Terraform Script for Wordpress

ATTACH AmazonEC2FullAccess, AmazonElasticFileSystemFullAccess, AmazonRDSFullAccess role to your ec2

```
mkdir terraform-complex

cd terraform-complex



vim providers.tf

###################providers.tf

provider "aws" {

            region = "us-east-1"

}


vim vpcmain.tf

########################vpcmain.tf


resource "aws_vpc" "Main" {

  cidr_block       = var.main_vpc_cidr

  instance_tenancy = "default"

 }


resource "aws_internet_gateway" "IGW" {

   vpc_id =  aws_vpc.Main.id

 }



resource "aws_subnet" "publicsubnet1" {

  vpc_id =  aws_vpc.Main.id

  cidr_block = "${var.public_subnet1}"

  tags = {

    Name = "publicsubnetA"
  }
 }

resource "aws_subnet" "publicsubnet2" {
  vpc_id =  aws_vpc.Main.id
  cidr_block = "${var.public_subnet2}"
  tags = {
    Name = "publicsubnetB"
  }
 }

resource "aws_subnet" "privatesubnet1" {
  vpc_id =  aws_vpc.Main.id
  cidr_block = "${var.private_subnet1}"
  tags = {
    Name = "privatesubnetA"
  }
 }

resource "aws_subnet" "privatesubnet2" {
  vpc_id =  aws_vpc.Main.id
  cidr_block = "${var.private_subnet2}"
  tags = {
    Name = "privatesubnetB"
  }
 }

 resource "aws_route_table" "PublicRT" {
    vpc_id =  aws_vpc.Main.id
         route {
    cidr_block = "0.0.0.0/0"
    gateway_id = aws_internet_gateway.IGW.id
     }
 }

resource "aws_route_table_association" "PublicRTassociation" {
  subnet_id = aws_subnet.publicsubnet1.id
  route_table_id = aws_route_table.PublicRT.id
}

resource "aws_route_table_association" "PublicRTassociation1" {
  subnet_id = aws_subnet.publicsubnet2.id
  route_table_id = aws_route_table.PublicRT.id
}


##########################################################################################

vim variables.tf
#################variables.tf

variable "region" {}
variable "main_vpc_cidr" {}
variable "public_subnet1" {}
variable "public_subnet2" {}
variable "private_subnet1" {}
variable "private_subnet2" {}

variable "engine" {}
variable "engine_version" {}
variable "instance_class" {}
variable "name"  {}
variable "username" {}

variable "password" {}

variable "parameter_group_name" {}



##############################################################################
#


vim terraform.tfvars

##################terraform.tfvars


region = "us-east-1"

main_vpc_cidr = "10.0.0.0/16"

public_subnet1 = "10.0.0.0/24"

public_subnet2 = "10.0.2.0/24"

private_subnet1 = "10.0.1.0/24"

private_subnet2 = "10.0.3.0/24"

engine            = "mysql"

engine_version      = "5.7"

instance_class      = "db.t3.micro"

name              = "mydb"

username           = "user1"

password           = "password"

parameter_group_name = "default.mysql5.7"


##############################################################################
#



vim ec2main.tf

######################ec2main.tf
```

```
resource "aws_instance" "testinstance" {
   ami = "ami-04505e74c0741db8d"
   instance_type = "t2.micro"
   subnet_id = aws_subnet.publicsubnet1.id
   vpc_security_group_ids = [ aws_security_group.ec2.id ]
   key_name="ab"
   tags= {
      Name = "testinstance"
   }
}

##########################################################################################
##

vim securitymain.tf
################################securitymain.tf

resource "aws_security_group" "ec2" {
  name        = "allow_efs"
  description = "Allow efs outbound traffic"
  vpc_id      = aws_vpc.Main.id
  ingress {
    cidr_blocks = ["0.0.0.0/0"]
    from_port = 22
    to_port = 22
    protocol = "tcp"
   }
  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 = "allow_efs"
  }
}

resource "aws_security_group" "efs" {
   name = "efs-sg"
   description= "Allos inbound efs traffic from ec2"
   vpc_id = aws_vpc.Main.id

   ingress {
     security_groups = [aws_security_group.ec2.id]
     from_port = 2049
     to_port = 2049
     protocol = "tcp"
   }

   egress {
     security_groups = [aws_security_group.ec2.id]
     from_port = 0
     to_port = 0
     protocol = "-1"
   }
```

```
 }



vim rdsmain.tf
######################################rdsmain.tf
###############add amazonrdsfullaccess role to ec2############

resource "aws_db_instance" "default" {
  allocated_storage    = 10
  engine             = var.engine
  engine_version       = var.engine_version
  instance_class       = var.instance_class
  db_name               = var.name
  username           = var.username
  password           = var.password
  parameter_group_name = var.parameter_group_name
  db_subnet_group_name   = aws_db_subnet_group.default.name
  vpc_security_group_ids = [ aws_security_group.ec2.id ]
  skip_final_snapshot       = true



}

resource "aws_db_subnet_group" "default" {
  name       = "main"
  subnet_ids = [aws_subnet.privatesubnet1.id, aws_subnet.privatesubnet2.id]

  tags = {
    Name = "My DB subnet group"
  }
```

```
}



vim efsmain.tf
###################efsmain.tf

resource "aws_efs_file_system" "efs" {
   creation_token = "efs"
   performance_mode = "generalPurpose"
   throughput_mode = "bursting"
   encrypted = "true"
 tags = {
     Name = "EFS"
   }
 }



resource "aws_efs_mount_target" "efs-mt" {

   file_system_id  = aws_efs_file_system.efs.id
   subnet_id = aws_subnet.publicsubnets.id
   security_groups = [aws_security_group.efs.id]
 }

resource "null_resource" "configure_nfs" {
  depends_on = [aws_efs_mount_target.efs-mt]
  connection {
    type     = "ssh"
    user     = "ubuntu"
    host     = aws_instance.demo.public_ip
    private_key = tls_private_key.my_key.private_key_pem
 }
  provisioner "remote-exec" {
  inline = [
    "sudo apt update",
    "sudo apt install apache2 -y",
    "sudo systemctl start apache2",
    "sudo systemctl enable apache2",
    "sudo apt install nfs-common -y -q",


      "cd /var/www/html",
      "sudo wget https://wordpress.org/latest.zip",
      "sudo unzip latest.zip"
    "sudo mount -t nfs -o
nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport
${aws_efs_file_system.efs.dns_name}:/  /var/www/html",
    "sudo chmod 666 /etc/fstab",
    "sudo echo '${aws_efs_file_system.efs.dns_name}:/ /var/www/html nfs4 defaults,_netdev 0 0' >>
/etc/fstab",
  ]
 }
}




vim elbmain.tf
#######################elbmain.tf
resource "aws_elb" "classicbar" {
  name              = "classicelb"
  availability_zones = ["us-east-1a", "us-east-1b", "us-east-1c"]



listener {
    instance_port     = 8000
    instance_protocol = "http"
    lb_port           = 80
    lb_protocol       = "http"
  }



  health_check {
    healthy_threshold   = 2
    unhealthy_threshold = 2
    timeout             = 3
    target              = "HTTP:8000/"
    interval            = 30
  }


  cross_zone_load_balancing   = true
  idle_timeout                = 400
  connection_draining         = true
  connection_draining_timeout = 400

  tags = {
    Name = "classicelb"
  }
}
```