Terraform 2 tier Application project

In this project the code written in HCL will do as follows as we apply the command “TERRAFORM APPLY” then terraform configuration sets up a VPC with a large IP address range (10.0.0.0/16) and two subnets in different availability zones (ap-southeast-1a and ap-southeast-1b) within that VPC. The VPC is linked to an internet gateway, which is used to route traffic from the internet. A route table is created to handle routing for outbound traffic and is associated with one of the subnets, ensuring that instances in that subnet can access the internet.

Two security groups are defined: one allows HTTP and SSH access from anywhere on the internet, and another manages inbound and outbound traffic. There are also two EC2 instances set up in different subnets, both configured to use the security group that permits HTTP and SSH traffic.

An application load balancer (ALB) is created and associated with the two subnets to distribute traffic across the instances. A target group is set up to route traffic to these instances on port 80, though there is a mistake here as there’s a duplication in the aws_instance resource names which needs to be resolved. Finally, the configuration specifies the AWS provider and its region for the resources.

main.tf

main.tf > resource "aws_vpc" "main"

```
resource "aws_vpc" "main" {
  cidr_block       = "10.0.0.0/16"
  instance_tenancy = "default"

  tags = {
    Name = "main"
  }
}
resource "aws_subnet" "main1" {
  vpc_id     = aws_vpc.main.id
  cidr_block = "10.0.1.0/24"
  availability_zone = "ap-southeast-1a"

  tags = {
    Name = "Main"
  }
}
resource "aws_subnet" "main2" {
  vpc_id     = aws_vpc.main.id
  cidr_block = "10.0.2.0/24"
  availability_zone = "ap-southeast-1b"

  tags = {
    Name = "Main"
  }
}
resource "aws_internet_gateway" "gw" {
  vpc_id = aws_vpc.main.id

  tags = {
    Name = "main"
  }
}
resource "aws_route_table" "public_rt" {
  vpc_id = aws_vpc.main.id

```

main.tf > resource "aws_vpc" "main"

```
resource "aws_route_table" "public_rt" {

  route {
    cidr_block = "0.0.0.0/0"
    gateway_id = aws_internet_gateway.gw.id
  }

  route {
    ipv6_cidr_block = "::/0"
    gateway_id      = aws_internet_gateway.gw.id
  }

  tags = {
    Name = "Public Route Table"
  }
}

resource "aws_route_table_association" "public_1_rt_a" {
  subnet_id      = aws_subnet.main1.id
  route_table_id = aws_route_table.public_rt.id
}




resource "aws_security_group" "web_sg" {
  name   = "HTTP and SSH"
  vpc_id = aws_vpc.main.id

  ingress {
    from_port   = 80
    to_port     = 80
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }

  ingress {
```

main.tf

main.tf > resource "aws_vpc" "main"

```
60  resource "aws_security_group" "web_sg" {
71    ingress {
72      from_port   = 22
73      to_port     = 22
74      protocol    = "tcp"
75      cidr_blocks = ["0.0.0.0/0"]
76    }
77
78    egress {
79      from_port   = 0
80      to_port     = 0
81      protocol    = -1
82      cidr_blocks = ["0.0.0.0/0"]
83    }
84  }
85
86  resource "aws_instance" "web_instance" {
87    ami           = "ami-0e97ea97a2f374e3d"
88    instance_type = "t2.micro"
89    key_name      = "ubuntu1"
90
91    subnet_id                   = aws_subnet.main1.id
92    vpc_security_group_ids      = [aws_security_group.web_sg.id]
93    associate_public_ip_address = true
94  }
95
96  resource "aws_instance" "web_instance" {
97    ami           = "ami-060e277c0d4cce553"
98    instance_type = "t2.micro"
99    key_name      = "ubuntu1"
100
101   subnet_id                   = aws_subnet.main2.id
102   vpc_security_group_ids      = [aws_security_group.web_sg.id]
103   associate_public_ip_address = true
104 }
105
106 resource "aws_lb" "my_alb" {
```

main.tf

main.tf > resource "aws_vpc" "main"

```
86  resource "aws_instance" "web_instance" {
91    subnet_id                   = aws_subnet.main1.id
92    vpc_security_group_ids      = [aws_security_group.web_sg.id]
93    associate_public_ip_address = true
94  }
95
96  resource "aws_instance" "web_instance" {
97    ami           = "ami-060e277c0d4cce553"
98    instance_type = "t2.micro"
99    key_name      = "ubuntu1"
100
101   subnet_id                   = aws_subnet.main2.id
102   vpc_security_group_ids      = [aws_security_group.web_sg.id]
103   associate_public_ip_address = true
104 }
105
106 resource "aws_lb" "my_alb" {
107  name               = "my-alb"
108  internal           = false
109  load_balancer_type = "application"
110  security_groups    = [aws_security_group.web_sg]
111  subnets            = [aws_subnet.main1,aws_subnet.main2]
112
113  tags = {
114    Environment = "dev"
115  }
116 }
117
118 resource "aws_lb_target_group" "test" {
119   name     = "tf-example-lb-tg"
120   port     = 80
121   protocol = "HTTP"
122   vpc_id   = aws_vpc.main.id
123 }
124
125 resource "aws_lb_target_group_attachment" "test" {
126   target_group_arn = aws_lb_target_group.test.arn
```

main.tf

main.tf > terraform > required_providers

```
116  }
117
118  resource "aws_lb_target_group" "test" {
119    name     = "tf-example-lb-tg"
120    port     = 80
121    protocol = "HTTP"
122    vpc_id   = aws_vpc.main.id
123  }
124
125  resource "aws_lb_target_group_attachment" "test" {
126    target_group_arn = aws_lb_target_group.test.arn
127    target_id        = aws_instance.web_instance1.id
128    port             = 80
129  }
130
131
132  terraform {
133    required_providers {
134      aws = {
135        source = "hashicorp/aws"
136        version = "5.58.0"
137      }
138    }
139  }
140
141  provider "aws" {
142    profile = "default"
143    region  = "us-east-1"
144  }
```

Instances (4) Info

Connect | Instance state | Actions | Launch instances

Find Instance by attribute or tag (case-sensitive) | All states | 1

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
	i-0b55b0534b9fc7e1a	Running	t2.micro	2/2 checks passed	View alarms	ap-southeast-1a	–
terraform	i-08225cb02803b93e6	Running	t2.micro	2/2 checks passed	View alarms	ap-southeast-1b	ec2-3-1-204-20
	i-049546b62b95da355	Running	t2.micro	2/2 checks passed	View alarms	ap-southeast-1b	–
	i-0ba92f7497df15482	Terminated	t2.micro	–	View alarms	ap-southeast-1b	–

Select an instance

Your VPCs (1/2) Info

Last updated 10 minutes ago | Actions | Create VP

Search

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DH
–	vpc-02645952f977a24e8	Available	172.31.0.0/16	–	do
main	vpc-0b830e1b9ef9edeca	Available	10.0.0.0/16	–	do

vpc-0b830e1b9ef9edeca / main

Subnets (5) Info

Last updated 10 minutes ago | Actions | Create subnet

Find resources by attribute or tag

Name	Subnet ID	State	VPC	IPv4 CIDR	IPv
–	subnet-0583e666e8ce1572b	Available	vpc-02645952f977a24e8	172.31.32.0/20	–
Main	subnet-06f625aa55374186d	Available	vpc-0b830e1b9ef9edeca | main	10.0.2.0/24	–
–	subnet-080281b720a748d9f	Available	vpc-02645952f977a24e8	172.31.16.0/20	–
Main	subnet-0f521ee7690e4a2c2	Available	vpc-0b830e1b9ef9edeca | main	10.0.1.0/24	–
–	subnet-01fde5b363bfe16f3	Available	vpc-02645952f977a24e8	172.31.0.0/20	–

Route tables (3) Info

Last updated 11 minutes ago | Actions | Create route table

Find resources by attribute or tag

Name	Route table ID	Explicit subnet associ...	Edge associations	Main	VPC
–	rtb-0c7384f77c3441cce	–	–	Yes	vpc-02645952f977a24e8
–	rtb-0a4b93663426190d2	–	–	Yes	vpc-0b830e1b9ef9edeca | m
Public Route Table	rtb-0953b9911f75a7d3a	subnet-0f521ee7690e4a...	–	No	vpc-0b830e1b9ef9edeca | m

Internet gateways (2) Info

Actions | Create internet gateway

Search

Name	Internet gateway ID	State	VPC ID	Owner
–	igw-05a10d9f333703d4c	Attached	vpc-02645952f977a24e8	211125322230
main	igw-066e9a6199c4839cf	Attached	vpc-0b830e1b9ef9edeca | main	211125322230

```
Terraform has compared your real infrastructure against your configuration and found no differences, so no changes are needed.
[root@ip-172-31-30-83 Terraform]# terraform apply
aws_vpc.main: Refreshing state... [id=vpc-0b830e1b9ef9edeca]
aws_lb_target_group.test: Refreshing state... [id=arn:aws:elasticloadbalancing:ap-southeast-1:211125322230:targetgroup/tf-example-lb-tg/30506820b210ef90]
aws_internet_gateway.gw: Refreshing state... [id=igw-066e9a6199c4839cf]
aws_security_group.web_sg: Refreshing state... [id=sg-07b16034872e2c867]
aws_subnet.main2: Refreshing state... [id=subnet-06f625aa55374186d]
aws_subnet.main1: Refreshing state... [id=subnet-0f521ee7690e4a2c2]
aws_route_table.public_rt: Refreshing state... [id=rtb-0953b9911f75a7d3a]
aws_instance.web_instance1: Refreshing state... [id=i-0b55b0534b9fc7e1a]
aws_instance.web_instance2: Refreshing state... [id=i-049546b62b95da355]
aws_lb.my_alb: Refreshing state... [id=arn:aws:elasticloadbalancing:ap-southeast-1:211125322230:loadbalancer/app/my-alb/99e8d0d1985d8244]
aws_route_table_association.public_1_rt_a: Refreshing state... [id=rtbassoc-053e087e7447b3ed6]
aws_lb_target_group_attachment.test: Refreshing state... [id=arn:aws:elasticloadbalancing:ap-southeast-1:211125322230:targetgroup/tf-example-lb-tg/30506820b210ef90-20240718063238918600000002]

No changes. Your infrastructure matches the configuration.

Terraform has compared your real infrastructure against your configuration and found no differences, so no changes are needed.

Apply complete! Resources: 0 added, 0 changed, 0 destroyed.
```