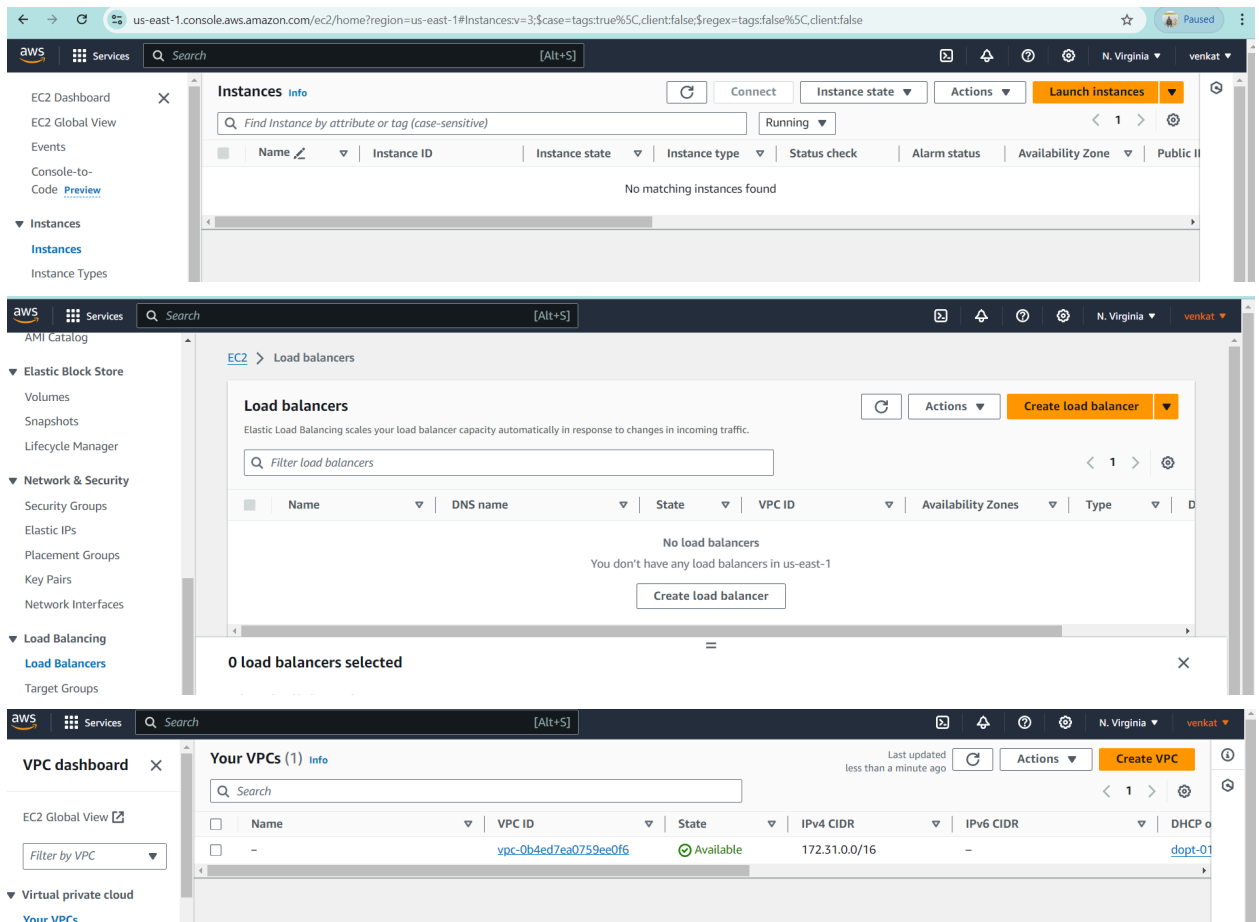


# CREATE LOAD BALANCER USING TERRAFORM (us-east-1)

## 1) Login aws account

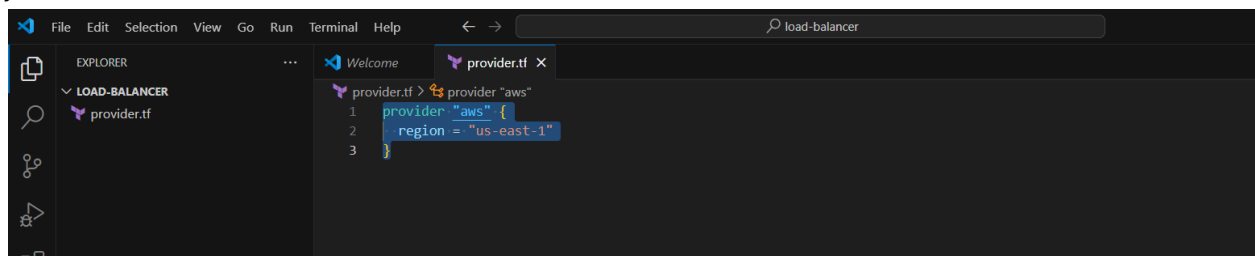


## 2) Now open visual studio code

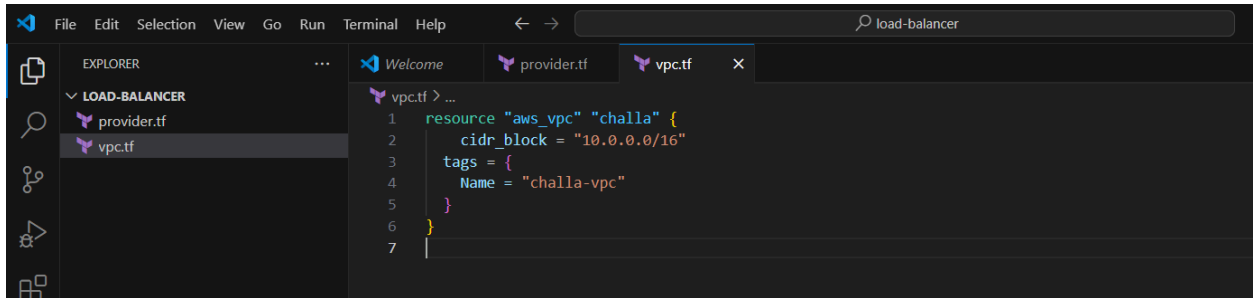
## 3) Select folder

## 4) Create file provider.tf

```
provider "aws" {  
  region = "us-east-1"  
}
```



## 5) Now create vpc.tf



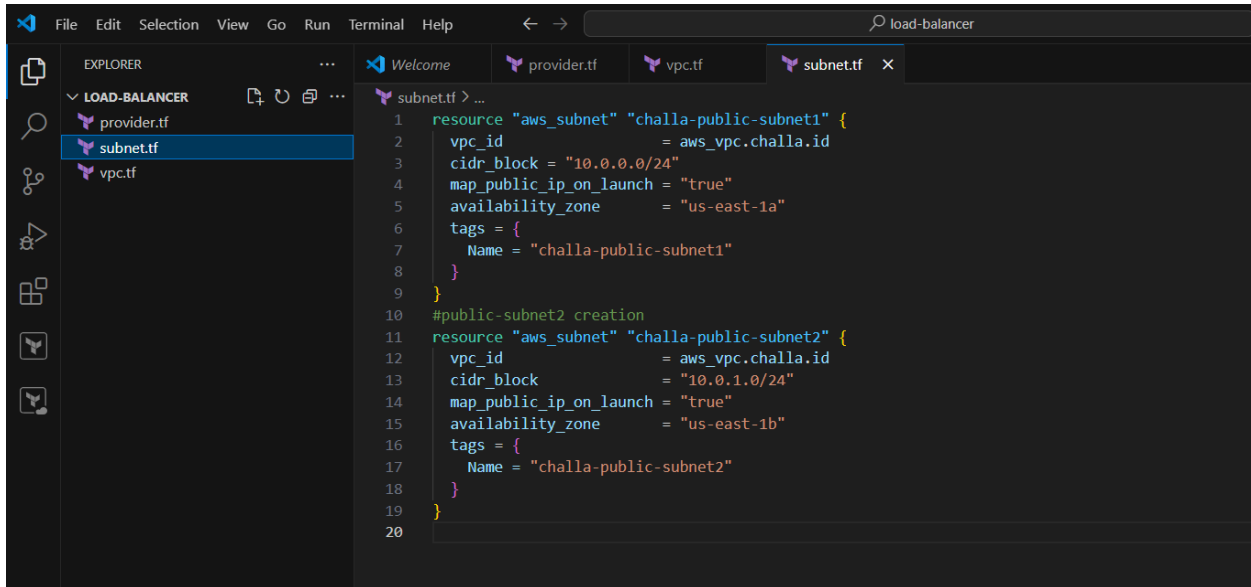
```
resource "aws_vpc" "challa" {  
  cidr_block = "10.0.0.0/16"  
  tags = {  
    Name = "challa-vpc"  
  }  
}
```

6) Now create subnet.tf

```
resource "aws_subnet" "challa-public-subnet1" {  
  vpc_id      = aws_vpc.challa.id  
  cidr_block  = "10.0.0.0/24"  
  map_public_ip_on_launch = "true"  
  availability_zone = "us-east-1a"  
  tags = {  
    Name = "challa-public-subnet1"  
  }  
}
```

#public-subnet2 creation

```
resource "aws_subnet" "challa-public-subnet2" {  
  vpc_id      = aws_vpc.challa.id  
  cidr_block  = "10.0.1.0/24"  
  map_public_ip_on_launch = "true"  
  availability_zone = "us-east-1b"  
  tags = {  
    Name = "challa-public-subnet2"  
  }  
}
```

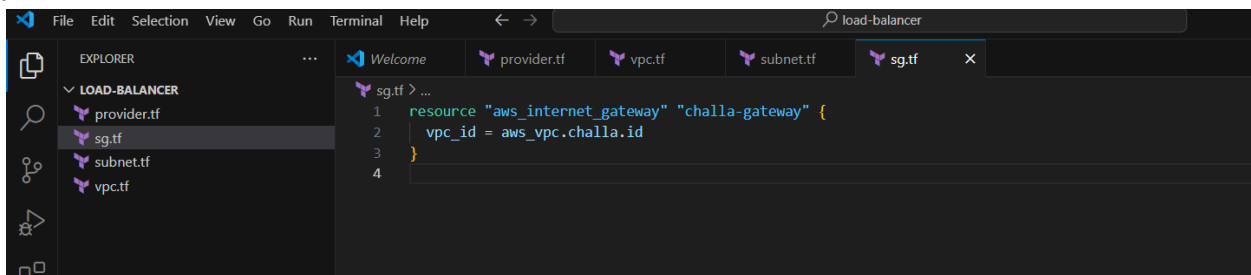


7) Now create igw.tf

```

resource "aws_internet_gateway" "challa-gateway" {
  vpc_id = aws_vpc.challa.id
}

```



8) Now create route table add route and subnet association

9) route.tf

```

resource "aws_route_table" "challa-route" {
  vpc_id = aws_vpc.challa.id

  route {
    cidr_block = "0.0.0.0/0"
    gateway_id = aws_internet_gateway.challa-gateway.id
  }
  tags = {
    Name = "route to internet"
  }
}

#route 1
resource "aws_route_table_association" "challa-route1" {
  subnet_id = aws_subnet.challa-public-subnet1.id
  route_table_id = aws_route_table.challa-route.id
}

```

```

}
#route 2
resource "aws_route_table_association" "challa-route2" {
  subnet_id      = aws_subnet.challa-public-subnet2.id
  route_table_id = aws_route_table.challa-route.id
}

```



10) Now create security group sg.tf

```

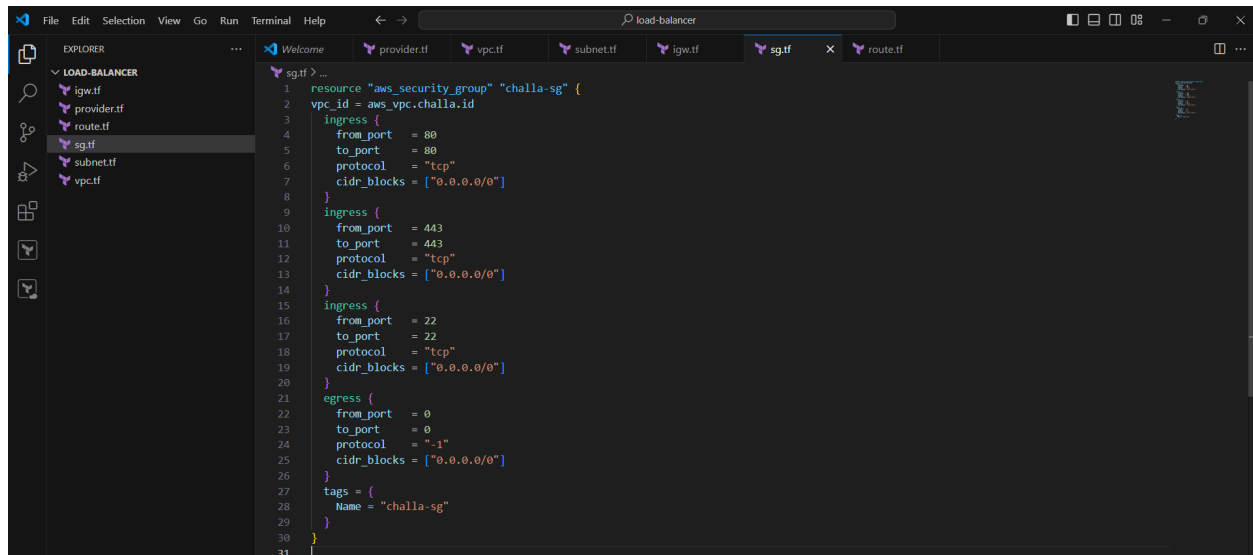
resource "aws_security_group" "challa-sg" {
  vpc_id = aws_vpc.challa.id
  ingress {
    from_port = 80
    to_port   = 80
    protocol  = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }
  ingress {
    from_port = 443
    to_port   = 443
    protocol  = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }
  ingress {
    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 = "challa-sg"
  }
}

```



11) Now create 2 ec2 instance ec2.tf

```

resource "aws_instance" "challa" {
  ami           = "ami-0e86e20dae9224db8" #use your ubuntu ami because user data write
  instance_type = "t2.micro"
  key_name      = "challa"
  vpc_security_group_ids = [aws_security_group.challa-sg.id]
  subnet_id     = aws_subnet.challa-public-subnet1.id
  associate_public_ip_address = true
  user_data     = file("${path.module}/appache1.sh")
  tags = {
    Name = "challa"
  }
}

```

```

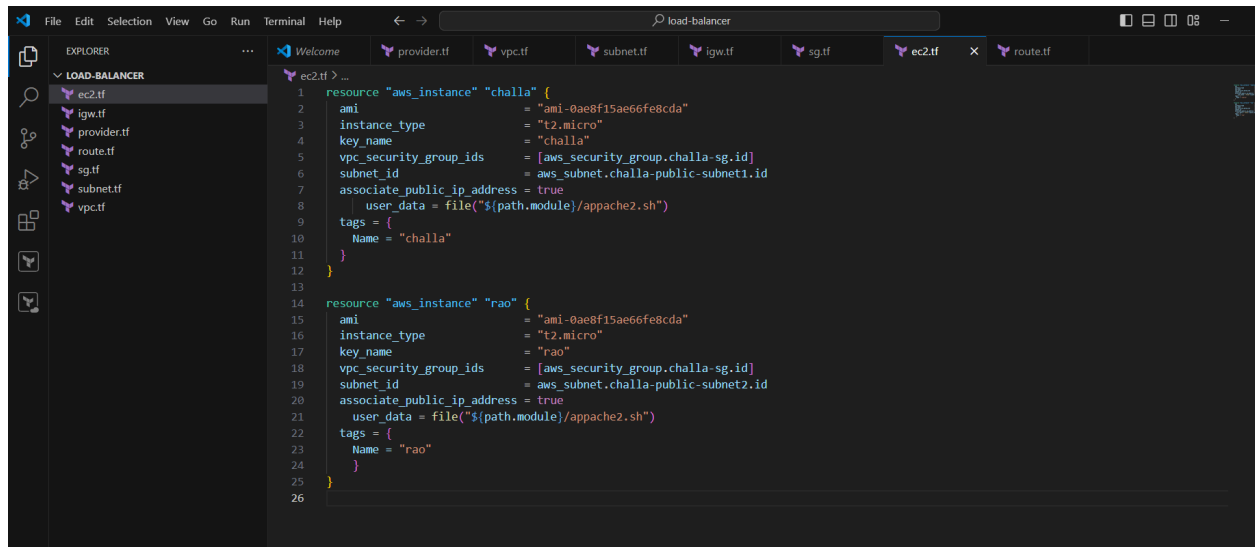
resource "aws_instance" "rao" {
  ami           = "ami-0e86e20dae9224db8" #use your ubuntu ami because user data write
  instance_type = "t2.micro"
  key_name      = "rao"
  vpc_security_group_ids = [aws_security_group.challa-sg.id]
  subnet_id     = aws_subnet.challa-public-subnet2.id
  associate_public_ip_address = true
}

```

```

    user_data = file("${path.module}/apache2.sh")
  tags = {
    Name = "rao"
  }
}

```

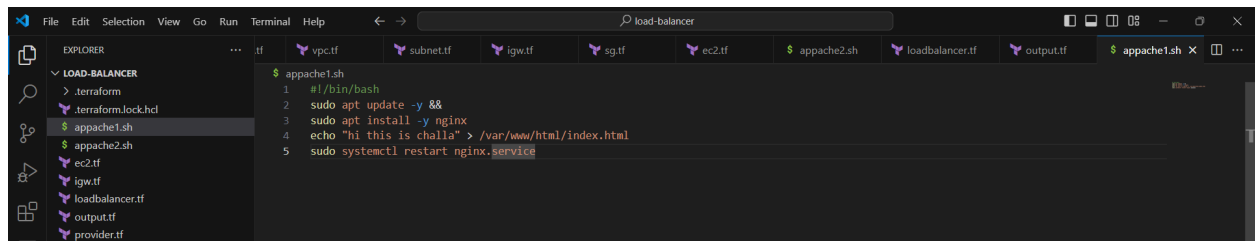


12) Now create apache1.sh to install nginx for challa instance

```

#apache1.sh
#!/bin/bash
sudo apt update -y &&
sudo apt install -y nginx
echo "hi this is challa" > /var/www/html/index.html
sudo systemctl restart nginx.service

```



13) Now create apache2.sh to install nginx for rao instance

```

#apache2.sh
#!/bin/bash
sudo apt update -y &&
sudo apt install -y nginx
echo "hi this is rao" > /var/www/html/index.html
sudo systemctl restart nginx.service

```

```
... .tf vpc.tf subnet.tf igw.tf sg.tf ec2.tf $ apache2.sh
$ apache2.sh
1  #!/bin/bash
2  sudo apt update -y &&
3  sudo apt install -y nginx
4  echo "hi this is rao" > /var/www/html/index.html
5  sudo systemctl restart nginx.service
6
```

14) Now create load balancer and target group

```
resource "aws_lb" "challa-alb" {
  name          = "challa-LB"
  internal      = false
  load_balancer_type = "application"
  security_groups = [aws_security_group.challa-sg.id]
  subnets      = [aws_subnet.challa-public-subnet1.id, aws_subnet.challa-public-subnet2.id]
}

resource "aws_lb_target_group" "challa-tg" {
  name     = "challa-TG"
  port     = 80
  protocol = "HTTP"
  vpc_id   = aws_vpc.challa.id
  health_check {
    path     = "/health"
    port     = 80
    protocol = "HTTP"
  }
}

resource "aws_lb_target_group_attachment" "challas" {
  target_group_arn = aws_lb_target_group.challa-tg.arn
  target_id        = aws_instance.challa.id
  port             = 80
  depends_on = [
    aws_lb_target_group.challa-tg,
    aws_instance.challa,
  ]
}

resource "aws_lb_target_group_attachment" "raos" {
  target_group_arn = aws_lb_target_group.challa-tg.arn
  target_id        = aws_instance.rao.id
  port             = 80
  depends_on = [
    aws_lb_target_group.challa-tg,
    aws_instance.rao,
  ]
}
```

```

}
resource "aws_lb_listener" "listener_elb" {
  load_balancer_arn = aws_lb.challa-alb.arn
  port              = 80
  protocol          = "HTTP"
  default_action {
    type            = "forward"
    target_group_arn = aws_lb_target_group.challa-tg.arn
  }
}
}

```

```

1 resource "aws_lb" "challa-alb" {
2   name           = "challa-lb"
3   internal       = false
4   load_balancer_type = "application"
5   security_groups = [aws_security_group.challa-sg.id]
6   subnets       = [aws_subnet.challa-public-subnet1.id, aws_subnet.challa-public-subnet2.id]
7 }
8 resource "aws_lb_target_group" "challa-tg" {
9   name           = "challa-tg"
10  port           = 80
11  protocol       = "HTTP"
12  vpc_id         = aws_vpc.challa.id
13  health_check {
14    path         = "/health"
15    port         = 80
16    protocol     = "HTTP"
17  }
18 }
19 resource "aws_lb_target_group_attachment" "challas" {
20   target_group_arn = aws_lb_target_group.challa-tg.arn
21   target_id        = aws_instance.challa.id
22   port             = 80
23   depends_on = [
24     aws_lb_target_group.challa-tg,
25     aws_instance.challa,
26   ]
27 }
28 resource "aws_lb_target_group_attachment" "raos" {
29   target_group_arn = aws_lb_target_group.challa-tg.arn
30   target_id        = aws_instance.rao.id
31   port             = 80
32   depends_on = [
33     aws_lb_target_group.challa-tg,
34     aws_instance.rao,
35   ]
36 }
37 resource "aws_lb_listener" "listener_elb" {

```

```

18 resource "aws_lb_target_group" "challa-tg" {
19 }
20 resource "aws_lb_target_group_attachment" "challas" {
21   target_group_arn = aws_lb_target_group.challa-tg.arn
22   target_id        = aws_instance.challa.id
23   port             = 80
24   depends_on = [
25     aws_lb_target_group.challa-tg,
26     aws_instance.challa,
27   ]
28 }
29 resource "aws_lb_target_group_attachment" "raos" {
30   target_group_arn = aws_lb_target_group.challa-tg.arn
31   target_id        = aws_instance.rao.id
32   port             = 80
33   depends_on = [
34     aws_lb_target_group.challa-tg,
35     aws_instance.rao,
36   ]
37 }
38 resource "aws_lb_listener" "listener_elb" {
39   load_balancer_arn = aws_lb.challa-alb.arn
40   port              = 80
41   protocol          = "HTTP"
42   default_action {
43     type            = "forward"
44     target_group_arn = aws_lb_target_group.challa-tg.arn
45   }
46 }

```

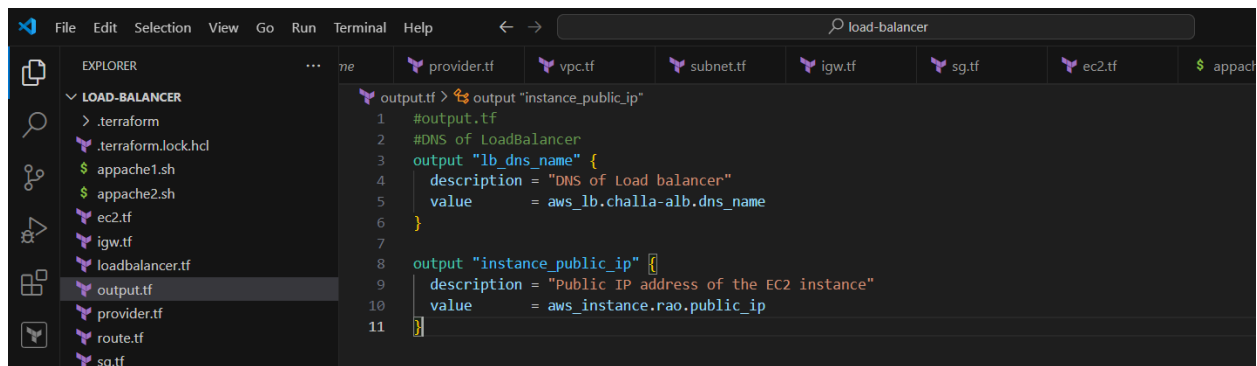
15) Now create output.tf  
#output.tf



#DNS of LoadBalancer

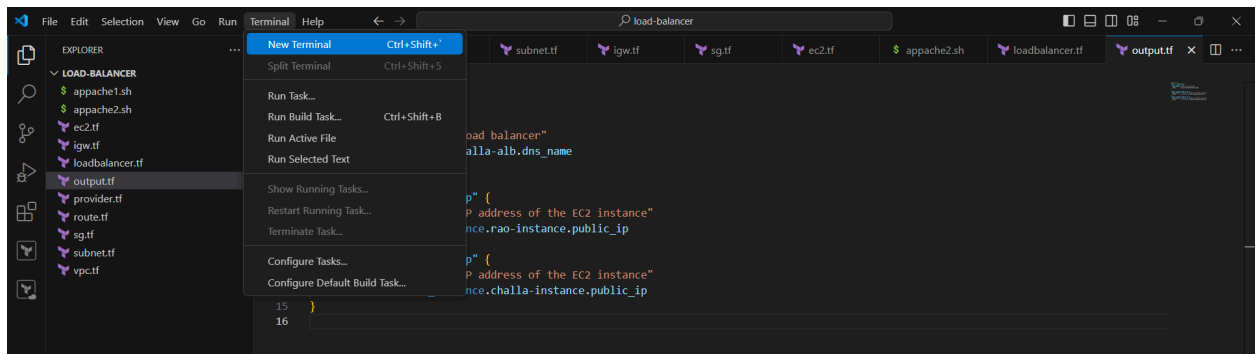
```
output "lb_dns_name" {  
  description = "DNS of Load balancer"  
  value      = aws_lb.challa-alb.dns_name  
}
```

```
output "instance_public_ip" {  
  description = "Public IP address of the EC2 instance"  
  value      = aws_instance.rao.public_ip  
}
```

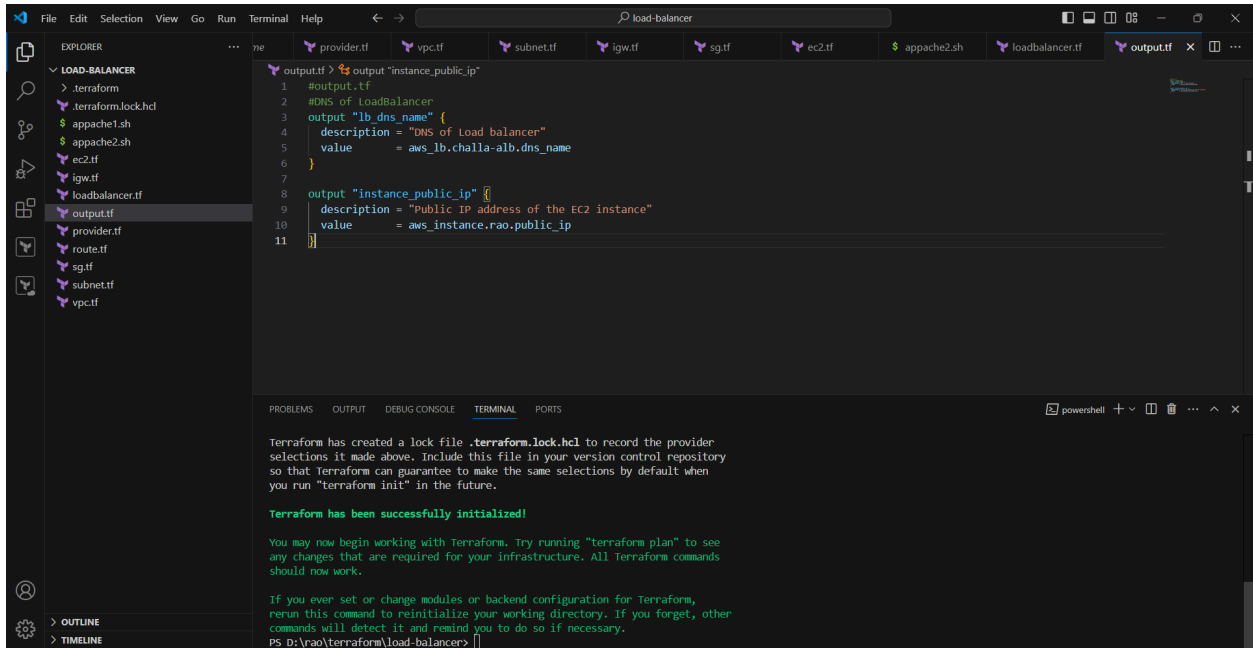


16) Now save all

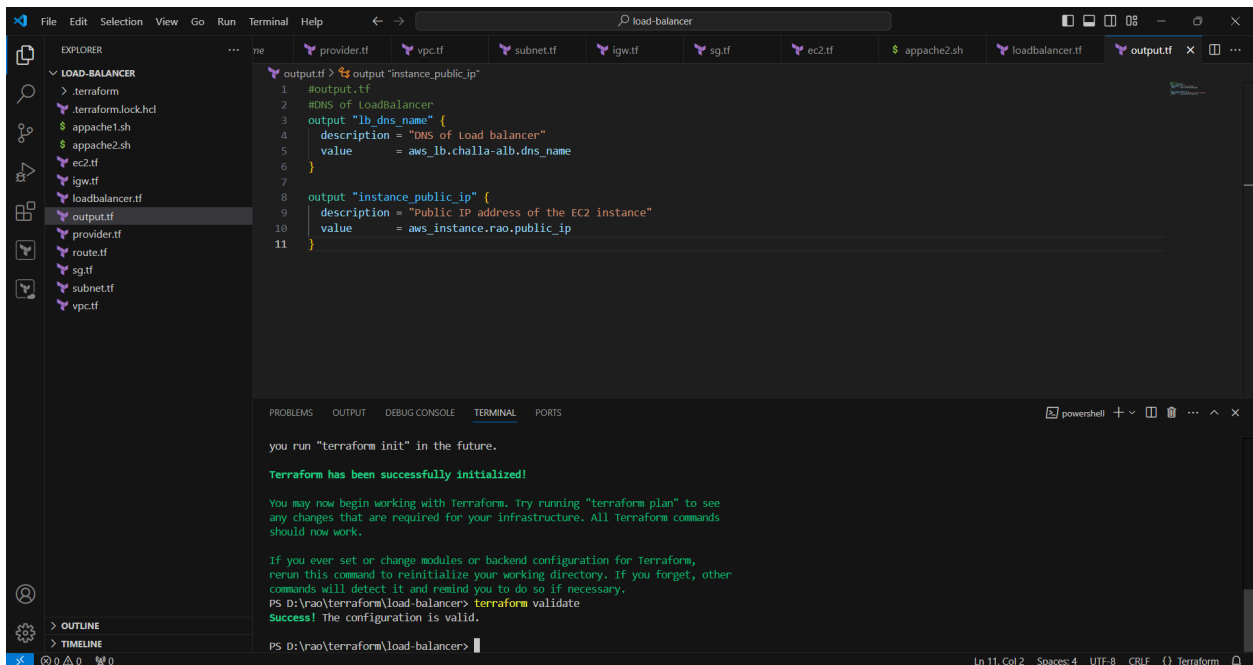
17) Now click on terminal select new terminal



18) Now click on terminal and use #terraform init



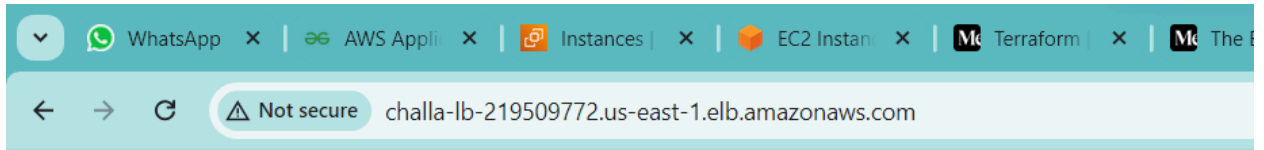
## 19) #terraform validate



## 20) Now terraform plan #terraform plan

## 21) Now terraform apply #terraform apply --auto-approve

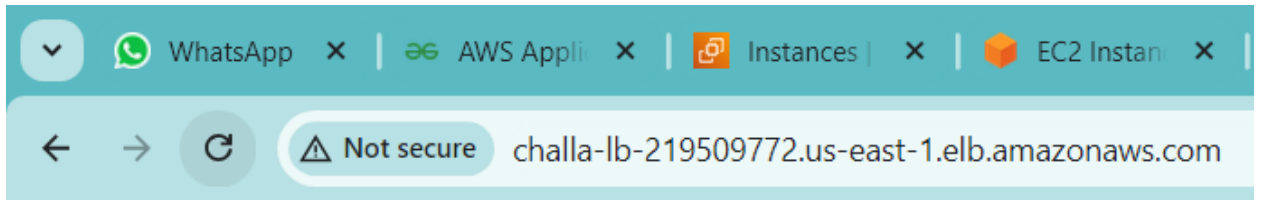
## 22) Copy dns and paste in google



this is challa

23)

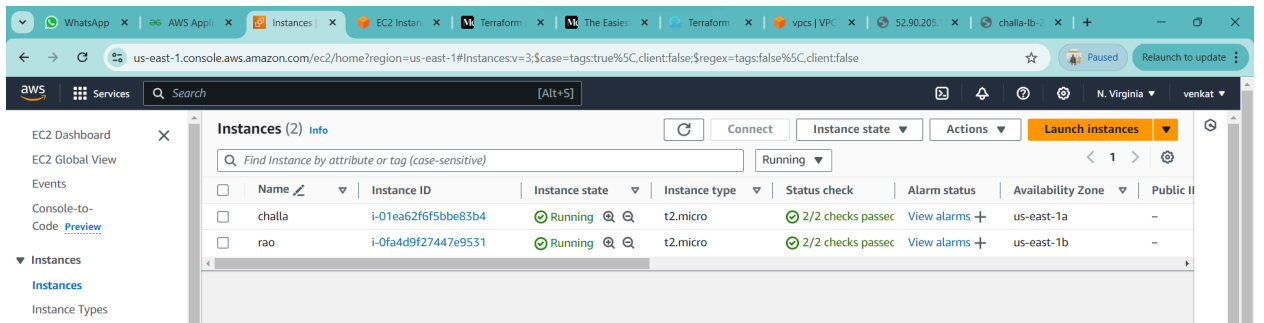
24) Refresh



hi this is rao

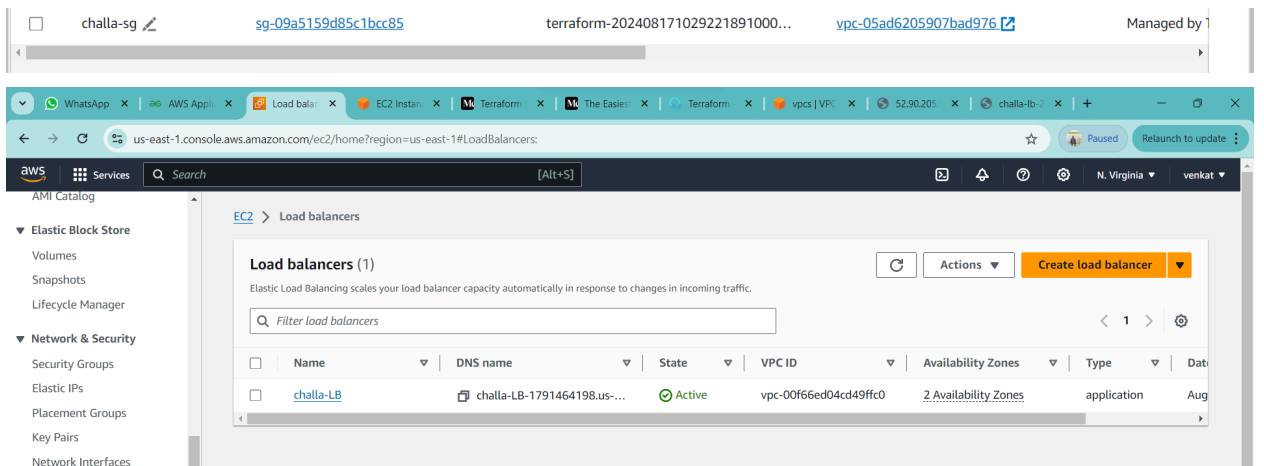
25)

26) Once go and check instances and load balancers



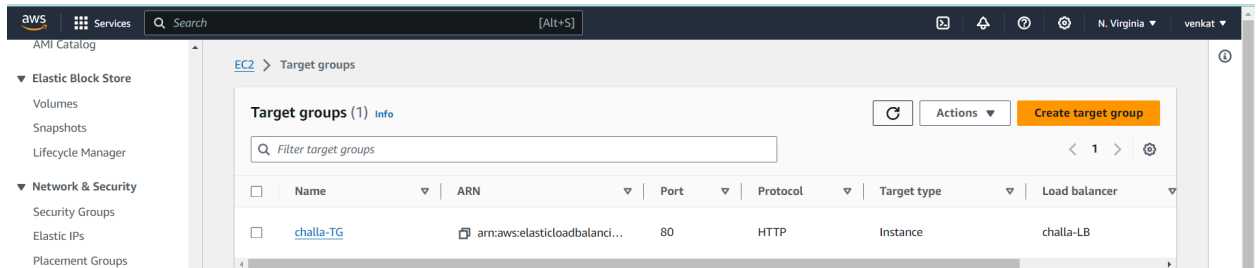
27)

28)

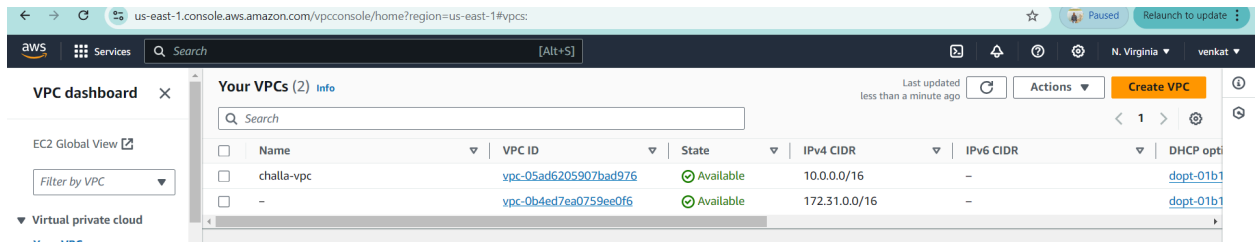


29)

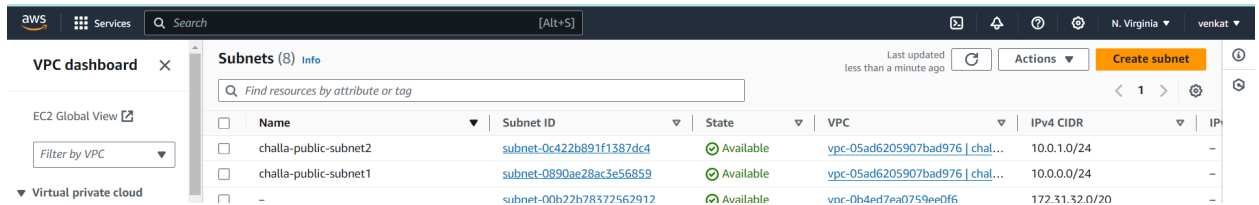
30)



31)



32)



=====

If you want destroy use `#terraform destroy`

```

aws_instance.rao: Still modifying... [id=i-0fa4d9f27447e9531, 30s elapsed]
aws_instance.challa: Still modifying... [id=i-01ea62f6f5bbe83b4, 40s elapsed]
aws_instance.rao: Still modifying... [id=i-0fa4d9f27447e9531, 40s elapsed]
aws_instance.rao: Still modifying... [id=i-0fa4d9f27447e9531, 50s elapsed]
aws_instance.challa: Still modifying... [id=i-01ea62f6f5bbe83b4, 50s elapsed]
aws_instance.challa: Still modifying... [id=i-01ea62f6f5bbe83b4, 1m0s elapsed]
aws_instance.rao: Still modifying... [id=i-0fa4d9f27447e9531, 1m0s elapsed]
aws_instance.rao: Still modifying... [id=i-0fa4d9f27447e9531, 1m10s elapsed]
aws_instance.challa: Still modifying... [id=i-01ea62f6f5bbe83b4, 1m10s elapsed]
aws_instance.rao: Modifications complete after 1m17s [id=i-0fa4d9f27447e9531]
aws_instance.challa: Modifications complete after 1m18s [id=i-01ea62f6f5bbe83b4]

Apply complete! Resources: 0 added, 2 changed, 0 destroyed.

Outputs:

instance_public_ip = "52.90.205.15"
lb_dns_name = "challa-LB-219509772.us-east-1.elb.amazonaws.com"
PS D:\rao\terraform\load-balancer> terraform destroy
aws_vpc.challa: Refreshing state... [id=vpc-05ad6205907bad976]

```

```

- owner_id = "058264331590" -> null
- tags = {
  - "Name" = "challa-vpc"
} -> null
- tags_all = {
  - "Name" = "challa-vpc"
} -> null
# (4 unchanged attributes hidden)
}

```

Plan: 0 to add, 0 to change, 15 to destroy.

Changes to Outputs:

```

- instance_public_ip = "54.91.167.113" -> null
- lb_dns_name = "challa-LB-219509772.us-east-1.elb.amazonaws.com" -> null

```

Do you really want to destroy all resources?

Terraform will destroy all your managed infrastructure, as shown above.  
There is no undo. Only 'yes' will be accepted to confirm.

Enter a value: yes

Ln 6,

The screenshot shows a VS Code editor with a file explorer on the left containing files: sg.tf, subnet.tf, terraform.tfstate, terraform.tfstate.backup, and vpc.tf. The main terminal window displays the output of a Terraform destroy command. The output shows the destruction of various AWS resources, including instances, subnets, internet gateways, and security groups. The terminal output ends with 'Destroy complete! Resources: 15 destroyed.' and a PowerShell prompt 'PS D:\rao\terraform\load-balancer>'.

```

aws_instance.rao: Still destroying... [id=i-0fa4d9f27447e9531, 30s elapsed]
aws_instance.challa: Still destroying... [id=i-01ea62f6f5bbe83b4, 30s elapsed]
aws_internet_gateway.challa-gateway: Still destroying... [id=igw-0fcde3869d4263edf, 30s elapsed]
aws_instance.rao: Still destroying... [id=i-0fa4d9f27447e9531, 40s elapsed]
aws_instance.challa: Still destroying... [id=i-01ea62f6f5bbe83b4, 40s elapsed]
aws_internet_gateway.challa-gateway: Still destroying... [id=igw-0fcde3869d4263edf, 40s elapsed]
aws_internet_gateway.challa-gateway: Destruction complete after 41s
aws_instance.rao: Destruction complete after 43s
aws_subnet.challa-public-subnet2: Destroying... [id=subnet-0c422b891f1387dc4]
aws_subnet.challa-public-subnet2: Destruction complete after 1s
aws_instance.challa: Still destroying... [id=i-01ea62f6f5bbe83b4, 50s elapsed]
aws_instance.challa: Destruction complete after 53s
aws_subnet.challa-public-subnet1: Destroying... [id=subnet-0890ae28ac3e56859]
aws_security_group.challa-sg: Destroying... [id=sg-09a5159d85c1bcc85]
aws_subnet.challa-public-subnet1: Destruction complete after 0s
aws_security_group.challa-sg: Destruction complete after 1s
aws_vpc.challa: Destroying... [id=vpc-05ad6205907bad976]
aws_vpc.challa: Destruction complete after 1s

Destroy complete! Resources: 15 destroyed.
PS D:\rao\terraform\load-balancer>

```