VMware Cloud on AWS

VMware Cloud on AWS Terraform deployment – Phase 3

Gilles Chekroun
July 15, 2022

After SDDC deployment in phase 1 and networking and security deployment in phase 2, we are now ready to deploy Virtual Machines in our SDDC. The goal of this final phase is to use Terraform vSphere provider to deploy a Content Library in VMware Cloud on AWS and clone VMs to the 2 NSX segments created earlier. As noted in the phase 1, all source files are available for download here.

Main.tf

Our `main.tf` module is simple and is used to initiate our state file's locations and the Virtual Machine module.

Initialization

Similar to phase 1 and 2, we will set a local path for phase 3 state file and read phase 1 and phase 2 data.

```
terraform {
  backend "local" {
    path = "../../phase3.tfstate"
  }
}
# Import the state from phase 1and 2 and read the outputs
data "terraform_remote_state" "phase1" {
  backend = "local"
  config = {
    path    = "../../phase1.tfstate"
  }
}
data "terraform_remote_state" "phase2" {
  backend = "local"
  config = {
    path    = "../../phase2.tfstate"
  }
}
```

vSphere Terraform Provider Setup

The vSphere provider will use vCenter username, password, and vCenter URL from phase 1. The NSX-T provider uses the NSX proxy URL and VMC token from phase 1 as well.

```
provider "vsphere" {
  user                 =
data.terraform_remote_state.phase1.outputs.cloud_username
  password             =
data.terraform_remote_state.phase1.outputs.cloud_password
  vsphere_server       = data.terraform_remote_state.phase1.outputs.vc_url
  allow_unverified_ssl = true
}

provider "nsxt" {
  host                 =
data.terraform_remote_state.phase1.outputs.proxy_url
  vmc_token            = var.vmc_token
  allow_unverified_ssl = true
  enforcement_point    = "vmc-enforcementpoint"
}
```

Virtual Machine Module

```
module "VMs" {
  source = "../VMs"

  data_center          = var.data_center
  cluster              = var.cluster
  workload_datastore   = var.workload_datastore
  compute_pool         = var.compute_pool
  Subnet12_name        =
data.terraform_remote_state.phase2.outputs.segment12_name
  subnet12             = data.terraform_remote_state.phase2.outputs.subnet12
  Subnet13_name        =
data.terraform_remote_state.phase2.outputs.segment13_name
  subnet13             = data.terraform_remote_state.phase2.outputs.subnet13
}
```

In this module, we read the outputs from phase 2 and use the two segments created (subnet 12 and 13) where Virtual Machines will be attached.

Variables

In the `variables.tf` files we will code the standard names for VMware Cloud on AWS data center SDDC

```
variable "data_center"          {default = "SDDC-Datacenter"}
variable "cluster"              {default = "Cluster-1"}
variable "workload_datastore" {default = "WorkloadDatastore"}
variable "compute_pool"         {default = "Compute-ResourcePool"}
variable "vmc_token"            {}
```

SDDC data

First step is to read the SDDC data using vSphere provider.

```
data "vsphere_datacenter" "dc" {
  name          = var.data_center
}
data "vsphere_compute_cluster" "cluster" {
  name          = var.cluster
  datacenter_id = data.vsphere_datacenter.dc.id
}
data "vsphere_datastore" "datastore" {
  name          = var.workload_datastore
  datacenter_id = data.vsphere_datacenter.dc.id
}
data "vsphere_resource_pool" "pool" {
  name          = var.compute_pool
  datacenter_id = data.vsphere_datacenter.dc.id
}
data "vsphere_network" "network12" {
  name          = var.Subnet12_name
  datacenter_id = data.vsphere_datacenter.dc.id
}
data "vsphere_network" "network13" {
  name          = var.Subnet13_name
  datacenter_id = data.vsphere_datacenter.dc.id
}
```

Create a Content Library

We will use a Content library from AWS S3 bucket in Frankfurt.

```
resource "vsphere_content_library" "S3CL" {
  name            = "S3CL"
  storage_backing = [data.vsphere_datastore.datastore.id]
  description     = "AWS S3 Content Library"

  subscription {
    subscription_url = "https://s3.eu-central-1.amazonaws.com/s3-cl-
frankfurt/lib.json"
    authentication_method = "NONE"
    automatic_sync = false
    on_demand = true
  }
}
```

We will wait 30 seconds and read the “blue VM” data.

```
  depends_on = [vsphere_content_library.S3CL]
  create_duration = "30s"
}

data "vsphere_content_library_item" "blue" {
  depends_on = [time_sleep.wait_30_seconds]
  name       = "blue"
  library_id = vsphere_content_library.S3CL.id
  type = "OVA"
}
```

Then we will:

- Clone 3 blue Virtual Machines using the demo-count set to 3
- Attach them to network 12
- Set a fixed IP address starting at .10
- Set a default gateway at .1

Note the 2 CPUs, 1GB RAM, and 20 GB disk.

```
resource "vsphere_virtual_machine" "Blue" {
  lifecycle {
    ignore_changes = [annotation]
  }
  count              = var.demo_count
  name               = "Blue-VM-${count.index + 1}"
  resource_pool_id   = data.vsphere_resource_pool.pool.id
  datastore_id       = data.vsphere_datastore.datastore.id
  folder             = "Workloads"

  num_cpus = 2
  memory   = 1024
  guest_id = "other3xLinux64Guest"

  network_interface {
    network_id = data.vsphere_network.network12.id
  }
  disk {
    label = "disk0"
    size  = 20
    thin_provisioned = true
  }
  clone {
    template_uuid = data.vsphere_content_library_item.blue.id
    customize {
      linux_options {
        host_name = "Photon"
        domain    = "vmc.local"
      }
      network_interface {
        ipv4_address = cidrhost(var.subnet12, 10 + count.index) #fixed IP
address
        ipv4_netmask = 24
      }
      ipv4_gateway = cidrhost(var.subnet12, 1)
    }
  }
}
```

We then repeat the process for 3 “Red Virtual Machines”.

NSX tags

Apply NSX tags using NSXT provider as shown below.

```
resource "nsxt_policy_vm_tags" "NSX_Blue_tag" {
  count = var.demo_count
  instance_id = vsphere_virtual_machine.Blue[count.index].id
  tag {
    tag   = "NSX_tag"
    scope = "Blue"
  }
}
resource "nsxt_policy_vm_tags" "NSX_Red_tag" {
  count = var.demo_count
  instance_id = vsphere_virtual_machine.Red[count.index].id
  tag {
    tag   = "NSX_tag"
    scope = "Red"
  }
}
```

vSphere tags

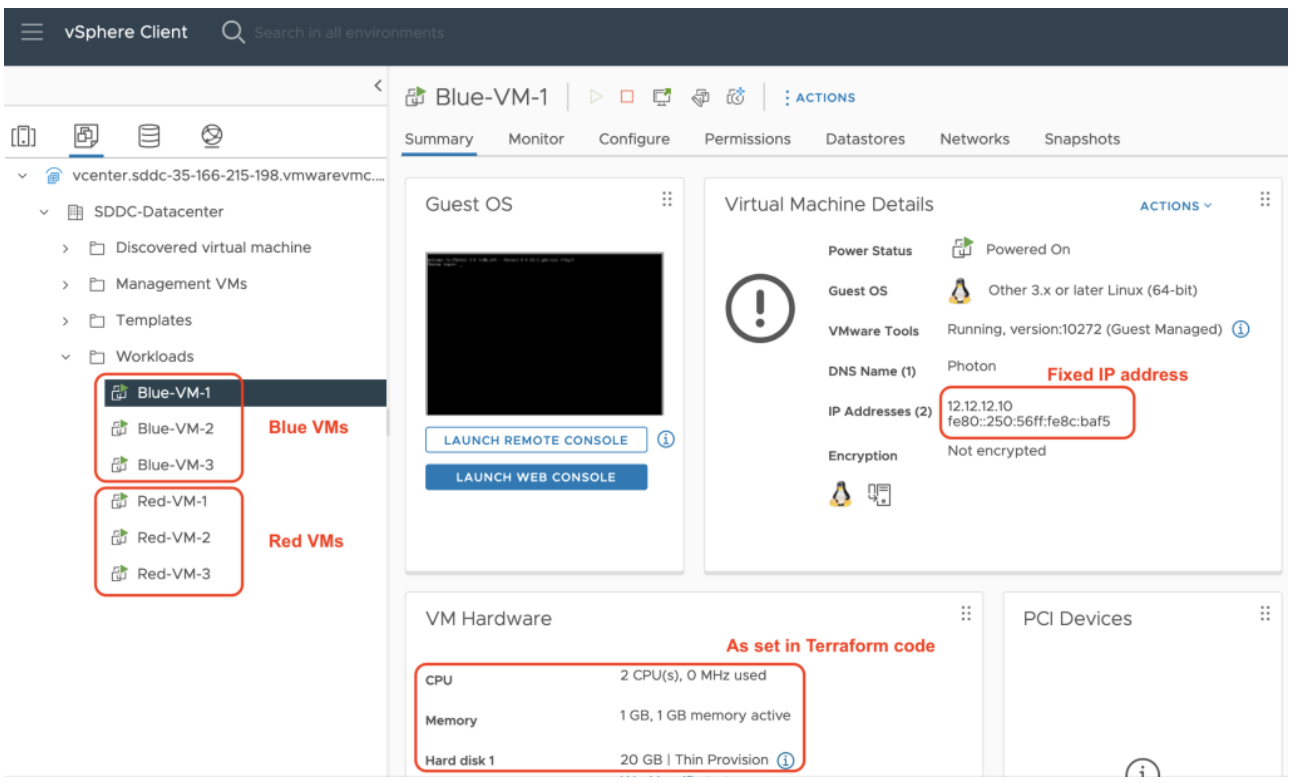
Apply vSphere tags that are used in the distributed firewall (DFW) rules that we set in phase2.

```
resource "vsphere_tag_category" "category" {
  name        = "ColoredVMs"
  cardinality = "SINGLE"
  description = "Managed by Terraform"

  associable_types = [
    "VirtualMachine"
  ]
}

resource "vsphere_tag" "tag" {
  name        = "vSphere_tag"
  category_id = vsphere_tag_category.category.id
  description = "Managed by Terraform"
}
```

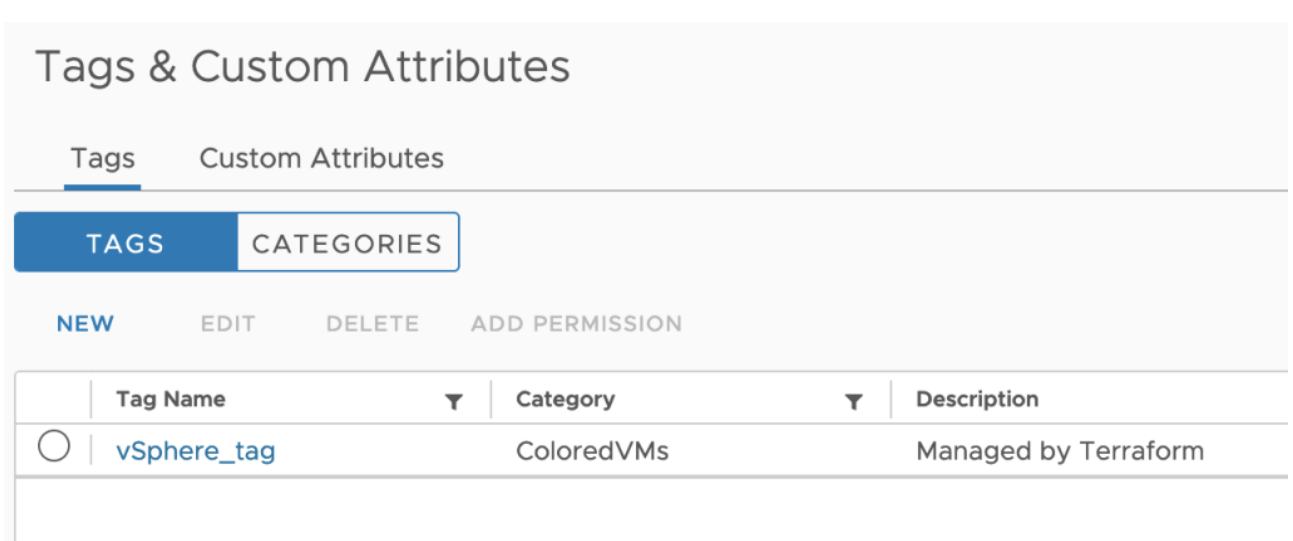
Deployment

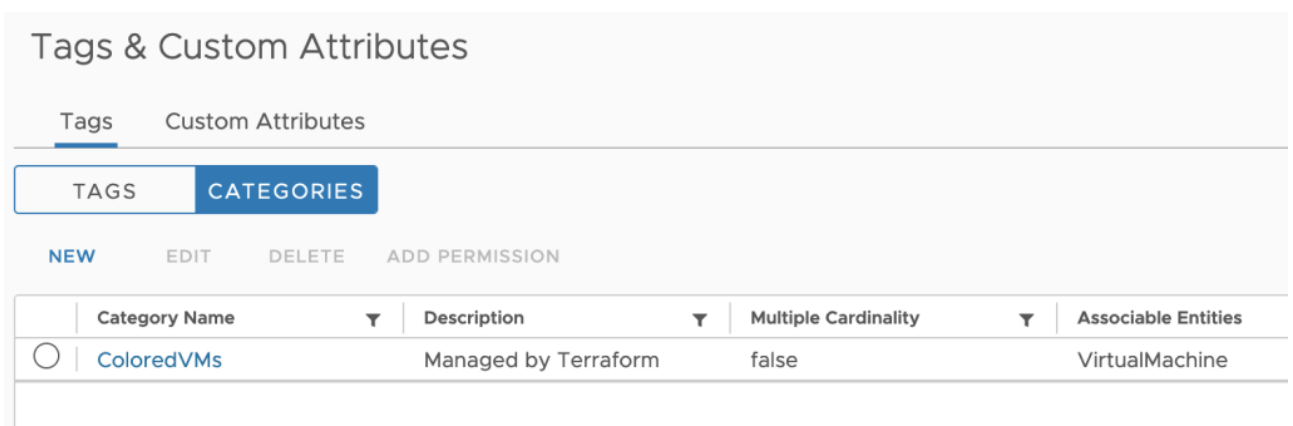
The `deploy.sh` script will execute terraform init and terraform apply. After 1m30s our 6 VMs are deployed and powered on. Note that with Terraform, a Virtual Machine has only 2 states: ON or GONE!

vCenter view



Tags





This concludes our blog post series on deploying a VMware Cloud on AWS environment from scratch using Terraform. We hope this series helps you in automating some or all your Day 1 and Day 2 tasks. As always, any feedback or comments are welcome!

Gilles Chekroun

Gilles Chekroun is Lead VMware Cloud on AWS Solutions Architect in the European team. He joined VMware in 2015 after spending 20 years at Cisco in the Data Centre Network...

Related Articles

VMware Cloud on AWS

VMware Cloud on AWS Free Trial is available now! Start your hybrid cloud journey today!

Nancy Cheng
July 11, 2023

VMware Cloud on AWS

VMware Lab Platform with VMware Cloud on AWS- Delivering on-demand labs anytime and anywhere

noeldnerb
July 6, 2023

VMware Cloud on AWS

Introducing the New and Improved Networking Experience for VMware Cloud Console

Nancy Cheng
June 9, 2023

VMware Cloud on AWS

Now Live – vCenter Federation for VMware Cloud on AWS

Aakash Chandhoke
June 8, 2023

VMware Cloud on AWS

Custom Managed Elastic DRS Policy: Improving Cost-Efficiency and Performance in VMware Cloud on AWS

Amitha Shetty
May 22, 2023

vmware®

Company

About Us

Executive Leadership

News & Stories

Investor Relations

Customer Stories

Diversity, Equity & Inclusion

Environment, Social & Governance

Careers

Blogs

Communities

Acquisitions

Office Locations

VMware Cloud Trust Center

COVID-19 Resources

Support

VMware Customer Connect

Support Policies

Product Documentation

Compatibility Guide

Terms & Conditions

California Transparency Act Statement

Twitter

YouTube

Facebook

LinkedIn

Contact Sales

Terms of Use

Your California Privacy Rights

Privacy

Accessibility

Trademarks

Glossary

Help

Feedback