



Cloud Volumes ONTAP for AWS

NetApp Automation

NetApp
November 18, 2025

Table of Contents

Cloud Volumes ONTAP for AWS	1
Cloud Volumes ONTAP for AWS - Burst to cloud.	1
Step 1: Install Docker and Docker Compose	1
Step 2: Prepare the Docker image	2
Step 3: Create environment variable files	4
Step 4: Sign up for NetApp Intelligent Services	4
Step 5: Create an external volume	5
Step 6: Deploy Cloud Volumes ONTAP for AWS	6

Cloud Volumes ONTAP for AWS

Cloud Volumes ONTAP for AWS - Burst to cloud

This article supports the NetApp Cloud Volumes ONTAP for AWS Automation Solution, which is available to NetApp customers from the NetApp Console automation hub.

The Cloud Volumes ONTAP for AWS Automation Solution automates the containerized deployment of Cloud Volumes ONTAP for AWS using Terraform, enabling you to deploy Cloud Volumes ONTAP for AWS rapidly, without any manual intervention.

Before you begin

- You must download the [Cloud Volumes ONTAP AWS - Burst to cloud](#) automation solution through the Console web UI. The solution is packaged as `cvo_aws_flexcache.zip`.
- You must install a Linux VM on the same network as Cloud Volumes ONTAP.
- After you install the Linux VM, you must follow the steps in this solution to install the required dependencies.

Step 1: Install Docker and Docker Compose

Install Docker

The following steps use Ubuntu 20.04 Debian Linux distribution software as an example. The commands you run depend on the Linux distribution software that you are using. Refer to the specific Linux distribution software documentation for your configuration.

Steps

1. Install Docker by running the following `sudo` commands:

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

2. Verify the installation:

```
docker -version
```

3. Verify that a group named "docker" has been created on your Linux system. If necessary, create the group:

```
sudo groupadd docker
```

4. Add the user that needs to access Docker to the group:

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

5. Your changes are applied after you log out and log back in to the terminal. Alternatively, you can apply the changes immediately:

```
newgrp docker
```

Install Docker Compose

Steps

1. Install Docker Compose by running the following `sudo` commands:

```
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. Verify the installation:

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

Step 2: Prepare the Docker image

Steps

1. Copy the `cvo_aws_flexcache.zip` folder to the Linux VM that you want to use to deploy Cloud Volumes ONTAP:

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

- `private-key.pem` is your private key file for login without a password.
- `awsuser` is the VM username.
- `IP_ADDRESS_OF_VM` is the VM IP address.
- `LOCATION_TO_BE_COPIED` is the location where the folder will be copied.

2. Extract the `cvo_aws_flexcache.zip` folder. You can extract the folder in the current directory or in a custom location.

To extract the folder in the current directory, run:

```
unzip cvo_aws_flexcache.zip
```

To extract the folder in a custom location, run:

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

3. After you extract the content, navigate to the `CVO_Aws_Deployment` folder and run the following command to view the files:

```
ls -la
```

You should see a list of files, similar to the following example:

```
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
```

4. Locate the `cvo_aws_flexcache_ubuntu_image.tar` file. This contains the Docker image required to deploy Cloud Volumes ONTAP for AWS.

5. Untar the file:

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

6. Wait a few minutes for the Docker image to load, and then validate that the Docker image loaded successfully:

```
docker images
```

You should see a Docker image named `cvo_aws_flexcache_ubuntu_image` with the `latest` tag, as shown in the following example:

REPOSITORY	TAG	IMAGE ID	CREATED
<code>cvo_aws_flexcache_ubuntu_image</code>	<code>latest</code>	<code>18db15a4d59c</code>	<code>2 weeks ago</code>
1.14GB			



You can change the Docker image name if required. If you change the Docker image name, make sure to update the Docker image name in the `docker-compose-deploy` and `docker-compose-destroy` files.

Step 3: Create environment variable files

At this stage, you must create two environment variable files. One file is for authentication of AWS Resource Manager APIs using the AWS access and secret keys. The second file is for setting environment variables to enable the Console Terraform modules to locate and authenticate AWS APIs.

Steps

1. Create the `awsauth.env` file in the following location:

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

- i. Add the following content to the `awsauth.env` file:

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

The format **must** be exactly as shown above.

2. Add the absolute file path to the `.env` file.

Enter the absolute path for the `awsauth.env` environment file that corresponds to the `AWS_CREDS` environment variable.

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

3. Navigate to the `cvo_aws_variable` folder and update the access and secret key in the credentials file.

Add the following content to the file:

```
aws_access_key_id=<>
aws_secret_access_key=<>
```

The format **must** be exactly as shown above.

Step 4: Sign up for NetApp Intelligent Services

Sign up for NetApp Intelligent Services through your cloud provider to pay hourly (PAYGO) or through an

annual contract. NetApp Intelligent Services include NetApp Backup and Recovery, Cloud Volumes ONTAP, NetApp Cloud Tiering, NetApp Ransomware Resilience and NetApp Disaster Recovery. NetApp Data Classification is included with your subscription at no additional cost.

Steps

1. From the Amazon Web Services (AWS) portal, navigate to **SaaS** and select **Subscribe to NetApp Intelligent Services**.

You can either use the same resource group as Cloud Volumes ONTAP or a different resource group.

2. Configure the NetApp Console portal to import the SaaS subscription to the Console.

You can configure this directly from the AWS portal.

You are redirected to the Console portal to confirm the configuration.

3. Confirm the configuration in the Console portal by selecting **Save**.

Step 5: Create an external volume

You should create an external volume to keep the Terraform state files, and other important files persistent. You must make sure that the files are available for Terraform to run the workflow and deployments.

Steps

1. Create an external volume outside of Docker Compose:

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

Example:

```
docker volume create cvo_aws_volume_dst
```

2. Use one of the following options:

- a. Add an external volume path to the `.env` environment file.

You must follow the exact format shown below.

Format:

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

Example:

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

- b. Add NFS shares as an external volume.

Make sure that the Docker container can communicate with the NFS shares and that the correct permissions, such as read/write, are configured.

- i. Add the NFS shares path as the path to the external volume in the Docker Compose file, as shown

below:

Format:

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

Example:

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

3. Navigate to the `cvo_aws_variables` folder.

You should see the following variable file in the folder:

- `terraform.tfvars`
- `variables.tf`

4. Change the values inside the `terraform.tfvars` file according to your requirements.

You must read the specific supporting documentation when modifying any of the variable values in the `terraform.tfvars` file. The values can vary depending on region, availability zones, and other factors supported by Cloud Volumes ONTAP for AWS. This includes licenses, disk size, and VM size for single nodes and high availability (HA) pairs.

All supporting variables for the Console agent and Cloud Volumes ONTAP Terraform modules are already defined in the `variables.tf` file. You must refer to the variable names in the `variables.tf` file before adding to the `terraform.tfvars` file.

5. Depending on your requirements, you can enable or disable FlexCache and FlexClone by setting the following options to `true` or `false`.

The following examples enable FlexCache and FlexClone:

- `is_flexcache_required = true`
- `is_flexclone_required = true`

Step 6: Deploy Cloud Volumes ONTAP for AWS

Use the following steps to deploy Cloud Volumes ONTAP for AWS.

Steps

1. From the root folder, run the following command to trigger deployment:

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

Two containers are triggered, the first container deploys Cloud Volumes ONTAP and the second container sends telemetry data to AutoSupport.

The second container waits until the first container completes all of the steps successfully.

2. Monitor progress of the deployment process using the log files:

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

This command provides output in real time and captures the data in the following log files:
deployment.log

telemetry_asup.log

You can change the name of these log files by editing the .env file using the following environment variables:

DEPLOYMENT_LOGS

TELEMETRY_ASUP_LOGS

The following examples show how to change the log file names:

DEPLOYMENT_LOGS=<your_deployment_log_filename>.log

TELEMETRY_ASUP_LOGS=<your_telemetry_asup_log_filename>.log

After you finish

You can use the following steps to remove the temporary environment and clean up items that were created during the deployment process.

Steps

1. If you deployed FlexCache, set the following option in the `terraform.tfvars` variable file, this cleans up FlexCache volumes and removes the temporary environment that was created earlier.

`flexcache_operation = "destroy"`



The possible options are `deploy` and `destroy`.

2. If you deployed FlexClone, set the following option in the `terraform.tfvars` variable file, this cleans up FlexClone volumes and removes the temporary environment that was created earlier.

`flexclone_operation = "destroy"`



The possible options are `deploy` and `destroy`.

Copyright information

Copyright © 2025 NetApp, Inc. All Rights Reserved. Printed in the U.S. No part of this document covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—without prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP “AS IS” AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

LIMITED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (b)(3) of the Rights in Technical Data -Noncommercial Items at DFARS 252.227-7013 (FEB 2014) and FAR 52.227-19 (DEC 2007).

Data contained herein pertains to a commercial product and/or commercial service (as defined in FAR 2.101) and is proprietary to NetApp, Inc. All NetApp technical data and computer software provided under this Agreement is commercial in nature and developed solely at private expense. The U.S. Government has a non-exclusive, non-transferrable, nonsublicensable, worldwide, limited irrevocable license to use the Data only in connection with and in support of the U.S. Government contract under which the Data was delivered. Except as provided herein, the Data may not be used, disclosed, reproduced, modified, performed, or displayed without the prior written approval of NetApp, Inc. United States Government license rights for the Department of Defense are limited to those rights identified in DFARS clause 252.227-7015(b) (FEB 2014).

Trademark information

NETAPP, the NETAPP logo, and the marks listed at <http://www.netapp.com/TM> are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners.