**Main.tf**

terraform {

required\_providers {

aws = {

source = "hashicorp/aws"

version = "~> 3.5.0"

}

}

}

# Configure the AWS Provider

provider "aws"{

region = "eu-west-2"

}

# Create a Key pair

resource "aws\_key\_pair" "abdul-aws-key-pair" {

key\_name = "abdul-aws-key-pair"

public\_key = tls\_private\_key.rsa.public\_key\_openssh

}

# RSA key of size 4096 bits

resource "tls\_private\_key" "rsa" {

algorithm = "RSA"

rsa\_bits = 4096

}

# Create a local file to store the key pair which will be used to access the instances cia ssh

resource "local\_file" "abdul-aws-key-pair" {

content = tls\_private\_key.rsa.private\_key\_pem

filename = "abdul-aws-key-pair"

}

# Get latest Amazon Linux 2 AMI

data "aws\_ami" "amazon-linux-2" {

most\_recent = true

owners = ["amazon"]

filter {

name = "name"

values = ["amzn2-ami-hvm\*"]

}

}

# Define the security group for the EC2 Instance

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

name = "aws-sg-webserver"

description = "Allow incoming connections"

vpc\_id = aws\_default\_vpc.default.id

ingress {

from\_port = 80

to\_port = 80

protocol = "tcp"

cidr\_blocks = ["0.0.0.0/0"]

description = "Allow incoming HTTP connections"

}

ingress {

from\_port = 22

to\_port = 22

protocol = "tcp"

cidr\_blocks = ["0.0.0.0/0"]

description = "Allow incoming SSH connections (Linux)"

}

egress {

from\_port = 0

to\_port = 0

protocol = "-1"

cidr\_blocks = ["0.0.0.0/0"]

}

tags = {

Name = "Webserver-sg"

}

}

resource "aws\_default\_vpc" "default" {

}

resource "aws\_instance" "vm-server" {

ami = data.aws\_ami.amazon-linux-2.id

instance\_type = "t2.micro"

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

key\_name = aws\_key\_pair.abdul-aws-key-pair.key\_name

user\_data = file("userdata.tpl")

tags = {

Name = "aws-webserver-demo"

}

}

resource "aws\_launch\_template" "aws-launch-template" {

name = "aws-launch-template"

image\_id = data.aws\_ami.amazon-linux-2.id

instance\_type = "t2.micro"

key\_name = aws\_key\_pair.abdul-aws-key-pair.key\_name

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

tag\_specifications {

resource\_type = "instance"

tags = {

Name = "aws-webserver-demo"

}

}

user\_data = filebase64("userdata.tpl")

}

# Use an autoscaling group of 2 minimum EC2 instances in 3 availability zones

resource "aws\_autoscaling\_group" "aws-autoscaling-group" {

availability\_zones = ["eu-west-2a", "eu-west-2b", "eu-west-2c"]

desired\_capacity = 2

max\_size = 3

min\_size = 2

launch\_template {

id = aws\_launch\_template.aws-launch-template.id

version = "$Latest"

}

}

Userdata.tpl

#!/bin/sh

# Install a LAMP stack

amazon-linux-extras install -y lamp-mariadb10.2-php7.2 php7.2

yum -y install httpd php-mbstring

# Start the web server

chkconfig httpd on

systemctl start httpd

# Install the web pages for our lab

if [ ! -f /var/www/html/immersion-day-app-php7.tar.gz ]; then

cd /var/www/html

wget

https://aws-joozero.s3.ap-northeast-2.amazonaws.com/immersion-day-app-p

hp7.tar.gz

tar xvfz immersion-day-app-php7.tar.gz

fi

# Install the AWS SDK for PHP

if [ ! -f /var/www/html/aws.zip ]; then

cd /var/www/html

mkdir vendor

cd vendor

wget https://docs.aws.amazon.com/aws-sdk-php/v3/download/aws.zip

unzip aws.zip

fi

# Update existing packages

yum -y update