***Use a Terraform provisioner to install httpd on an EC2 instance and also create a security group.***

IAC( Infrastructure As Code)

#I have use static credientals u can use shared credientals

provider "aws" {

  region     = "us-east-1"

  access\_key = "input\_access\_key"

  secret\_key = "input\_secret\_key"

}

#locals value for tags

locals {

    common\_tag = {

        Name = "dev-Dept"

        Owner = "phenoix"

    }

}

#variable for instance type

variable "instancetype" {

    type = map

    default = {

        "dev-team"  = "t2.micro",

        "test-team" = "t2.nano",

        "prod-team" = "t2.small"

    }

}

#variable for image

variable "image" {

    type = list

    default = ["ami-00beae93a2d981137", "ami-0fe630eb857a6ec83", "ami-04b70fa74e45c3917"]

}

# key pair

resource "aws\_key\_pair" "devloper" {

    key\_name = "dev"

    public\_key = "ssh-rsa  devops@DESKTOP-6AJ7SR1"

}

# Security Group

resource "aws\_security\_group" "Provisioner-TF"{

    name = "Provis\_SG"

    description = "Security group using Terraform for provisioner instance."

    tags = local.common\_tag

}

# S/G inbound(ingress) Rule for HTTPS

resource "aws\_vpc\_security\_group\_ingress\_rule" "HTTPS" {

  security\_group\_id = aws\_security\_group.Provisioner-TF.id

  cidr\_ipv4         = "0.0.0.0/0"

  from\_port         = 443

  ip\_protocol       = "tcp"

  to\_port           = 443

}

#  S/G inbound(ingress) Rule for HTTP

resource "aws\_vpc\_security\_group\_ingress\_rule" "HTTP" {

  security\_group\_id = aws\_security\_group.Provisioner-TF.id

  cidr\_ipv4         = "0.0.0.0/0"

  from\_port         = 80

  ip\_protocol       = "tcp"

  to\_port           = 80

}

#  S/G inbound(ingress) Rule for SSH

resource "aws\_vpc\_security\_group\_ingress\_rule" "SSH" {

  security\_group\_id = aws\_security\_group.Provisioner-TF.id

  cidr\_ipv4         = "0.0.0.0/0"

  from\_port         = 22

  ip\_protocol       = "tcp"

  to\_port           = 22

}

# S/G outbound (egress) Rule allow all

resource "aws\_vpc\_security\_group\_egress\_rule" "allow\_all\_traffic\_ipv4" {

    security\_group\_id = aws\_security\_group.Provisioner-TF.id

    cidr\_ipv4 = "0.0.0.0/0"

    ip\_protocol = "-1"

}

#resource for instance

resource "aws\_instance" "dev" {

    ami = var.image[0]

    instance\_type = var.instancetype["dev-team"]

    key\_name = "dev"

    vpc\_security\_group\_ids = [aws\_security\_group.Provisioner-TF.id]

    tags = local.common\_tag

    # File provisioner

    provisioner "file" {

        source = "D:/Terraform/Provisioners/index.html"

        destination = "/tmp/index.html"

    }

    # remote Provisioner

    provisioner "remote-exec"{

        inline = [

            "sudo yum install httpd -y",

            "sudo systemctl start httpd",

            "sudo systemctl enable httpd",

            "sudo cp /tmp/index.html /var/www/html",

            "sudo systemctl restart httpd"

        ]

    }

    connection{

        host = self.public\_ip

        user = "ec2-user"

        type = "ssh"

        private\_key = file("./dev")

    }

}

# output

output "instance-id" {

    value = aws\_instance.dev.id

}















