TERRAFORM ON OPENSTACK

To create a virtual network on openstack.

Providers.tf

provider "openstack" {

user\_name = "xxxxx"

tenant\_name = "xxxxxxx"

password = "xxxxxxxxx"

auth\_url = “xxxxxxxxxxxxxxxxxxxxxxx"

region = "xxxxxxxxx"

}

Main.tf

resource "openstack\_compute\_keypair\_v2" "terraform" {

name = "terraform"

public\_key = "${file("${var.ssh\_key\_file}.pub")}"

}

resource "openstack\_networking\_network\_v2" "terraform" {

name = "terraform"

admin\_state\_up = "true"

}

resource "openstack\_networking\_subnet\_v2" "terraform" {

name = "terraform"

network\_id = "${openstack\_networking\_network\_v2.terraform.id}"

cidr = "10.0.0.0/24"

ip\_version = 4

dns\_nameservers = ["8.8.8.8", "8.8.4.4"]

}

resource "openstack\_networking\_router\_v2" "terraform" {

name = "terraform"

admin\_state\_up = "true"

external\_network\_id = "${data.openstack\_networking\_network\_v2.terraform.id}"

}

resource "openstack\_networking\_router\_interface\_v2" "terraform" {

router\_id = "${openstack\_networking\_router\_v2.terraform.id}"

subnet\_id = "${openstack\_networking\_subnet\_v2.terraform.id}"

}

resource "openstack\_networking\_secgroup\_v2" "terraform" {

name = "terraform"

description = "Security group for the Terraform example instances"

}

resource "openstack\_networking\_secgroup\_rule\_v2" "terraform\_22" {

direction = "ingress"

ethertype = "IPv4"

protocol = "tcp"

port\_range\_min = 22

port\_range\_max = 22

remote\_ip\_prefix = "0.0.0.0/0"

security\_group\_id = "${openstack\_networking\_secgroup\_v2.terraform.id}"

}

resource "openstack\_networking\_secgroup\_rule\_v2" "terraform\_80" {

direction = "ingress"

ethertype = "IPv4"

protocol = "tcp"

port\_range\_min = 80

port\_range\_max = 80

remote\_ip\_prefix = "0.0.0.0/0"

security\_group\_id = "${openstack\_networking\_secgroup\_v2.terraform.id}"

}

resource "openstack\_networking\_secgroup\_rule\_v2" "terraform" {

direction = "ingress"

ethertype = "IPv4"

protocol = "icmp"

remote\_ip\_prefix = "0.0.0.0/0"

security\_group\_id = "${openstack\_networking\_secgroup\_v2.terraform.id}"

}

resource "openstack\_networking\_floatingip\_v2" "terraform" {

pool = "${var.pool}"

}

resource "openstack\_compute\_instance\_v2" "terraform" {

name = "terraform"

image\_name = "${var.image}"

flavor\_name = "${var.flavor}"

key\_pair = "${openstack\_compute\_keypair\_v2.terraform.name}"

security\_groups = ["${openstack\_networking\_secgroup\_v2.terraform.name}"]

network {

uuid = "${openstack\_networking\_network\_v2.terraform.id}"

}

}

resource "openstack\_compute\_floatingip\_associate\_v2" "terraform" {

floating\_ip = "${openstack\_networking\_floatingip\_v2.terraform.address}"

instance\_id = "${openstack\_compute\_instance\_v2.terraform.id}"

provisioner "remote-exec" {

connection {

host = "openstack\_networking\_floatingip\_v2.terraform.address"

user = "var.ssh\_user\_name"

private\_key = "file(var.ssh\_key\_file)"

}

inline = [

"sudo apt-get -y update",

"sudo apt-get -y install nginx",

"sudo service nginx start",

]

}

}

Datasources.tf

data "openstack\_networking\_network\_v2" "terraform" {

name = "${var.pool}"

}

Outputs.tf

output "address" {

value = "${openstack\_networking\_floatingip\_v2.terraform.address}"

}

Vars.tf

variable "image" {

default = "ubuntu-ssh"

}

variable "flavor" {

default = "tiny"

}

variable "ssh\_key\_file" {

default = "~/.ssh/id\_rsa"

}

variable "ssh\_user\_name" {

default = "ubuntu"

}

variable "pool" {

default = "external"

}