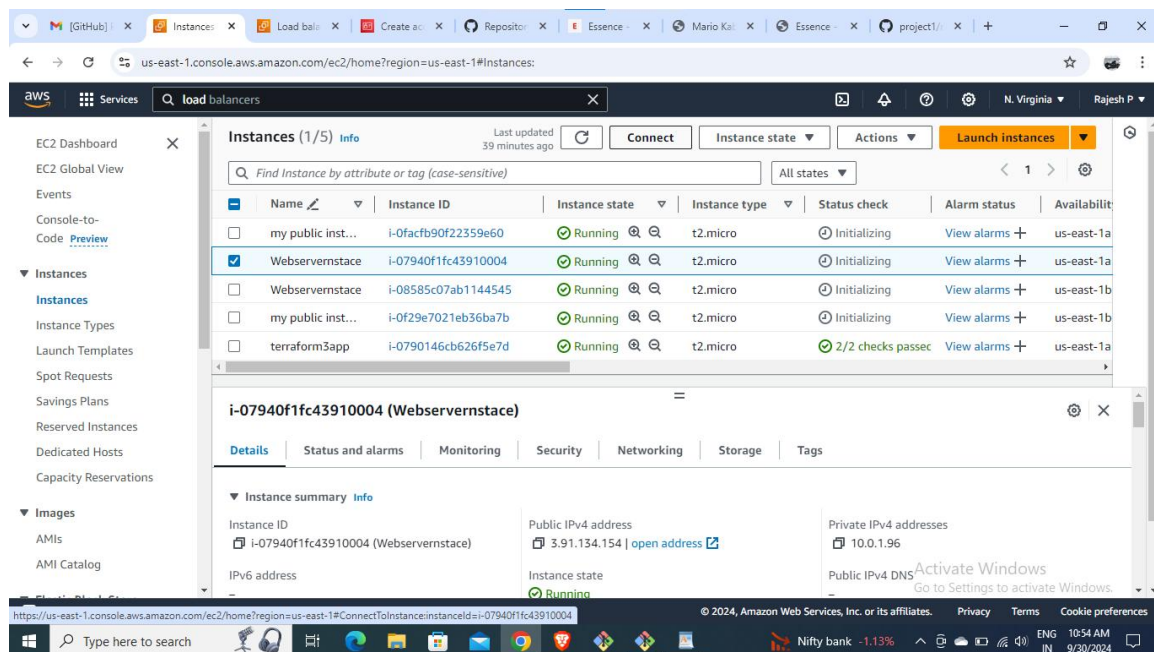


# MAJOR PROJECT-1

## CREATION OF THREE TIER APPLICATION USING TERRAFORM

CREATE AN INSTANCE AND INSTALL TERRAFORM





us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#vpcs:

Virtual private cloud

Your VPCs (1/3) Info

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR
-	vpc-0ea7498f28852b07c	Available	172.31.0.0/16	-
vpc	vpc-034bf63537c14d481	Available	10.0.0.0/16	-
demo VPC	vpc-00fb619ba412225a3	Available	10.0.0.0/16	-

vpc-00fb619ba412225a3 / demo VPC

Details Resource map CIDRs Flow logs Tags Integrations

Details

VPC ID: vpc-00fb619ba412225a3 State: Available DNS hostnames: Disabled DNS resolution: Enabled

## STEP2:SUBNET CREATION

```
ec2-user@ip-172-31-93-245:~/project1
#creating 1st web subnet
resource "aws_subnet" "public_subnet_1" {
  vpc_id            = aws_vpc.demovpc.id
  cidr_block        = var.subnet1_cidr
  map_public_ip_on_launch = true
  availability_zone  = "us-east-1a"
  tags = {
    Name = "web subnet 1"
  }
}

#creating 2st web subnet
resource "aws_subnet" "public_subnet_2" {
  vpc_id            = aws_vpc.demovpc.id
  cidr_block        = var.subnet2_cidr
  map_public_ip_on_launch = true
  availability_zone  = "us-east-1b"
  tags = {
    Name = "web subnet 2"
  }
}

#creating 1st application subnet
resource "aws_subnet" "application_subnet_1" {
  vpc_id            = aws_vpc.demovpc.id
  cidr_block        = var.subnet3_cidr
  map_public_ip_on_launch = false
  availability_zone  = "us-east-1a"
  tags = {
    Name = "application subnet 1"
  }
}

"subnet.tf" 60L, 1738B
```

Activate Windows  
Go to Settings to activate Windows.





us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#RouteTables:

Route tables (1/7) Info

Find resources by attribute or tag

Name	Route table ID	Explicit subnet associ...	Edge associations	Main
-	rtb-00e37c86c234840f2	-	-	Yes
rtb-private2-us-east-1b	rtb-0e3ca40d1bb40b79f2	subnet-0f800a8caa661cc...	-	No
-	rtb-0b33466ef75766ea5	-	-	Yes
rtb-public	rtb-008989227a10b6259	2 subnets	-	No
route to internet	rtb-07308faba40da56cb	2 subnets	-	No
rtb-private1-us-east-1a	rtb-044574507ce877d07	subnet-01ae7576cad057	-	No

rtb-07308faba40da56cb / route to internet

Details Routes Subnet associations Edge associations Route propagation Tags

Details

Route table ID	Main	Explicit subnet associations	Edge associations
rtb-07308faba40da56cb	No	2 subnets	Go to Settings to activate Windows

## STEP5:CREATION OF TWO EC2 INSTANCES

```
ec2-user@ip-172-31-93-245:~/project1
resource "aws_instance" "demoinstance" {
  ami           = "ami-0ebfd941bbafe70c6"
  instance_type = "t2.micro"
  key_name      = "3app"
  vpc_security_group_ids = ["${aws_security_group.demosg.id}"]
  subnet_id     = aws_subnet.public_subnet_1.id
  associate_public_ip_address = true
  user_data     = file("data.sh")
  tags = {
    Name = "my public instance1"
  }
}
```

"ec2.tf" 12L, 444B

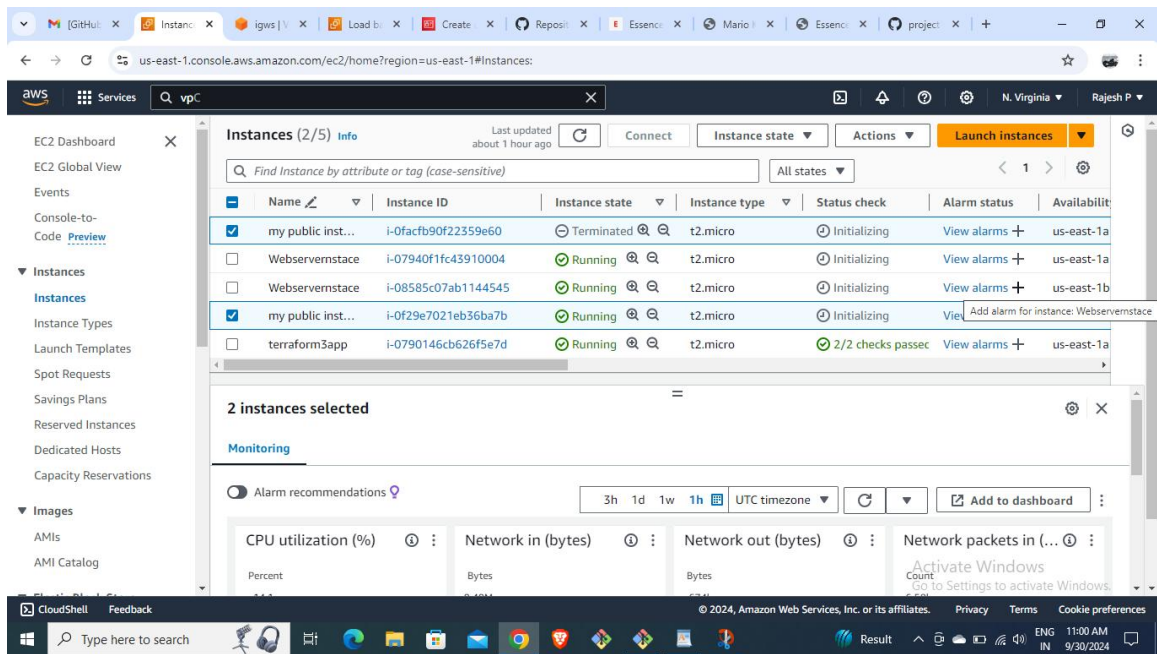
Activate Windows  
Go to Settings to activate Windows.  
4,37 A11



```
ec2-user@ip-172-31-93-245:~/project1
resource "aws_instance" "demoinstance1" {
  ami           = "ami-0ebfd941bbafe70c6"
  instance_type = "t2.micro"
  key_name      = "3app"
  vpc_security_group_ids = ["${aws_security_group.demosg.id}"]
  subnet_id     = aws_subnet.public_subnet_2.id
  associate_public_ip_address = true
  user_data     = file("data2.sh")
  tags = {
    Name = "my public instance2"
  }
}
```

"insta.tf" 13L, 447B

Activate Windows  
Go to Settings to activate Windows.  
8,43 A11



## STEP6:CREATION OF SECURITY GROUPS

```
ec2-user@ip-172-31-93-245:~/project1
resource "aws_security_group" "demosg" {
  vpc_id = aws_vpc.demovpc.id
  #inbound rules
  #HTTPS access from anywhere
  ingress {
    from_port = 80
    to_port   = 80
    protocol  = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }
  #http access from any where
  ingress {
    from_port = 443
    to_port   = 443
    protocol  = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }
  #http access from any where
  ingress {
    from_port = 22
    to_port   = 22
    protocol  = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }
  #outbound rules
  #internet access to any where
  egress {
    from_port = 0
    to_port   = 0
  }
}
"web_sg.tf" 36L, 719B

Activate Windows
Go to Settings to activate Windows.
1,1 Top
```

```
ec2-user@ip-172-31-93-245:~/project1
#Create database sg
resource "aws_security_group" "database_sg" {
  name = "database_sg"
  description = "allow inbound traffic from application layer"
  vpc_id = aws_vpc.demovpc.id

  ingress {
    description = "allow traffic from application layer"
    from_port = 3306
    to_port = 3306
    protocol = "tcp"
    security_groups = [aws_security_group.demosg.id]
  }

  egress {
    from_port = 32768
    to_port = 65535
    protocol = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }

  tags = {
    Name = "database_sg"
  }
}
~
~
~
"database_sg.tf" 25L, 570B

Activate Windows
Go to Settings to activate Windows.
1,1 A11
```

## STEP7:CREATION OF LOAD BALANCER



```
ec2-user@ip-172-31-93-245:~/project1
resource "aws_lb" "external_alb" {
  name            = "external-alb"
  internal        = false
  load_balancer_type = "application"
  security_groups = [aws_security_group.demosg.id]
  subnets        = [aws_subnet.public_subnet_1.id, aws_subnet.public_subnet_2.id]
}

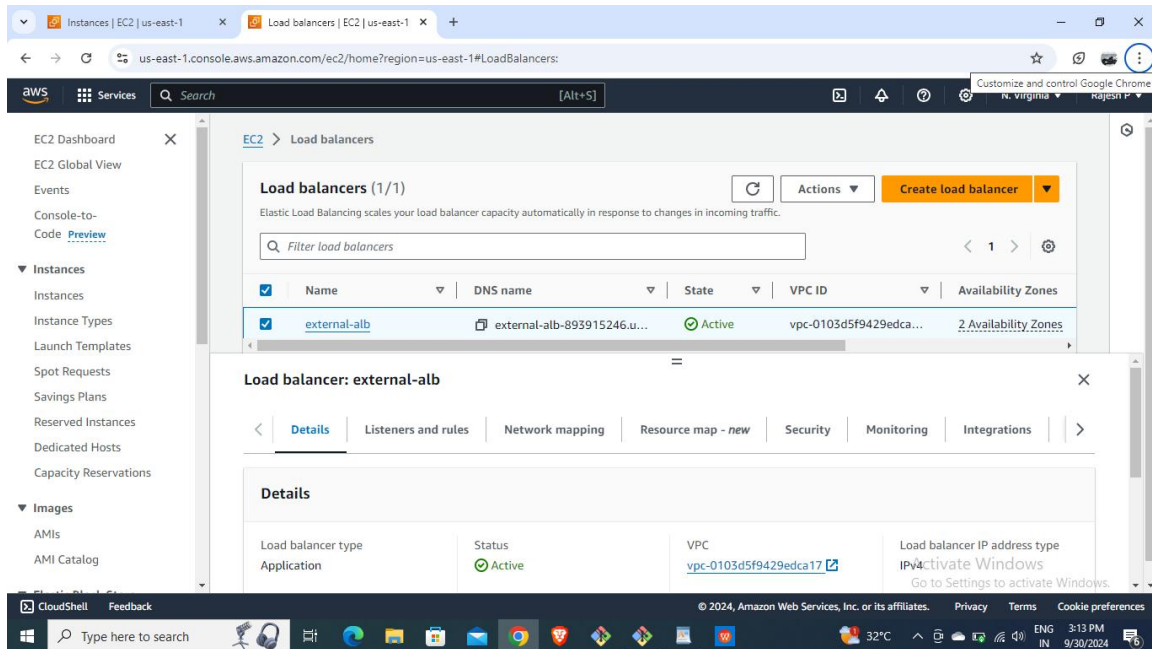
resource "aws_lb_target_group" "target_elb" {
  name     = "ALB-TG"
  port     = 80
  protocol = "HTTP"
  vpc_id   = aws_vpc.demovpc.id
}

resource "aws_lb_target_group_attachment" "attachment_instance" {
  target_group_arn = aws_lb_target_group.target_elb.arn
  target_id        = aws_instance.demoinstance.id
  port             = 80
  depends_on       = [aws_instance.demoinstance]
}

resource "aws_lb_target_group_attachment" "attachment_instance_1" {
  target_group_arn = aws_lb_target_group.target_elb.arn
  target_id        = aws_instance.demoinstance1.id
  port             = 80
  depends_on       = [aws_instance.demoinstance1]
}

"alb.tf" 39L, 1182B
```

Activate Windows  
Go to Settings to activate Windows.  
1,1 Top



## STEP8: CREATION IF AUTO SCALING GROUPS

```
ec2-user@ip-172-31-93-245:~/project1
resource "aws_launch_template" "web_launch_template" {
  name_prefix      = "web-launch-template"
  image_id         = "ami-0ebfd941bbafe70c6"
  instance_type    = "t2.micro"
  key_name         = "3app"
  vpc_security_group_ids = [aws_security_group.demosg.id]
  tag_specifications {
    resource_type = "instance"
    tags = {
      Name = "webinstance"
    }
  }
}

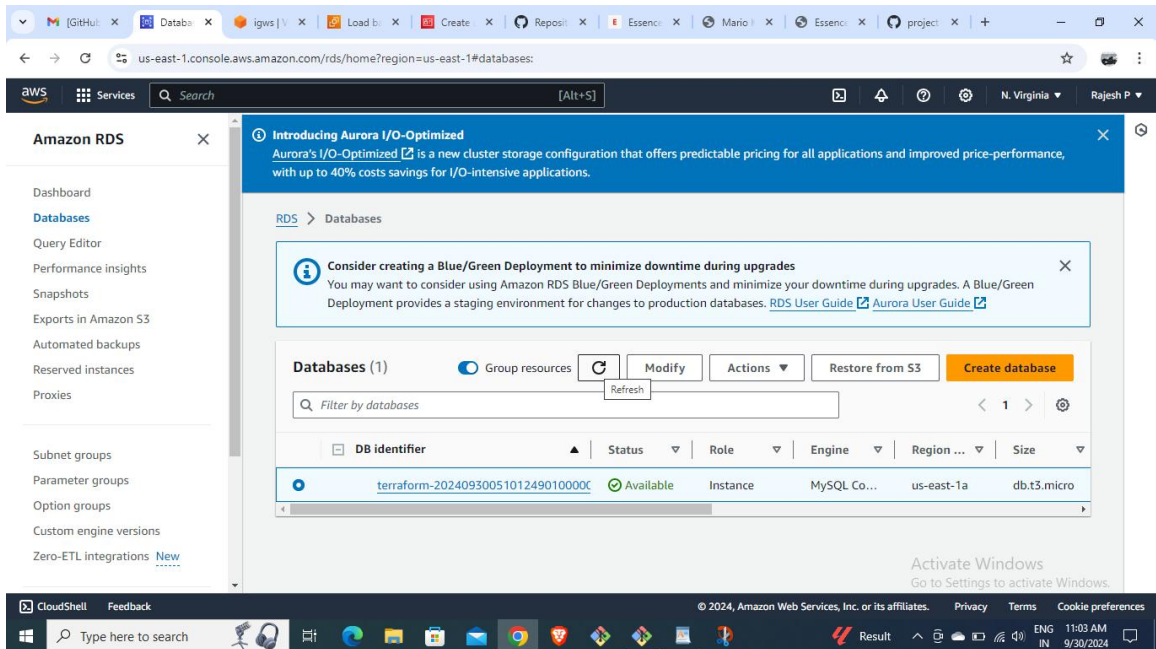
resource "aws_autoscaling_group" "web_asg" {
  desired_capacity = 2
  max_size        = 5
  min_size        = 1
  launch_template {
    id      = aws_launch_template.web_launch_template.id
    version = "$Latest"
  }
  vpc_zone_identifier = [aws_subnet.public_subnet_1.id, aws_subnet.public_subnet_2.id]
  health_check_type   = "EC2"
  health_check_grace_period = 300
  tag {
    key      = "Name"
    value    = "Webserverinstance"
    propagate_at_launch = true
  }
}

"asg.tf" 34L, 1081B
```

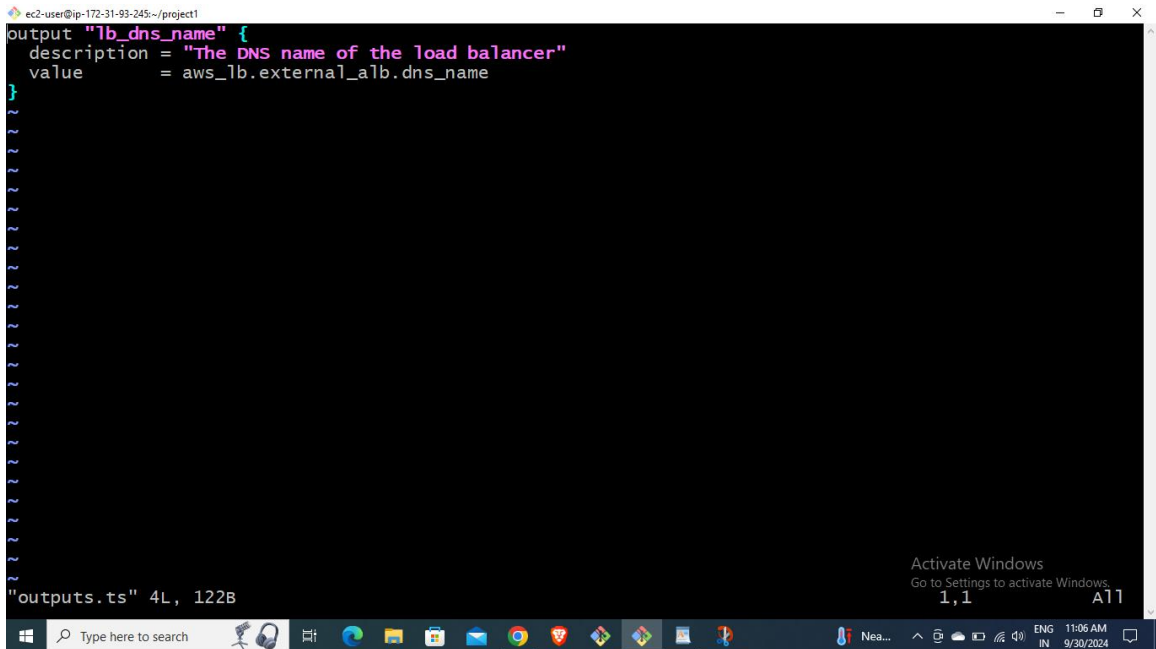
Activate Windows  
Go to Settings to activate Windows.  
6,28 Top

27°C 11:01 AM 9/30/2024





## STEP10:FILE FOR OUTPUT



STEP11:BROWSE THE URL OF LB AND IT SHOWS LOAD IS BALANCING B/W TWO INSTANCES.

