A screenshot of a computer

Description automatically generated

Steps

One vpc

2 public subnets, 2 private subnets

One public route table, 2 private route tables

2 elastic ip

2 nat gateways

Map subnets to route tables

Create one bastion server in any of the public subnets

Create launch templates for applications in private subnets

Create target groups

Create autoscaling groups and attach to target groups

Create load balancers

Create listeners

provider "aws" {

  region = "us-east-1"

}

## Create VPC

resource "aws\_vpc" "myvpc" {

  cidr\_block = "10.0.0.0/16"

  tags = {

    Name = "My VPC"

  }

}

##  Create public and private subnets

resource "aws\_subnet" "pubsub1" {

  vpc\_id                  = aws\_vpc.myvpc.id

  cidr\_block              = "10.0.0.0/24"

  availability\_zone       = "us-east-1a"

  map\_public\_ip\_on\_launch = true

  tags = {

    Name = "Public Subnet1"

  }

}

resource "aws\_subnet" "pubsub2" {

  vpc\_id                  = aws\_vpc.myvpc.id

  cidr\_block              = "10.0.1.0/24"

  availability\_zone       = "us-east-1b"

  map\_public\_ip\_on\_launch = true

  tags = {

    Name = "Public Subnet2"

  }

}

resource "aws\_subnet" "prvsub1" {

  vpc\_id                  = aws\_vpc.myvpc.id

  cidr\_block              = "10.0.2.0/24"

  availability\_zone       = "us-east-1a"

  map\_public\_ip\_on\_launch = false

  tags = {

    Name = "Private Subnet1"

  }

}

resource "aws\_subnet" "prvsub2" {

  vpc\_id                  = aws\_vpc.myvpc.id

  cidr\_block              = "10.0.3.0/24"

  availability\_zone       = "us-east-1b"

  map\_public\_ip\_on\_launch = false

  tags = {

    Name = "Private Subnet2"

  }

}

## Create Internet gateway

resource "aws\_internet\_gateway" "igw" {

  vpc\_id = aws\_vpc.myvpc.id

  tags = {

    Name = "IGW"

  }

}

# Create public route table

resource "aws\_route\_table" "RTpub" {

  vpc\_id = aws\_vpc.myvpc.id

  route {

    cidr\_block = "0.0.0.0/0"

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

  }

  tags = {

    Name = "Public Route"

  }

}

# Create Elastic IP and Nat gateways

resource "aws\_eip" "nateip1" {

  domain = "vpc"

}

resource "aws\_eip" "nateip2" {

  domain = "vpc"

}

resource "aws\_nat\_gateway" "nat1" {

  allocation\_id = aws\_eip.nateip1.id

  subnet\_id = aws\_subnet.pubsub1.id

  tags = {

    Name = "Nat1 VPC"

  }

}

resource "aws\_nat\_gateway" "nat2" {

  allocation\_id = aws\_eip.nateip2.id

  subnet\_id = aws\_subnet.pubsub2.id

  tags = {

    Name = "Nat2 VPC"

  }

}

# Create Private route table and attach the NAt gateway route

resource "aws\_route\_table" "RTprv1" {

  vpc\_id = aws\_vpc.myvpc.id

  route {

    cidr\_block = "0.0.0.0/0"

    nat\_gateway\_id = aws\_nat\_gateway.nat1.id

  }

  tags = {

    Name = "Private Route1"

  }

}

resource "aws\_route\_table" "RTprv2" {

  vpc\_id = aws\_vpc.myvpc.id

  route {

    cidr\_block = "0.0.0.0/0"

    nat\_gateway\_id = aws\_nat\_gateway.nat2.id

  }

  tags = {

    Name = "Private Route2"

  }

}

## Attach the subnets to the route tables

resource "aws\_route\_table\_association" "rta1" {

  subnet\_id      = aws\_subnet.pubsub1.id

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

}

resource "aws\_route\_table\_association" "rta2" {

  subnet\_id      = aws\_subnet.pubsub2.id

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

}

resource "aws\_route\_table\_association" "rta3" {

  subnet\_id      = aws\_subnet.prvsub1.id

  route\_table\_id = aws\_route\_table.RTprv1.id

}

resource "aws\_route\_table\_association" "rta4" {

  subnet\_id      = aws\_subnet.prvsub2.id

  route\_table\_id = aws\_route\_table.RTprv2.id

}

## CReate security groups

resource "aws\_security\_group" "webSg" {

  name   = "web"

  vpc\_id = aws\_vpc.myvpc.id

  ingress {

    description = "HTTP from VPC"

    from\_port   = 80

    to\_port     = 80

    protocol    = "tcp"

    cidr\_blocks = ["0.0.0.0/0"]

  }

  ingress {

    description = "SSH"

    from\_port   = 22

    to\_port     = 22

    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 = "Web-sg"

  }

}

# resource "aws\_s3\_bucket" "example" {

#   bucket = "abhisheksterraform2023project"

# }

## Create Key pair for instances

resource "aws\_key\_pair" "kp" {

  key\_name = "key1"

  public\_key = file("C:/Users/Vijesh/.ssh/id\_rsa.pub")

}

## Create a bastion instance to login to the private instances

resource "aws\_instance" "instance2vpc01" {

  ami = "ami-080e1f13689e07408"

  instance\_type = "t2.micro"

  subnet\_id = aws\_subnet.pubsub1.id

  key\_name = aws\_key\_pair.kp.key\_name

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

  tags = {

    Name = "Bastion Server"

  }

}

## Create a launch template to be used in the auoscaling group

resource "aws\_launch\_template" "launchtemp" {

  name = "lt1"

  instance\_type = "t2.micro"

  key\_name = aws\_key\_pair.kp.key\_name

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

  image\_id = "ami-080e1f13689e07408"

  user\_data = base64encode(file("userdata.sh"))

}

## Create target groups for load balancer

resource "aws\_lb\_target\_group" "tg" {

  name = "TG1"

  protocol = "HTTP"

  port = 80

  vpc\_id = aws\_vpc.myvpc.id

  health\_check {

    path = "/"

    port = "traffic-port"

  }

}

## Create autoscaling group and attach the target group

resource "aws\_autoscaling\_group" "asg" {

  name = "ASG"

  vpc\_zone\_identifier = [aws\_subnet.prvsub2.id,aws\_subnet.prvsub1.id]

  min\_size = 2

  max\_size = 4

  desired\_capacity = 2

  health\_check\_grace\_period = 300

  launch\_template {

    id = aws\_launch\_template.launchtemp.id

    version = "$Default"

  }

  target\_group\_arns = [aws\_lb\_target\_group.tg.arn]

}

## Create load balancer

resource "aws\_lb" "lb" {

  name = "Load-balancer1"

  internal = false

  ip\_address\_type = "ipv4"

  load\_balancer\_type = "application"

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

  subnets = [aws\_subnet.pubsub1.id,aws\_subnet.pubsub2.id]

}

## Create a listener to route the traffic to the target groups

resource "aws\_lb\_listener" "listen" {

  load\_balancer\_arn = aws\_lb.lb.arn

  protocol = "HTTP"

  port = 80

  default\_action {

    type = "forward"

    target\_group\_arn = aws\_lb\_target\_group.tg.arn

  }

}