resource "aws\_security\_group" "TerraSecGrp"

{

name = "Wordpress-Secgrp"

description = "Allow all inbound traffic"

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

ingress

{

from\_port = 22

to\_port = 22

protocol = "tcp"

cidr\_blocks = ["0.0.0.0/0"]

}

ingress

{

from\_port = 80

to\_port = 80

protocol = "tcp"

cidr\_blocks = ["0.0.0.0/0"]

}

ingress

{

from\_port = 3306

to\_port = 3306

protocol = "tcp"

cidr\_blocks = ["10.20.0.0/16"]

}

egress

{

from\_port = 0

to\_port = 0

protocol = "-1"

cidr\_blocks = ["0.0.0.0/0"]

}

tags

{

Name = "CreatedByTerraform"

}

}

resource "aws\_security\_group" "ELBSecGrp"

{

name = "Wordpresselb-Secgrp"

description = "Allow 80 traffic"

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

ingress

{

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"]

}

tags

{

Name = "WordpressELB-Secgrp"

}

}

resource "aws\_security\_group" "DB-Secgroup"

{

name = "DB-Secgroup"

description = "for database"

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

ingress

{

from\_port = 3306

to\_port = 3306

protocol = "tcp"

cidr\_blocks = ["10.20.0.0/16"]

}

tags

{

Name = "DB-Secgrp"

}

}

resource "aws\_instance" "wordpress"

{

ami = "ami-8d948ced"

depends\_on = ["aws\_security\_group.TerraSecGrp","aws\_key\_pair.wordpress-KP"]

instance\_type = "t2.micro"

associate\_public\_ip\_address = "true"

subnet\_id = "${aws\_subnet.public\_subnet.id}"

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

key\_name = "${aws\_key\_pair.wordpress-KP.id}"

tags

{

Name = "Wordpress through terrafrom"

Owner = "Ramanuj Shastri"

Application = "Wordpress"

Created = "This instance is created through terraform"

}

user\_data = "${file("/home/ubuntu/terraform/two-tier-architecture/install.sh")}"

}

resource "aws\_key\_pair" "wordpress-KP"

{

key\_name = "wordpress"

public\_key = "${file("${var.ssh\_key}")}"

}

resource "aws\_db\_subnet\_group" "dbsubnet-grp"

{

# name = "main"

subnet\_ids = ["${aws\_subnet.private\_subnet1.id}","${aws\_subnet.private\_subnet2.id}"]

tags {

Name = "My DB subnet group"

}

}

resource "aws\_db\_instance" "Wordpress-DB"

{

depends\_on = ["aws\_db\_subnet\_group.dbsubnet-grp"]

identifier = "wordpress-db"

allocated\_storage = 10

storage\_type = "gp2"

engine = "mysql"

engine\_version = "5.7"

db\_subnet\_group\_name = "${aws\_db\_subnet\_group.dbsubnet-grp.id}"

instance\_class = "db.t2.micro"

name = "WordpressDB"

username = "wordpress"

password = "wordpress"

parameter\_group\_name = "default.mysql5.7"

publicly\_accessible = "false"

vpc\_security\_group\_ids = ["${aws\_security\_group.DB-Secgroup.id}"]

skip\_final\_snapshot = "true"

}

resource "aws\_elb" "Wordpress-ELB"

{

depends\_on = ["aws\_security\_group.ELBSecGrp"]

name = "wordpress-elb"

subnets = ["${aws\_subnet.public\_subnet.id}"]

security\_groups = ["${aws\_security\_group.ELBSecGrp.id}"]

listener

{

instance\_port = 80

instance\_protocol = "http"

lb\_port = 80

lb\_protocol = "http"

}

health\_check {

healthy\_threshold = 5

unhealthy\_threshold = 2

timeout = 3

target = "HTTP:80/"

interval = 30

}

instances = ["${aws\_instance.wordpress.id}"]

cross\_zone\_load\_balancing = false

idle\_timeout = 400

connection\_draining = true

connection\_draining\_timeout = 400

tags {

Name = "wordpress-elb"

}

}