#This Terraform Code Deploys Basic VPC Infra.

provider "aws" {

access\_key = "${var.aws\_access\_key}"

secret\_key = "${var.aws\_secret\_key}"

region = "${var.aws\_region}"

}

resource "aws\_vpc" "default" {

cidr\_block = "${var.vpc\_cidr}"

enable\_dns\_hostnames = true

tags = {

Name = "${var.vpc\_name}"

}

}

resource "aws\_internet\_gateway" "default" {

vpc\_id = "${aws\_vpc.default.id}"

tags = {

Name = "${var.IGW\_name}"

}

}

resource "aws\_subnet" "subnets" {

count = 3 #0,1,2

vpc\_id = "${aws\_vpc.default.id}"

cidr\_block = "${element(var.cidrs, count.index)}"

availability\_zone = "${element(var.azs, count.index)}"

tags = {

Name = "Public-Subnet-${count.index+1}"

}

}

resource "aws\_route\_table" "terraform-public" {

vpc\_id = "${aws\_vpc.default.id}"

route {

cidr\_block = "0.0.0.0/0"

gateway\_id = "${aws\_internet\_gateway.default.id}"

}

tags = {

Name = "${var.Main\_Routing\_Table}"

}

}

resource "aws\_route\_table\_association" "routetable-assos1" {

count = 3

subnet\_id = "${element(aws\_subnet.subnets.\*.id, count.index)}"

route\_table\_id = "${aws\_route\_table.terraform-public.id}"

}

resource "aws\_security\_group" "allow\_all" {

name = "allow\_all"

description = "Allow all inbound traffic"

vpc\_id = "${aws\_vpc.default.id}"

ingress {

from\_port = 0

to\_port = 0

protocol = "-1"

cidr\_blocks = ["0.0.0.0/0"]

}

egress {

from\_port = 0

to\_port = 0

protocol = "-1"

cidr\_blocks = ["0.0.0.0/0"]

}

}

data "aws\_ami" "my\_ami" {

most\_recent = true

#name\_regex = "^mavrick"

owners = ["721834156908"]

}

resource "aws\_instance" "web-1" {

count = 1

#ami = "${data.aws\_ami.my\_ami.id}"

ami = "${lookup(var.amis,var.aws\_region,"us-east-1")}"

#availability\_zone = "us-east-1a"

instance\_type = "t2.micro"

key\_name = "LaptopKey"

subnet\_id = "${element(aws\_subnet.subnets.\*.id, count.index)}"

vpc\_security\_group\_ids = ["${aws\_security\_group.allow\_all.id}"]

associate\_public\_ip\_address = true

tags = {

Name = "Server-${count.index+1}"

Env = "Prod"

Owner = "Sree"

}

}

resource "null\_resource" "cluster" {

provisioner "file" {

source = "script.sh"

destination = "/tmp/script.sh"

connection {

type = "ssh"

user = "ec2-user"

#password = "India@123456"

private\_key = "${file("LaptopKey.pem")}"

host = "${aws\_instance.web-1.0.public\_ip}"

}

}

provisioner "remote-exec" {

inline = [

"chmod 700 /tmp/script.sh",

"sudo ./tmp/script.sh",

"sudo yum update -y",

"sudo yum install nginx -y",

"sudo service nginx start"

]

connection {

type = "ssh"

user = "ec2-user"

#password = "India@123456"

private\_key = "${file("LaptopKey.pem")}"

host = "${aws\_instance.web-1.0.public\_ip}"

}

}

provisioner "local-exec" {

command = <<EOH

echo "${aws\_instance.web-1.0.public\_ip}" >> details && echo "${aws\_instance.web-1.0.private\_ip}" >> details,

EOH

}

}

<https://stackoverflow.com/questions/52897221/terraform-count-index-function-used-in-provisioner-error>