Text

Description automatically generated

**Main.tf**

#Create and bootstrap webserver

resource "aws\_instance" "webserver" {

ami = data.aws\_ssm\_parameter.webserver-ami.value

instance\_type = "t3.micro"

key\_name = aws\_key\_pair.webserver-key.key\_name

associate\_public\_ip\_address = true

vpc\_security\_group\_ids = [aws\_security\_group.sg.id]

subnet\_id = aws\_subnet.subnet.id

provisioner "remote-exec" {

inline = [

"sudo yum -y install httpd && sudo systemctl start httpd",

"echo '<h1><center>My Test Website With Help From Terraform Provisioner</center></h1>' > index.html",

"sudo mv index.html /var/www/html/"

]

connection {

type = "ssh"

user = "ec2-user"

private\_key = file("~/.ssh/id\_rsa")

host = self.public\_ip

}

}

tags = {

Name = "webserver"

}

}

**Setup.tf:--**

provider "aws" {

region = "us-east-1"

}

#Create key-pair for logging into EC2 in us-east-1

resource "aws\_key\_pair" "webserver-key" {

key\_name = "webserver-key"

public\_key = file("~/.ssh/id\_rsa.pub")

}

#Get Linux AMI ID using SSM Parameter endpoint in us-east-1

data "aws\_ssm\_parameter" "webserver-ami" {

name = "/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86\_64-gp2"

}

#Create VPC in us-east-1

resource "aws\_vpc" "vpc" {

cidr\_block = "10.0.0.0/16"

enable\_dns\_support = true

enable\_dns\_hostnames = true

tags = {

Name = "terraform-vpc"

}

}

#Create IGW in us-east-1

resource "aws\_internet\_gateway" "igw" {

vpc\_id = aws\_vpc.vpc.id

}

#Get main route table to modify

data "aws\_route\_table" "main\_route\_table" {

filter {

name = "association.main"

values = ["true"]

}

filter {

name = "vpc-id"

values = [aws\_vpc.vpc.id]

}

}

#Create route table in us-east-1

resource "aws\_default\_route\_table" "internet\_route" {

default\_route\_table\_id = data.aws\_route\_table.main\_route\_table.id

route {

cidr\_block = "0.0.0.0/0"

gateway\_id = aws\_internet\_gateway.igw.id

}

tags = {

Name = "Terraform-RouteTable"

}

}

#Get all available AZ's in VPC for master region

data "aws\_availability\_zones" "azs" {

state = "available"

}

#Create subnet # 1 in us-east-1

resource "aws\_subnet" "subnet" {

availability\_zone = element(data.aws\_availability\_zones.azs.names, 0)

vpc\_id = aws\_vpc.vpc.id

cidr\_block = "10.0.1.0/24"

}

#Create SG for allowing TCP/80 & TCP/22

resource "aws\_security\_group" "sg" {

name = "sg"

description = "Allow TCP/80 & TCP/22"

vpc\_id = aws\_vpc.vpc.id

ingress {

description = "Allow SSH traffic"

from\_port = 22

to\_port = 22

protocol = "tcp"

cidr\_blocks = ["0.0.0.0/0"]

}

ingress {

description = "allow traffic from TCP/80"

from\_port = 80

to\_port = 80

protocol = "tcp"

cidr\_blocks = ["0.0.0.0/0"]

}

egress {

from\_port = 0

to\_port = 0

protocol = "-1"

cidr\_blocks = ["0.0.0.0/0"]

}

}

output "Webserver-Public-IP" {

value = aws\_instance.webserver.public\_ip

}

Text

Description automatically generated

Graphical user interface, application

Description automatically generated

Text

Description automatically generated