Solution Automation
NetApp Solutions
NetApp
May 01, 2024
# 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)

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

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

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

2. Paste the below content in the Terminal

   ```bash
   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

```bash
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:

   ```yaml
   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:

   ```yaml
   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.

   ![Access keys](image)

   **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](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: EAAfPTMCu4QjI9hR2PTrT75Lswr0fHp4BheEJ2XFsHt
```

Acquiring Client ID

1. Access the API page to copy Client ID at [https://services.cloud.netapp.com/developer-hub](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".
Acquiring Key Pair from AWS

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

How to Acquire a JWT Access Token via regular token access

1. Make an HTTP POST request to the endpoint
   https://netapp-cloud-account.actie8.com/oauth/token
   Include the header Content-Type: application/json
   Include the body:

   ```json
   {
     "grant_type": "password",
     "username": "YOUR_EMAIL_ADDRESS",
     "password": "YOUR_PASSWORD",
     "audience": "https://api.cloud.netapp.com",
     "client_id": 
   }
   ```

Acquiring Key Pair from AWS

1. In AWS console, search for “Key Pair” and create a key pair with “pem”. Remember the name of your key_pair, we will use it to deploy the connector.
Acquiring Account ID

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

```
account-LwO1qR76
Account ID
```

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.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>aws_connctor_deployment_bool</code></td>
<td>Bool</td>
<td>(Required) Check for Connector deployment.</td>
</tr>
<tr>
<td><code>aws_connctor_name</code></td>
<td>String</td>
<td>(Required) The name of the Cloud Manager Connector.</td>
</tr>
<tr>
<td><code>aws_connctor_region</code></td>
<td>String</td>
<td>(Required) The region where the Cloud Manager Connector will be created.</td>
</tr>
<tr>
<td><code>aws_connctor_key_name</code></td>
<td>String</td>
<td>(Required) The name of the key pair to use for the Connector instance.</td>
</tr>
<tr>
<td><code>aws_connctor_company</code></td>
<td>String</td>
<td>(Required) The name of the company of the user.</td>
</tr>
<tr>
<td><code>aws_connctor_instance_type</code></td>
<td>String</td>
<td>(Required) The type of instance (for example, t3.xlarge). At least 4 CPU and 16 GB of memory are required.</td>
</tr>
<tr>
<td><code>aws_connctor_subnet_id</code></td>
<td>String</td>
<td>(Required) The ID of the subnet for the instance.</td>
</tr>
<tr>
<td><code>aws_connctor_security_group_id</code></td>
<td>String</td>
<td>(Required) The ID of the security group for the instance, multiple security groups can be provided separated by ','.</td>
</tr>
<tr>
<td><code>aws_connctor_iam_instance_profile_name</code></td>
<td>String</td>
<td>(Required) The name of the instance profile for the Connector.</td>
</tr>
</tbody>
</table>
aws_connector_account_id

(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

(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.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cvo_name</td>
<td>String</td>
<td>(Required) The name of the Cloud Volumes ONTAP working environment.</td>
</tr>
<tr>
<td>cvo_region</td>
<td>String</td>
<td>(Required) The region where the working environment will be created.</td>
</tr>
<tr>
<td>cvo_subnet_id</td>
<td>String</td>
<td>(Required) The subnet id where the working environment will be created.</td>
</tr>
<tr>
<td>cvo_vpc_id</td>
<td>String</td>
<td>(Optional) The VPC ID where the working environment will be created.</td>
</tr>
<tr>
<td>cvo_svm_password</td>
<td>String</td>
<td>(Required) The admin password for Cloud Volumes ONTAP.</td>
</tr>
<tr>
<td>cvo_writing_speed_state</td>
<td>String</td>
<td>(Optional) The write speed setting for Cloud Volumes ONTAP: ['NORMAL','HIGH']. The default is 'NORMAL'.</td>
</tr>
</tbody>
</table>

CVO HA Deployment

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

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

Terraforn 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
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
```
## Connector

Terraform variables for NetApp AWS connector instance for CVO deployment.

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

## HA Pair

Terraform variables for NetApp CVO instances in HA Pair.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cvo_is_ha</td>
<td>Bool</td>
<td>(Optional) Indicate whether the working environment is an HA pair or not [true, false]. The default is false.</td>
</tr>
<tr>
<td>cvo_name</td>
<td>String</td>
<td>(Required) The name of the Cloud Volumes ONTAP working environment.</td>
</tr>
<tr>
<td>cvo_region</td>
<td>String</td>
<td>(Required) The region where the working environment will be created.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>--------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>cvo_node1_subnet_id</td>
<td>String</td>
<td>(Required) The subnet id where the first node will be created.</td>
</tr>
<tr>
<td>cvo_node2_subnet_id</td>
<td>String</td>
<td>(Required) The subnet id where the second node will be created.</td>
</tr>
<tr>
<td>cvo_vpc_id</td>
<td>String</td>
<td>(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.</td>
</tr>
<tr>
<td>cvo_svm_password</td>
<td>String</td>
<td>(Required) The admin password for Cloud Volumes ONTAP.</td>
</tr>
<tr>
<td>cvo.failover_mode</td>
<td>String</td>
<td>(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.</td>
</tr>
<tr>
<td>cvo_mediator_subnet_id</td>
<td>String</td>
<td>(Optional) For HA, the subnet ID of the mediator.</td>
</tr>
<tr>
<td>cvo_mediator_key_pair_name</td>
<td>String</td>
<td>(Optional) For HA, the key pair name for the mediator instance.</td>
</tr>
<tr>
<td>cvo_cluster_floating_ip</td>
<td>String</td>
<td>(Optional) For HA FloatingIP, the cluster management floating IP address.</td>
</tr>
<tr>
<td>cvo_data_floating_ip</td>
<td>String</td>
<td>(Optional) For HA FloatingIP, the data floating IP address.</td>
</tr>
<tr>
<td>cvo_data_floating_ip2</td>
<td>String</td>
<td>(Optional) For HA FloatingIP, the data floating IP address.</td>
</tr>
<tr>
<td>cvo_svm_floating_ip</td>
<td>String</td>
<td>(Optional) For HA FloatingIP, the SVM management floating IP address.</td>
</tr>
<tr>
<td>cvo_route_table_ids</td>
<td>List</td>
<td>(Optional) For HA FloatingIP, the list of route table IDs that will be updated with the floating IPs.</td>
</tr>
</tbody>
</table>

**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).


**Procedure**

In order to run the template:

a. Clone the repository.

```bash
    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.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>aws_connector_deploy_bool</td>
<td>Bool</td>
<td>(Required) Check for Connector deployment.</td>
</tr>
<tr>
<td>aws.connector_name</td>
<td>String</td>
<td>(Required) The name of the Cloud Manager Connector.</td>
</tr>
<tr>
<td>aws.connector_region</td>
<td>String</td>
<td>(Required) The region where the Cloud Manager Connector will be created.</td>
</tr>
<tr>
<td>aws.connector_key_name</td>
<td>String</td>
<td>(Required) The name of the key pair to use for the Connector instance.</td>
</tr>
<tr>
<td>aws.connector_company</td>
<td>String</td>
<td>(Required) The name of the company of the user.</td>
</tr>
<tr>
<td>aws.connector_instance_type</td>
<td>String</td>
<td>(Required) The type of instance (for example, t3.xlarge). At least 4 CPU and 16 GB of memory are required.</td>
</tr>
<tr>
<td>aws.connector_subnet_id</td>
<td>String</td>
<td>(Required) The ID of the subnet for the instance.</td>
</tr>
</tbody>
</table>
| 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](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.

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

```bash
git clone https://github.com/NetApp/na_cloud_volumes_automation.git
```

b. Navigate to the desired folder

```bash
cd na_cloud_volumes_automation
```

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

```bash
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.

```bash
terraform init
```

f. Verify the terraform files using terraform validate command.

```bash
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

Recipes:
Single Node Instance

Terraform variables for single NetApp ANF Volume.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>az_location</td>
<td>String</td>
<td>(Required) Specifies the supported Azure location where the resource exists. Changing this forces a new resource to be created.</td>
</tr>
<tr>
<td>az_prefix</td>
<td>String</td>
<td>(Required) The name of the resource group where the NetApp Volume should be created. Changing this forces a new resource to be created.</td>
</tr>
<tr>
<td>az_vnet_address_space</td>
<td>String</td>
<td>(Required) The address space to be used by the newly created vnet for ANF volume deployment.</td>
</tr>
<tr>
<td>az_subnet_address_prefix</td>
<td>String</td>
<td>(Required) The subnet address prefix to be used by the newly created vnet for ANF volume deployment.</td>
</tr>
<tr>
<td>az_volume_path</td>
<td>String</td>
<td>(Required) A unique file path for the volume. Used when creating mount targets. Changing this forces a new resource to be created.</td>
</tr>
<tr>
<td>az_capacity_pool_size</td>
<td>Integer</td>
<td>(Required) Capacity Pool Size mentioned in TB.</td>
</tr>
<tr>
<td>az_vnet_creation_bool</td>
<td>Boolean</td>
<td>(Required) Set this boolean to true if you want to create a new vnet. Set it to false to use an existing vnet.</td>
</tr>
<tr>
<td>az_subnet_creation_bool</td>
<td>Boolean</td>
<td>(Required) Set this boolean to true to create a new subnet. Set it to false to use an existing subnet.</td>
</tr>
<tr>
<td>az_subnet_id_for_anf_vol</td>
<td>String</td>
<td>(Required) Mention the subnet id in case you decide to use an existing subnet by setting subnet_creation_bool to true. If set to false, leave it at the default value.</td>
</tr>
<tr>
<td>az_netapp_pool_service_level</td>
<td>String</td>
<td>(Required) The target performance of the file system. Valid values include Premium, Standard, or Ultra.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>--------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>az_netapp_vol_service_level</td>
<td>String</td>
<td>(Required) The target performance of the file system. Valid values include Premium, Standard, or Ultra.</td>
</tr>
<tr>
<td>az_netapp_vol_protocol</td>
<td>String</td>
<td>(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.</td>
</tr>
<tr>
<td>az_netapp_vol_security_style</td>
<td>String</td>
<td>(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.</td>
</tr>
<tr>
<td>az_netapp_vol_storage_quota</td>
<td>String</td>
<td>(Required) The maximum Storage Quota allowed for a file system in Gigabytes.</td>
</tr>
</tbody>
</table>

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.

```bash
git clone https://github.com/NetApp/na_cloud_volumes_automation.git
```

b. Navigate to the desired folder

```bash
cd na_cloud_volumes_automation
```

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

```bash
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
```

**Recipes:**

**ANF Data Protection**

Terraform variables for single ANF Volume with data protection enabled.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>az_location</td>
<td>String</td>
<td>(Required) Specifies the supported Azure location where the resource exists. Changing this forces a new resource to be created.</td>
</tr>
<tr>
<td>az_alt_location</td>
<td>String</td>
<td>(Required) The Azure location where the secondary volume will be created.</td>
</tr>
<tr>
<td>az_prefix</td>
<td>String</td>
<td>(Required) The name of the resource group where the NetApp Volume should be created. Changing this forces a new resource to be created.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>-----------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>az_vnet_primary_address_space</td>
<td>String</td>
<td>(Required) The address space to be used by the newly created vnet for ANF primary volume deployment.</td>
</tr>
<tr>
<td>az_vnet_secondary_address_space</td>
<td>String</td>
<td>(Required) The address space to be used by the newly created vnet for ANF secondary volume deployment.</td>
</tr>
<tr>
<td>az_subnet_primary_address_prefix</td>
<td>String</td>
<td>(Required) The subnet address prefix to be used by the newly created vnet for ANF primary volume deployment.</td>
</tr>
<tr>
<td>az_subnet_secondary_address_prefix</td>
<td>String</td>
<td>(Required) The subnet address prefix to be used by the newly created vnet for ANF secondary volume deployment.</td>
</tr>
<tr>
<td>az_volume_path_primary</td>
<td>String</td>
<td>(Required) A unique file path for the primary volume. Used when creating mount targets. Changing this forces a new resource to be created.</td>
</tr>
<tr>
<td>az_volume_path_secondary</td>
<td>String</td>
<td>(Required) A unique file path for the secondary volume. Used when creating mount targets. Changing this forces a new resource to be created.</td>
</tr>
<tr>
<td>az_capacity_pool_size_primary</td>
<td>Integer</td>
<td>(Required) Capacity Pool Size mentioned in TB.</td>
</tr>
<tr>
<td>az_capacity_pool_size_secondary</td>
<td>Integer</td>
<td>(Required) Capacity Pool Size mentioned in TB.</td>
</tr>
<tr>
<td>az_vnet_primary_creation_bool</td>
<td>Boolean</td>
<td>(Required) Set this boolean to true if you want to create a new vnet for primary volume. Set it to false to use an existing vnet.</td>
</tr>
<tr>
<td>az_vnet_secondary_creation_bool</td>
<td>Boolean</td>
<td>(Required) Set this boolean to true if you want to create a new vnet for secondary volume. Set it to false to use an existing vnet.</td>
</tr>
<tr>
<td>az_subnet_primary_creation_bool</td>
<td>Boolean</td>
<td>(Required) Set this boolean to true to create a new subnet for primary volume. Set it to false to use an existing subnet.</td>
</tr>
<tr>
<td>az_subnet_secondary_creation_bool</td>
<td>Boolean</td>
<td>(Required) Set this boolean to true to create a new subnet for secondary volume. Set it to false to use an existing subnet.</td>
</tr>
<tr>
<td>az_primary_subnet_id_for_anf_vol</td>
<td>String</td>
<td>(Required) Mention the subnet id in case you decide to use an existing subnet by setting subnet_primary_creation_bool to true. If set to false, leave it at the default value.</td>
</tr>
<tr>
<td>az_secondary_subnet_id_for_anf_vol</td>
<td>String</td>
<td>(Required) Mention the subnet id in case you decide to use an existing subnet by setting subnet_secondary_creation_bool to true. If set to false, leave it at the default value.</td>
</tr>
<tr>
<td>az_netapp_pool_service_level_primary</td>
<td>String</td>
<td>(Required) The target performance of the file system. Valid values include Premium, Standard, or Ultra.</td>
</tr>
<tr>
<td>az_netapp_pool_service_level_secondary</td>
<td>String</td>
<td>(Required) The target performance of the file system. Valid values include Premium, Standard, or Ultra.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>--------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>az_netapp_vol_service_level_primary</td>
<td>String</td>
<td>(Required) The target performance of the file system. Valid values include Premium, Standard, or Ultra.</td>
</tr>
<tr>
<td>az_netapp_vol_service_level_secondary</td>
<td>String</td>
<td>(Required) The target performance of the file system. Valid values include Premium, Standard, or Ultra.</td>
</tr>
<tr>
<td>az_netapp_vol_protocol_primary</td>
<td>String</td>
<td>(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.</td>
</tr>
<tr>
<td>az_netapp_vol_protocol_secondary</td>
<td>String</td>
<td>(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.</td>
</tr>
<tr>
<td>az_netapp_vol_storage_quota_primary</td>
<td>String</td>
<td>(Required) The maximum Storage Quota allowed for a file system in Gigabytes.</td>
</tr>
<tr>
<td>az_netapp_vol_storage_quota_secondary</td>
<td>String</td>
<td>(Required) The maximum Storage Quota allowed for a file system in Gigabytes.</td>
</tr>
<tr>
<td>az_dp_replication_frequency</td>
<td>String</td>
<td>(Required) Replication frequency, supported values are 10minutes, hourly, daily, values are case sensitive.</td>
</tr>
</tbody>
</table>

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](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](https://registry.terraform.io/providers/hashicorp/azurerm/latest/docs)

**Procedure**

In order to run the template:

a. Clone the repository.

    ```shell
    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.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>az_location</td>
<td>String</td>
<td>(Required) Specifies the supported Azure location where the resource exists. Changing this forces a new resource to be created.</td>
</tr>
<tr>
<td>az_prefix</td>
<td>String</td>
<td>(Required) The name of the resource group where the NetApp Volume should be created. Changing this forces a new resource to be created.</td>
</tr>
<tr>
<td>az_vnet_address_space</td>
<td>String</td>
<td>(Required) The address space to be used by the newly created vnet for ANF volume deployment.</td>
</tr>
<tr>
<td>az_subnet_address_prefix</td>
<td>String</td>
<td>(Required) The subnet address prefix to be used by the newly created vnet for ANF volume deployment.</td>
</tr>
<tr>
<td>az_volume_path</td>
<td>String</td>
<td>(Required) A unique file path for the volume. Used when creating mount targets. Changing this forces a new resource to be created.</td>
</tr>
<tr>
<td>az_capacity_pool_size</td>
<td>Integer</td>
<td>(Required) Capacity Pool Size mentioned in TB.</td>
</tr>
<tr>
<td>az_vnet_creation_bool</td>
<td>Boolean</td>
<td>(Required) Set this boolean to true if you want to create a new vnet. Set it to false to use an existing vnet.</td>
</tr>
<tr>
<td>az_subnet_creation_bool</td>
<td>Boolean</td>
<td>(Required) Set this boolean to true to create a new subnet. Set it to false to use an existing subnet.</td>
</tr>
<tr>
<td>az_subnet_id_for_anf_vol</td>
<td>String</td>
<td>(Required) Mention the subnet id in case you decide to use an existing subnet by setting subnet_creation_bool to true. If set to false, leave it at the default value.</td>
</tr>
<tr>
<td>az_netapp_pool_service_level</td>
<td>String</td>
<td>(Required) The target performance of the file system. Valid values include Premium, Standard, or Ultra.</td>
</tr>
<tr>
<td>az_netapp_vol_service_level</td>
<td>String</td>
<td>(Required) The target performance of the file system. Valid values include Premium, Standard, or Ultra.</td>
</tr>
<tr>
<td>az_netapp_vol_protocol1</td>
<td>String</td>
<td>(Required) 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.</td>
</tr>
<tr>
<td>az_netapp_vol_protocol2</td>
<td>String</td>
<td>(Required) 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.</td>
</tr>
<tr>
<td>az_netapp_vol_storage_quota</td>
<td>String</td>
<td>(Required) The maximum Storage Quota allowed for a file system in Gigabytes.</td>
</tr>
<tr>
<td>az_smb_server_username</td>
<td>String</td>
<td>(Required) Username to create ActiveDirectory object.</td>
</tr>
<tr>
<td>az_smb_server_password</td>
<td>String</td>
<td>(Required) User Password to create ActiveDirectory object.</td>
</tr>
<tr>
<td>az_smb_server_name</td>
<td>String</td>
<td>(Required) Server Name to create ActiveDirectory object.</td>
</tr>
</tbody>
</table>
### Name | Type | Description
---|---|---
az_smb_dns_servers| 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](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](https://registry.terraform.io/providers/hashicorp/azurerm/latest/docs)

#### Procedure

In order to run the template:

1. Clone the repository.

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

2. Navigate to the desired folder

   ```
   cd na_cloud_volumes_automation
   ```

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

   ```
   az login
   ```

4. 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.

5. 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.

```bash
terraform validate
```

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

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

d. Run the deployment

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

To delete the deployment

```bash
terraform destroy
```

**Recipes:**

**Single Node Instance**

Terraform variables for single ANF Volume using snapshot.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>az_location</td>
<td>String</td>
<td>(Required) Specifies the supported Azure location where the resource exists. Changing this forces a new resource to be created.</td>
</tr>
<tr>
<td>az_prefix</td>
<td>String</td>
<td>(Required) The name of the resource group where the NetApp Volume should be created. Changing this forces a new resource to be created.</td>
</tr>
<tr>
<td>az_vnet_address_space</td>
<td>String</td>
<td>(Required) The address space to be used by the newly created vnet for ANF volume deployment.</td>
</tr>
<tr>
<td>az_subnet_address_prefix</td>
<td>String</td>
<td>(Required) The subnet address prefix to be used by the newly created vnet for ANF volume deployment.</td>
</tr>
<tr>
<td>az_volume_path</td>
<td>String</td>
<td>(Required) A unique file path for the volume. Used when creating mount targets. Changing this forces a new resource to be created.</td>
</tr>
<tr>
<td>az_capacity_pool_size</td>
<td>Integer</td>
<td>(Required) Capacity Pool Size mentioned in TB.</td>
</tr>
<tr>
<td>az_vnet_creation_bool</td>
<td>Boolean</td>
<td>(Required) Set this boolean to true if you want to create a new vnet. Set it to false to use an existing vnet.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>--------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>az_subnet_creation_bool</td>
<td>Boolean</td>
<td>(Required) Set this boolean to true to create a new subnet. Set it to false to use an existing subnet.</td>
</tr>
<tr>
<td>az_subnet_id_for_anf_vol</td>
<td>String</td>
<td>(Required) Mention the subnet id in case you decide to use an existing subnet by setting subnet_creation_bool to true. If set to false, leave it at the default value.</td>
</tr>
<tr>
<td>az_netapp_pool_service_level</td>
<td>String</td>
<td>(Required) The target performance of the file system. Valid values include Premium, Standard, or Ultra.</td>
</tr>
<tr>
<td>az_netapp_vol_service_level</td>
<td>String</td>
<td>(Required) The target performance of the file system. Valid values include Premium, Standard, or Ultra.</td>
</tr>
<tr>
<td>az_netapp_vol_protocol</td>
<td>String</td>
<td>(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.</td>
</tr>
<tr>
<td>az_netapp_vol_storage_quota</td>
<td>String</td>
<td>(Required) The maximum Storage Quota allowed for a file system in Gigabytes.</td>
</tr>
<tr>
<td>az_snapshot_id</td>
<td>String</td>
<td>(Required) Snapshot ID using which new ANF volume will be created.</td>
</tr>
</tbody>
</table>

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](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.


Procedure

In order to run the template:

a. Clone the repository.

```bash
git clone https://github.com/NetApp/na_cloud_volumes_automation.git
```

b. Navigate to the desired folder

```bash
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.az_cvo_single_node_deployment" -var
-file="vars\azure_cvo_single_node_deployment.tfvars"
```

h. Run the deployment

```
terraform apply
-target="module.az_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).

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>refresh_token</td>
<td>String</td>
<td>(Required) The refresh token of NetApp cloud manager. This can be generated from netapp Cloud Central.</td>
</tr>
<tr>
<td>az_connector_name</td>
<td>String</td>
<td>(Required) The name of the Cloud Manager Connector.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------------</td>
<td>-----------</td>
<td>--------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>az_connector_location</td>
<td>String</td>
<td>(Required) The location where the Cloud Manager Connector will be created.</td>
</tr>
<tr>
<td>az_connector_subscription_id</td>
<td>String</td>
<td>(Required) The ID of the Azure subscription.</td>
</tr>
<tr>
<td>az_connector_company</td>
<td>String</td>
<td>(Required) The name of the company of the user.</td>
</tr>
<tr>
<td>az_connector_resource_group</td>
<td>Integer</td>
<td>(Required) The resource group in Azure where the resources will be created.</td>
</tr>
<tr>
<td>az_connector_subnet_id</td>
<td>String</td>
<td>(Required) The name of the subnet for the virtual machine.</td>
</tr>
<tr>
<td>az_connector_vnet_id</td>
<td>String</td>
<td>(Required) The name of the virtual network.</td>
</tr>
<tr>
<td>az_connector_network_security_group_name</td>
<td>String</td>
<td>(Required) The name of the security group for the instance.</td>
</tr>
<tr>
<td>az_connector_associate_public_ip_address</td>
<td>String</td>
<td>(Required) Indicates whether to associate the public IP address to the virtual machine.</td>
</tr>
<tr>
<td>az_connector_account_id</td>
<td>String</td>
<td>(Required) The NetApp account ID that the Connector will be associated with. If not provided, Cloud Manager uses the first account. If no account exists, Cloud Manager creates a new account. You can find the account ID in the account tab of Cloud Manager at <a href="https://cloudmanager.netapp.com">https://cloudmanager.netapp.com</a>.</td>
</tr>
<tr>
<td>az_connector_admin_password</td>
<td>String</td>
<td>(Required) The password for the Connector.</td>
</tr>
<tr>
<td>az_connector_admin_username</td>
<td>String</td>
<td>(Required) The user name for the Connector.</td>
</tr>
<tr>
<td>az_cvo_name</td>
<td>String</td>
<td>(Required) The name of the Cloud Volumes ONTAP working environment.</td>
</tr>
<tr>
<td>az_cvo_location</td>
<td>String</td>
<td>(Required) The location where the working environment will be created.</td>
</tr>
<tr>
<td>az_cvo_subnet_id</td>
<td>String</td>
<td>(Required) The name of the subnet for the Cloud Volumes ONTAP system.</td>
</tr>
<tr>
<td>az_cvo_vnet_id</td>
<td>String</td>
<td>(Required) The name of the virtual network.</td>
</tr>
<tr>
<td>az_cvo_vnet_resource_group</td>
<td>String</td>
<td>(Required) The resource group in Azure associated to the virtual network.</td>
</tr>
<tr>
<td>az_cvo_data_encryption_type</td>
<td>String</td>
<td>(Required) The type of encryption to use for the working environment: [AZURE, NONE]. The default is AZURE.</td>
</tr>
<tr>
<td>az_cvo_storage_type</td>
<td>String</td>
<td>(Required) The type of storage for the first data aggregate: [Premium_LRS, Standard_LRS, StandardSSD_LRS]. The default is Premium_LRS</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>---------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>az_cvo_svm_password</td>
<td>String</td>
<td>(Required) The admin password for Cloud Volumes ONTAP.</td>
</tr>
<tr>
<td>az_cvo_workspace_id</td>
<td>String</td>
<td>(Required) The ID of the Cloud Manager workspace where you want to deploy Cloud Volumes ONTAP. If not provided, Cloud Manager uses the first workspace. You can find the ID from the Workspace tab on <a href="https://cloudmanager.netapp.com">https://cloudmanager.netapp.com</a>.</td>
</tr>
<tr>
<td>az_cvo_capacity_tier</td>
<td>String</td>
<td>(Required) Whether to enable data tiering for the first data aggregate: [Blob, NONE]. The default is BLOB.</td>
</tr>
<tr>
<td>az_cvo_writing_speed_state</td>
<td>String</td>
<td>(Required) The write speed setting for Cloud Volumes ONTAP: [NORMAL, HIGH]. The default is NORMAL. This argument is not relevant for HA pairs.</td>
</tr>
<tr>
<td>az_cvo_ontap_version</td>
<td>String</td>
<td>(Required) The required ONTAP version. Ignored if 'use_latest_version' is set to true. The default is to use the latest version.</td>
</tr>
<tr>
<td>az_cvo_instance_type</td>
<td>String</td>
<td>(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.</td>
</tr>
<tr>
<td>az_cvo_license_type</td>
<td>String</td>
<td>(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.</td>
</tr>
<tr>
<td>az_cvo_nss_account</td>
<td>String</td>
<td>(Required) the 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.</td>
</tr>
<tr>
<td>az_tenant_id</td>
<td>String</td>
<td>(Required) Tenant ID of the application/service principal registered in Azure.</td>
</tr>
<tr>
<td>az_application_id</td>
<td>String</td>
<td>(Required) Application ID of the application/service principal registered in Azure.</td>
</tr>
<tr>
<td>az_application_key</td>
<td>String</td>
<td>(Required) The Application Key of the application/service principal registered in Azure.</td>
</tr>
</tbody>
</table>

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.

```bash
git clone https://github.com/NetApp/na_cloud_volumes_automation.git
```

- b. Navigate to the desired folder

```bash
cd na_cloud_volumes_automation
```

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

```bash
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.

```bash
terraform init
```

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

```bash
terraform validate
```

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

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

- h. Run the deployment

```bash
terraform apply -target="module.az_cvo_ha_deployment" -var -file="vars\azure_cvo_ha_deployment.tfvars"
```
To delete the deployment

```bash
terraform destroy
```

**Recipes:**

**HA Pair Instance**

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>refresh_token</td>
<td>String</td>
<td>(Required) The refresh token of NetApp cloud manager. This can be generated from netapp Cloud Central.</td>
</tr>
<tr>
<td>az_connector_name</td>
<td>String</td>
<td>(Required) The name of the Cloud Manager Connector.</td>
</tr>
<tr>
<td>az_connector_location</td>
<td>String</td>
<td>(Required) The location where the Cloud Manager Connector will be created.</td>
</tr>
<tr>
<td>az_connector_subscription_id</td>
<td>String</td>
<td>(Required) The ID of the Azure subscription.</td>
</tr>
<tr>
<td>az_company</td>
<td>String</td>
<td>(Required) The name of the company of the user.</td>
</tr>
<tr>
<td>az_resource_group</td>
<td>Integer</td>
<td>(Required) The resource group in Azure where the resources will be created.</td>
</tr>
<tr>
<td>az_subnet_id</td>
<td>String</td>
<td>(Required) The name of the subnet for the virtual machine.</td>
</tr>
<tr>
<td>az_vnet_id</td>
<td>String</td>
<td>(Required) The name of the virtual network.</td>
</tr>
<tr>
<td>az_network_security_group_name</td>
<td>String</td>
<td>(Required) The name of the security group for the instance.</td>
</tr>
<tr>
<td>az_associate_public_ip_address</td>
<td>String</td>
<td>(Required) Indicates whether to associate the public IP address to the virtual machine.</td>
</tr>
<tr>
<td>az_account_id</td>
<td>String</td>
<td>(Required) The NetApp account ID that the Connector will be associated with. If not provided, Cloud Manager uses the first account. If no account exists, Cloud Manager creates a new account. You can find the account ID in the account tab of Cloud Manager at <a href="https://cloudmanager.netapp.com">https://cloudmanager.netapp.com</a>.</td>
</tr>
<tr>
<td>az_admin_password</td>
<td>String</td>
<td>(Required) The password for the Connector.</td>
</tr>
<tr>
<td>az_admin_username</td>
<td>String</td>
<td>(Required) The user name for the Connector.</td>
</tr>
<tr>
<td>az_cvo_name</td>
<td>String</td>
<td>(Required) The name of the Cloud Volumes ONTAP working environment.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>az_cvo_location</td>
<td>String</td>
<td>(Required) The location where the working environment will be created.</td>
</tr>
<tr>
<td>az_cvo_subnet_id</td>
<td>String</td>
<td>(Required) The name of the subnet for the Cloud Volumes ONTAP system.</td>
</tr>
<tr>
<td>az_cvo_vnet_id</td>
<td>String</td>
<td>(Required) The name of the virtual network.</td>
</tr>
<tr>
<td>az_cvo_vnet_resource_group</td>
<td>String</td>
<td>(Required) The resource group in Azure associated to the virtual network.</td>
</tr>
<tr>
<td>az_cvo_data_encryption_type</td>
<td>String</td>
<td>(Required) The type of encryption to use for the working environment: [AZURE, NONE]. The default is AZURE.</td>
</tr>
<tr>
<td>az_cvo_storage_type</td>
<td>String</td>
<td>(Required) The type of storage for the first data aggregate: [Premium_LRS, Standard_LRS, StandardSSD_LRS]. The default is Premium_LRS</td>
</tr>
<tr>
<td>az_cvo_svm_password</td>
<td>String</td>
<td>(Required) The admin password for Cloud Volumes ONTAP.</td>
</tr>
<tr>
<td>az_cvo_workspace_id</td>
<td>String</td>
<td>(Required) The ID of the Cloud Manager workspace where you want to deploy Cloud Volumes ONTAP. If not provided, Cloud Manager uses the first workspace. You can find the ID from the Workspace tab on <a href="https://cloudmanager.netapp.com">https://cloudmanager.netapp.com</a>.</td>
</tr>
<tr>
<td>az_cvo_capacity_tier</td>
<td>String</td>
<td>(Required) Whether to enable data tiering for the first data aggregate: [Blob, NONE]. The default is BLOB.</td>
</tr>
<tr>
<td>az_cvo_writing_speed_state</td>
<td>String</td>
<td>(Required) The write speed setting for Cloud Volumes ONTAP: [NORMAL, HIGH]. The default is NORMAL. This argument is not relevant for HA pairs.</td>
</tr>
<tr>
<td>az_cvo_ontap_version</td>
<td>String</td>
<td>(Required) The required ONTAP version. Ignored if 'use_latest_version' is set to true. The default is to use the latest version.</td>
</tr>
<tr>
<td>az_cvo_instance_type</td>
<td>String</td>
<td>(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.</td>
</tr>
<tr>
<td>az_cvo_license_type</td>
<td>String</td>
<td>(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.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>az_cvo_nss_account</td>
<td>String</td>
<td>(Required) The 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.</td>
</tr>
<tr>
<td>az_tenant_id</td>
<td>String</td>
<td>(Required) Tenant ID of the application/service principal registered in Azure.</td>
</tr>
<tr>
<td>az_application_id</td>
<td>String</td>
<td>(Required) Application ID of the application/service principal registered in Azure.</td>
</tr>
<tr>
<td>az_application_key</td>
<td>String</td>
<td>(Required) The Application Key of the application/service principal registered in Azure.</td>
</tr>
</tbody>
</table>
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

Recipes:

Connector

Terraform variables for NetApp GCP connector instance for CVO deployment.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>gcp_connector_deploy_bool</td>
<td>Bool</td>
<td>(Required) Check for Connector deployment.</td>
</tr>
<tr>
<td>gcp_connector_name</td>
<td>String</td>
<td>(Required) The name of the Cloud Manager Connector.</td>
</tr>
<tr>
<td>gcp_connector_project_id</td>
<td>String</td>
<td>(Required) The GCP project_id where the connector will be created.</td>
</tr>
<tr>
<td>gcp_connector_zone</td>
<td>String</td>
<td>(Required) The GCP zone where the Connector will be created.</td>
</tr>
<tr>
<td>gcp_connector_company</td>
<td>String</td>
<td>(Required) The name of the company of the user.</td>
</tr>
<tr>
<td>gcp_connector_service_account_email</td>
<td>String</td>
<td>(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.</td>
</tr>
<tr>
<td>gcp_connector_service_account_path</td>
<td>String</td>
<td>(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.</td>
</tr>
<tr>
<td>gcp_connector_account_id</td>
<td>String</td>
<td>(Optional) The NetApp account ID that the Connector will be associated with. If not provided, Cloud Manager uses the first account. If no account exists, Cloud Manager creates a new account. You can find the account ID in the account tab of Cloud Manager at <a href="https://cloudmanager.netapp.com">https://cloudmanager.netapp.com</a>.</td>
</tr>
</tbody>
</table>

Single Node Instance

Terraform variables for single NetApp CVO instance on GCP.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>gcp_cvo_name</td>
<td>String</td>
<td>(Required) The name of the Cloud Volumes ONTAP working environment.</td>
</tr>
</tbody>
</table>
### 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](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`.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
</table>
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

Recipes:
Connector

Terraform variables for NetApp GCP connector instance for CVO deployment.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>gcp_connector_deploy_bool</td>
<td>Bool</td>
<td>(Required) Check for Connector deployment.</td>
</tr>
<tr>
<td>gcp_connector_name</td>
<td>String</td>
<td>(Required) The name of the Cloud Manager Connector.</td>
</tr>
<tr>
<td>gcp_connector_project_id</td>
<td>String</td>
<td>(Required) The GCP project_id where the connector will be created.</td>
</tr>
<tr>
<td>gcp_connector_zone</td>
<td>String</td>
<td>(Required) The GCP zone where the Connector will be created.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-----------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>gcp_connector_company</td>
<td>String</td>
<td>(Required) The name of the company of the user.</td>
</tr>
<tr>
<td>gcp_connector_service_account_email</td>
<td>String</td>
<td>(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.</td>
</tr>
<tr>
<td>gcp_connector_service_account_path</td>
<td>String</td>
<td>(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.</td>
</tr>
<tr>
<td>gcp_connector_account_id</td>
<td>String</td>
<td>(Optional) The NetApp account ID that the Connector will be associated with. If not provided, Cloud Manager uses the first account. If no account exists, Cloud Manager creates a new account. You can find the account ID in the account tab of Cloud Manager at <a href="https://cloudmanager.netapp.com">https://cloudmanager.netapp.com</a>.</td>
</tr>
</tbody>
</table>

**HA Pair**

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>gcp_cvo_is_ha</td>
<td>Bool</td>
<td>(Optional) Indicate whether the working environment is an HA pair or not [true, false]. The default is false.</td>
</tr>
<tr>
<td>gcp_cvo_name</td>
<td>String</td>
<td>(Required) The name of the Cloud Volumes ONTAP working environment.</td>
</tr>
<tr>
<td>gcp_cvo_project_id</td>
<td>String</td>
<td>(Required) The ID of the GCP project.</td>
</tr>
<tr>
<td>gcp_cvo_zone</td>
<td>String</td>
<td>(Required) The zone of the region where the working environment will be created.</td>
</tr>
<tr>
<td>gcp_cvo_node1_zone</td>
<td>String</td>
<td>(Optional) Zone for node 1.</td>
</tr>
<tr>
<td>gcp_cvo_node2_zone</td>
<td>String</td>
<td>(Optional) Zone for node 2.</td>
</tr>
<tr>
<td>gcp_cvo_mediator_zone</td>
<td>String</td>
<td>(Optional) Zone for mediator.</td>
</tr>
<tr>
<td>gcp_cvo_vpc_id</td>
<td>String</td>
<td>(Optional) The name of the VPC.</td>
</tr>
<tr>
<td>gcp_cvo_subnet_id</td>
<td>String</td>
<td>(Optional) The name of the subnet for Cloud Volumes ONTAP. The default is: 'default'.</td>
</tr>
<tr>
<td>gcp_cvo_vpc0_node_and_data_connectivity</td>
<td>String</td>
<td>(Optional) VPC path for nic1, required for node and data connectivity. If using shared VPC, netwrok_project_id must be provided.</td>
</tr>
<tr>
<td>gcp_cvo_vpc1_cluster_connectivity</td>
<td>String</td>
<td>(Optional) VPC path for nic2, required for cluster connectivity.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>-----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>gcp_cvo_vpc2_ha_connectivity</td>
<td>String</td>
<td>(Optional) VPC path for nic3, required for HA connectivity.</td>
</tr>
<tr>
<td>gcp_cvo_vpc3_data_replication</td>
<td>String</td>
<td>(Optional) VPC path for nic4, required for data replication.</td>
</tr>
<tr>
<td>gcp_cvo_subnet0_node_and_data_connectivity</td>
<td>String</td>
<td>(Optional) Subnet path for nic1, required for node and data connectivity. If using shared VPC, netwrok_project_id must be provided.</td>
</tr>
<tr>
<td>gcp_cvo_subnet1_cluster_connectivity</td>
<td>String</td>
<td>(Optional) Subnet path for nic2, required for cluster connectivity.</td>
</tr>
<tr>
<td>gcp_cvo_subnet2_ha_connectivity</td>
<td>String</td>
<td>(Optional) Subnet path for nic3, required for HA connectivity.</td>
</tr>
<tr>
<td>gcp_cvo_subnet3_data_replication</td>
<td>String</td>
<td>(Optional) Subnet path for nic4, required for data replication.</td>
</tr>
<tr>
<td>gcp_cvo_gcp_service_account</td>
<td>String</td>
<td>(Required) The gcp_service_account email in order to enable tiering of cold data to Google Cloud Storage.</td>
</tr>
<tr>
<td>gcp_cvo_svm_password</td>
<td>String</td>
<td>(Required) The admin password for Cloud Volumes ONTAP.</td>
</tr>
<tr>
<td>gcp_cvo_workspace_id</td>
<td>String</td>
<td>(Optional) The ID of the Cloud Manager workspace where you want to deploy Cloud Volumes ONTAP. If not provided, Cloud Manager uses the first workspace. You can find the ID from the Workspace tab on <a href="https://cloudmanager.netapp.com">https://cloudmanager.netapp.com</a>.</td>
</tr>
<tr>
<td>gcp_cvo_capacity_package_name</td>
<td>String</td>
<td>(Optional) The capacity package name: ['Essential', 'Professional', 'Freemium']. Default is 'Essential'.</td>
</tr>
<tr>
<td>gcp_cvo_gcp_volume_size</td>
<td>String</td>
<td>(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'.</td>
</tr>
<tr>
<td>gcp_cvo_gcp_volume_size_unit</td>
<td>String</td>
<td>(Optional) ['GB' or 'TB']. The default is ‘TB’.</td>
</tr>
</tbody>
</table>

**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).

Procedure
In order to run the template:

a. Clone the repository.

```bash
git clone
https://github.com/NetApp/na_cloud_volumes_automation.git
```

b. Navigate to the desired folder

```bash
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.

```bash
terraform init
```

f. Verify the terraform files using terraform validate command.

```bash
terraform validate
```

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

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

h. Run the deployment

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

To delete the deployment

```bash
terraform destroy
```

Recipes:
CVS Volume
Terraform variables for NetApp GCP CVS Volume.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>gcp_cvs_name</td>
<td>String</td>
<td>(Required) The name of the NetApp CVS volume.</td>
</tr>
<tr>
<td>gcp_cvs_project_id</td>
<td>String</td>
<td>(Required) The GCP project_id where the CVS Volume will be created.</td>
</tr>
<tr>
<td>gcp_cvs_gcp_service_account_path</td>
<td>String</td>
<td>(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.</td>
</tr>
<tr>
<td>gcp_cvs_region</td>
<td>String</td>
<td>(Required) The GCP zone where the CVS Volume will be created.</td>
</tr>
<tr>
<td>gcp_cvs_network</td>
<td>String</td>
<td>(Required) The network VPC of the volume.</td>
</tr>
<tr>
<td>gcp_cvs_size</td>
<td>Integer</td>
<td>(Required) The size of volume is between 1024 to 102400 inclusive (in GiB).</td>
</tr>
<tr>
<td>gcp_cvs_volume_path</td>
<td>String</td>
<td>(Optional) The name of the volume path for volume.</td>
</tr>
<tr>
<td>gcp_cvs_protocol_types</td>
<td>String</td>
<td>(Required) The protocol_type of the volume. For NFS use 'NFSv3' or 'NFSv4' and for SMB use 'CIFS' or 'SMB'.</td>
</tr>
</tbody>
</table>

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

ℹ️ 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.
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.

   ![Create User](image)

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

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.
3. 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.

4. 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.
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.
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.

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.
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:

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>&lt;Any name for the vpc endpoint&gt;</td>
</tr>
<tr>
<td>Service category</td>
<td>AWS Services</td>
</tr>
<tr>
<td>Services</td>
<td>com.amazonaws.&lt;region&gt;.fsx</td>
</tr>
<tr>
<td>vpc</td>
<td>&lt;select the vpc where lambda will be deployed&gt;</td>
</tr>
<tr>
<td>subnets</td>
<td>&lt;select the subnets where lambda will be deployed&gt;</td>
</tr>
<tr>
<td>Security groups</td>
<td>&lt;select the security group&gt;</td>
</tr>
<tr>
<td>Policy</td>
<td>&lt;Either choose Full access or set your own custom policy&gt;</td>
</tr>
</tbody>
</table>

Click on Create endpoint.
Create endpoint

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.
  - fxsEndpoint

- **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)**

- **Service Name**: com.amazonaws.us-west-1.fxs
  - **Owner**: amazon
  - **Type**: Interface
2. 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.
3. Navigate to the newly created Lambda function > Scroll down to the **Layers** section and click on **Add a layer**.

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**.
6. Navigate back to AWS Lambda function > Add Layer > Custom Layers and add the utilities layer.
7. 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**.

8. 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**.
   a. Click on the JSON tab and paste the contents of the file policy.json from the GitHub repo.
   b. Replace every occurrence of `${AWS::AccountId}` with your account ID and click on **Review Policy**
   c. Provide a Name for the policy and click on **Create policy**

9. 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.

10. 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**:
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>fsxList</td>
<td>List</td>
<td>(Required) List of all FSx for ONTAP filesystems to be monitored. Include all the file systems in the list for monitoring and auto-resizing.</td>
</tr>
<tr>
<td>fsxMgmtIp</td>
<td>String</td>
<td>(Required) Enter the &quot;Management endpoint - IP address&quot; from the FSx for ONTAP console on AWS.</td>
</tr>
<tr>
<td>fsxId</td>
<td>String</td>
<td>(Required) Enter the &quot;File system ID&quot; from the FSx for ONTAP console on AWS.</td>
</tr>
<tr>
<td>username</td>
<td>String</td>
<td>(Required) Enter the FSx for ONTAP &quot;ONTAP administrator username&quot; from FSx for ONTAP console on AWS.</td>
</tr>
<tr>
<td>resize_threshold</td>
<td>Integer</td>
<td>(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.</td>
</tr>
<tr>
<td>fsx_password_ssm_parameter</td>
<td>String</td>
<td>(Required) Enter the path name used in AWS Parameter Store for storing &quot;fsxadmin&quot; password.</td>
</tr>
<tr>
<td>warn_notification</td>
<td>Bool</td>
<td>(Required) Set this variable to True to receive a notification when Storage Capacity/Volume/LUN usage exceeds 75% but is less than the threshold.</td>
</tr>
<tr>
<td>enable_snapshot_deletion</td>
<td>Bool</td>
<td>(Required) Set this variable to True to enable volume level snapshot deletion for snapshots older than the value specified in &quot;snapshot_age_threshold_in_days&quot;.</td>
</tr>
<tr>
<td>snapshot_age_threshold_in_days</td>
<td>Integer</td>
<td>(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.</td>
</tr>
<tr>
<td><strong>internet_access</strong></td>
<td><strong>Bool</strong></td>
<td>(Required) Set this variable to True if internet access is available from the subnet where this lambda is deployed. Otherwise set it to False.</td>
</tr>
<tr>
<td>--------------------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>smtp_region</strong></td>
<td><strong>String</strong></td>
<td>(Optional) If &quot;internet_access&quot; variable is set to False, enter the region in which lambda is deployed. E.g. us-east-1 (in this format)</td>
</tr>
<tr>
<td><strong>smtp_username_ssm_parameter</strong></td>
<td><strong>String</strong></td>
<td>(Optional) If &quot;internet_access&quot; variable is set to False, enter the path name used in AWS Parameter Store for storing the SMTP username.</td>
</tr>
<tr>
<td><strong>smtp_password_ssm_parameter</strong></td>
<td><strong>String</strong></td>
<td>(Optional) If &quot;internet_access&quot; variable is set to False, enter the path name used in AWS Parameter Store for storing the SMTP password.</td>
</tr>
<tr>
<td><strong>sender_email</strong></td>
<td><strong>String</strong></td>
<td>(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.</td>
</tr>
<tr>
<td><strong>recipient_email</strong></td>
<td><strong>String</strong></td>
<td>(Required) Enter the email ID on which you want to receive the alert notifications.</td>
</tr>
</tbody>
</table>

11. 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.

12. Once tested successfully, navigate to **Configuration > Triggers > Add Trigger**.
Select a Source: EventBridge
Rule: Create a new rule
Rule name: <Enter any name>
Rule type: Schedule expression
Schedule expression: <Use "rate(1 day)" if you want the function to run daily or add your own cron expression>

Click on Add.

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.