**#Terraform code to create a vpc, subnet, internatgateway, route tables, associating the route table with the subnets, creating security groups and launching ec2 instances using variables**

terraform {

required\_providers {

aws = {

source = "hashicorp/aws"

version = "5.54.1"

}

}

}

provider "aws" {

region = var.region

access\_key = var.access\_key

secret\_key = var.secret\_key

}

variable "region" {}

variable "access\_key" {}

variable "secret\_key" {}

variable "vpc\_cidr" {}

variable "subnet\_cidr" {}

variable "availability\_zone" {}

variable "alltraffic" {}

variable "security\_group\_name" {}

variable "all\_ports" {}

variable "protocol" {}

variable "ami" {}

variable "instance\_type" {}

variable "public\_ip" {}

variable "pemkey" {}

variable "instance\_count" {}

variable "vpc\_name" {}

variable "subnet\_name" {}

variable "igw\_name" {}

variable "route\_table\_name" {}

variable "instance\_name" {}

resource "aws\_vpc" "newvpc" {

cidr\_block = var.vpc\_cidr

tags = {

Name = var.vpc\_name

}

}

resource "aws\_subnet" "newsubnet" {

vpc\_id = aws\_vpc.newvpc.id

cidr\_block = var.subnet\_cidr

availability\_zone = var.availability\_zone

tags = {

Name = var.subnet\_name

}

}

resource "aws\_internet\_gateway" "newgateway" {

vpc\_id = aws\_vpc.newvpc.id

tags = {

Name = var.igw\_name

}

}

resource "aws\_route\_table" "newroutetable" {

vpc\_id = aws\_vpc.newvpc.id

route {

cidr\_block = var.alltraffic

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

}

tags = {

Name = var.route\_table\_name

}

}

resource "aws\_route\_table\_association" "newassociation" {

route\_table\_id = aws\_route\_table.newroutetable.id

subnet\_id = aws\_subnet.newsubnet.id

}

resource "aws\_security\_group" "newsecurity" {

name = var.security\_group\_name

vpc\_id = aws\_vpc.newvpc.id

ingress {

from\_port = var.all\_ports

protocol = var.protocol

to\_port = var.all\_ports

cidr\_blocks = [var.alltraffic]

}

egress {

from\_port = var.all\_ports

protocol = var.protocol

to\_port = var.all\_ports

cidr\_blocks = [var.alltraffic]

}

}

resource "aws\_instance" "newinstance" {

ami = var.ami

instance\_type = var.instance\_type

availability\_zone = var.availability\_zone

subnet\_id = aws\_subnet.newsubnet.id

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

associate\_public\_ip\_address = var.public\_ip

key\_name = var.pemkey

count = var.instance\_count

tags = {

Name = var.instance\_name

}

}

**Variables file for the above program**

=================================

region = "us-east-2"

access\_key = "AKIAUS66WDVERQYFL5EA"

secret\_key = "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"

vpc\_cidr = "10.0.0.0/24"

subnet\_cidr = "10.0.0.0/25"

availability\_zone = "us-east-2b"

alltraffic = "0.0.0.0/0"

security\_group\_name = "new-security"

all\_ports = 0

protocol = "-1"

ami = "ami-04f167a56786e4b09"

instance\_type = "t2.micro"

public\_ip = true

pemkey = "ohiokey"

instance\_count = 1

vpc\_name = "newvpc"

subnet\_name = "newsubnet"

igw\_name = "newgateway"

route\_table\_name = "newroutetable"

instance\_name = "newinstance"

=========================================================================================================

**Terraform code to create an AMI based on the previous code and create an autoscaling group by creating the launch configuration template**

terraform {

required\_providers {

aws = {

source = "hashicorp/aws"

version = "5.54.1"

}

}

}

provider "aws" {

region = var.region

access\_key = var.access\_key

secret\_key = var.secret\_key

}

variable "region" {}

variable "access\_key" {}

variable "secret\_key" {}

variable "vpc\_cidr" {}

variable "subnet\_cidr" {}

variable "availability\_zone" {}

variable "alltraffic" {}

variable "security\_group\_name" {}

variable "all\_ports" {}

variable "protocol" {}

variable "ami" {}

variable "instance\_type" {}

variable "public\_ip" {}

variable "pemkey" {}

variable "vpc\_name" {}

variable "subnet\_name" {}

variable "igw\_name" {}

variable "route\_table\_name" {}

variable "instance\_name" {}

variable "launch\_template\_name\_prefix" {}

variable "resource\_type" {}

variable "ami\_name" {}

variable "launch\_template\_name" {}

variable "asg\_count" {}

resource "aws\_vpc" "newvpc" {

cidr\_block = var.vpc\_cidr

tags = {

Name = var.vpc\_name

}

}

resource "aws\_subnet" "newsubnet" {

vpc\_id = aws\_vpc.newvpc.id

cidr\_block = var.subnet\_cidr

availability\_zone = var.availability\_zone

tags = {

Name = var.subnet\_name

}

}

resource "aws\_internet\_gateway" "newgateway" {

vpc\_id = aws\_vpc.newvpc.id

tags = {

Name = var.igw\_name

}

}

resource "aws\_route\_table" "newroutetable" {

vpc\_id = aws\_vpc.newvpc.id

route {

cidr\_block = var.alltraffic

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

}

tags = {

Name = var.route\_table\_name

}

}

resource "aws\_route\_table\_association" "newassociation" {

route\_table\_id = aws\_route\_table.newroutetable.id

subnet\_id = aws\_subnet.newsubnet.id

}

resource "aws\_security\_group" "newsecurity" {

name = var.security\_group\_name

vpc\_id = aws\_vpc.newvpc.id

ingress {

from\_port = var.all\_ports

protocol = var.protocol

to\_port = var.all\_ports

cidr\_blocks = [var.alltraffic]

}

egress {

from\_port = var.all\_ports

protocol = var.protocol

to\_port = var.all\_ports

cidr\_blocks = [var.alltraffic]

}

}

resource "aws\_instance" "newinstance" {

ami = var.ami

instance\_type = var.instance\_type

availability\_zone = var.availability\_zone

subnet\_id = aws\_subnet.newsubnet.id

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

associate\_public\_ip\_address = var.public\_ip

key\_name = var.pemkey

user\_data = <<-EOF

#!/bin/bash

sudo apt-get update

sudo apt-get install -y tomcat10

sudo apt-get install -y tomcat10-admin

EOF

tags = {

Name = var.instance\_name

}

}

resource "aws\_ami\_from\_instance" "tomcat" {

name = var.ami\_name

source\_instance\_id = aws\_instance.newinstance.id

tags = {

Name = var.ami\_name

}

}

resource "aws\_launch\_template" "newtemplate" {

name\_prefix = var.launch\_template\_name\_prefix

image\_id = aws\_ami\_from\_instance.tomcat.id

instance\_type = var.instance\_type

key\_name = var.pemkey

network\_interfaces {

associate\_public\_ip\_address = var.public\_ip

security\_groups = [aws\_security\_group.newsecurity.id]

}

tag\_specifications {

resource\_type = var.resource\_type

tags = {

Name= var.launch\_template\_name

}

}

}

resource "aws\_autoscaling\_group" "newscalling" {

max\_size = var.asg\_count

min\_size = var.asg\_count

desired\_capacity = var.asg\_count

vpc\_zone\_identifier = [aws\_subnet.newsubnet.id]

launch\_template {

id = aws\_launch\_template.newtemplate.id

version = "$Latest"

}

tag {

key = "Name"

propagate\_at\_launch = true

value = "ASG\_Instance"

}

}

**Variables for the above file** =============================

region = "us-east-2"

access\_key = "AKIAUS66WDVERQYFL5EA"

secret\_key = "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"

vpc\_cidr = "10.0.0.0/24"

subnet\_cidr = "10.0.0.0/25"

availability\_zone = "us-east-2b"

alltraffic = "0.0.0.0/0"

security\_group\_name = "new-security"

all\_ports = 0

protocol = "-1"

ami = "ami-04f167a56786e4b09"

instance\_type = "t2.micro"

public\_ip = true

pemkey = "ohiokey"

vpc\_name = "newvpc"

subnet\_name = "newsubnet"

igw\_name = "newgateway"

route\_table\_name = "newroutetable"

instance\_name = "newinstance"

launch\_template\_name\_prefix = "newtemplate-lt"

resource\_type = "instance"

ami\_name = "tomcat"

launch\_template\_name = "newtemplate"

asg\_count = 2