Terraform Lab with Libvirt and KVM

Objective

Learn to deploy virtual machines automatically using **Terraform** and **Libvirt/KVM** on a local Linux system. Students will write Infrastructure as Code (IaC) to define and provision VMs.

Prerequisites

- Linux machine with virtualization support (Ubuntu/Debian/Fedora)
- Installed:
  - `qemu-kvm`, `libvirt`, `virt-manager`
  - `terraform`
  - `terraform-provider-libvirt`
- Downloaded cloud image (e.g., Ubuntu cloud-init image)

Part 1 – Environment Setup

1.1 Install KVM and Libvirt

```
#Update your packet list.
sudo apt update
#Install Terraform.
mkdir terraform
cd terraform
```

```
wget
https://releases.hashicorp.com/terraform/1.8.5/terraform_1.8.5_linux_amd
64.zip
sudo apt-get intall unzip
unzip terraform_1.8.5_linux_amd64.zip
sudo mv terraform /usr/local/bin/
#Check your installation.
terraform version

#Install apparmor tools to handle access to VM images.
sudo apt install apparmor-utils

# Install Libvirt and KVM
sudo apt install qemu-kvm libvirt-daemon-system libvirt-clients bridge-
utils virt-manager
```

1.2 Add your user to the libvirt group

```
#User setting configuration
sudo usermod -aG libvirt $(whoami)
newgrp libvirt
```

1.3 Verify KVM is active

```
kvm-ok  # or
lsmod | grep kvm
```

Part 2 – Download a Cloud Image

```
#We download a base cloud image we will use in our configuration.
mkdir -p ~/VMs/images
cd ~/VMs/images
wget https://cloud-images.ubuntu.com/focal/current/focal-server-
cloudimg-amd64.img
```

Part 3 – Terraform directory

```
#Create the Terraform  directory that will host all files.
cd ~
mkdir ~/terraform-libvirt && cd ~/terraform-libvirt
touch main.tf variables.tf cloud_init.cfg
```

3.1 `main.tf`

```
#Main terraform file.
#Configure provider that will allow us to connect to local libvirt.
terraform {
  required_providers {
    libvirt = {
      source  = "dmacvicar/libvirt"
      version = "~> 0.6.3"
    }
  }
}

#We will use the local libvirt
#If remote machine, we should configure either a Tcp ou ssh
configuration.
provider "libvirt" {
  uri = "qemu:///system"
}

#The volume of our VM.
resource "libvirt_volume" "ubuntu_img" {
```

```
  name   = "ubuntu_vm.qcow2"
  pool   = "userpool"
  source = var.image_path
  format = "qcow2"
}

#Our Cloud init configuration to set the VM password "toto"
data "template_file" "cloud_init" {
  template = file("${path.module}/cloud_init.cfg")
}

#Cloud init iso to configure the VM.
resource "libvirt_cloudinit_disk" "cloudinit" {
  name      = "cloudinit.iso"
  pool      = "userpool"
  user_data = data.template_file.cloud_init.rendered
}

#Configuration of the VM.
resource "libvirt_domain" "vm" {
  name   = "vm"
  memory = 1024
  vcpu   = 1

  disk {
    volume_id = libvirt_volume.ubuntu_img.id
  }

  cloudinit = libvirt_cloudinit_disk.cloudinit.id

  network_interface {
    network_name = "default"
    wait_for_lease = true
  }

  console {
```

```
    type        = "pty"
    target_port = "0"
    target_type = "serial"
  }

  graphics {
    type = "vnc"
    listen_type = "address"
    autoport = true
  }
}
```

3.2 variables.tf

```
#Variable definition, the path to VM image we downloaded.
variable "image_path" {
  default = "/home/ubuntu/VMs/images/focal-server-cloudimg-amd64.img"
}
```

3.3 Cloud_init.cfg

```
#cloud-config
hostname: ubuntu
bootcmd:
  - echo 'ttyS0' >> /etc/securetty
users:
  - name: ubuntu
    gecos: Ubuntu User
    sudo: ["ALL=(ALL) NOPASSWD:ALL"]
    shell: /bin/bash
    lock_passwd: false
    passwd:
"$6$rLuihULIOlNPtf/4$K6n5xMUS4bVpoeUd2OOfklX5qXoxVitdAXV6BStR8FsEogz2R8Z
CT0kAuGPPnAVPX7csYmlf/xNEt.cFHuuwx0"
chpasswd:
  expire: false
```

Part 4 – Deploying the VM

4.1 Initialize Libvirt Default Storage Pool

```
cd ~
mkdir libvirt-pool
virsh  --connect qemu:///system pool-define-as --name userpool --type
dir --target ~/libvirt-pool
virsh --connect qemu:///system pool-start userpool
virsh --connect qemu:///system pool-autostart userpool
virsh --connect qemu:///system pool-info userpool
virsh --connect qemu:///system pool-list --all
sudo aa-complain /etc/apparmor.d/libvirt/*
```

4.2 Execute our terraform

```
cd terraform-libvirt/
terraform init
terraform plan
terraform apply
```

Part 5 - Check the VM

Check the VM IP address and connect to the VM. The password is "toto".

```
virsh domifaddr test-vm
ssh console vm
```