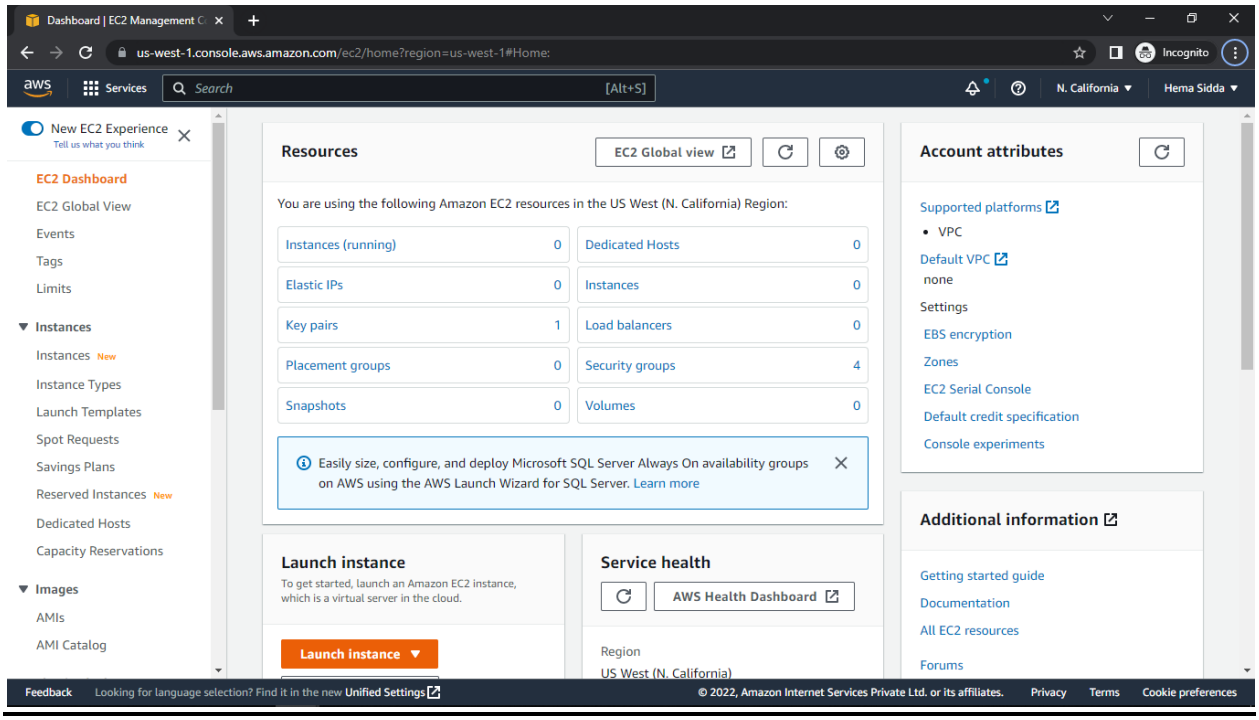
DEPLOY THREE-TIER ARCHITECHTURE IN AWS BY USING TERRAFORM

Method-1

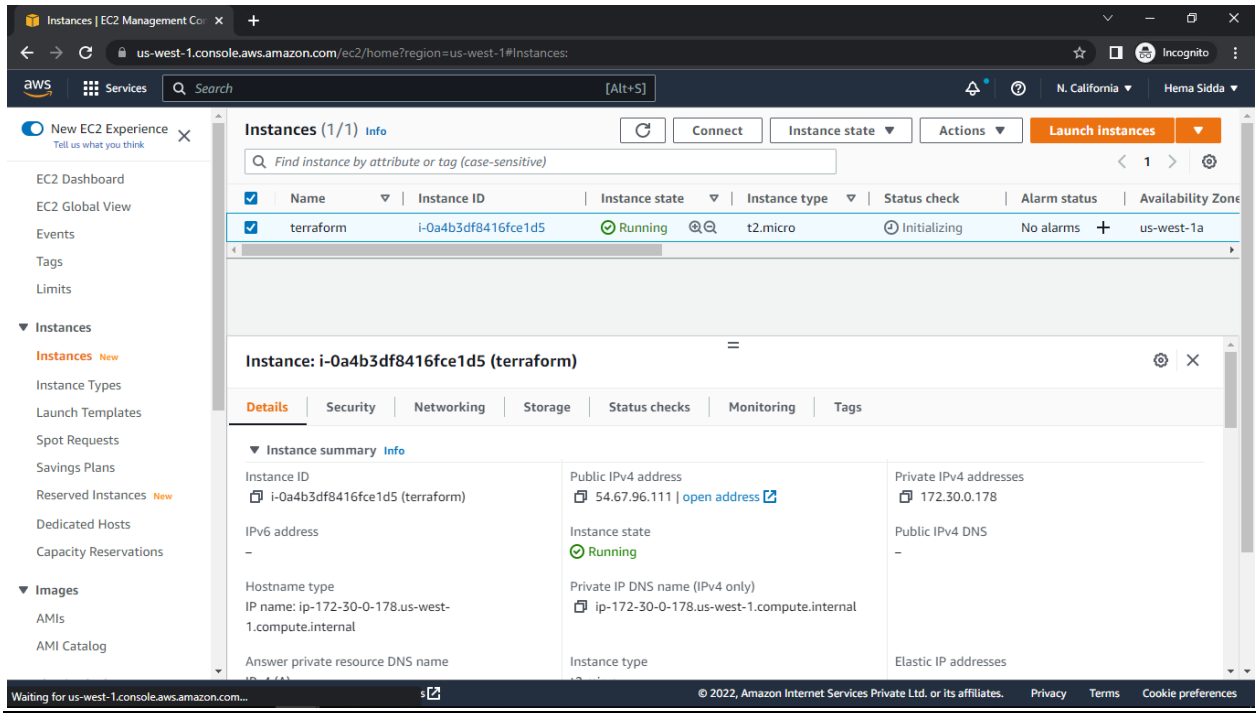
- First open the AWS account by using credentials and go with EC2 instances.



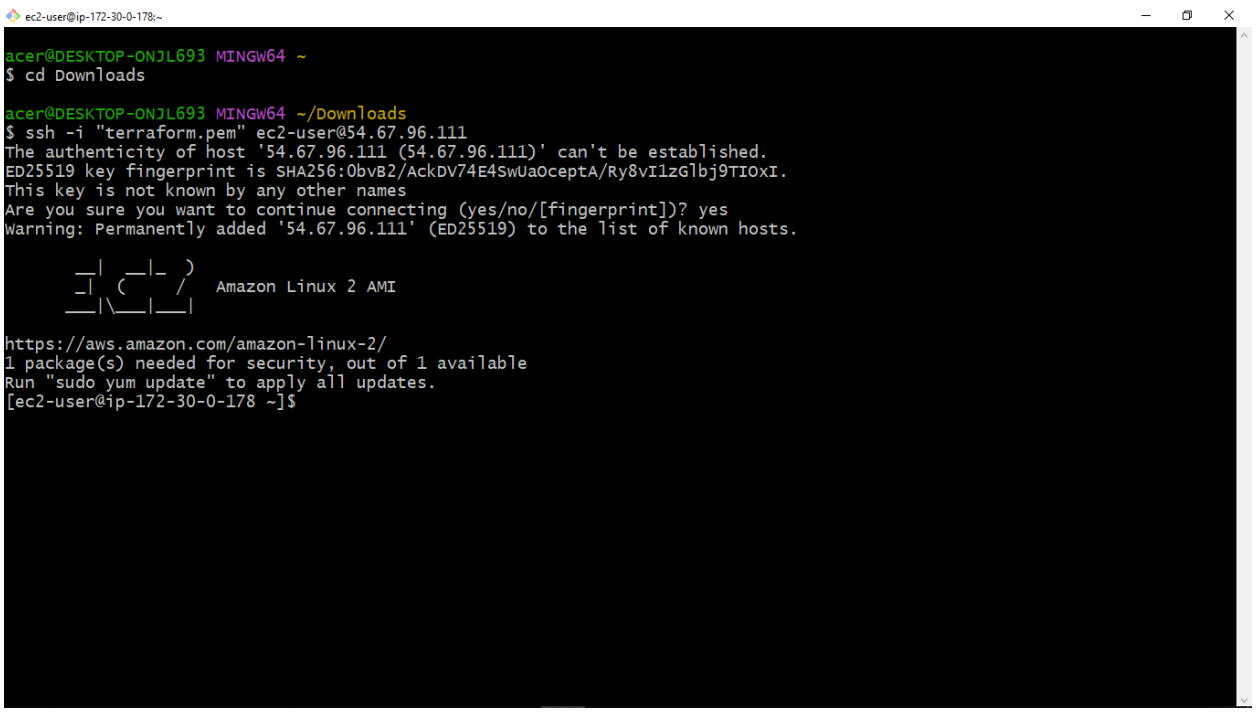
- Now create or launch the server by selecting AMI's and key-pairs and also give the port ranges like SSH(22) and HTTP(80).

SHAIK GOUSERABBANI

DEPLOY THREE-TIER ARCHITECHTURE IN AWS BY USING TERRAFORM



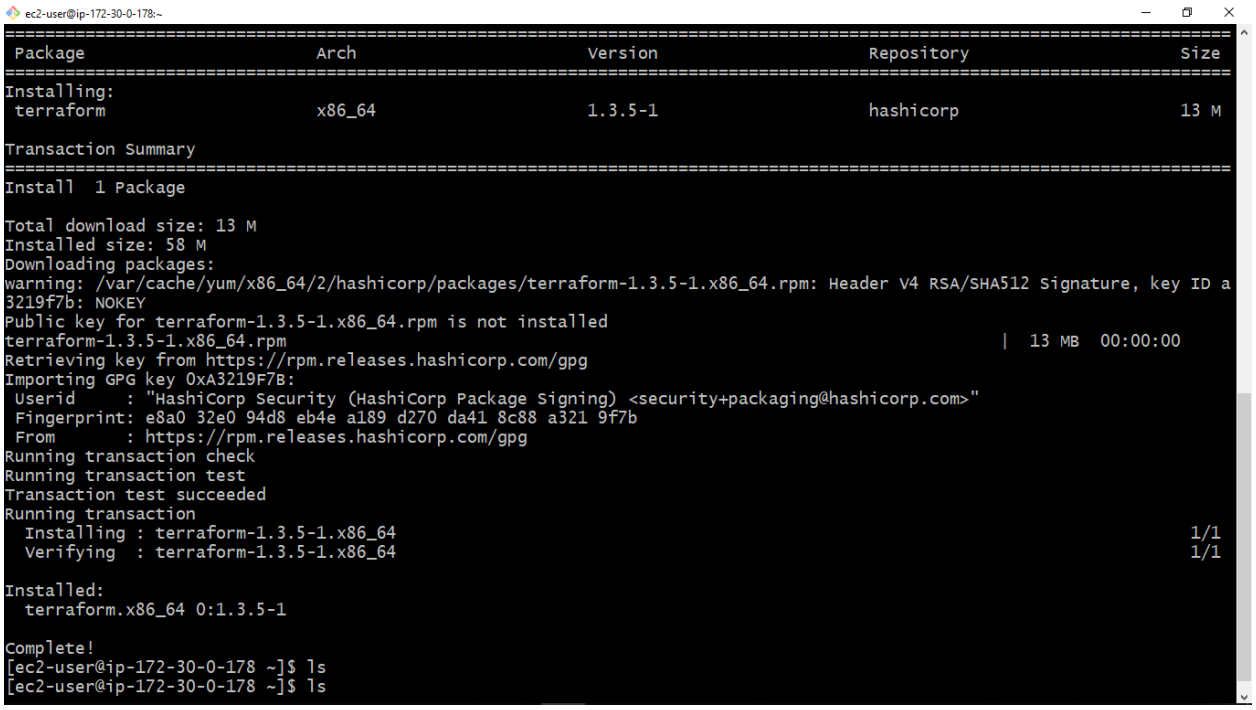
➢ Now connect the launched server through GIT-Bash terminal.



SHAIK GOUSERABBANI

DEPLOY THREE-TIER ARCHITECHTURE IN AWS BY USING TERRAFORM

- Now install the terraform by using commands those are available in Google browse it by the name <Terraform download>.

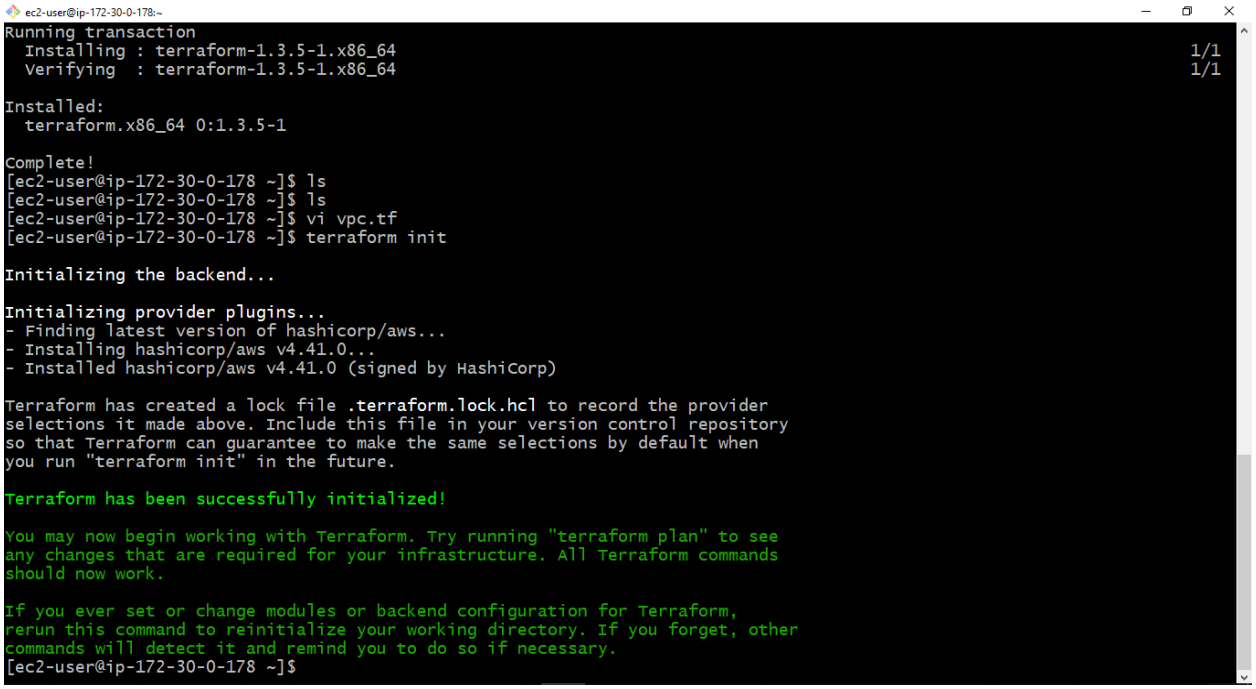


- Now create vpc by using yaml scripting for that create a file as <vi file.tf> and write the yaml script to create vpc and save it by using ":wq".

SHAIK GOUSERABBANI

DEPLOY THREE-TIER ARCHITECHTURE IN AWS BY USING TERRAFORM

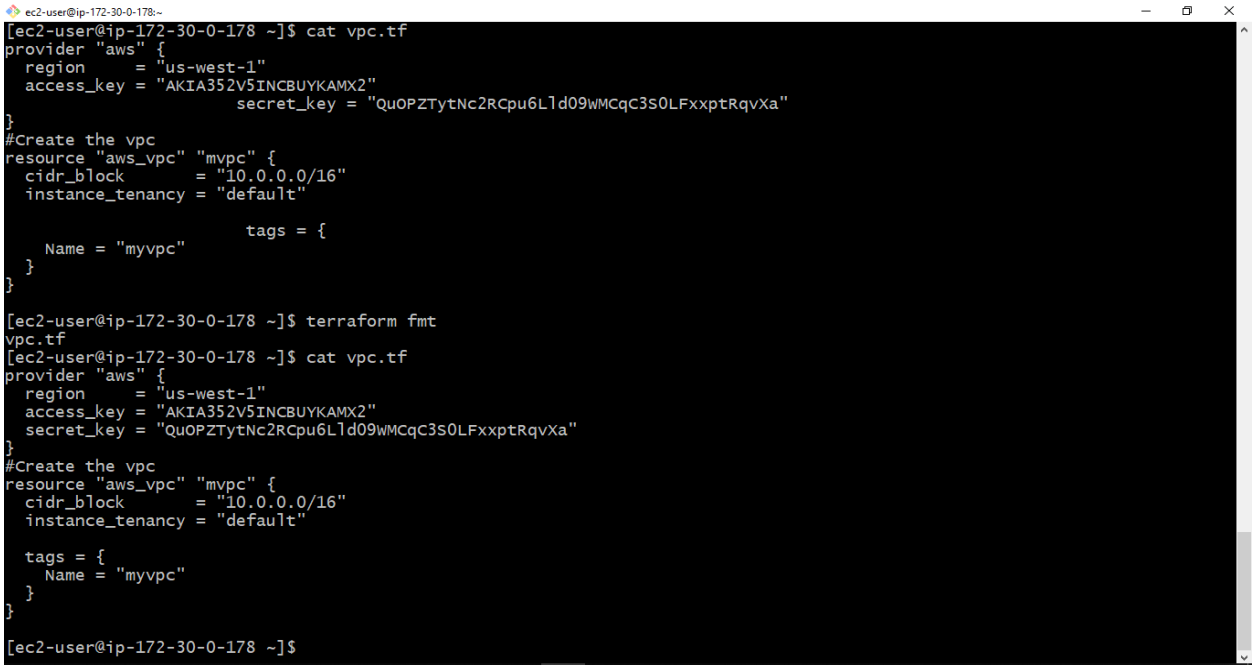
- Now change the environment into terraform mode by using command as <terraform init>.



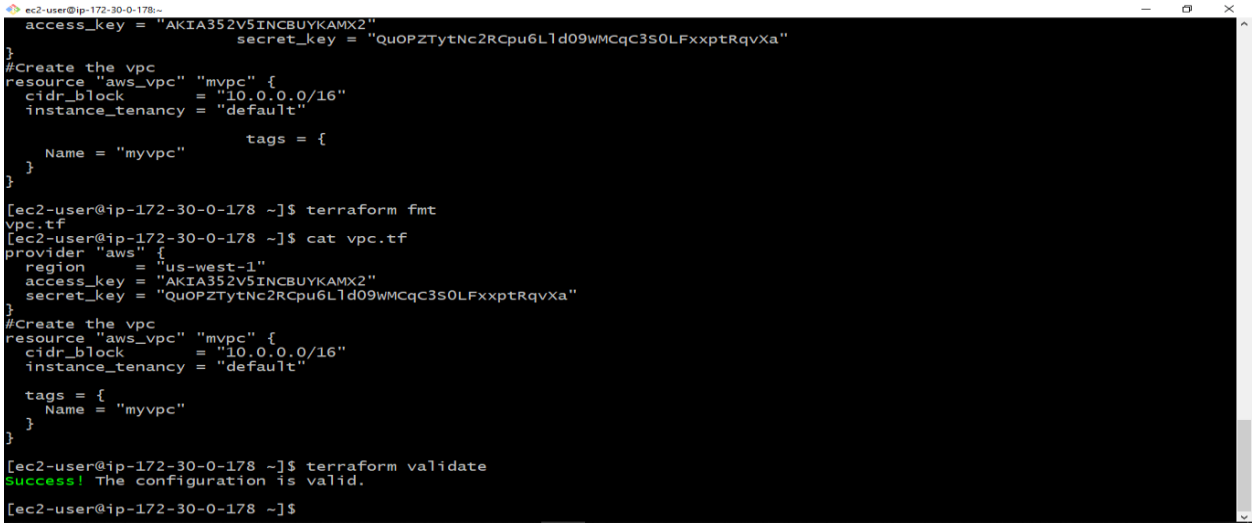
SHAIK GOUSERABBANI

DEPLOY THREE-TIER ARCHITECHTURE IN AWS BY USING TERRAFORM

- Now set the yaml script alignment by using command as <terraform fmt>.

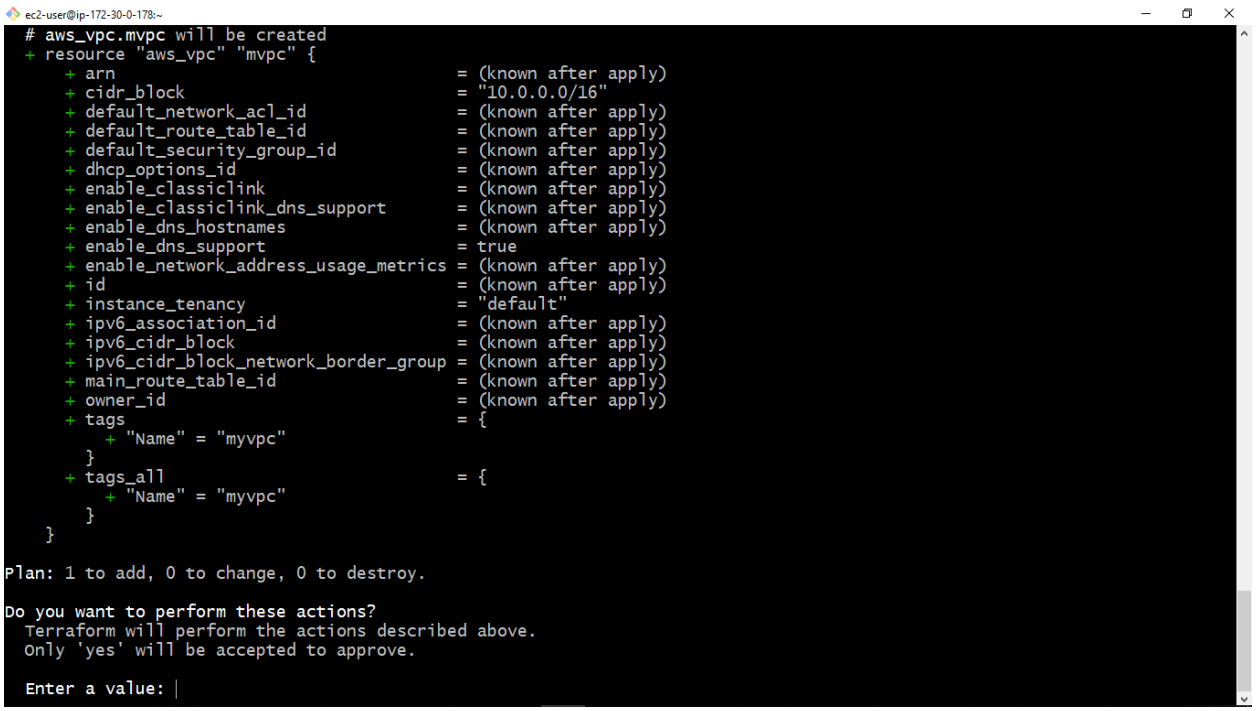


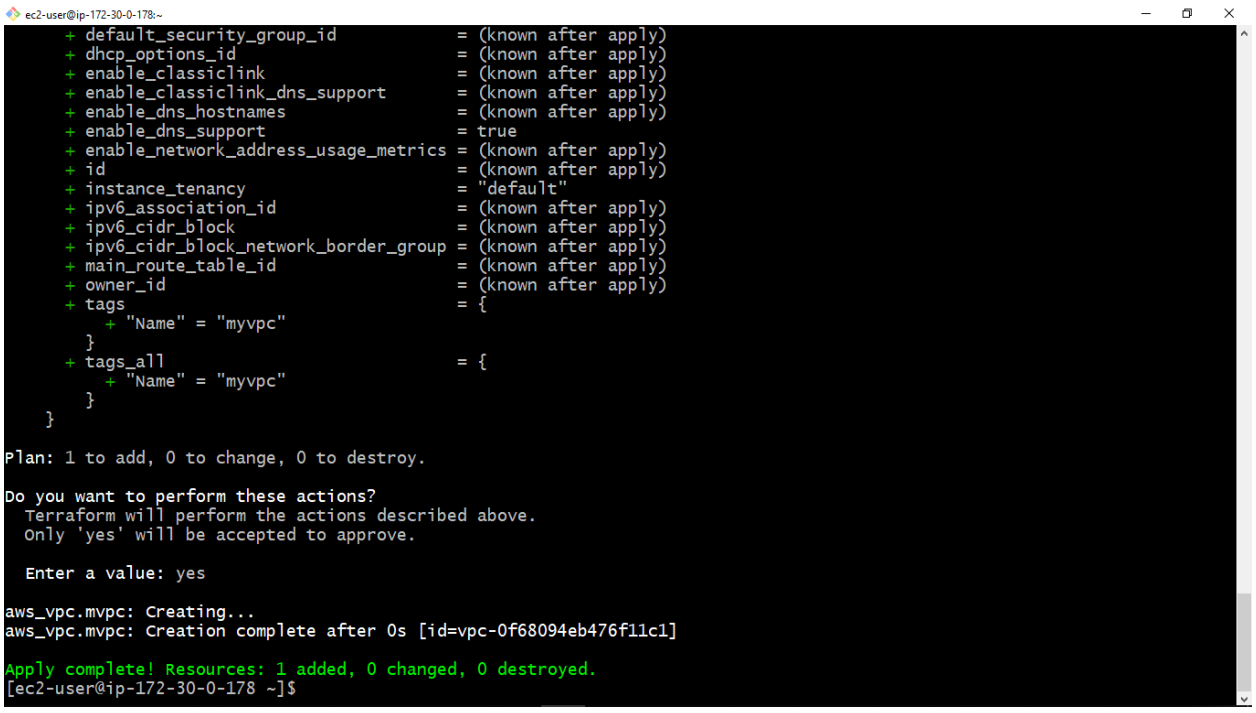
- Now validate the script for correcting the spelling mistakes by using command as <terraform validate>.



DEPLOY THREE-TIER ARCHITECHTURE IN AWS BY USING TERRAFORM

➢ Now create the vpc by using command as <terraform apply>.

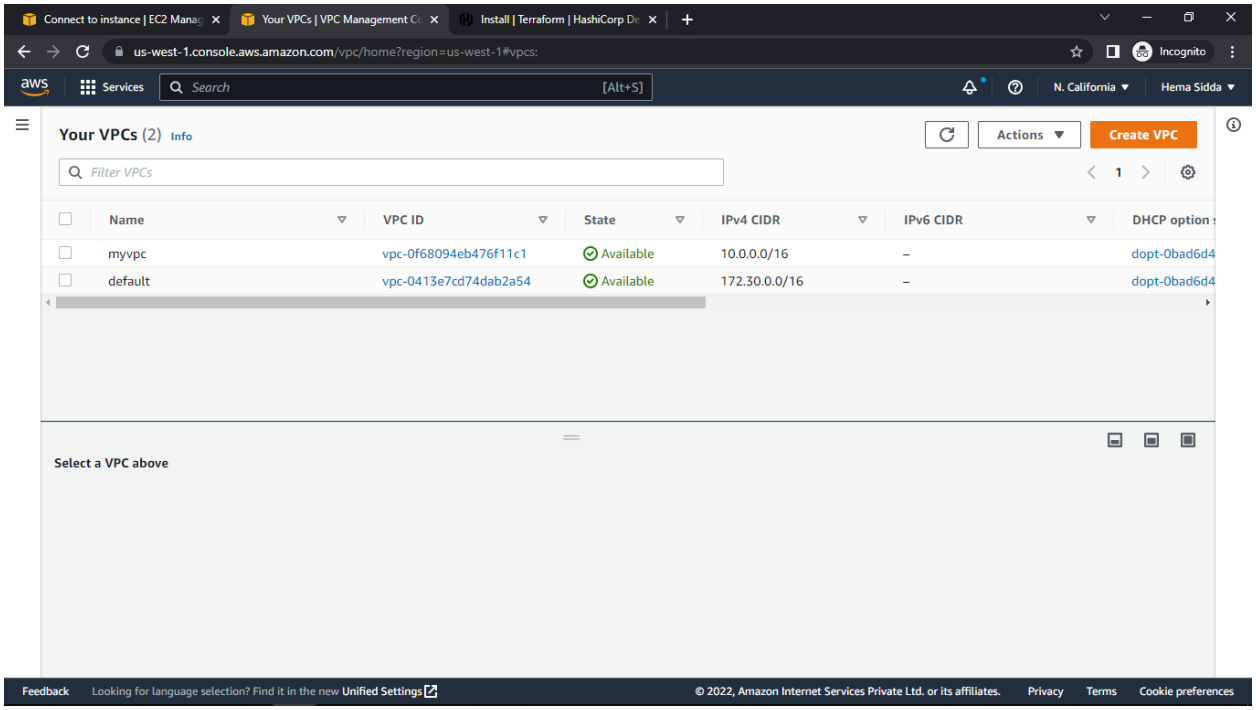




SHAIK GOUSERABBANI

DEPLOY THREE-TIER ARCHITECHTURE IN AWS BY USING TERRAFORM

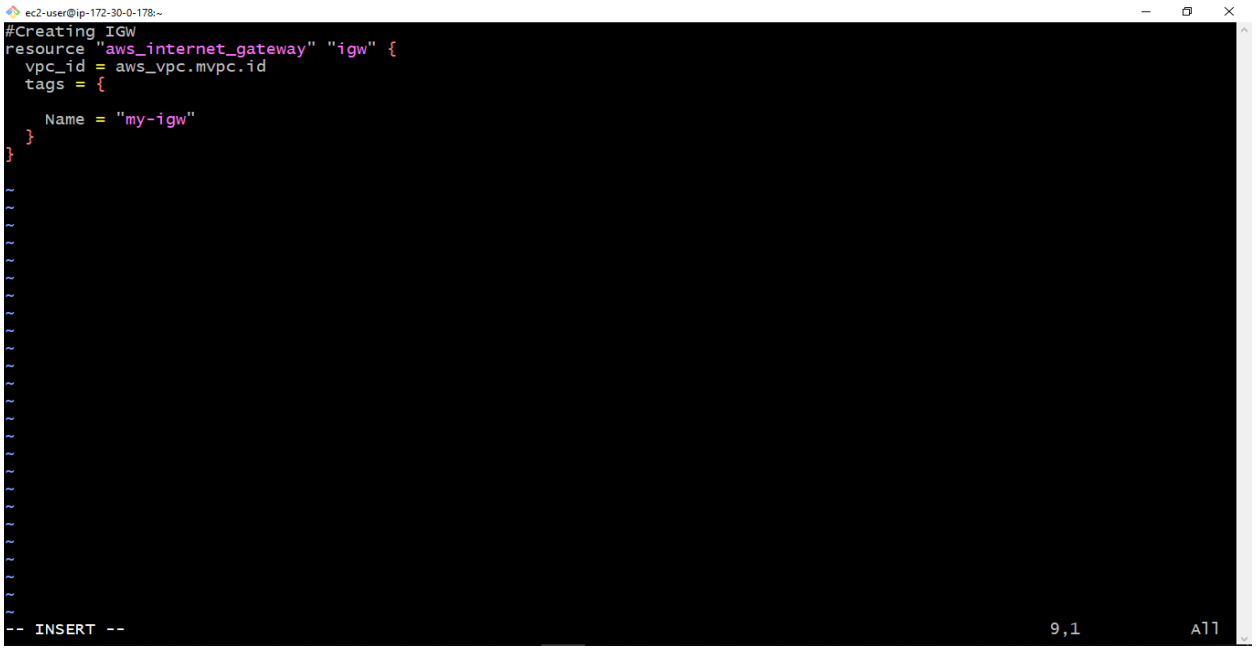
- Now go to your AWS account and open vpc services and check your vpc created or not.

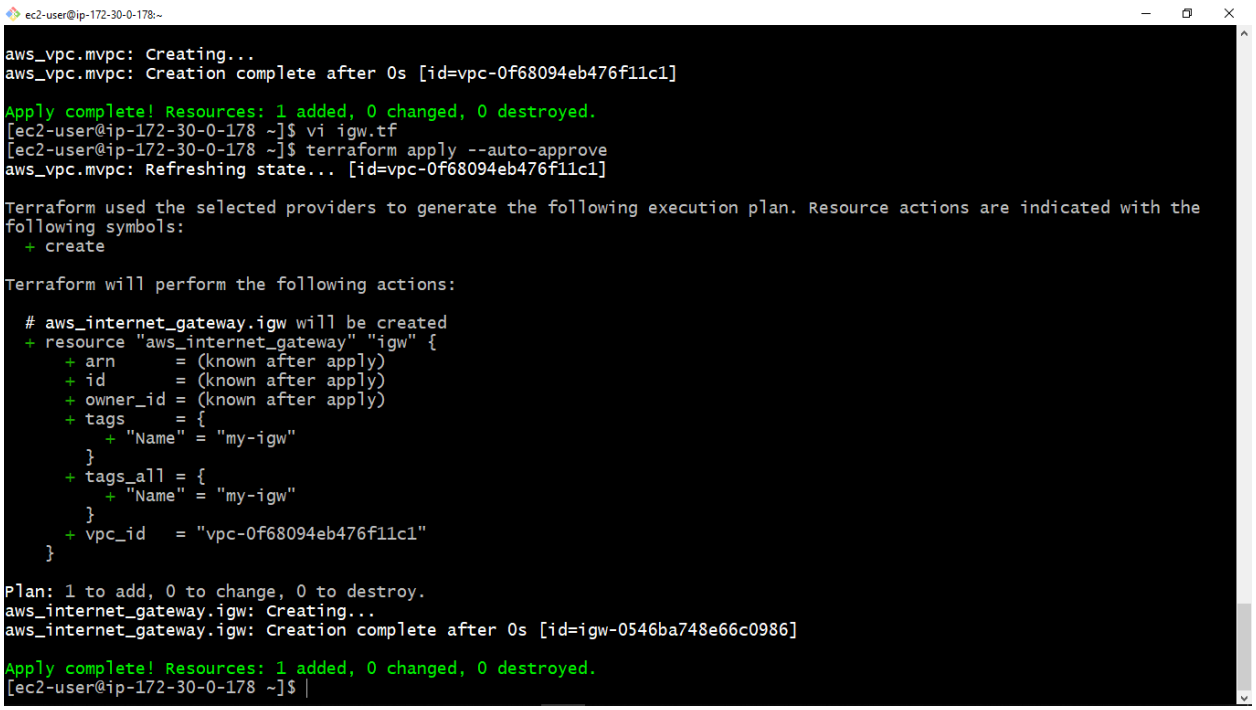


- Now create the IGW and attach it to created vpc for that create a file as <vi igw.tf>

SHAIK GOUSERABBANI

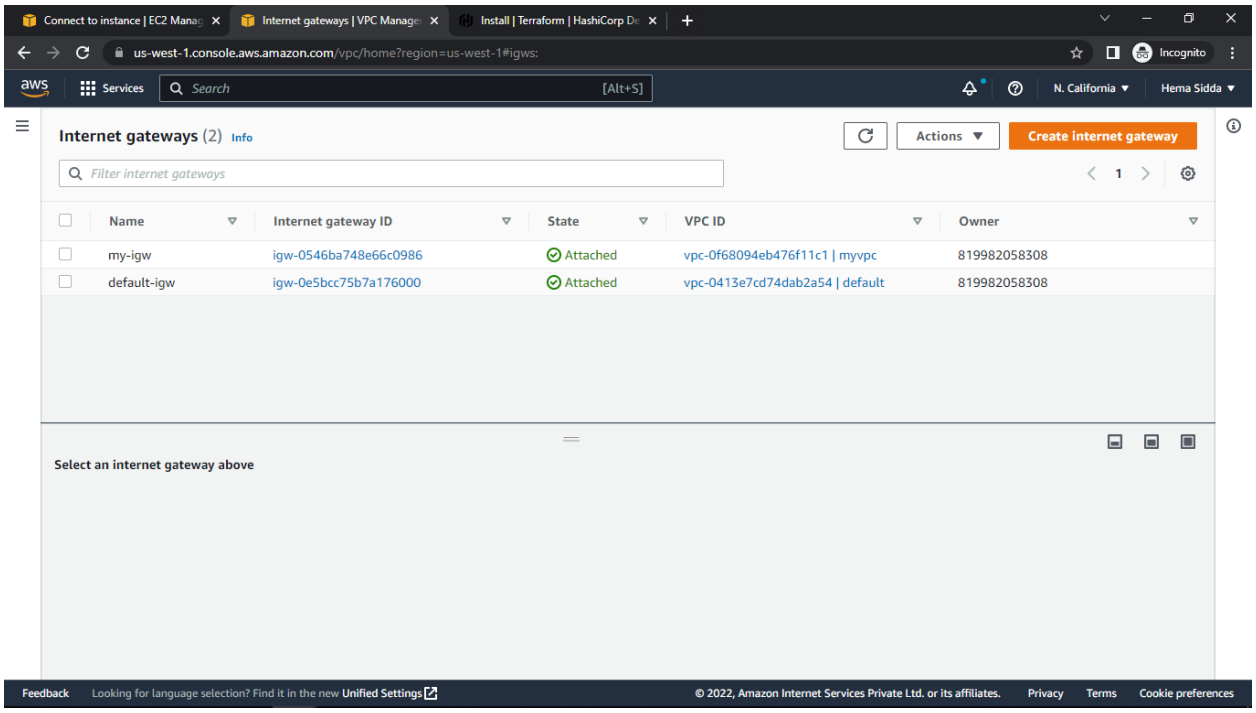
DEPLOY THREE-TIER ARCHITECHTURE IN AWS BY USING TERRAFORM



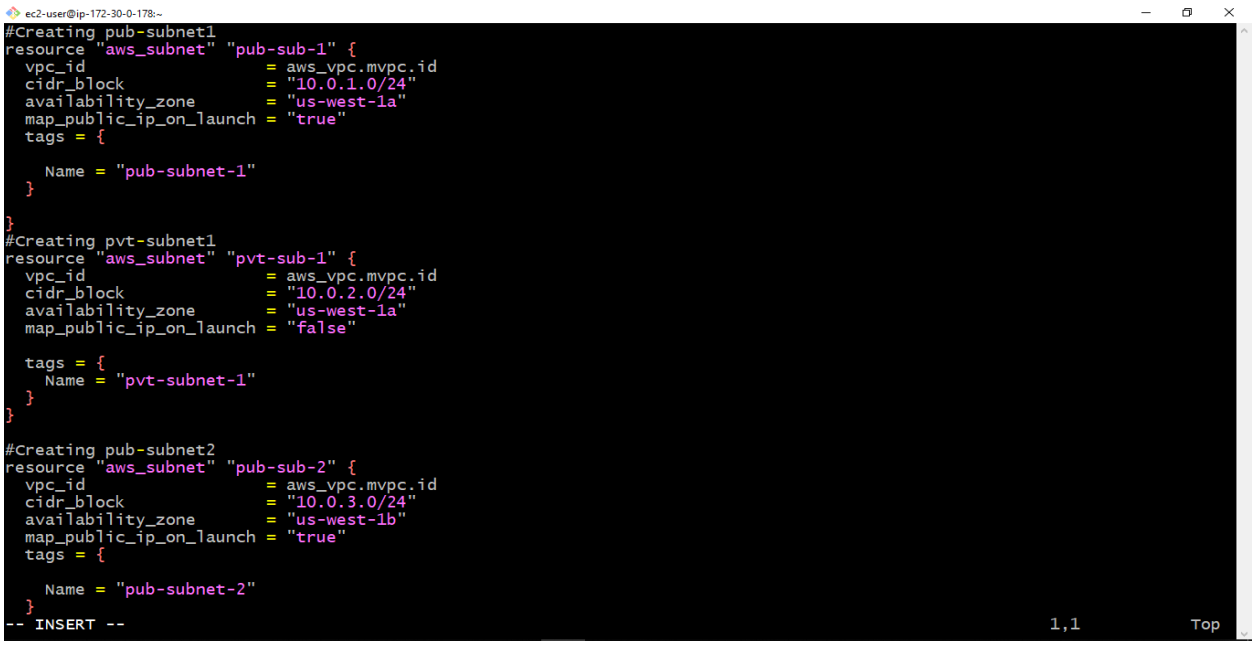


SHAIK GOUSERABBANI

DEPLOY THREE-TIER ARCHITECHTURE IN AWS BY USING TERRAFORM



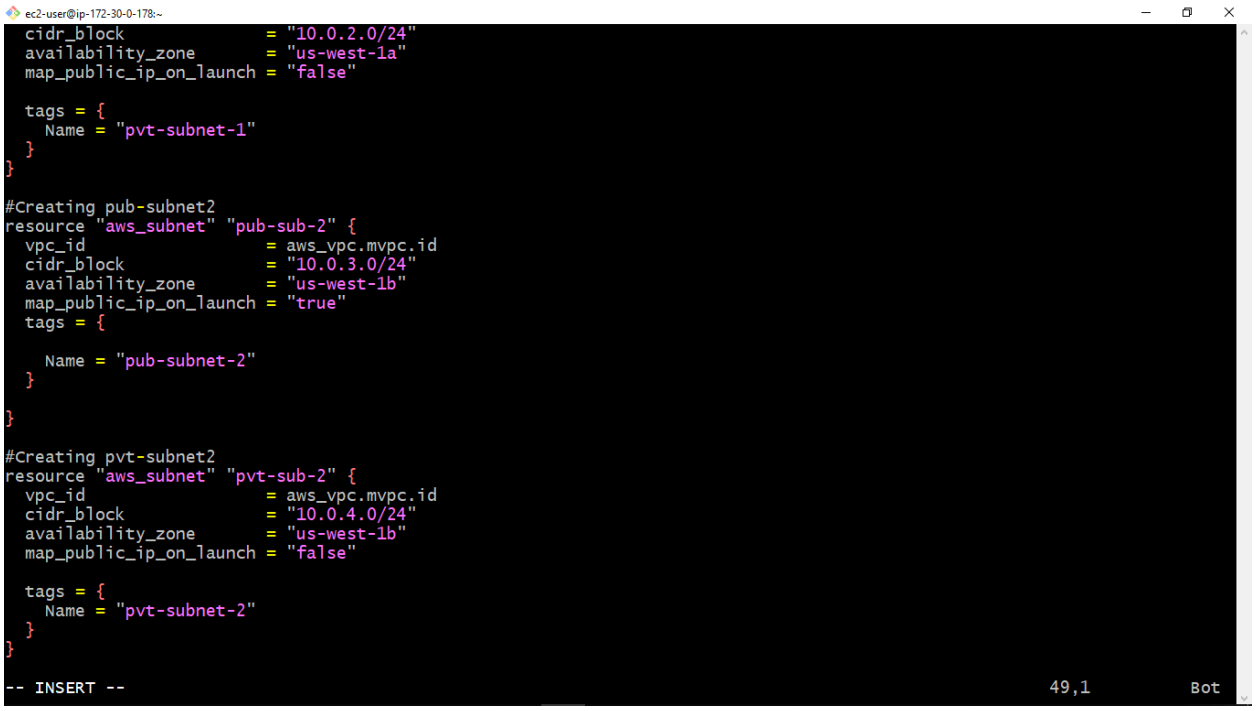
➢ Now create the subnets by using yaml script for that use this command <vi subnets.tf>



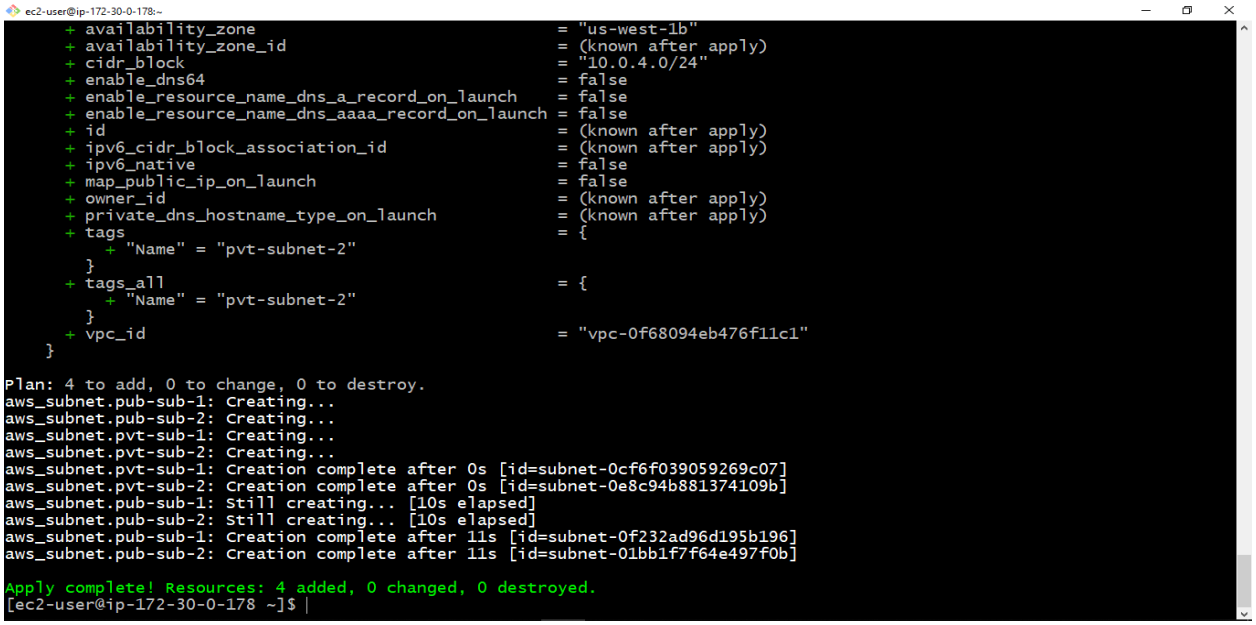
```
#Creating pub-subnet1
resource "aws_subnet" "pub-sub-1" {
  vpc_id                  = aws_vpc.mvpc.id
  cidr_block              = "10.0.1.0/24"
  availability_zone       = "us-west-1a"
  map_public_ip_on_launch = "true"
  tags = {

    Name = "pub-subnet-1"
  }

}
#Creating pvt-subnet1
resource "aws_subnet" "pvt-sub-1" {
  vpc_id                  = aws_vpc.mvpc.id
  cidr_block              = "10.0.2.0/24"
  availability_zone       = "us-west-1a"
  map_public_ip_on_launch = "false"

  tags = {
    Name = "pvt-subnet-1"
  }
}

#Creating pub-subnet2
resource "aws_subnet" "pub-sub-2" {
  vpc_id                  = aws_vpc.mvpc.id
  cidr_block              = "10.0.3.0/24"
  availability_zone       = "us-west-1b"
  map_public_ip_on_launch = "true"
  tags = {

    Name = "pub-subnet-2"
  }
-- INSERT --                                                    1,1           Top
```

SHAIK GOUSERABBANI

DEPLOY THREE-TIER ARCHITECHTURE IN AWS BY USING TERRAFORM

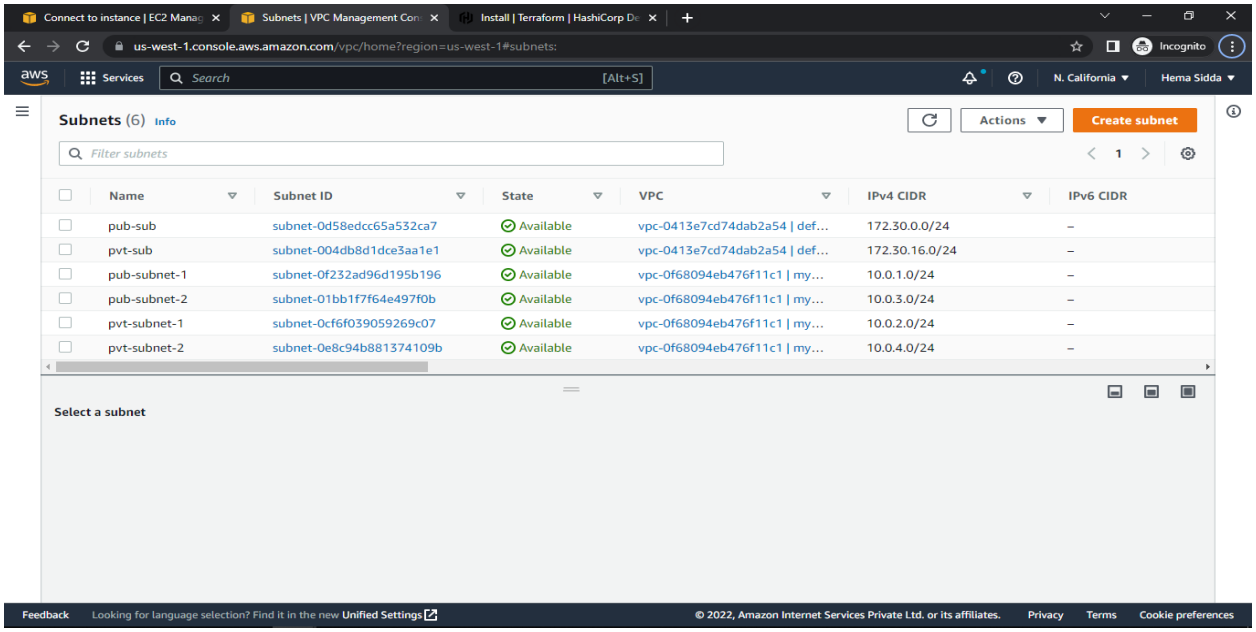


- Now by using single command I can create the resource that commad is <terraform apply --auto-approve>



SHAIK GOUSERABBANI

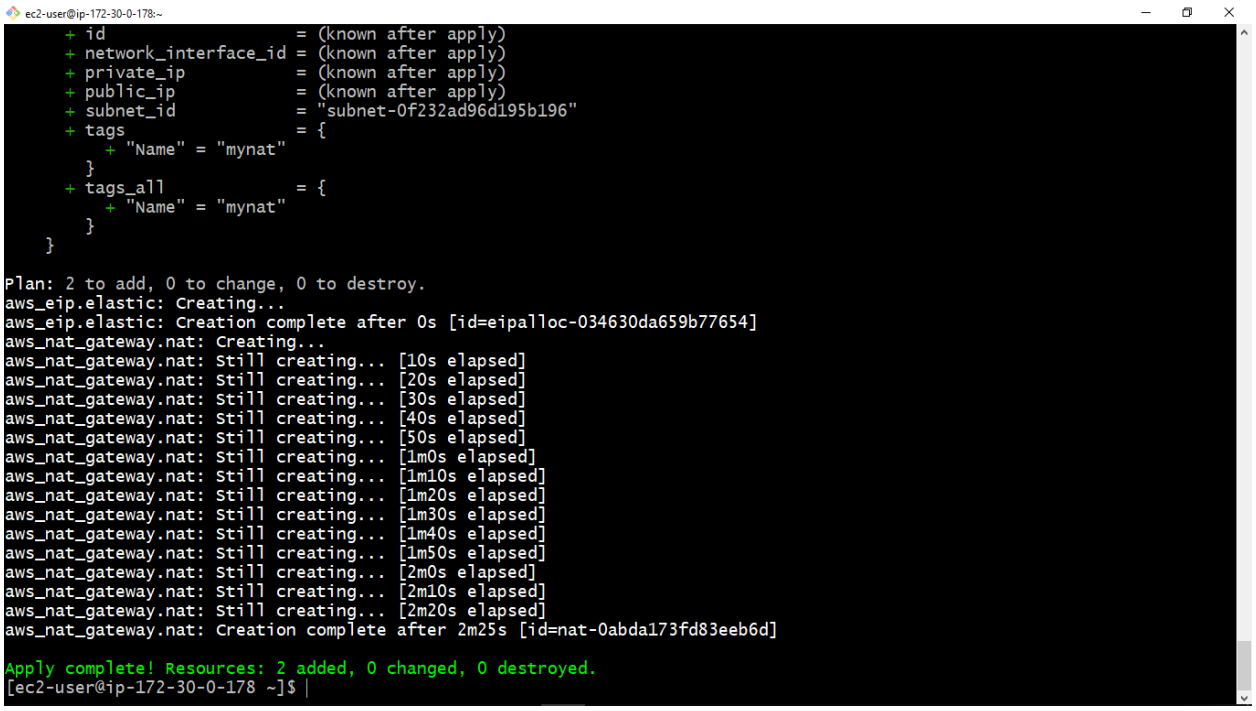
DEPLOY THREE-TIER ARCHITECHTURE IN AWS BY USING TERRAFORM

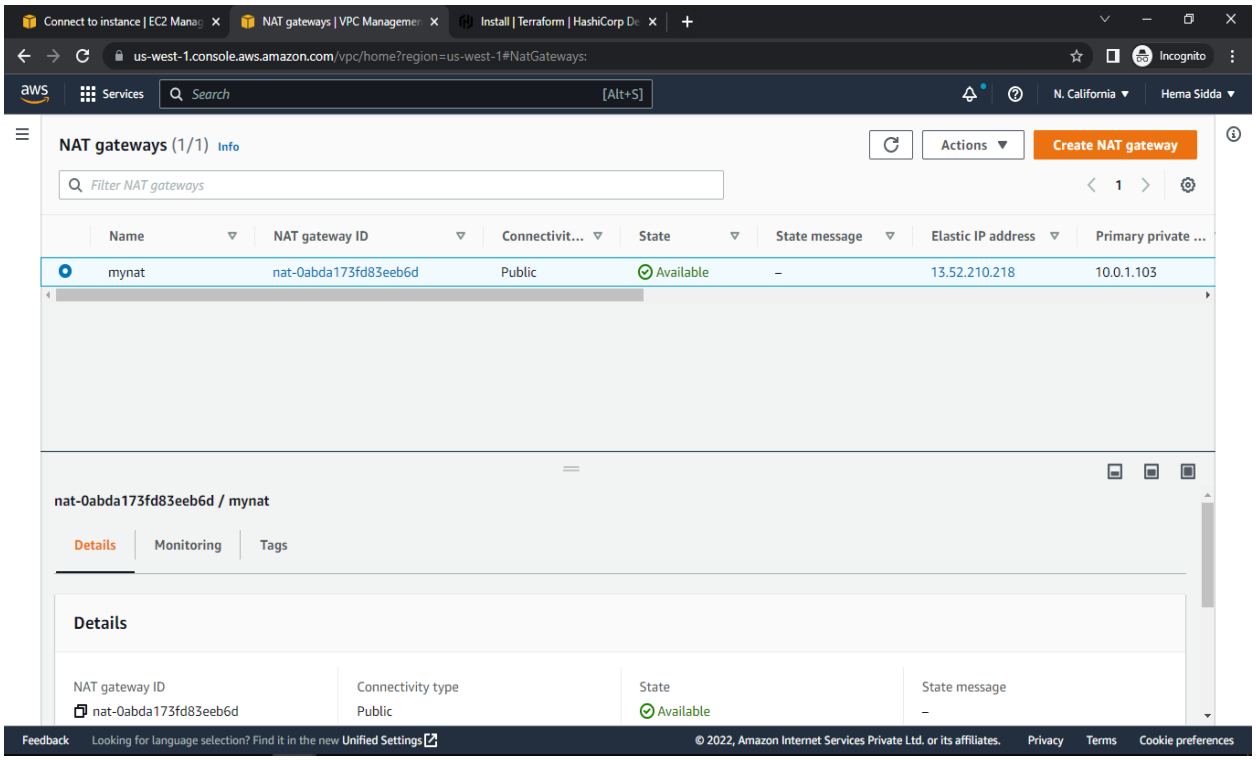


➢ Now create the NAT-gateway for that create a .tf file and enter the yaml script.

SHAIK GOUSERABBANI

DEPLOY THREE-TIER ARCHITECHTURE IN AWS BY USING TERRAFORM





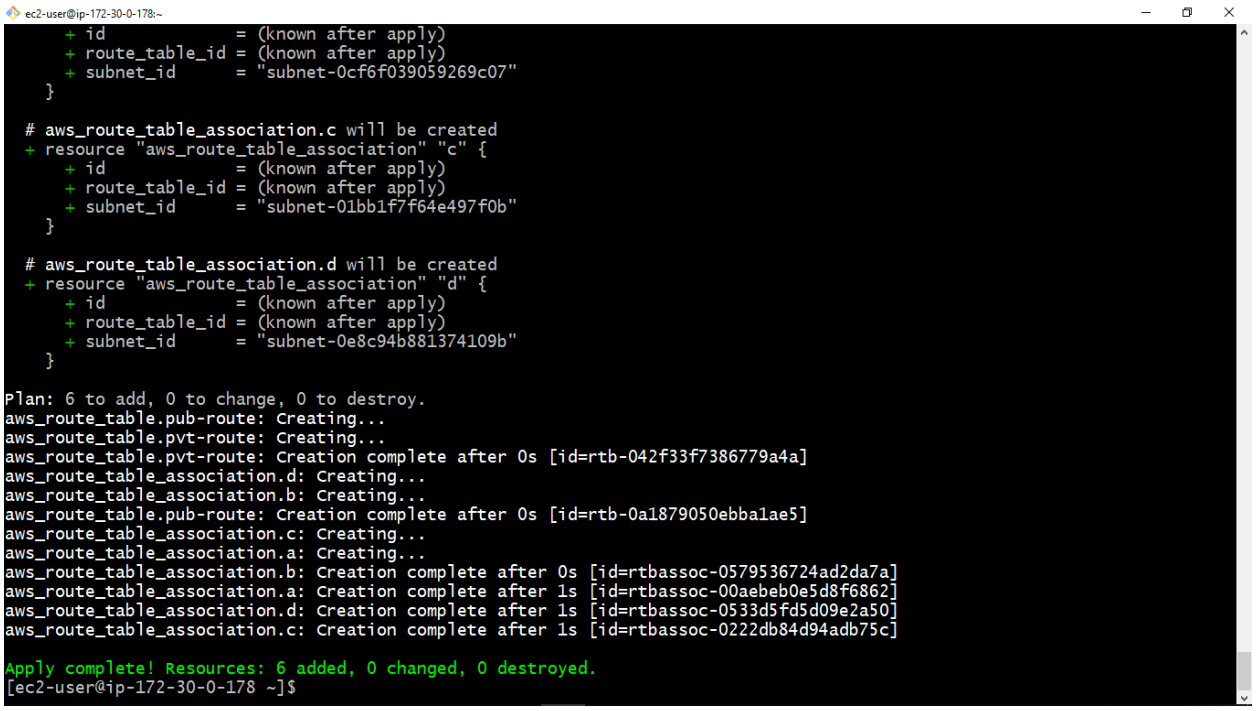
SHAIK GOUSERABBANI

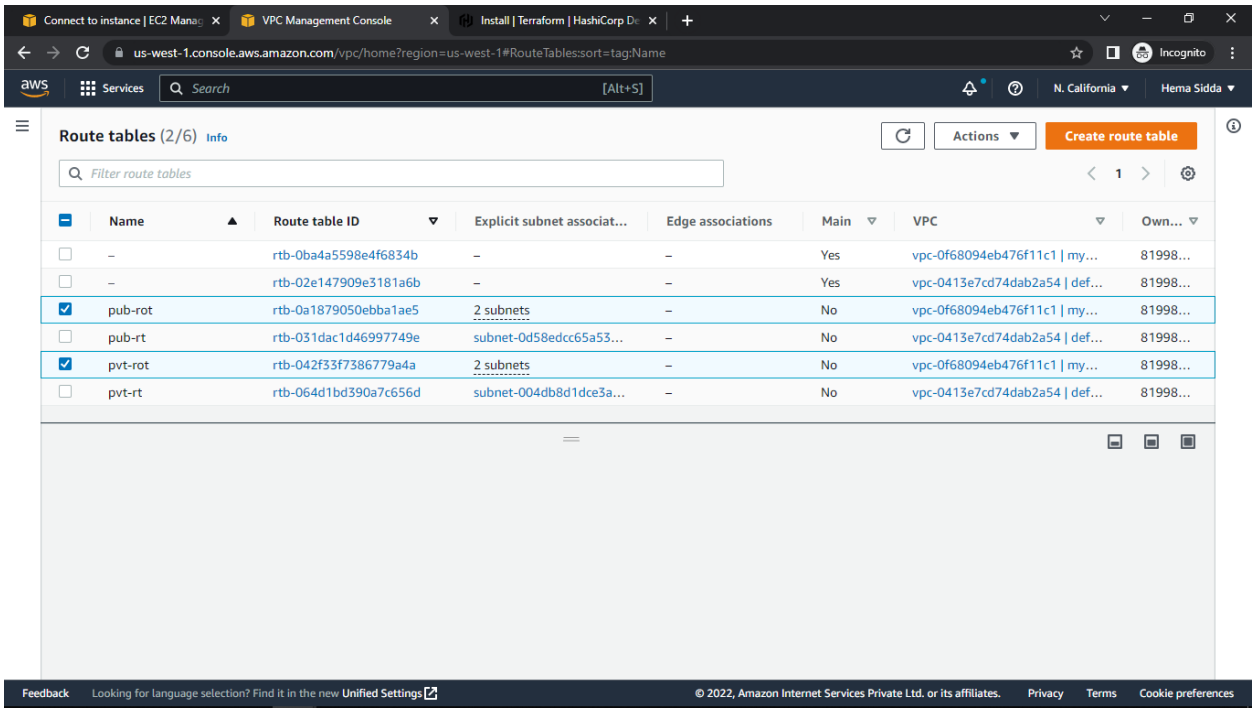
DEPLOY THREE-TIER ARCHITECHTURE IN AWS BY USING TERRAFORM

- Now go to create the route table and associate subnets, Internet gateway and NAT gate way for that create a .tf file by using the vi mode as <vi route.tf>

SHAIK GOUSERABBANI

DEPLOY THREE-TIER ARCHITECHTURE IN AWS BY USING TERRAFORM

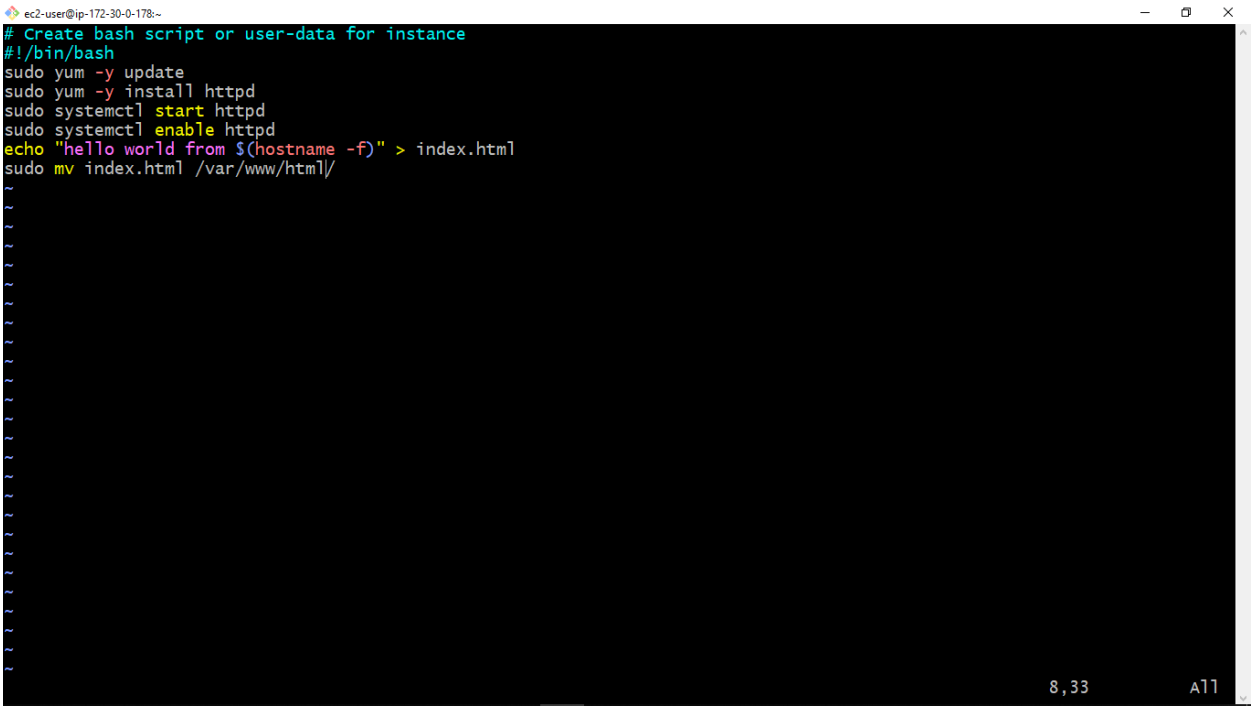




➢ Now create the bash script file by the data.sh

SHAIK GOUSERABBANI

DEPLOY THREE-TIER ARCHITECHTURE IN AWS BY USING TERRAFORM

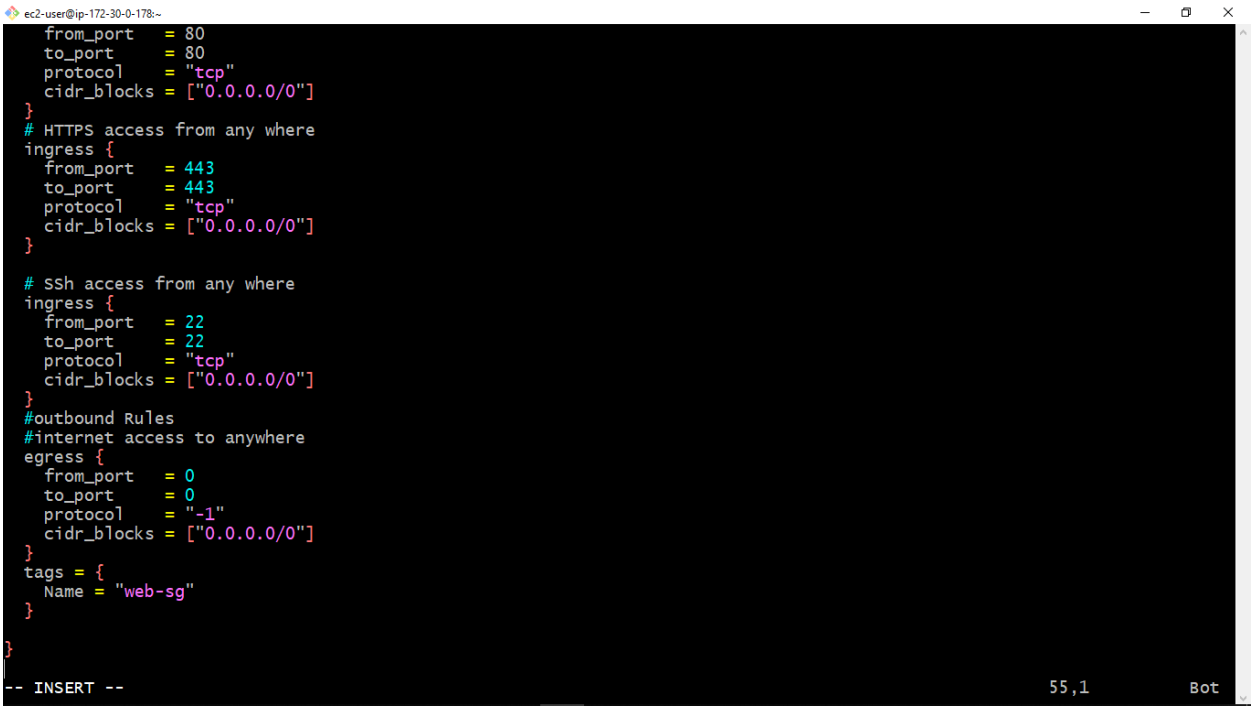


- Now create the public instance by using yaml scripting.

SHAIK GOUSERABBANI

DEPLOY THREE-TIER ARCHITECHTURE IN AWS BY USING TERRAFORM

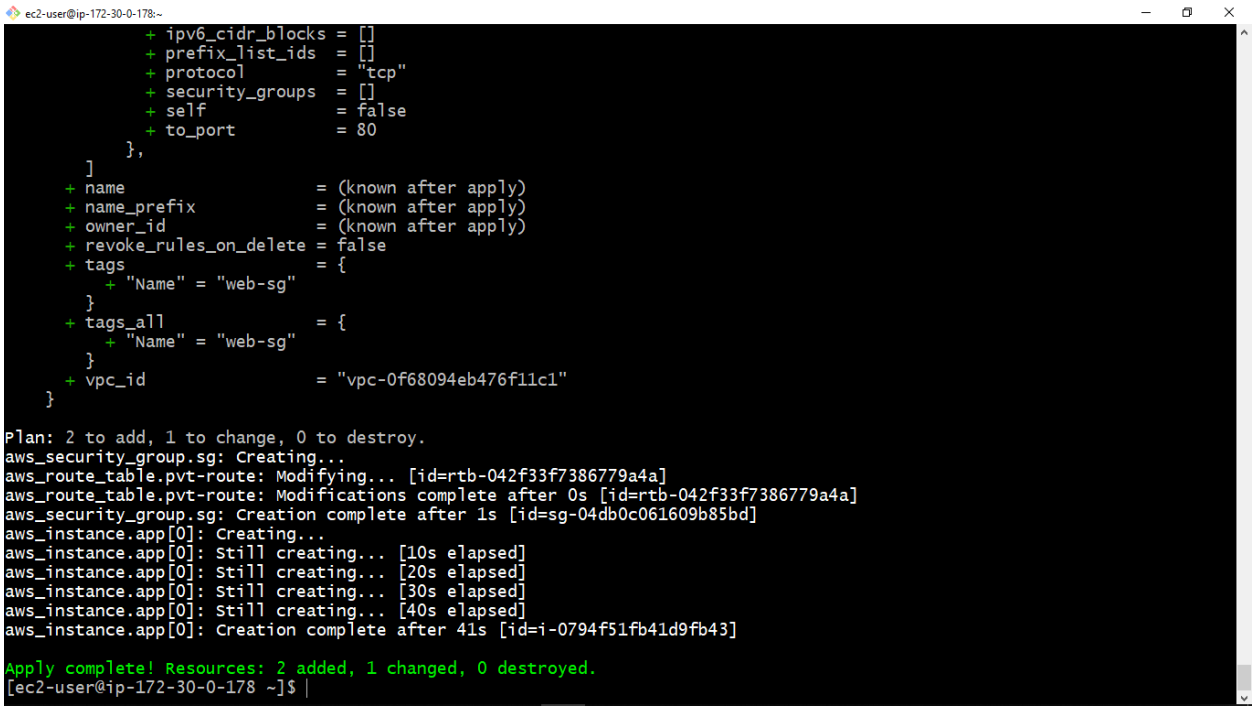
```
ec2-user@ip-172-30-0-178:~
# Create instance
resource "aws_instance" "app" {
  ami                    = "ami-0f5ee8a042c8bfcd5e"
  instance_type          = "t2.micro"
  count                  = 1
  key_name               = "terraform"
  subnet_id              = aws_subnet.pub-sub-1.id
  vpc_security_group_ids = ["${aws_security_group.sg.id}"]
  user_data              = "${file("data.sh")}"
  tags = {
    Name = "webapplication"
  }
}

# Create Security-group
resource "aws_security_group" "sg" {
  vpc_id = aws_vpc.mvpc.id

  #Inbound Rules
  # HTTP acces from anywhere
  ingress {
    from_port   = 80
    to_port     = 80
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }
  # HTTPS access from any where
  ingress {
    from_port   = 443
    to_port     = 443
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }
-- INSERT --                                                    1,1           Top
```

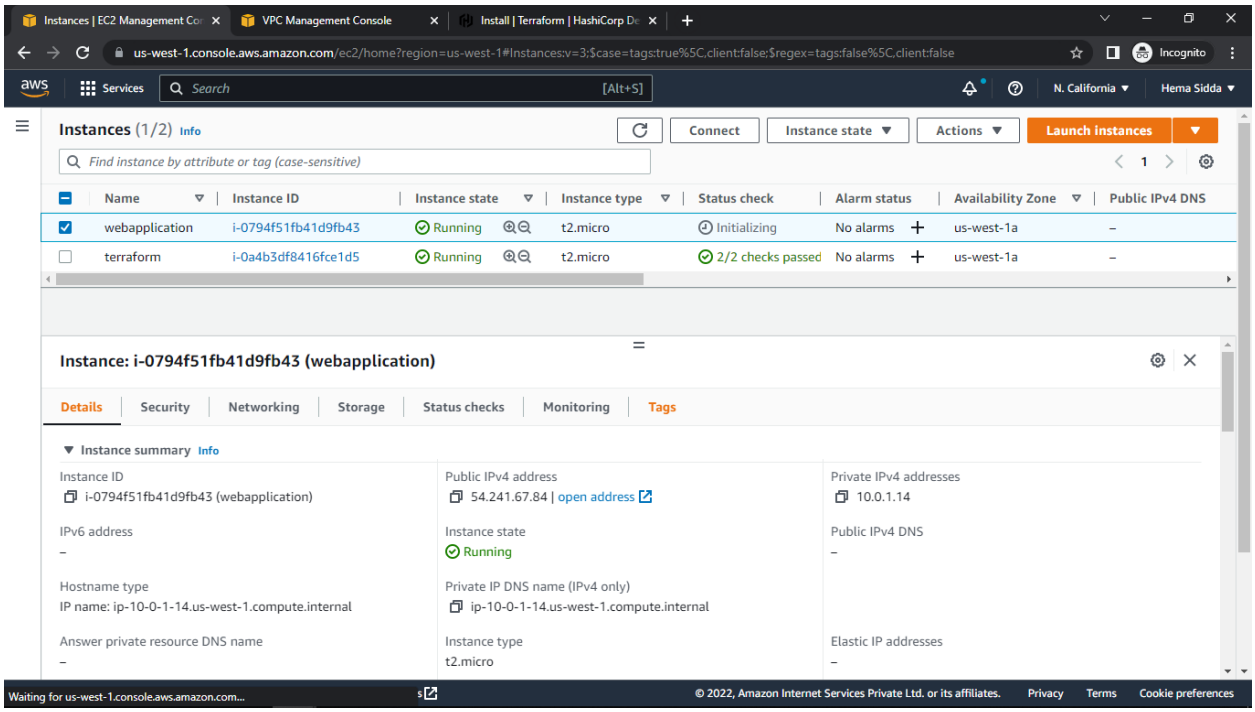


```
ec2-user@ip-172-30-0-178:~
    from_port   = 80
    to_port     = 80
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }
  # HTTPS access from any where
  ingress {
    from_port   = 443
    to_port     = 443
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }

  # SSh 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 anywhere
  egress {
    from_port   = 0
    to_port     = 0
    protocol    = "-1"
    cidr_blocks = ["0.0.0.0/0"]
  }
  tags = {
    Name = "web-sg"
  }

}
-- INSERT --                                                    55,1          Bot
```

SHAIK GOUSERABBANI

DEPLOY THREE-TIER ARCHITECHTURE IN AWS BY USING TERRAFORM





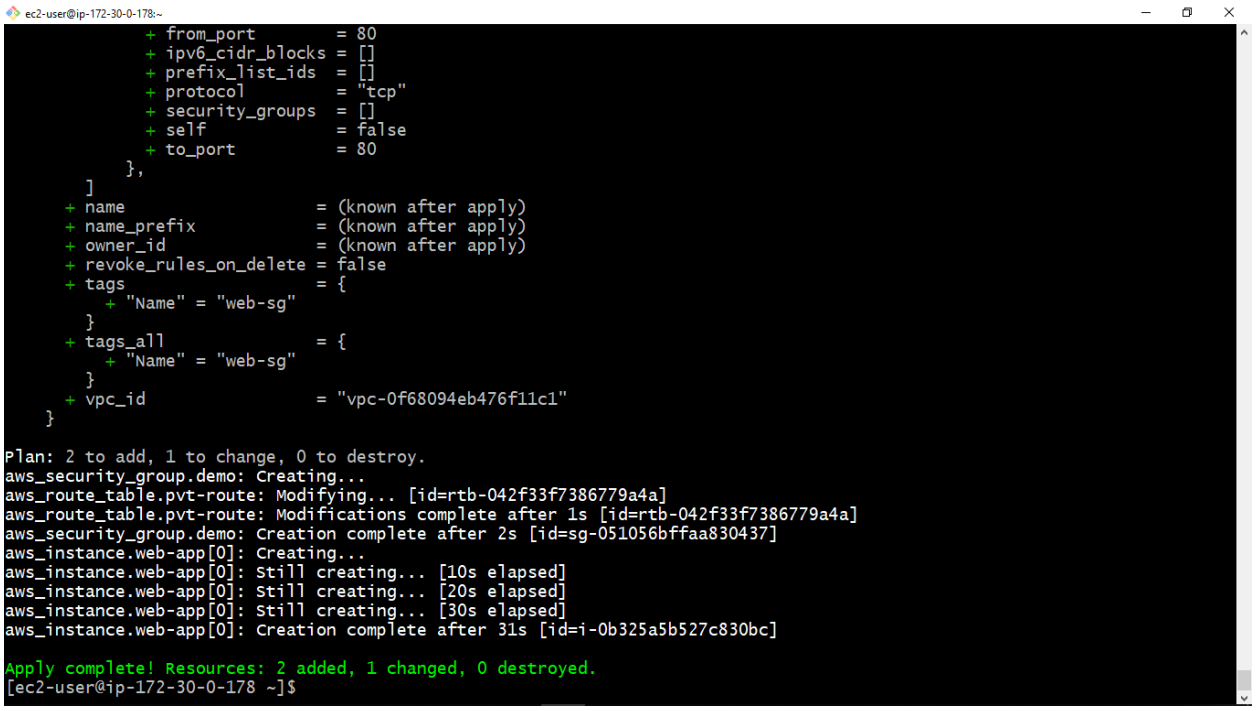
SHAIK GOUSERABBANI

DEPLOY THREE-TIER ARCHITECHTURE IN AWS BY USING TERRAFORM

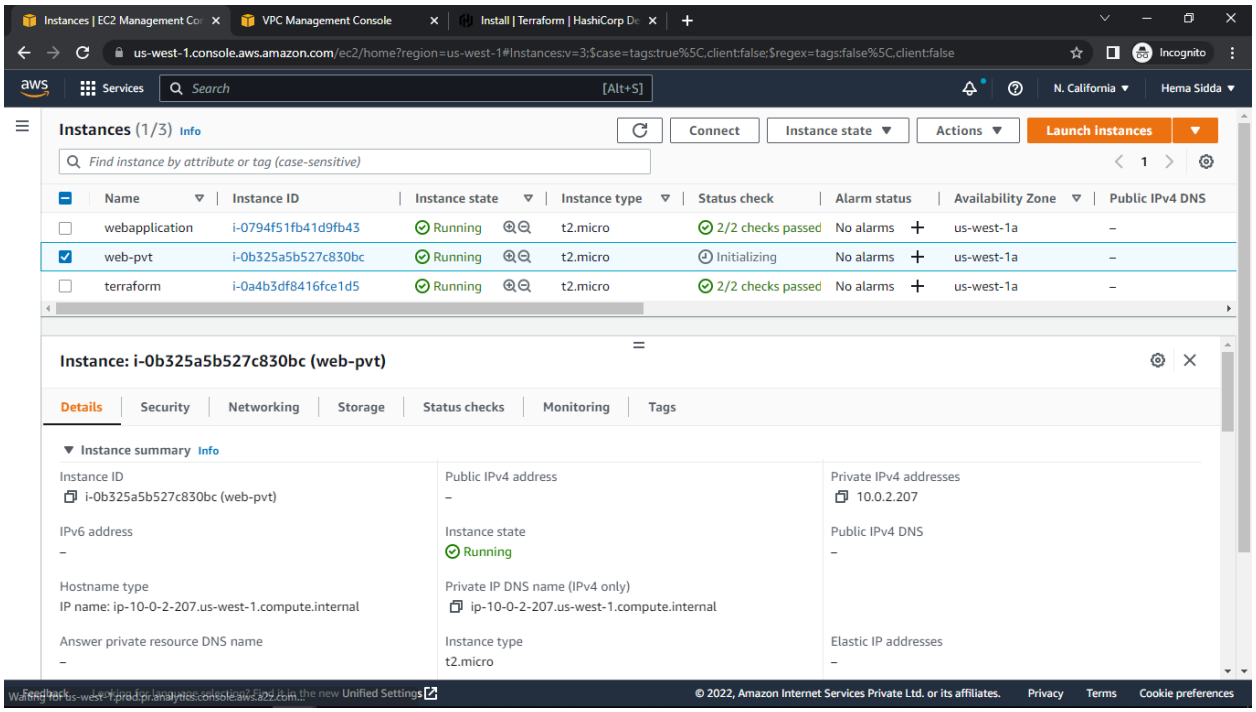
- Now create the pvt instance by using the yaml script of pvt instance .

SHAIK GOUSERABBANI

DEPLOY THREE-TIER ARCHITECHTURE IN AWS BY USING TERRAFORM



```
ec2-user@ip-172-30-0-178:~
            + from_port        = 80
            + ipv6_cidr_blocks = []
            + prefix_list_ids  = []
            + protocol         = "tcp"
            + security_groups  = []
            + self             = false
            + to_port          = 80
          },
      ]
    + name                   = (known after apply)
    + name_prefix            = (known after apply)
    + owner_id               = (known after apply)
    + revoke_rules_on_delete = false
    + tags                   = {
        + "Name" = "web-sg"
      }
    + tags_all               = {
        + "Name" = "web-sg"
      }
    + vpc_id                 = "vpc-0f68094eb476f11c1"
  }

Plan: 2 to add, 1 to change, 0 to destroy.
aws_security_group.demo: Creating...
aws_route_table.pvt-route: Modifying... [id=rtb-042f33f7386779a4a]
aws_route_table.pvt-route: Modifications complete after 1s [id=rtb-042f33f7386779a4a]
aws_security_group.demo: Creation complete after 2s [id=sg-051056bffaa830437]
aws_instance.web-app[0]: Creating...
aws_instance.web-app[0]: Still creating... [10s elapsed]
aws_instance.web-app[0]: Still creating... [20s elapsed]
aws_instance.web-app[0]: Still creating... [30s elapsed]
aws_instance.web-app[0]: Creation complete after 31s [id=i-0b325a5b527c830bc]

Apply complete! Resources: 2 added, 1 changed, 0 destroyed.
[ec2-user@ip-172-30-0-178 ~]$
```



SHAIK GOUSERABBANI

DEPLOY THREE-TIER ARCHITECHTURE IN AWS BY USING TERRAFORM

- Now create the RDS(relational database) service by using yaml scripting.

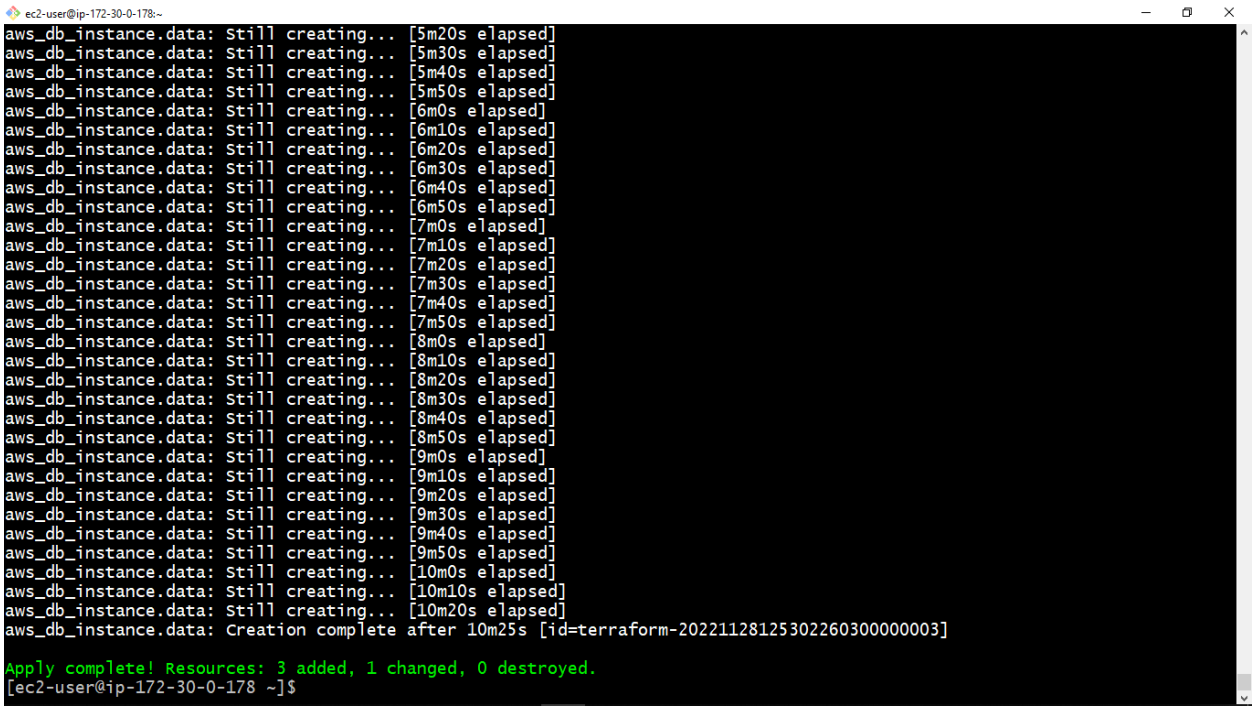
ec2-user@ip-172-30-0-178:~

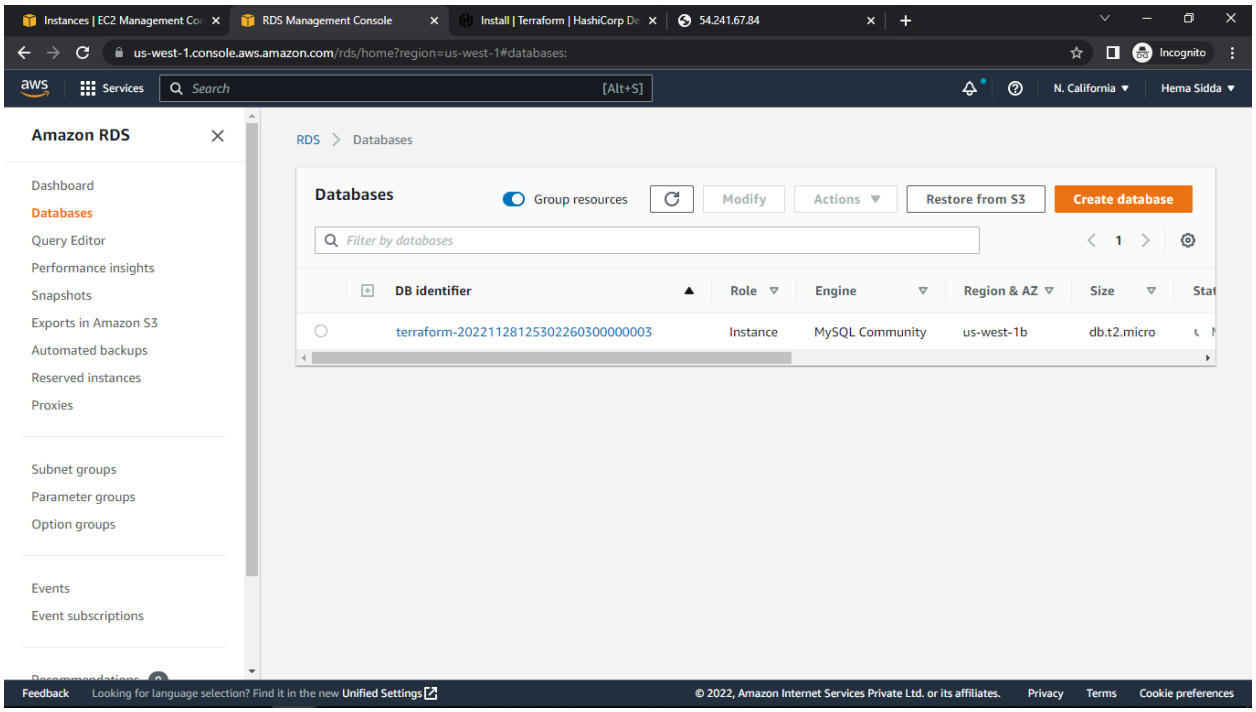
```
# Create subnet groups
resource "aws_db_subnet_group" "db-sub" {
subnet_ids = [aws_subnet.pvt-sub-1.id, aws_subnet.pvt-sub-2.id]
tags = {
        Name = "dbsubnet"
        }
}
# Create the data-base
resource "aws_db_instance" "data" {
 allocated_storage = "10"
        db_subnet_group_name = aws_db_subnet_group.db-sub.id
        db_name = "mydata"
        engine = "mysql"
        engine_version = "8.0.28"
    instance_class = "db.t2.micro"
    multi_az = "true"
    username = "admin"
    password = "admin1230"
    vpc_security_group_ids = ["${aws_security_group.db-sg.id}"]
    skip_final_snapshot = "true"

}

# Create the security groups
resource "aws_security_group" "db-sg" {
        vpc_id = aws_vpc.mvpc.id

  ingress {
  from_port = 3306
  to_port = 3306
  protocol = "tcp"
  security_groups = [aws_security_group.demo.id]
  }
-- INSERT --                                                    1,1           Top
```

ec2-user@ip-172-30-0-178:~

```
        engine = "mysql"
        engine_version = "8.0.28"
    instance_class = "db.t2.micro"
    multi_az = "true"
    username = "admin"
    password = "admin1230"
    vpc_security_group_ids = ["${aws_security_group.db-sg.id}"]
    skip_final_snapshot = "true"

}

# Create the security groups
resource "aws_security_group" "db-sg" {
        vpc_id = aws_vpc.mvpc.id

  ingress {
  from_port = 3306
  to_port = 3306
  protocol = "tcp"
  security_groups = [aws_security_group.demo.id]
  }

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

  }
  tags = {
  Name = "database-sg"
  }
}
-- INSERT --                                                    46,1          Bot
```

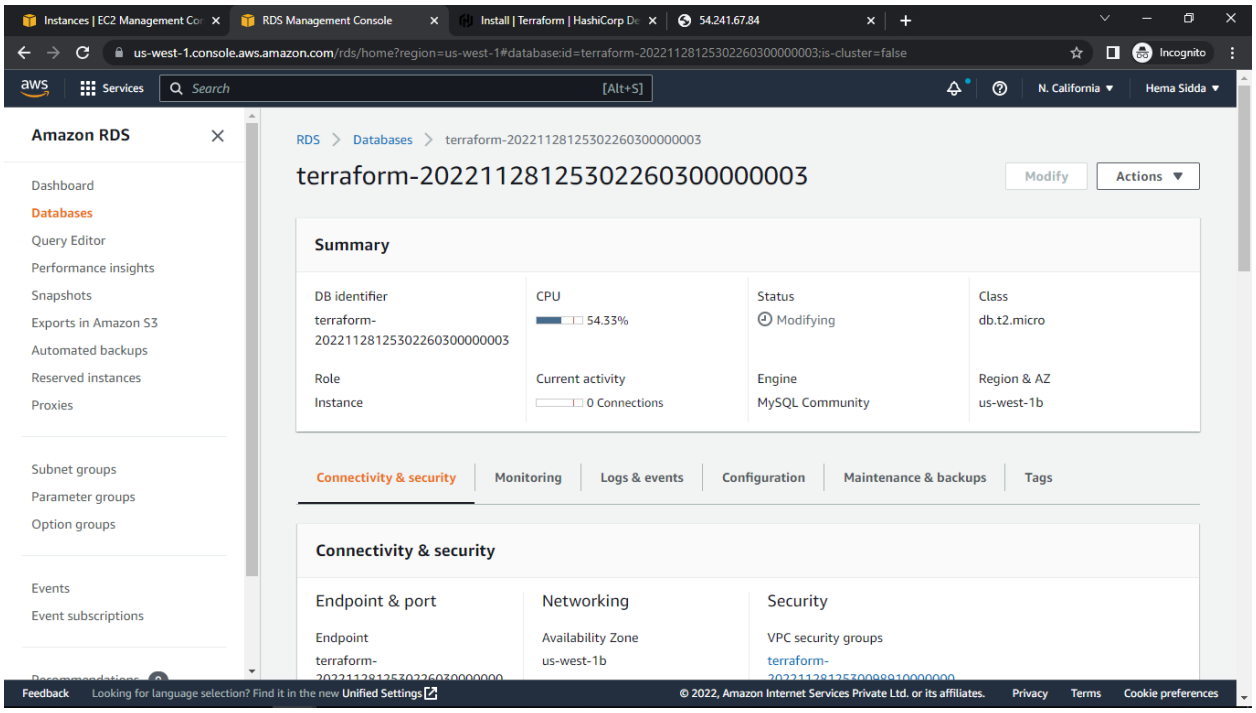
DEPLOY THREE-TIER ARCHITECHTURE IN AWS BY USING TERRAFORM





SHAIK GOUSERABBANI

DEPLOY THREE-TIER ARCHITECHTURE IN AWS BY USING TERRAFORM

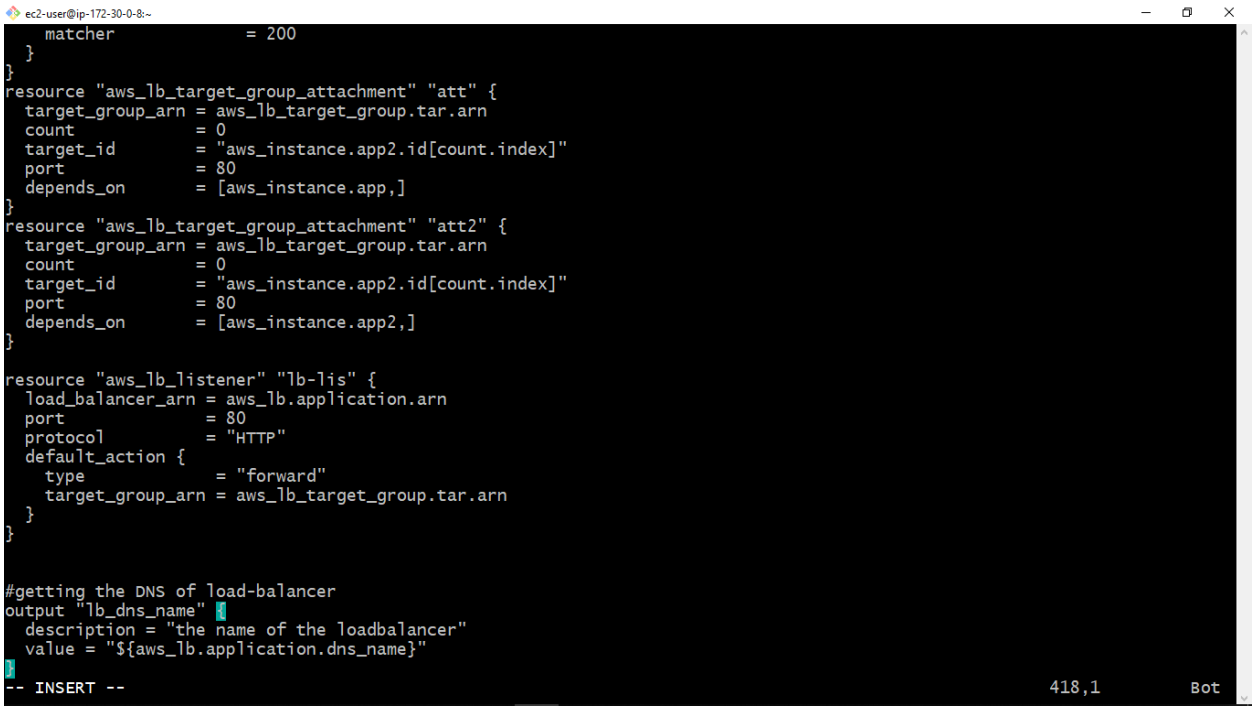


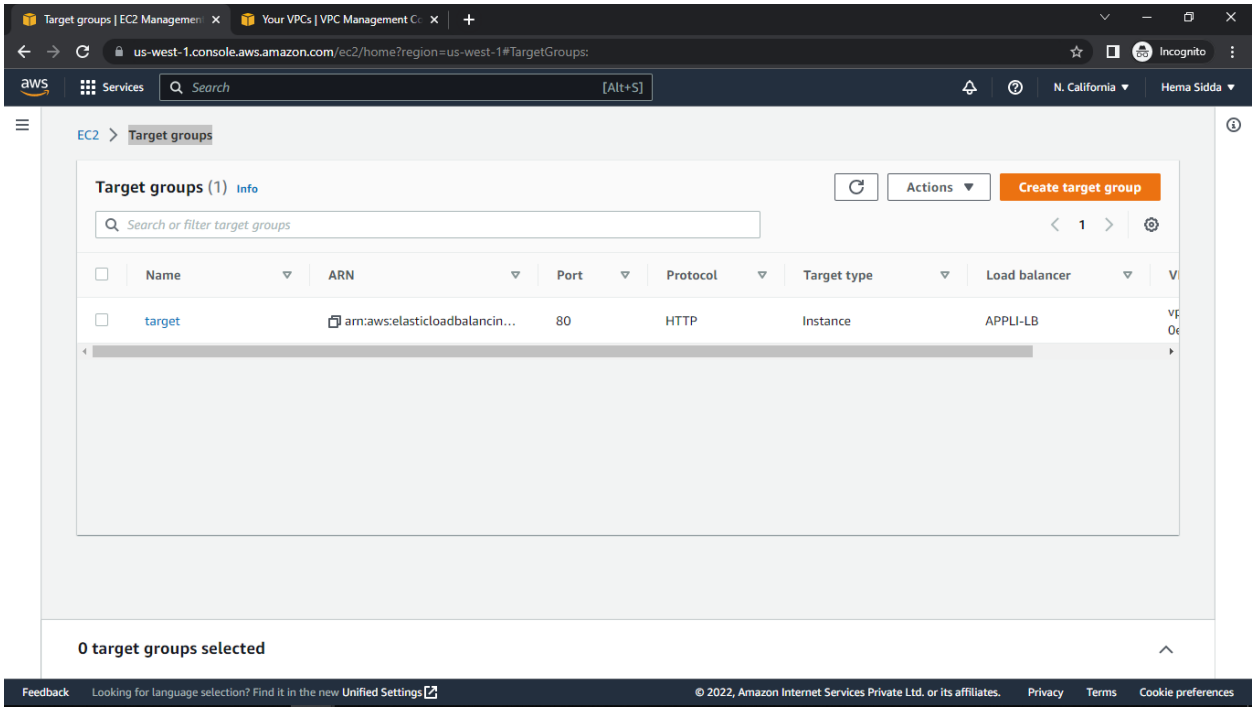
➢ Now create the load-balancer by using yaml script.

```
ec2-user@ip-172-30-0-8:~
# Create load_balancer
resource "aws_lb" "application" {
  internal           = false
  name               = "APPLI-LB"
  load_balancer_type = "application"
  security_groups    = [aws_security_group.sg.id]
  subnets            = [aws_subnet.pub-sub-1.id, aws_subnet.pub-sub-2.id]
}

resource "aws_lb_target_group" "tar" {
  name     = "target"
  port     = 80
  protocol = "HTTP"
  vpc_id   = aws_vpc.mvpc.id
  health_check {
    healthy_threshold   = 2
    unhealthy_threshold = 2
    timeout             = 3
    interval            = 30
    protocol            = "HTTP"
    port                = 80
    path                = "/ping"
    matcher             = 200
  }
}
resource "aws_lb_target_group_attachment" "att" {
  target_group_arn = aws_lb_target_group.tar.arn
  count            = 0
  target_id        = "aws_instance.app2.id[count.index]"
  port             = 80
  depends_on       = [aws_instance.app,]
}
resource "aws_lb_target_group_attachment" "att2" {
  target_group_arn = aws_lb_target_group.tar.arn
-- INSERT --                                                    391,1         94%
```

SHAIK GOUSERABBANI

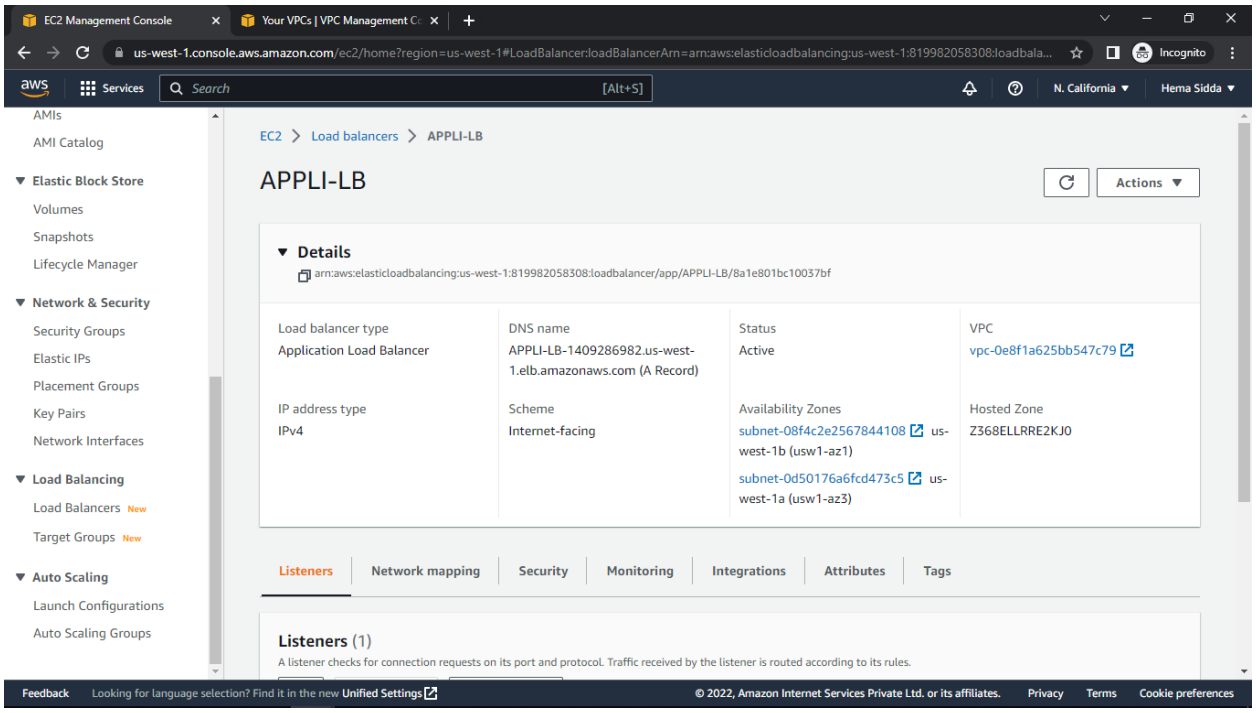
DEPLOY THREE-TIER ARCHITECHTURE IN AWS BY USING TERRAFORM

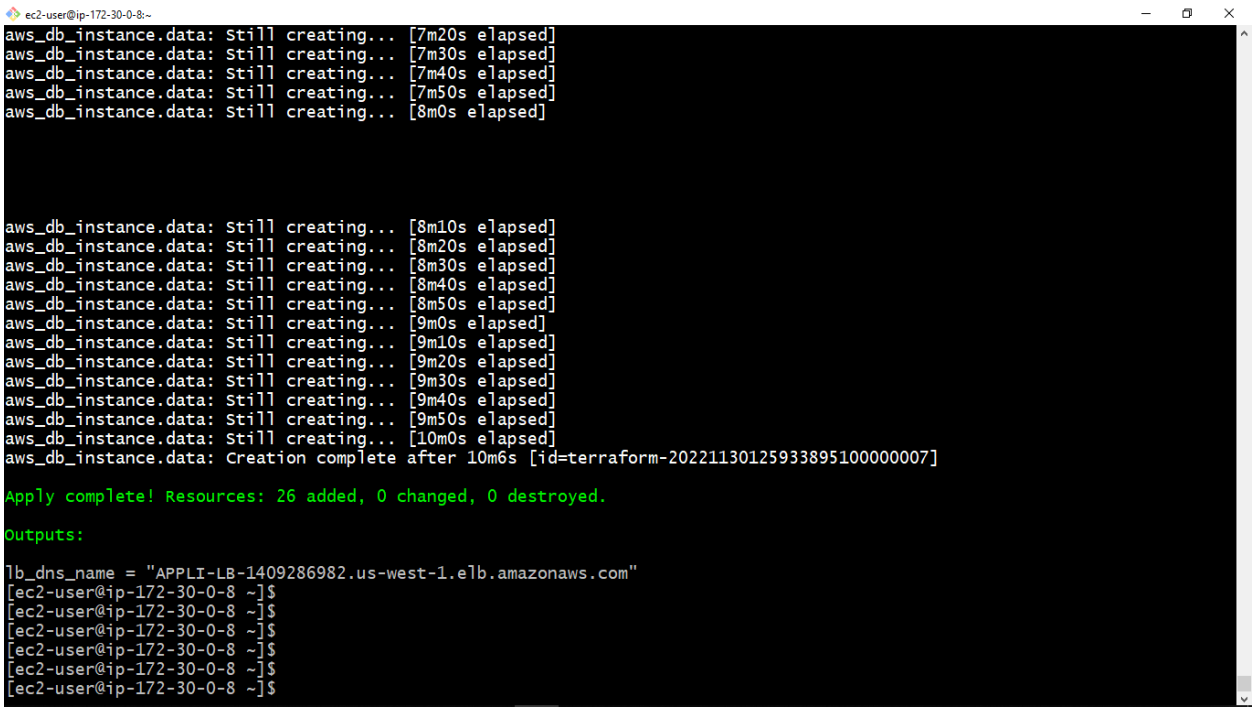




SHAIK GOUSERABBANI

DEPLOY THREE-TIER ARCHITECHTURE IN AWS BY USING TERRAFORM

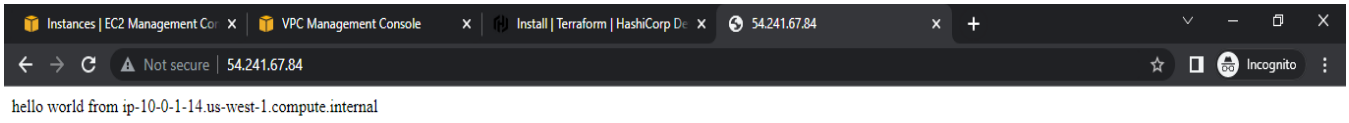




SHAIK GOUSERABBANI

DEPLOY THREE-TIER ARCHITECHTURE IN AWS BY USING TERRAFORM

- Now go to EC2 instances and select the public ip(ipv4) and browse it on Google.



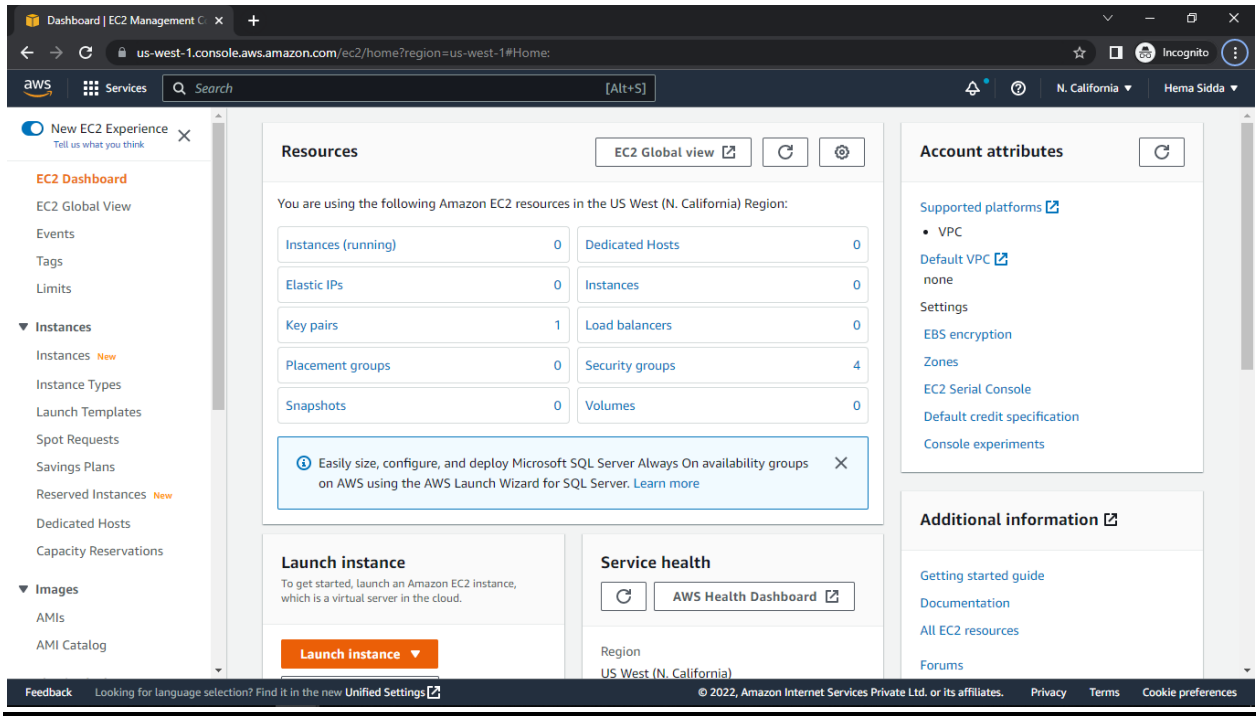
hello world from ip-10-0-1-14.us-west-1.compute.internal

SHAIK GOUSERABBANI

DEPLOY THREE-TIER ARCHITECHTURE IN AWS BY USING TERRAFORM

METHOD-2

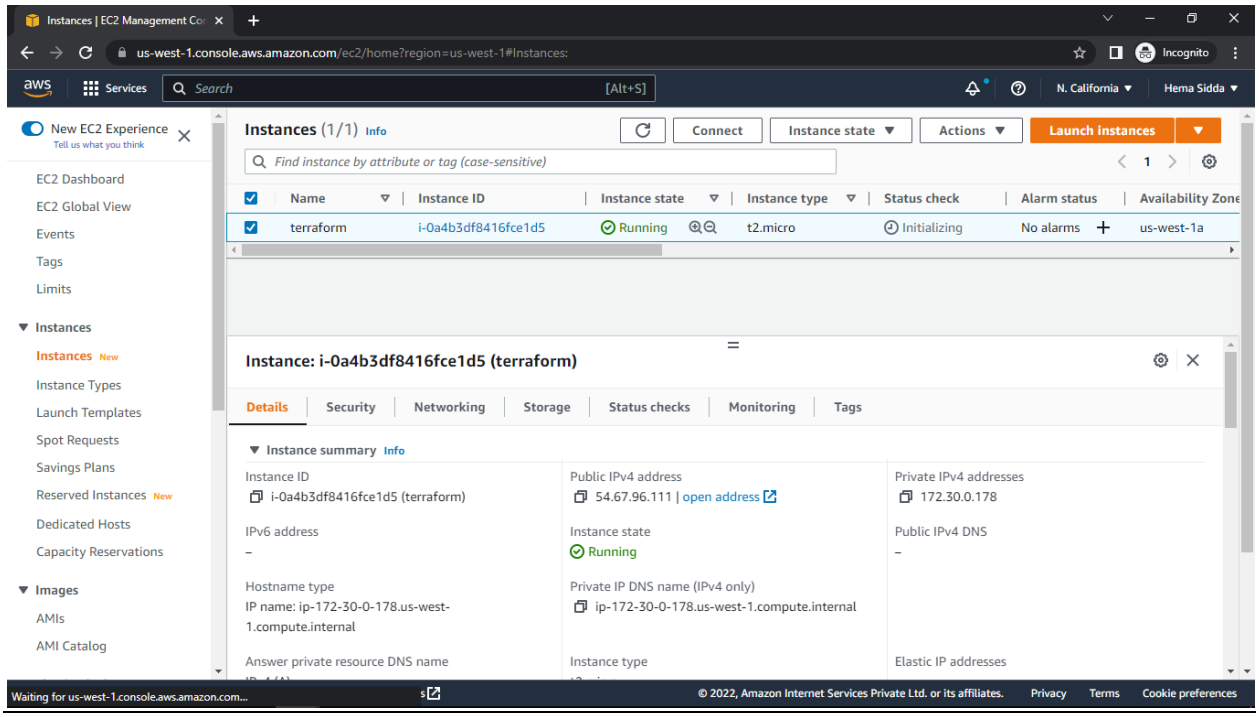
- First open the AWS account by using credentials and go with EC2 instances.



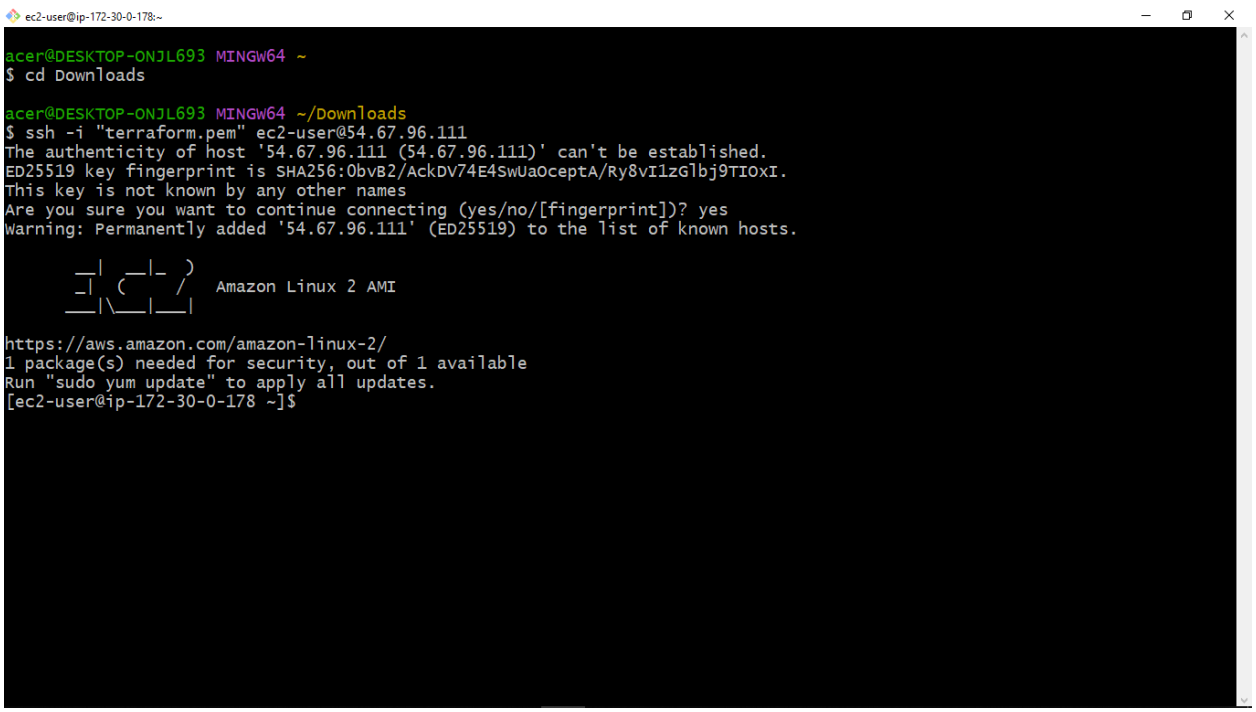
- Now create or launch the server by selecting AMI's and key-pairs and also give the port ranges like SSH(22) and HTTP(80).

SHAIK GOUSERABBANI

DEPLOY THREE-TIER ARCHITECHTURE IN AWS BY USING TERRAFORM



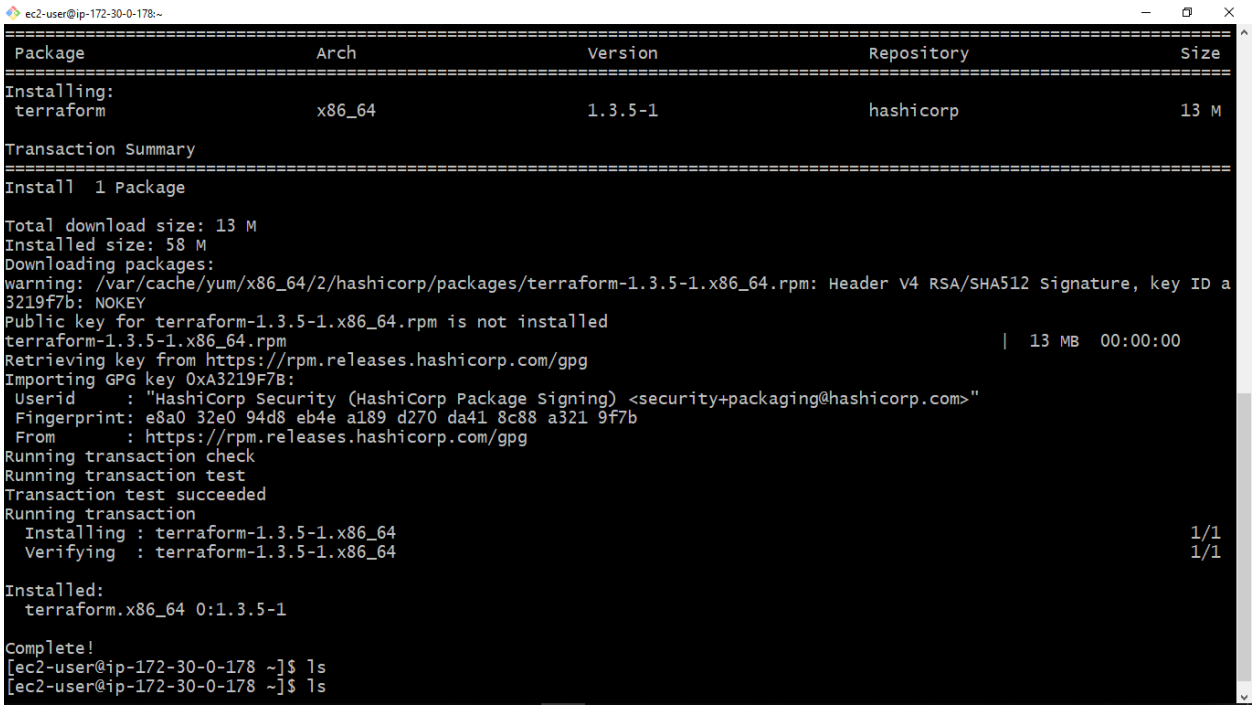
➢ Now connect the launched server through GIT-Bash terminal.



SHAIK GOUSERABBANI

DEPLOY THREE-TIER ARCHITECHTURE IN AWS BY USING TERRAFORM

- Now install the terraform by using commands those are available in Google browse it by the name <Terraform download>.

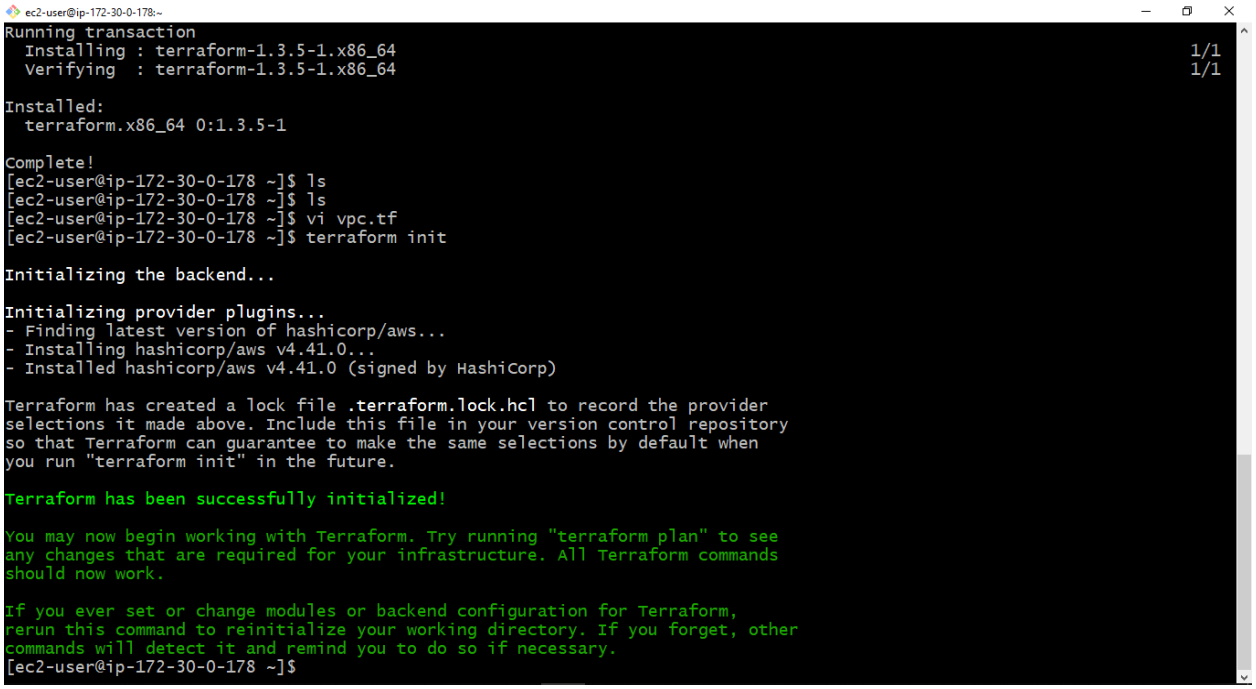


- Now create vpc by using yaml scripting for that create a file as <vi file.tf> and write the yaml script to create vpc and save it by using ":wq".

SHAIK GOUSERABBANI

DEPLOY THREE-TIER ARCHITECHTURE IN AWS BY USING TERRAFORM

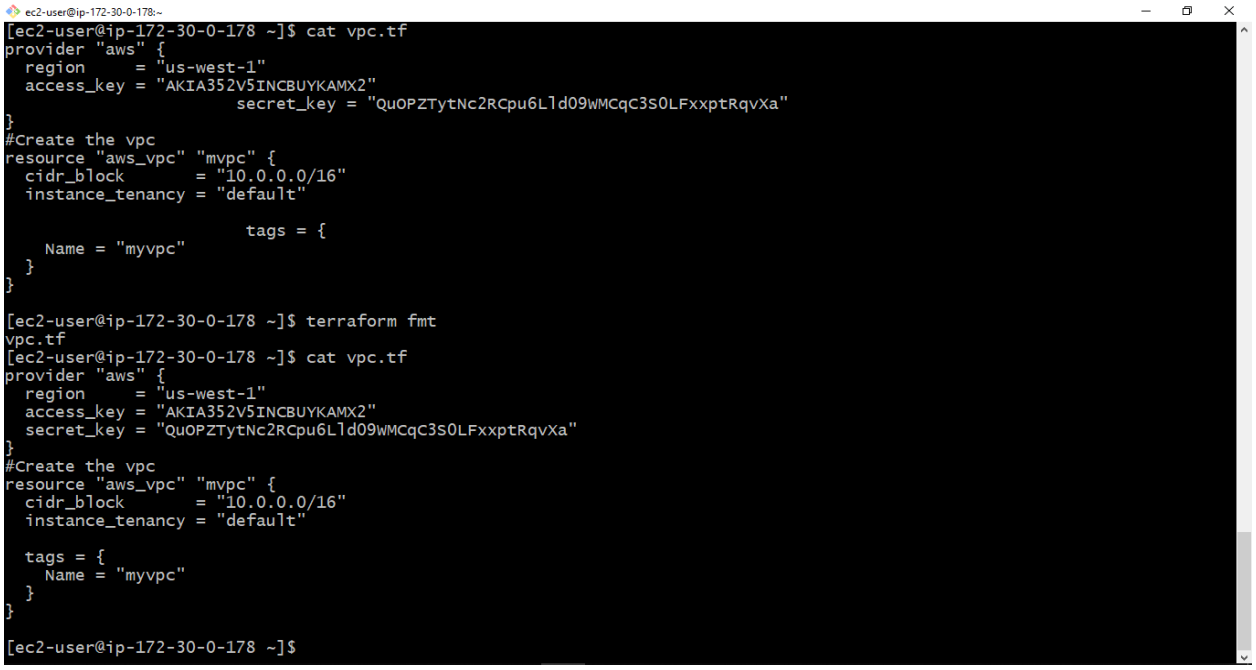
- Now change the environment into terraform mode by using command as <terraform init>.



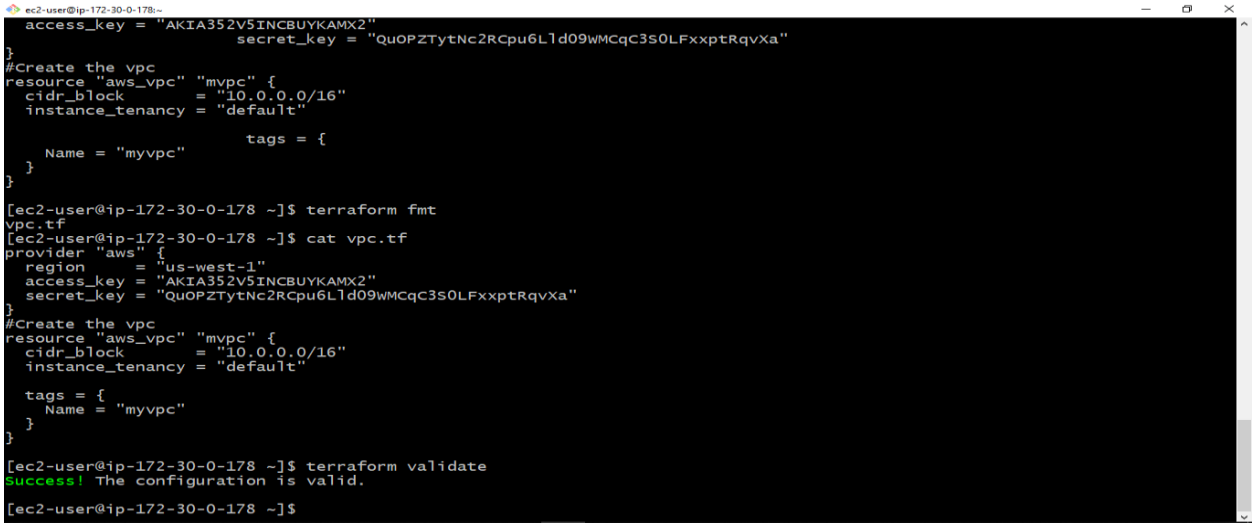
SHAIK GOUSERABBANI

DEPLOY THREE-TIER ARCHITECHTURE IN AWS BY USING TERRAFORM

- Now set the yaml script alignment by using command as <terraform fmt>.

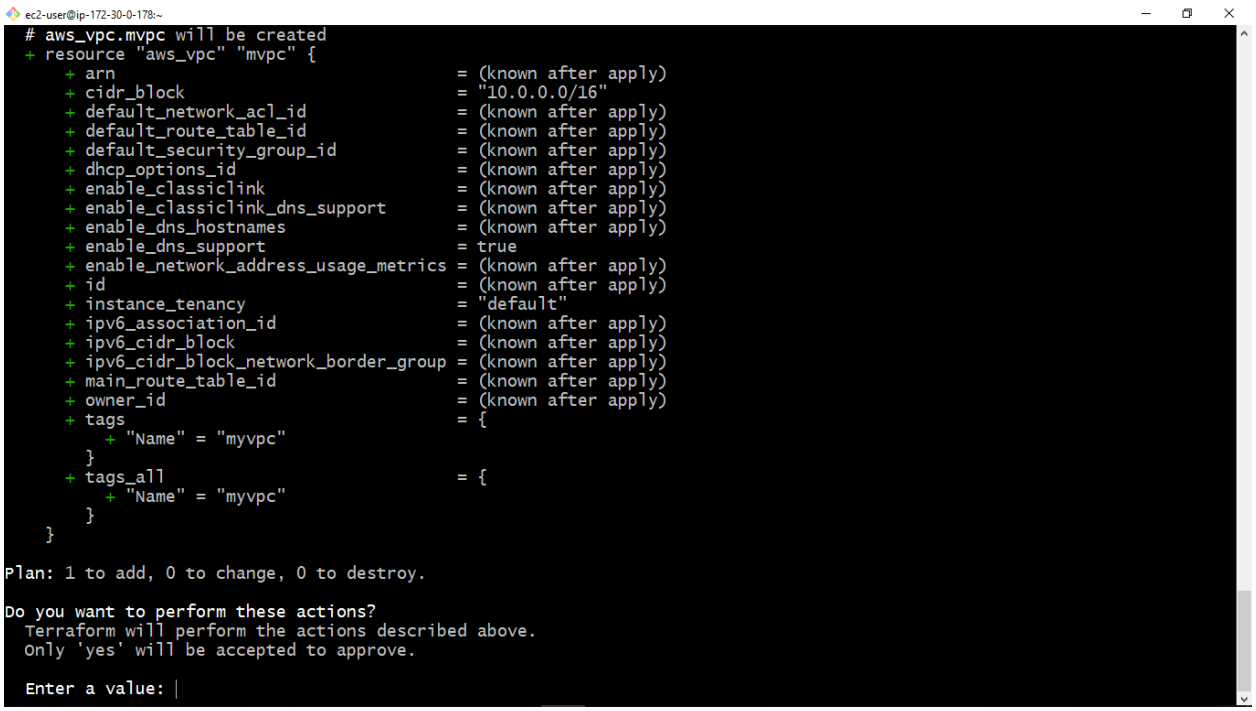


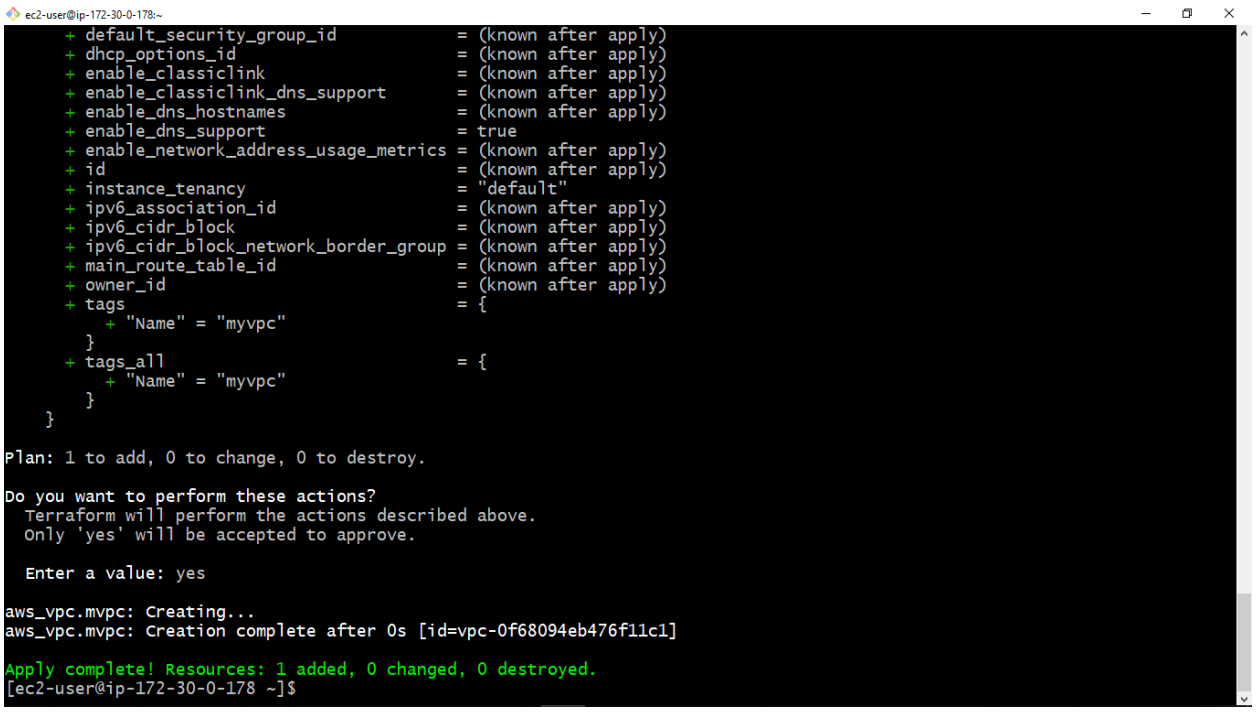
- Now validate the script for correcting the spelling mistakes by using command as <terraform validate>.



DEPLOY THREE-TIER ARCHITECHTURE IN AWS BY USING TERRAFORM

➢ Now create the vpc by using command as <terraform apply>.

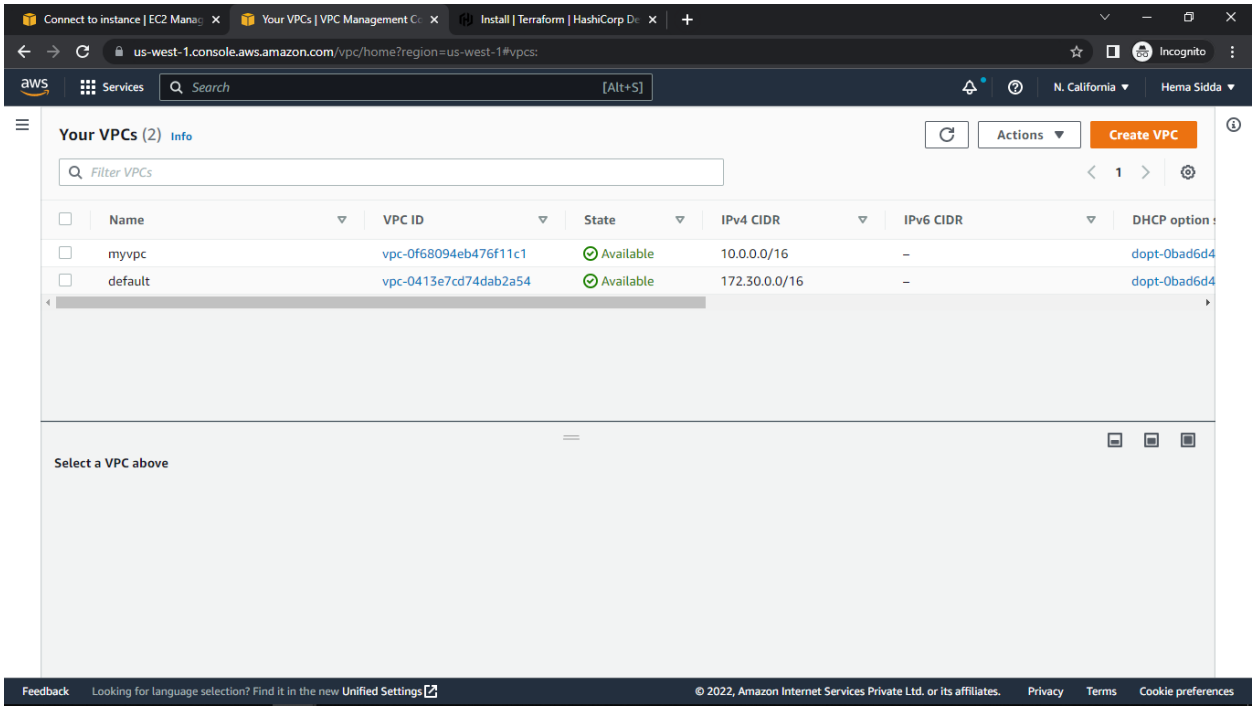




SHAIK GOUSERABBANI

DEPLOY THREE-TIER ARCHITECHTURE IN AWS BY USING TERRAFORM

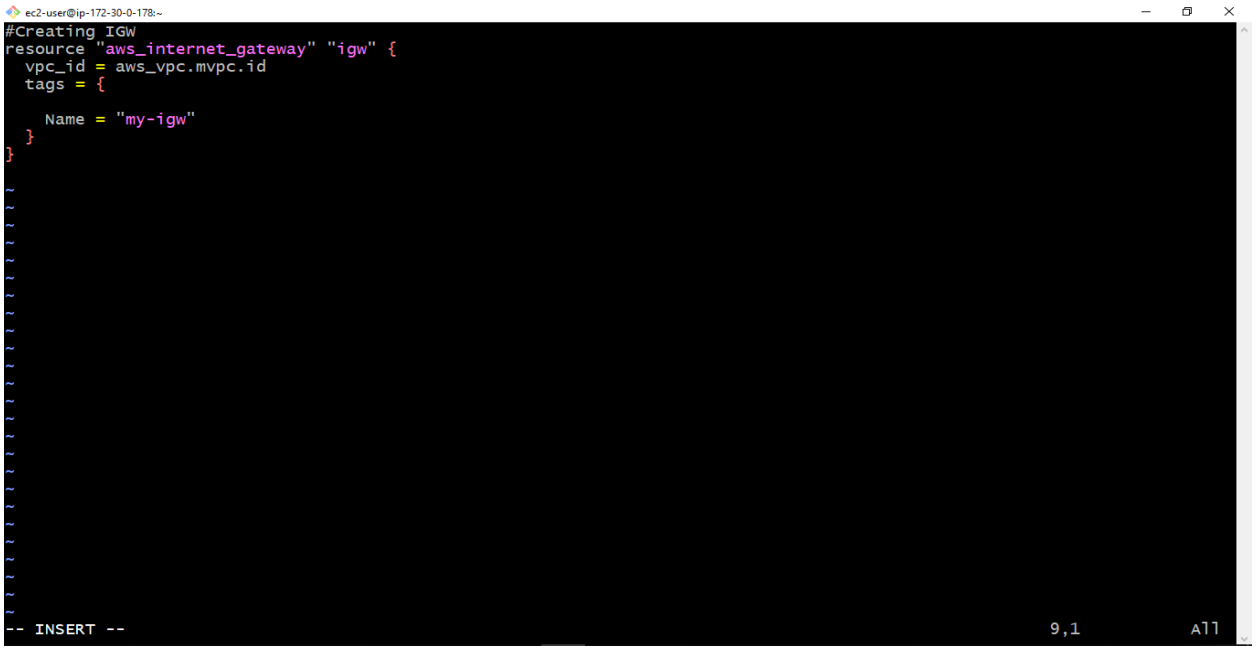
- Now go to your AWS account and open vpc services and check your vpc created or not.

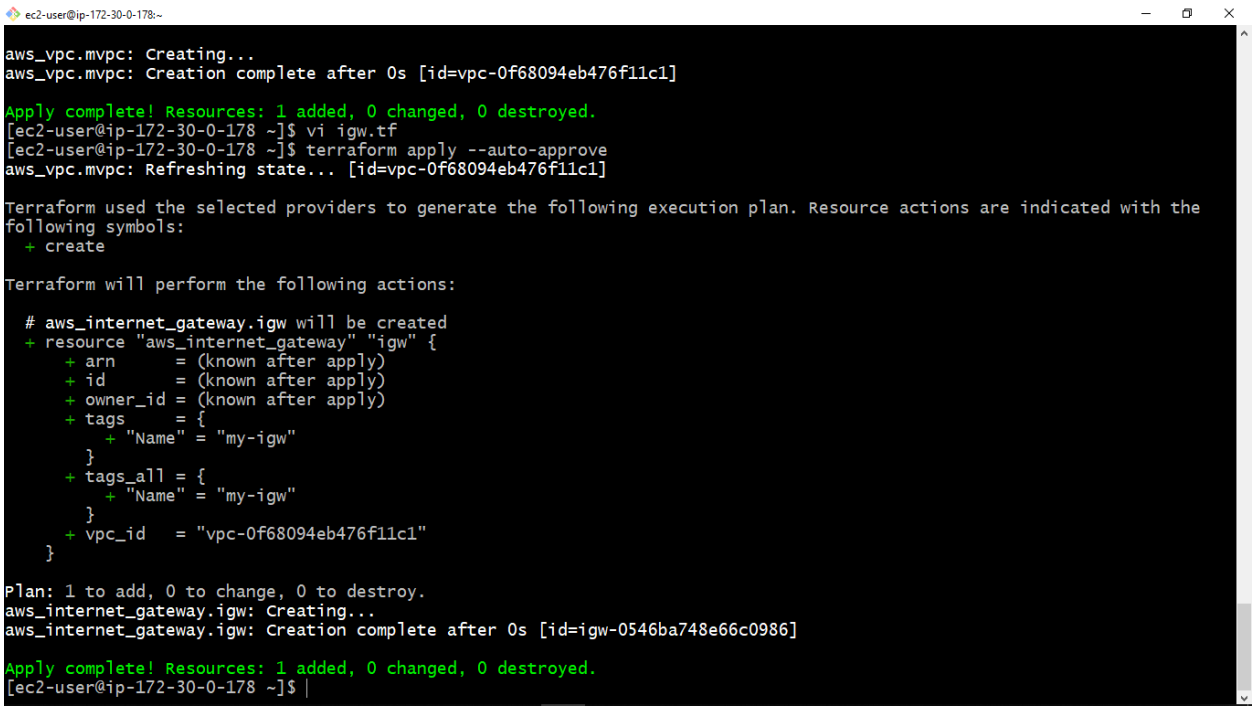


- Now create the IGW and attach it to created vpc for that create a file as <vi igw.tf>

SHAIK GOUSERABBANI

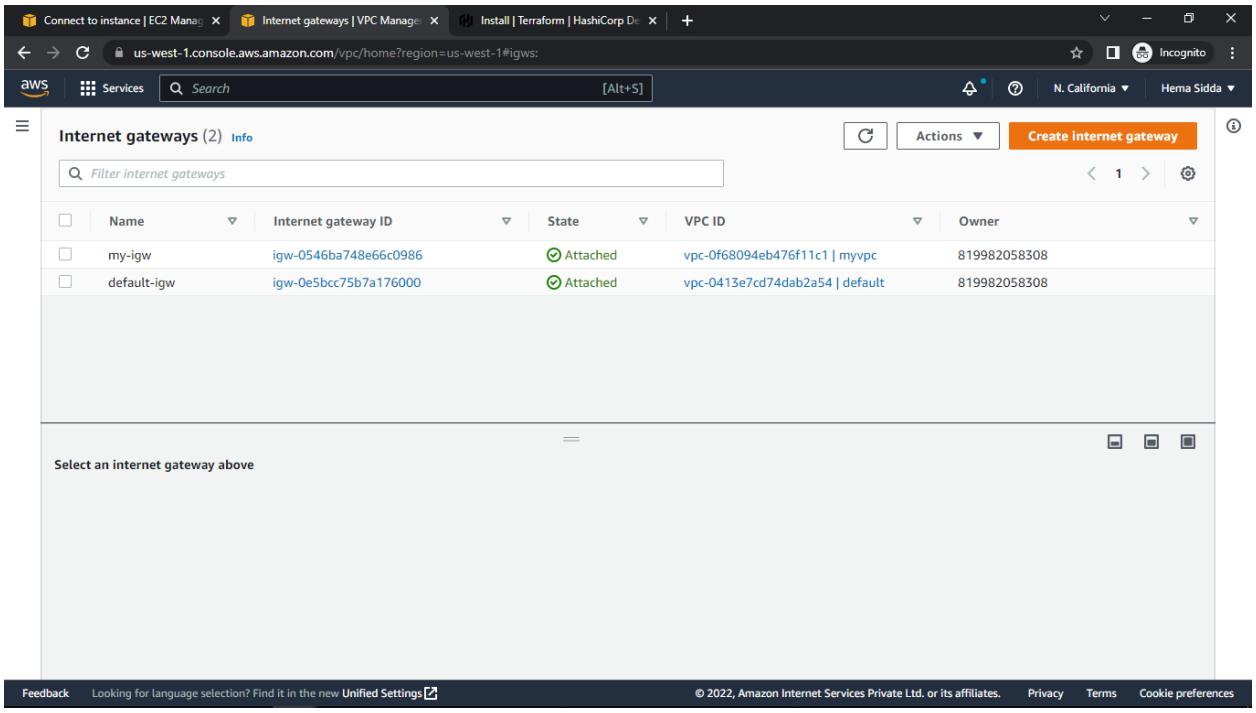
DEPLOY THREE-TIER ARCHITECHTURE IN AWS BY USING TERRAFORM



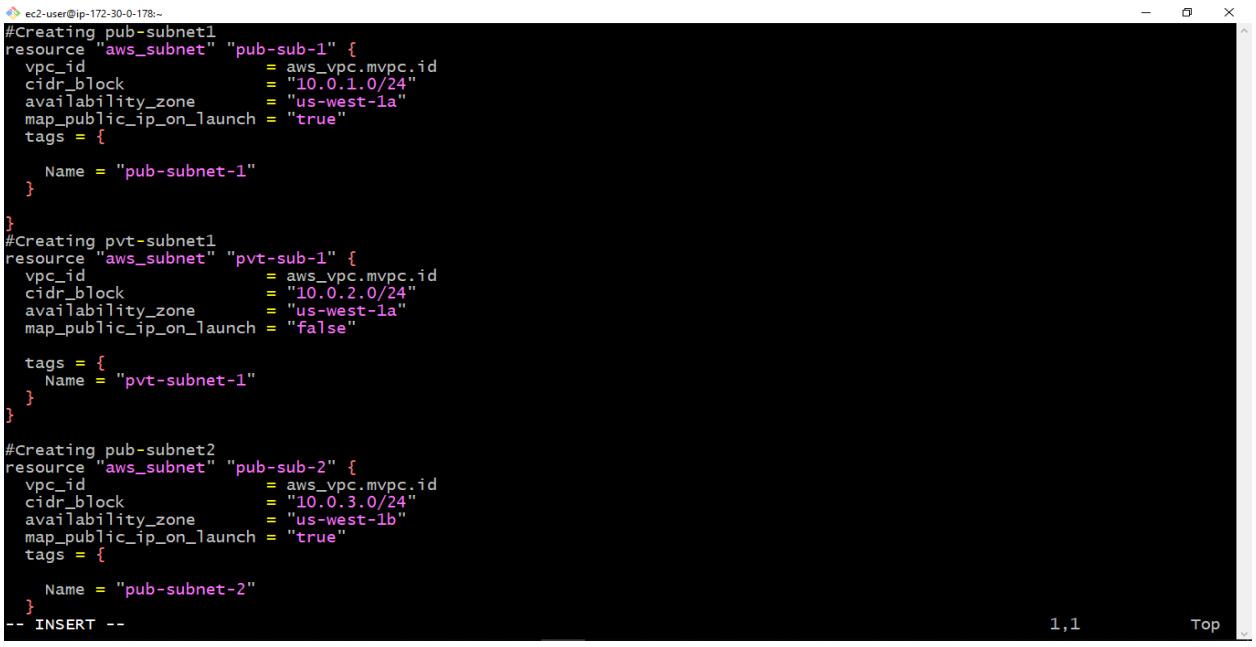


SHAIK GOUSERABBANI

DEPLOY THREE-TIER ARCHITECHTURE IN AWS BY USING TERRAFORM



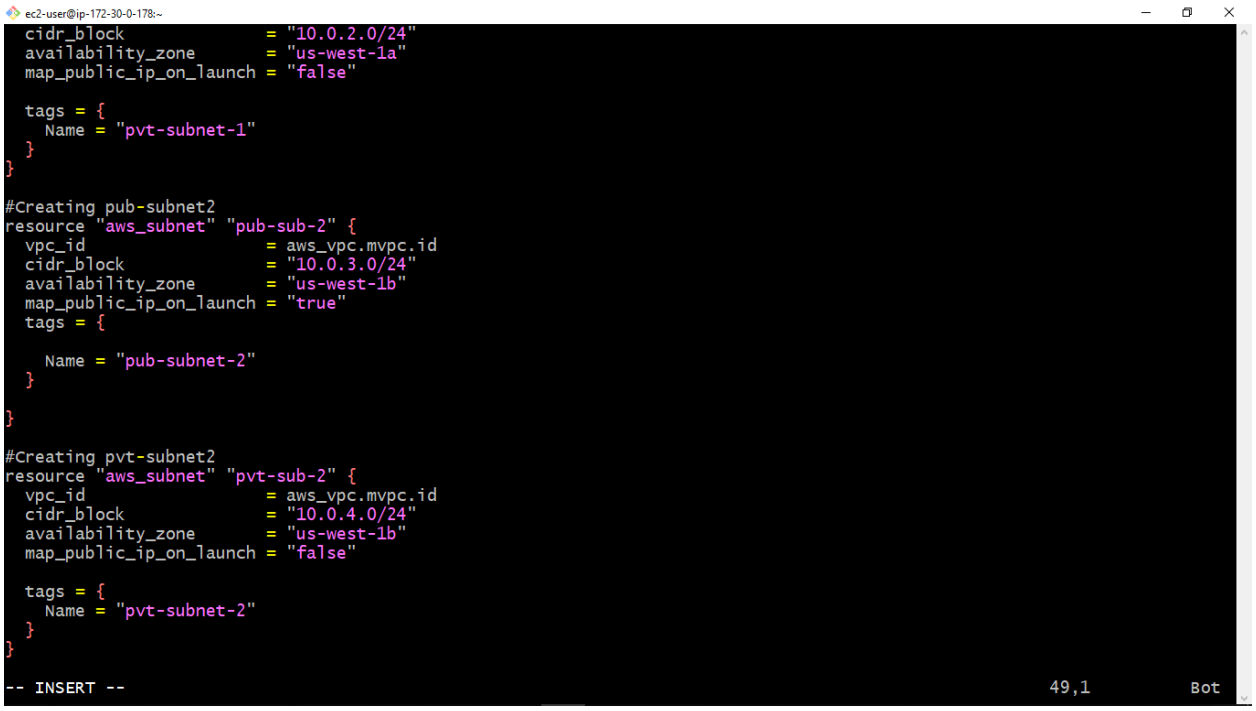
- Now create the subnets by using yaml script for that use this command <vi subnets.tf>



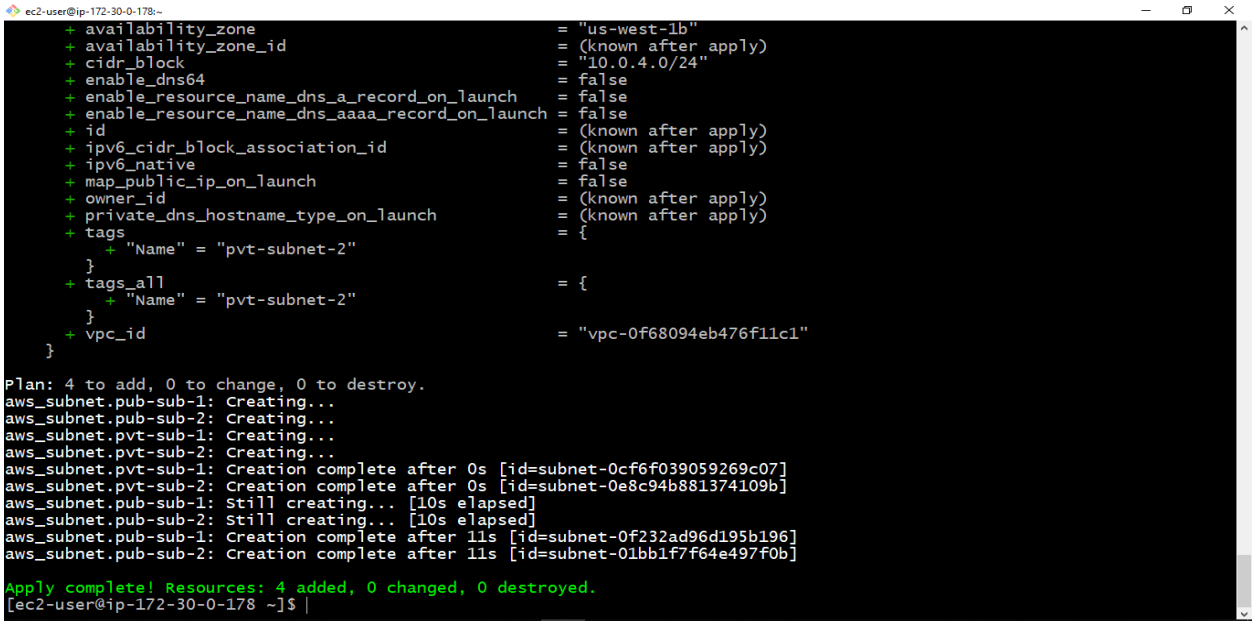
```
ec2-user@ip-172-30-0-178:~
#Creating pub-subnet1
resource "aws_subnet" "pub-sub-1" {
  vpc_id                  = aws_vpc.mvpc.id
  cidr_block              = "10.0.1.0/24"
  availability_zone       = "us-west-1a"
  map_public_ip_on_launch = "true"
  tags = {

    Name = "pub-subnet-1"
  }

}
#Creating pvt-subnet1
resource "aws_subnet" "pvt-sub-1" {
  vpc_id                  = aws_vpc.mvpc.id
  cidr_block              = "10.0.2.0/24"
  availability_zone       = "us-west-1a"
  map_public_ip_on_launch = "false"

  tags = {
    Name = "pvt-subnet-1"
  }
}

#Creating pub-subnet2
resource "aws_subnet" "pub-sub-2" {
  vpc_id                  = aws_vpc.mvpc.id
  cidr_block              = "10.0.3.0/24"
  availability_zone       = "us-west-1b"
  map_public_ip_on_launch = "true"
  tags = {

    Name = "pub-subnet-2"
  }
-- INSERT --                                                    1,1           Top
```

SHAIK GOUSERABBANI

DEPLOY THREE-TIER ARCHITECHTURE IN AWS BY USING TERRAFORM

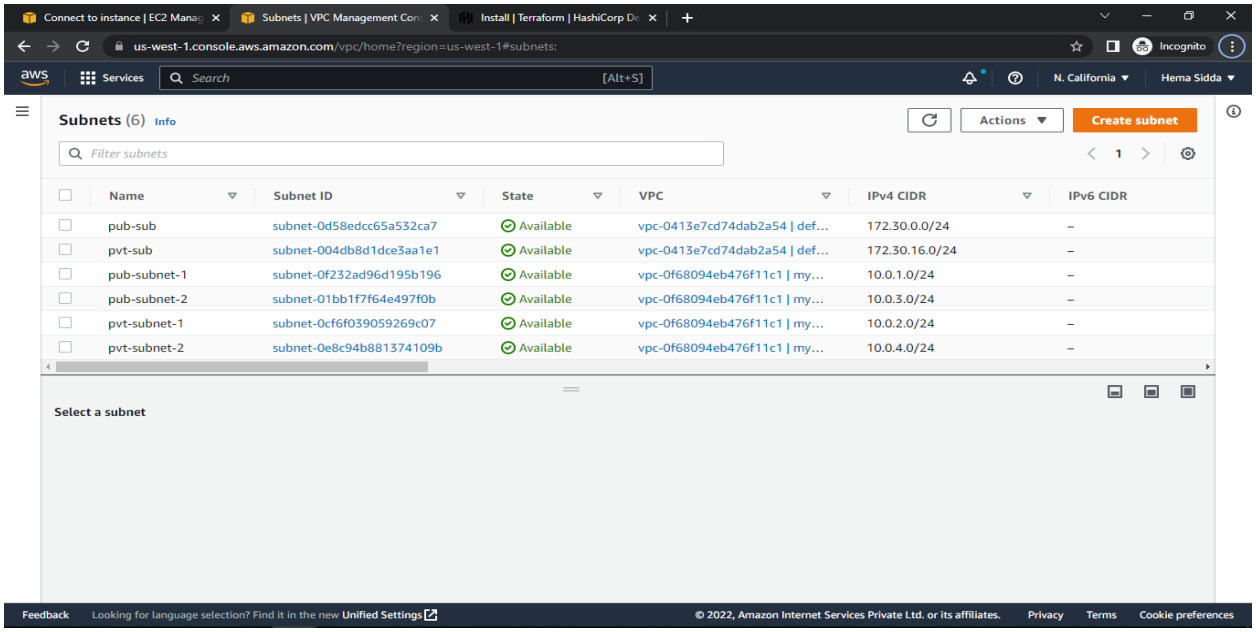


- Now by using single command I can create the resource that commad is <terraform apply --auto-approve>



SHAIK GOUSERABBANI

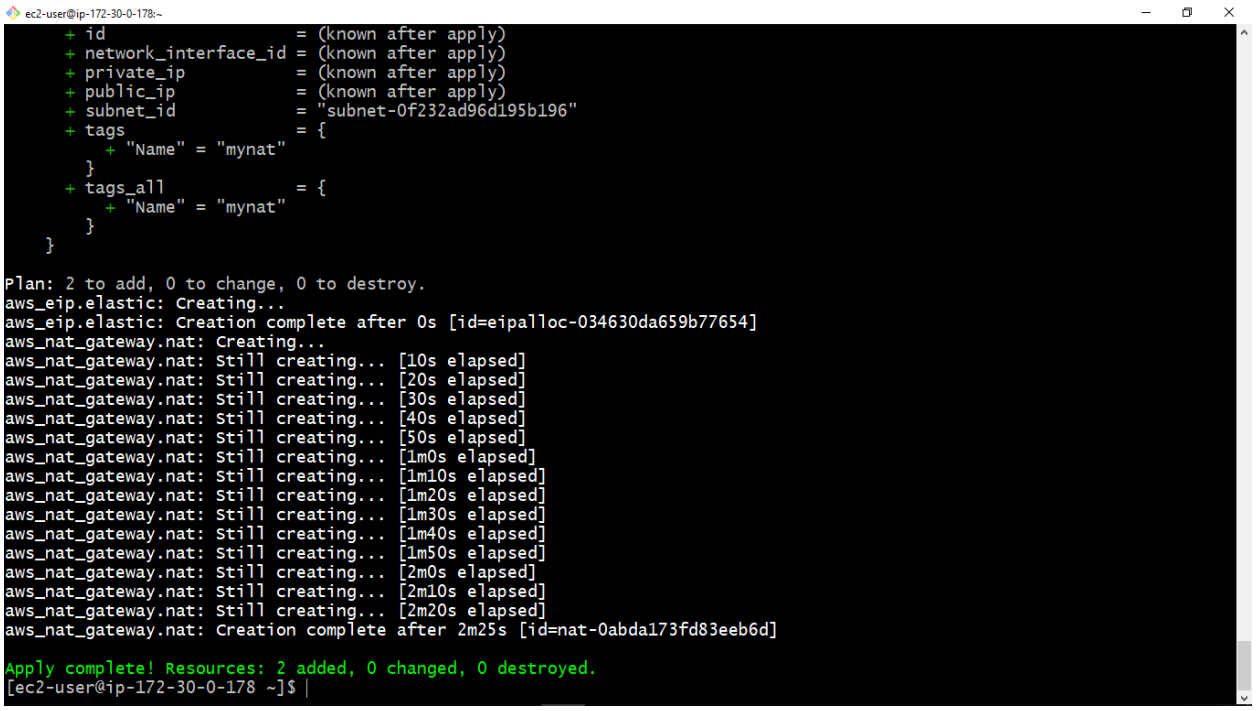
DEPLOY THREE-TIER ARCHITECHTURE IN AWS BY USING TERRAFORM

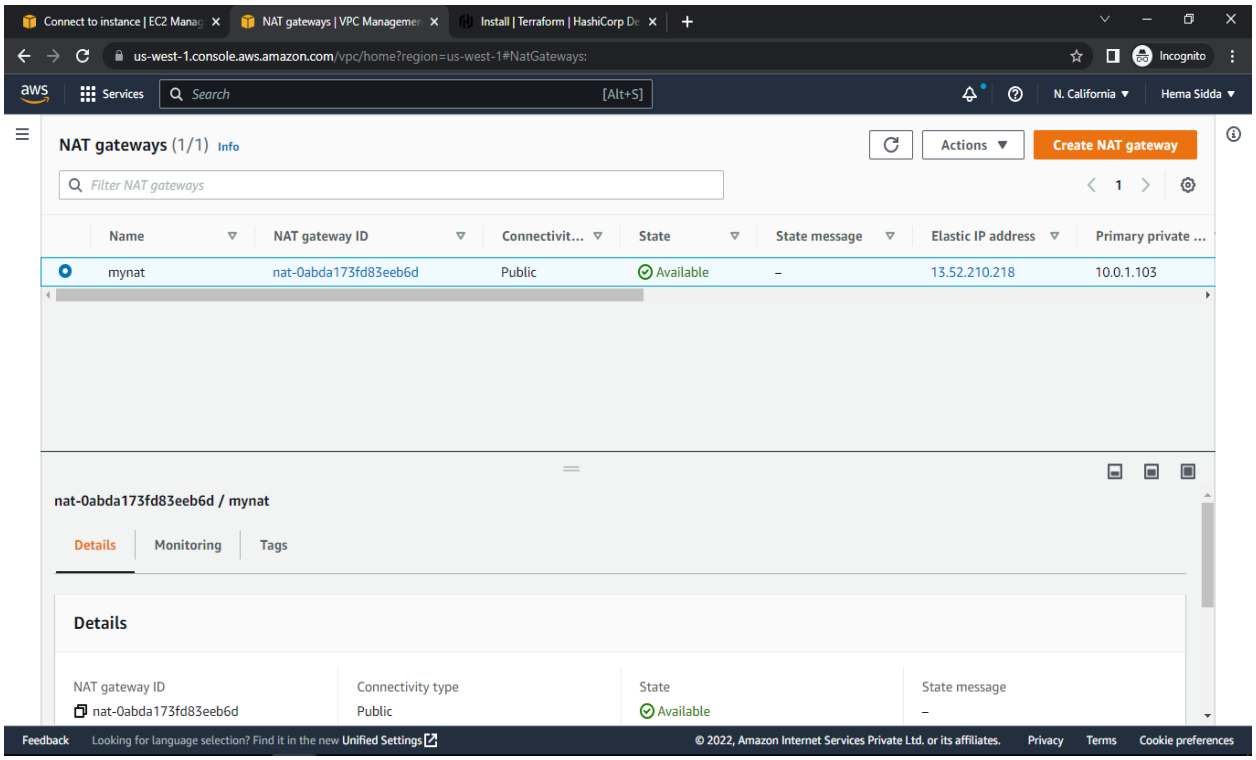


➢ Now create the NAT-gateway for that create a .tf file and enter the yaml script.

SHAIK GOUSERABBANI

DEPLOY THREE-TIER ARCHITECHTURE IN AWS BY USING TERRAFORM





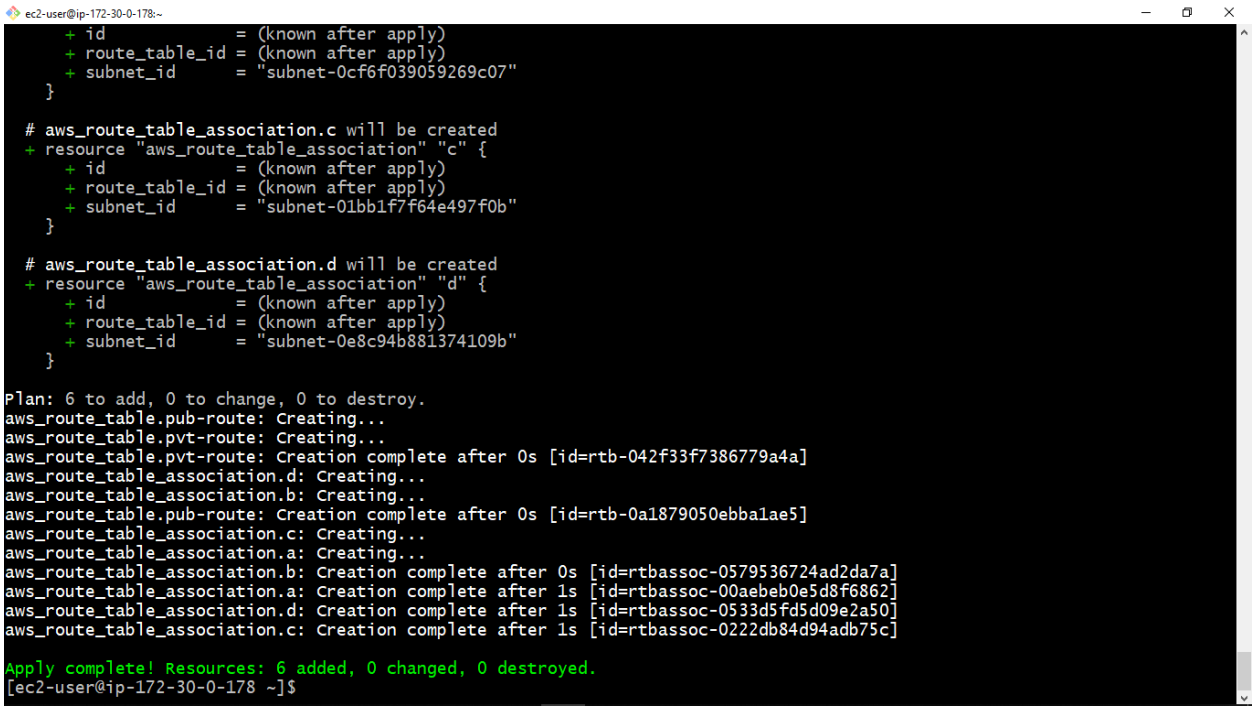
SHAIK GOUSERABBANI

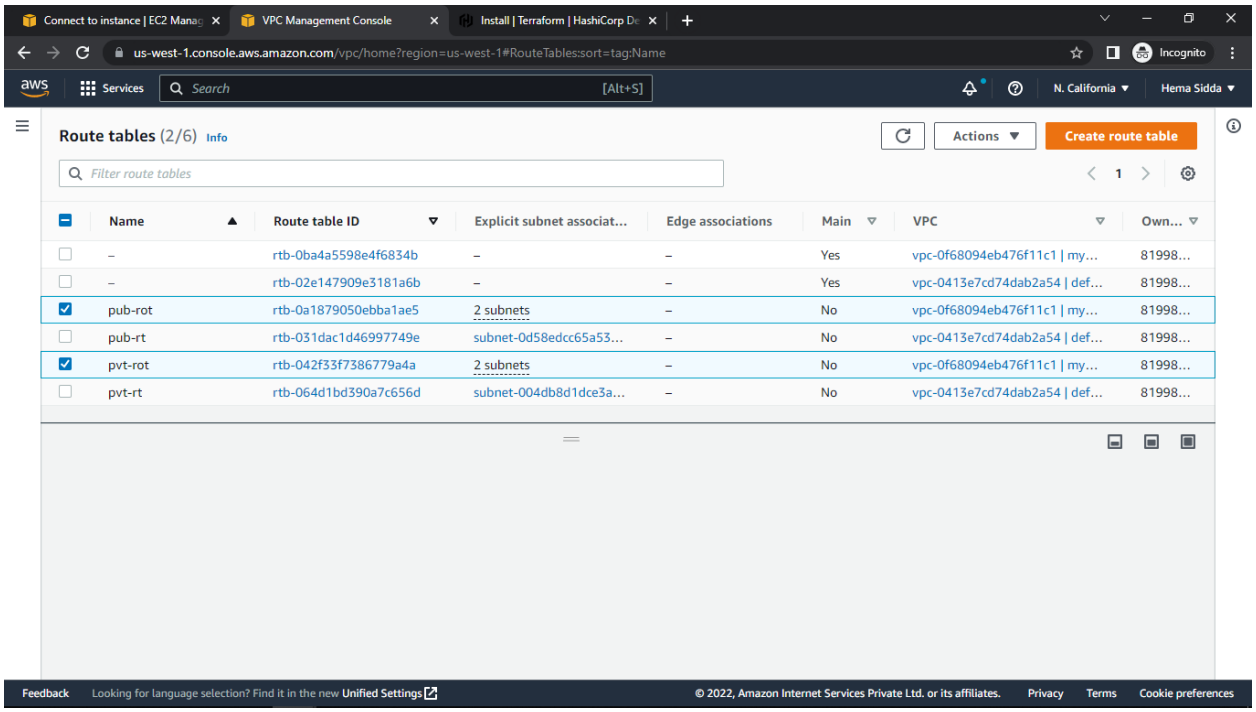
DEPLOY THREE-TIER ARCHITECHTURE IN AWS BY USING TERRAFORM

- Now go to create the route table and associate subnets, Internet gateway and NAT gate way for that create a .tf file by using the vi mode as <vi route.tf>

SHAIK GOUSERABBANI

DEPLOY THREE-TIER ARCHITECHTURE IN AWS BY USING TERRAFORM

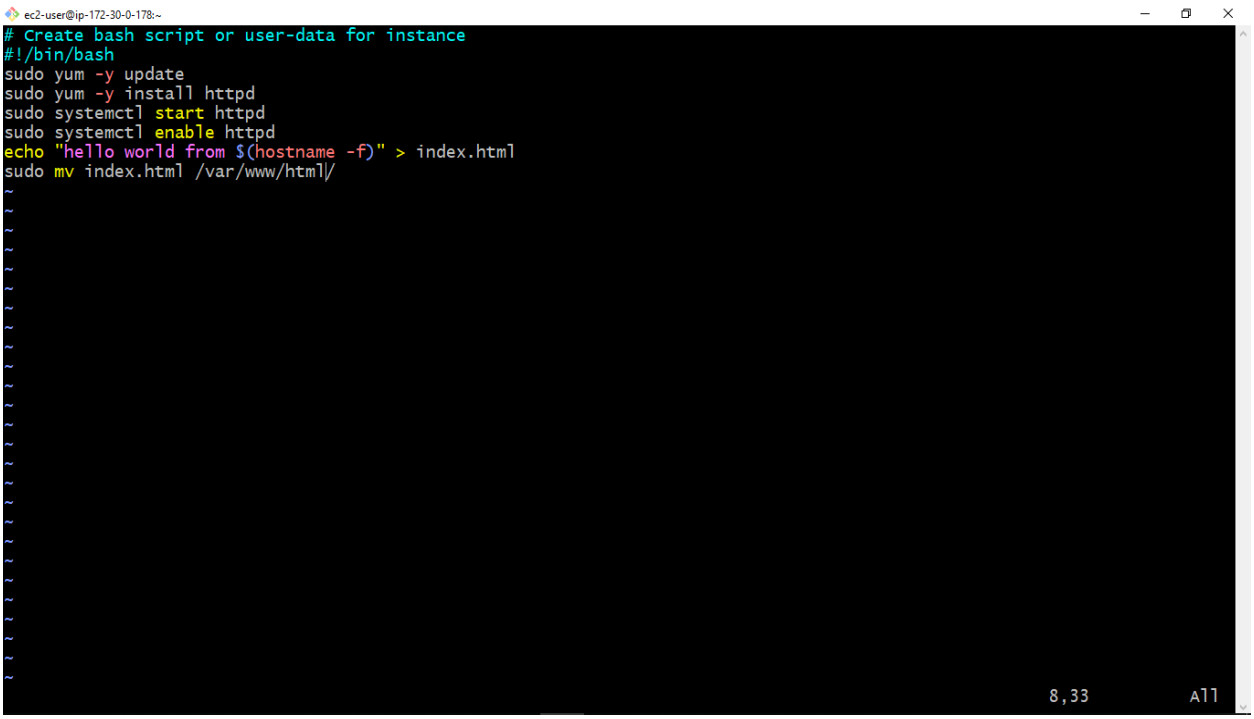




SHAIK GOUSERABBANI

DEPLOY THREE-TIER ARCHITECHTURE IN AWS BY USING TERRAFORM

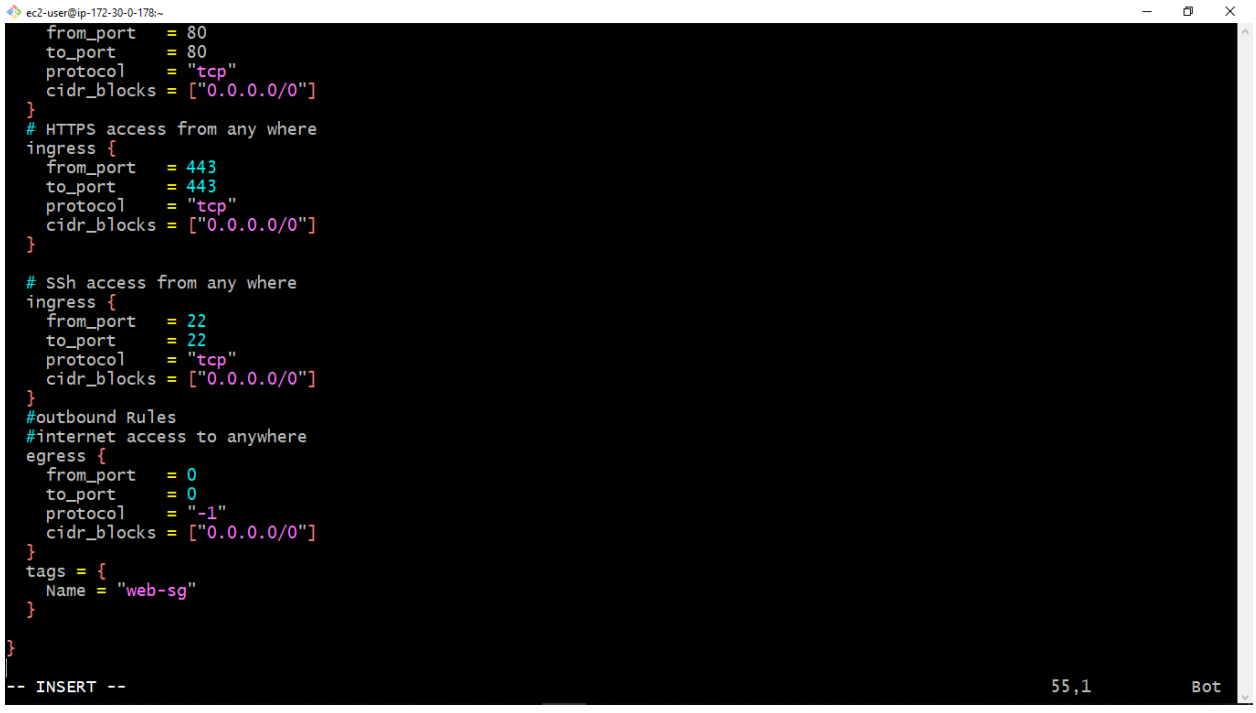
- Now create the bash script file by the data.sh



- Now create the public instance by using yaml scripting.

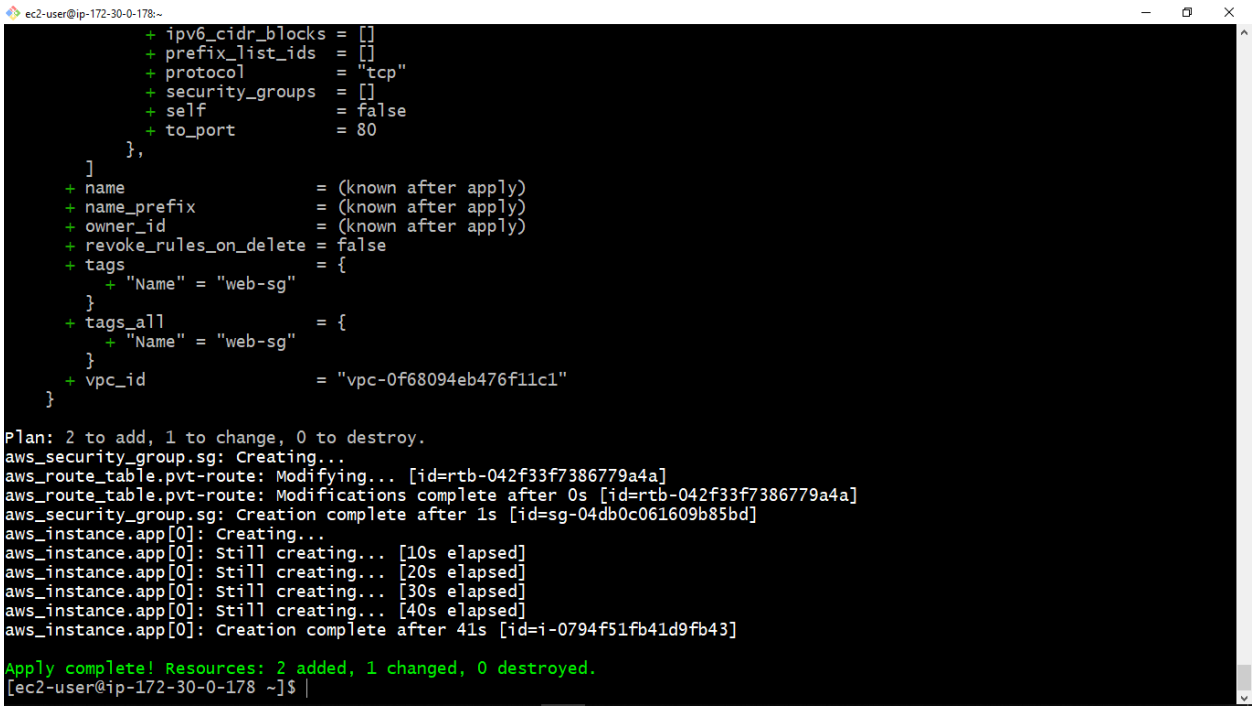
SHAIK GOUSERABBANI

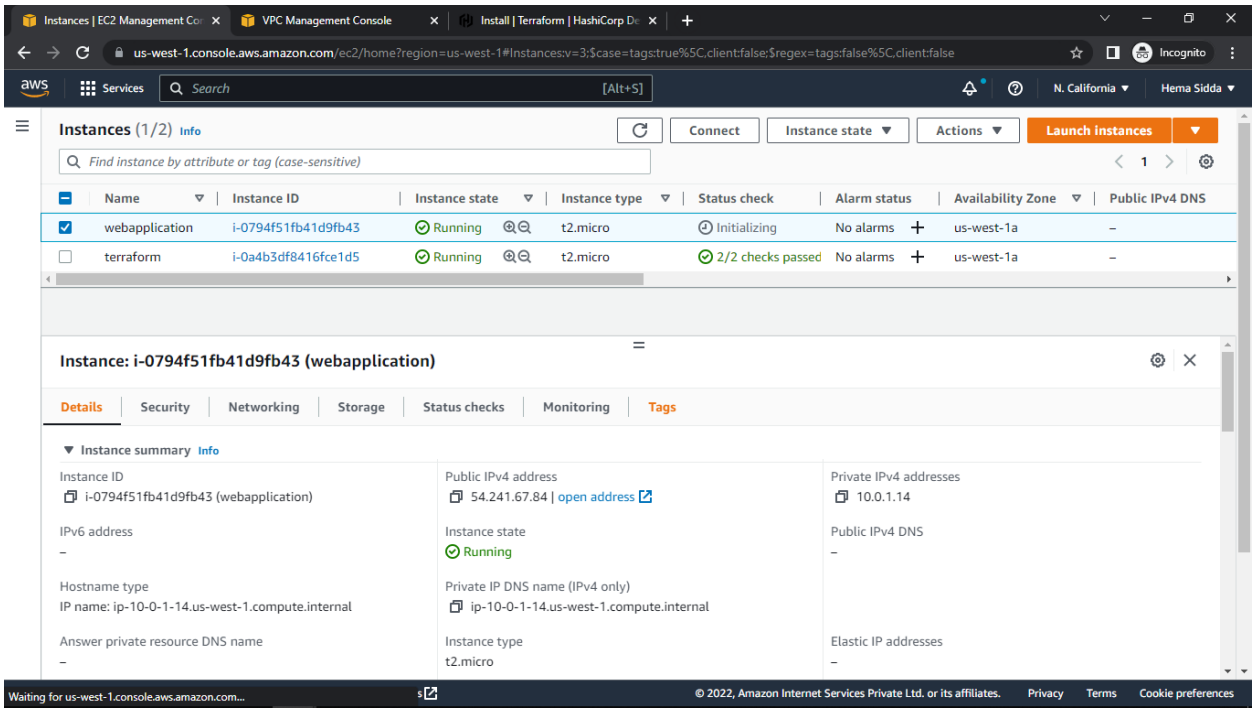
DEPLOY THREE-TIER ARCHITECHTURE IN AWS BY USING TERRAFORM



SHAIK GOUSERABBANI

DEPLOY THREE-TIER ARCHITECHTURE IN AWS BY USING TERRAFORM





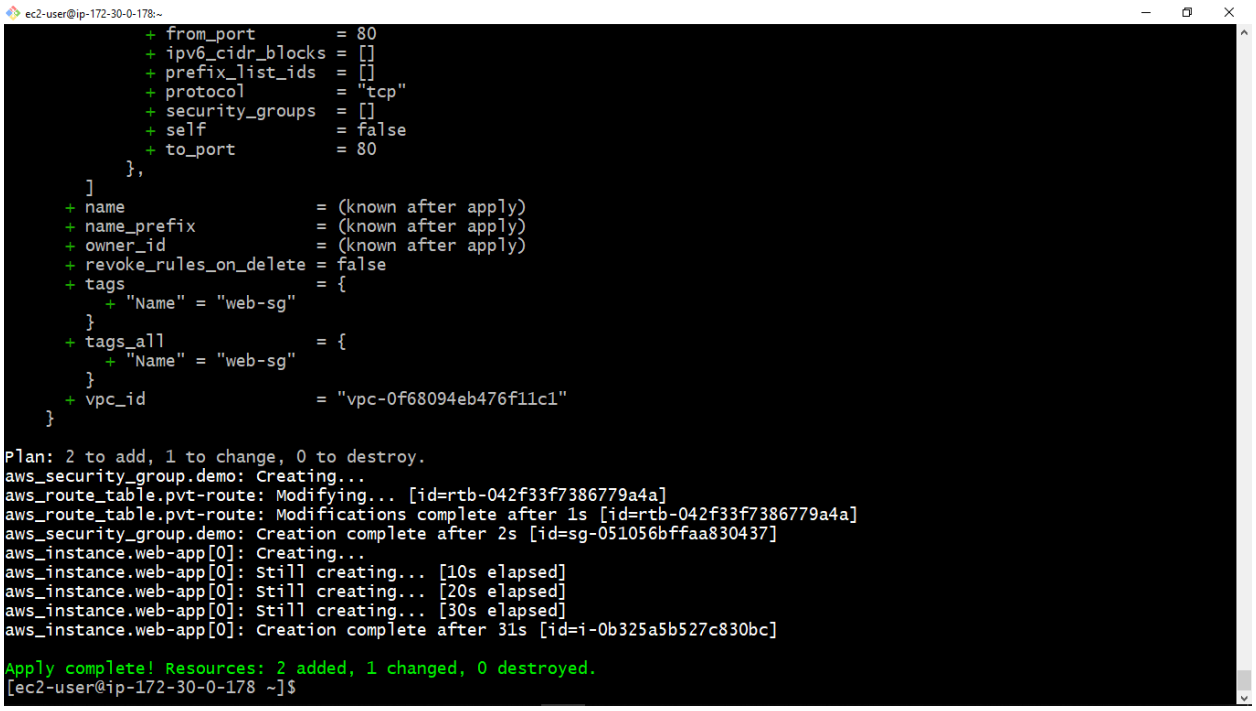
SHAIK GOUSERABBANI

DEPLOY THREE-TIER ARCHITECHTURE IN AWS BY USING TERRAFORM

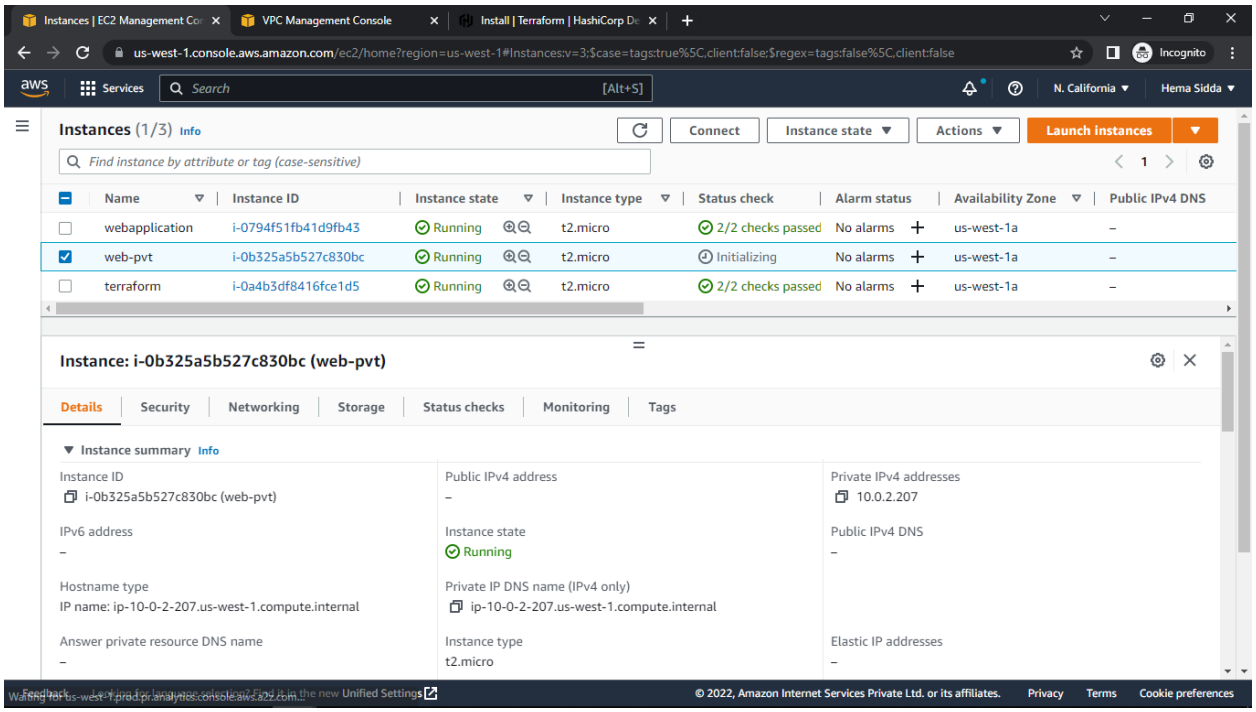
- Now create the pvt instance by using the yaml script of pvt instance .

SHAIK GOUSERABBANI

DEPLOY THREE-TIER ARCHITECHTURE IN AWS BY USING TERRAFORM



```
ec2-user@ip-172-30-0-178:~
            + from_port        = 80
            + ipv6_cidr_blocks = []
            + prefix_list_ids  = []
            + protocol         = "tcp"
            + security_groups  = []
            + self             = false
            + to_port          = 80
          },
      ]
    + name                   = (known after apply)
    + name_prefix            = (known after apply)
    + owner_id               = (known after apply)
    + revoke_rules_on_delete = false
    + tags                   = {
        + "Name" = "web-sg"
      }
    + tags_all               = {
        + "Name" = "web-sg"
      }
    + vpc_id                 = "vpc-0f68094eb476f11c1"
  }

Plan: 2 to add, 1 to change, 0 to destroy.
aws_security_group.demo: Creating...
aws_route_table.pvt-route: Modifying... [id=rtb-042f33f7386779a4a]
aws_route_table.pvt-route: Modifications complete after 1s [id=rtb-042f33f7386779a4a]
aws_security_group.demo: Creation complete after 2s [id=sg-051056bffaa830437]
aws_instance.web-app[0]: Creating...
aws_instance.web-app[0]: Still creating... [10s elapsed]
aws_instance.web-app[0]: Still creating... [20s elapsed]
aws_instance.web-app[0]: Still creating... [30s elapsed]
aws_instance.web-app[0]: Creation complete after 31s [id=i-0b325a5b527c830bc]

Apply complete! Resources: 2 added, 1 changed, 0 destroyed.
[ec2-user@ip-172-30-0-178 ~]$
```



SHAIK GOUSERABBANI

DEPLOY THREE-TIER ARCHITECHTURE IN AWS BY USING TERRAFORM

- Now create the RDS(relational database) service by using yaml scripting.

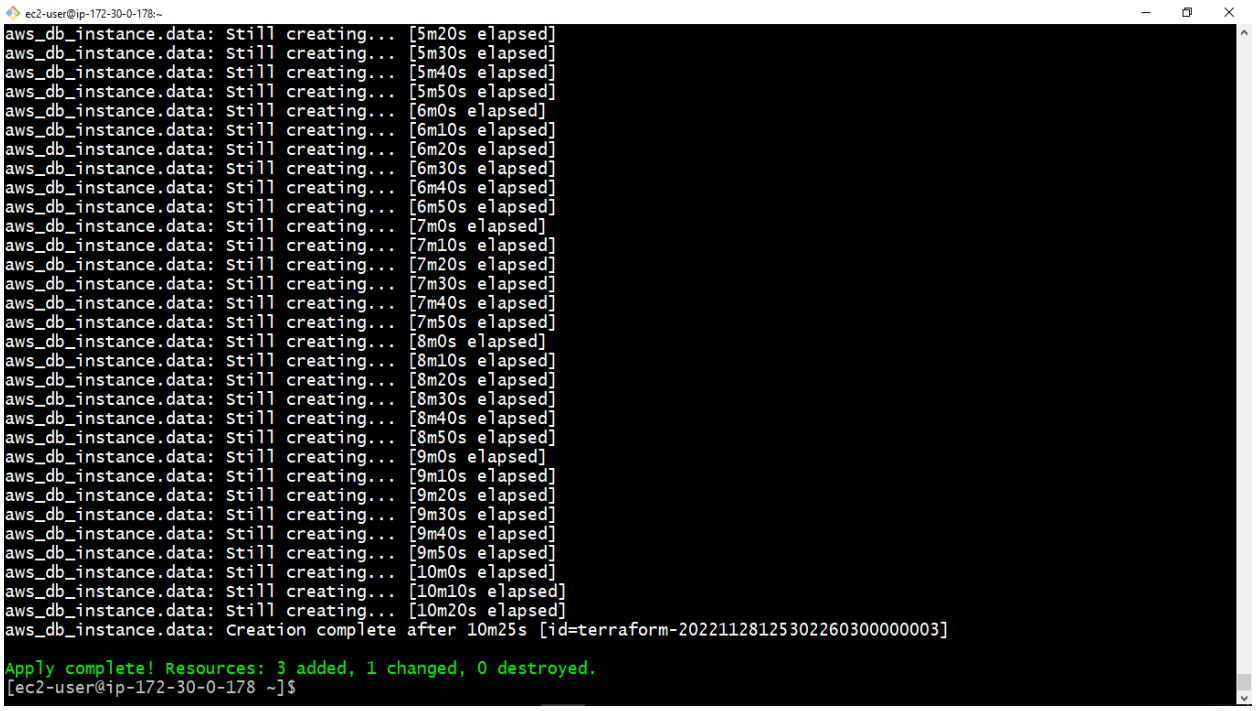
ec2-user@ip-172-30-0-178:~

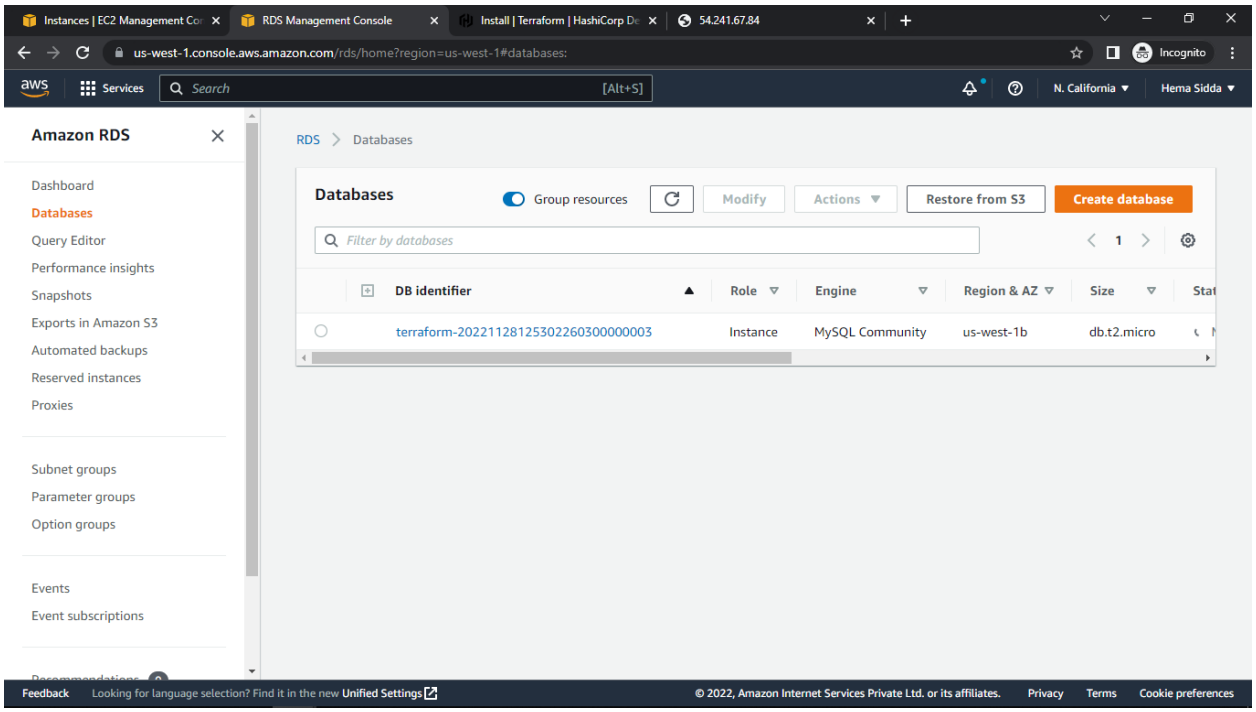
```
# Create subnet groups
resource "aws_db_subnet_group" "db-sub" {
subnet_ids = [aws_subnet.pvt-sub-1.id, aws_subnet.pvt-sub-2.id]
tags = {
        Name = "dbsubnet"
        }
}
# Create the data-base
resource "aws_db_instance" "data" {
 allocated_storage = "10"
        db_subnet_group_name = aws_db_subnet_group.db-sub.id
        db_name = "mydata"
        engine = "mysql"
        engine_version = "8.0.28"
    instance_class = "db.t2.micro"
    multi_az = "true"
    username = "admin"
    password = "admin1230"
    vpc_security_group_ids = ["${aws_security_group.db-sg.id}"]
    skip_final_snapshot = "true"

}

# Create the security groups
resource "aws_security_group" "db-sg" {
        vpc_id = aws_vpc.mvpc.id

  ingress {
  from_port = 3306
  to_port = 3306
  protocol = "tcp"
  security_groups = [aws_security_group.demo.id]
  }
-- INSERT --                                                    1,1           Top
```

ec2-user@ip-172-30-0-178:~

```
        engine = "mysql"
        engine_version = "8.0.28"
    instance_class = "db.t2.micro"
    multi_az = "true"
    username = "admin"
    password = "admin1230"
    vpc_security_group_ids = ["${aws_security_group.db-sg.id}"]
    skip_final_snapshot = "true"

}

# Create the security groups
resource "aws_security_group" "db-sg" {
        vpc_id = aws_vpc.mvpc.id

  ingress {
  from_port = 3306
  to_port = 3306
  protocol = "tcp"
  security_groups = [aws_security_group.demo.id]
  }

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

  }
  tags = {
  Name = "database-sg"
  }
}
-- INSERT --                                                    46,1          Bot
```

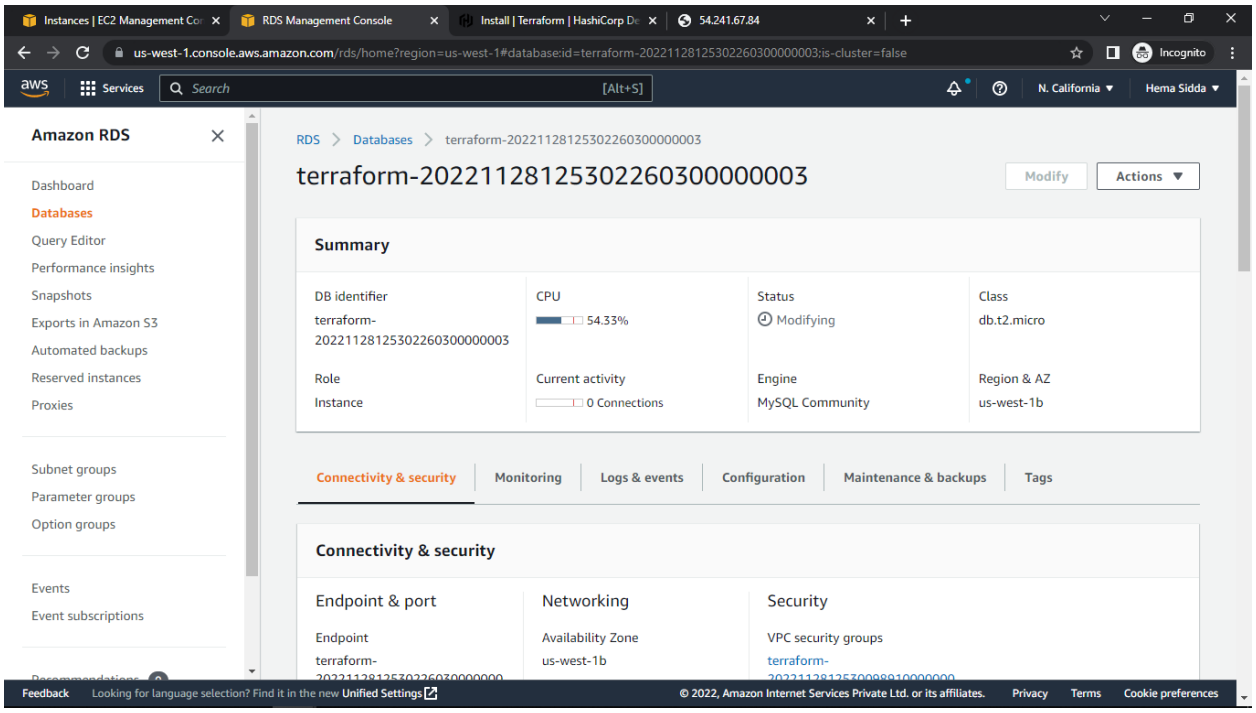
DEPLOY THREE-TIER ARCHITECHTURE IN AWS BY USING TERRAFORM



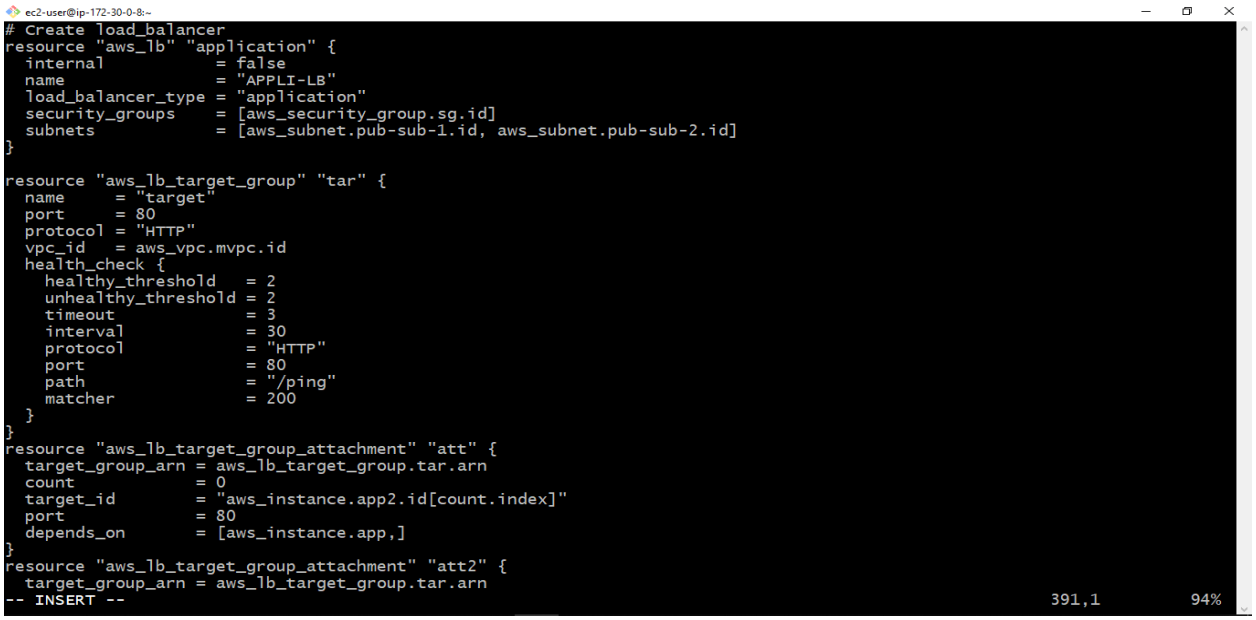


SHAIK GOUSERABBANI

DEPLOY THREE-TIER ARCHITECHTURE IN AWS BY USING TERRAFORM

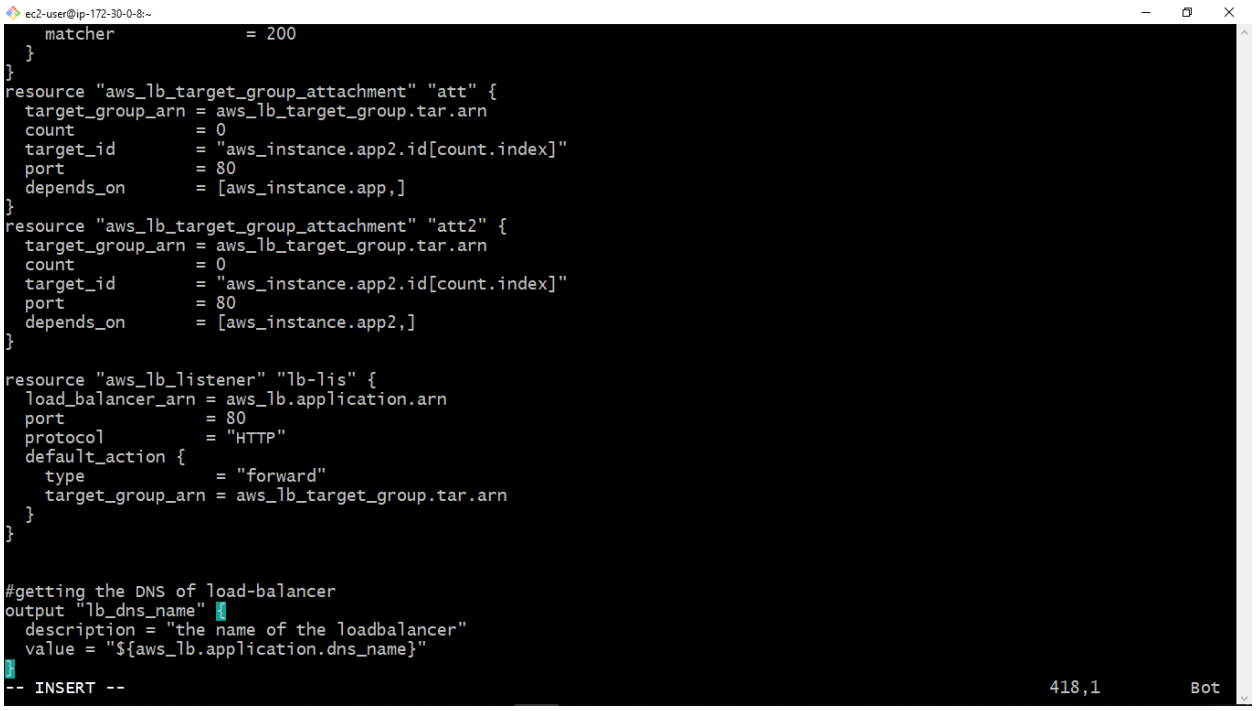


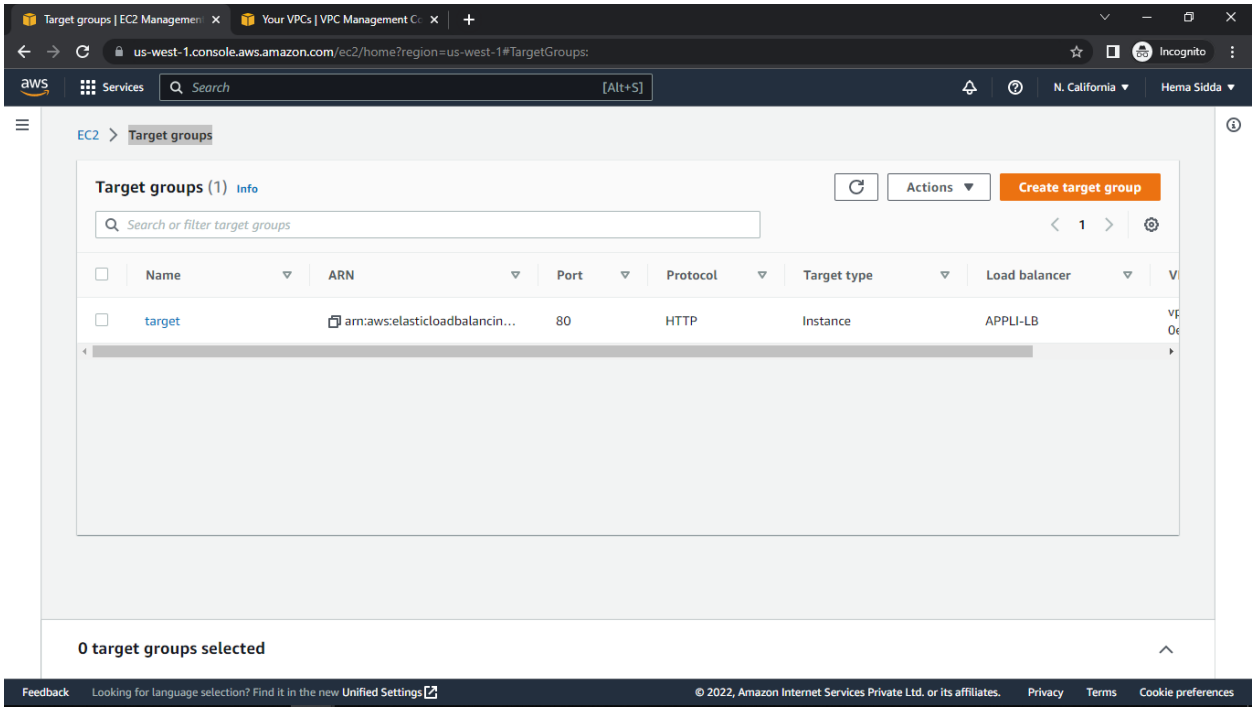
➢ Now create the load-balancer by using yaml script.



SHAIK GOUSERABBANI

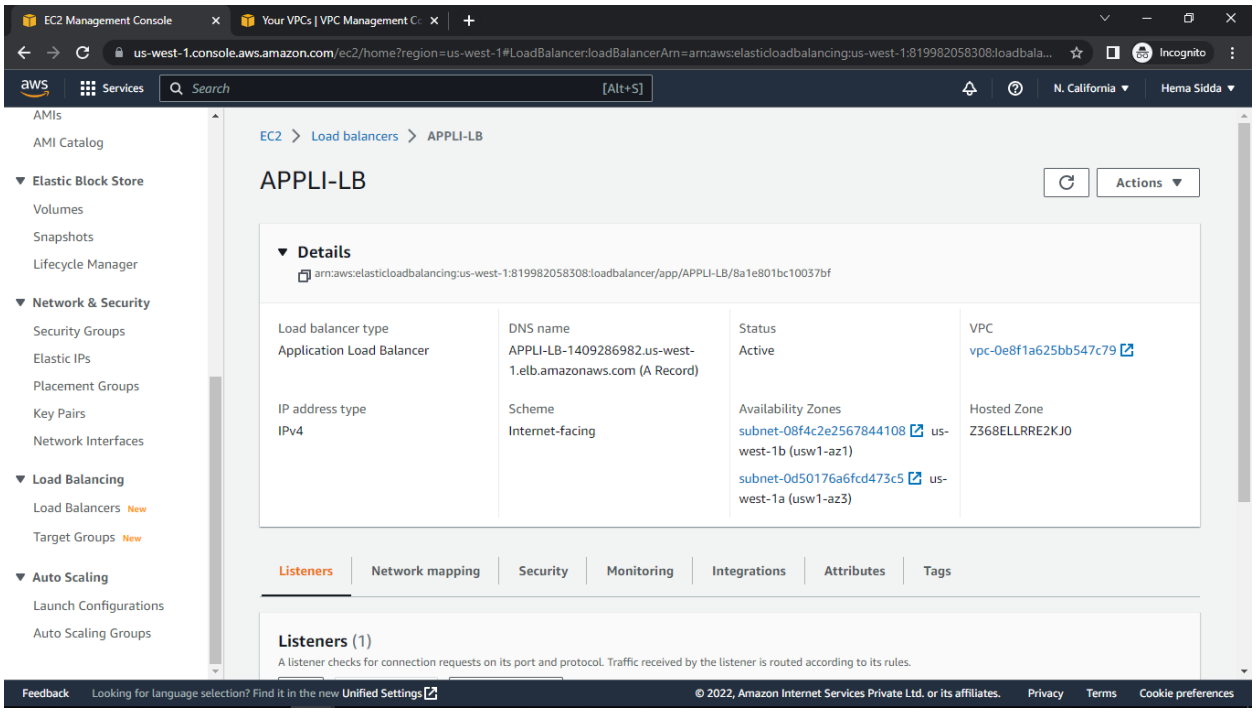
DEPLOY THREE-TIER ARCHITECHTURE IN AWS BY USING TERRAFORM

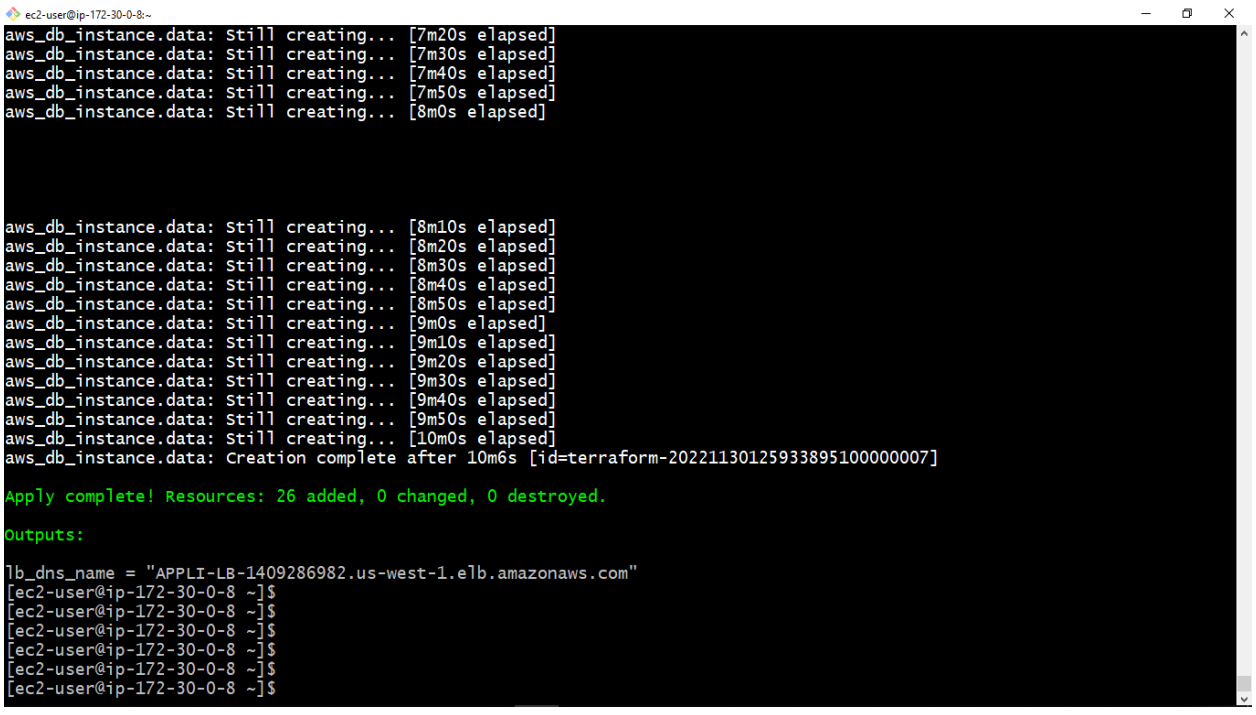




SHAIK GOUSERABBANI

DEPLOY THREE-TIER ARCHITECHTURE IN AWS BY USING TERRAFORM

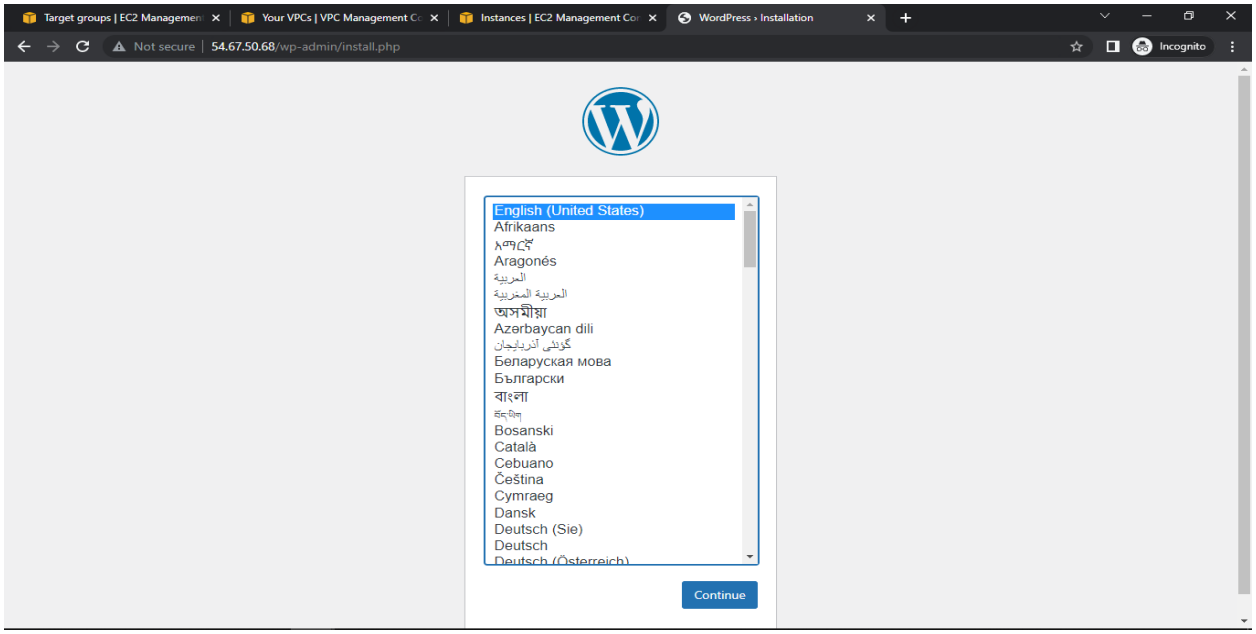


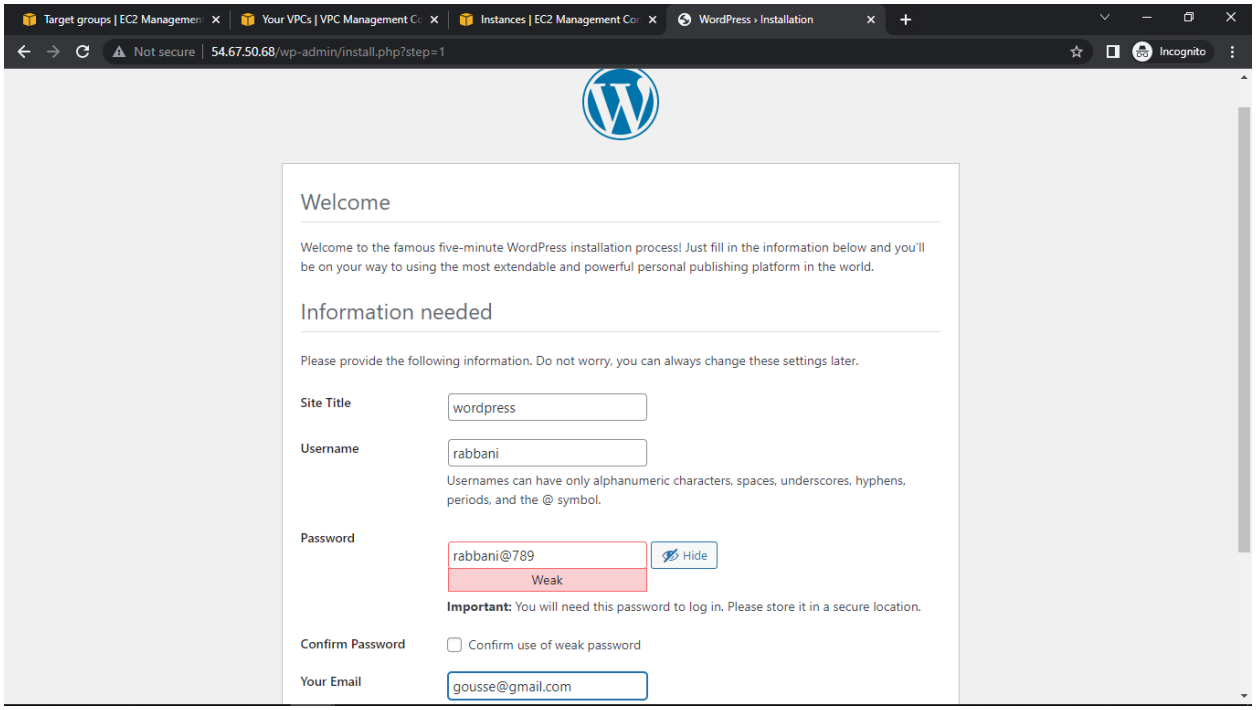


SHAIK GOUSERABBANI

DEPLOY THREE-TIER ARCHITECHTURE IN AWS BY USING TERRAFORM

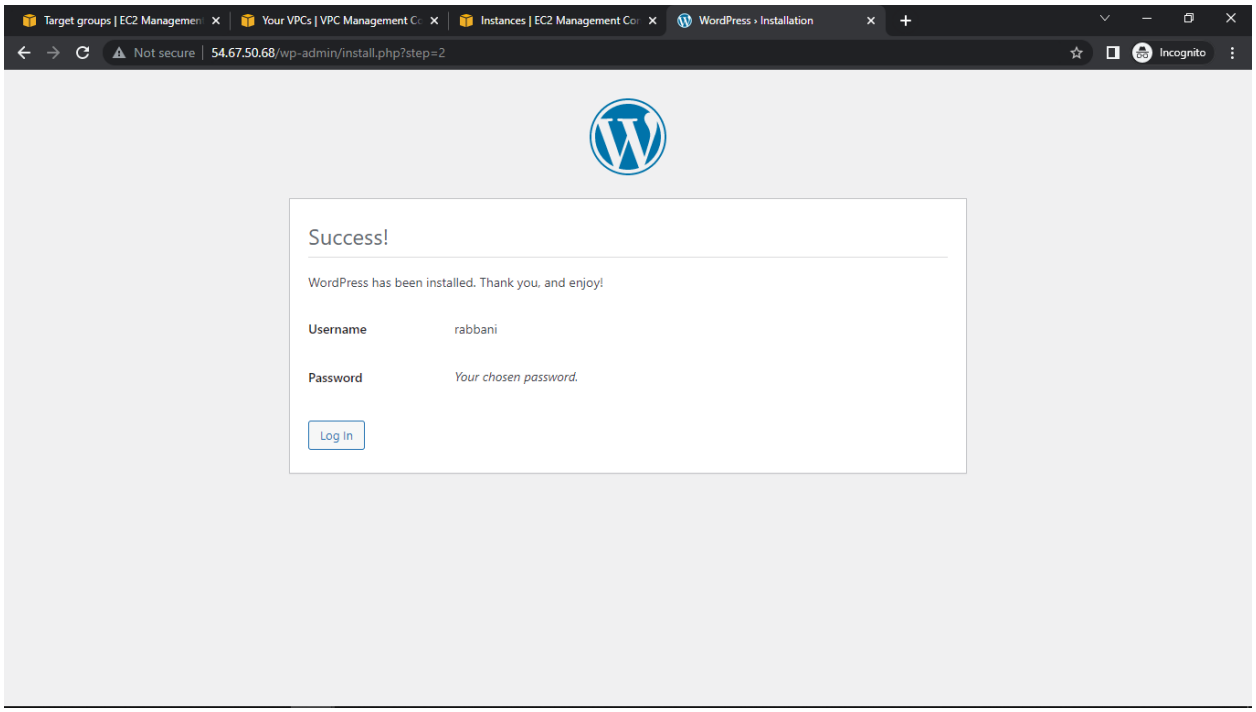
- Now go to EC2 instances and select the public ip(ipv4) and browse it on Google.

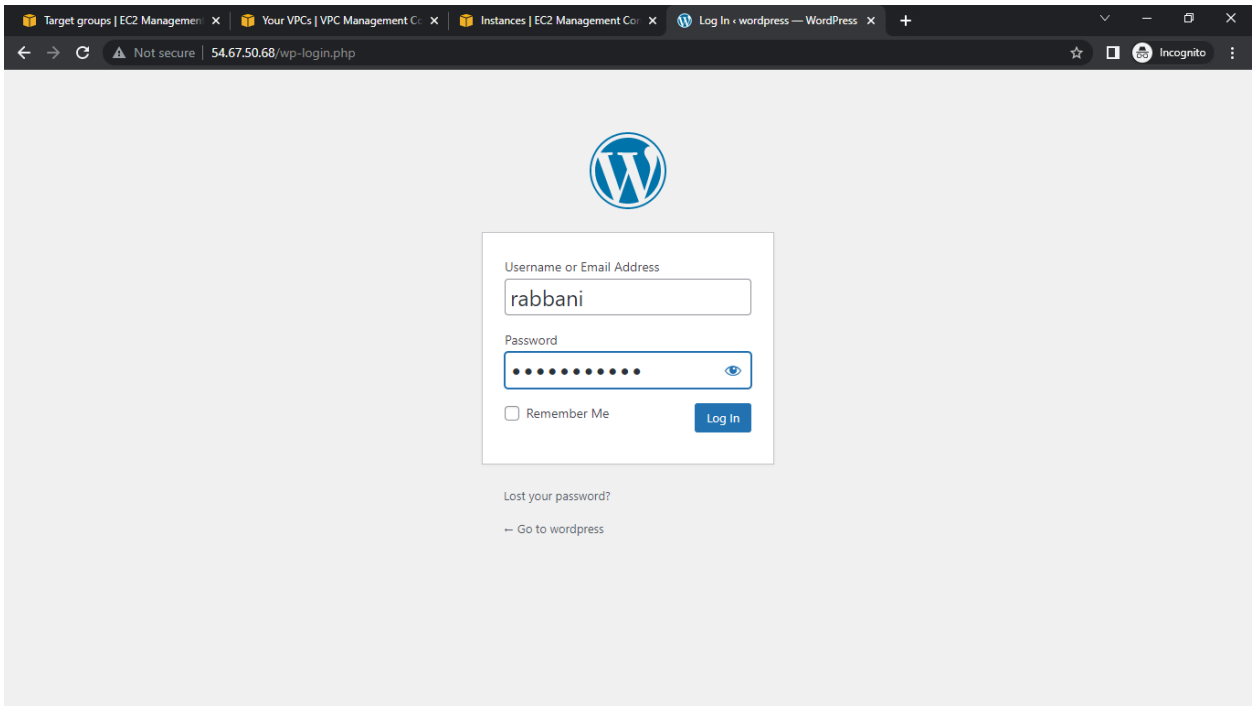




SHAIK GOUSERABBANI

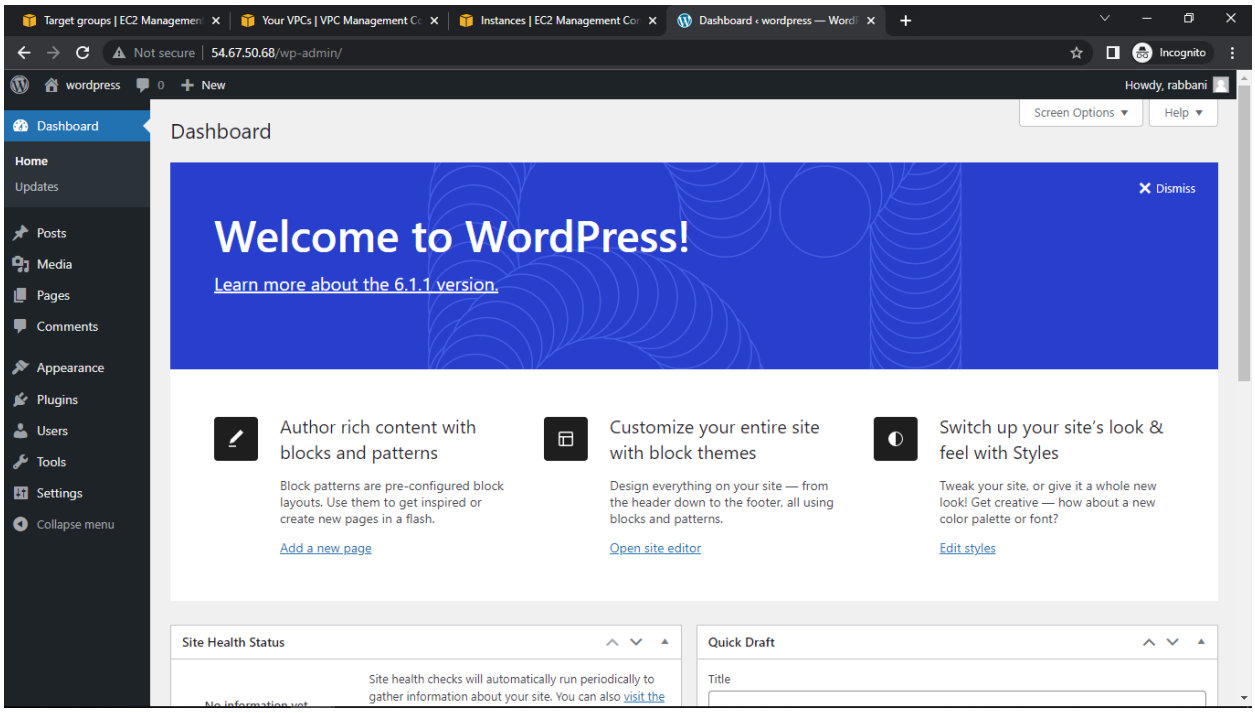
DEPLOY THREE-TIER ARCHITECHTURE IN AWS BY USING TERRAFORM





SHAIK GOUSERABBANI

DEPLOY THREE-TIER ARCHITECHTURE IN AWS BY USING TERRAFORM



- Now copy the DNS of load-balancer browse in Google and see the result.

SHAIK GOUSERABBANI