# **Terraform Guide: Creating Single and Multiple VMs on vSphere (ESXi/vCenter)**

## **Prerequisites**

Before starting, make sure you have:

* Installed Terraform on your system
* Access to vSphere/ESXi or vCenter Server
* A working Ubuntu ISO uploaded to the datastore (e.g., ubuntu-22.04.iso)
* Basic knowledge of command-line usage

## **Folder Setup**

Create a folder for your Terraform project:

mkdir terraform-vm-demo  
cd terraform-vm-demo

## **Part 1: Creating a Single VM using Terraform**

### **Step 1: Write main.tf**

provider "vsphere" {  
 vsphere\_server = "----------"  
 user = "------"  
 password = "--------"  
 allow\_unverified\_ssl = true  
}  
  
data "vsphere\_datacenter" "dc" {  
 name = "ha-datacenter"  
}  
  
data "vsphere\_datastore" "datastore" {  
 name = "datastore1"  
 datacenter\_id = data.vsphere\_datacenter.dc.id  
}  
  
data "vsphere\_resource\_pool" "pool" {  
 datacenter\_id = data.vsphere\_datacenter.dc.id  
}  
  
data "vsphere\_network" "net" {  
 name = "VM Network"  
 datacenter\_id = data.vsphere\_datacenter.dc.id  
}  
  
resource "vsphere\_virtual\_machine" "ubuntu\_vm" {  
 name = "Aswini\_VM"  
 resource\_pool\_id = data.vsphere\_resource\_pool.pool.id  
 datastore\_id = data.vsphere\_datastore.datastore.id  
  
 num\_cpus = 1  
 memory = 1024  
 guest\_id = "ubuntu64Guest"  
 firmware = "bios"  
  
 network\_interface {  
 network\_id = data.vsphere\_network.net.id  
 adapter\_type = "vmxnet3"  
 }  
  
 disk {  
 label = "disk0"  
 size = 20  
 thin\_provisioned = true  
 }  
  
 cdrom {  
 datastore\_id = data.vsphere\_datastore.datastore.id  
 path = "[datastore1] ISO/ubuntu-22.04.iso"  
 }  
}

### **Step 2: Initialize and Apply**

terraform init  
terraform plan  
terraform apply

## **Part 2: Creating Multiple VMs using for\_each**

### **Step 1: Full Terraform Script to Create Multiple VMs**

provider "vsphere" {  
 vsphere\_server = "---------"  
 user = "------"  
 password = "---------"  
 allow\_unverified\_ssl = true  
}  
  
# Define the list of VMs  
locals {  
 vm\_list = [  
 { name = "Aswini\_VM\_1", memory = 1024, cpu = 1 },  
 { name = "Aswini\_VM\_2", memory = 2048, cpu = 2 },  
 { name = "Aswini\_VM\_3", memory = 4096, cpu = 2 }  
 ]  
}  
  
# Data Sources  
data "vsphere\_datacenter" "dc" {  
 name = "ha-datacenter"  
}  
  
data "vsphere\_datastore" "datastore" {  
 name = "datastore1"  
 datacenter\_id = data.vsphere\_datacenter.dc.id  
}  
  
data "vsphere\_resource\_pool" "pool" {  
 datacenter\_id = data.vsphere\_datacenter.dc.id  
}  
  
data "vsphere\_network" "net" {  
 name = "VM Network"  
 datacenter\_id = data.vsphere\_datacenter.dc.id  
}  
  
# Create multiple VMs with for\_each  
resource "vsphere\_virtual\_machine" "ubuntu\_vm" {  
 for\_each = { for vm in local.vm\_list : vm.name => vm }  
 name = each.value.name  
 resource\_pool\_id = data.vsphere\_resource\_pool.pool.id  
 datastore\_id = data.vsphere\_datastore.datastore.id  
  
 num\_cpus = each.value.cpu  
 memory = each.value.memory  
 guest\_id = "ubuntu64Guest"  
 firmware = "bios"  
  
 network\_interface {  
 network\_id = data.vsphere\_network.net.id  
 adapter\_type = "vmxnet3"  
 }  
  
 disk {  
 label = "disk0"  
 size = 20  
 thin\_provisioned = true  
 }  
  
 cdrom {  
 datastore\_id = data.vsphere\_datastore.datastore.id  
 path = "[datastore1] ISO/ubuntu-22.04.iso"  
 }  
}

### **Step 2: Initialize and Apply**

terraform init  
terraform plan  
terraform apply

## Explanation of Key Concepts

* provider: Specifies connection to vSphere (ESXi/vCenter)
* data sources: Fetch existing infrastructure info like datastore, network
* resource: Defines what to create (virtual machines)
* locals: Used to define reusable lists/values
* for\_each: Used to loop and create multiple VMs dynamically
* terraform init: Prepares your directory by downloading providers
* terraform apply: Creates the resources defined

## Notes

* If running .\terraform on Windows fails, add Terraform to your system PATH.
* Ensure your ISO file path is correct in cdrom.path
* You can update CPU, memory, or disk sizes as per your VM requirements