



# **Solution Automation**

## **NetApp Solutions**

NetApp  
April 11, 2023

This PDF was generated from [https://docs.netapp.com/us-en/netapp-solutions/automation/automation\\_introduction.html](https://docs.netapp.com/us-en/netapp-solutions/automation/automation_introduction.html) on April 11, 2023. Always check docs.netapp.com for the latest.

# Table of Contents

- Solution Automation . . . . . 1
  - NetApp Solution Automation . . . . . 1
    - Getting Started with NetApp solution automation . . . . . 1
    - NetApp Solution Automation . . . . . 6
    - Cloud Volumes Automation via Terraform . . . . . 9
    - FSx for ONTAP Monitoring and Auto-Resizing using AWS Lambda Function . . . . . 55

# Solution Automation

## NetApp Solution Automation

### Introduction

In providing solutions to meet today's business challenges, NetApp delivers solutions with the following goals:

- Providing validated deployment and configuration steps,
- Providing solutions that are easily consumable,
- Providing solution deployment that has a predictable outcome, is easily repeated, and scalable across a customer's enterprise.

In order to achieve these goals, it is paramount that the deployment and configuration of infrastructure and/or applications delivered through our solutions is simplified through automation. NetApp is committed to simplifying solution consumption through automation.

Utilizing open-source automation tools such as Red Hat Ansible, HashiCorp Terraform, or Microsoft Powershell, NetApp solutions have the ability to automate application deployment, cloud provisioning, configuration management, and many other common IT tasks. NetApp's solutions take advantage of publicly available automation artifacts - as well as providing NetApp authored automation - to simplify the overall deployment of a solution.

Where automation capabilities are available, the solution collateral will guide the user through the process for automating the solution or solution steps via the specific automation tool(s).

## Getting Started with NetApp solution automation

NetApp solution automation provides simplicity and repeatability for many of the common tasks utilized by the NetApp Solutions.

Prior to running any solution automation, the environment must be configured for how the automation will be executed. There are options to run the automation from the command line or through a tool such as AWX or tower.

The following sections will outline the steps required to configure the environment for each of the specified environments.

## Setup the Ansible Control Node for CLI deployments on RHEL / CentOS

1. Requirements for the Ansible control node,:
  - a. A RHEL/CentOS machine with the following packages installed:
    - i. Python3
    - ii. Pip3
    - iii. Ansible (version greater than 2.10.0)
    - iv. Git

If you have a fresh RHEL/CentOS machine without the above requirements installed, follow the below steps to setup that machine as the Ansible control node:

1. Enable the Ansible repository for RHEL-8/RHEL-7
  - a. For RHEL-8 (run the below command as root)

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

- b. For RHEL-7 (run the below command as root)

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

2. Create a .sh file

```
vi setup.sh
```

3. Paste the below content in the file

```
#!/bin/bash  
echo "Installing Python ----->"  
sudo yum -y install python3 >/dev/null  
echo "Installing Python Pip ----->"  
sudo yum -y install python3-pip >/dev/null  
echo "Installing Ansible ----->"  
python3 -W ignore -m pip --disable-pip-version-check install ansible  
>/dev/null  
echo "Installing git ----->"  
sudo yum -y install git >/dev/null
```

4. Make the file executable

```
chmod +x setup.sh
```

5. Run the script (as root)

```
./setup.sh
```

## Setup the Ansible Control Node for CLI deployments on Ubuntu / Debian

1. Requirements for the Ansible control node,:
  - a. A Ubuntu/Debian machine with the following packages installed:
    - i. Python3
    - ii. Pip3
    - iii. Ansible (version greater than 2.10.0)
    - iv. Git

If you have a fresh Ubuntu/Debian machine without the above requirements installed, follow the below steps to setup that machine as the Ansible control node:

1. Create a .sh file

```
vi setup.sh
```

2. Paste the below content in the file

```
#!/bin/bash
echo "Installing Python ----->"
sudo apt-get -y install python3 >/dev/null
echo "Installing Python Pip ----->"
sudo apt-get -y install python3-pip >/dev/null
echo "Installing Ansible ----->"
python3 -W ignore -m pip --disable-pip-version-check install ansible
>/dev/null
echo "Installing git ----->"
sudo apt-get -y install git >/dev/null
```

3. Make the file executable

```
chmod +x setup.sh
```

4. Run the script (as root)

```
./setup.sh
```

## Setup Ansible Tower or AWX for Tower / AWX deployments

This section describes the steps required to configure the parameters in AWX/Ansible Tower that prepare the environment for consuming NetApp automated solutions.

1. Configure the inventory.
  - a. Navigate to Resources → Inventories → Add and click Add Inventory.
  - b. Provide name and organization details and click Save.
  - c. In the Inventories page, click the inventory resources you just created.
  - d. If there are any inventory variables, paste them into the variables field.
  - e. Go to the Groups sub-menu and click Add.
  - f. Provide the name of the group, copy in the group variables (if necessary), and click Save.
  - g. Click the group created, go to the Hosts sub-menu and click Add New Host.
  - h. Provide the hostname and IP address of the host, paste in the host variables (if necessary), and click Save.
2. Create credential types. For solutions involving ONTAP, Element, VMware, or any other HTTPS-based transport connection, you must configure the credential type to match the username and password entries.
  - a. Navigate to Administration → Credential Types and click Add.
  - b. Provide the name and description.
  - c. Paste the following content into the Input Configuration:

```
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
```

- a. Paste the following content into the Injector Configuration:

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

1. Configure credentials.
  - a. Navigate to Resources → Credentials and click Add.
  - b. Enter the name and organization details.
  - c. Select the correct credential type; if you intend to use the standard SSH login, select the type Machine or alternatively select the custom credential type that you created.
  - d. Enter the other corresponding details and click Save.
2. Configure the project.
  - a. Navigate to Resources → Projects and click Add.
  - b. Enter the name and organization details.
  - c. Select Git for the Source Control Credential Type.
  - d. Paste the source control URL (or git clone URL) corresponding to the specific solution.
  - e. Optionally, if the Git URL is access controlled, create and attach the corresponding credential in Source Control Credential.
  - f. Click Save.
3. Configure the job template.
  - a. Navigate to Resources → Templates → Add and click Add Job Template.
  - b. Enter the name and description.
  - c. Select the Job type; Run configures the system based on a playbook and Check performs a dry run of the playbook without actually configuring the system.
  - d. Select the corresponding inventory, project, and credentials for the playbook.
  - e. Select the playbook that you would like to run as a part of the job template.
  - f. Usually the variables are pasted during runtime. Therefore, to get the prompt to populate the variables during runtime, make sure to tick the checkbox Prompt on Launch corresponding to the Variable field.
  - g. Provide any other details as required and click Save.
4. Launch the job template.
  - a. Navigate to Resources → Templates.
  - b. Click the desired template and then click Launch.
  - c. Fill in any variables if prompted on launch and then click Launch again.

## NetApp Solution Automation

### AWS Authentication Requirements for CVO and Connector Using NetApp Cloud Manager

To configure automated Deployments of CVO and Connectors using Ansible playbooks via AWX/Ansible Tower, the following information is needed:

#### Acquiring Access/Secret Keys from AWS

1. To deploy CVO and Connector in Cloud Manager, we need AWS Access/Secret Key. Acquire the keys in AWS console by launching IAM→Users→your username→security credentials→Create Access key.



2. Copy access keys and keep them secured to use in Connector and CVO deployment.



If you lose your key, you can create another access key and delete the one you lost



## Acquiring Refresh Token from NetApp Cloud Central

1. Login into your cloud central account using your account credentials at <https://services.cloud.netapp.com/refresh-token>
2. Generate a refresh Token and save it for deployments.

### 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

## Acquiring Client ID

1. Access the API page to copy Client ID at <https://services.cloud.netapp.com/developer-hub>.
2. Click on "learn How to Authenticate", in the top right corner.
3. From the Authentication window that pops up, copy the Client ID from Regular Access if you require a username/password to login. Federated users with SSO should copy the client ID from the "Refresh Token Tab".

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

## Acquiring Key Pair from AWS

1. In AWS console, search for “Key Pair” and create a key pair with “pem”. Remember the name of you key\_pair, we will use it to deploy the connector.

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.

## Acquiring Account ID

1. In Cloud Manager, click on Account → Manage Accounts and then copy the account id for use in variables for AWX.



## Cloud Volumes Automation via Terraform

This solution documents the automated deployments of Cloud Volumes on AWS (CVO Single Node, CVO HA and FSX ONTAP) and Azure (CVO Single Node, CVO HA and ANF) using Terraform modules. The code can be found at [https://github.com/NetApp/NetApp-Automation/na\\_cloud\\_volumes\\_automation](https://github.com/NetApp/NetApp-Automation/na_cloud_volumes_automation)

### Pre-requisites

1. Terraform  $\geq$  0.13
2. Cloud Manager Account
3. Cloud Provider Account – AWS, Azure
4. Host machine (any OS supported by Terraform)

### Provider documentation

The documentation of Terraform provider for Cloud Manager is available at: <https://registry.terraform.io/providers/NetApp/netapp-cloudmanager/latest/docs>

### Controlling the provider version

Note that you can also control the provider version. This is controlled by a `required_providers` block in your Terraform configuration.

The syntax is as follows:

```
terraform {
  required_providers {
    netapp-cloudmanager = {
      source = "NetApp/netapp-cloudmanager"
      version = "20.10.0"
    }
  }
}
```

Read more on provider version control.

## **Running Specific Modules**



## CVO Single Node Deployment

### Terraform configuration files for deployment of NetApp CVO (Single Node Instance) on AWS

This section contains various Terraform configuration files to deploy/configure single node NetApp CVO (Cloud Volumes ONTAP) on AWS (Amazon Web Services).

Terraform Documentation: <https://registry.terraform.io/providers/NetApp/netapp-cloudmanager/latest/docs>

#### Procedure

In order to run the template:

- a. Clone the repository.

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

- b. Navigate to the desired folder

```
cd na_cloud_volumes_automation/
```

- c. Configure AWS credentials from the CLI.

```
aws configure
```

- AWS Access Key ID [None]: accesskey
- AWS Secret Access Key [None]: secretkey
- Default region name [None]: us-west-2
- Default output format [None]: json

- d. Update the variable values in `vars/aws_cvo_single_node_deployment.tfvar`



You can choose to deploy the connector by setting the variable "aws\_connector\_deploy\_bool" value to true/false.

- e. Initialize the Terraform repository to install all the pre-requisites and prepare for deployment.

```
terraform init
```

- f. Verify the terraform files using terraform validate command.

```
terraform validate
```

- g. Make a dry run of the configuration to get a preview of all the changes expected by the deployment.

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

- h. Run the deployment

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

To delete the deployment

```
terraform destroy
```

### Recipies:

Connector

Terraform variables for NetApp AWS connector instance for CVO deployment.

Name	Type	Description
<b>aws_connector_deploy_bool</b>	Bool	(Required) Check for Connector deployment.
<b>aws_connector_name</b>	String	(Required) The name of the Cloud Manager Connector.
<b>aws_connector_region</b>	String	(Required) The region where the Cloud Manager Connector will be created.
<b>aws_connector_key_name</b>	String	(Required) The name of the key pair to use for the Connector instance.
<b>aws_connector_company</b>	String	(Required) The name of the company of the user.
<b>aws_connector_instance_type</b>	String	(Required) The type of instance (for example, t3.xlarge). At least 4 CPU and 16 GB of memory are required.
<b>aws_connector_subnet_id</b>	String	(Required) The ID of the subnet for the instance.
<b>aws_connector_security_group_id</b>	String	(Required) The ID of the security group for the instance, multiple security groups can be provided separated by ','.

<b>aws_connector_iam_instance_profile_name</b>	String	(Required) The name of the instance profile for the Connector.
<b>aws_connector_account_id</b>	String	(Optional) The NetApp account ID that the Connector will be associated with. If not provided, Cloud Manager uses the first account. If no account exists, Cloud Manager creates a new account. You can find the account ID in the account tab of Cloud Manager at <a href="https://cloudmanager.netapp.com">https://cloudmanager.netapp.com</a> .
<b>aws_connector_public_ip_bootstrap</b>	Bool	(Optional) Indicates whether to associate a public IP address to the instance. If not provided, the association will be done based on the subnet's configuration.

#### Single Node Instance

Terraform variables for single NetApp CVO instance.

Name	Type	Description
<b>cvo_name</b>	String	(Required) The name of the Cloud Volumes ONTAP working environment.
<b>cvo_region</b>	String	(Required) The region where the working environment will be created.
<b>cvo_subnet_id</b>	String	(Required) The subnet id where the working environment will be created.
<b>cvo_vpc_id</b>	String	(Optional) The VPC ID where the working environment will be created. If this argument isn't provided, the VPC will be calculated by using the provided subnet ID.
<b>cvo_svm_password</b>	String	(Required) The admin password for Cloud Volumes ONTAP.
<b>cvo_writing_speed_state</b>	String	(Optional) The write speed setting for Cloud Volumes ONTAP: ['NORMAL','HIGH']. The default is 'NORMAL'.

#### CVO HA Deployment



## Terraform configuration files for deployment of NetApp CVO (HA Pair) on AWS

This section contains various Terraform configuration files to deploy/configure NetApp CVO (Cloud Volumes ONTAP) in high availability pair on AWS (Amazon Web Services).

Terraform Documentation: <https://registry.terraform.io/providers/NetApp/netapp-cloudmanager/latest/docs>

### Procedure

In order to run the template:

- a. Clone the repository.

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

- b. Navigate to the desired folder

```
cd na_cloud_volumes_automation/
```

- c. Configure AWS credentials from the CLI.

```
aws configure
```

- AWS Access Key ID [None]: accesskey
- AWS Secret Access Key [None]: secretkey
- Default region name [None]: us-west-2
- Default output format [None]: json

- d. Update the variable values in `vars/aws_cvo_ha_deployment.tfvars`.



You can choose to deploy the connector by setting the variable "aws\_connector\_deploy\_bool" value to true/false.

- e. Initialize the Terraform repository to install all the pre-requisites and prepare for deployment.

```
terraform init
```

- f. Verify the terraform files using terraform validate command.

```
terraform validate
```

- g. Make a dry run of the configuration to get a preview of all the changes expected by the deployment.

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

h. Run the deployment

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

To delete the deployment

```
terraform destroy
```

### Recipies:

Connector

Terraform variables for NetApp AWS connector instance for CVO deployment.

Name	Type	Description
<b>aws_connector_deploy_bool</b>	Bool	(Required) Check for Connector deployment.
<b>aws_connector_name</b>	String	(Required) The name of the Cloud Manager Connector.
<b>aws_connector_region</b>	String	(Required) The region where the Cloud Manager Connector will be created.
<b>aws_connector_key_name</b>	String	(Required) The name of the key pair to use for the Connector instance.
<b>aws_connector_company</b>	String	(Required) The name of the company of the user.
<b>aws_connector_instance_type</b>	String	(Required) The type of instance (for example, t3.xlarge). At least 4 CPU and 16 GB of memory are required.
<b>aws_connector_subnet_id</b>	String	(Required) The ID of the subnet for the instance.
<b>aws_connector_security_group_id</b>	String	(Required) The ID of the security group for the instance, multiple security groups can be provided separated by ','.

<b>aws_connector_iam_instance_profile_name</b>	String	(Required) The name of the instance profile for the Connector.
<b>aws_connector_account_id</b>	String	(Optional) The NetApp account ID that the Connector will be associated with. If not provided, Cloud Manager uses the first account. If no account exists, Cloud Manager creates a new account. You can find the account ID in the account tab of Cloud Manager at <a href="https://cloudmanager.netapp.com">https://cloudmanager.netapp.com</a> .
<b>aws_connector_public_ip_bootstrap</b>	Bool	(Optional) Indicates whether to associate a public IP address to the instance. If not provided, the association will be done based on the subnet's configuration.

HA Pair

Terraform variables for NetApp CVO instances in HA Pair.

Name	Type	Description
<b>cvo_is_ha</b>	Bool	(Optional) Indicate whether the working environment is an HA pair or not [true, false]. The default is false.
<b>cvo_name</b>	String	(Required) The name of the Cloud Volumes ONTAP working environment.
<b>cvo_region</b>	String	(Required) The region where the working environment will be created.
<b>cvo_node1_subnet_id</b>	String	(Required) The subnet id where the first node will be created.
<b>cvo_node2_subnet_id</b>	String	(Required) The subnet id where the second node will be created.
<b>cvo_vpc_id</b>	String	(Optional) The VPC ID where the working environment will be created. If this argument isn't provided, the VPC will be calculated by using the provided subnet ID.
<b>cvo_svm_password</b>	String	(Required) The admin password for Cloud Volumes ONTAP.

<b>cvo_failover_mode</b>	String	(Optional) For HA, the failover mode for the HA pair: ['PrivateIP', 'FloatingIP']. 'PrivateIP' is for a single availability zone and 'FloatingIP' is for multiple availability zones.
<b>cvo_mediator_subnet_id</b>	String	(Optional) For HA, the subnet ID of the mediator.
<b>cvo_mediator_key_pair_name</b>	String	(Optional) For HA, the key pair name for the mediator instance.
<b>cvo_cluster_floating_ip</b>	String	(Optional) For HA FloatingIP, the cluster management floating IP address.
<b>cvo_data_floating_ip</b>	String	(Optional) For HA FloatingIP, the data floating IP address.
<b>cvo_data_floating_ip2</b>	String	(Optional) For HA FloatingIP, the data floating IP address.
<b>cvo_svm_floating_ip</b>	String	(Optional) For HA FloatingIP, the SVM management floating IP address.
<b>cvo_route_table_ids</b>	List	(Optional) For HA FloatingIP, the list of route table IDs that will be updated with the floating IPs.

## FSx Deployment

### Terraform configuration files for deployment of NetApp ONTAP FSx on AWS

This section contains various Terraform configuration files to deploy/configure NetApp ONTAP FSx on AWS (Amazon Web Services).

Terraform Documentation: <https://registry.terraform.io/providers/NetApp/netapp-cloudmanager/latest/docs>

### Procedure

In order to run the template:

- a. Clone the repository.

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

- b. Navigate to the desired folder

```
cd na_cloud_volumes_automation/
```

c. Configure AWS credentials from the CLI.

```
aws configure
```

- AWS Access Key ID [None]: accesskey
- AWS Secret Access Key [None]: secretkey
- Default region name [None]: us-west-2
- Default output format [None]:

d. Update the variable values in `vars/aws_fsx_deployment.tfvars`



You can choose to deploy the connector by setting the variable "aws\_connector\_deploy\_bool" value to true/false.

e. Initialize the Terraform repository to install all the pre-requisites and prepare for deployment.

```
terraform init
```

f. Verify the terraform files using terraform validate command.

```
terraform validate
```

g. Make a dry run of the configuration to get a preview of all the changes expected by the deployment.

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

h. Run the deployment

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

To delete the deployment

```
terraform destroy
```

**Recipes:**

Connector

Terraform variables for NetApp AWS connector instance.

Name	Type	Description
<b>aws_connector_deploy_bool</b>	Bool	(Required) Check for Connector deployment.
<b>aws_connector_name</b>	String	(Required) The name of the Cloud Manager Connector.
<b>aws_connector_region</b>	String	(Required) The region where the Cloud Manager Connector will be created.
<b>aws_connector_key_name</b>	String	(Required) The name of the key pair to use for the Connector instance.
<b>aws_connector_company</b>	String	(Required) The name of the company of the user.
<b>aws_connector_instance_type</b>	String	(Required) The type of instance (for example, t3.xlarge). At least 4 CPU and 16 GB of memory are required.
<b>aws_connector_subnet_id</b>	String	(Required) The ID of the subnet for the instance.
<b>aws_connector_security_group_id</b>	String	(Required) The ID of the security group for the instance, multiple security groups can be provided separated by ','.
<b>aws_connector_iam_instance_profile_name</b>	String	(Required) The name of the instance profile for the Connector.
<b>aws_connector_account_id</b>	String	(Optional) The NetApp account ID that the Connector will be associated with. If not provided, Cloud Manager uses the first account. If no account exists, Cloud Manager creates a new account. You can find the account ID in the account tab of Cloud Manager at <a href="https://cloudmanager.netapp.com">https://cloudmanager.netapp.com</a> .
<b>aws_connector_public_ip_bool</b>	Bool	(Optional) Indicates whether to associate a public IP address to the instance. If not provided, the association will be done based on the subnet's configuration.

FSx Instance

Terraform variables for NetApp ONTAP FSx instance.

Name	Type	Description
<b>fsx_name</b>	String	(Required) The name of the Cloud Volumes ONTAP working environment.
<b>fsx_region</b>	String	(Required) The region where the working environment will be created.
<b>fsx_primary_subnet_id</b>	String	(Required) The primary subnet id where the working environment will be created.
<b>fsx_secondary_subnet_id</b>	String	(Required) The secondary subnet id where the working environment will be created.
<b>fsx_account_id</b>	String	(Required) The NetApp account ID that the FSx instance will be associated with. If not provided, Cloud Manager uses the first account. If no account exists, Cloud Manager creates a new account. You can find the account ID in the account tab of Cloud Manager at <a href="https://cloudmanager.netapp.com">https://cloudmanager.netapp.com</a> .
<b>fsx_workspace_id</b>	String	(Required) The ID of the Cloud Manager workspace of working environment.
<b>fsx_admin_password</b>	String	(Required) The admin password for Cloud Volumes ONTAP.
<b>fsx_throughput_capacity</b>	String	(Optional) capacity of the throughput.
<b>fsx_storage_capacity_size</b>	String	(Optional) EBS volume size for the first data aggregate. For GB, the unit can be: [100 or 500]. For TB, the unit can be: [1,2,4,8,16]. The default is '1'
<b>fsx_storage_capacity_size_unit</b>	String	(Optional) ['GB' or 'TB']. The default is 'TB'.
<b>fsx_cloudmanager_aws_credential_name</b>	String	(Required) The name of the AWS Credentials account name.





## ANF

### Terraform configuration files for deployment of ANF Volume on Azure

This section contains various Terraform configuration files to deploy/configure ANF (Azure Netapp Files) Volume on Azure.

Terraform Documentation: <https://registry.terraform.io/providers/hashicorp/azurerm/latest/docs>

#### Procedure

In order to run the template:

- a. Clone the repository.

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

- b. Navigate to the desired folder

```
cd na_cloud_volumes_automation
```

- c. Login to your Azure CLI (Azure CLI must be installed).

```
az login
```

- d. Update the variable values in `vars/azure_anf.tfvars`.



You can choose to deploy the ANF volume using an existing vnet and subnet by setting the variable "vnet\_creation\_bool" and "subnet\_creation\_bool" value to false and supplying the "subnet\_id\_for\_anf\_vol". You can also set those values to true and create a new vnet and subnet in which case, the subnet ID will automatically be taken from the newly created subnet.

- e. Initialize the Terraform repository to install all the pre-requisites and prepare for deployment.

```
terraform init
```

- f. Verify the terraform files using terraform validate command.

```
terraform validate
```

- g. Make a dry run of the configuration to get a preview of all the changes expected by the deployment.

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

#### h. Run the deployment

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

To delete the deployment

```
terraform destroy
```

#### Recipies:

Single Node Instance

Terraform variables for single NetApp ANF Volume.

Name	Type	Description
<b>az_location</b>	String	(Required) Specifies the supported Azure location where the resource exists. Changing this forces a new resource to be created.
<b>az_prefix</b>	String	(Required) The name of the resource group where the NetApp Volume should be created. Changing this forces a new resource to be created.
<b>az_vnet_address_space</b>	String	(Required) The address space to be used by the newly created vnet for ANF volume deployment.
<b>az_subnet_address_prefix</b>	String	(Required) The subnet address prefix to be used by the newly created vnet for ANF volume deployment.
<b>az_volume_path</b>	String	(Required) A unique file path for the volume. Used when creating mount targets. Changing this forces a new resource to be created.
<b>az_capacity_pool_size</b>	Integer	(Required) Capacity Pool Size mentioned in TB.

<b>az_vnet_creation_bool</b>	Boolean	(Required) Set this boolean to <code>true</code> if you want to create a new vnet. Set it to <code>false</code> to use an existing vnet.
<b>az_subnet_creation_bool</b>	Boolean	(Required) Set this boolean to <code>true</code> to create a new subnet. Set it to <code>false</code> to use an existing subnet.
<b>az_subnet_id_for_anf_vol</b>	String	(Required) Mention the subnet id in case you decide to use an existing subnet by setting <code>subnet_creation_bool</code> to <code>true</code> . If set to <code>false</code> , leave it at the default value.
<b>az_netapp_pool_service_level</b>	String	(Required) The target performance of the file system. Valid values include <code>Premium</code> , <code>Standard</code> , or <code>Ultra</code> .
<b>az_netapp_vol_service_level</b>	String	(Required) The target performance of the file system. Valid values include <code>Premium</code> , <code>Standard</code> , or <code>Ultra</code> .
<b>az_netapp_vol_protocol</b>	String	(Optional) The target volume protocol expressed as a list. Supported single value include <code>CIFS</code> , <code>NFSv3</code> , or <code>NFSv4.1</code> . If argument is not defined it will default to <code>NFSv3</code> . Changing this forces a new resource to be created and data will be lost.
<b>az_netapp_vol_security_style</b>	String	(Optional) Volume security style, accepted values are <code>Unix</code> or <code>Ntfs</code> . If not provided, single-protocol volume is created defaulting to <code>Unix</code> if it is <code>NFSv3</code> or <code>NFSv4.1</code> volume, if <code>CIFS</code> , it will default to <code>Ntfs</code> . In a dual-protocol volume, if not provided, its value will be <code>Ntfs</code> .
<b>az_netapp_vol_storage_quota</b>	String	(Required) The maximum Storage Quota allowed for a file system in Gigabytes.

## ANF Data Protection

### Terraform configuration files for deployment of ANF Volume with Data Protection on Azure

This section contains various Terraform configuration files to deploy/configure ANF (Azure Netapp Files) Volume with Data Protection on Azure.

Terraform Documentation: <https://registry.terraform.io/providers/hashicorp/azurerm/latest/docs>

## Procedure

In order to run the template:

- a. Clone the repository.

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

- b. Navigate to the desired folder

```
cd na_cloud_volumes_automation
```

- c. Login to your Azure CLI (Azure CLI must be installed).

```
az login
```

- d. Update the variable values in vars/azure\_anf\_data\_protection.tfvars.



You can choose to deploy the ANF volume using an existing vnet and subnet by setting the variable "vnet\_creation\_bool" and "subnet\_creation\_bool" value to false and supplying the "subnet\_id\_for\_anf\_vol". You can also set those values to true and create a new vnet and subnet in which case, the subnet ID will automatically be taken from the newly created subnet.

- e. Initialize the Terraform repository to install all the pre-requisites and prepare for deployment.

```
terraform init
```

- f. Verify the terraform files using terraform validate command.

```
terraform validate
```

- g. Make a dry run of the configuration to get a preview of all the changes expected by the deployment.

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

- h. Run the deployment

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

To delete the deployment

```
terraform destroy
```

### Recipies:

ANF Data Protection

Terraform variables for single ANF Volume with data protection enabled.

Name	Type	Description
<b>az_location</b>	String	(Required) Specifies the supported Azure location where the resource exists. Changing this forces a new resource to be created.
<b>az_alt_location</b>	String	(Required) The Azure location where the secondary volume will be created
<b>az_prefix</b>	String	(Required) The name of the resource group where the NetApp Volume should be created. Changing this forces a new resource to be created.
<b>az_vnet_primary_address_space</b>	String	(Required) The address space to be used by the newly created vnet for ANF primary volume deployment.
<b>az_vnet_secondary_address_space</b>	String	(Required) The address space to be used by the newly created vnet for ANF secondary volume deployment.
<b>az_subnet_primary_address_prefix</b>	String	(Required) The subnet address prefix to be used by the newly created vnet for ANF primary volume deployment.
<b>az_subnet_secondary_addresses_prefix</b>	String	(Required) The subnet address prefix to be used by the newly created vnet for ANF secondary volume deployment.

<b>az_volume_path_primary</b>	String	(Required) A unique file path for the primary volume. Used when creating mount targets. Changing this forces a new resource to be created.
<b>az_volume_path_secondary</b>	String	(Required) A unique file path for the secondary volume. Used when creating mount targets. Changing this forces a new resource to be created.
<b>az_capacity_pool_size_primary</b>	Integer	(Required) Capacity Pool Size mentioned in TB.
<b>az_capacity_pool_size_secondary</b>	Integer	(Required) Capacity Pool Size mentioned in TB.
<b>az_vnet_primary_creation_bool</b>	Boolean	(Required) Set this boolean to <code>true</code> if you want to create a new vnet for primary volume. Set it to <code>false</code> to use an existing vnet.
<b>az_vnet_secondary_creation_bool</b>	Boolean	(Required) Set this boolean to <code>true</code> if you want to create a new vnet for secondary volume. Set it to <code>false</code> to use an existing vnet.
<b>az_subnet_primary_creation_bool</b>	Boolean	(Required) Set this boolean to <code>true</code> to create a new subnet for primary volume. Set it to <code>false</code> to use an existing subnet.
<b>az_subnet_secondary_creation_bool</b>	Boolean	(Required) Set this boolean to <code>true</code> to create a new subnet for secondary volume. Set it to <code>false</code> to use an existing subnet.
<b>az_primary_subnet_id_for_anf_vol</b>	String	(Required) Mention the subnet id in case you decide to use an existing subnet by setting <code>subnet_primary_creation_bool</code> to <code>true</code> . If set to <code>false</code> , leave it at the default value.
<b>az_secondary_subnet_id_for_anf_vol</b>	String	(Required) Mention the subnet id in case you decide to use an existing subnet by setting <code>subnet_secondary_creation_bool</code> to <code>true</code> . If set to <code>false</code> , leave it at the default value.

<b>az_netapp_pool_service_level_primary</b>	String	(Required) The target performance of the file system. Valid values include Premium , Standard , or Ultra.
<b>az_netapp_pool_service_level_secondary</b>	String	(Required) The target performance of the file system. Valid values include Premium , Standard , or Ultra.
<b>az_netapp_vol_service_level_primary</b>	String	(Required) The target performance of the file system. Valid values include Premium , Standard , or Ultra.
<b>az_netapp_vol_service_level_secondary</b>	String	(Required) The target performance of the file system. Valid values include Premium , Standard , or Ultra.
<b>az_netapp_vol_protocol_primary</b>	String	(Optional) The target volume protocol expressed as a list. Supported single value include CIFS, NFSv3, or NFSv4 . 1. If argument is not defined it will default to NFSv3. Changing this forces a new resource to be created and data will be lost.
<b>az_netapp_vol_protocol_secondary</b>	String	(Optional) The target volume protocol expressed as a list. Supported single value include CIFS, NFSv3, or NFSv4 . 1. If argument is not defined it will default to NFSv3. Changing this forces a new resource to be created and data will be lost.
<b>az_netapp_vol_storage_quota_primary</b>	String	(Required) The maximum Storage Quota allowed for a file system in Gigabytes.
<b>az_netapp_vol_storage_quota_secondary</b>	String	(Required) The maximum Storage Quota allowed for a file system in Gigabytes.
<b>az_dp_replication_frequency</b>	String	(Required) Replication frequency, supported values are 10minutes, hourly, daily, values are case sensitive.

## ANF Dual Protocol

### Terraform configuration files for deployment of ANF Volume with dual protocol on Azure

This section contains various Terraform configuration files to deploy/configure ANF (Azure Netapp Files) Volume with dual protocol enabled on Azure.

## Procedure

In order to run the template:

- a. Clone the repository.

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

- b. Navigate to the desired folder

```
cd na_cloud_volumes_automation
```

- c. Login to your Azure CLI (Azure CLI must be installed).

```
az login
```

- d. Update the variable values in vars/azure\_anf\_dual\_protocol.tfvars.



You can choose to deploy the ANF volume using an existing vnet and subnet by setting the variable "vnet\_creation\_bool" and "subnet\_creation\_bool" value to false and supplying the "subnet\_id\_for\_anf\_vol". You can also set those values to true and create a new vnet and subnet in which case, the subnet ID will automatically be taken from the newly created subnet.

- e. Initialize the Terraform repository to install all the pre-requisites and prepare for deployment.

```
terraform init
```

- f. Verify the terraform files using terraform validate command.

```
terraform validate
```

- g. Make a dry run of the configuration to get a preview of all the changes expected by the deployment.

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

- h. Run the deployment



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

To delete the deployment

```
terraform destroy
```

### Recipies:

Single Node Instance

Terraform variables for single ANF Volume with dual protocol enabled.

Name	Type	Description
<b>az_location</b>	String	(Required) Specifies the supported Azure location where the resource exists. Changing this forces a new resource to be created.
<b>az_prefix</b>	String	(Required) The name of the resource group where the NetApp Volume should be created. Changing this forces a new resource to be created.
<b>az_vnet_address_space</b>	String	(Required) The address space to be used by the newly created vnet for ANF volume deployment.
<b>az_subnet_address_prefix</b>	String	(Required) The subnet address prefix to be used by the newly created vnet for ANF volume deployment.
<b>az_volume_path</b>	String	(Required) A unique file path for the volume. Used when creating mount targets. Changing this forces a new resource to be created.
<b>az_capacity_pool_size</b>	Integer	(Required) Capacity Pool Size mentioned in TB.
<b>az_vnet_creation_bool</b>	Boolean	(Required) Set this boolean to <code>true</code> if you want to create a new vnet. Set it to <code>false</code> to use an existing vnet.

<b>az_subnet_creation_bool</b>	Boolean	(Required) Set this boolean to <code>true</code> to create a new subnet. Set it to <code>false</code> to use an existing subnet.
<b>az_subnet_id_for_anf_vol</b>	String	(Required) Mention the subnet id in case you decide to use an existing subnet by setting <code>subnet_creation_bool</code> to <code>true</code> . If set to <code>false</code> , leave it at the default value.
<b>az_netapp_pool_service_level</b>	String	(Required) The target performance of the file system. Valid values include <code>Premium</code> , <code>Standard</code> , or <code>Ultra</code> .
<b>az_netapp_vol_service_level</b>	String	(Required) The target performance of the file system. Valid values include <code>Premium</code> , <code>Standard</code> , or <code>Ultra</code> .
<b>az_netapp_vol_protocol1</b>	String	(Required) The target volume protocol expressed as a list. Supported single value include <code>CIFS</code> , <code>NFSv3</code> , or <code>NFSv4.1</code> . If argument is not defined it will default to <code>NFSv3</code> . Changing this forces a new resource to be created and data will be lost.
<b>az_netapp_vol_protocol2</b>	String	(Required) The target volume protocol expressed as a list. Supported single value include <code>CIFS</code> , <code>NFSv3</code> , or <code>NFSv4.1</code> . If argument is not defined it will default to <code>NFSv3</code> . Changing this forces a new resource to be created and data will be lost.
<b>az_netapp_vol_storage_quota</b>	String	(Required) The maximum Storage Quota allowed for a file system in Gigabytes.
<b>az_smb_server_username</b>	String	(Required) Username to create ActiveDirectory object.
<b>az_smb_server_password</b>	String	(Required) User Password to create ActiveDirectory object.
<b>az_smb_server_name</b>	String	(Required) Server Name to create ActiveDirectory object.
<b>az_smb_dns_servers</b>	String	(Required) DNS Server IP to create ActiveDirectory object.

#### ANF Volume From Snapshot

## Terraform configuration files for deployment of ANF Volume from Snapshot on Azure

This section contains various Terraform configuration files to deploy/configure ANF (Azure Netapp Files) Volume from Snapshot on Azure.

Terraform Documentation: <https://registry.terraform.io/providers/hashicorp/azurerm/latest/docs>

### Procedure

In order to run the template:

- a. Clone the repository.

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

- b. Navigate to the desired folder

```
cd na_cloud_volumes_automation
```

- c. Login to your Azure CLI (Azure CLI must be installed).

```
az login
```

- d. Update the variable values in vars/azure\_anf\_volume\_from\_snapshot.tfvars.



You can choose to deploy the ANF volume using an existing vnet and subnet by setting the variable "vnet\_creation\_bool" and "subnet\_creation\_bool" value to false and supplying the "subnet\_id\_for\_anf\_vol". You can also set those values to true and create a new vnet and subnet in which case, the subnet ID will automatically be taken from the newly created subnet.

- a. Initialize the Terraform repository to install all the pre-requisites and prepare for deployment.

```
terraform init
```

- b. Verify the terraform files using terraform validate command.

```
terraform validate
```

- c. Make a dry run of the configuration to get a preview of all the changes expected by the deployment.

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

d. Run the deployment

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

To delete the deployment

```
terraform destroy
```

**Recipies:**

Single Node Instance

Terraform variables for single ANF Volume using snapshot.

Name	Type	Description
<b>az_location</b>	String	(Required) Specifies the supported Azure location where the resource exists. Changing this forces a new resource to be created.
<b>az_prefix</b>	String	(Required) The name of the resource group where the NetApp Volume should be created. Changing this forces a new resource to be created.
<b>az_vnet_address_space</b>	String	(Required) The address space to be used by the newly created vnet for ANF volume deployment.
<b>az_subnet_address_prefix</b>	String	(Required) The subnet address prefix to be used by the newly created vnet for ANF volume deployment.
<b>az_volume_path</b>	String	(Required) A unique file path for the volume. Used when creating mount targets. Changing this forces a new resource to be created.
<b>az_capacity_pool_size</b>	Integer	(Required) Capacity Pool Size mentioned in TB.
<b>az_vnet_creation_bool</b>	Boolean	(Required) Set this boolean to <code>true</code> if you want to create a new vnet. Set it to <code>false</code> to use an existing vnet.

<b>az_subnet_creation_bool</b>	Boolean	(Required) Set this boolean to <code>true</code> to create a new subnet. Set it to <code>false</code> to use an existing subnet.
<b>az_subnet_id_for_anf_vol</b>	String	(Required) Mention the subnet id in case you decide to use an existing subnet by setting <code>subnet_creation_bool</code> to <code>true</code> . If set to <code>false</code> , leave it at the default value.
<b>az_netapp_pool_service_level</b>	String	(Required) The target performance of the file system. Valid values include <code>Premium</code> , <code>Standard</code> , or <code>Ultra</code> .
<b>az_netapp_vol_service_level</b>	String	(Required) The target performance of the file system. Valid values include <code>Premium</code> , <code>Standard</code> , or <code>Ultra</code> .
<b>az_netapp_vol_protocol</b>	String	(Optional) The target volume protocol expressed as a list. Supported single value include <code>CIFS</code> , <code>NFSv3</code> , or <code>NFSv4.1</code> . If argument is not defined it will default to <code>NFSv3</code> . Changing this forces a new resource to be created and data will be lost.
<b>az_netapp_vol_storage_quota</b>	String	(Required) The maximum Storage Quota allowed for a file system in Gigabytes.
<b>az_snapshot_id</b>	String	(Required) Snapshot ID using which new ANF volume will be created.

## CVO Single Node Deployment

### Terraform configuration files for deployment of Single Node CVO on Azure

This section contains various Terraform configuration files to deploy/configure Single Node CVO (Cloud Volumes ONTAP) on Azure.

Terraform Documentation: <https://registry.terraform.io/providers/NetApp/netapp-cloudmanager/latest/docs>

### Procedure

In order to run the template:

- a. Clone the repository.

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

b. Navigate to the desired folder

```
cd na_cloud_volumes_automation
```

c. Login to your Azure CLI (Azure CLI must be installed).

```
az login
```

d. Update the variables in vars\azure\_cvo\_single\_node\_deployment.tfvars.

e. Initialize the Terraform repository to install all the pre-requisites and prepare for deployment.

```
terraform init
```

f. Verify the terraform files using terraform validate command.

```
terraform validate
```

g. Make a dry run of the configuration to get a preview of all the changes expected by the deployment.

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

h. Run the deployment

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

To delete the deployment

```
terraform destroy
```

**Recipies:**

## Single Node Instance

Terraform variables for single node Cloud Volumes ONTAP (CVO).

Name	Type	Description
<b>refresh_token</b>	String	(Required) The refresh token of NetApp cloud manager. This can be generated from netapp Cloud Central.
<b>az_connector_name</b>	String	(Required) The name of the Cloud Manager Connector.
<b>az_connector_location</b>	String	(Required) The location where the Cloud Manager Connector will be created.
<b>az_connector_subscription_id</b>	String	(Required) The ID of the Azure subscription.
<b>az_connector_company</b>	String	(Required) The name of the company of the user.
<b>az_connector_resource_group</b>	Integer	(Required) The resource group in Azure where the resources will be created.
<b>az_connector_subnet_id</b>	String	(Required) The name of the subnet for the virtual machine.
<b>az_connector_vnet_id</b>	String	(Required) The name of the virtual network.
<b>az_connector_network_security_group_name</b>	String	(Required) The name of the security group for the instance.
<b>az_connector_associate_public_ip_address</b>	String	(Required) Indicates whether to associate the public IP address to the virtual machine.
<b>az_connector_account_id</b>	String	(Required) The NetApp account ID that the Connector will be associated with. If not provided, Cloud Manager uses the first account. If no account exists, Cloud Manager creates a new account. You can find the account ID in the account tab of Cloud Manager at <a href="https://cloudmanager.netapp.com">https://cloudmanager.netapp.com</a> .
<b>az_connector_admin_password</b>	String	(Required) The password for the Connector.
<b>az_connector_admin_username</b>	String	(Required) The user name for the Connector.

<b>az_cvo_name</b>	String	(Required) The name of the Cloud Volumes ONTAP working environment.
<b>az_cvo_location</b>	String	(Required) The location where the working environment will be created.
<b>az_cvo_subnet_id</b>	String	(Required) The name of the subnet for the Cloud Volumes ONTAP system.
<b>az_cvo_vnet_id</b>	String	(Required) The name of the virtual network.
<b>az_cvo_vnet_resource_group</b>	String	(Required) The resource group in Azure associated to the virtual network.
<b>az_cvo_data_encryption_type</b>	String	(Required) The type of encryption to use for the working environment: [AZURE, NONE]. The default is AZURE.
<b>az_cvo_storage_type</b>	String	(Required) The type of storage for the first data aggregate: [Premium_LRS, Standard_LRS, StandardSSD_LRS]. The default is Premium_LRS
<b>az_cvo_svm_password</b>	String	(Required) The admin password for Cloud Volumes ONTAP.
<b>az_cvo_workspace_id</b>	String	(Required) The ID of the Cloud Manager workspace where you want to deploy Cloud Volumes ONTAP. If not provided, Cloud Manager uses the first workspace. You can find the ID from the Workspace tab on <a href="https://cloudmanager.netapp.com">https://cloudmanager.netapp.com</a> .
<b>az_cvo_capacity_tier</b>	String	(Required) Whether to enable data tiering for the first data aggregate: [Blob, NONE]. The default is BLOB.
<b>az_cvo_writing_speed_state</b>	String	(Required) The write speed setting for Cloud Volumes ONTAP: [NORMAL , HIGH]. The default is NORMAL. This argument is not relevant for HA pairs.



<b>az_cvo_ontap_version</b>	String	(Required) The required ONTAP version. Ignored if 'use_latest_version' is set to true. The default is to use the latest version.
<b>az_cvo_instance_type</b>	String	(Required) The type of instance to use, which depends on the license type you chose: Explore:[Standard_DS3_v2], Standard:[Standard_DS4_v2, Standard_DS13_v2, Standard_L8s_v2], Premium:[Standard_DS5_v2, Standard_DS14_v2], BYOL: all instance types defined for PayGo. For more supported instance types, refer to Cloud Volumes ONTAP Release Notes. The default is Standard_DS4_v2 .
<b>az_cvo_license_type</b>	String	(Required) The type of license to be use. For single node: [azure-cot-explore-paygo, azure-cot-standard-paygo, azure-cot-premium-paygo, azure-cot-premium-byol, capacity-paygo]. For HA: [azure-ha-cot-standard-paygo, azure-ha-cot-premium-paygo, azure-ha-cot-premium-byol, ha-capacity-paygo]. The default is azure-cot-standard-paygo. Use capacity-paygo or ha-capacity-paygo for HA on selecting Bring Your Own License type Capacity-Based or Freemium. Use azure-cot-premium-byol or azure-ha-cot-premium-byol for HA on selecting Bring Your Own License type Node-Based.
<b>az_cvo_nss_account</b>	String	(Required) he NetApp Support Site account ID to use with this Cloud Volumes ONTAP system. If the license type is BYOL and an NSS account isn't provided, Cloud Manager tries to use the first existing NSS account.

<b>az_tenant_id</b>	String	(Required) Tenant ID of the application/service principal registered in Azure.
<b>az_application_id</b>	String	(Required) Application ID of the application/service principal registered in Azure.
<b>az_application_key</b>	String	(Required) The Application Key of the application/service principal registered in Azure.

## CVO HA Deployment

### Terraform configuration files for deployment of CVO HA on Azure

This section contains various Terraform configuration files to deploy/configure CVO (Cloud Volumes ONTAP) HA (High Availability) on Azure.

Terraform Documentation: <https://registry.terraform.io/providers/NetApp/netapp-cloudmanager/latest/docs>

### Procedure

In order to run the template:

- a. Clone the repository.

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

- b. Navigate to the desired folder

```
cd na_cloud_volumes_automation
```

- c. Login to your Azure CLI (Azure CLI must be installed).

```
az login
```

- d. Update the variables in `vars\azure_cvo_ha_deployment.tfvars`.

- e. Initialize the Terraform repository to install all the pre-requisites and prepare for deployment.

```
terraform init
```

- f. Verify the terraform files using terraform validate command.

```
terraform validate
```

- g. Make a dry run of the configuration to get a preview of all the changes expected by the deployment.

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

- h. Run the deployment

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

To delete the deployment

```
terraform destroy
```

#### Recipies:

HA Pair Instance

Terraform variables for HA pair Cloud Volumes ONTAP (CVO).

Name	Type	Description
<b>refresh_token</b>	String	(Required) The refresh token of NetApp cloud manager. This can be generated from netapp Cloud Central.
<b>az_connector_name</b>	String	(Required) The name of the Cloud Manager Connector.
<b>az_connector_location</b>	String	(Required) The location where the Cloud Manager Connector will be created.
<b>az_connector_subscription_id</b>	String	(Required) The ID of the Azure subscription.
<b>az_connector_company</b>	String	(Required) The name of the company of the user.
<b>az_connector_resource_group</b>	Integer	(Required) The resource group in Azure where the resources will be created.
<b>az_connector_subnet_id</b>	String	(Required) The name of the subnet for the virtual machine.
<b>az_connector_vnet_id</b>	String	(Required) The name of the virtual network.

<b>az_connector_network_security_group_name</b>	String	(Required) The name of the security group for the instance.
<b>az_connector_associate_public_ip_address</b>	String	(Required) Indicates whether to associate the public IP address to the virtual machine.
<b>az_connector_account_id</b>	String	(Required) The NetApp account ID that the Connector will be associated with. If not provided, Cloud Manager uses the first account. If no account exists, Cloud Manager creates a new account. You can find the account ID in the account tab of Cloud Manager at <a href="https://cloudmanager.netapp.com">https://cloudmanager.netapp.com</a> .
<b>az_connector_admin_password</b>	String	(Required) The password for the Connector.
<b>az_connector_admin_username</b>	String	(Required) The user name for the Connector.
<b>az_cvo_name</b>	String	(Required) The name of the Cloud Volumes ONTAP working environment.
<b>az_cvo_location</b>	String	(Required) The location where the working environment will be created.
<b>az_cvo_subnet_id</b>	String	(Required) The name of the subnet for the Cloud Volumes ONTAP system.
<b>az_cvo_vnet_id</b>	String	(Required) The name of the virtual network.
<b>az_cvo_vnet_resource_group</b>	String	(Required) The resource group in Azure associated to the virtual network.
<b>az_cvo_data_encryption_type</b>	String	(Required) The type of encryption to use for the working environment: [AZURE, NONE]. The default is AZURE.
<b>az_cvo_storage_type</b>	String	(Required) The type of storage for the first data aggregate: [Premium_LRS, Standard_LRS, StandardSSD_LRS]. The default is Premium_LRS
<b>az_cvo_svm_password</b>	String	(Required) The admin password for Cloud Volumes ONTAP.

<b>az_cvo_workspace_id</b>	String	(Required) The ID of the Cloud Manager workspace where you want to deploy Cloud Volumes ONTAP. If not provided, Cloud Manager uses the first workspace. You can find the ID from the Workspace tab on <a href="https://cloudmanager.netapp.com">https://cloudmanager.netapp.com</a> .
<b>az_cvo_capacity_tier</b>	String	(Required) Whether to enable data tiering for the first data aggregate: [Blob, NONE]. The default is BLOB.
<b>az_cvo_writing_speed_state</b>	String	(Required) The write speed setting for Cloud Volumes ONTAP: [NORMAL , HIGH]. The default is NORMAL. This argument is not relevant for HA pairs.
<b>az_cvo_ontap_version</b>	String	(Required) The required ONTAP version. Ignored if 'use_latest_version' is set to true. The default is to use the latest version.
<b>az_cvo_instance_type</b>	String	(Required) The type of instance to use, which depends on the license type you chose: Explore:[Standard_DS3_v2], Standard:[Standard_DS4_v2, Standard_DS13_v2, Standard_L8s_v2], Premium:[Standard_DS5_v2, Standard_DS14_v2], BYOL: all instance types defined for PayGo. For more supported instance types, refer to Cloud Volumes ONTAP Release Notes. The default is Standard_DS4_v2 .

<b>az_cvo_license_type</b>	String	(Required) The type of license to be use. For single node: [azure-cot-explore-paygo, azure-cot-standard-paygo, azure-cot-premium-paygo, azure-cot-premium-byol, capacity-paygo]. For HA: [azure-ha-cot-standard-paygo, azure-ha-cot-premium-paygo, azure-ha-cot-premium-byol, ha-capacity-paygo]. The default is azure-cot-standard-paygo. Use capacity-paygo or ha-capacity-paygo for HA on selecting Bring Your Own License type Capacity-Based or Freemium. Use azure-cot-premium-byol or azure-ha-cot-premium-byol for HA on selecting Bring Your Own License type Node-Based.
<b>az_cvo_nss_account</b>	String	(Required) he NetApp Support Site account ID to use with this Cloud Volumes ONTAP system. If the license type is BYOL and an NSS account isn't provided, Cloud Manager tries to use the first existing NSS account.
<b>az_tenant_id</b>	String	(Required) Tenant ID of the application/service principal registered in Azure.
<b>az_application_id</b>	String	(Required) Application ID of the application/service principal registered in Azure.
<b>az_application_key</b>	String	(Required) The Application Key of the application/service principal registered in Azure.



## CVO Single Node Deployment

### Terraform configuration files for deployment of NetApp CVO (Single Node Instance) on GCP

This section contains various Terraform configuration files to deploy/configure single node NetApp CVO (Cloud Volumes ONTAP) on GCP (Google Cloud Platform).

Terraform Documentation: <https://registry.terraform.io/providers/NetApp/netapp-cloudmanager/latest/docs>

#### Procedure

In order to run the template:

- a. Clone the repository.

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

- b. Navigate to the desired folder

```
cd na_cloud_volumes_automation/
```

- c. Save the GCP authentication key JSON file in the directory.
- d. Update the variable values in `vars/gcp_cvo_single_node_deployment.tfvar`



You can choose to deploy the connector by setting the variable "gcp\_connector\_deploy\_bool" value to true/false.

- e. Initialize the Terraform repository to install all the pre-requisites and prepare for deployment.

```
terraform init
```

- f. Verify the terraform files using terraform validate command.

```
terraform validate
```

- g. Make a dry run of the configuration to get a preview of all the changes expected by the deployment.

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

- h. Run the deployment



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

To delete the deployment

```
terraform destroy
```

### Recipies:

Connector

Terraform variables for NetApp GCP connector instance for CVO deployment.

Name	Type	Description
<b>gcp_connector_deploy_bool</b>	Bool	(Required) Check for Connector deployment.
<b>gcp_connector_name</b>	String	(Required) The name of the Cloud Manager Connector.
<b>gcp_connector_project_id</b>	String	(Required) The GCP project_id where the connector will be created.
<b>gcp_connector_zone</b>	String	(Required) The GCP zone where the Connector will be created.
<b>gcp_connector_company</b>	String	(Required) The name of the company of the user.
<b>gcp_connector_service_account_email</b>	String	(Required) The email of the service_account for the connector instance. This service account is used to allow the Connector to create Cloud Volume ONTAP.
<b>gcp_connector_service_account_path</b>	String	(Required) The local path of the service_account JSON file for GCP authorization purposes. This service account is used to create the Connector in GCP.

<b>gcp_connector_account_id</b>	String	(Optional) The NetApp account ID that the Connector will be associated with. If not provided, Cloud Manager uses the first account. If no account exists, Cloud Manager creates a new account. You can find the account ID in the account tab of Cloud Manager at <a href="https://cloudmanager.netapp.com">https://cloudmanager.netapp.com</a> .
---------------------------------	--------	---

#### Single Node Instance

Terraform variables for single NetApp CVO instance on GCP.

Name	Type	Description
<b>gcp_cvo_name</b>	String	(Required) The name of the Cloud Volumes ONTAP working environment.
<b>gcp_cvo_project_id</b>	String	(Required) The ID of the GCP project.
<b>gcp_cvo_zone</b>	String	(Required) The zone of the region where the working environment will be created.
<b>gcp_cvo_gcp_service_account</b>	String	(Required) The gcp_service_account email in order to enable tiering of cold data to Google Cloud Storage.
<b>gcp_cvo_svm_password</b>	String	(Required) The admin password for Cloud Volumes ONTAP.
<b>gcp_cvo_workspace_id</b>	String	(Optional) The ID of the Cloud Manager workspace where you want to deploy Cloud Volumes ONTAP. If not provided, Cloud Manager uses the first workspace. You can find the ID from the Workspace tab on <a href="https://cloudmanager.netapp.com">https://cloudmanager.netapp.com</a> .

<b>gcp_cvo_license_type</b>	String	(Optional) The type of license to use. For single node: ['capacity-paygo', 'gcp-cot-explore-paygo', 'gcp-cot-standard-paygo', 'gcp-cot-premium-paygo', 'gcp-cot-premium-byol'], For HA: ['ha-capacity-paygo', 'gcp-ha-cot-explore-paygo', 'gcp-ha-cot-standard-paygo', 'gcp-ha-cot-premium-paygo', 'gcp-ha-cot-premium-byol']. The default is 'capacity-paygo' for single node, and 'ha-capacity-paygo' for HA.
<b>gcp_cvo_capacity_package_name</b>	String	(Optional) The capacity package name: ['Essential', 'Professional', 'Freemium']. Default is 'Essential'.

## CVO HA Deployment

### Terraform configuration files for deployment of NetApp CVO (HA Pair) on GCP

This section contains various Terraform configuration files to deploy/configure NetApp CVO (Cloud Volumes ONTAP) in high availability pair on GCP (Google Cloud Platform).

Terraform Documentation: <https://registry.terraform.io/providers/NetApp/netapp-cloudmanager/latest/docs>

### Procedure

In order to run the template:

- a. Clone the repository.

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

- b. Navigate to the desired folder

```
cd na_cloud_volumes_automation/
```

- c. Save the GCP authentication key JSON file in the directory.
- d. Update the variable values in `vars/gcp_cvo_ha_deployment.tfvars`.



You can choose to deploy the connector by setting the variable "gcp\_connector\_deploy\_bool" value to true/false.

- e. Initialize the Terraform repository to install all the pre-requisites and prepare for deployment.

```
terraform init
```

- f. Verify the terraform files using terraform validate command.

```
terraform validate
```

- g. Make a dry run of the configuration to get a preview of all the changes expected by the deployment.

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

- h. Run the deployment

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

To delete the deployment

```
terraform destroy
```

### Recipies:

Connector

Terraform variables for NetApp GCP connector instance for CVO deployment.

Name	Type	Description
<b>gcp_connector_deploy_bool</b>	Bool	(Required) Check for Connector deployment.
<b>gcp_connector_name</b>	String	(Required) The name of the Cloud Manager Connector.
<b>gcp_connector_project_id</b>	String	(Required) The GCP project_id where the connector will be created.
<b>gcp_connector_zone</b>	String	(Required) The GCP zone where the Connector will be created.
<b>gcp_connector_company</b>	String	(Required) The name of the company of the user.

<b>gcp_connector_service_account_email</b>	String	(Required) The email of the service_account for the connector instance. This service account is used to allow the Connector to create Cloud Volume ONTAP.
<b>gcp_connector_service_account_path</b>	String	(Required) The local path of the service_account JSON file for GCP authorization purposes. This service account is used to create the Connector in GCP.
<b>gcp_connector_account_id</b>	String	(Optional) The NetApp account ID that the Connector will be associated with. If not provided, Cloud Manager uses the first account. If no account exists, Cloud Manager creates a new account. You can find the account ID in the account tab of Cloud Manager at <a href="https://cloudmanager.netapp.com">https://cloudmanager.netapp.com</a> .

#### HA Pair

Terraform variables for NetApp CVO instances in HA Pair on GCP.

Name	Type	Description
<b>gcp_cvo_is_ha</b>	Bool	(Optional) Indicate whether the working environment is an HA pair or not [true, false]. The default is false.
<b>gcp_cvo_name</b>	String	(Required) The name of the Cloud Volumes ONTAP working environment.
<b>gcp_cvo_project_id</b>	String	(Required) The ID of the GCP project.
<b>gcp_cvo_zone</b>	String	(Required) The zone of the region where the working environment will be created.
<b>gcp_cvo_node1_zone</b>	String	(Optional) Zone for node 1.
<b>gcp_cvo_node2_zone</b>	String	(Optional) Zone for node 2.
<b>gcp_cvo_mediator_zone</b>	String	(Optional) Zone for mediator.
<b>gcp_cvo_vpc_id</b>	String	(Optional) The name of the VPC.

<b>gcp_cvo_subnet_id</b>	String	(Optional) The name of the subnet for Cloud Volumes ONTAP. The default is: 'default'.
<b>gcp_cvo_vpc0_node_and_data_connectivity</b>	String	(Optional) VPC path for nic1, required for node and data connectivity. If using shared VPC, network_project_id must be provided.
<b>gcp_cvo_vpc1_cluster_connectivity</b>	String	(Optional) VPC path for nic2, required for cluster connectivity.
<b>gcp_cvo_vpc2_ha_connectivity</b>	String	(Optional) VPC path for nic3, required for HA connectivity.
<b>gcp_cvo_vpc3_data_replication</b>	String	(Optional) VPC path for nic4, required for data replication.
<b>gcp_cvo_subnet0_node_and_data_connectivity</b>	String	(Optional) Subnet path for nic1, required for node and data connectivity. If using shared VPC, network_project_id must be provided.
<b>gcp_cvo_subnet1_cluster_connectivity</b>	String	(Optional) Subnet path for nic2, required for cluster connectivity.
<b>gcp_cvo_subnet2_ha_connectivity</b>	String	(Optional) Subnet path for nic3, required for HA connectivity.
<b>gcp_cvo_subnet3_data_replication</b>	String	(Optional) Subnet path for nic4, required for data replication.
<b>gcp_cvo_gcp_service_account</b>	String	(Required) The gcp_service_account email in order to enable tiering of cold data to Google Cloud Storage.
<b>gcp_cvo_svm_password</b>	String	(Required) The admin password for Cloud Volumes ONTAP.
<b>gcp_cvo_workspace_id</b>	String	(Optional) The ID of the Cloud Manager workspace where you want to deploy Cloud Volumes ONTAP. If not provided, Cloud Manager uses the first workspace. You can find the ID from the Workspace tab on <a href="https://cloudmanager.netapp.com">https://cloudmanager.netapp.com</a> .

<b>gcp_cvo_license_type</b>	String	(Optional) The type of license to use. For single node: ['capacity-paygo', 'gcp-cot-explore-paygo', 'gcp-cot-standard-paygo', 'gcp-cot-premium-paygo', 'gcp-cot-premium-byol'], For HA: ['ha-capacity-paygo', 'gcp-ha-cot-explore-paygo', 'gcp-ha-cot-standard-paygo', 'gcp-ha-cot-premium-paygo', 'gcp-ha-cot-premium-byol']. The default is 'capacity-paygo' for single node, and 'ha-capacity-paygo' for HA.
<b>gcp_cvo_capacity_package_name</b>	String	(Optional) The capacity package name: ['Essential', 'Professional', 'Freemium']. Default is 'Essential'.
<b>gcp_cvo_gcp_volume_size</b>	String	(Optional) The GCP volume size for the first data aggregate. For GB, the unit can be: [100 or 500]. For TB, the unit can be: [1,2,4,8]. The default is '1' .
<b>gcp_cvo_gcp_volume_size_unit</b>	String	(Optional) ['GB' or 'TB']. The default is 'TB'.

## CVS Volume

### Terraform configuration files for deployment of NetApp CVS Volume on GCP

This section contains various Terraform configuration files to deploy/configure NetApp CVS (Cloud Volumes Services) Volume on GCP (Google Cloud Platform).

Terraform Documentation: <https://registry.terraform.io/providers/NetApp/netapp-gcp/latest/docs>

### Procedure

In order to run the template:

- a. Clone the repository.

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

- b. Navigate to the desired folder

```
cd na_cloud_volumes_automation/
```

- c. Save the GCP authentication key JSON file in the directory.
- d. Update the variable values in vars/gcp\_cvs\_volume.tfvars.
- e. Initialize the Terraform repository to install all the pre-requisites and prepare for deployment.

```
terraform init
```

- f. Verify the terraform files using terraform validate command.

```
terraform validate
```

- g. Make a dry run of the configuration to get a preview of all the changes expected by the deployment.

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

- h. Run the deployment

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

To delete the deployment

```
terraform destroy
```

### Recipies:

CVS Volume

Terraform variables for NetApp GCP CVS Volume.

Name	Type	Description
<b>gcp_cvs_name</b>	String	(Required) The name of the NetApp CVS volume.
<b>gcp_cvs_project_id</b>	String	(Required) The GCP project_id where the CVS Volume will be created.
<b>gcp_cvs_gcp_service_account_path</b>	String	(Required) The local path of the service_account JSON file for GCP authorization purposes. This service account is used to create the CVS Volume in GCP.
<b>gcp_cvs_region</b>	String	(Required) The GCP zone where the CVS Volume will be created.



<b>gcp_cvs_network</b>	String	(Required) The network VPC of the volume.
<b>gcp_cvs_size</b>	Integer	(Required) The size of volume is between 1024 to 102400 inclusive (in GiB).
<b>gcp_cvs_volume_path</b>	String	(Optional) The name of the volume path for volume.
<b>gcp_cvs_protocol_types</b>	String	(Required) The protocol_type of the volume. For NFS use 'NFSv3' or 'NFSv4' and for SMB use 'CIFS' or 'SMB'.

## FSx for ONTAP Monitoring and Auto-Resizing using AWS Lambda Function

Author(s): Dhruv Tyagi, Niyaz Mohamed

### Overview: Monitoring and Auto-Resizing FSx for ONTAP via AWS Lambda function

FSx for ONTAP is a first party enterprise-grade cloud storage service available on AWS that provides highly reliable, scalable, high-performing and feature-rich file storage built on the popular NetApp ONTAP file system.

FSx for ONTAP provides a seamless deployment and management experience. No storage expertise is required to get started. To simplify monitoring, an AWS lambda function (to automate resizing of total storage capacity, volume size or LUN size based on threshold) can be used. This document provides a step by step guide to create an automated setup that monitors FSx for ONTAP at regular intervals, notifies and resizes when a user-specified threshold is crossed and notifies the administrator of the resizing activity.

### Features

The solution provides the following features:

- Ability to monitor:
  - Usage of overall Storage Capacity of FSx for ONTAP
  - Usage of each volume (thin provisioned / thick provisioned)
  - Usage of each LUN (thin provisioned / thick provisioned)
- Ability to resize any of the above when a user-defined threshold is breached
- Alerting mechanism to receive usage warning and resizing notifications via email
- Ability to delete snapshots older than user-defined threshold
- Ability to get a list of FlexClone volumes and snapshots associated
- Ability to run the checks at a regular interval

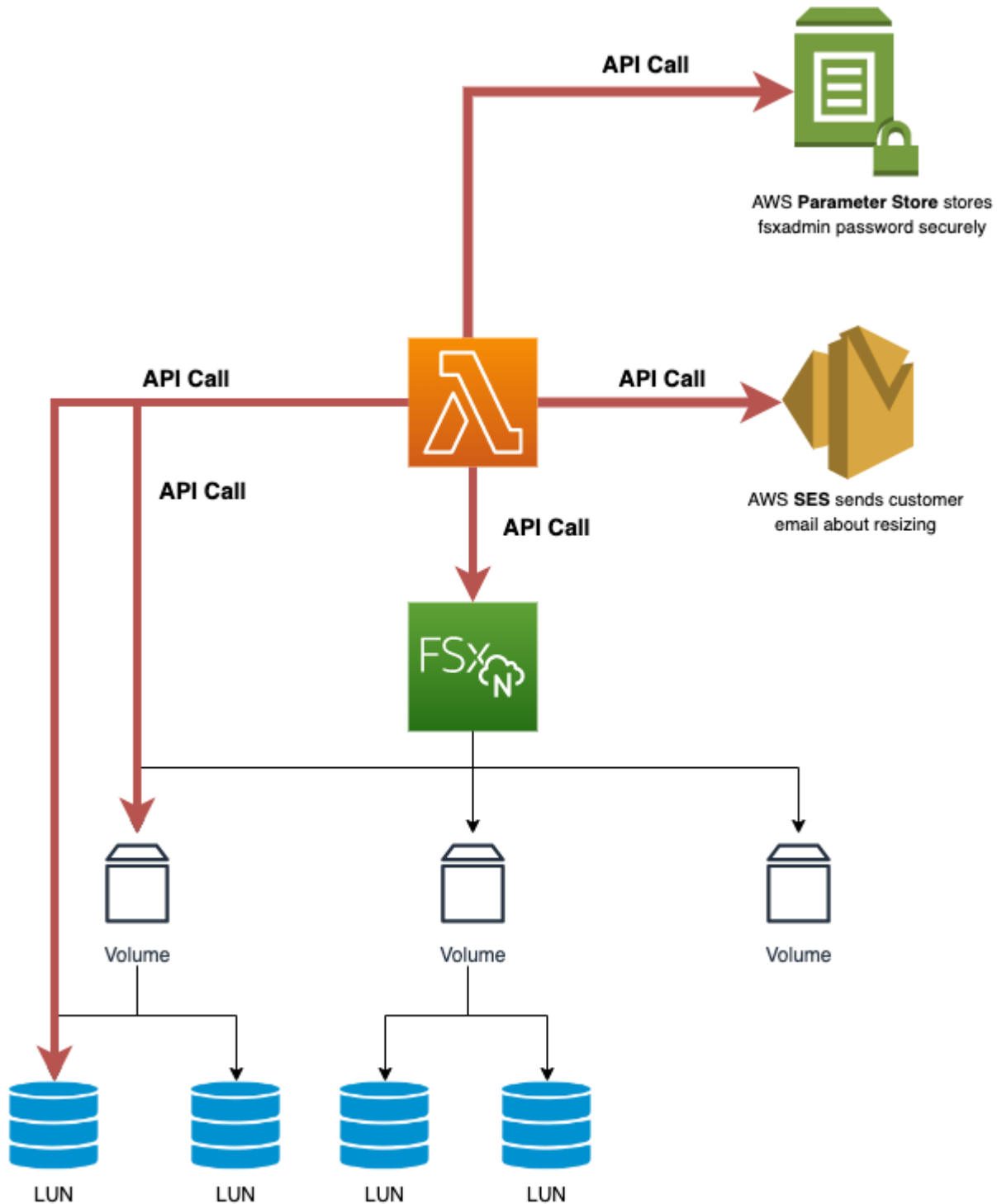
## Pre-requisites

Before you begin, ensure that the following prerequisites are met:

- FSx for ONTAP is deployed
- A Private Subnet with a NAT gateway attached is required by the lambda function for internet connectivity
- The private subnet should also have connectivity to FSx for ONTAP
- "fsxadmin" password has been set for FSx for ONTAP

## High Level Architecture

- AWS Lambda Function makes API calls to FSx for ONTAP for retrieving and updating the size of Storage Capacity, Volumes and LUNs.
- "fsxadmin" password stored as secure string in AWS SSM Parameter Store for added layer of security.
- AWS SES (Simple Email Service) is used to notify end-users when a resizing event occurs.



## Solution Deployment

Follow the series of steps to complete the deployment of this solution:

### Step 1: Clone the GitHub repository

Clone the GitHub repository on your local system:

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

## Step 2: Create SSM parameter for fsxadmin password

Navigate to AWS Console > **Parameter Store** and click on **Create Parameter**.

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>

Click on **Create parameter**.

The screenshot shows the AWS Management Console interface for creating a new parameter. The breadcrumb navigation at the top indicates the path: AWS Systems Manager > Parameter Store > Create parameter. The main heading is 'Create parameter'. Below this, the 'Parameter details' section contains several fields and options:

- Name:** A text input field containing '/fsxn/password/'.
- Description — Optional:** An empty text input field.
- Tier:** Two radio button options: 'Standard' (selected) and 'Advanced'. The 'Standard' option includes a note: 'Limit of 10,000 parameters. Parameter value size up to 4 KB. Parameter policies are not available. No additional charge.' The 'Advanced' option includes a note: 'Can create more than 10,000 parameters. Parameter value size up to 8 KB. Parameter policies are available. Charges apply.'
- Type:** Three radio button options: 'String' (Any string value), 'StringList' (Separate strings using commas), and 'SecureString' (selected, Encrypt sensitive data using KMS keys from your account or another account).
- KMS key source:** Two radio button options: 'My current account' (selected) and 'Another account'. The 'My current account' option has a sub-note: 'Use the default KMS key for this account or specify a customer-managed key for this account. [Learn more](#)'.
- KMS Key ID:** A dropdown menu showing 'alias/aws/ssm'.
- Value:** A text input field with masked characters (dots).

A blue information box is displayed below the KMS Key ID dropdown, stating: '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](#)'.

### Step 3: Setup Email Service

Navigate to AWS Console > **Simple Email Service (SES)** and click on **Create Identity**.

Identity type: Email address

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

Click on **Create identity**

The screenshot shows the AWS Management Console interface. At the top, there's a navigation bar with the AWS logo, 'Services', a search bar, and a '[Option+S]' button. Below this is a 'Resource Groups & Tag Editor' link. The main content area has a breadcrumb trail: 'Amazon SES > Configuration: Verified identities > Create identity'. The title 'Create identity' is prominently displayed, followed by a descriptive paragraph: '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.' The form is divided into two main sections. The first section, 'Identity details', 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 sub-description: 'To verify ownership of an email address, you must have access to its inbox to open the verification email.' Below this, there's a text input field for 'Email address' containing 'abc@xyz.com', with a note stating 'Email address can contain up to 320 characters, including plus signs (+), equals signs (=) and underscores (\_)'.

☐ **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**  
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**

## Step 4: Create and setup the AWS Lambda Function

1. Navigate to AWS Console > **AWS Lambda** and click on **Create function** in the same region as FSx for ONTAP
2. Use the default **Author from scratch** and update the following fields:

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 which have NAT gateway attached pointing to public subnets with internet gateway and subnets that have internet access>  
        Security Group: <Choose a Security Group>

Click on **Create function**.

The screenshot shows the 'Create function' page in the AWS Lambda console. At the top, there are three tabs: 'Author from scratch' (selected), 'Use a blueprint', and 'Container image'. Below the tabs, the 'Basic information' section is visible. It includes fields for 'Function name' (with 'DemoFunction' entered), 'Runtime' (set to 'Python 3.9'), and 'Architecture' (set to 'x86\_64'). The 'Permissions' section is expanded, showing options to 'Change default execution role'. The 'Execution role' section is also expanded, showing the option to 'Create a new role with basic Lambda permissions' selected. A blue box at the bottom of the permissions section states: 'Role creation might take a few minutes. Please do not delete the role or edit the trust or permissions policies in this role.' At the very bottom, a note says: 'Lambda will create an execution role named DemoFunction-role-zu7fjy2, with permission to upload logs to Amazon CloudWatch Logs.'

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: [selected] [refresh]

**Subnets**  
Select the VPC subnets for Lambda to use to set up your VPC configuration.

Choose subnets

subnet: [selected] us-east-1a X subnet: [selected] 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: [selected] X  
default vpc security group

3. Scroll down to the **Layers** section of the newly created Lambda function and click on **Add a layer**.

**Layers** [info](#) Edit Add a layer

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

4. Click on **create a new layer** under **Layer source**

5. Create 2 Layers - 1 for Requests and 1 for Paramiko and upload **Requests.zip** and **Paramiko.zip** files. Select **Python 3.9** as the compatible runtime and click on **Create**.





## Create layer

### Layer configuration

Name

paramiko

Description - *optional*

Description

☒ Upload a .zip file

☐ Upload a file from Amazon S3



Upload

paramiko.zip (6.9 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](#)

6. Navigate back to AWS Lambda **Add Layer > Custom Layers** and add the paramiko and requests layer one after the other.


[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.

Requests

Version

1

Cancel

Add

#### Layers [Info](#)

Edit

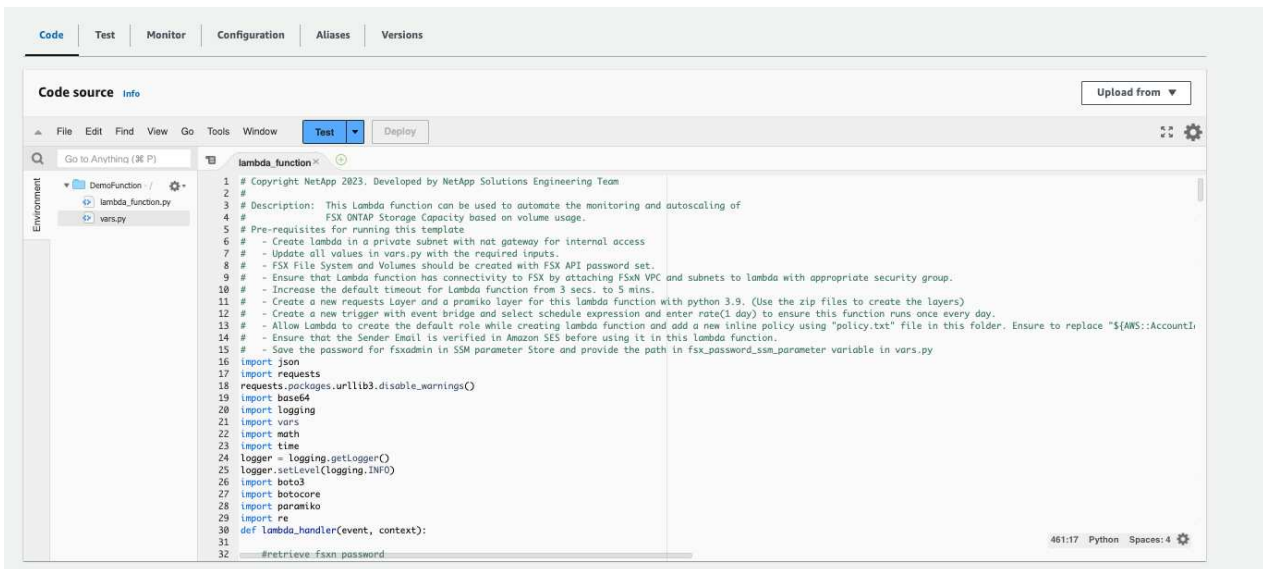
Add a layer

Merge order	Name	Layer version	Compatible runtimes	Compatible architectures	Version ARN
1	Requests	1	python3.9	x86_64	arn:aws:lambda:us-east-1:██████:layer:Requests:1
2	paramiko	1	python3.9	-	arn:aws:lambda:us-east-1:██████:layer:paramiko:1

- Navigate to the **Configuration** tab of the Lambda function and click on **Edit** under **General Configuration**. Change the Timeout to **5 mins** and click Save.
- Navigate to **Permissions** tab of the Lambda function and click on the role assigned. In the permissions tab of the role, click on **Add permissions** > **Create Inline policy**.
  - Click on the JSON tab and paste the contents of the file policy.json from the GitHub repo.
  - Replace every occurrence of `${AWS::AccountId}` with your account ID and click on **Review Policy**
  - Provide a Name for the policy and click on **Create policy**
- Copy the contents of `fsxn_monitoring_resizing_lambda.py` from the git repo to `lambda_function.py` in the AWS Lambda function Code Source section.
- Create a new file in the same level as `lambda_function.py` and name it `vars.py` and copy the contents of `vars.py` from the git repo to the lambda function `vars.py` file. Update the variable values in `vars.py`. Reference variable definitions below and click on **Deploy**:

Name	Type	Description
------	------	-------------

<b>fsxMgmtIp</b>	String	(Required) Enter the "Management endpoint - IP address" from the FSx for ONTAP console on AWS.
<b>fsxId</b>	String	(Required) Enter the "File system ID" from the FSx for ONTAP console on AWS.
<b>username</b>	String	(Required) Enter the FSx for ONTAP "ONTAP administrator username" from FSx for ONTAP console on AWS.
<b>resize_threshold</b>	Integer	(Required) 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.
<b>sender_email</b>	String	(Required) Enter the email ID registered on SES that will be used by the lambda function to send notification alerts related to monitoring and resizing.
<b>recipient_email</b>	String	(Required) Enter the email ID on which you want to receive the alert notifications.
<b>fsx_password_ssm_parameter</b>	String	(Required) Enter the path name used in AWS Parameter Store for storing "fsxadmin" password.
<b>warn_notification</b>	Bool	(Required) Set this variable to True to receive notification when Storage Capacity/Volume/LUN usage exceeds 75% but is less than threshold.
<b>enable_snapshot_deletion</b>	Bool	(Required) Set this variable to True to enable volume level snapshot deletion for snapshots older than the value specified in "snapshot_age_threshold_in_days".
<b>snapshot_age_threshold_in_days</b>	Integer	(Required) 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.



11. Click on **Test**, create an empty test event and run the test and check if the script is running properly.
12. Once tested successfully, navigate to **Configuration > Triggers > Add Trigger**.

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>

Click on Add.



Lambda &gt; Add trigger

## 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.

## Conclusion

With the provided solution, it is easy to setup a monitoring solution that regularly monitors FSx for ONTAP Storage, resizes it based on user-specified threshold and provides an alerting mechanism. This makes the process of using and monitoring FSx for ONTAP seamless freeing up administrators to focus on business-critical activities while storage grows automatically when required.

## Copyright information

Copyright © 2023 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.