terraform {

required\_providers {

netapp-cloudmanager = {

source = "NetApp/netapp-cloudmanager"

version = "21.1.1"

}

}

}

provider "netapp-cloudmanager" {

# Configuration options

}

Execute the terraform init command to initiate the provider for your deployment.

You can obtain the refresh token by logging into your NetApp Cloud Central account and going to this page: <https://services.cloud.netapp.com/refresh-token>.

**Cloud Manager Terraform Provider Supported Resources and Sample Configuration**

Cloud Manager Terraform provider can be used to create the following type of resources: Cloud Volumes ONTAP, aggregates, CIFS server, Cloud Volumes ONTAP storage volumes, NSS Accounts, and SnapMirror® relationships. Let’s look at how this works for each one of them.

**Cloud Volumes ONTAP**

In this section we’ll show you how to configure Cloud Volumes ONTAP using Terraform Provider for Cloud Manager**.**

As a prerequisite, you’ll need a Connecter in place. A Connector resource is required in each cloud provider environment for Cloud Manager to deploy/manage Cloud Volumes ONTAP resources and processes.

To deploy a Connector, you’ll need an account with permissions in the cloud environment you want to use and a Connector with policies configured for [AWS](https://s3.amazonaws.com/occm-sample-policies/Policy_for_Setup_As_Service.json), [Azure](https://s3.amazonaws.com/occm-sample-policies/Policy_for_Setup_As_Service_Azure.json) and [GCP](https://occm-sample-policies.s3.amazonaws.com/Policy_for_Cloud_Manager_3.8.0_GCP.yaml). Sample code for deployment of connectors can be found here:

[Sample code for connector deployment on AWS](https://registry.terraform.io/providers/NetApp/netapp-cloudmanager/latest/docs/resources/connector_aws)

[Sample code for connector deployment in Azure](https://registry.terraform.io/providers/NetApp/netapp-cloudmanager/latest/docs/resources/connector_azure)

[Sample code for connector deployment in GCP](https://registry.terraform.io/providers/NetApp/netapp-cloudmanager/latest/docs/resources/connector_gcp)

With your Connector set up, you can now use Terraform to deploy Cloud Volumes ONTAP. Let’s explore a [sample](https://registry.terraform.io/providers/NetApp/netapp-cloudmanager/latest/docs/resources/cvo_aws) Terraform code block that can be used for Cloud Volumes ONTAP single node deployment in AWS.

resource "netapp-cloudmanager\_cvo\_aws" "Cloud Volumes ONTAP-aws" {

provider = netapp-cloudmanager

name = "Cloud Volumes ONTAPSingleNode"

region = "us-west-2"

subnet\_id = "subnet-test"

vpc\_id = "vpc-0f46c06a"

aws\_tag {

tag\_key = "demo"

tag\_value = "env1"

}

aws\_tag {

tag\_key = "conf"

tag\_value = "singlenode"

}

svm\_password = "Test@1234!"

client\_id = netapp-cloudmanager\_connector\_aws.cm-aws.client\_id

writing\_speed\_state = "NORMAL"

}

The following are the arguments for this code

name indicates the name of the Cloud Volumes ONTAP environment

region is the AWS region where the Cloud Volumes ONTAP will be deployed

subnet\_id is the subnet id to which Cloud Volumes ONTAP will be connected

vpc\_id is the ID of the AWS VPC where the Cloud Volumes ONTAP will be created. This is an optional argument. If not provided it will be derived from the provided subnet ID.

client\_id is the ID that can be found from the connector tab in [cloudmanager web console](https://cloudmanager.netapp.com/) or can reference a connector created using Terraform as shown in this example

svm\_password is your Cloud Volumes ONTAP admin password

aws\_tag is the AWS tag key value pair to be added to the deployment

writing\_speed\_state defines the write speed setting for Cloud Volumes ONTAP. Values can be either NORMAL or HIGH. It is an optional argument and the default value is NORMAL.

**Aggregate**

The [architecture of Cloud Volumes ONTAP](https://docs.netapp.com/us-en/occm/concept_storage.html#overview)depends on the underlying cloud provider storage disks and groups them into aggregates. The aggregates are then used to provision Cloud Volumes ONTAP storage volumes. Cloud Manager aggregate deployment needs an existing Cloud Volumes ONTAP and Cloud Manager connector as prerequisites.  
[Sample code for deployment of an aggregate can be found here](https://registry.terraform.io/providers/NetApp/netapp-cloudmanager/latest/docs/resources/aggregate)

**CIFS Server**

Using Cloud Manager Terraform provider, you can provision a CIFS server resource based on an existing AD or workgroup on a Cloud Volumes ONTAP. It can then be used to create CIFS volumes. It also requires a Cloud Volumes ONTAP and Cloud Manager connector deployed as prerequisites.   
[Sample code for deployment of CIFS server can be found here](https://registry.terraform.io/providers/NetApp/netapp-cloudmanager/latest/docs/resources/cifs_server)

**Cloud Volumes ONTAP Storage Volumes**

This code can be used to create, update, or delete Cloud Volumes ONTAP volumes. You can create NFS, CIFS, and iSCSI volumes using the provider. Both a Cloud Manager Connector resource and Cloud Volumes ONTAP are prerequisites for this action.  
[Sample code for provisioning volumes can be found here](https://registry.terraform.io/providers/NetApp/netapp-cloudmanager/latest/docs/resources/cvo_volume)

**NSS Account**

Using Terraform provider, you can create or delete NetApp Support Site Account, that can be used to reach out to NetApp support team, access KB articles, and download software.  
[A sample code for deploying an NSS account can be found here](https://registry.terraform.io/providers/NetApp/netapp-cloudmanager/latest/docs/resources/nss_account)

**SnapMirror**

You can use NetApp [SnapMirror technology](https://cloud.netapp.com/blog/snapmirror-data-replication-aws) to replicate data between different Cloud Volumes ONTAP volumes. Cloud Manager Terraform provider can be leveraged to create new SnapMirror® configuration for Cloud Volumes ONTAP. Doing this requires a Cloud Volumes ONTAP system and a Cloud Manager connector as prerequisites.  
[Sample code to create new SnapMirror configurations for Cloud Volumes ONTAP can be found here.](https://registry.terraform.io/providers/NetApp/netapp-cloudmanager/latest/docs/resources/snapmirror)

**Conclusion**

NetApp Terraform Cloud provider enables fully supported IAC deployment for Cloud Volumes ONTAP environments in multiple cloud platforms. It can be easily integrated with your existing DevOps tools to build a comprehensive CI/CD solution for your cloud storage deployments. It enhances the agility offered by Cloud Volumes ONTAP to quickly provision and configure storage volumes for your enterprise workloads on demand.

To get started now, [sample configuration files for AWS/Azure/GCP are available in the GitHub](https://github.com/NetApp/terraform-provider-netapp-cloudmanager/tree/main/examples) for NetApp Cloud Manager Terraform Provider.

Sample code to

# Specify CVO resources

provider "azurerm" {

features {}

skip\_provider\_registration = true

}

terraform {

required\_providers {

netapp-cloudmanager = {

source = "NetApp/netapp-cloudmanager"

version = "20.10.0"

}

}

}

resource "azurerm\_resource\_group" "occm-rg" {

name = "CMTerraform"

location = "westus"

}

data "azurerm\_subscription" "primary" {

}

resource "netapp-cloudmanager\_connector\_azure" "cm-azure" {

depends\_on = [azurerm\_resource\_group.occm-rg]

provider = netapp-cloudmanager

name = "CMTerraform"

location = "westus"

subscription\_id = data.azurerm\_subscription.primary.subscription\_id

company = "NetApp"

resource\_group = "CMTerraform"

vnet\_resource\_group = "rg\_westus"

network\_security\_group\_name = "OCCM\_SG"

network\_security\_resource\_group = "rg\_westus"

subnet\_id = "Subnet1"

vnet\_id = "Vnet1"

account\_id = "account-xxxxxxx"

admin\_password = "P@ssword!"

admin\_username = "vmadmin"

}

data "azurerm\_virtual\_machine" "occm-vm" {

depends\_on = [netapp-cloudmanager\_connector\_azure.cm-azure]

name = "CMTerraform"

resource\_group\_name = "CMTerraform"

}

resource "azurerm\_role\_definition" "occm-role" {

role\_definition\_id = "12345678-0000-0000-b9d7-123456789012"

name = "Terraform-OCCM-Role"

scope = data.azurerm\_subscription.primary.id

description = "This is a custom role created via Terraform"

permissions {

actions = ["Microsoft.Compute/disks/delete",

"Microsoft.Compute/disks/read",

"Microsoft.Compute/disks/write",

"Microsoft.Compute/locations/operations/read",

"Microsoft.Compute/locations/vmSizes/read",

"Microsoft.Resources/subscriptions/locations/read",

"Microsoft.Compute/operations/read",

"Microsoft.Compute/virtualMachines/instanceView/read",

"Microsoft.Compute/virtualMachines/powerOff/action",

"Microsoft.Compute/virtualMachines/read",

"Microsoft.Compute/virtualMachines/restart/action",

"Microsoft.Compute/virtualMachines/deallocate/action",

"Microsoft.Compute/virtualMachines/start/action",

"Microsoft.Compute/virtualMachines/vmSizes/read",

"Microsoft.Compute/virtualMachines/write",

"Microsoft.Compute/images/write",

"Microsoft.Compute/images/read",

"Microsoft.Network/locations/operationResults/read",

"Microsoft.Network/locations/operations/read",

"Microsoft.Network/networkInterfaces/read",

"Microsoft.Network/networkInterfaces/write",

"Microsoft.Network/networkInterfaces/join/action",

"Microsoft.Network/networkSecurityGroups/read",

"Microsoft.Network/networkSecurityGroups/write",

"Microsoft.Network/networkSecurityGroups/join/action",

"Microsoft.Network/virtualNetworks/read",

"Microsoft.Network/virtualNetworks/checkIpAddressAvailability/read",

"Microsoft.Network/virtualNetworks/subnets/read",

"Microsoft.Network/virtualNetworks/subnets/write",

"Microsoft.Network/virtualNetworks/subnets/virtualMachines/read",

"Microsoft.Network/virtualNetworks/virtualMachines/read",

"Microsoft.Network/virtualNetworks/subnets/join/action",

"Microsoft.Resources/deployments/operations/read",

"Microsoft.Resources/deployments/read",

"Microsoft.Resources/deployments/write",

"Microsoft.Resources/resources/read",

"Microsoft.Resources/subscriptions/operationresults/read",

"Microsoft.Resources/subscriptions/resourceGroups/delete",

"Microsoft.Resources/subscriptions/resourceGroups/read",

"Microsoft.Resources/subscriptions/resourcegroups/resources/read",

"Microsoft.Resources/subscriptions/resourceGroups/write",

"Microsoft.Storage/checknameavailability/read",

"Microsoft.Storage/operations/read",

"Microsoft.Storage/storageAccounts/listkeys/action",

"Microsoft.Storage/storageAccounts/read",

"Microsoft.Storage/storageAccounts/delete",

"Microsoft.Storage/storageAccounts/regeneratekey/action",

"Microsoft.Storage/storageAccounts/write",

"Microsoft.Storage/usages/read",

"Microsoft.Compute/snapshots/write",

"Microsoft.Compute/snapshots/read",

"Microsoft.Compute/availabilitySets/write",

"Microsoft.Compute/availabilitySets/read",

"Microsoft.Compute/disks/beginGetAccess/action",

"Microsoft.MarketplaceOrdering/offertypes/publishers/offers/plans/agreements/read",

"Microsoft.MarketplaceOrdering/offertypes/publishers/offers/plans/agreements/write",

"Microsoft.Network/loadBalancers/read",

"Microsoft.Network/loadBalancers/write",

"Microsoft.Network/loadBalancers/delete",

"Microsoft.Network/loadBalancers/backendAddressPools/read",

"Microsoft.Network/loadBalancers/backendAddressPools/join/action",

"Microsoft.Network/loadBalancers/frontendIPConfigurations/read",

"Microsoft.Network/loadBalancers/loadBalancingRules/read",

"Microsoft.Network/loadBalancers/probes/read",

"Microsoft.Network/loadBalancers/probes/join/action",

"Microsoft.Authorization/locks/\*",

"Microsoft.Network/routeTables/join/action",

"Microsoft.NetApp/netAppAccounts/capacityPools/volumes/write",

"Microsoft.NetApp/netAppAccounts/capacityPools/volumes/read",

"Microsoft.NetApp/netAppAccounts/capacityPools/volumes/delete",

"Microsoft.NetApp/netAppAccounts/write",

"Microsoft.NetApp/netAppAccounts/read",

"Microsoft.NetApp/netAppAccounts/capacityPools/write",

"Microsoft.NetApp/netAppAccounts/capacityPools/read",

"Microsoft.NetApp/netAppAccounts/capacityPools/volumes/delete",

"Microsoft.Network/privateEndpoints/write",

"Microsoft.Storage/storageAccounts/PrivateEndpointConnectionsApproval/action",

"Microsoft.Storage/storageAccounts/privateEndpointConnections/read",

"Microsoft.Network/privateEndpoints/read",

"Microsoft.Network/privateDnsZones/write",

"Microsoft.Network/privateDnsZones/virtualNetworkLinks/write",

"Microsoft.Network/virtualNetworks/join/action",

"Microsoft.Network/privateDnsZones/A/write",

"Microsoft.Network/privateDnsZones/read",

"Microsoft.Network/privateDnsZones/virtualNetworkLinks/read",

"Microsoft.Resources/deployments/operationStatuses/read",

"Microsoft.Insights/Metrics/Read",

"Microsoft.Compute/virtualMachines/extensions/write",

"Microsoft.Compute/virtualMachines/extensions/read",

"Microsoft.Compute/virtualMachines/extensions/delete",

"Microsoft.Compute/virtualMachines/delete",

"Microsoft.Network/networkInterfaces/delete",

"Microsoft.Network/networkSecurityGroups/delete",

"Microsoft.Resources/deployments/delete",

"Microsoft.Compute/diskEncryptionSets/read"]

not\_actions = []

}

assignable\_scopes = [

data.azurerm\_subscription.primary.id, # /subscriptions/00000000-0000-0000-0000-000000000000

]

}

resource "azurerm\_role\_assignment" "occm-role-assignment" {

depends\_on = [azurerm\_role\_definition.occm-role]

scope = data.azurerm\_subscription.primary.id

role\_definition\_id = azurerm\_role\_definition.occm-role.role\_definition\_resource\_id

principal\_id = data.azurerm\_virtual\_machine.occm-vm.identity.0.principal\_id

}

resource "netapp-cloudmanager\_cvo\_azure" "cvo-azure" {

depends\_on = [azurerm\_role\_assignment.occm-role-assignment]

provider = netapp-cloudmanager

name = "TerraformCVOAzure"

location = "westus"

subscription\_id = data.azurerm\_subscription.primary.subscription\_id

subnet\_id = "Subnet1"

vnet\_id = "Vnet1"

vnet\_resource\_group = "rg\_westus"

data\_encryption\_type = "AZURE"

azure\_tag {

tag\_key = "abcd"

tag\_value = "ABCD"

}

azure\_tag {

tag\_key = "xxx"

tag\_value = "YYY"

}

storage\_type = "Premium\_LRS"

svm\_password = "\*\*\*\*\*\*\*\*"

client\_id = netapp-cloudmanager\_connector\_azure.cm-azure.client\_id

workspace\_id = "workspace-xxxxxx"

capacity\_tier = "Blob"

writing\_speed\_state = "NORMAL"

is\_ha = false

}

resource "netapp-cloudmanager\_cvo\_azure" "cvo-azure-ha" {

depends\_on = [azurerm\_role\_assignment.occm-role-assignment]

provider = netapp-cloudmanager

name = "TerraformCVOAzureHA"

location = "westus"

subscription\_id = data.azurerm\_subscription.primary.subscription\_id

subnet\_id = "Subnet1"

vnet\_id = "Vnet1"

vnet\_resource\_group = "rg\_westus"

data\_encryption\_type = "AZURE"

azure\_tag {

tag\_key = "abcd"

tag\_value = "ABCD"

}

azure\_tag {

tag\_key = "xxx"

tag\_value = "YYY"

}

storage\_type = "Premium\_LRS"

svm\_password = "\*\*\*\*\*\*\*\*"

client\_id = netapp-cloudmanager\_connector\_azure.cm-azure.client\_id

workspace\_id = "workspace-xxxxxx"

capacity\_tier = "Blob"

writing\_speed\_state = "NORMAL"

is\_ha = true

license\_type = "azure-ha-cot-standard-paygo"

}