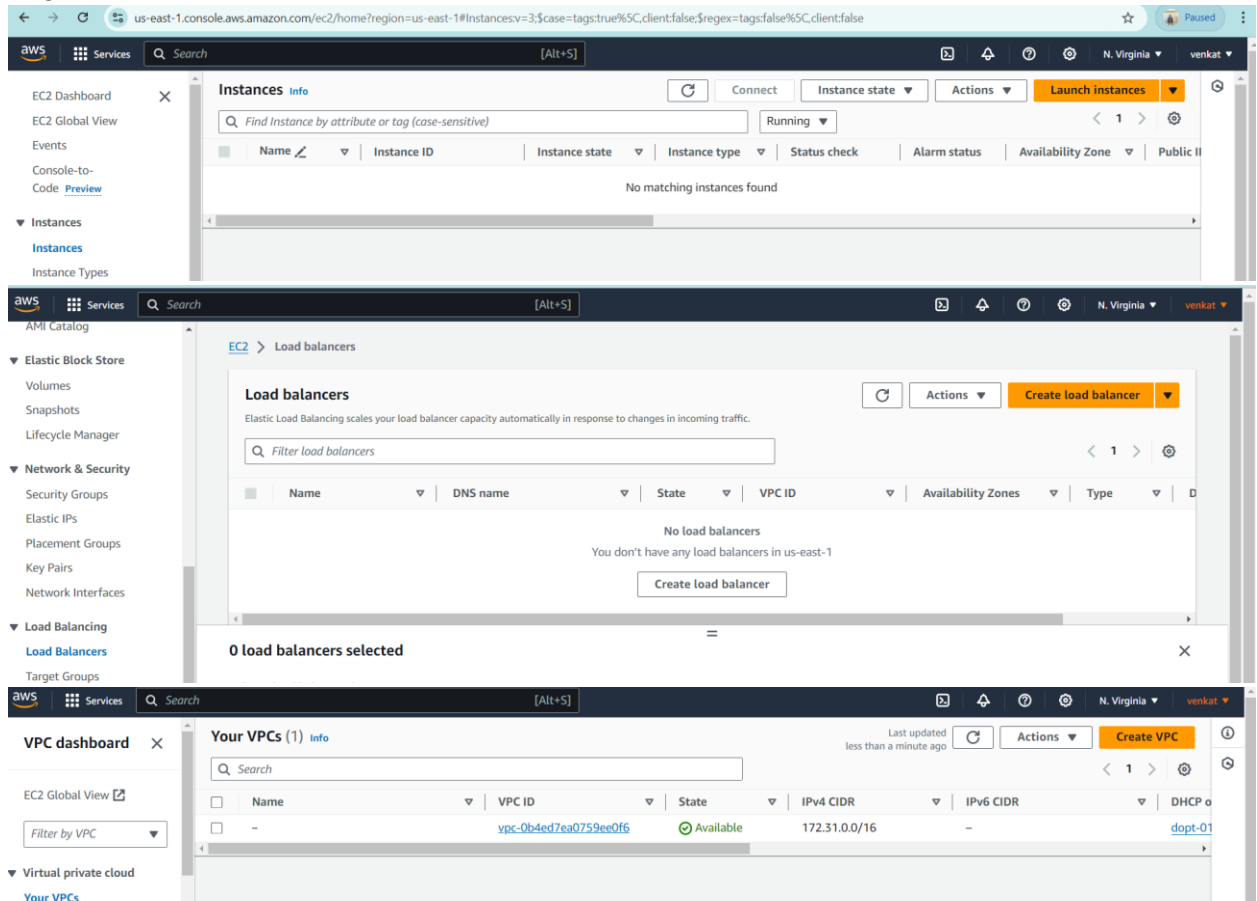


CREATE LOAD BALANCER USING TERRAFORM (us-east-1)

1) Login aws account

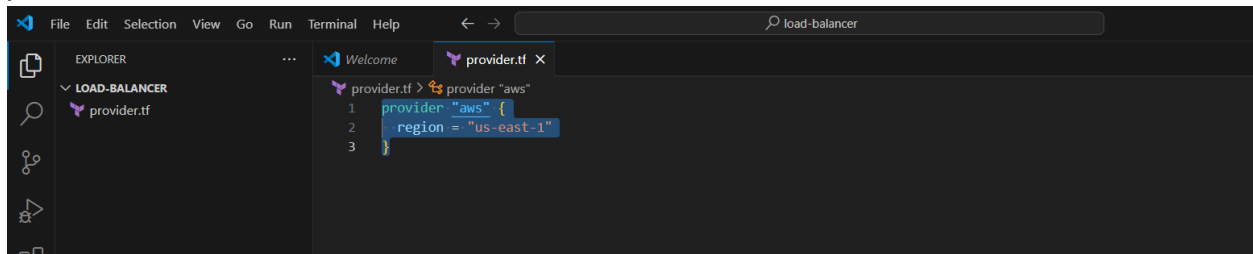


2) Now open visual studio code

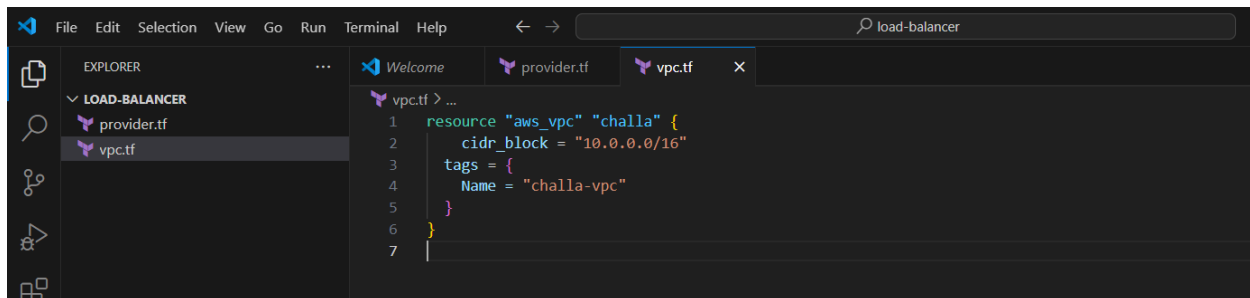
3) Select folder

4) Create file provider.tf

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



5) Now create vpc.tf



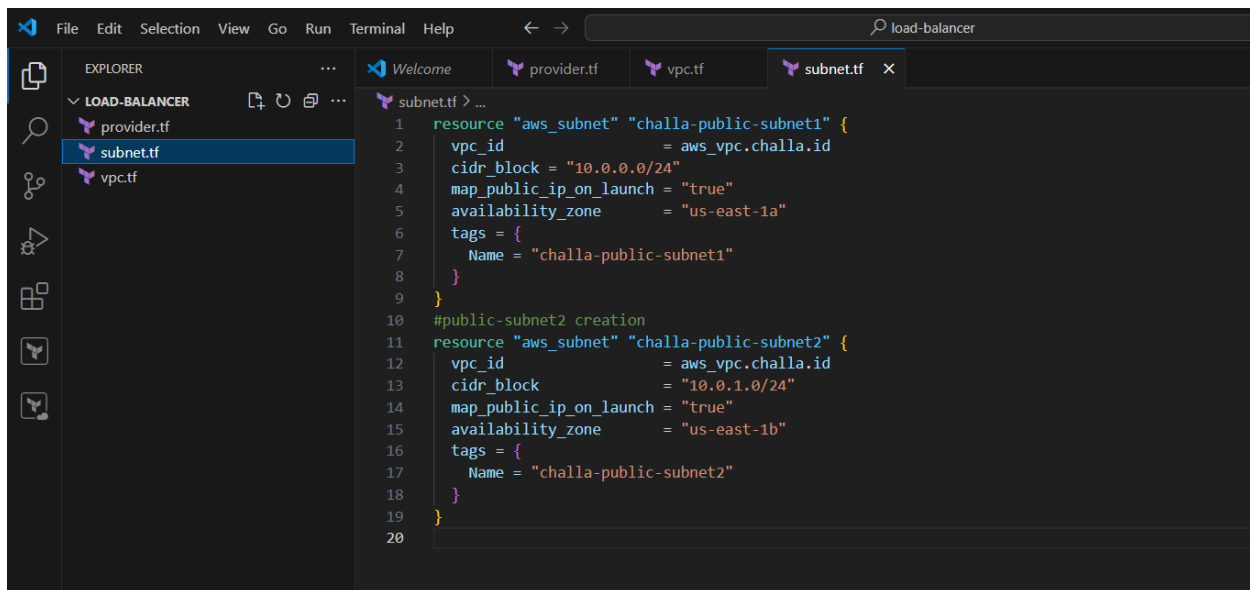
```
resource "aws_vpc" "challa" {  
  cidr_block = "10.0.0.0/16"  
  tags = {  
    Name = "challa-vpc"  
  }  
}
```

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"  
  }  
}
```

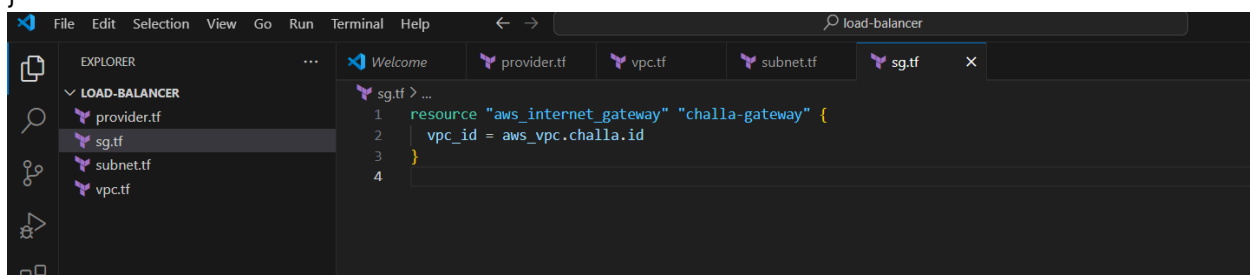


7) Now create igw.tf

```

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  
}
```



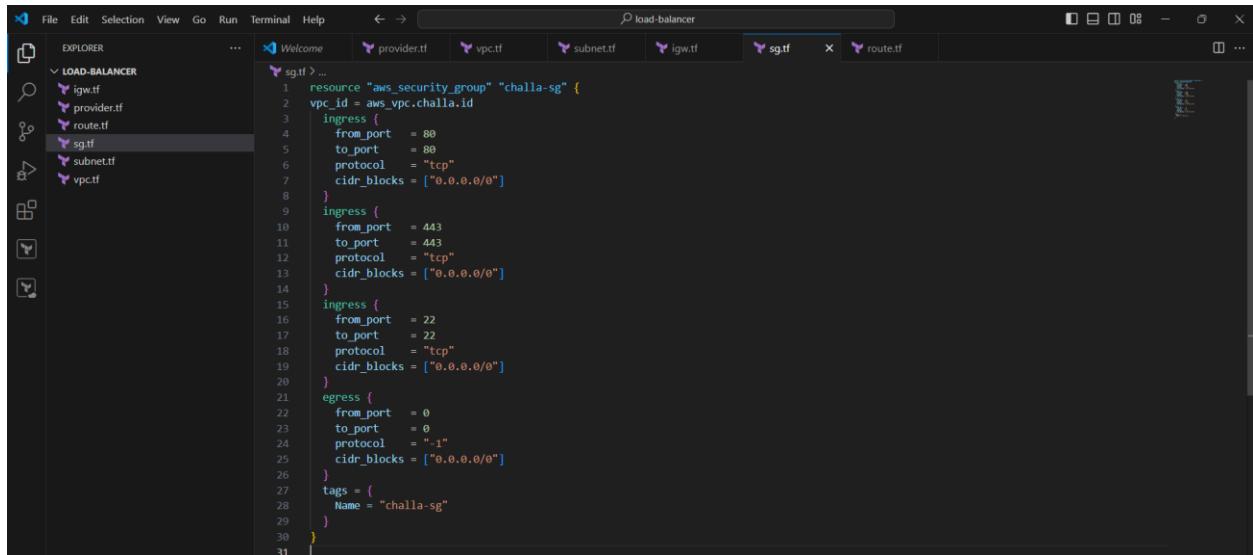
10) Now create security group sg.tf

```
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
  ubuntu script
  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}/apache1.sh")
  tags = {
    Name = "challa"
  }
}

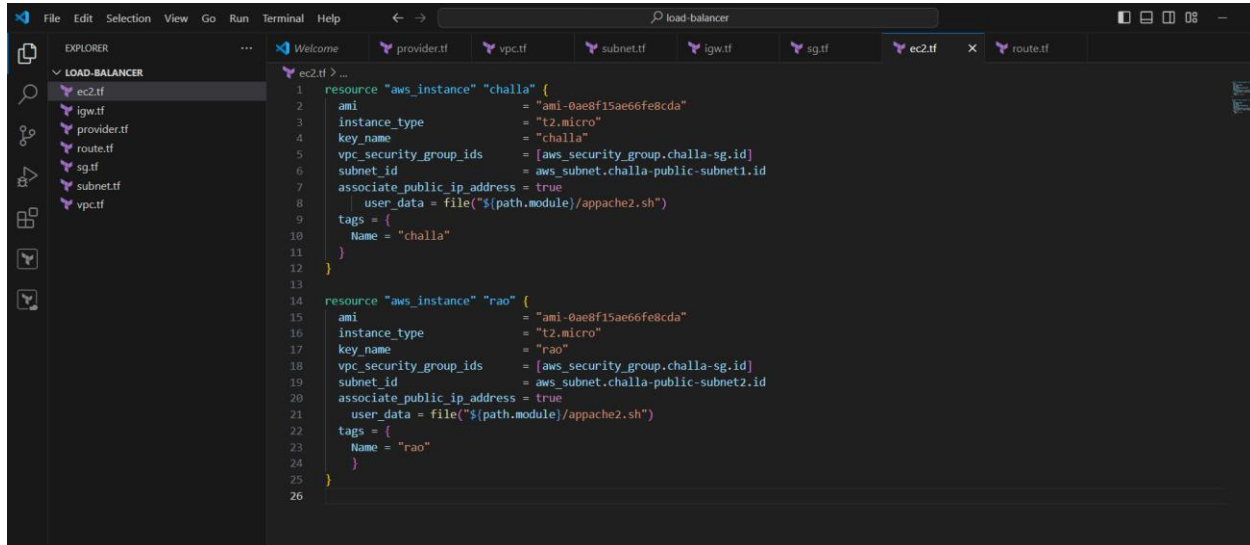
```

```

resource "aws_instance" "rao" {
  ami                = "ami-0e86e20dae9224db8" #use your ubuntu ami because user data write
  ubuntu script
  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")
}

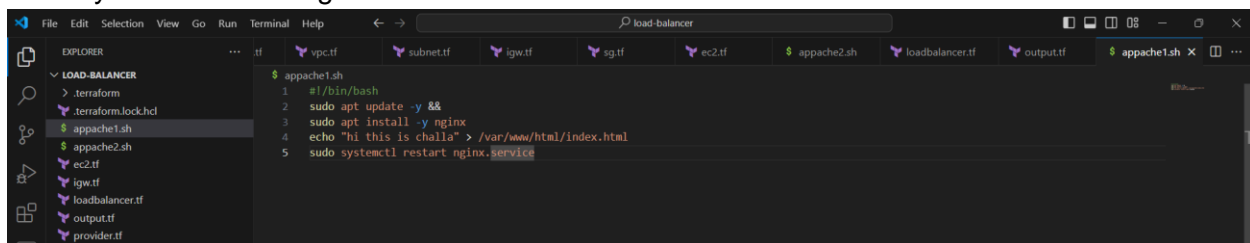
```

```
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.s
$ 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"
  }
}

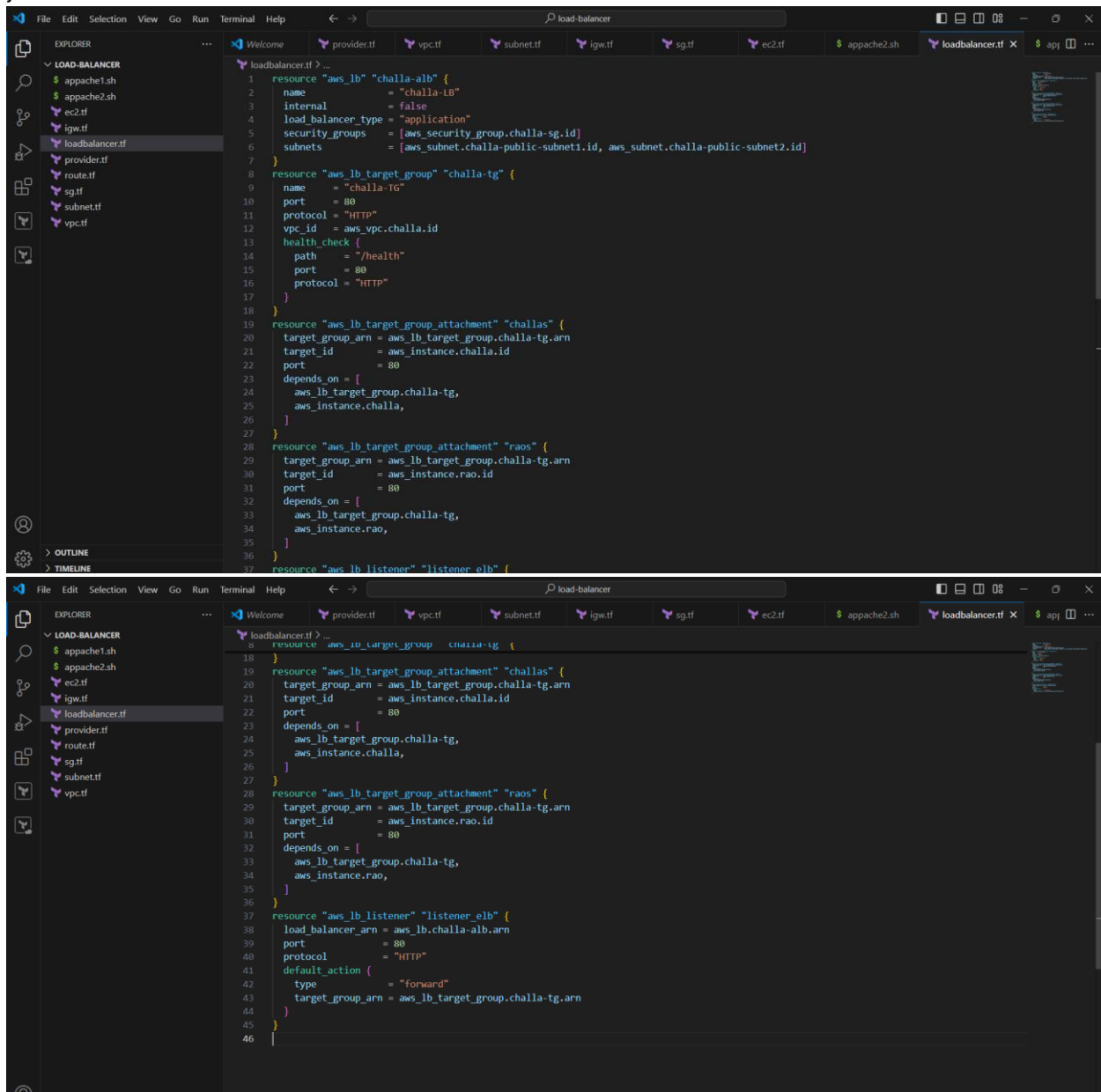
resource "aws_lb_target_group_attachment" "challas" {
  target_group_arn = aws_lb_target_group.challa-tg.arn
  target_id        = aws_instance.challa.id
  port             = 80
  depends_on = [
    aws_lb_target_group.challa-tg,
    aws_instance.challa,
  ]
}

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
  }
}
}

```



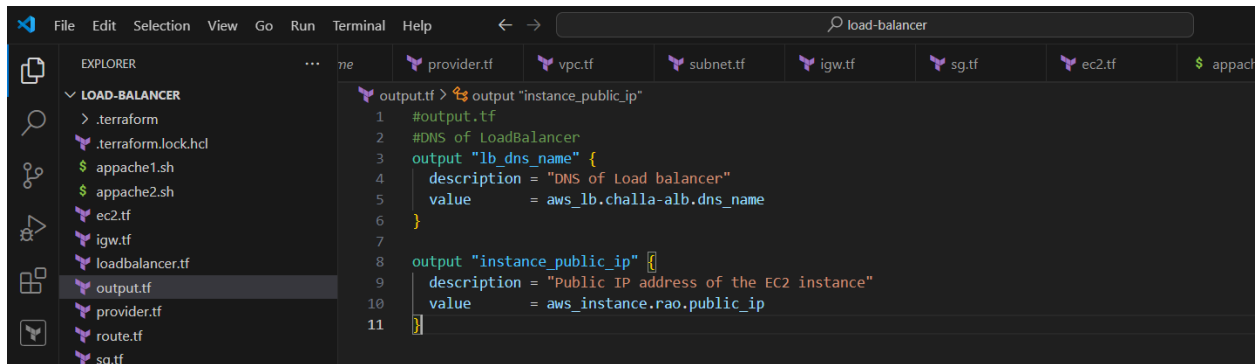
15) Now create output.tf

#output.tf

#DNS of LoadBalancer

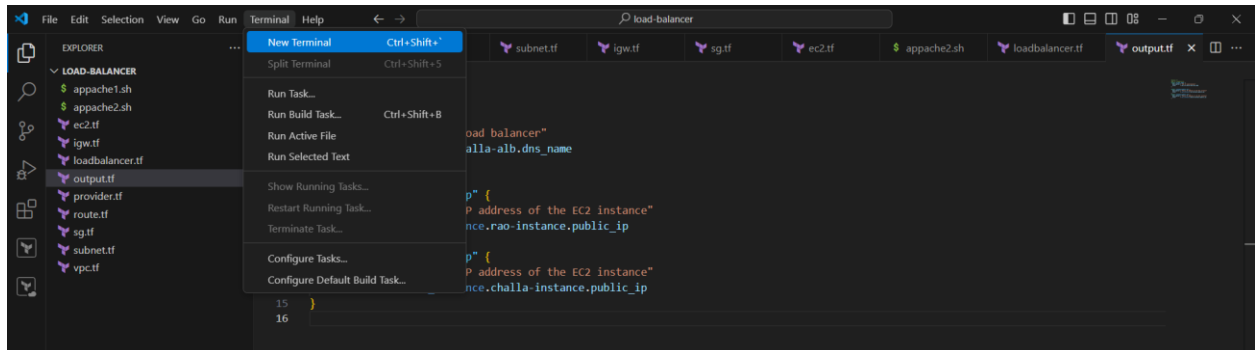
```
output "lb_dns_name" {  
  description = "DNS of Load balancer"  
  value      = aws_lb.challa-alb.dns_name  
}
```

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

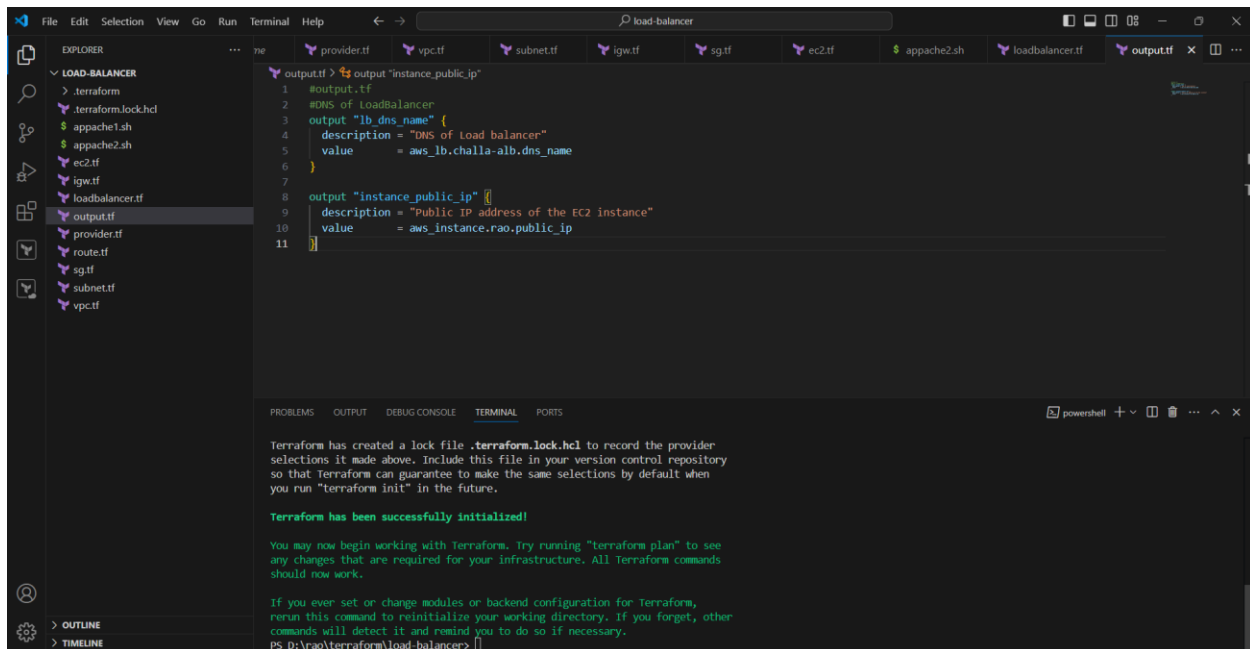


16) Now save all

17) Now click on terminal select new terminal



18) Now click on terminal and use #terraform init



The screenshot shows the Visual Studio Code interface with a file explorer on the left containing Terraform configuration files for a load balancer setup. The main editor displays the `output.tf` file, which defines two outputs: `lb_dns_name` and `instance_public_ip`. The terminal at the bottom shows the output of the `terraform init` command, indicating that Terraform has been successfully initialized and a lock file has been created.

```
output.tf> output "instance_public_ip"
1 #output.tf
2 #DNS of loadBalancer
3 output "lb_dns_name" {
4   description = "DNS of Load balancer"
5   value       = aws_lb.challa-alb.dns_name
6 }
7
8 output "instance_public_ip" {
9   description = "Public IP address of the EC2 instance"
10  value       = aws_instance.rao.public_ip
11 }
```

Terraform has created a lock file `.terraform.lock.hcl` to record the provider selections it made above. Include this file in your version control repository so that Terraform can guarantee to make the same selections by default when you run "terraform init" in the future.

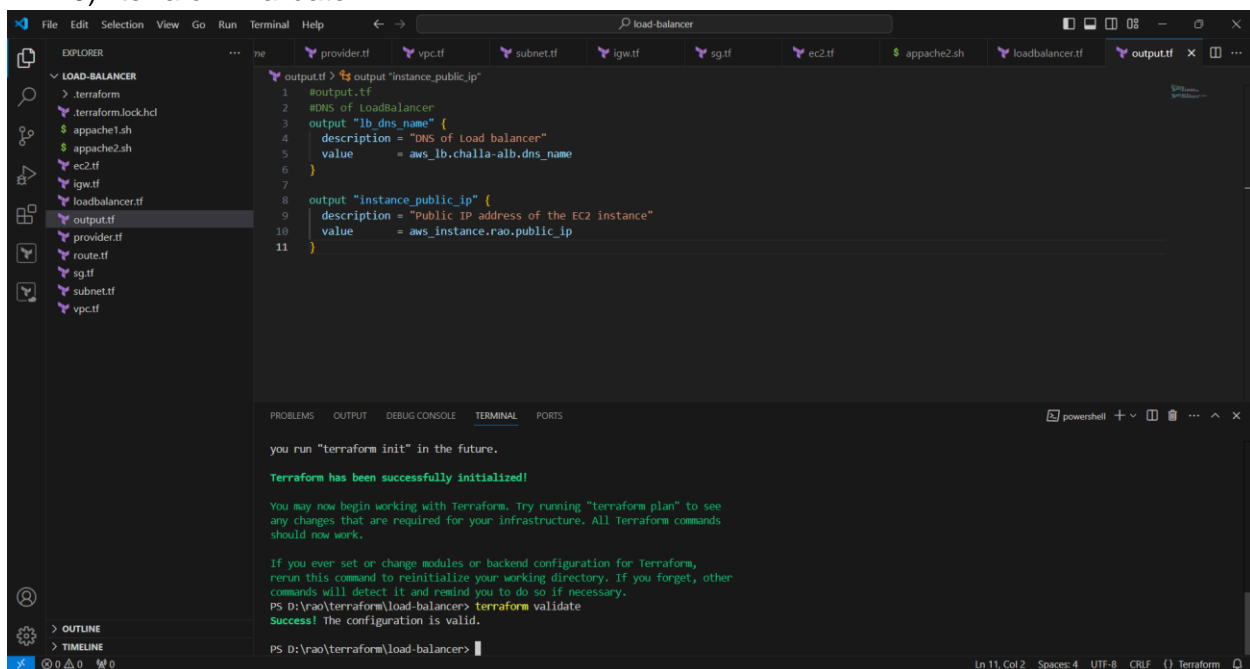
Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see any changes that are required for your infrastructure. All Terraform commands should now work.

If you ever set or change modules or backend configuration for Terraform, rerun this command to reinitialize your working directory. If you forget, other commands will detect it and remind you to do so if necessary.

PS D:\rao\terraform\load-balancer>

19) #terraform validate



This screenshot shows the same VS Code environment after running the `terraform validate` command. The terminal output confirms that the configuration is valid and reiterates the successful initialization of Terraform.

```
output.tf> output "instance_public_ip"
1 #output.tf
2 #DNS of loadBalancer
3 output "lb_dns_name" {
4   description = "DNS of Load balancer"
5   value       = aws_lb.challa-alb.dns_name
6 }
7
8 output "instance_public_ip" {
9   description = "Public IP address of the EC2 instance"
10  value       = aws_instance.rao.public_ip
11 }
```

you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see any changes that are required for your infrastructure. All Terraform commands should now work.

If you ever set or change modules or backend configuration for Terraform, rerun this command to reinitialize your working directory. If you forget, other commands will detect it and remind you to do so if necessary.

PS D:\rao\terraform\load-balancer> terraform validate
Success! The configuration is valid.

PS D:\rao\terraform\load-balancer>

20) Now terraform plan #terraform plan

The screenshot shows the VS Code interface with the Explorer pane on the left displaying a project structure for 'LOAD-BALANCER'. The main editor shows the 'output.tf' file with the following content:

```
1 output "instance_public_ip"
2 {
3   #DNS of LoadBalancer
4   output "lb_dns_name" {
5     description = "DNS of Load balancer"
6     value       = aws_lb.challa-alb.dns_name
7   }
8   output "instance_public_ip" {
9     description = "Public IP address of the EC2 instance"
10    value       = aws_instance.rao.public_ip
11  }
12 }
```

The bottom pane shows the output of the 'terraform plan' command. It indicates that there are no changes to be made to the infrastructure, as the resources already exist in the target state.

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

Changes to Outputs:
  + instance_public_ip = (known after apply)
  + lb_dns_name        = (known after apply)

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run "terraform apply" now.
PS D:\rao\terraform\load-balancer>
```

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

The screenshot shows the VS Code interface with the Explorer pane on the left displaying the project structure. The main editor shows the 'output.tf' file. The bottom pane shows the output of the 'terraform apply' command, which successfully creates the infrastructure resources.

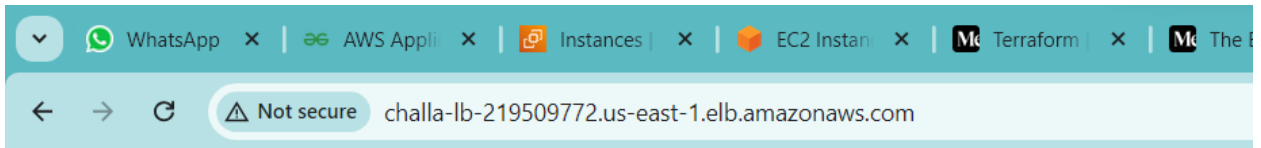
```
aws_internet_gateway.challa-gateway: Creating...
aws_subnet.challa-public-subnet2: Creating...
aws_lb_target_group.challa-tg: Creating...
aws_subnet.challa-public-subnet1: Creating...
aws_security_group.challa-sg: Creating...
aws_internet_gateway.challa-gateway: Creation complete after 2s [id=igw-00e436a28844b3b7]
aws_route_table.challa-route: Creating...
aws_lb_target_group.challa-tg: Creation complete after 3s [id=arn:aws:elasticloadbalancing:us-east-1:058264331590:targetgroup/challa-TG/9dc64907588778f4]
aws_route_table.challa-route: Creation complete after 2s [id=rtb-0d78a9c9a87c9df3]
aws_security_group.challa-sg: Creation complete after 6s [id=sg-00809dcb20f96f02f]
aws_subnet.challa-public-subnet2: Still creating... [10s elapsed]
aws_subnet.challa-public-subnet1: Still creating... [10s elapsed]
aws_subnet.challa-public-subnet2: Creation complete after 13s [id=subnet-0d9e5c663b0802bc]
aws_route_table_association.challa-route2: Creating...
aws_instance.rao: Creating...
aws_subnet.challa-public-subnet1: Creation complete after 13s [id=subnet-0581dcfcb98f215]
aws_route_table_association.challa-route1: Creating...
aws_lb.challa-alb: Creating...
aws_instance.challa: Creating...
aws_route_table_association.challa-route2: Creation complete after 0s [id=rtbassoc-031fe95bd2c091e]
aws_route_table_association.challa-route1: Creation complete after 1s [id=rtbassoc-03f5a1ddc4afe06e4]

aws_instance.challa: Still modifying... [id=i-01ea62f6f5bbe83b4, 20s elapsed]
aws_instance.challa: Still modifying... [id=i-01ea62f6f5bbe83b4, 30s elapsed]
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.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.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>
```

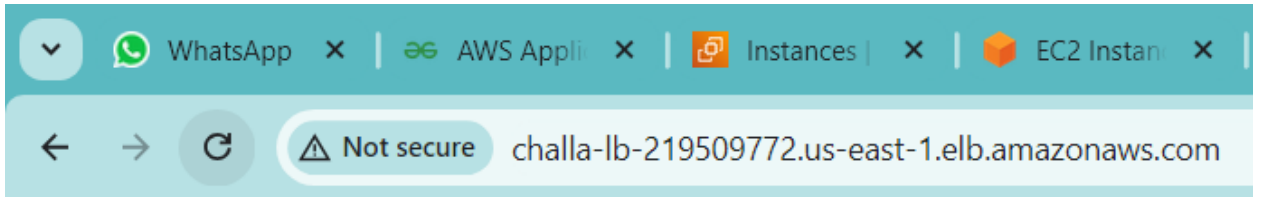
22) Copy dns and paste in google



this is challa

23)

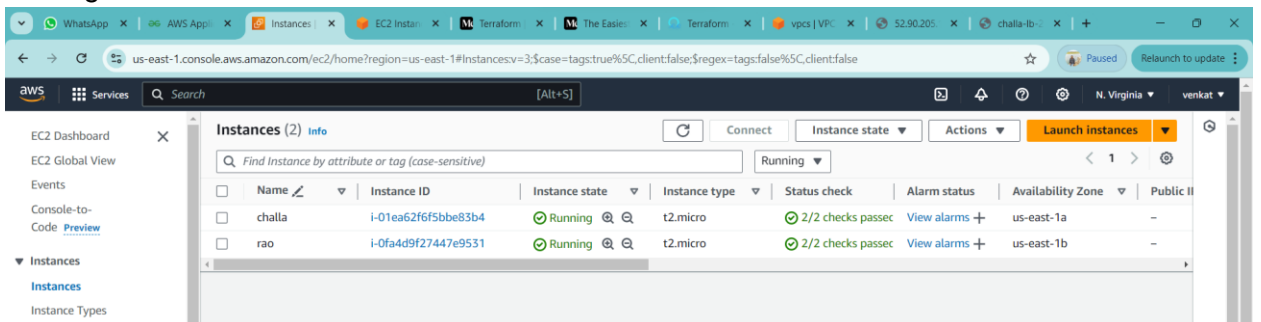
24) Refresh



hi this is rao

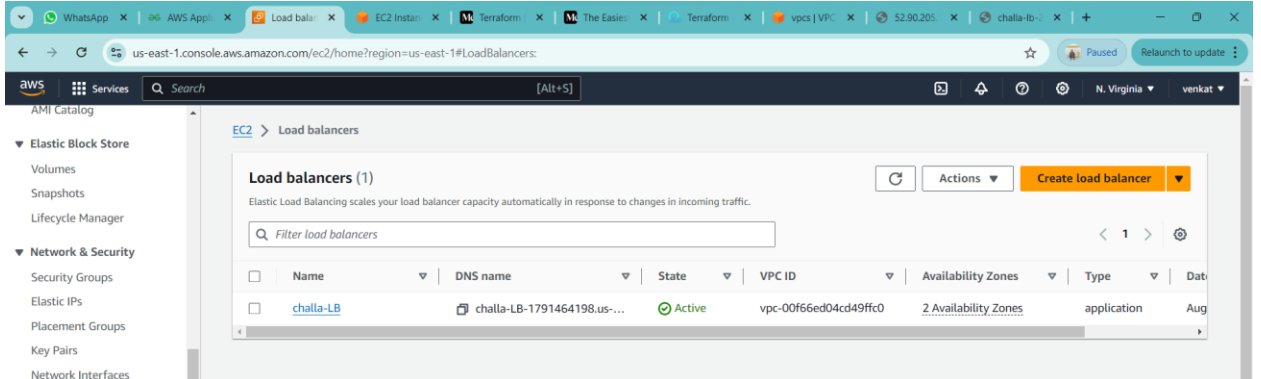
25)

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 interface with a terminal window open. The terminal displays the output of a Terraform destroy command. The output lists the following resources being destroyed:

- aws_instance.rao: Still destroying... [id=i-0fa4d9f27447e9531, 30s elapsed]
- aws_instance.challa: Still destroying... [id=i-01ea62f6f9b8e83b4, 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-01ea62f6f9b8e83b4, 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-01ea62f6f9b8e83b4, 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

The terminal concludes with the message: **Destroy complete! Resources: 15 destroyed.**