



Solution Automation

NetApp Solutions

NetApp
April 26, 2024

This PDF was generated from https://docs.netapp.com/us-en/netapp-solutions/automation/automation_introduction.html on April 26, 2024. 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 5
 - Cloud Volumes Automation via Terraform 8
 - FSx for ONTAP Monitoring and Auto-Resizing using AWS Lambda Function 47

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. Paste the below content in the Terminal

```
sudo yum -y install python3 >> install.log  
sudo yum -y install python3-pip >> install.log  
python3 -W ignore -m pip --disable-pip-version-check install ansible  
>> install.log  
sudo yum -y install git >> install.log
```

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. Paste the below content in the terminal

```
sudo apt-get -y install python3 >> outputlog.txt
sudo apt-get -y install python3-pip >> outputlog.txt
python3 -W ignore -m pip --disable-pip-version-check install ansible
>> outputlog.txt
sudo apt-get -y install git >> outputlog.txt
```

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/na_cloud_volumes_automation

Pre-requisites

1. Terraform ≥ 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 = ">= 23.0.0"
    }
  }
}
```

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/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
```

Recipes:

Connector

Terraform variables for NetApp AWS connector instance for CVO deployment.

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

Name	Type	Description
aws_connector_account_id	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 https://cloudmanager.netapp.com .
aws_connector_public_ip_bool	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
cvo_name	String	(Required) The name of the Cloud Volumes ONTAP working environment.
cvo_region	String	(Required) The region where the working environment will be created.
cvo_subnet_id	String	(Required) The subnet id where the working environment will be created.
cvo_vpc_id	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.
cvo_svm_passwd	String	(Required) The admin password for Cloud Volumes ONTAP.
cvo_writing_speed_state	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/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
aws_connector_deploy_bool	Bool	(Required) Check for Connector deployment.
aws_connector_name	String	(Required) The name of the Cloud Manager Connector.
aws_connector_region	String	(Required) The region where the Cloud Manager Connector will be created.
aws_connector_key_name	String	(Required) The name of the key pair to use for the Connector instance.
aws_connector_company	String	(Required) The name of the company of the user.
aws_connector_instance_type	String	(Required) The type of instance (for example, t3.xlarge). At least 4 CPU and 16 GB of memory are required.
aws_connector_subnet_id	String	(Required) The ID of the subnet for the instance.
aws_connector_security_group_id	String	(Required) The ID of the security group for the instance, multiple security groups can be provided separated by ','.
aws_connector_instance_profile_name	String	(Required) The name of the instance profile for the Connector.
aws_connector_account_id	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 https://cloudmanager.netapp.com .
aws_connector_public_ip_bool	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
cvo_is_ha	Bool	(Optional) Indicate whether the working environment is an HA pair or not [true, false]. The default is false.
cvo_name	String	(Required) The name of the Cloud Volumes ONTAP working environment.
cvo_region	String	(Required) The region where the working environment will be created.

Name	Type	Description
cvo_node1_subnet_id	String	(Required) The subnet id where the first node will be created.
cvo_node2_subnet_id	String	(Required) The subnet id where the second node will be created.
cvo_vpc_id	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.
cvo_svm_password	String	(Required) The admin password for Cloud Volumes ONTAP.
cvo_failover_mode	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.
cvo_mediator_subnet_id	String	(Optional) For HA, the subnet ID of the mediator.
cvo_mediator_key_pair_name	String	(Optional) For HA, the key pair name for the mediator instance.
cvo_cluster_floating_ip	String	(Optional) For HA FloatingIP, the cluster management floating IP address.
cvo_data_floating_ip	String	(Optional) For HA FloatingIP, the data floating IP address.
cvo_data_floating_ip2	String	(Optional) For HA FloatingIP, the data floating IP address.
cvo_svm_floating_ip	String	(Optional) For HA FloatingIP, the SVM management floating IP address.
cvo_route_table_ids	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/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
aws_connector_deploy_bool	Bool	(Required) Check for Connector deployment.
aws_connector_name	String	(Required) The name of the Cloud Manager Connector.
aws_connector_region	String	(Required) The region where the Cloud Manager Connector will be created.
aws_connector_key_name	String	(Required) The name of the key pair to use for the Connector instance.
aws_connector_company	String	(Required) The name of the company of the user.
aws_connector_instance_type	String	(Required) The type of instance (for example, t3.xlarge). At least 4 CPU and 16 GB of memory are required.
aws_connector_subnet_id	String	(Required) The ID of the subnet for the instance.
aws_connector_security_group_id	String	(Required) The ID of the security group for the instance, multiple security groups can be provided separated by ','.
aws_connector_instance_profile_name	String	(Required) The name of the instance profile for the Connector.
aws_connector_account_id	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 https://cloudmanager.netapp.com .
aws_connector_public_ip_bool	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
fsx_name	String	(Required) The name of the Cloud Volumes ONTAP working environment.
fsx_region	String	(Required) The region where the working environment will be created.
fsx_primary_subnet_id	String	(Required) The primary subnet id where the working environment will be created.

Name	Type	Description
fsx_secondary_subnet_id	String	(Required) The secondary subnet id where the working environment will be created.
fsx_account_id	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 https://cloudmanager.netapp.com .
fsx_workspace_id	String	(Required) The ID of the Cloud Manager workspace of working environment.
fsx_admin_password	String	(Required) The admin password for Cloud Volumes ONTAP.
fsx_throughput_capacity	String	(Optional) capacity of the throughput.
fsx_storage_capacity_size	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'
fsx_storage_capacity_size_unit	String	(Optional) ['GB' or 'TB']. The default is 'TB'.
fsx_cloudmanager_aws_credentials_name	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/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
az_location	String	(Required) Specifies the supported Azure location where the resource exists. Changing this forces a new resource to be created.
az_prefix	String	(Required) The name of the resource group where the NetApp Volume should be created. Changing this forces a new resource to be created.
az_vnet_address_space	String	(Required) The address space to be used by the newly created vnet for ANF volume deployment.
az_subnet_address_prefix	String	(Required) The subnet address prefix to be used by the newly created vnet for ANF volume deployment.
az_volume_path	String	(Required) A unique file path for the volume. Used when creating mount targets. Changing this forces a new resource to be created.
az_capacity_pool_size	Integer	(Required) Capacity Pool Size mentioned in TB.
az_vnet_creation_bool	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.
az_subnet_creation_bool	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.
az_subnet_id_for_anf_vol	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.
az_netapp_pool_service_level	String	(Required) The target performance of the file system. Valid values include <code>Premium</code> , <code>Standard</code> , or <code>Ultra</code> .

Name	Type	Description
az_netapp_vol_service_level	String	(Required) The target performance of the file system. Valid values include Premium , Standard , or Ultra.
az_netapp_vol_protocol	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.
az_netapp_vol_security_style	String	(Optional) Volume security style, accepted values are Unix or Ntfs. If not provided, single-protocol volume is created defaulting to Unix if it is NFSv3 or NFSv4.1 volume, if CIFS, it will default to Ntfs. In a dual-protocol volume, if not provided, its value will be Ntfs.
az_netapp_vol_storage_quota	String	(Required) The maximum Storage Quota allowed for a file system in Gigabytes.



As per the recommendation, this script uses the `prevent_destroy` lifecycle argument to mitigate the possibility of accidental data loss in the configuration file. For more information on the `prevent_destroy` lifecycle argument please see the terraform documentation: <https://developer.hashicorp.com/terraform/tutorials/state/resource-lifecycle#prevent-resource-deletion>.

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/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
az_location	String	(Required) Specifies the supported Azure location where the resource exists. Changing this forces a new resource to be created.
az_alt_location	String	(Required) The Azure location where the secondary volume will be created
az_prefix	String	(Required) The name of the resource group where the NetApp Volume should be created. Changing this forces a new resource to be created.

Name	Type	Description
az_vnet_primary_address_space	String	(Required) The address space to be used by the newly created vnet for ANF primary volume deployment.
az_vnet_secondary_address_space	String	(Required) The address space to be used by the newly created vnet for ANF secondary volume deployment.
az_subnet_primary_address_prefix	String	(Required) The subnet address prefix to be used by the newly created vnet for ANF primary volume deployment.
az_subnet_secondary_address_prefix	String	(Required) The subnet address prefix to be used by the newly created vnet for ANF secondary volume deployment.
az_volume_path_primary	String	(Required) A unique file path for the primary volume. Used when creating mount targets. Changing this forces a new resource to be created.
az_volume_path_secondary	String	(Required) A unique file path for the secondary volume. Used when creating mount targets. Changing this forces a new resource to be created.
az_capacity_pool_size_primary	Integer	(Required) Capacity Pool Size mentioned in TB.
az_capacity_pool_size_secondary	Integer	(Required) Capacity Pool Size mentioned in TB.
az_vnet_primary_creation_bool	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.
az_vnet_secondary_creation_bool	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.
az_subnet_primary_creation_bool	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.
az_subnet_secondary_creation_bool	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.
az_primary_subnet_id_for_anf_vol	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.
az_secondary_subnet_id_for_anf_vol	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.
az_netapp_pool_service_level_primary	String	(Required) The target performance of the file system. Valid values include <code>Premium</code> , <code>Standard</code> , or <code>Ultra</code> .
az_netapp_pool_service_level_secondary	String	(Required) The target performance of the file system. Valid values include <code>Premium</code> , <code>Standard</code> , or <code>Ultra</code> .

Name	Type	Description
az_netapp_vol_service_level_primary	String	(Required) The target performance of the file system. Valid values include Premium , Standard , or Ultra.
az_netapp_vol_service_level_secondary	String	(Required) The target performance of the file system. Valid values include Premium , Standard , or Ultra.
az_netapp_vol_protocol_primary	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.
az_netapp_vol_protocol_secondary	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.
az_netapp_vol_storage_quota_primary	String	(Required) The maximum Storage Quota allowed for a file system in Gigabytes.
az_netapp_vol_storage_quota_secondary	String	(Required) The maximum Storage Quota allowed for a file system in Gigabytes.
az_dp_replication_frequency	String	(Required) Replication frequency, supported values are 10minutes, hourly, daily, values are case sensitive.



As per the recommendation, this script uses the `prevent_destroy` lifecycle argument to mitigate the possibility of accidental data loss in the configuration file. For more information on the `prevent_destroy` lifecycle argument please see the terraform documentation: <https://developer.hashicorp.com/terraform/tutorials/state/resource-lifecycle#prevent-resource-deletion>.

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.

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/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
az_location	String	(Required) Specifies the supported Azure location where the resource exists. Changing this forces a new resource to be created.
az_prefix	String	(Required) The name of the resource group where the NetApp Volume should be created. Changing this forces a new resource to be created.
az_vnet_address_space	String	(Required) The address space to be used by the newly created vnet for ANF volume deployment.
az_subnet_address_prefix	String	(Required) The subnet address prefix to be used by the newly created vnet for ANF volume deployment.
az_volume_path	String	(Required) A unique file path for the volume. Used when creating mount targets. Changing this forces a new resource to be created.
az_capacity_pool_size	Integer	(Required) Capacity Pool Size mentioned in TB.
az_vnet_creation_bool	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.
az_subnet_creation_bool	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.
az_subnet_id_for_anf_vol	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.
az_netapp_pool_service_level	String	(Required) The target performance of the file system. Valid values include <code>Premium</code> , <code>Standard</code> , or <code>Ultra</code> .
az_netapp_vol_service_level	String	(Required) The target performance of the file system. Valid values include <code>Premium</code> , <code>Standard</code> , or <code>Ultra</code> .
az_netapp_vol_protocol1	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.
az_netapp_vol_protocol2	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.
az_netapp_vol_storage_quota	String	(Required) The maximum Storage Quota allowed for a file system in Gigabytes.
az_smb_server_username	String	(Required) Username to create ActiveDirectory object.
az_smb_server_password	String	(Required) User Password to create ActiveDirectory object.
az_smb_server_name	String	(Required) Server Name to create ActiveDirectory object.

Name	Type	Description
az_smb_dns_server	String	(Required) DNS Server IP to create ActiveDirectory object.



As per the recommendation, this script uses the `prevent_destroy` lifecycle argument to mitigate the possibility of accidental data loss in the configuration file. For more information on the `prevent_destroy` lifecycle argument please see the terraform documentation: <https://developer.hashicorp.com/terraform/tutorials/state/resource-lifecycle#prevent-resource-deletion>.

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

Name	Type	Description
az_subnet_creation_bool	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.
az_subnet_id_for_anf_vol	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.
az_netapp_pool_service_level	String	(Required) The target performance of the file system. Valid values include <code>Premium</code> , <code>Standard</code> , or <code>Ultra</code> .
az_netapp_vol_service_level	String	(Required) The target performance of the file system. Valid values include <code>Premium</code> , <code>Standard</code> , or <code>Ultra</code> .
az_netapp_vol_protocol	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.
az_netapp_vol_storage_quota	String	(Required) The maximum Storage Quota allowed for a file system in Gigabytes.
az_snapshot_id	String	(Required) Snapshot ID using which new ANF volume will be created.



As per the recommendation, this script uses the `prevent_destroy` lifecycle argument to mitigate the possibility of accidental data loss in the configuration file. For more information on the `prevent_destroy` lifecycle argument please see the terraform documentation: <https://developer.hashicorp.com/terraform/tutorials/state/resource-lifecycle#prevent-resource-deletion>.

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/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
```

Recipes:

Single Node Instance

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

Name	Type	Description
refresh_token	String	(Required) The refresh token of NetApp cloud manager. This can be generated from netapp Cloud Central.
az_connector_name	String	(Required) The name of the Cloud Manager Connector.

Name	Type	Description
az_connector_location	String	(Required) The location where the Cloud Manager Connector will be created.
az_connector_subscription_id	String	(Required) The ID of the Azure subscription.
az_connector_company	String	(Required) The name of the company of the user.
az_connector_resource_group	Integer	(Required) The resource group in Azure where the resources will be created.
az_connector_subnet_id	String	(Required) The name of the subnet for the virtual machine.
az_connector_vnet_id	String	(Required) The name of the virtual network.
az_connector_network_security_group_name	String	(Required) The name of the security group for the instance.
az_connector_associate_public_ip_address	String	(Required) Indicates whether to associate the public IP address to the virtual machine.
az_connector_account_id	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 https://cloudmanager.netapp.com .
az_connector_admin_password	String	(Required) The password for the Connector.
az_connector_admin_username	String	(Required) The user name for the Connector.
az_cvo_name	String	(Required) The name of the Cloud Volumes ONTAP working environment.
az_cvo_location	String	(Required) The location where the working environment will be created.
az_cvo_subnet_id	String	(Required) The name of the subnet for the Cloud Volumes ONTAP system.
az_cvo_vnet_id	String	(Required) The name of the virtual network.
az_cvo_vnet_resource_group	String	(Required) The resource group in Azure associated to the virtual network.
az_cvo_data_encryption_type	String	(Required) The type of encryption to use for the working environment: [AZURE, NONE]. The default is AZURE.
az_cvo_storage_type	String	(Required) The type of storage for the first data aggregate: [Premium_LRS, Standard_LRS, StandardSSD_LRS]. The default is Premium_LRS

Name	Type	Description
az_cvo_svm_password	String	(Required) The admin password for Cloud Volumes ONTAP.
az_cvo_workspace_id	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 https://cloudmanager.netapp.com .
az_cvo_capacity_tier	String	(Required) Whether to enable data tiering for the first data aggregate: [Blob, NONE]. The default is BLOB.
az_cvo_writing_speed_state	String	(Required) The write speed setting for Cloud Volumes ONTAP: [NORMAL , HIGH]. The default is NORMAL. This argument is not relevant for HA pairs.
az_cvo_ontap_version	String	(Required) The required ONTAP version. Ignored if 'use_latest_version' is set to true. The default is to use the latest version.
az_cvo_instance_type	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 .
az_cvo_license_type	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.
az_cvo_nss_account	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.
az_tenant_id	String	(Required) Tenant ID of the application/service principal registered in Azure.
az_application_id	String	(Required) Application ID of the application/service principal registered in Azure.
az_application_key	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/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
```

Recipes:

HA Pair Instance

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

Name	Type	Description
refresh_token	String	(Required) The refresh token of NetApp cloud manager. This can be generated from netapp Cloud Central.
az_connector_name	String	(Required) The name of the Cloud Manager Connector.
az_connector_location	String	(Required) The location where the Cloud Manager Connector will be created.
az_connector_subscription_id	String	(Required) The ID of the Azure subscription.
az_connector_company	String	(Required) The name of the company of the user.
az_connector_resource_group	Integer	(Required) The resource group in Azure where the resources will be created.
az_connector_subnet_id	String	(Required) The name of the subnet for the virtual machine.
az_connector_virtual_network_id	String	(Required) The name of the virtual network.
az_connector_network_security_group_name	String	(Required) The name of the security group for the instance.
az_connector_associate_public_ip_address	String	(Required) Indicates whether to associate the public IP address to the virtual machine.
az_connector_account_id	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 https://cloudmanager.netapp.com .
az_connector_admin_password	String	(Required) The password for the Connector.
az_connector_admin_username	String	(Required) The user name for the Connector.
az_cvo_name	String	(Required) The name of the Cloud Volumes ONTAP working environment.

Name	Type	Description
az_cvo_location	String	(Required) The location where the working environment will be created.
az_cvo_subnet_id	String	(Required) The name of the subnet for the Cloud Volumes ONTAP system.
az_cvo_vnet_id	String	(Required) The name of the virtual network.
az_cvo_vnet_resource_group	String	(Required) The resource group in Azure associated to the virtual network.
az_cvo_data_encryption_type	String	(Required) The type of encryption to use for the working environment: [AZURE, NONE]. The default is AZURE.
az_cvo_storage_type	String	(Required) The type of storage for the first data aggregate: [Premium_LRS, Standard_LRS, StandardSSD_LRS]. The default is Premium_LRS
az_cvo_svm_password	String	(Required) The admin password for Cloud Volumes ONTAP.
az_cvo_workspace_id	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 https://cloudmanager.netapp.com .
az_cvo_capacity_tier	String	(Required) Whether to enable data tiering for the first data aggregate: [Blob, NONE]. The default is BLOB.
az_cvo_writing_speed_state	String	(Required) The write speed setting for Cloud Volumes ONTAP: [NORMAL , HIGH]. The default is NORMAL. This argument is not relevant for HA pairs.
az_cvo_ontap_version	String	(Required) The required ONTAP version. Ignored if 'use_latest_version' is set to true. The default is to use the latest version.
az_cvo_instance_type	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 .
az_cvo_license_type	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.

Name	Type	Description
az_cvo_nss_account	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.
az_tenant_id	String	(Required) Tenant ID of the application/service principal registered in Azure.
az_application_id	String	(Required) Application ID of the application/service principal registered in Azure.
az_application_key	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/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
gcp_connector_deploy_bool	Bool	(Required) Check for Connector deployment.
gcp_connector_name	String	(Required) The name of the Cloud Manager Connector.
gcp_connector_project_id	String	(Required) The GCP project_id where the connector will be created.
gcp_connector_zone	String	(Required) The GCP zone where the Connector will be created.
gcp_connector_company	String	(Required) The name of the company of the user.
gcp_connector_service_account_email	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.
gcp_connector_service_account_path	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.
gcp_connector_account_id	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 https://cloudmanager.netapp.com .

Single Node Instance

Terraform variables for single NetApp CVO instance on GCP.

Name	Type	Description
gcp_cvo_name	String	(Required) The name of the Cloud Volumes ONTAP working environment.

Name	Type	Description
gcp_cvo_project_id	String	(Required) The ID of the GCP project.
gcp_cvo_zone	String	(Required) The zone of the region where the working environment will be created.
gcp_cvo_gcp_service_account	String	(Required) The gcp_service_account email in order to enable tiering of cold data to Google Cloud Storage.
gcp_cvo_svm_password	String	(Required) The admin password for Cloud Volumes ONTAP.
gcp_cvo_workspace_id	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 https://cloudmanager.netapp.com .
gcp_cvo_license_type	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.
gcp_cvo_capacity_package_name	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/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
gcp_connector_deploy_bool	Bool	(Required) Check for Connector deployment.
gcp_connector_name	String	(Required) The name of the Cloud Manager Connector.
gcp_connector_project_id	String	(Required) The GCP project_id where the connector will be created.
gcp_connector_zone	String	(Required) The GCP zone where the Connector will be created.

Name	Type	Description
gcp_connector_company	String	(Required) The name of the company of the user.
gcp_connector_service_account_email	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.
gcp_connector_service_account_path	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.
gcp_connector_account_id	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 https://cloudmanager.netapp.com .

HA Pair

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

Name	Type	Description
gcp_cvo_is_ha	Bool	(Optional) Indicate whether the working environment is an HA pair or not [true, false]. The default is false.
gcp_cvo_name	String	(Required) The name of the Cloud Volumes ONTAP working environment.
gcp_cvo_project_id	String	(Required) The ID of the GCP project.
gcp_cvo_zone	String	(Required) The zone of the region where the working environment will be created.
gcp_cvo_node1_zone	String	(Optional) Zone for node 1.
gcp_cvo_node2_zone	String	(Optional) Zone for node 2.
gcp_cvo_mediator_zone	String	(Optional) Zone for mediator.
gcp_cvo_vpc_id	String	(Optional) The name of the VPC.
gcp_cvo_subnet_id	String	(Optional) The name of the subnet for Cloud Volumes ONTAP. The default is: 'default'.
gcp_cvo_vpc0_node_and_data_connectivity	String	(Optional) VPC path for nic1, required for node and data connectivity. If using shared VPC, network_project_id must be provided.
gcp_cvo_vpc1_cluster_connectivity	String	(Optional) VPC path for nic2, required for cluster connectivity.

Name	Type	Description
gcp_cvo_vpc2_ha_connectivity	String	(Optional) VPC path for nic3, required for HA connectivity.
gcp_cvo_vpc3_data_replication	String	(Optional) VPC path for nic4, required for data replication.
gcp_cvo_subnet0_node_and_data_connectivity	String	(Optional) Subnet path for nic1, required for node and data connectivity. If using shared VPC, network_project_id must be provided.
gcp_cvo_subnet1_cluster_connectivity	String	(Optional) Subnet path for nic2, required for cluster connectivity.
gcp_cvo_subnet2_ha_connectivity	String	(Optional) Subnet path for nic3, required for HA connectivity.
gcp_cvo_subnet3_data_replication	String	(Optional) Subnet path for nic4, required for data replication.
gcp_cvo_gcp_service_account	String	(Required) The gcp_service_account email in order to enable tiering of cold data to Google Cloud Storage.
gcp_cvo_svm_password	String	(Required) The admin password for Cloud Volumes ONTAP.
gcp_cvo_workspace_id	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 https://cloudmanager.netapp.com .
gcp_cvo_license_type	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.
gcp_cvo_capacity_package_name	String	(Optional) The capacity package name: ['Essential', 'Professional', 'Freemium']. Default is 'Essential'.
gcp_cvo_gcp_volume_size	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'.
gcp_cvo_gcp_volume_size_unit	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/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
gcp_cvs_name	String	(Required) The name of the NetApp CVS volume.
gcp_cvs_project_id	String	(Required) The GCP project_id where the CVS Volume will be created.
gcp_cvs_gcp_service_account_path	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.
gcp_cvs_region	String	(Required) The GCP zone where the CVS Volume will be created.
gcp_cvs_network	String	(Required) The network VPC of the volume.
gcp_cvs_size	Integer	(Required) The size of volume is between 1024 to 102400 inclusive (in GiB).
gcp_cvs_volume_path	String	(Optional) The name of the volume path for volume.
gcp_cvs_protocol_types	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

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 creating 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 warnings and resizing notifications via email
- Ability to delete snapshots older than the user-defined threshold
- Ability to get a list of FlexClone volumes and snapshots associated
- Ability to monitor the checks at a regular interval
- Ability to use the solution with or without internet access
- Ability to deploy manually or using AWS CloudFormation Template
- Ability to monitor multiple FSx for ONTAP filesystems in a single VPC

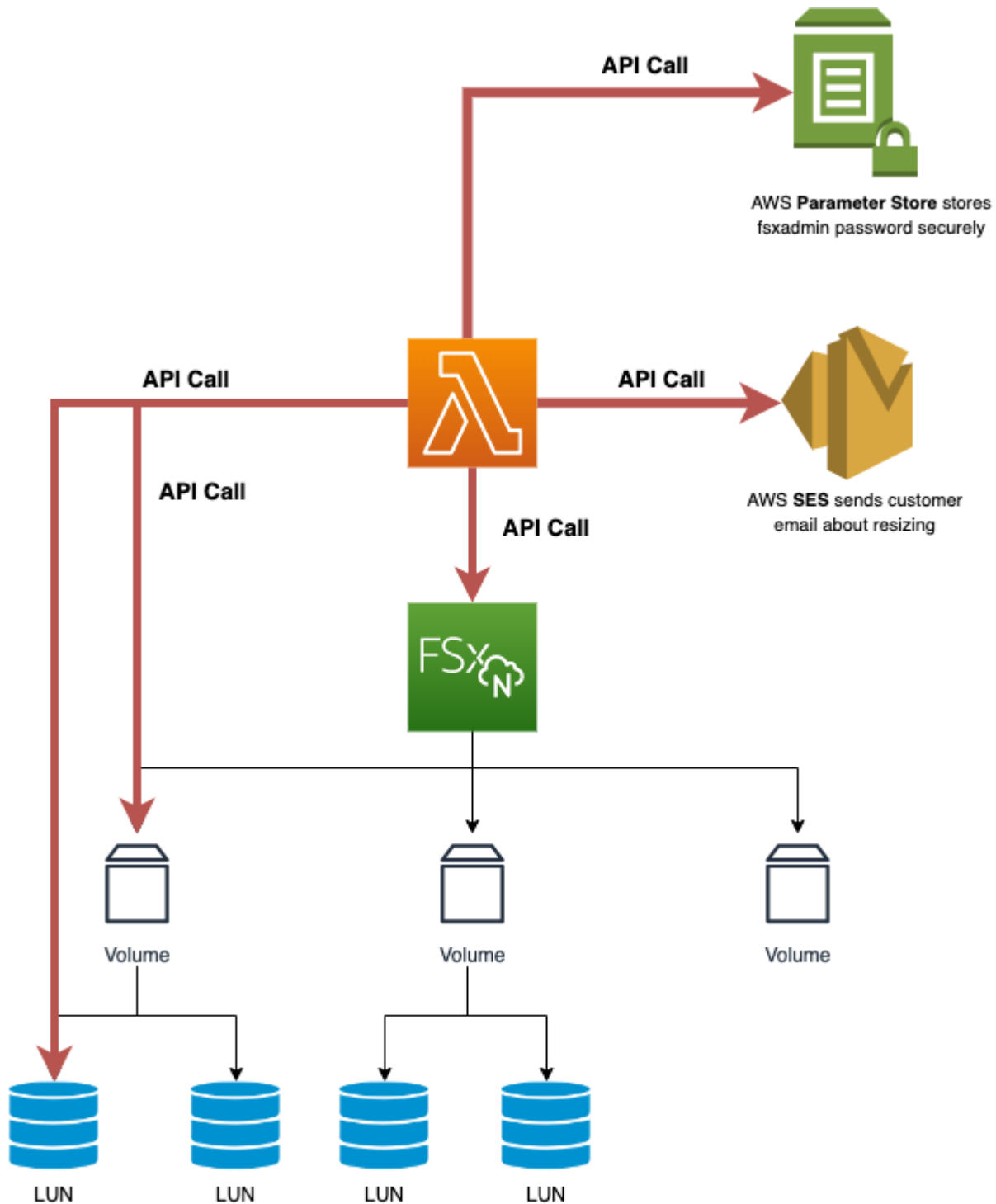
Pre-requisites

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

- FSx for ONTAP is deployed
- Private subnet with 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 an added layer of security.
- AWS SES (Simple Email Service) is used to notify end-users when a resizing event occurs.
- If deploying the solution in a VPC without internet access, VPC Endpoints for AWS SSM, FSx and SES are set up to allow Lambda to reach these services via AWS internal network.



Solution Deployment

Automated Deployment



Covers single FSx for ONTAP filesystem.

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

Step 1: Clone the GitHub repository

Clone the GitHub repository on your local system:

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

Step 2: Setup an AWS S3 bucket

1. Navigate to AWS Console > **S3** and click on **Create bucket**. Create the bucket with the default settings.
2. Once inside the bucket, click on **Upload** > **Add files** and select **Utilities.zip** from the cloned GitHub repository on your system.

The screenshot shows the AWS S3 console interface. At the top, there's a navigation bar with the AWS logo, 'Services', a search bar, and a '[Option+S]' shortcut. Below the navigation bar, the breadcrumb trail reads 'Amazon S3 > Buckets > vedant-fsx-bucket > Upload'. The main heading is 'Upload' with an 'Info' link. A descriptive text states: 'Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. [Learn more](#)'. Below this is a dashed box with the instruction: 'Drag and drop files and folders you want to upload here, or choose **Add files** or **Add folder**.' The 'Files and folders' section shows '1 Total, 27.4 MB'. It includes a 'Remove' button, 'Add files' and 'Add folder' buttons, and a search bar labeled 'Find by name'. A table lists the uploaded file:

<input type="checkbox"/>	Name	Folder	Type	Size
<input type="checkbox"/>	Utilities.zip	-	application/zip	27.4 MB

Below the table, the 'Destination' section shows the path 's3://vedant-fsx-bucket'. A 'Destination details' section is partially visible, indicating it shows bucket settings that impact new objects stored in the specified destination.

Step 3: AWS SES SMTP Setup (required if no internet access available)

Follow this step if you want to deploy the solution without internet access (Note: There will be added costs associated with VPC endpoints being set up.)

1. Navigate to AWS Console > **AWS Simple Email Service (SES)** > **SMTP Settings** and click on **Create SMTP credentials**
2. Enter an IAM User Name or leave it at the default value and click on **Create User**. Save the **SMTP user name** and **SMTP password** for further use.



Skip this step if SES SMTP setup is already in place.

Specify user details

Create user for SMTP

Create an IAM user with SMTP credentials for SMTP authentication with Amazon SES.

User name

ses-smtp-user.20230904-201255

The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and * , _ , @ , - (hyphen)

Permissions policy for user

This permissions policy gives the user permissions to access AWS SES.

```
1- {  
2-   "Version": "2012-10-17",  
3-   "Statement": [  
4-     {  
5-       "Effect": "Allow",  
6-       "Action": "ses:SendRawEmail",  
7-       "Resource": "*" }  
8-   ]  
9- }  
10 }
```

Tags - optional

Tags are key-value pairs you can add to AWS resources to help identify, organize, or search for resources. Choose any tags you want to associate with this user.

No tags associated with the resource.

Add new tag

You can add up to 50 more tags.

Cancel

Create user

Step 4: AWS CloudFormation Deployment

1. Navigate to AWS Console > **CloudFormation** > Create stack > With New Resources (Standard).

```
Prepare template: Template is ready
Specify template: Upload a template file
Choose file: Browse to the cloned GitHub repo and select fsxn-
monitoring-solution.yaml
```

The screenshot shows the 'Create stack' wizard in the AWS CloudFormation console. The left sidebar indicates the current step is 'Step 1: Create stack'. The main content area is titled 'Create stack' and contains a 'Prerequisite - Prepare template' section. Under 'Prepare template', the 'Template is ready' radio button is selected. Below this is the 'Specify template' section, which includes a 'Template source' section where 'Upload a template file' is selected. The 'Upload a template file' section shows a 'Choose file' button and the filename 'fsxn-monitoring-solution.yaml'. At the bottom, the 'S3 URL' is displayed, and a 'View in Designer' button is available. 'Cancel' and 'Next' buttons are at the bottom right.

Click on Next

2. Enter the stack details. Click on Next and check the checkbox for "I acknowledge that AWS CloudFormation might create IAM resources" and click on Submit.



If "Does VPC have internet access?" is set to False, "SMTP Username for AWS SES" and "SMTP Password for AWS SES" are required. Otherwise, they can be left empty.

The screenshot shows the 'Specify stack details' step of the 'Create stack' wizard. The left sidebar indicates the current step is 'Step 2: Specify stack details'. The main content area is titled 'Specify stack details' and contains a 'Stack name' section where 'DemoFxnMonitoringSolution' is entered. Below this is the 'Parameters' section, which includes a 'Network Configuration' section. Under 'Network Configuration', there are three dropdown menus: 'VPC', 'Private Subnet 1', and 'Private Subnet 2'. The 'VPC' dropdown is currently selected. At the bottom, there are 'Cancel' and 'Next' buttons.

FSx for ONTAP Configuration

Management IP address
Enter the "Management endpoint - IP address" from the FSx for ONTAP console on AWS.

10.10.10.10

File System ID
Enter the "File system ID" from the FSx for ONTAP console on AWS.

fs-██████████

ONTAP Administrator Username
Enter the FSx for ONTAP "ONTAP administrator username" from FSx for ONTAP console on AWS.

fsxadmin

Password for ONTAP Administrator Account
Enter the password set for ONTAP Administrator user for FSx for ONTAP.

Resize Threshold (%)
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.

90

Enable Warning Notifications
Set this variable to True to receive notification when Storage Capacity/Volume/LUN usage exceeds 75% but is less than threshold.

True

Enable Snapshot Deletion
Set this variable to True to enable volume level snapshot deletion for snapshots older than the value specified in "Snapshot Age Threshold for Deletion (No. of Days)".

True

Snapshot Age Threshold for Deletion (No. of Days)
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.

30

General Configuration

S3 Bucket Name
Enter the name of the S3 Bucket where paramiko.zip and requests.zip is uploaded. Ensure S3 key for paramiko.zip is paramiko.zip and for requests.zip is requests.zip.

DemoFSxMonitoringBucket

Does VPC have Internet access?
Set this variable to True if the VPC used for deploying this solution has access to Internet. Set to False otherwise.

True

Does SSM VPC Endpoint already exist for the selected VPC?
If Internet access is not available, set this variable to True if the VPC Endpoint for SSM already exists in the VPC. Set to False otherwise.

False

Does FSx VPC Endpoint already exist for the selected VPC?
If Internet access is not available, set this variable to True if the VPC Endpoint for FSx already exists in the VPC. Set to False otherwise.

False

Does SES VPC Endpoint already exist for the selected VPC?
If Internet access is not available, set this variable to True if the VPC Endpoint for SES already exists in the VPC. Set to False otherwise.

False

SMTP Username for AWS SES
If Internet access is not available, enter the smtp username for AWS SES.

Enter String

SMTP Password for AWS SES
If Internet access is not available, enter the smtp password for AWS SES.

Enter String

Sender Email ID
Enter the email ID registered on SES that will be used by the lambda function to send notification alerts related to monitoring and resizing.

abc@xyz.com

Receiver Email ID
Enter the email ID on which you want to receive the alert notifications.

abc@xyz.com

Schedule Expression for frequency of running the solution
Self-trigger your target on an automated schedule using Cron or rate expressions. Cron expressions are in UTC. e.g. rate(1 day), cron(0 17 ? * MON-FRI ?).

rate(1 day)

Cancel Previous **Next**

- Once the CloudFormation deployment starts, the email ID mentioned in the "sender email ID" will get an email asking them to authorize the use of the email address with AWS SES. Click on the link to verify the email address.
- Once the CloudFormation stack deployment is completed, if there are any warnings/notifications, an email will be sent to the recipient email ID with the notification details.

FSx for ONTAP Monitoring

File System Storage Capacity Notification

Storage Capacity used is greater than 90%. File System Storage Capacity resized to: 1240 GB

Volume Notification

Volume Name	Use %	Notification Type	Updated Size
clonevol3	88.39%	Warning	
vol2	88.39%	Warning	
clonevol2	88.39%	Warning	
vol1	78.43%	Warning	

Snapshot Notification

Snapshot Name	Volume Name	Snapshot Age	Space Freed Up	Status
clone_clonevol2.2023-03-22_095434.0	vol2	1 day	296KB	Deleted
clone_clonevol3.2023-03-22_170720.0	clonevol2	1 day	392KB	Deleted

Clone Information

Volume Name	Parent Snapshot	Snapshot Size
clonevol2	clone_clonevol3.2023-03-22_170720.0	392.0KB
vol2	clone_clonevol2.2023-03-22_095434.0	296.0KB

Manual Deployment



Supports monitoring multiple FSx for ONTAP filesystems in a single VPC.

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

Step 1: Clone the GitHub repository

Clone the GitHub repository on your local system:

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

Step 2: AWS SES SMTP Setup (required if no internet access available)

Follow this step if you want to deploy the solution without internet access (Note: There will be added costs associated with VPC endpoints being set up.)

1. Navigate to AWS Console > **AWS Simple Email Service (SES)** > SMTP Settings and click on **Create SMTP credentials**
2. Enter an IAM User Name or leave it at the default value and click on Create. Save the username and password for further use.

Specify user details

Create user for SMTP
Create an IAM user with SMTP credentials for SMTP authentication with Amazon SES.

User name
ses-smtp-user.20230904-201255
The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and * + , @ _ - (hyphen)

Permissions policy for user
This permissions policy gives the user permissions to access AWS SES.

```
1 {  
2   "Version": "2012-10-17",  
3   "Statement": [  
4     {  
5       "Effect": "Allow",  
6       "Action": "ses:SendRawEmail",  
7       "Resource": "*" }  
8   ]  
9 }  
10 }
```

Tags - optional
Tags are key-value pairs you can add to AWS resources to help identify, organize, or search for resources. Choose any tags you want to associate with this user.

No tags associated with the resource.

Add new tag
You can add up to 50 more tags.

Cancel Create user

Step 3: 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**.

Repeat the above steps for all FSx for ONTAP filesystems to be monitored.

The screenshot shows the AWS Console interface for creating a new parameter. The breadcrumb trail is 'AWS Systems Manager > Parameter Store > Create parameter'. The 'Create parameter' page has a 'Parameter details' section with the following 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 has 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 has 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 note: 'Use the default KMS key for this account or specify a customer-managed key for this account. [Learn more](#)'. The 'Another account' option has a note: 'Use a KMS key from another account [Learn more](#)'.
- KMS Key ID:** A dropdown menu showing 'alias/aws/ssm'.
- Value:** A text input field with masked characters (asterisks).

A warning message is displayed below the KMS Key ID field: '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](#)'.

Perform the same steps for storing smtp username and smtp password if deploying the solution without internet access. Otherwise, skip adding these 2 parameters.

Step 4: 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 email ID mentioned in the "sender email ID" will get an email asking the owner to authorize the use of the email address with AWS SES. Click on the link to verify the email address.

aws Services Search [Option+S]
Resource Groups & Tag Editor

Amazon SES > Configuration: Verified identities > Create identity

Create identity

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.

Identity details [Info](#)

Identity type

☐ 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 [Info](#)

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 5: Setup VPC Endpoints (required if no internet access is available)



Required only if deployed without internet access. There will be additional costs associated with VPC endpoints.

1. Navigate to AWS Console > **VPC** > **Endpoints** and click on **Create Endpoint** and enter the following details:

```
Name: <Any name for the vpc endpoint>
Service category: AWS Services
Services: com.amazonaws.<region>.fsx
vpc: <select the vpc where lambda will be deployed>
subnets: <select the subnets where lambda will be deployed>
Security groups: <select the security group>
Policy: <Either choose Full access or set your own custom policy>
```

Click on Create endpoint.

[VPC](#) > [Endpoints](#) > Create endpoint

Create endpoint [Info](#)

There are three types of VPC endpoints – Interface endpoints, Gateway Load Balancer endpoints, and Gateway endpoints. Interface endpoints and Gateway Load Balancer endpoints are powered by AWS PrivateLink, and use an Elastic Network Interface (ENI) as an entry point for traffic destined to the service. Interface endpoints are typically accessed using the public or private DNS name associated with the service, while Gateway endpoints and Gateway Load Balancer endpoints serve as a target for a route in your route table for traffic destined for the service.

Endpoint settings

Name tag - optional

Creates a tag with a key of 'Name' and a value that you specify.

Service category

Select the service category



AWS services

Services provided by Amazon



PrivateLink Ready partner services

Services with an AWS Service Ready designation



AWS Marketplace services


Services that you've purchased through AWS Marketplace



Other endpoint services

Find services shared with you by service name

Services (1/1)

< 1 >  

Service Name	Owner	Type
 com.amazonaws.us-west-1.fsx	amazon	Interface

VPC

Select the VPC in which to create the endpoint

VPC

The VPC in which to create your endpoint.

vpc- [REDACTED] (DemoFSxN-vpc) ▼



► Additional settings

Subnets (2/2) Info

<input checked="" type="checkbox"/>	Availability Zone ▼	Subnet ID ▼
<input checked="" type="checkbox"/>	us-west-1a (usw1-az3)	subnet-[REDACTED] ▼
<input checked="" type="checkbox"/>	us-west-1b (usw1-az1)	subnet-[REDACTED] ▼

subnet-[REDACTED] ✕
DemoFSxN-subnet-private1-us-west-1a

subnet-[REDACTED] ✕
DemoFSxN-subnet-private2-us-west-1b

IP address type

- ☒ IPv4
☐ IPv6
☐ Dualstack

Security groups (1/1) Info

Find resources by attribute or tag

< 1 > ⚙

<input checked="" type="checkbox"/>	Group ID ▼	Group name ▼	VPC ID ▼
<input checked="" type="checkbox"/>	sg-[REDACTED]	default	vpc-[REDACTED]

sg-[REDACTED] ✕

- Follow the same process for creating SES and SSM VPC endpoints. All parameters remain the same as above except Services which will correspond to **com.amazonaws.<region>.smtp** and **com.amazonaws.<region>.ssm** respectively.

Step 6: Create and setup the AWS Lambda Function

1. Navigate to AWS Console > **AWS Lambda > Functions** 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 that 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 a hint to use lowercase letters, numbers, hyphens, or underscores), 'Runtime' (set to 'Python 3.9'), and 'Architecture' (set to 'x86_64'). The 'Permissions' section shows a message: 'By default, Lambda roles are created with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role after adding triggers.' Below this, there is a section for 'Change default execution role' with options to 'Create a new role with basic Lambda permissions' (selected), 'Use an existing role', or 'Create a new role from AWS policy templates'. A note at the bottom states: 'Lambda will create an execution role named `SamFunction-role-20250221`, 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] [X] [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. Navigate to the newly created Lambda function > Scroll down to the **Layers** section 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 a Layer and upload **Utilities.zip** file. Select **Python 3.9** as the compatible runtime and click on **Create**.

aws

Services

Search

[Option+S]

Lambda > Layers > Create layer

Create layer

Layer configuration

Name

Utilities

Description - optional

Paramiko and Requests Libraries

☒ Upload a .zip file

☐ Upload a file from Amazon S3

Upload

Utilities.zip
28.70 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 function > **Add Layer** > **Custom Layers** and add the utilities layer.

Services
Search
[Option+S]

Lambda > Layers > Add layer

Add layer

Function runtime settings

Runtime	Architecture
Python 3.9	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.

Utilities

Version
2

Cancel
Add

Layers

[Info](#)

Edit
Add a layer

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

- 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
fsxList	List	(Required) List of all FSx for ONTAP filesystems to be monitored. Include all the file systems in the list for monitoring and auto-resizing.
fsxMgmtIp	String	(Required) Enter the "Management endpoint - IP address" from the FSx for ONTAP console on AWS.
fsxId	String	(Required) Enter the "File system ID" from the FSx for ONTAP console on AWS.
username	String	(Required) Enter the FSx for ONTAP "ONTAP administrator username" from FSx for ONTAP console on AWS.
resize_threshold	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.
fsx_password_ssm_parameter	String	(Required) Enter the path name used in AWS Parameter Store for storing "fsxadmin" password.
warn_notification	Bool	(Required) Set this variable to True to receive a notification when Storage Capacity/Volume/LUN usage exceeds 75% but is less than the threshold.
enable_snapshot_deletion	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".
snapshot_age_threshold_in_days	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.

internet_access	Bool	(Required) Set this variable to True if internet access is available from the subnet where this lambda is deployed. Otherwise set it to False.
smtp_region	String	(Optional) If "internet_access" variable is set to False, enter the region in which lambda is deployed. E.g. us-east-1 (in this format)
smtp_username_ssm_parameter	String	(Optional) If "internet_access" variable is set to False, enter the path name used in AWS Parameter Store for storing the SMTP username.
smtp_password_ssm_parameter	String	(Optional) If "internet_access" variable is set to False, enter the path name used in AWS Parameter Store for storing the SMTP password.
sender_email	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.
recipient_email	String	(Required) Enter the email ID on which you want to receive the alert notifications.

```

1 # Copyright NetApp 2023. Developed by NetApp Solutions Engineering Team
2 #
3 # Description: This Lambda function can be used to automate the monitoring and autoscaling of
4 # FSX ONTAP Storage Capacity based on volume usage.
5 # Pre-requisites for running this template
6 # - Create lambda in a private subnet with nat gateway for internet access
7 # - Update all values in vars.py with the required inputs.
8 # - FSX File System and Volumes should be created with FSX API password set.
9 # - Ensure that Lambda function has connectivity to FSX by attaching FSX VPC and subnets to lambda with appropriate security group.
10 # - Increase the default timeout for Lambda function from 3 secs. to 5 mins.
11 # - Create a new requests layer and a prometheus layer for this lambda function with python 3.9. (Use the zip files to create the layers)
12 # - Create a new trigger with event bridge and select schedule expression and enter rate(1 day) to ensure this function runs once every day.
13 # - Allow lambda to create the default role while creating lambda function and add a new inline policy using "policy.txt" file in this folder. Ensure to replace "${AWS::AccountID}" with your account ID.
14 # - Ensure that the Sender Email is verified in Amazon SES before using it in this lambda function.
15 # - Save the password for fsxadmin in SSM parameter store and provide the path in fsx_password_ssm_parameter variable in vars.py.
16
17 import json
18 import requests
19 requests.packages.urllib3.disable_warnings()
20 import base64
21 import logging
22 import vars
23 import math
24 import time
25 logger = logging.getLogger()
26 logger.setLevel(logging.INFO)
27 import boto3
28 import boto3.ec2
29 import re
30
31 def lambda_handler(event, context):
32     #Retrieve fsx password

```

- Click on **Test**, create a test event with an empty JSON object and run the test by clicking **Invoke** to check if the script is running properly.
- 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.

The screenshot shows the 'Add trigger' configuration page in the AWS Lambda console. The page is titled 'Add trigger' and is part of the 'Resource Groups & Tag Editor' interface. The 'Trigger configuration' section is active, showing the 'EventBridge (CloudWatch Events)' source. Under the 'Rule' section, the 'Create a new rule' option is selected. The 'Rule name' field is filled with 'DemoFSxNRule'. The 'Rule description' field is empty. Under the 'Rule type' section, the 'Schedule expression' option is selected. The 'Schedule expression' field is filled with 'rate(1 day)'. Below the field, there is a note: 'Self-trigger your target on an automated schedule using [Cron or rate expressions](#). Cron expressions are in UTC. e.g. rate(1 day), cron(0 17 ? * MON-FRI *)'. At the bottom, a message states: '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 set up a monitoring solution that regularly monitors FSx for ONTAP Storage, resizes it based on a 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 © 2024 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.