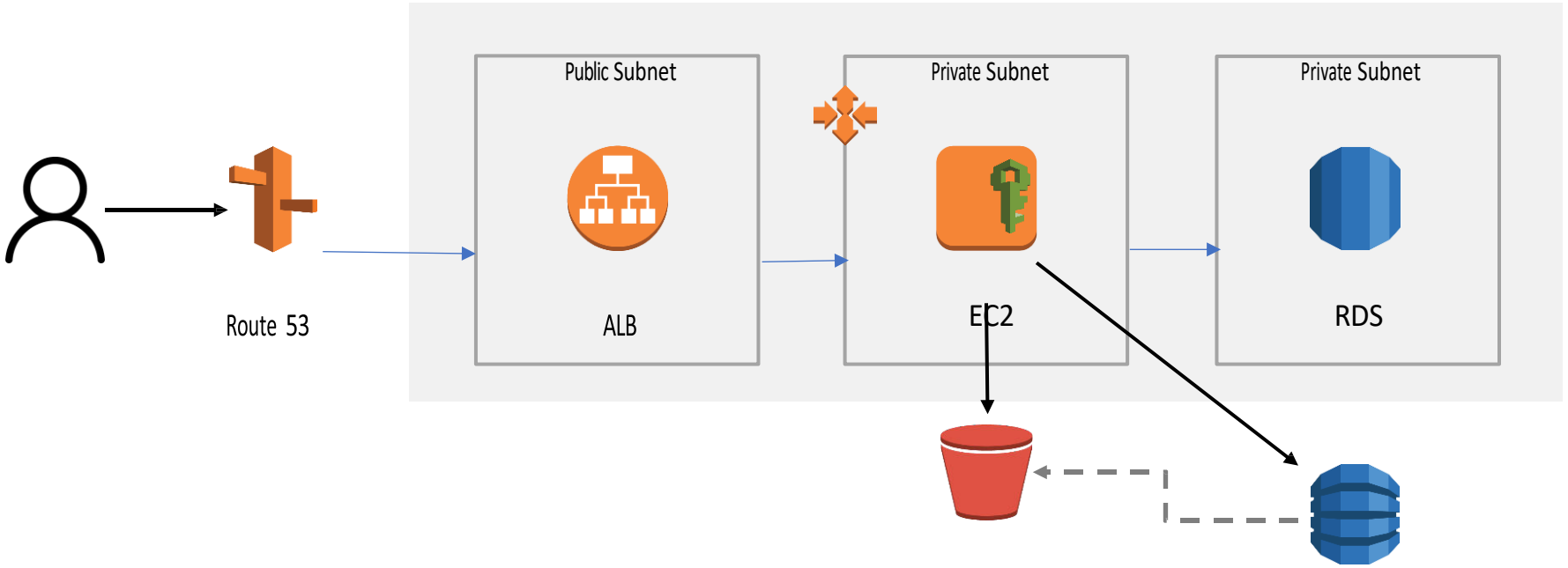
Problem Statement

Employee profile of XYZ company – New employees input their information and upload photos. Photos to be stored into S3 bucket and other information to be stored into RDS. Metadata to be stored into DynamoDB.

Shubham Bhamare

Technical Architecture

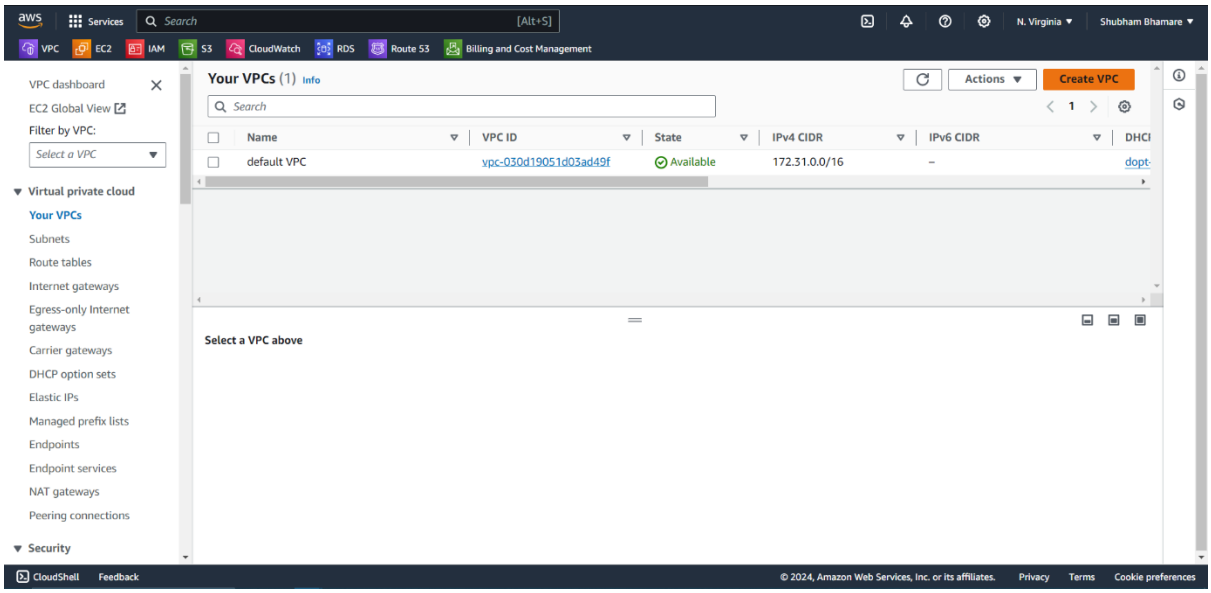


Shubham Bhamare

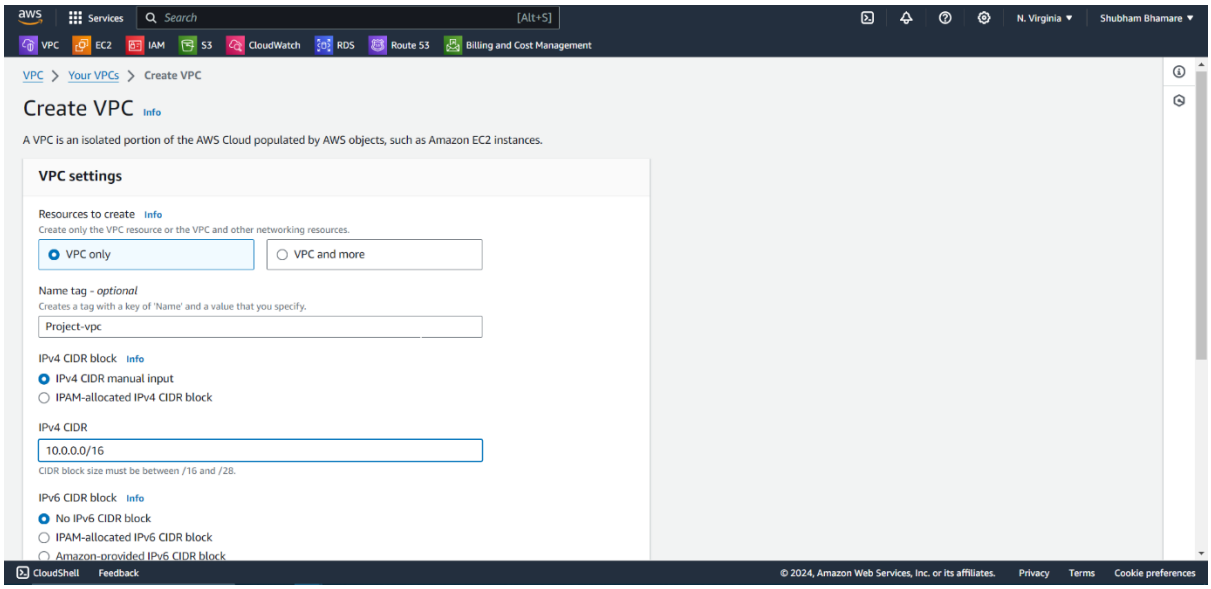
AWS Setup

- Setup VPC for Load Balancer, Application EC2 instance and RDS Database - one public and one private subnets.
- Create Load balancer.
- Create RDS DB instance and DynamoDB table.
- S3 Bucket
- Get domain name and map it with Load Balancer
- Create instance profile that has to be attached to the EC2 instances being launched. Instance profile should have permission to access RDS, DynamoDB and S3 bucket.

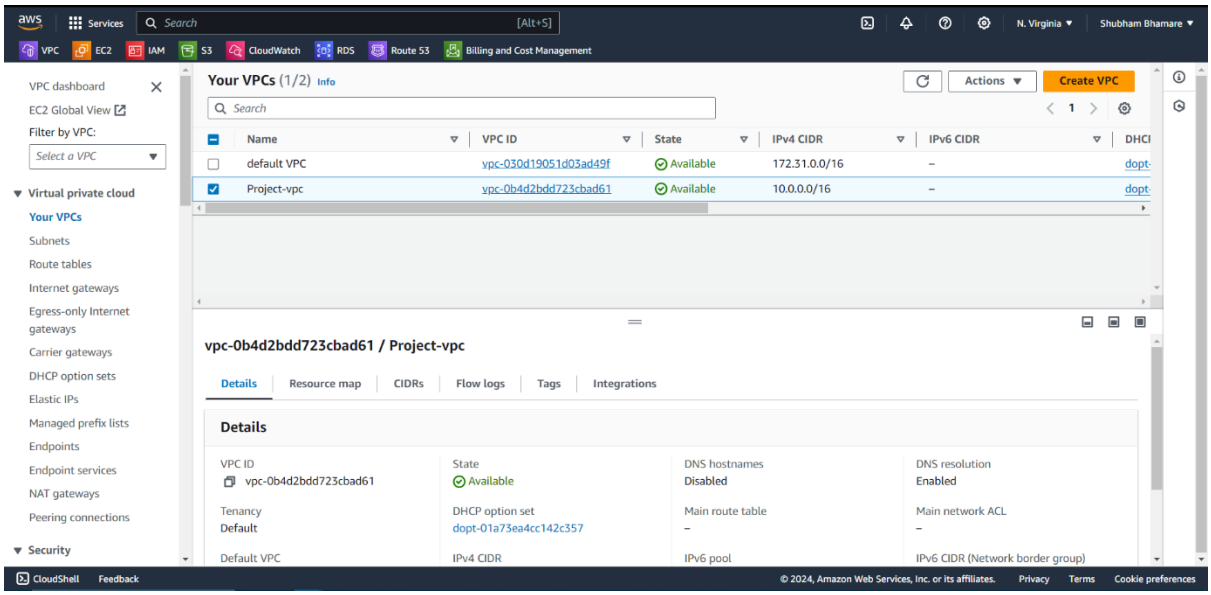
Step 1: Go to VPC Dashboard and click on “Create VPC” to create a new private VPC.



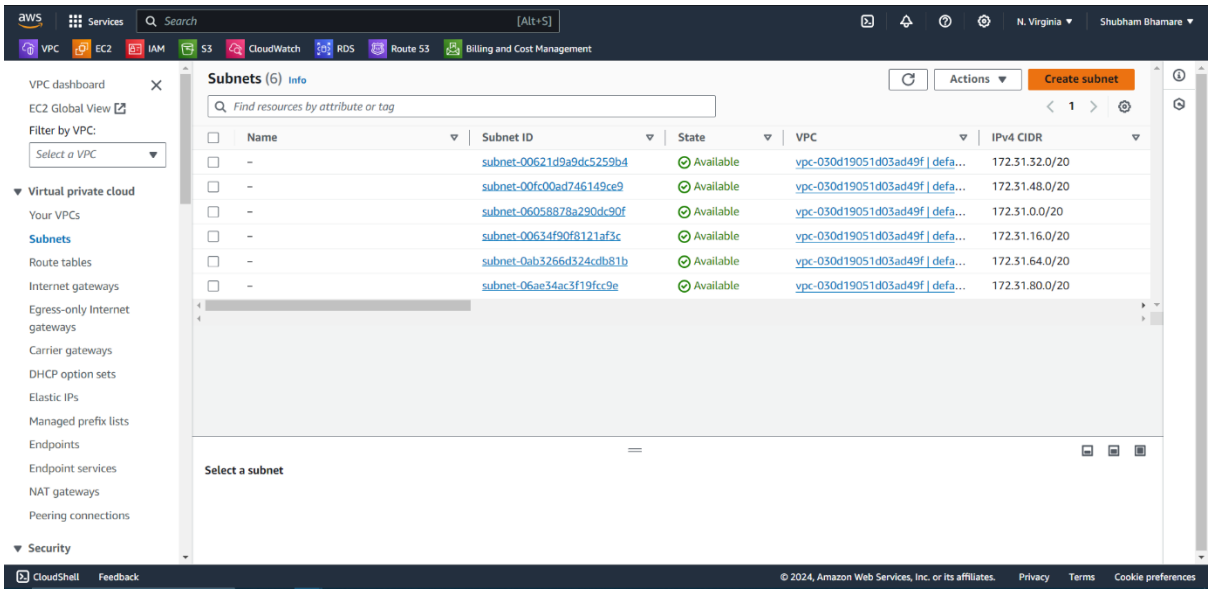
Step 2: Specify the name, CIDR for your VPC.



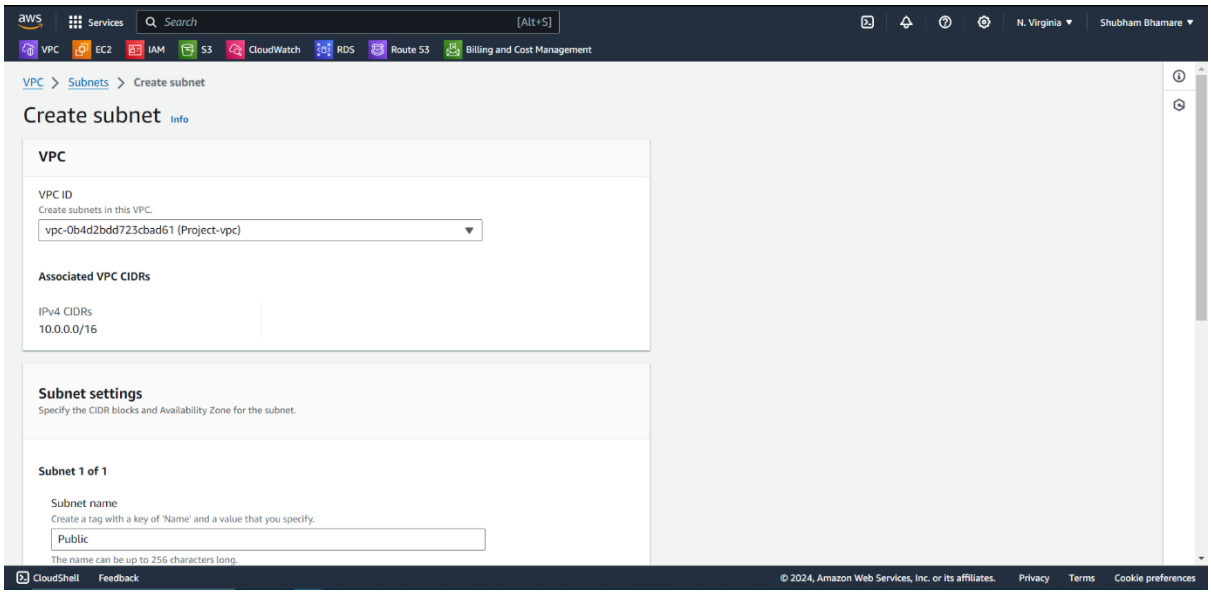
Step 3: VPC is created successfully.



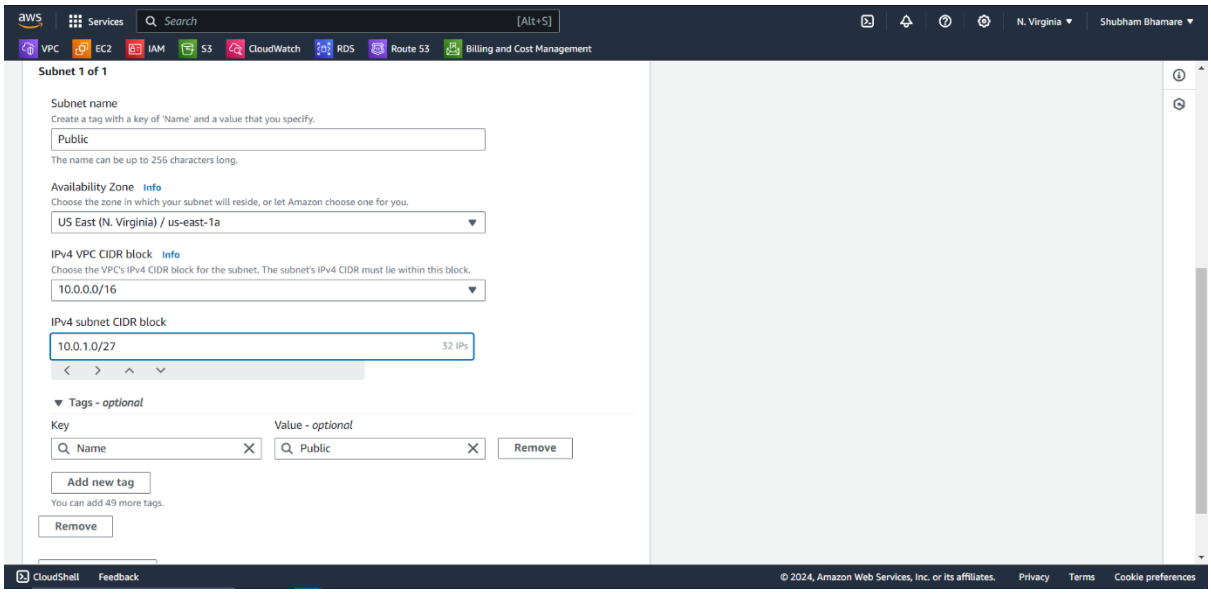
Step 4: Now go to subnet and click on “Create Subnet” to create one public and one private subnet.



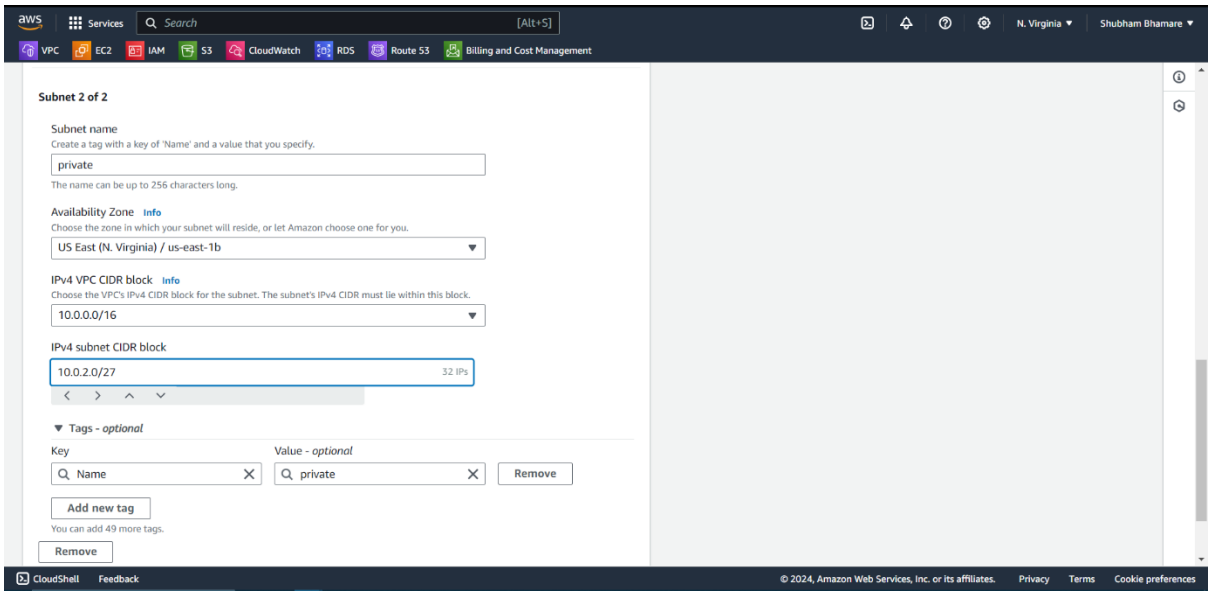
Step 5: Select your VPC in which you want to create subnet.



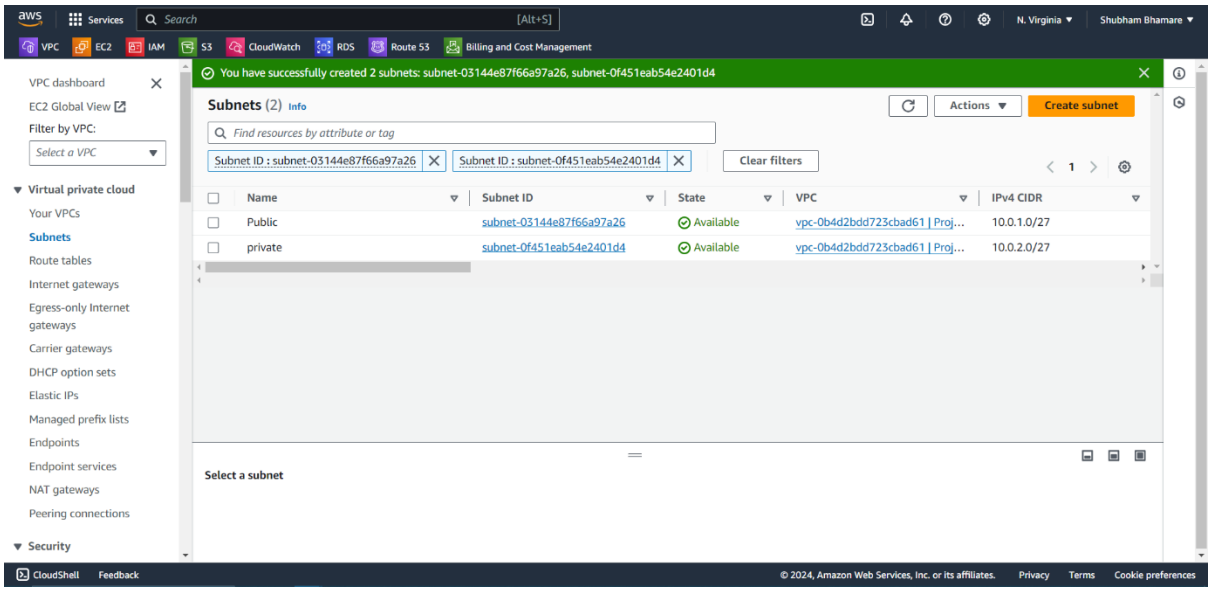
Step 6: Mention the name to your subnet , Select AZ, Specify CIDR for your subnet.



Step 7: Again add another subnet and now create one private subnet and mention name, AZ, CIDR to your subnet.

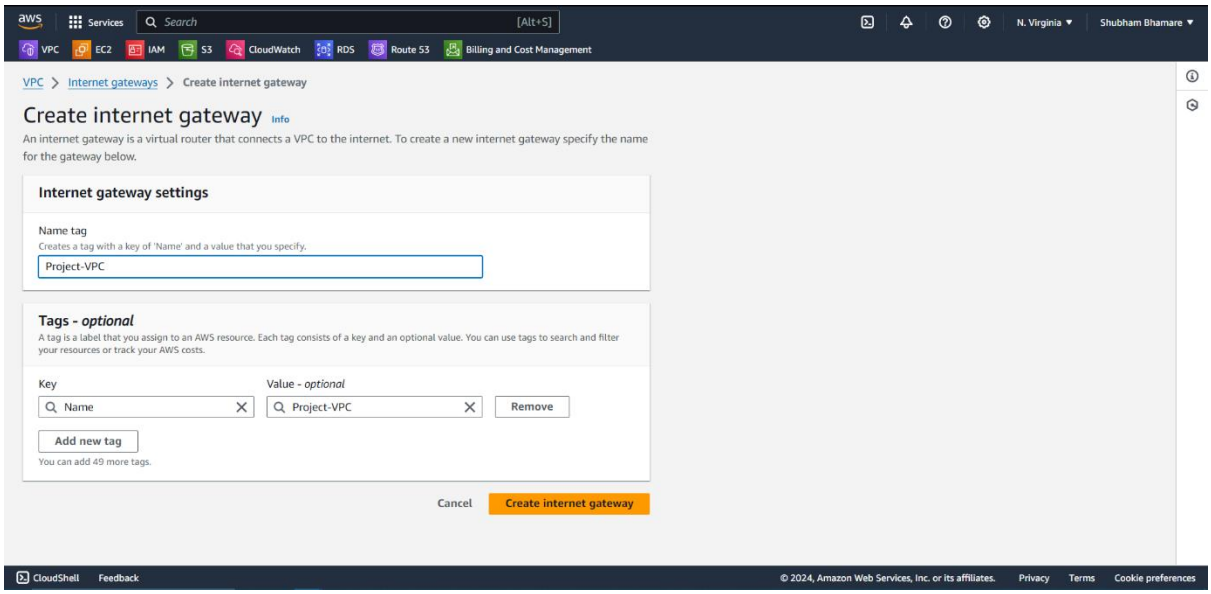


Step 8: Two subnets are created successfully.

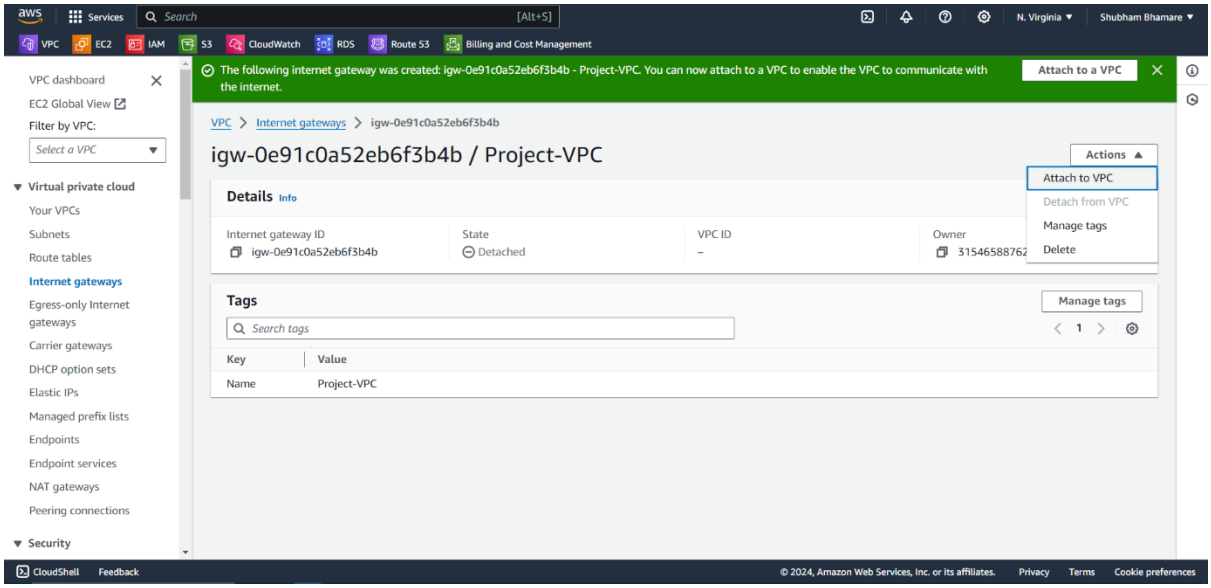


Shubham Bhamare

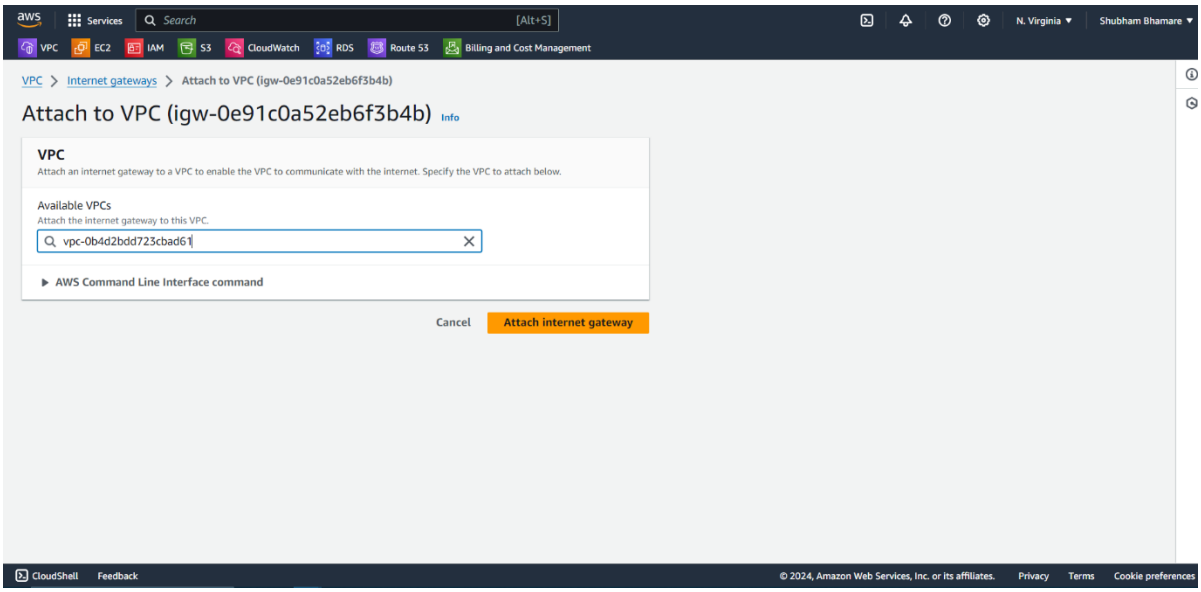
Step 9: Now go to Internet Gateway and create one internet gateway.



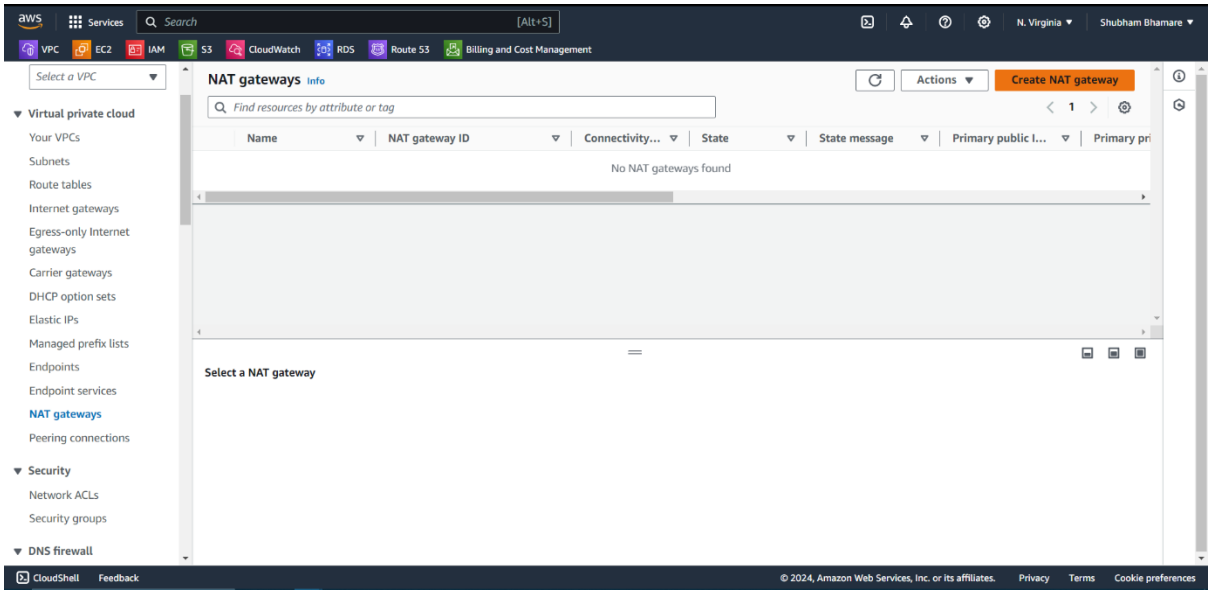
Step 10: Now attach that internet gateway to your VPC.



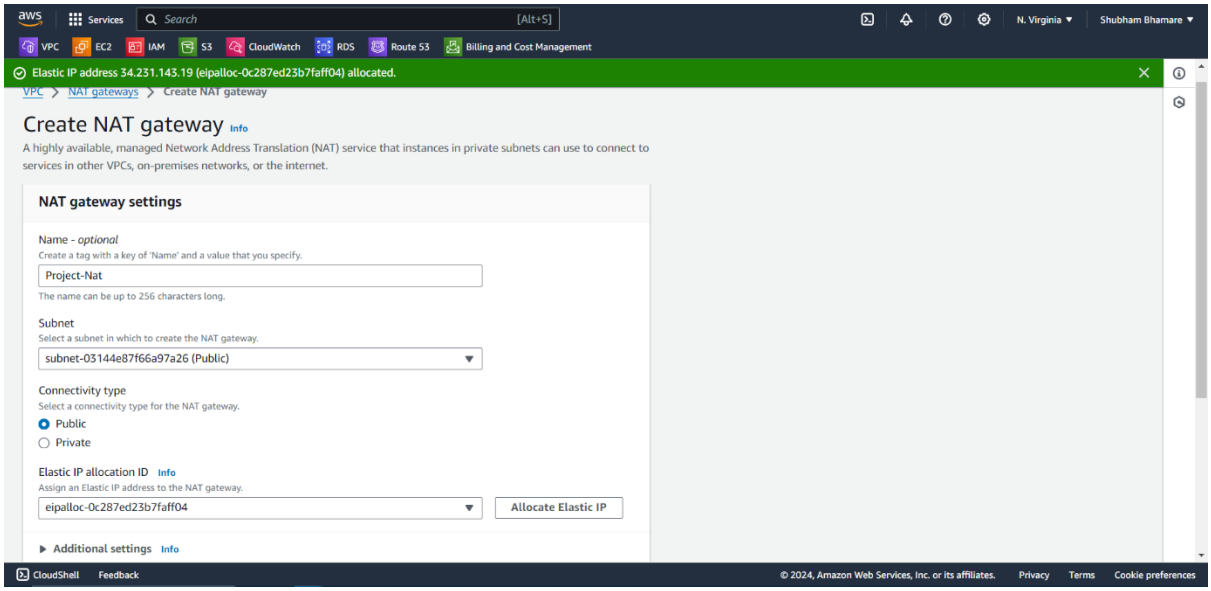
Step 11: Select your VPC and click on “Attach Internet Gateway”.



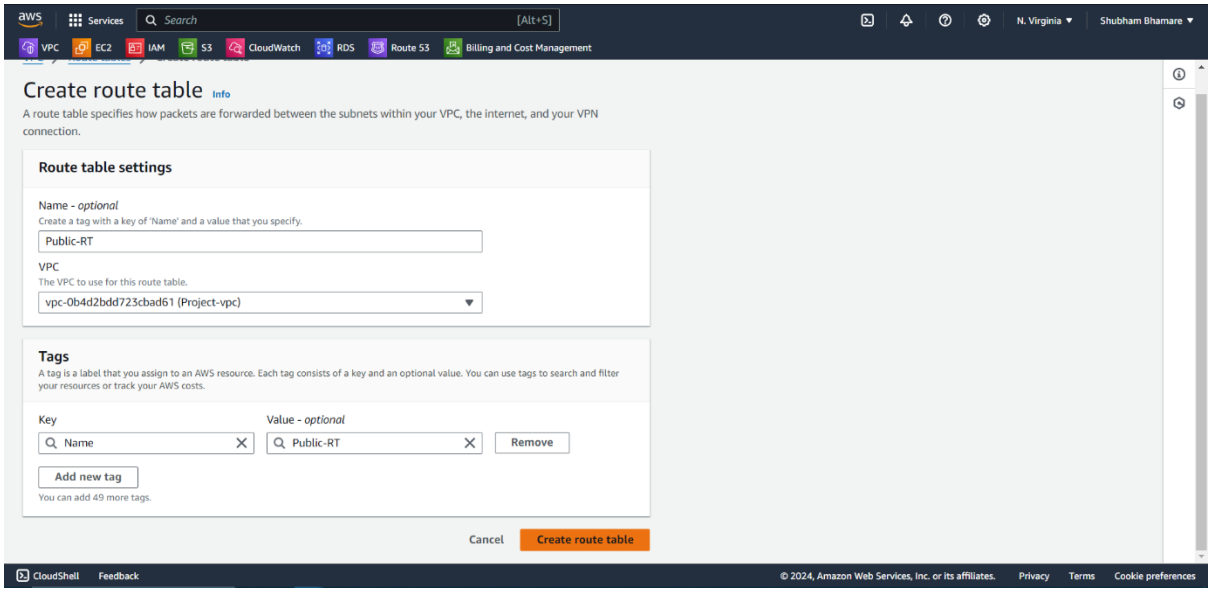
Step 12: Now go to NAT gateway and click on “Create NAT gateway”.



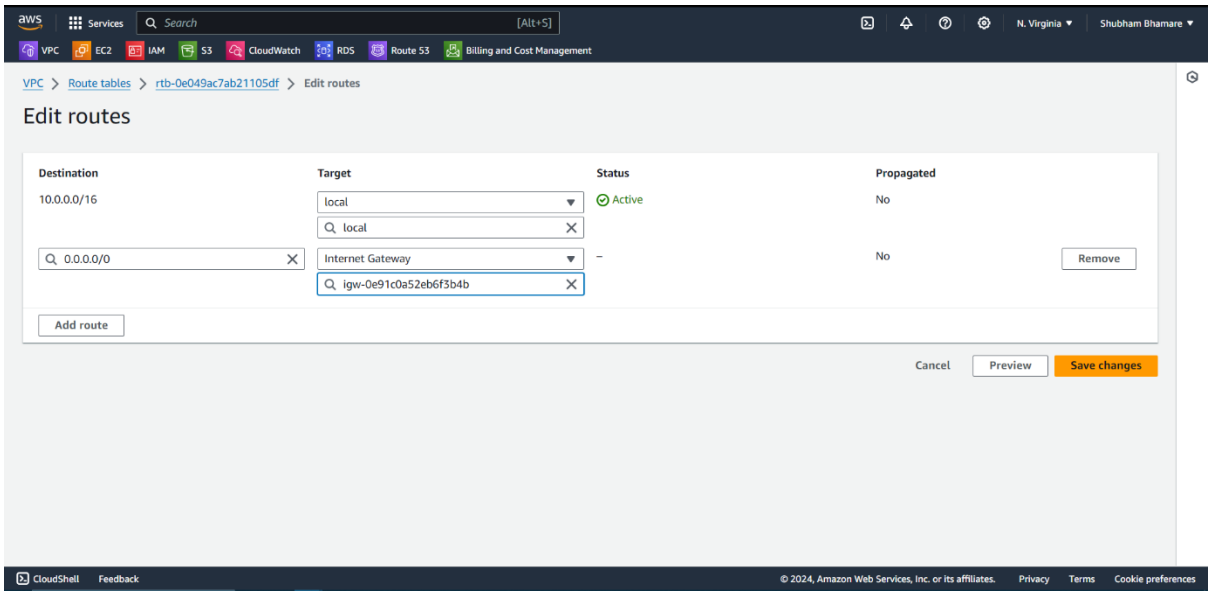
Step 13: Specify a name to your NAT gateway , In subnet select the private subnet because NAT gateway is accessing internet through that public subnet, Allocate elastic IP to your NAT gateway.



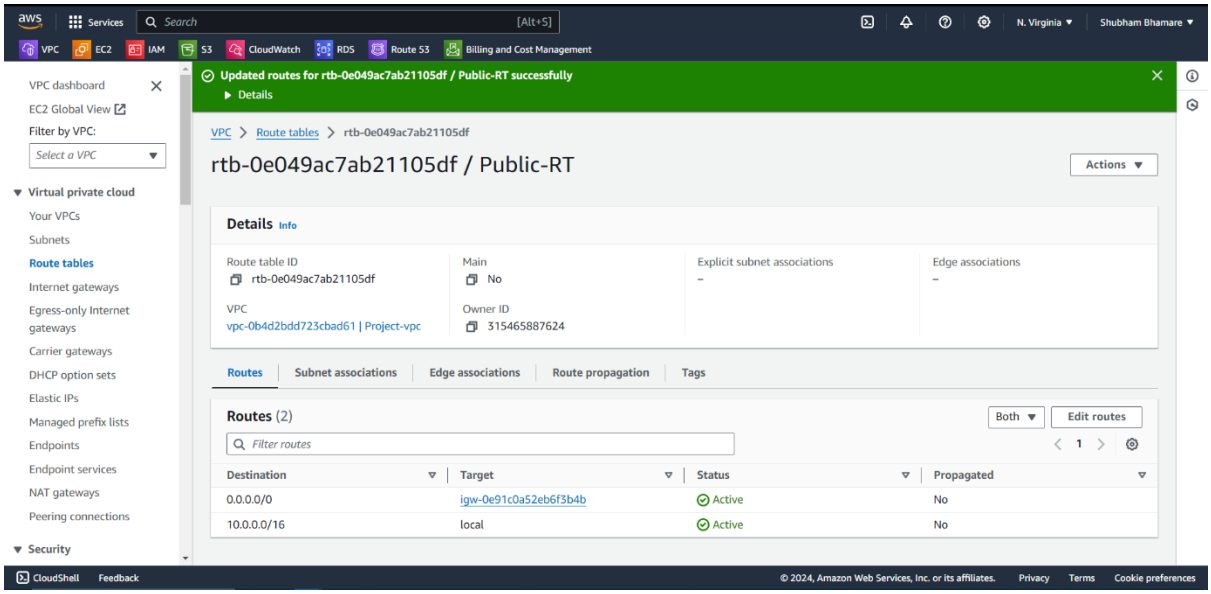
Step 14: Now go to Route table and create one public route table.



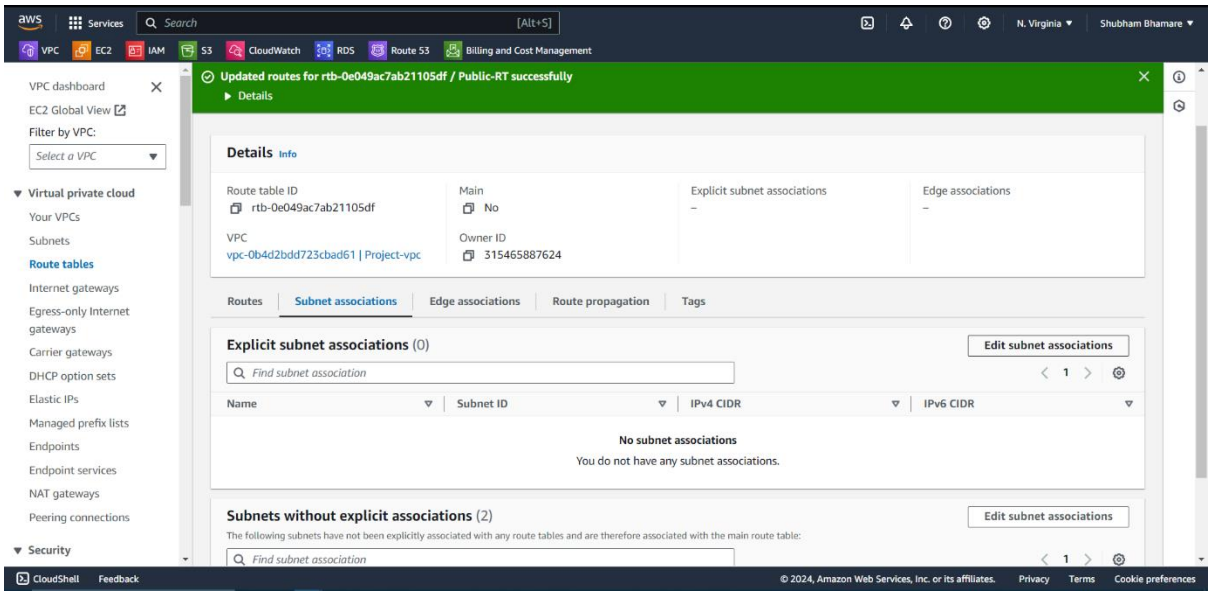
Step 15: Now go to edit routes of the public route table and add Internet gateway to your public route table.



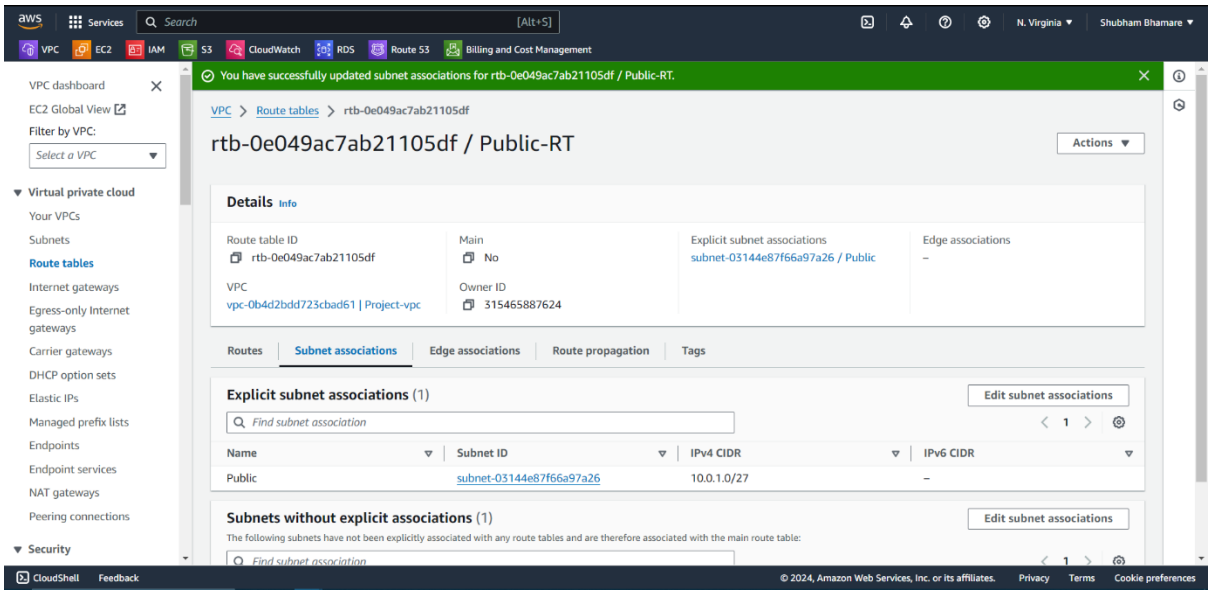
Step 16: Internet gateway is attached successfully to your public route table.



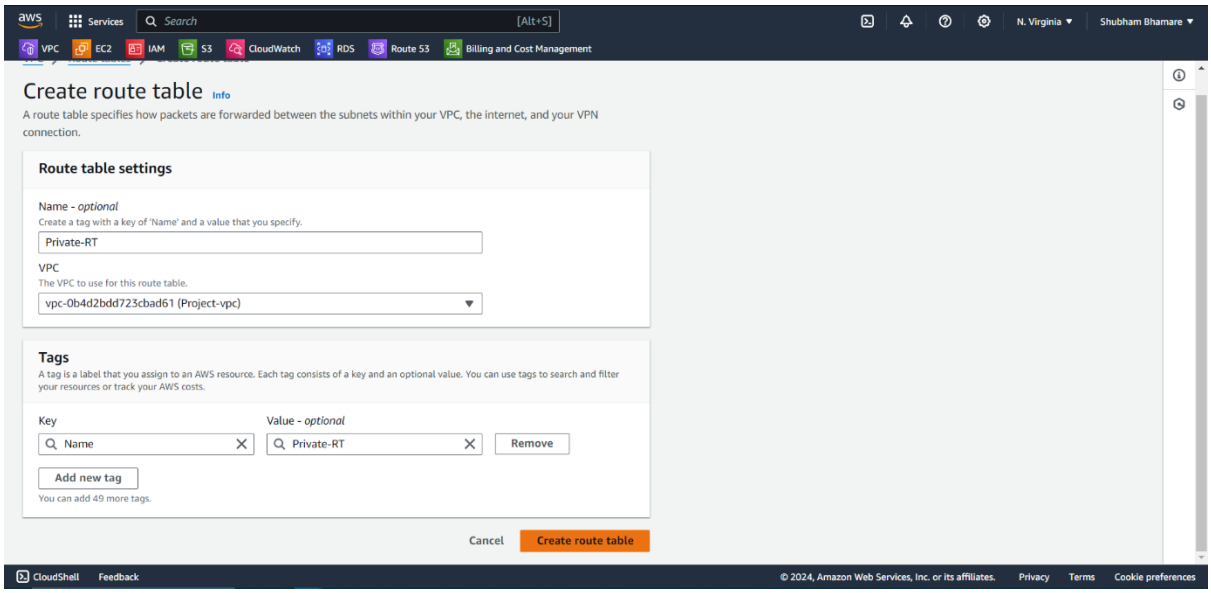
Step 17: Now go to subnet association of public route table and click on “Edit subnet association”.



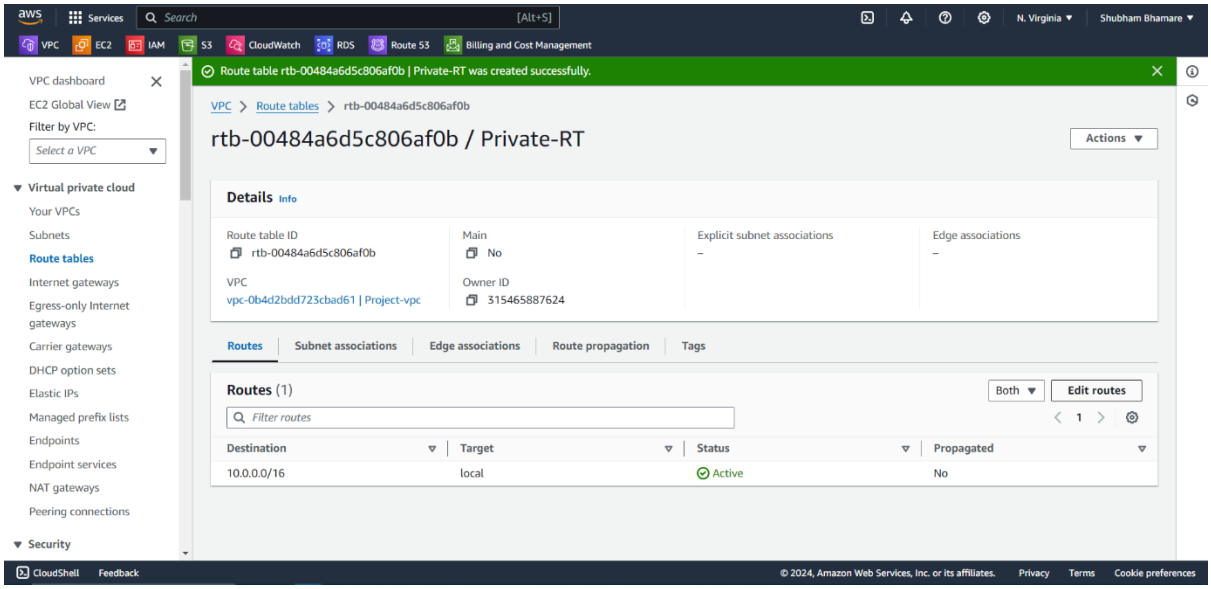
Step 18: Public subnet is associates to your public route table.



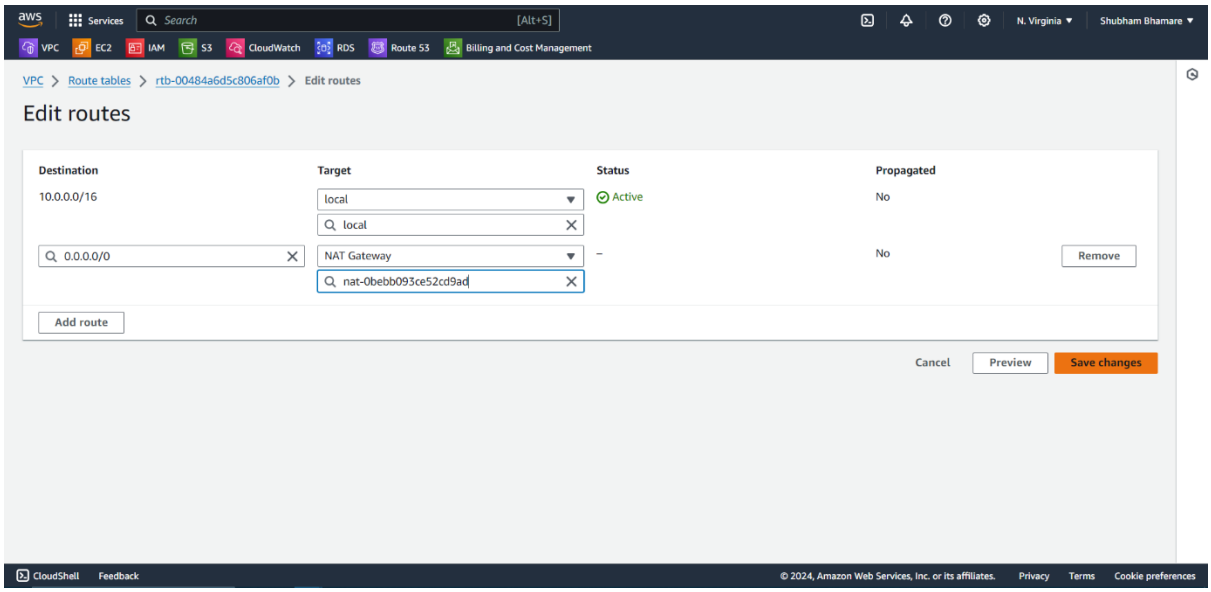
Step 19: Again create one private route table.



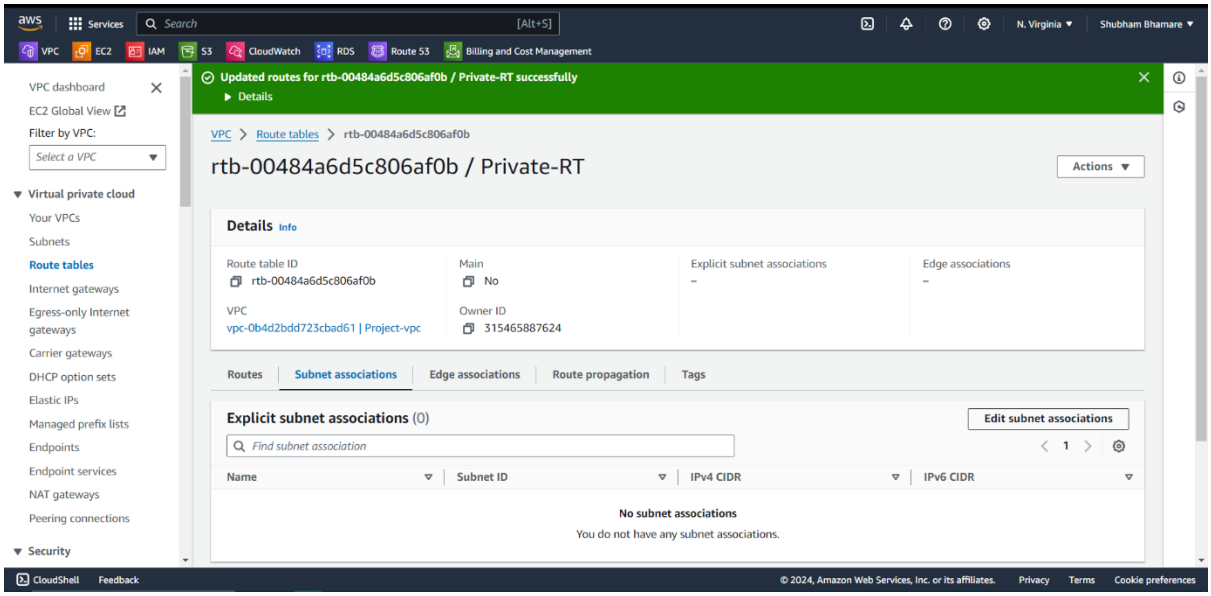
Step 20 : In private route table, go to Routes and click on “Edit routes”.



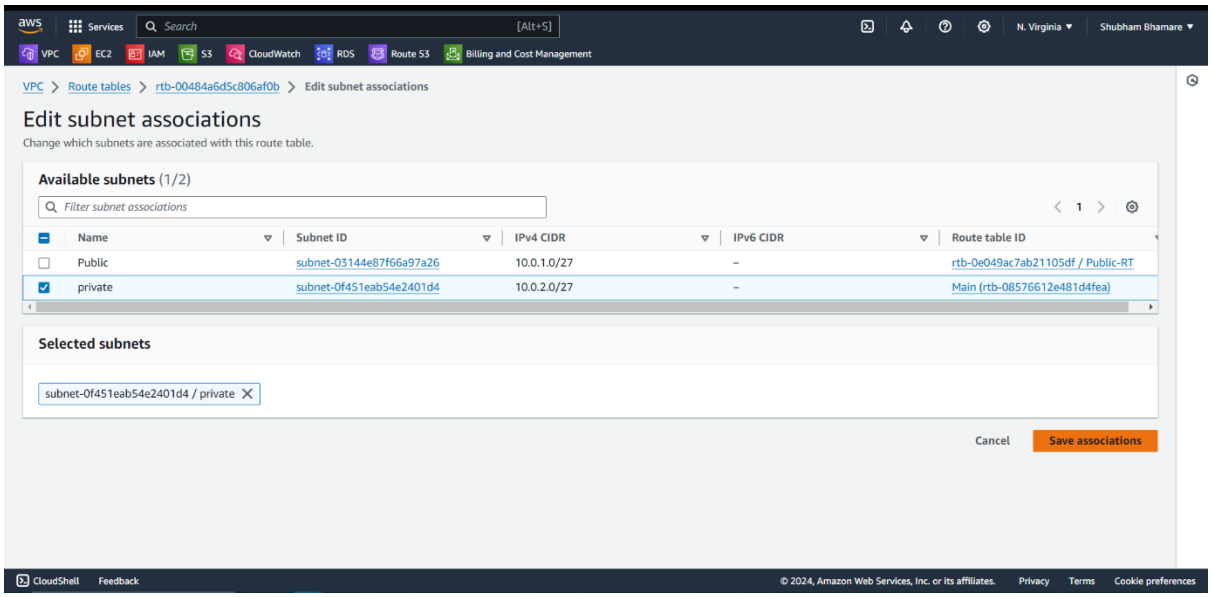
Step 21: Attach NAT gateway to your private route table.



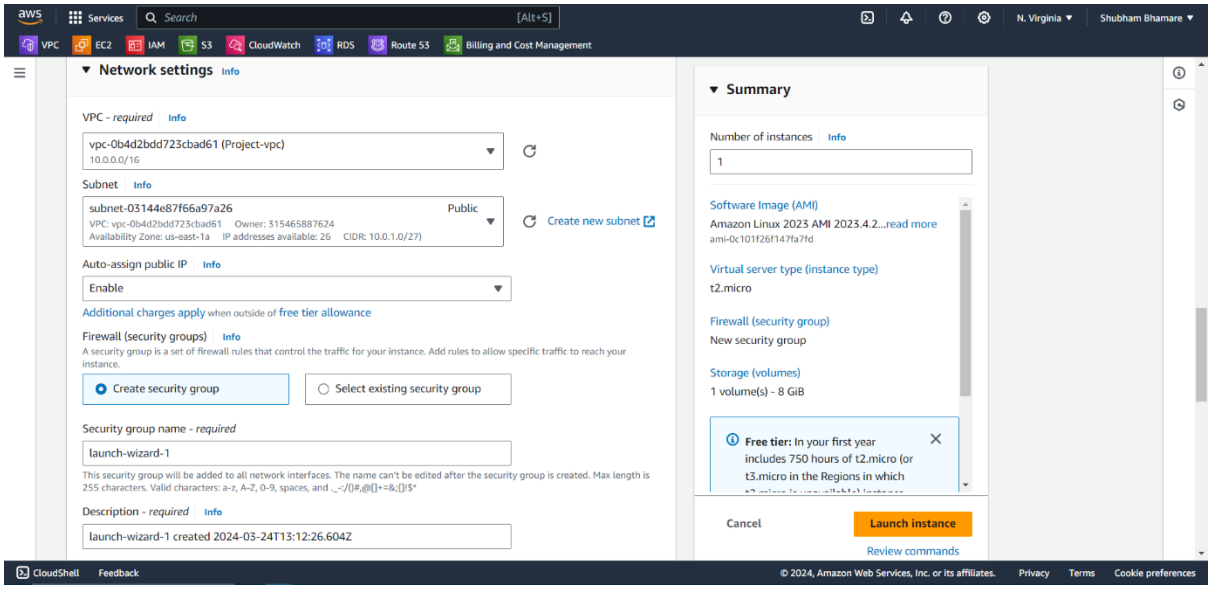
Step 22: Now go to subnet association of your private route table and click on “Edit subnet association”.



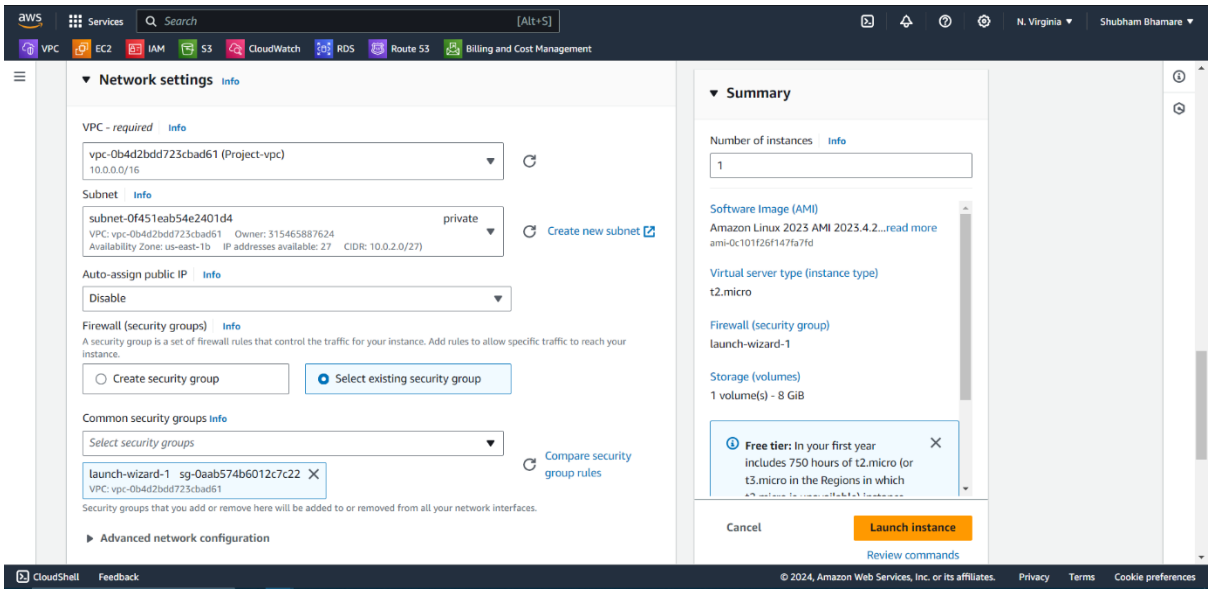
Step 23: Attach private subnet to private route table.



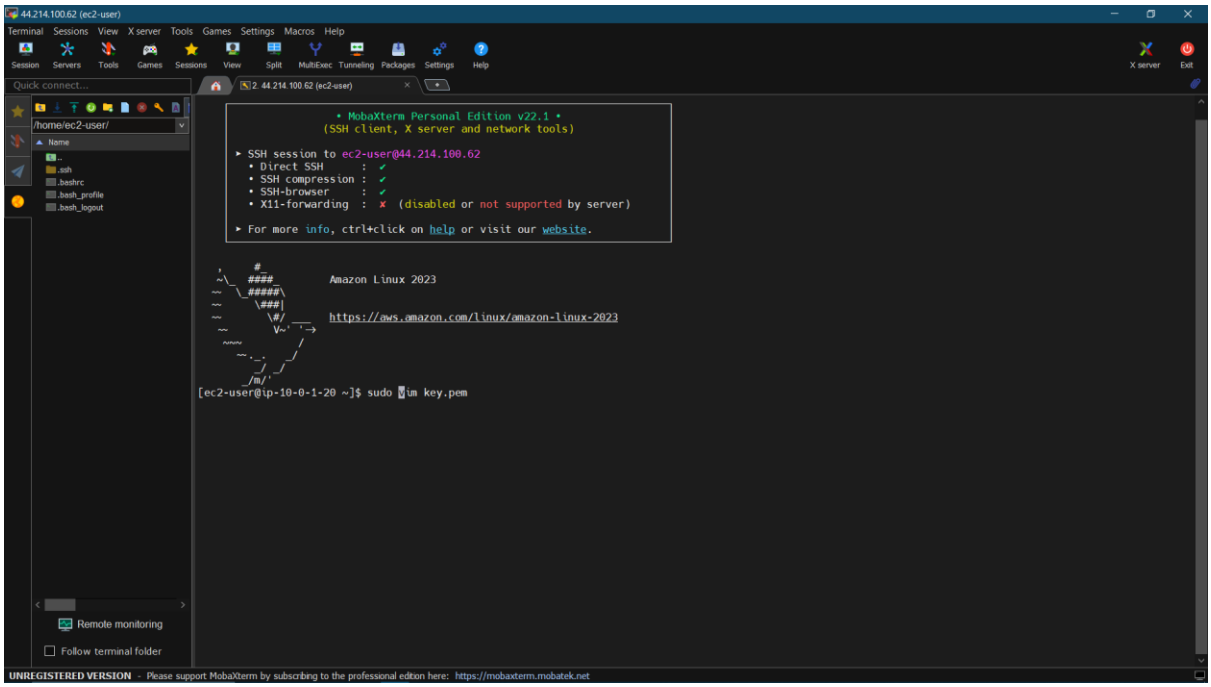
Step 24: Now Go to EC2 service and launch one public instance with public subnet and enable public IP.



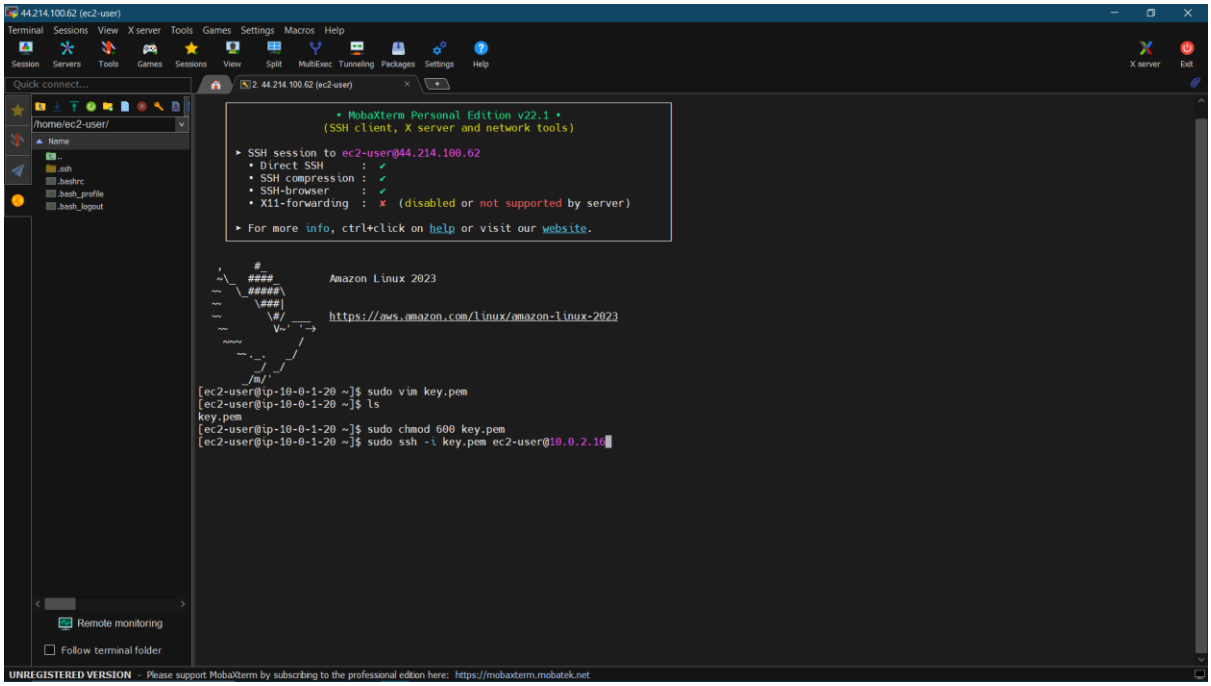
Step 25: Again launch one private instance in your private subnet.



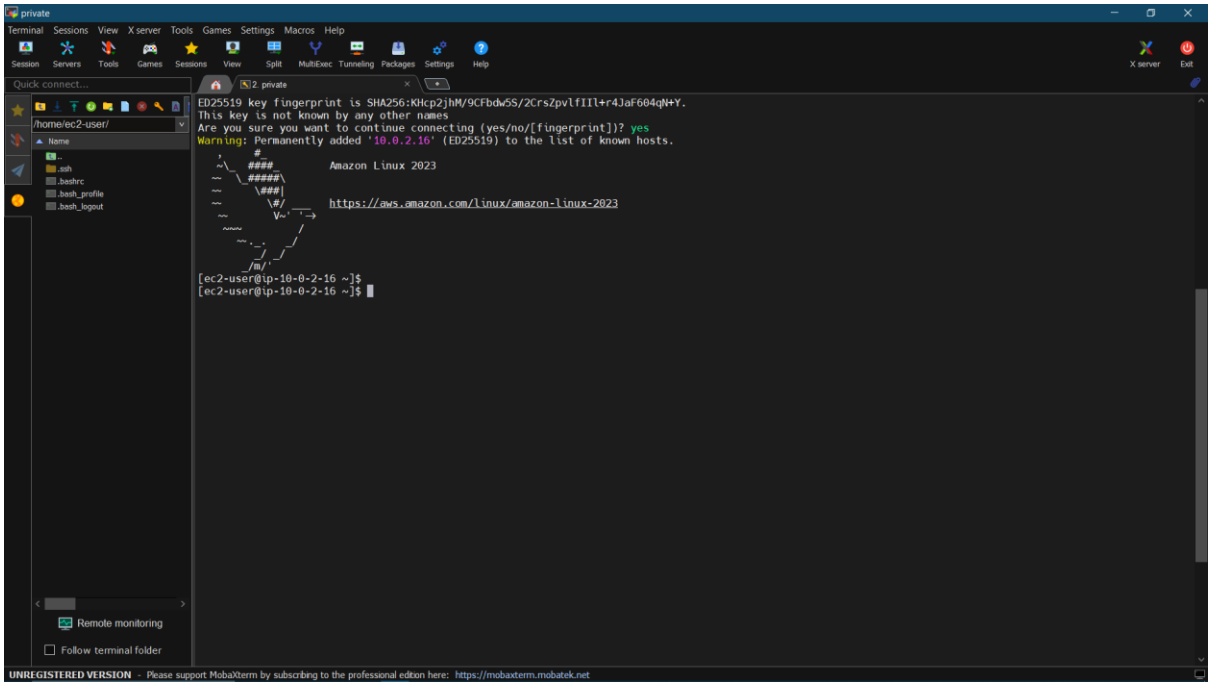
Step 26: Take SSH to your public Instance and create one key file and paste the content of instance key into that file for taking SSH access to your private instance.



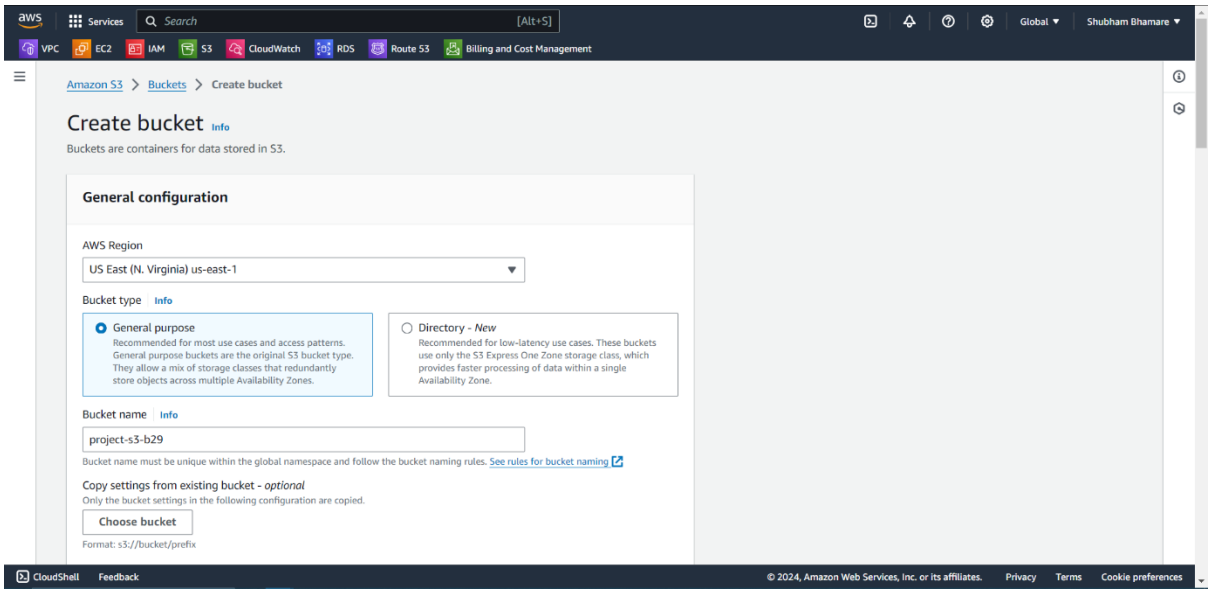
Step 27: Key is created successfully. Now change the permission for that key using “chmod 600 <filename>” command. Then run command “sudo ssh -I <keyfile_name> ec2-user@<private_ip_private_Instance>”.



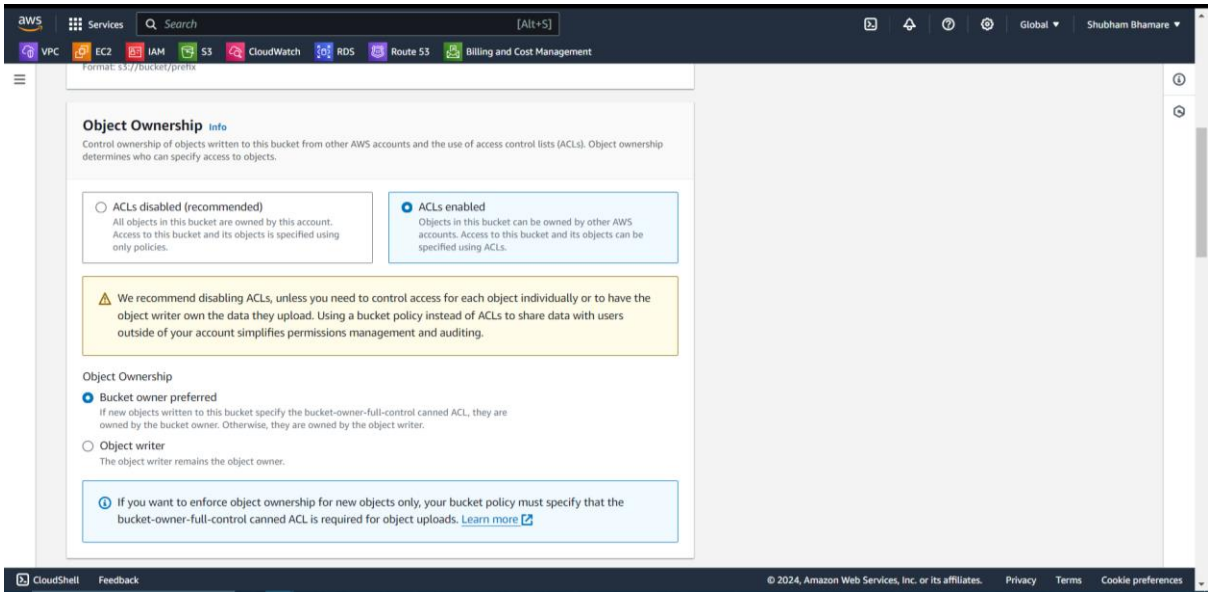
Step 28 : Now you are logged into your private instance.



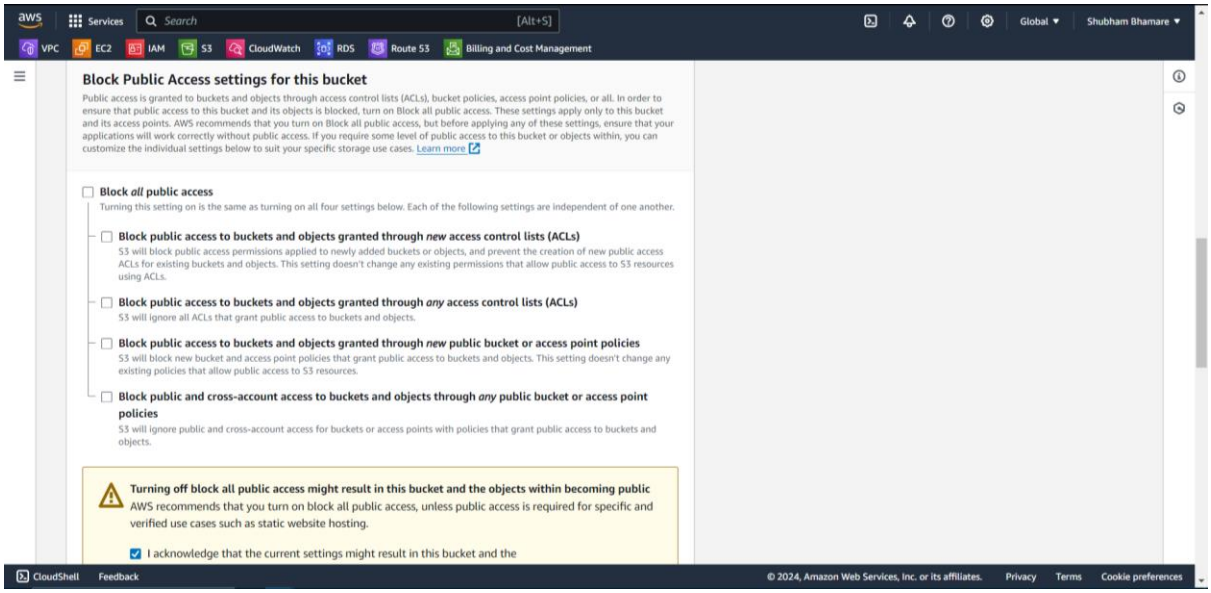
Step 29: Go to S3 bucket and create one public bucket.



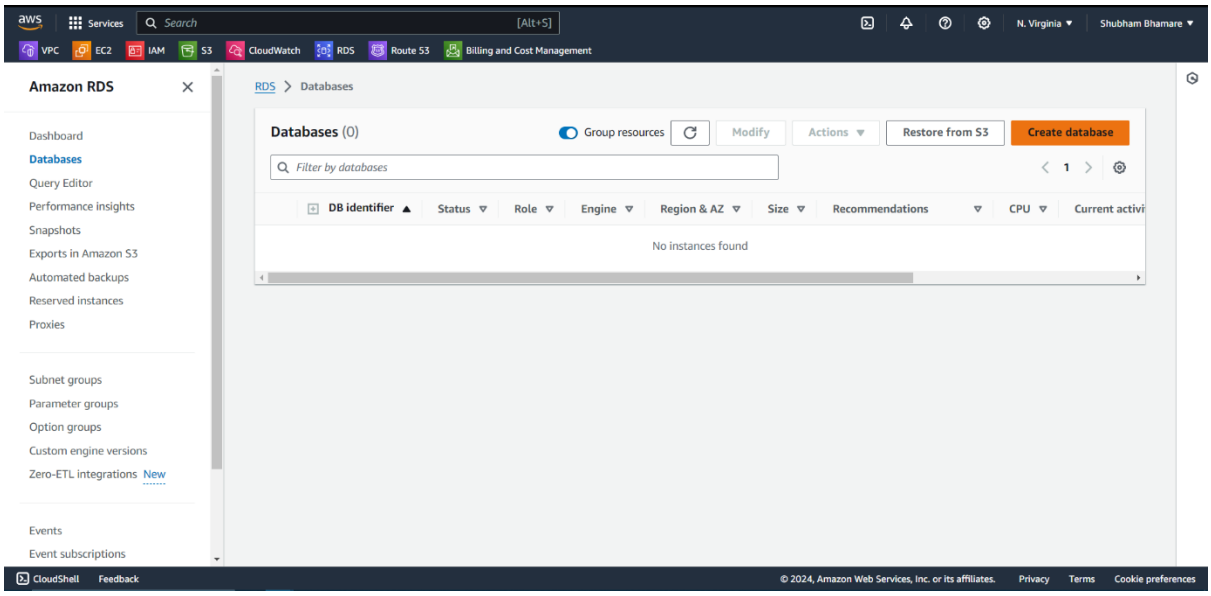
Step 30 : Enable ACLs.



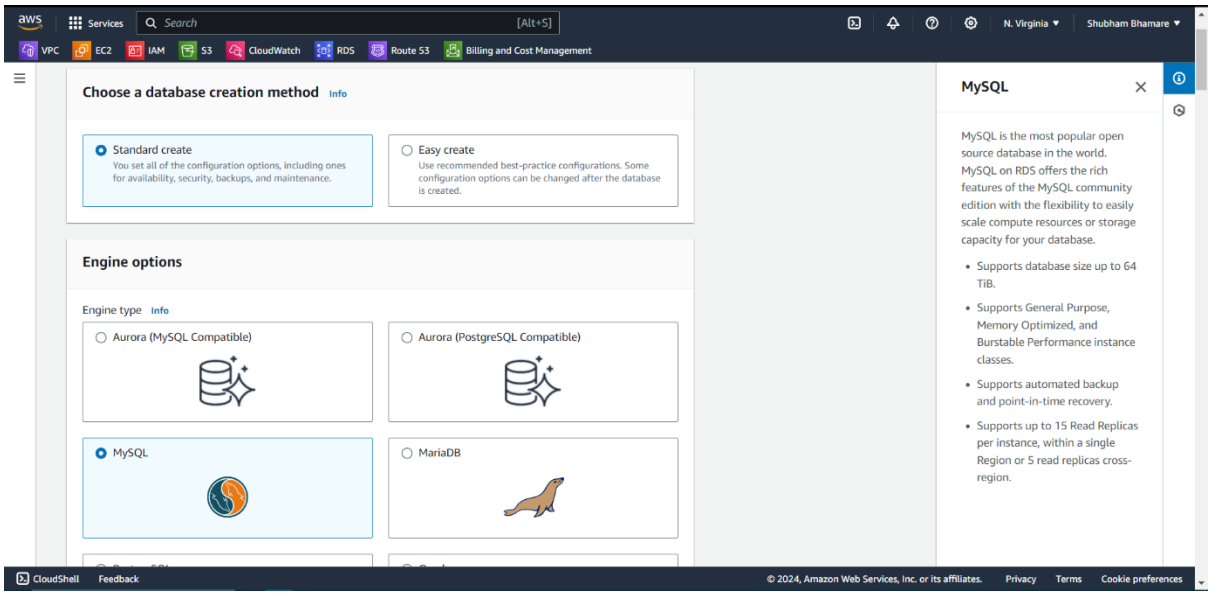
Step 31: Untick “block all public access”, and create bucket.



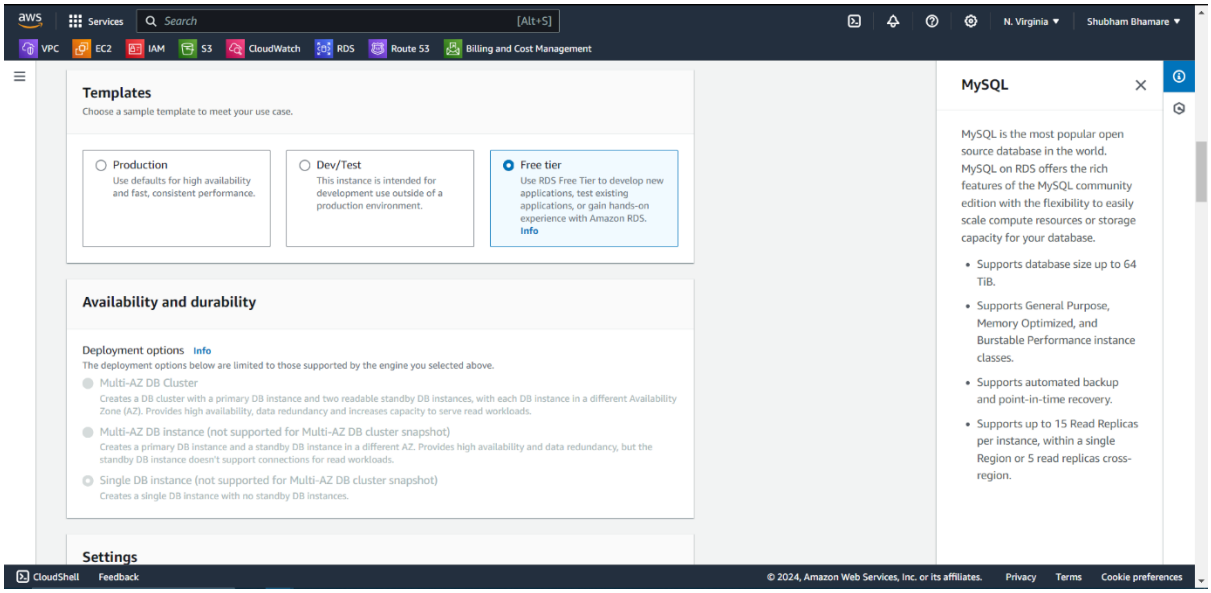
Step 32: Go to RDS and click on “Create database”.



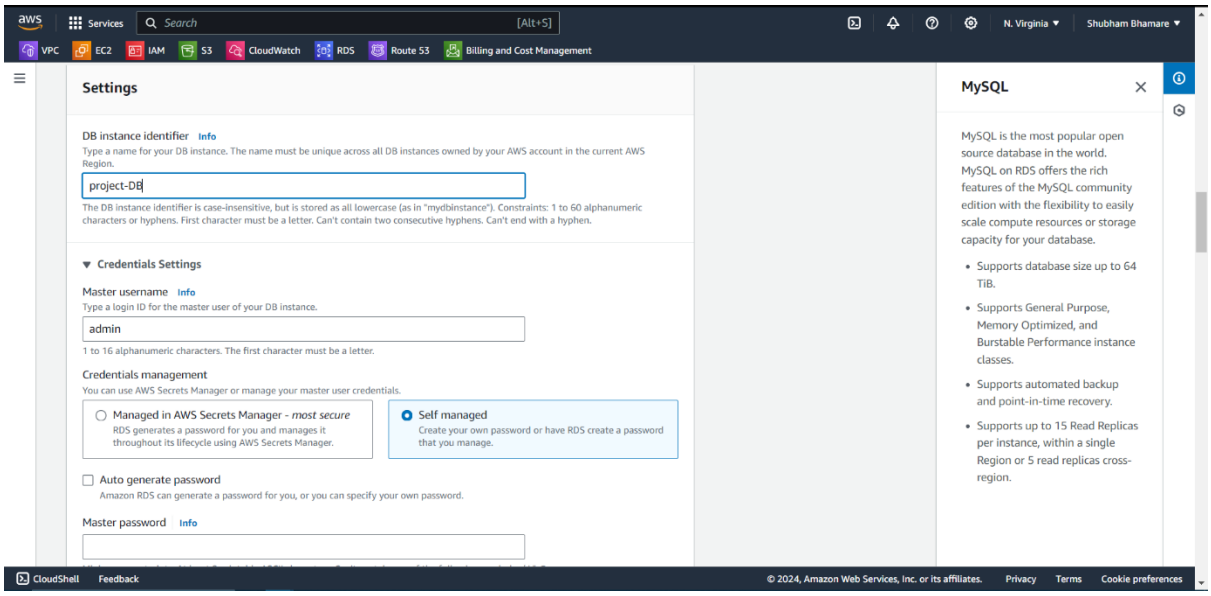
Step 33: Select the engine for your database.



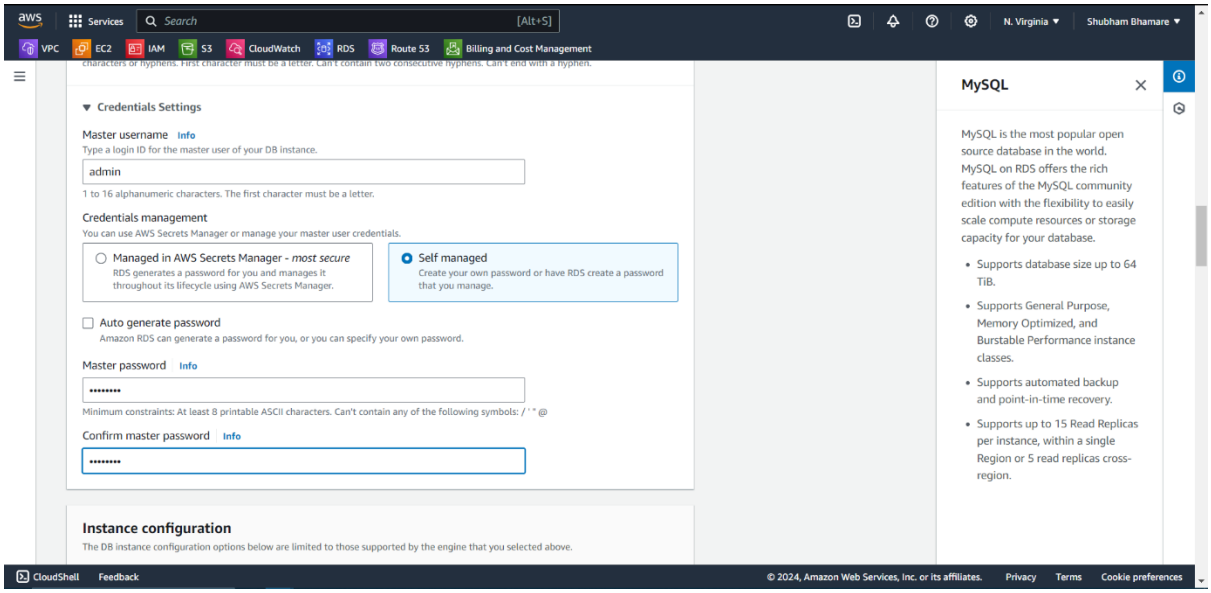
Step 34: Select a template as per your requirement.



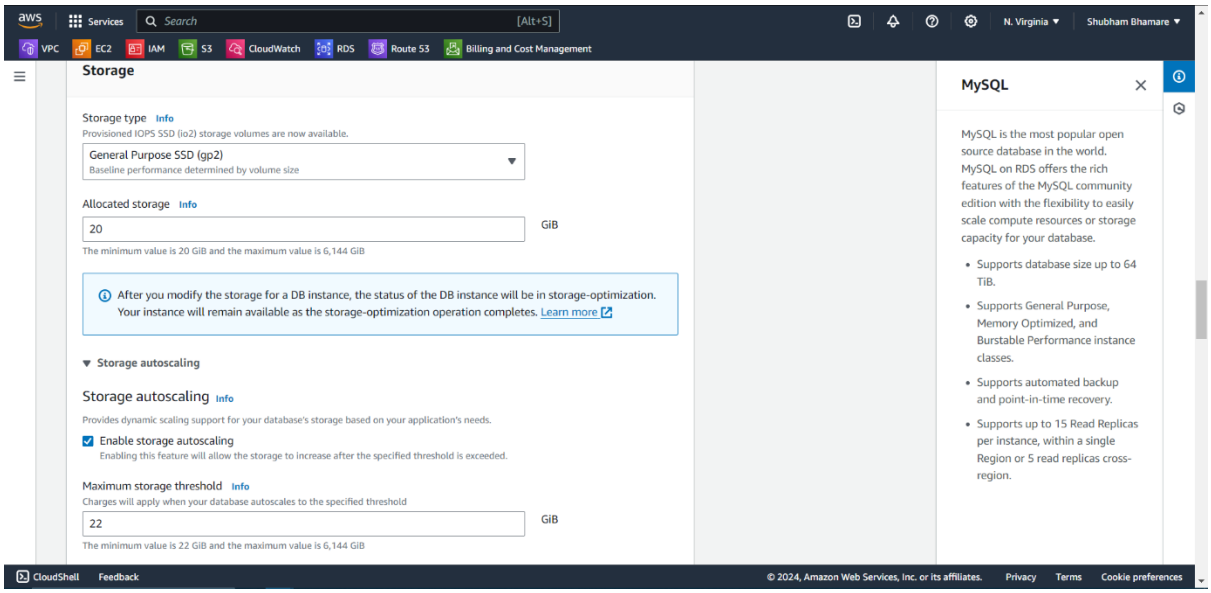
Step 35: Specify name for your DB instance.



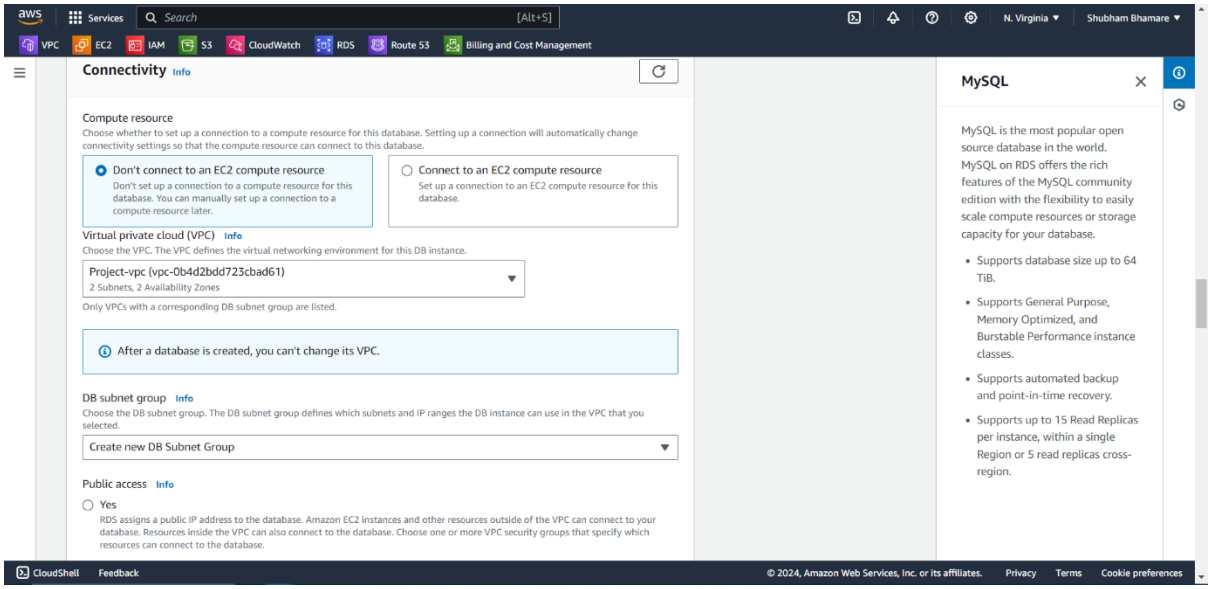
Step 36: Specify username and password for your DB.



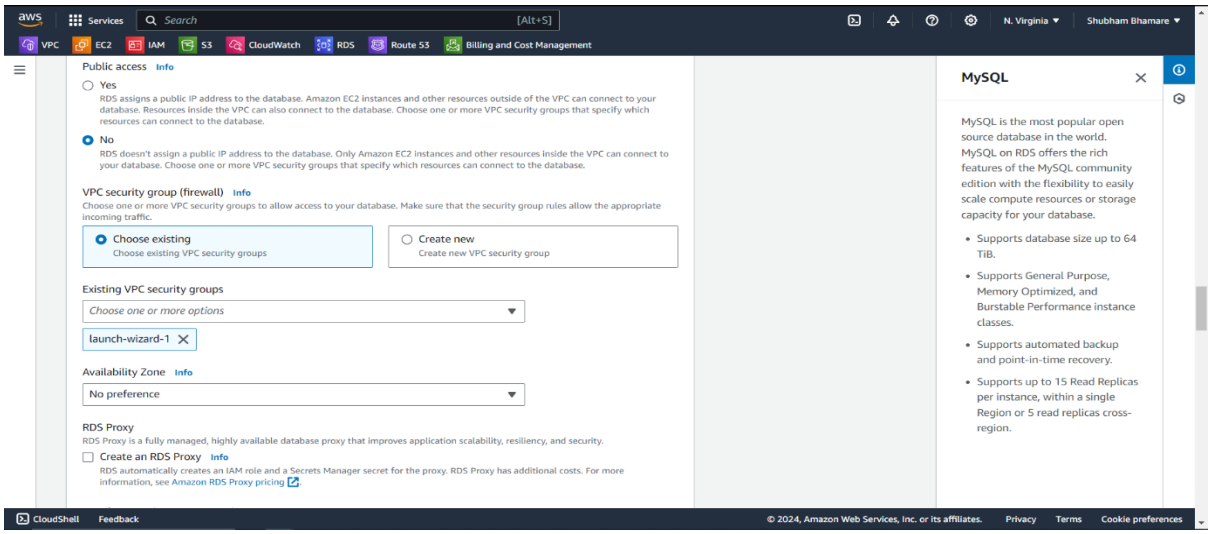
Step 37: You can specify storage as per your requirement.



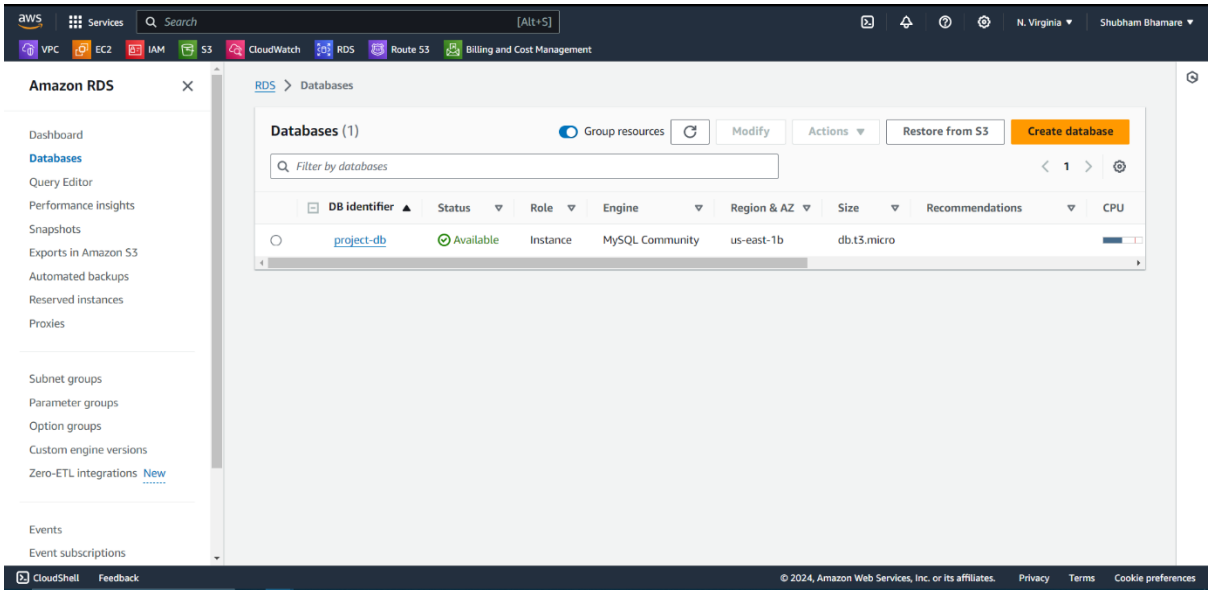
Step 38: In connectivity select your VPC.



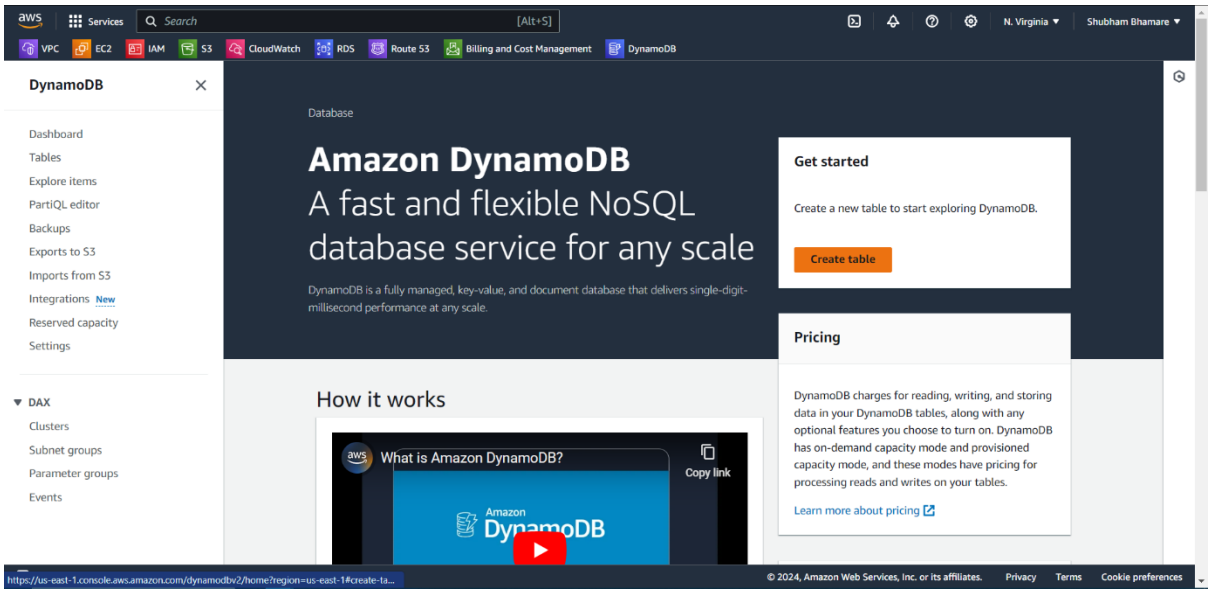
Step 39: Restrict public access for your DB.



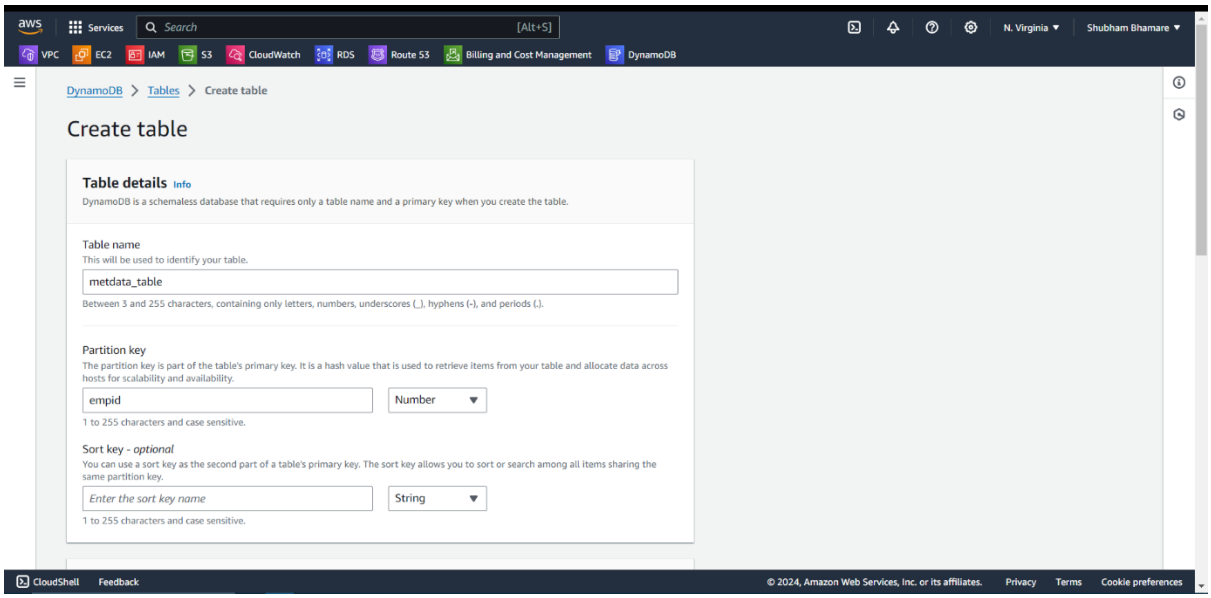
Step 40: Your database is created successfully.



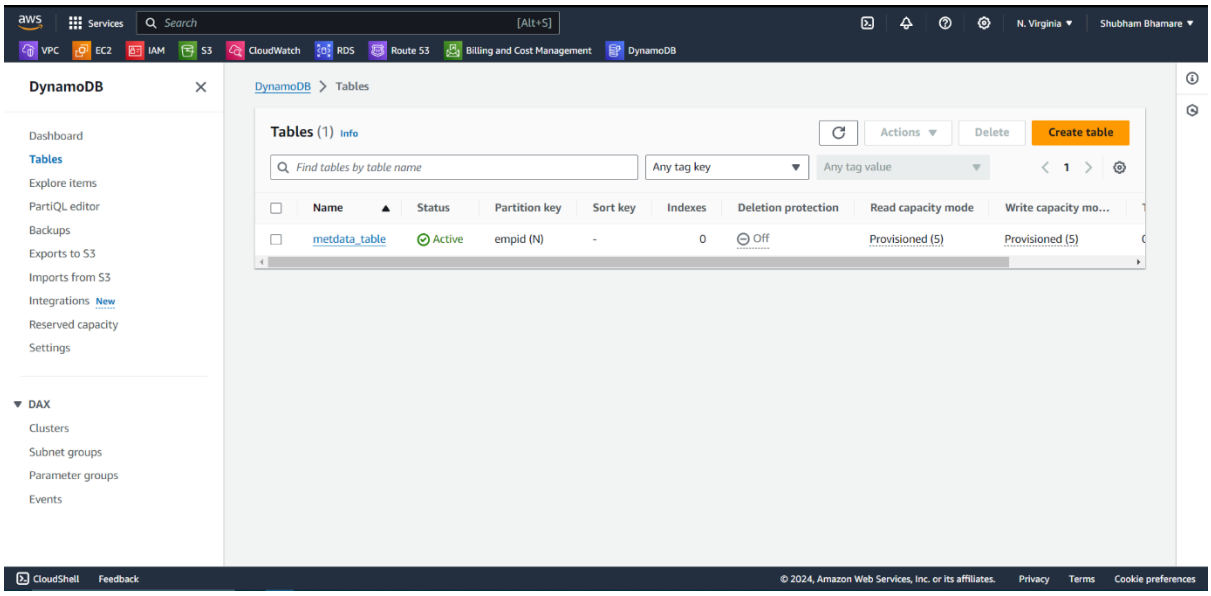
Step 41: Now to do DynamoDB and click on “create table”.



Step 42: Specify Name for your table, select partition key and datatype for that partition key.

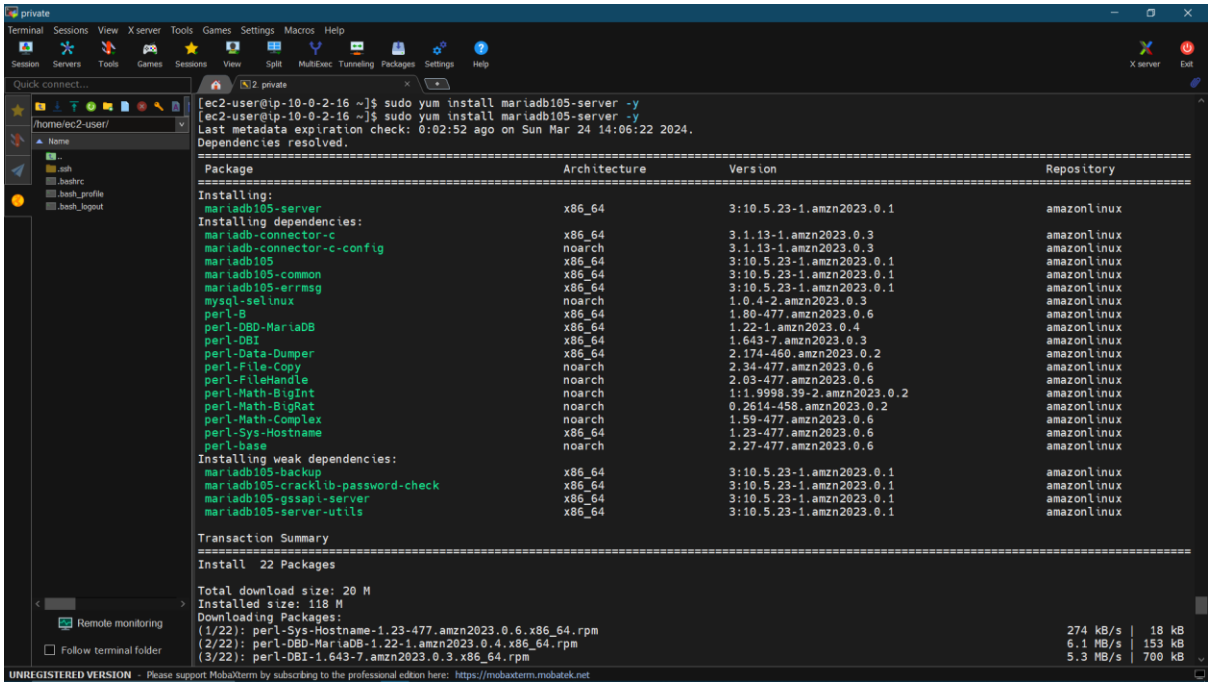


Step 43: Table is created successfully.



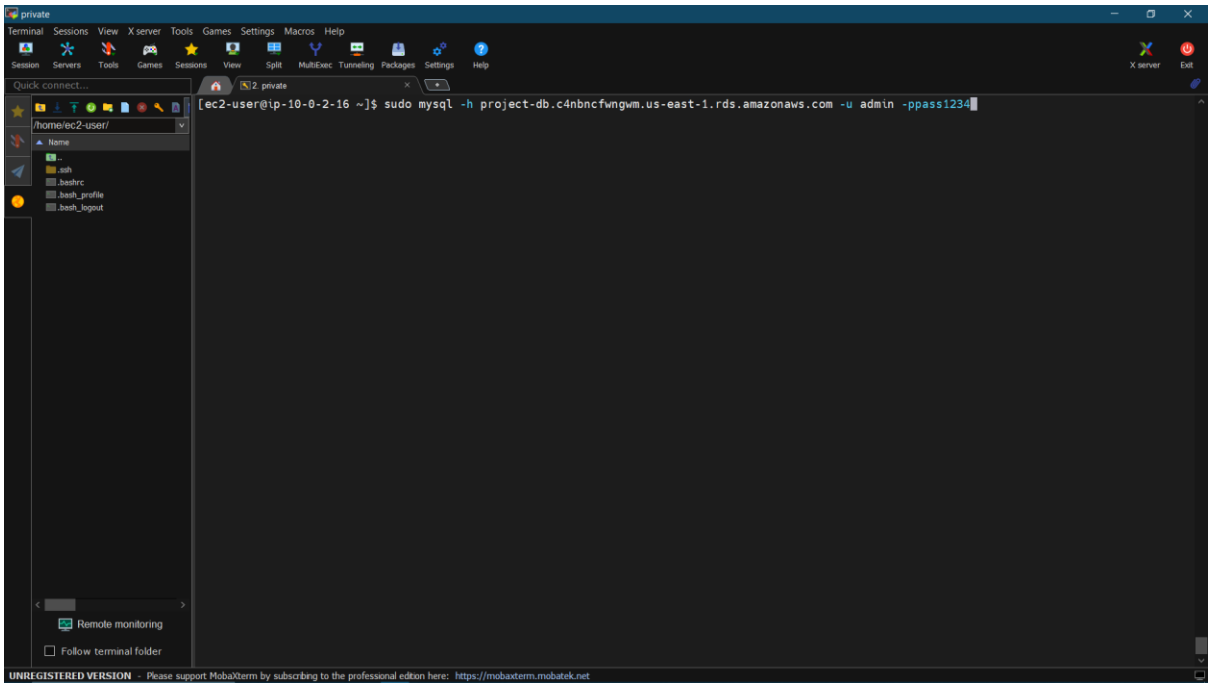
Step 44: Now go to your private instance and install mariadb using

“sudo yum install mariadb105-server -y”.

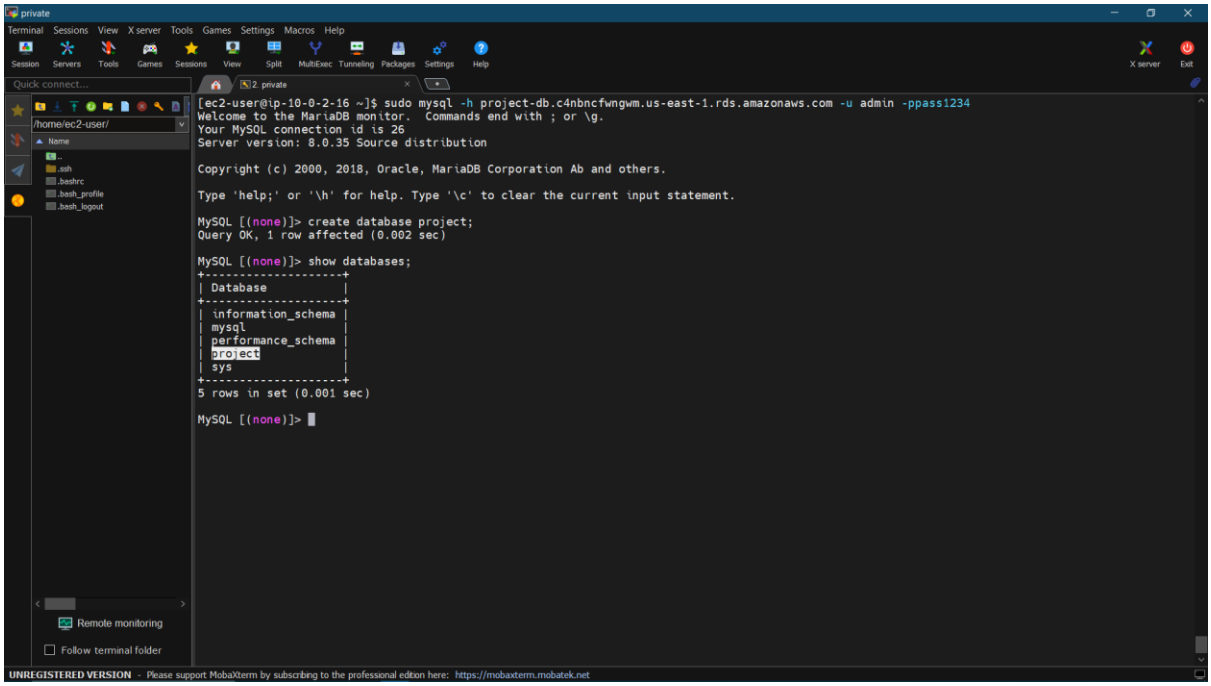


Step 45: Now login into your database using command

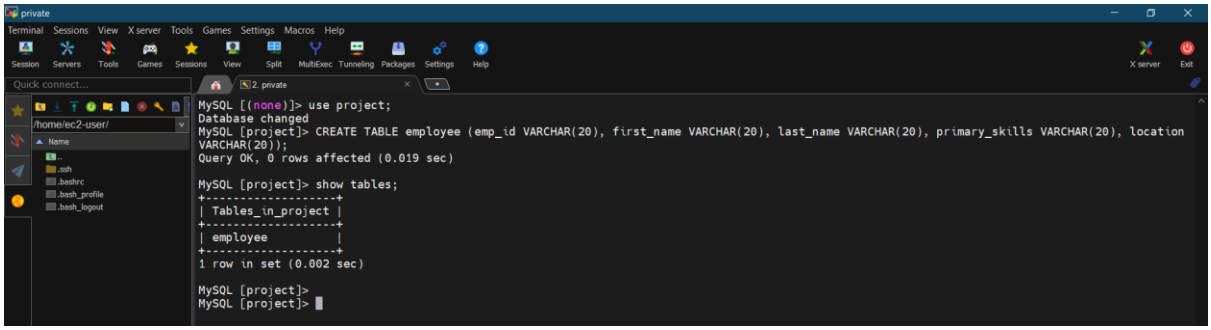
“sudo mysql -h <DBendpoint> -u <username> -p<password>”.



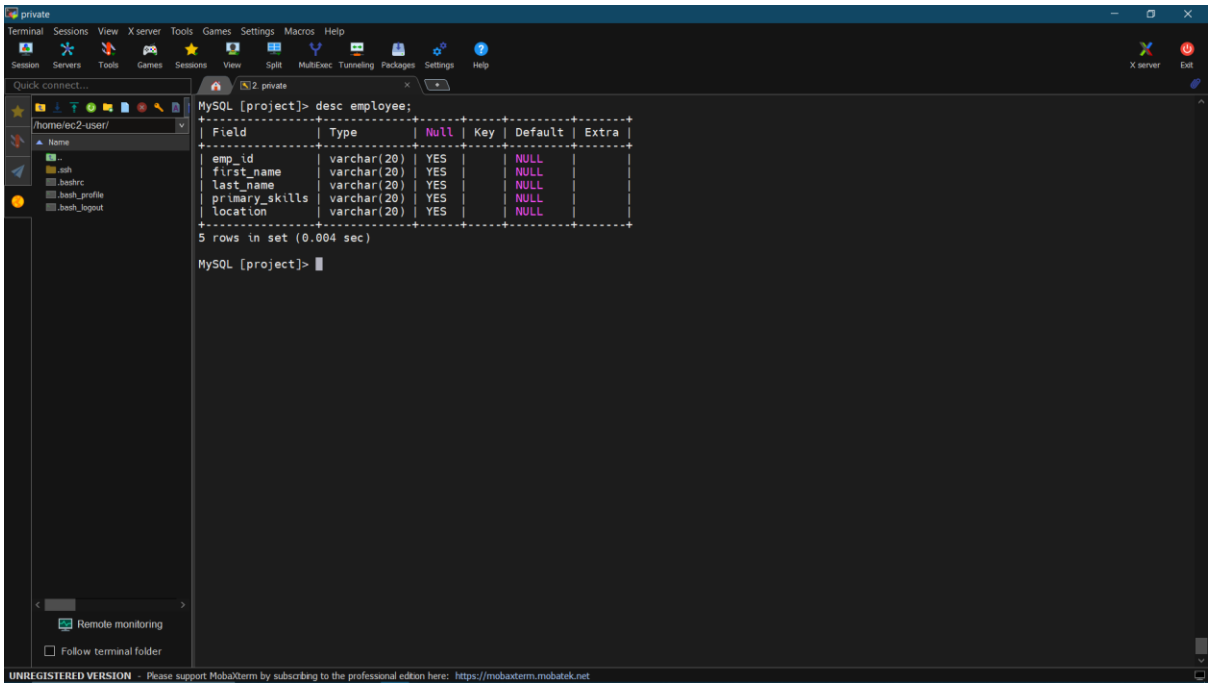
Step 46: Run command “create database <DBname>;” to create DB and run command “show databases;” to list all databases.



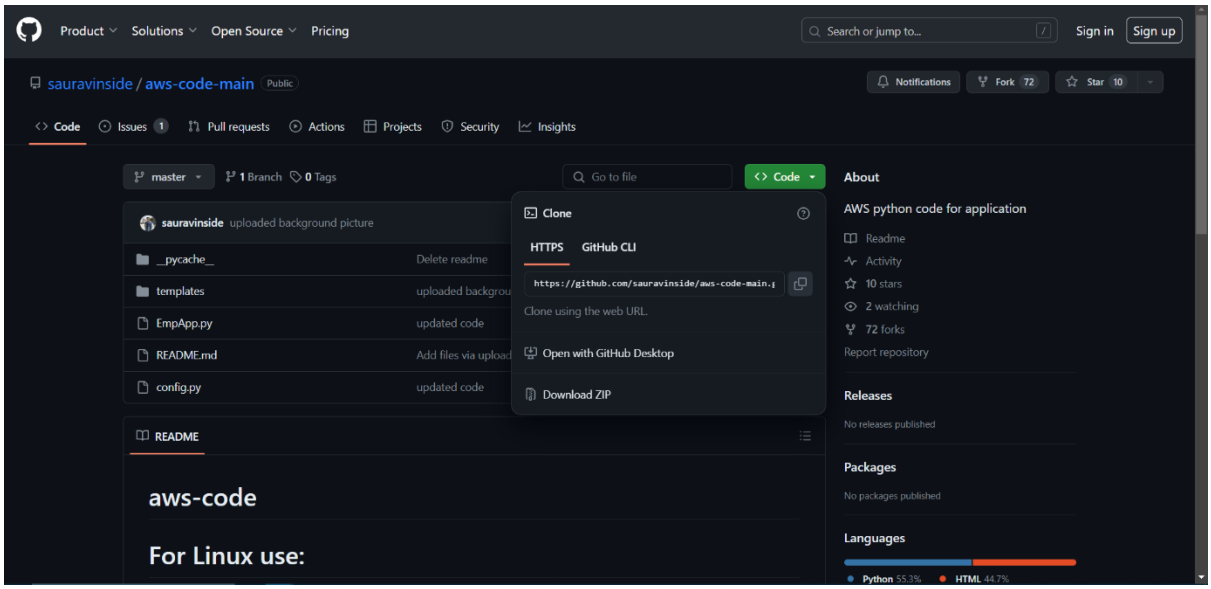
Step 47: Now run command “use <db-name>;” to enter into that database then run command “CREATE TABLE employee (emp_id VARCHAR(20), first_name VARCHAR(20), last_name VARCHAR (20), primary_skills VARCHAR(20), location VARCHAR(20));” to create table., Now run command “show tables;” to list all tables in your DB.



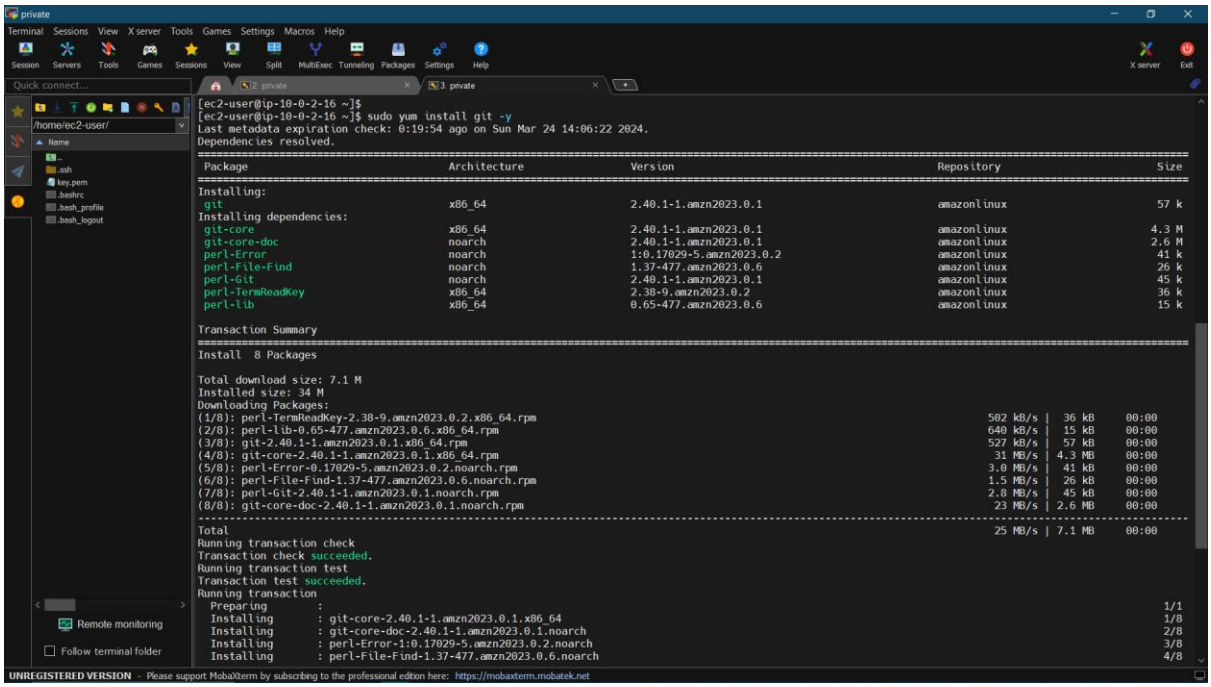
Step 48: Run command “desc <table_name>;” to describe table.



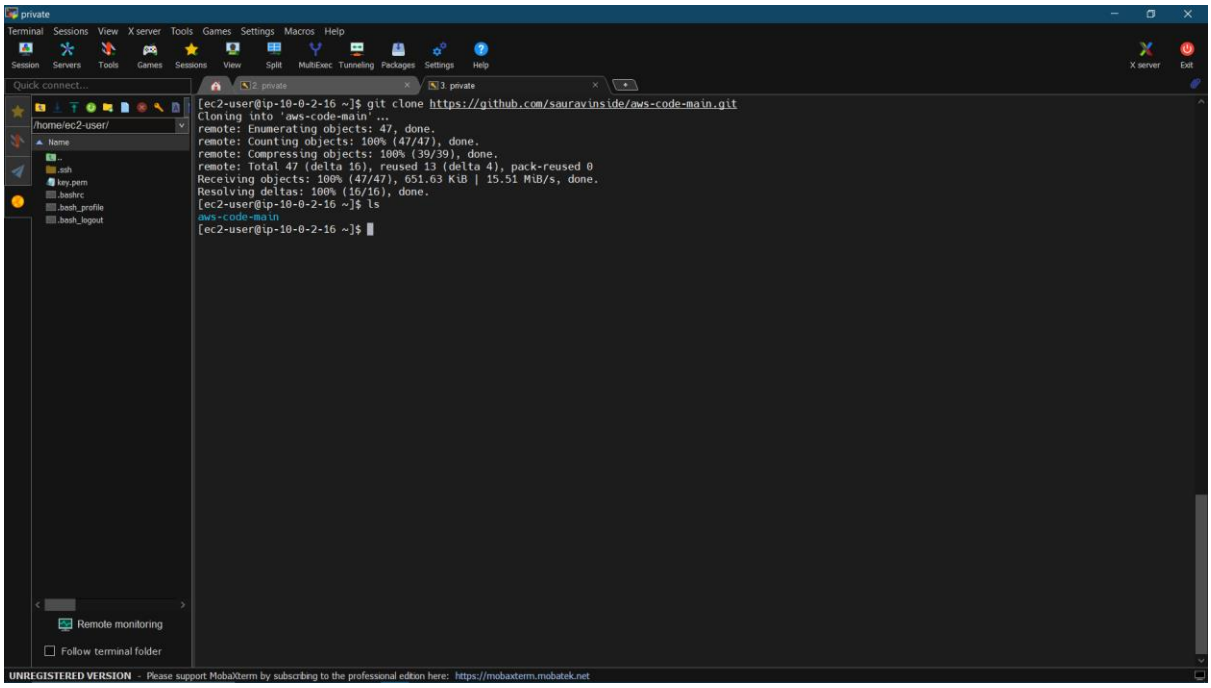
Step 49: Now go to https://github.com/sauravinside/aws-code-main then click on “code” and the url.



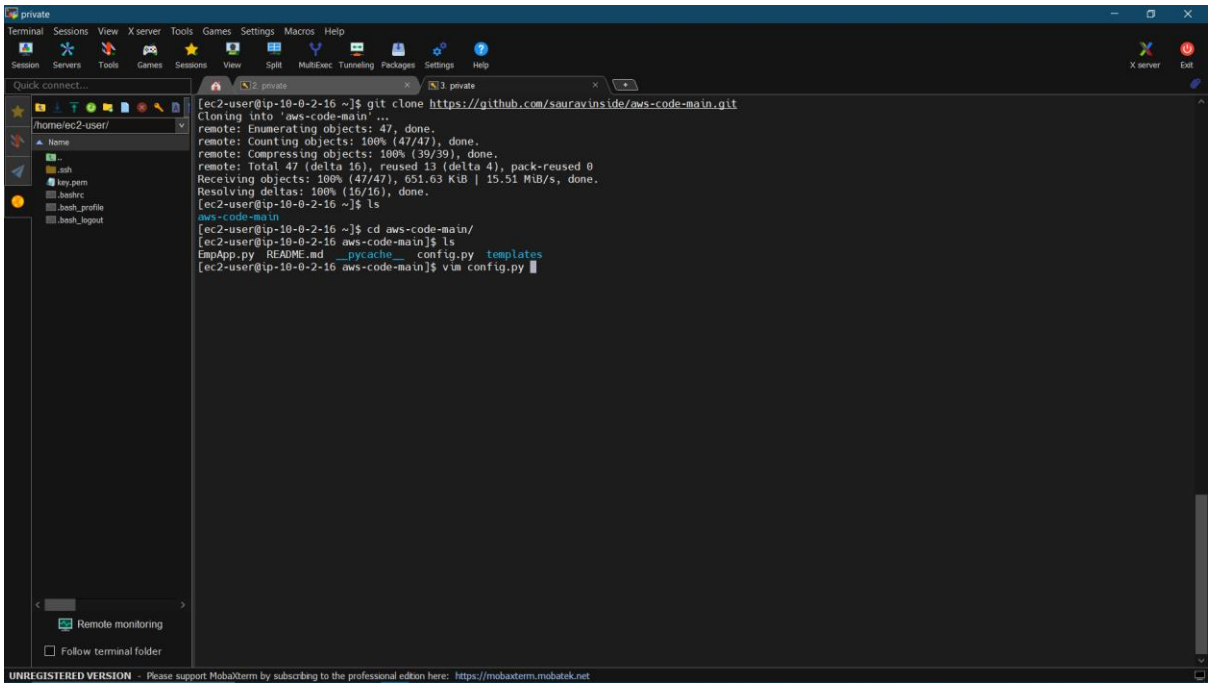
Step 50: On private instance run command “sudo yum install git -y” to install git on your private instance.



Step 51: Run command “ git clone <copied git-URL>” to download the application from git repository.



Step 52: Run command “vim config.py” to open configuration file.



Step 53: Now in configuration file change all configuration as per your resources.

In customhost -Enter your RDS endpoint

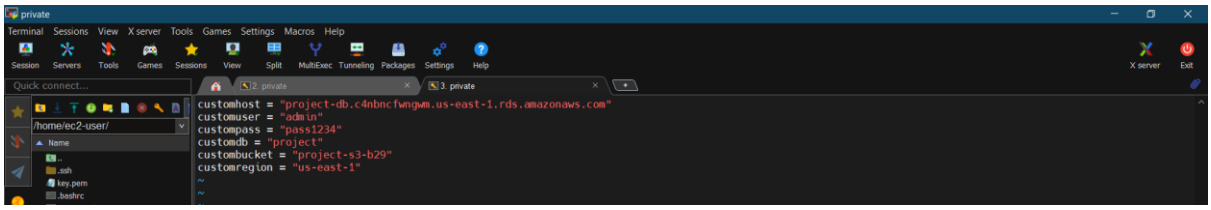
Customuser- username

Custompass- password of RDS

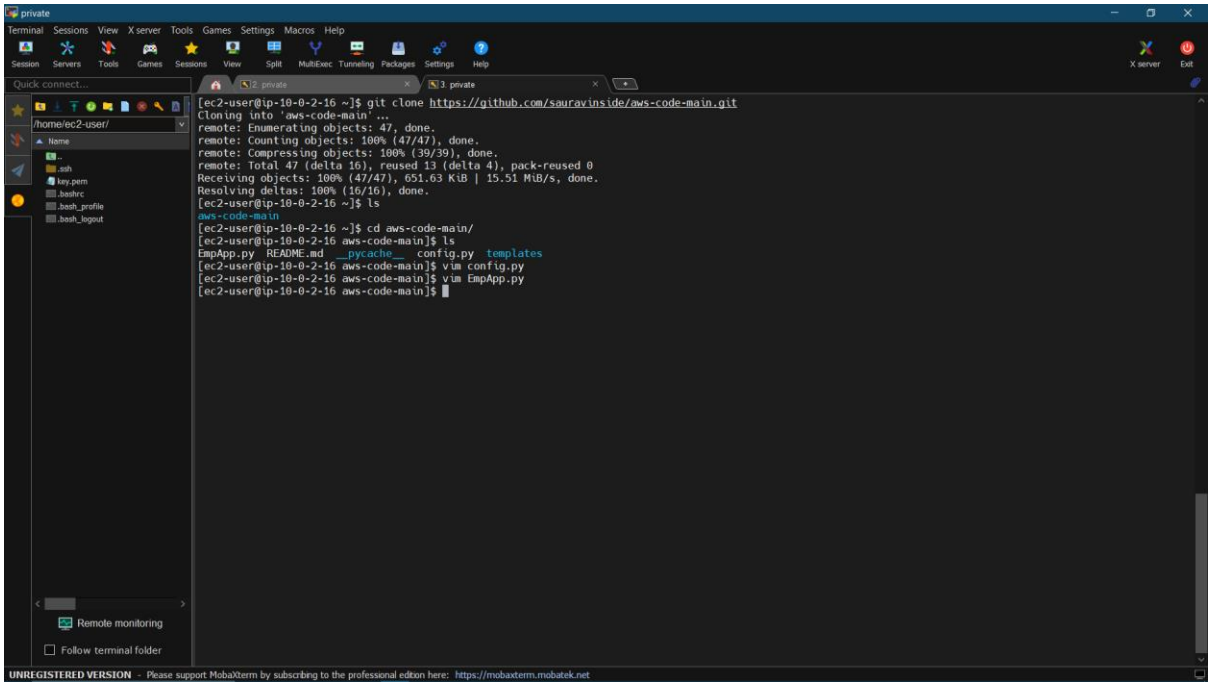
Customdb- database name in your RDS

Custombucket – S3 bucket name

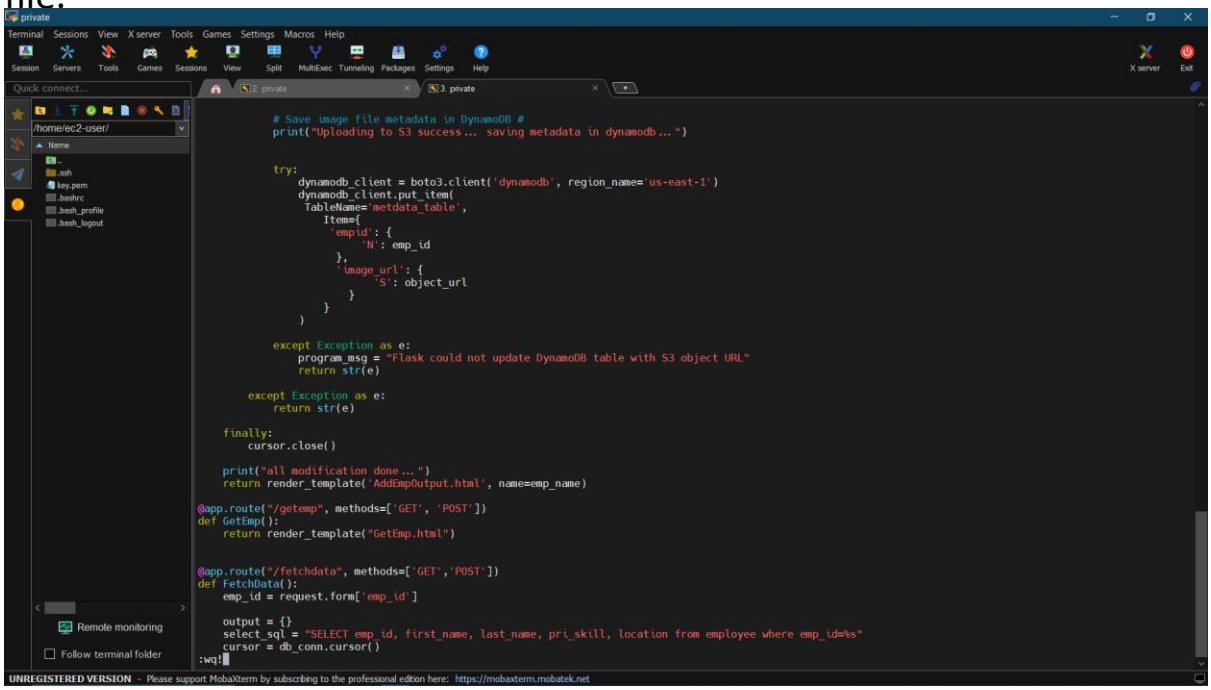
Customreigion – region in which you bucket is present and save file.



Step 54: Now run command “vim EMpApp.py” to open the DynamoDB configuration file.

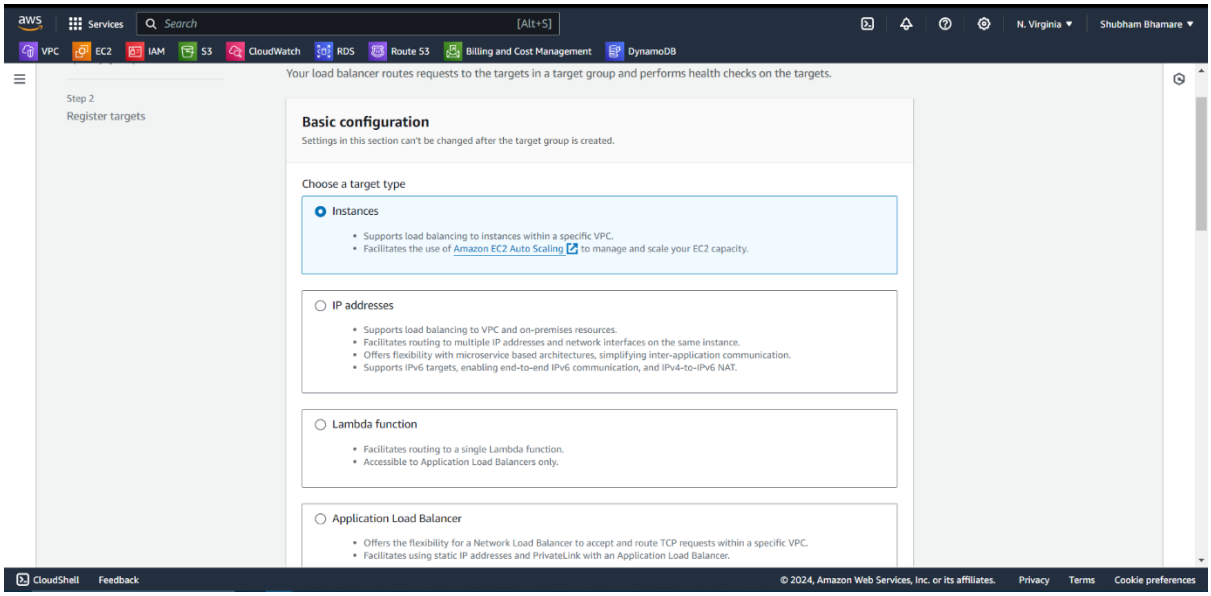


Step 55: Now in DynamoDB block change the table name to your table name which you have created in DynamoDb console and then save the file.

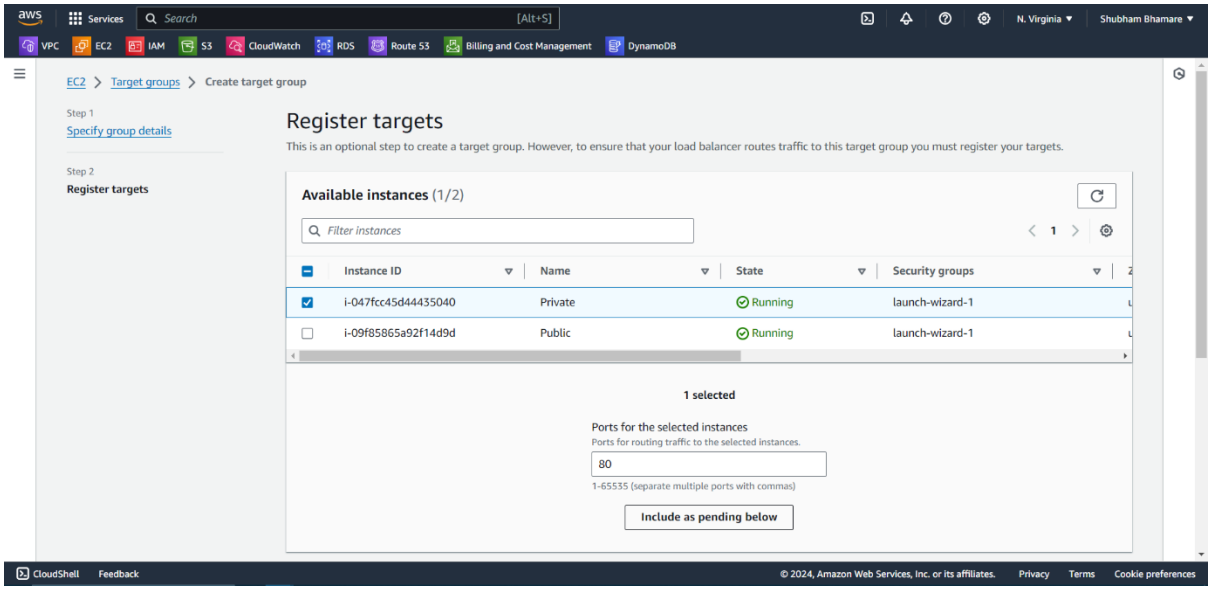


```
            # Save image file metadata in DynamoDB #
            print("Uploading to S3 success... saving metadata in dynamodb...")


            try:
                dynamodb_client = boto3.client('dynamodb', region_name='us-east-1')
                dynamodb_client.put_item(
                 TableName='metdata_table',
                    Item={
                     'empid': {
                         'N': emp_id
                     },
                     'image_url': {
                          'S': object_url
                       }
                    }
                )

            except Exception as e:
                program_msg = "Flask could not update DynamoDB table with S3 object URL"
                return str(e)

        except Exception as e:
            return str(e)

    finally:
        cursor.close()

    print("all modification done...")
    return render_template('AddEmpOutput.html', name=emp_name)

@app.route("/getemp", methods=['GET', 'POST'])
def GetEmp():
    return render_template("GetEmp.html")


@app.route("/fetchdata", methods=['GET','POST'])
def FetchData():
    emp_id = request.form['emp_id']

    output = {}
    select_sql = "SELECT emp_id, first_name, last_name, pri_skill, location from employee where emp_id=%s"
    cursor = db_conn.cursor()
:wq!
```

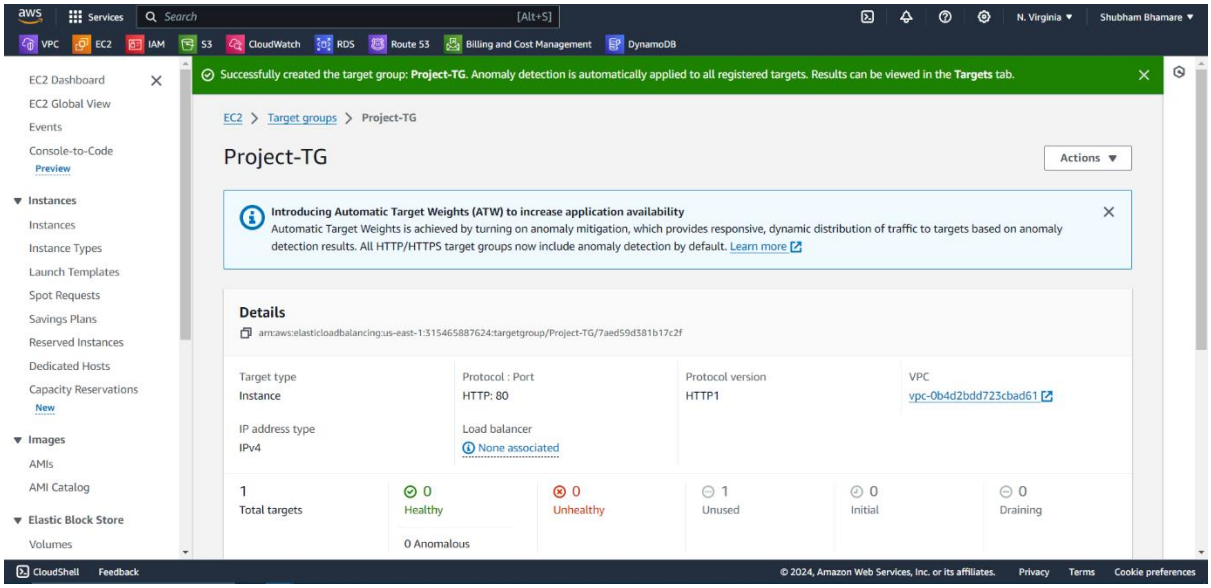
Step 66: Now go to target group and create new target group for your instance.



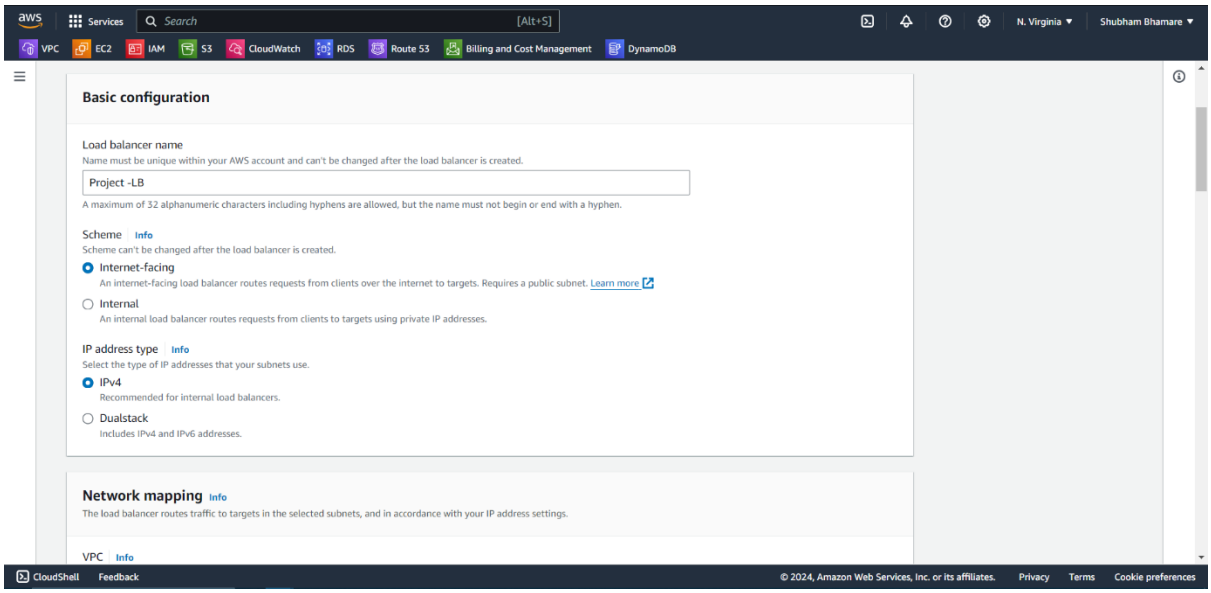
Step 67: Select your private instance only.



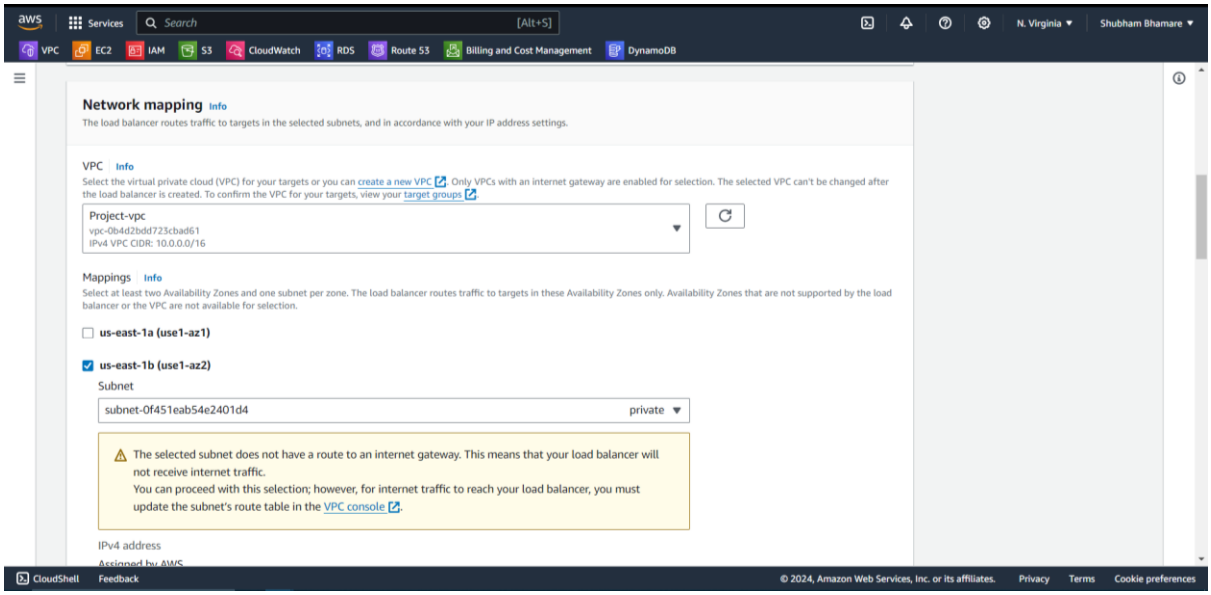
Step 68: Target group is created successfully.



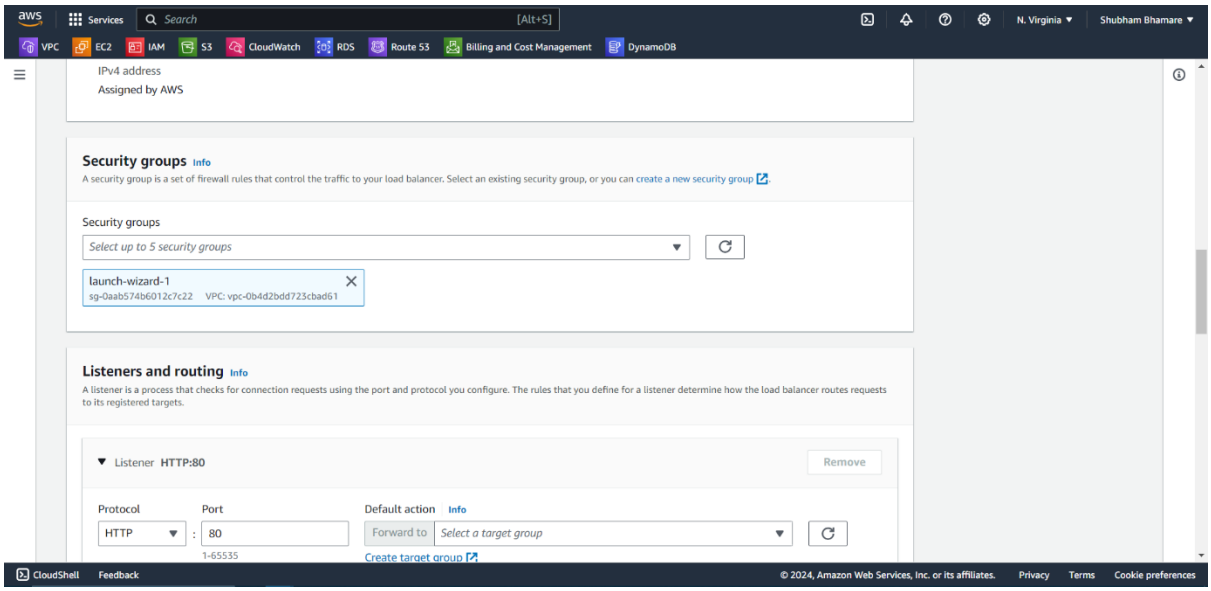
Step 69: Go to load balance and create one application load balancer.



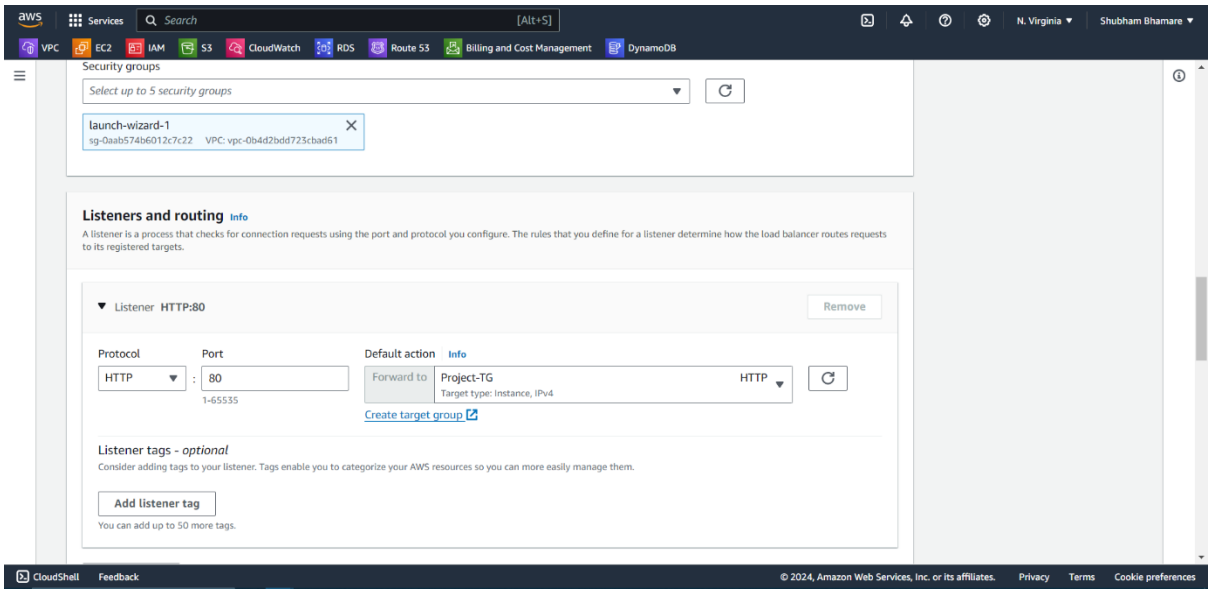
Step 70: In networking select your VPC and both subnets.



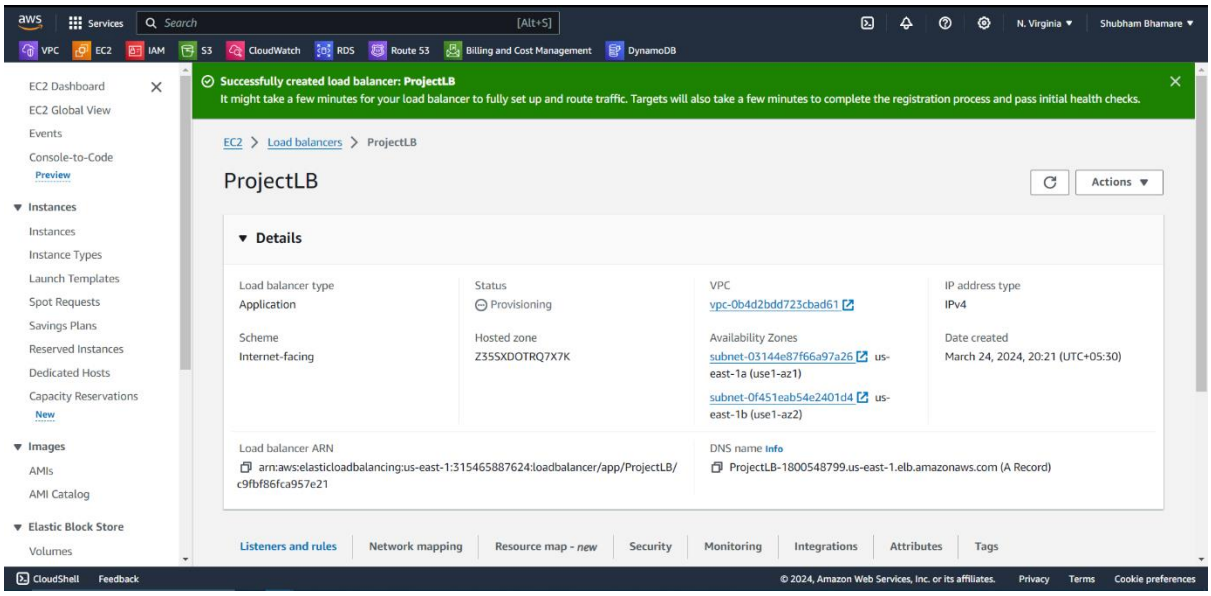
Step 71: Select the security group (All Traffic)which is also attached to your instances.



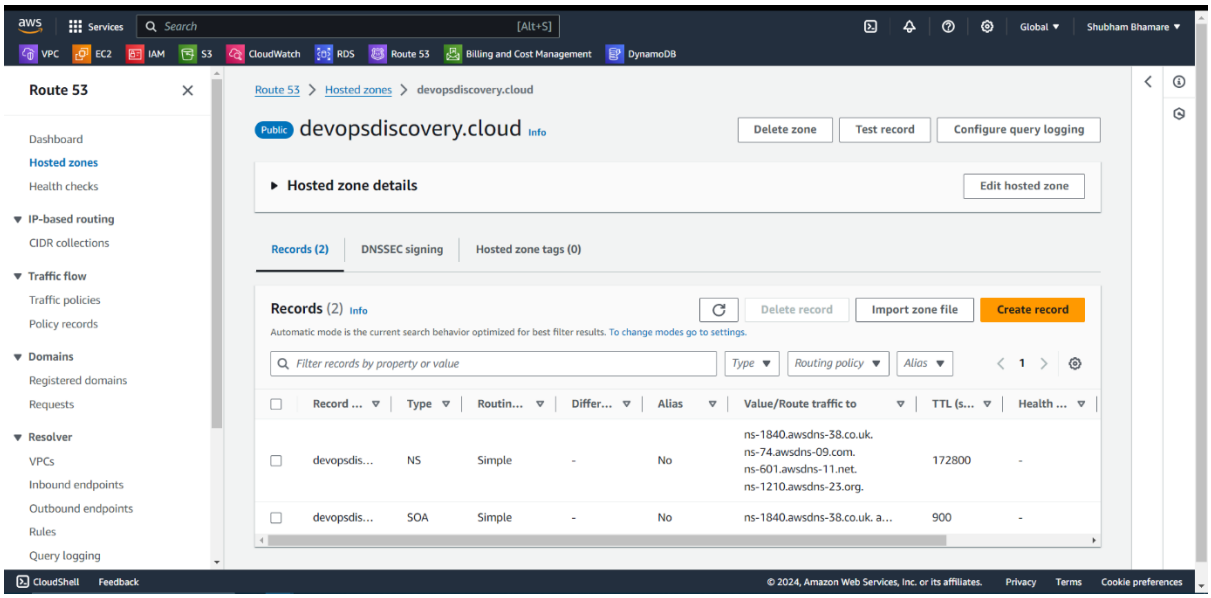
Step 72: Select your target group.



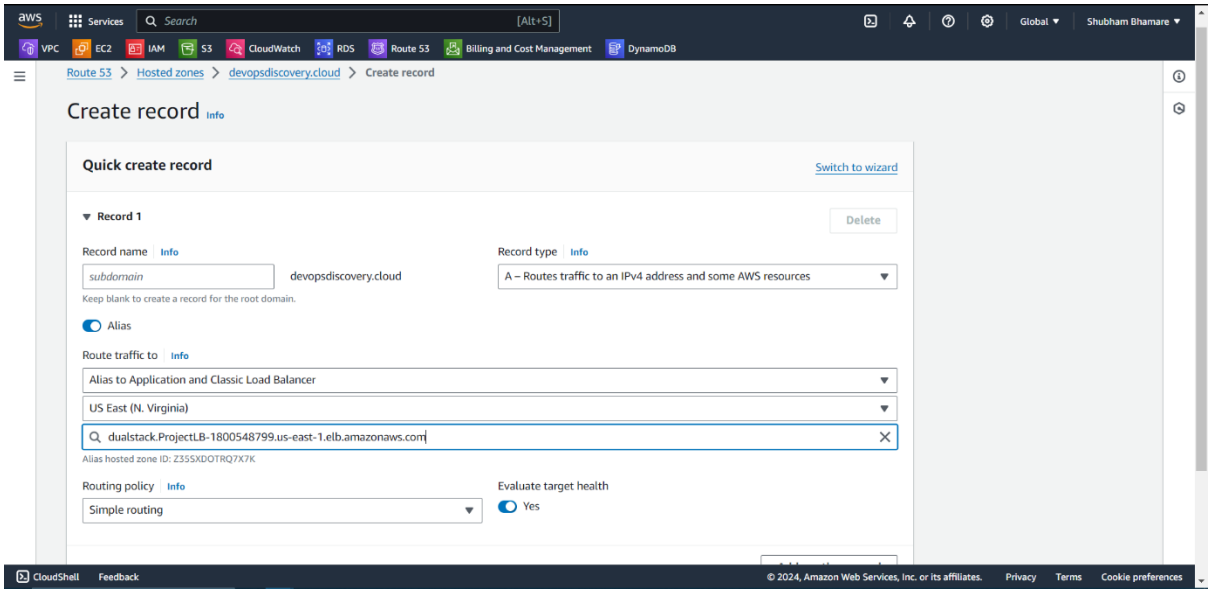
Step 73: Load balancer is created successfully.



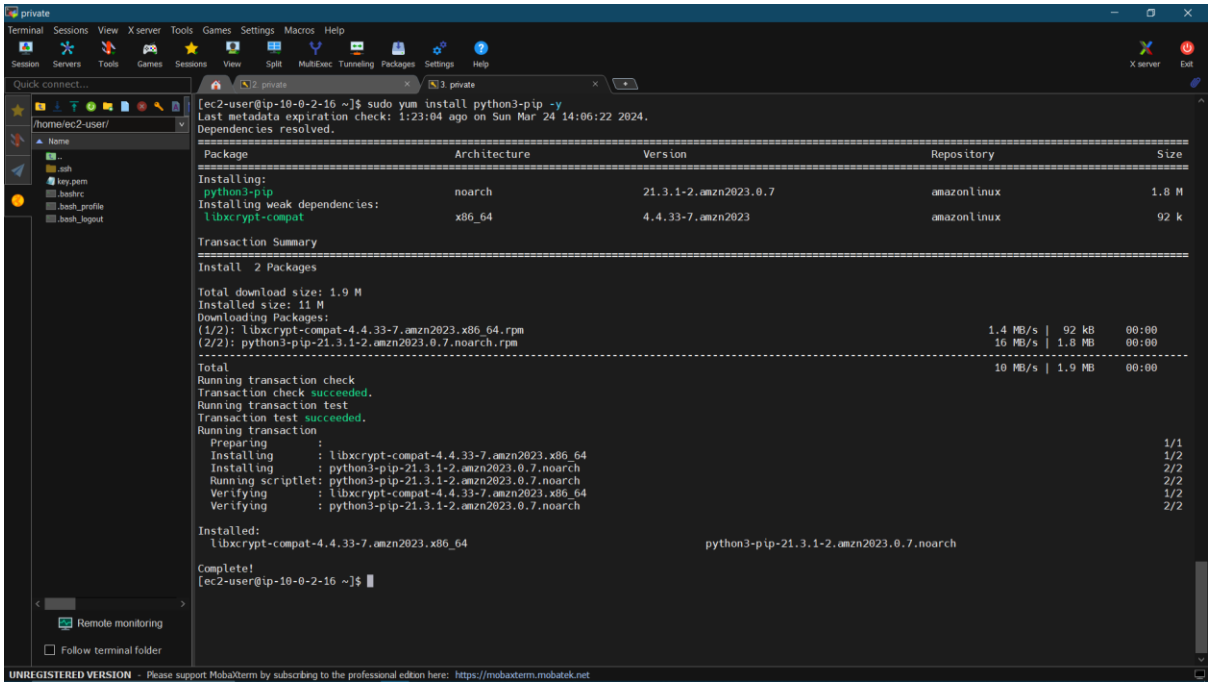
Step 74: Go to Route 53 and create one hosted zone for your Domain. Change name servers for your domain into your domain provider.



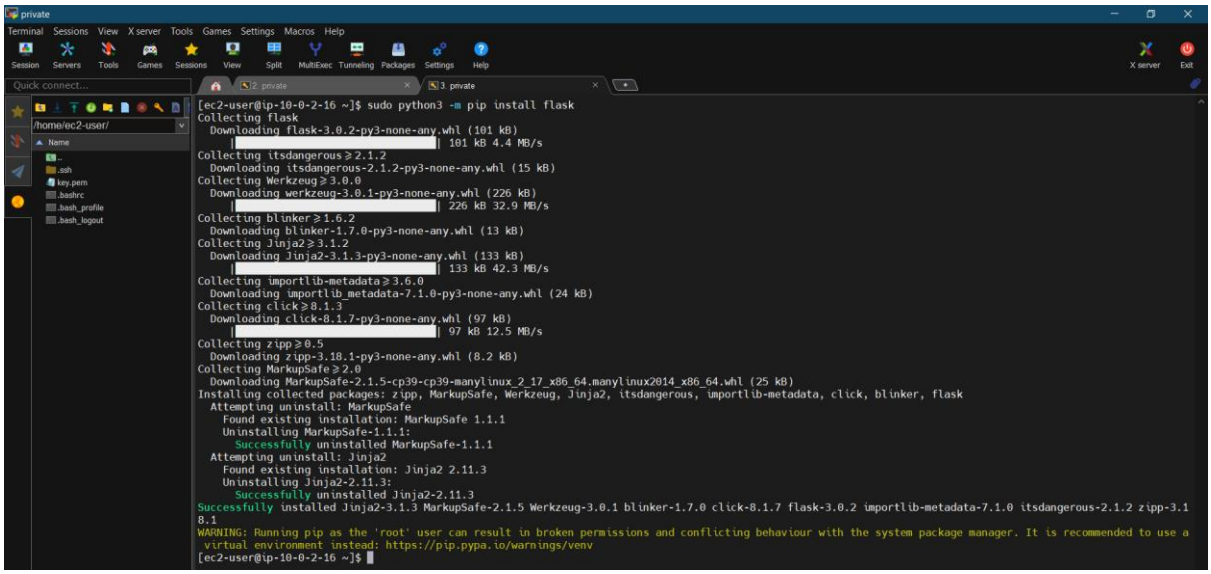
Step 75: Create one record with any subdomain , record type – A, enable alias, route traffic to the application load balancer, Select region, and select your Load balancer and create record.



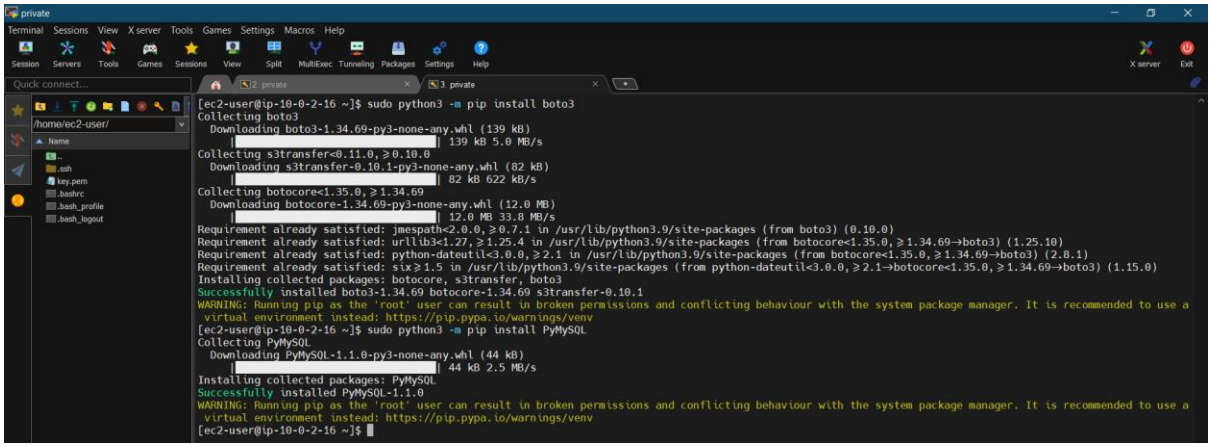
Step 76: Now run command “sudo yum install python3-pip -y” to install python on your private instance.



Step 77: Now run command “sudo python3 -m pip install flask” to download dependencies required for running python application.

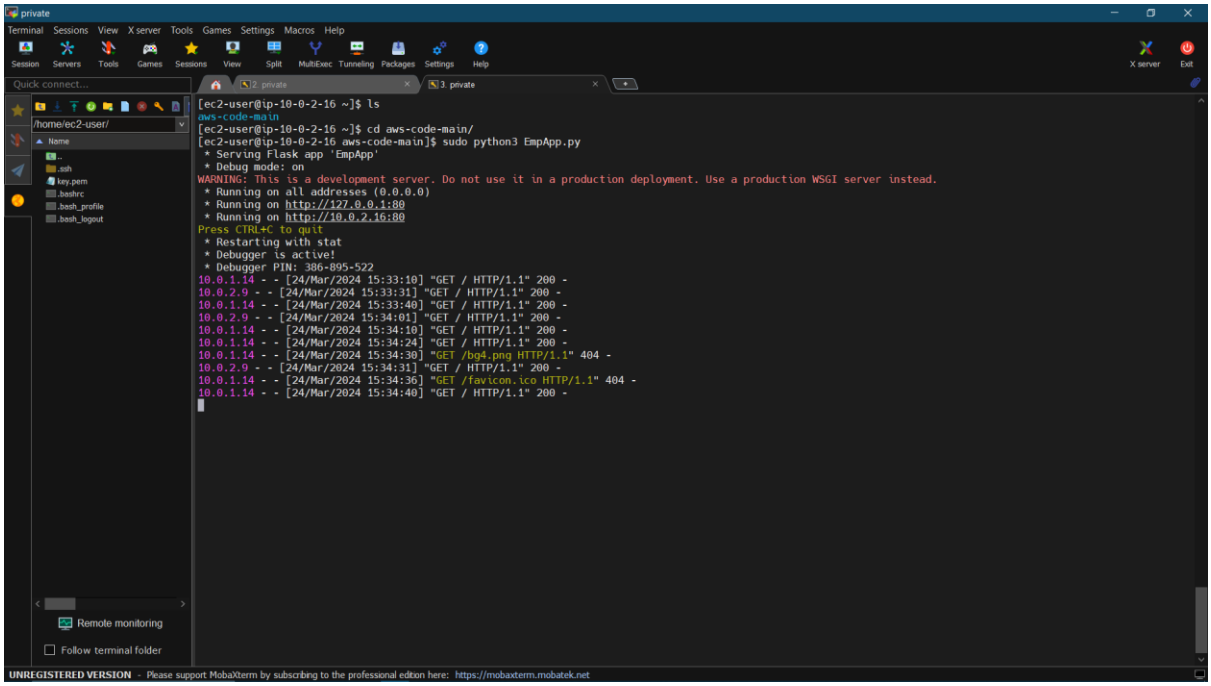


Step 78: “sudo python3 -m pip install PyMySQL” , “sudo python3 -m pip install boto3” to install required dependencies.

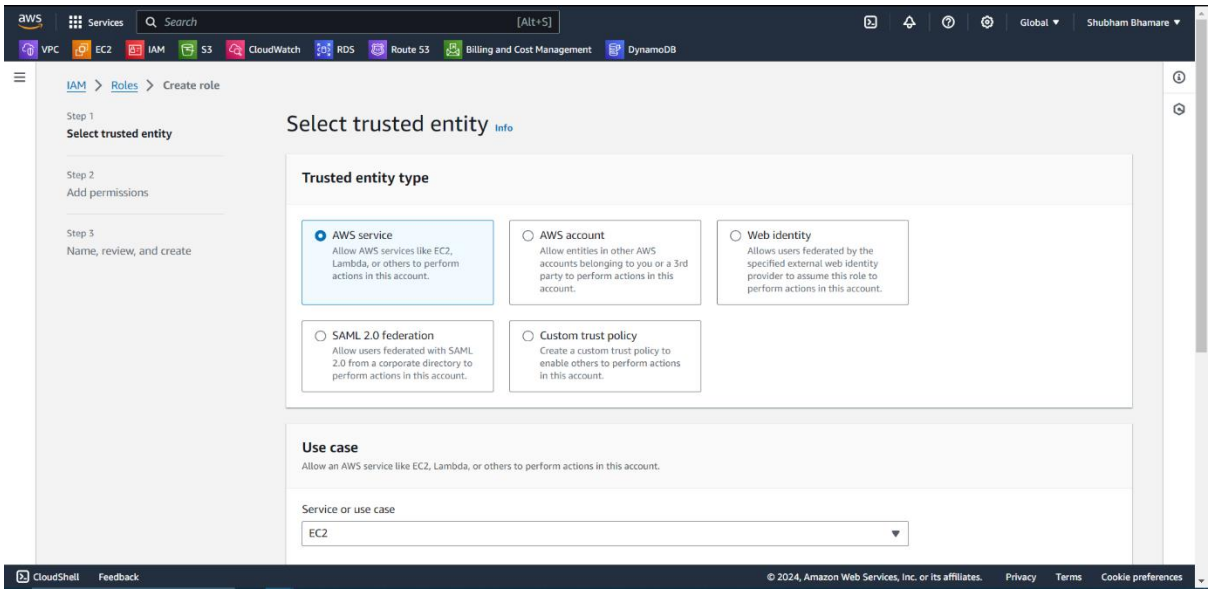


Step 79: Now to to application directory and run command

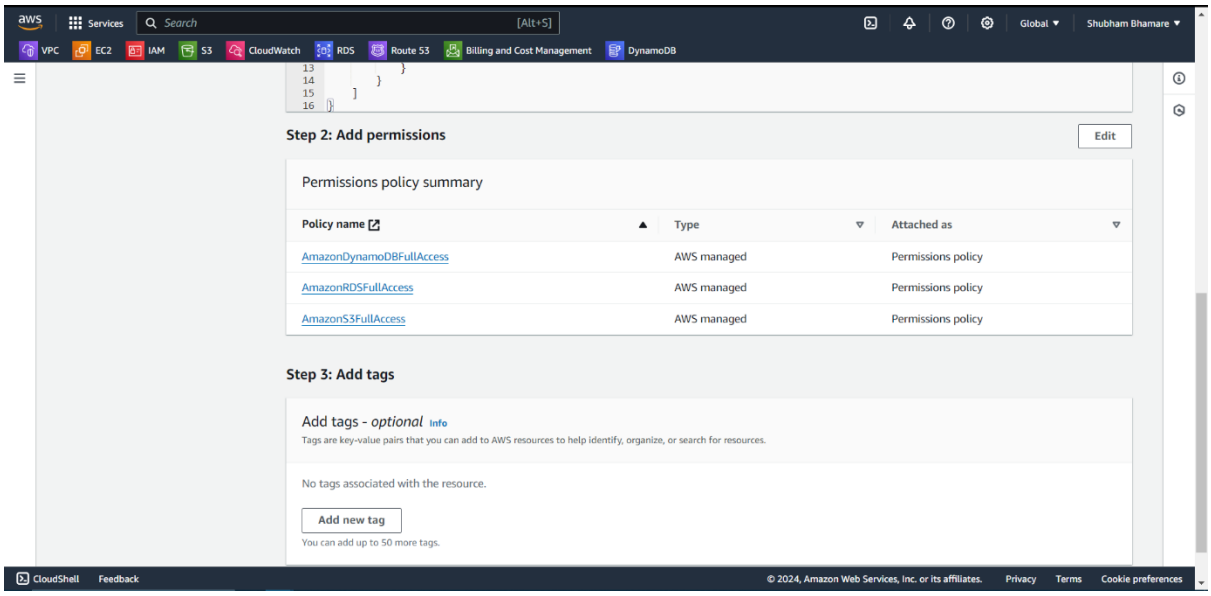
“sudo python3 EmpApp.py” on your private instance to start python application.



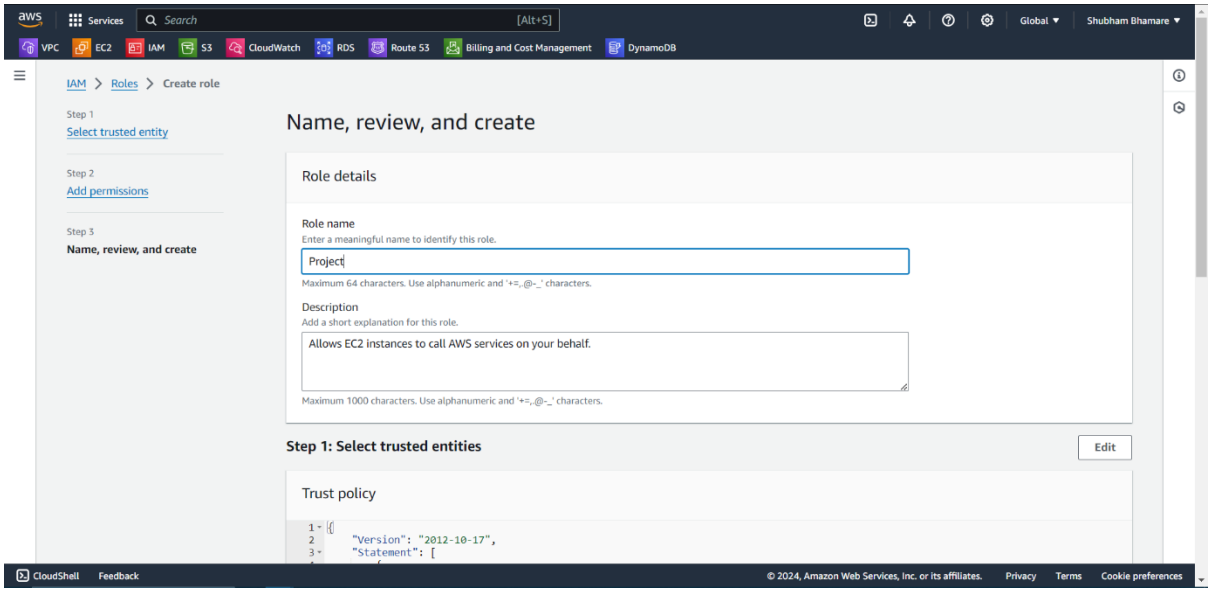
Step 80: Now go to IAM console and create one IAM role for EC2.



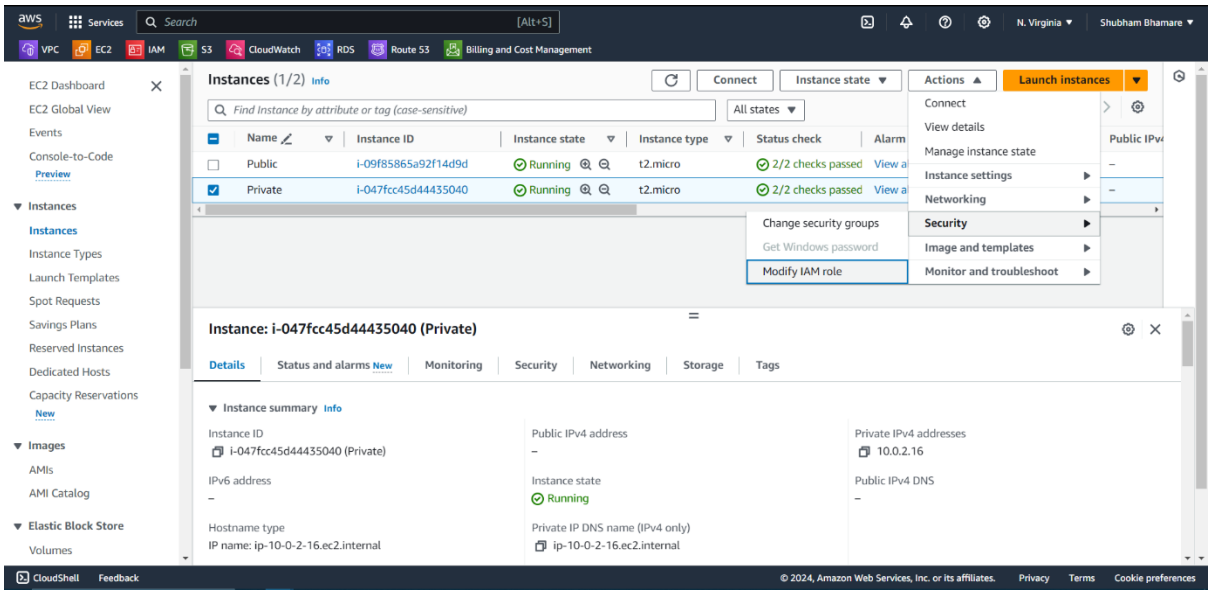
Step 81: Create role with DynamoDB,RDS,S3 full access.



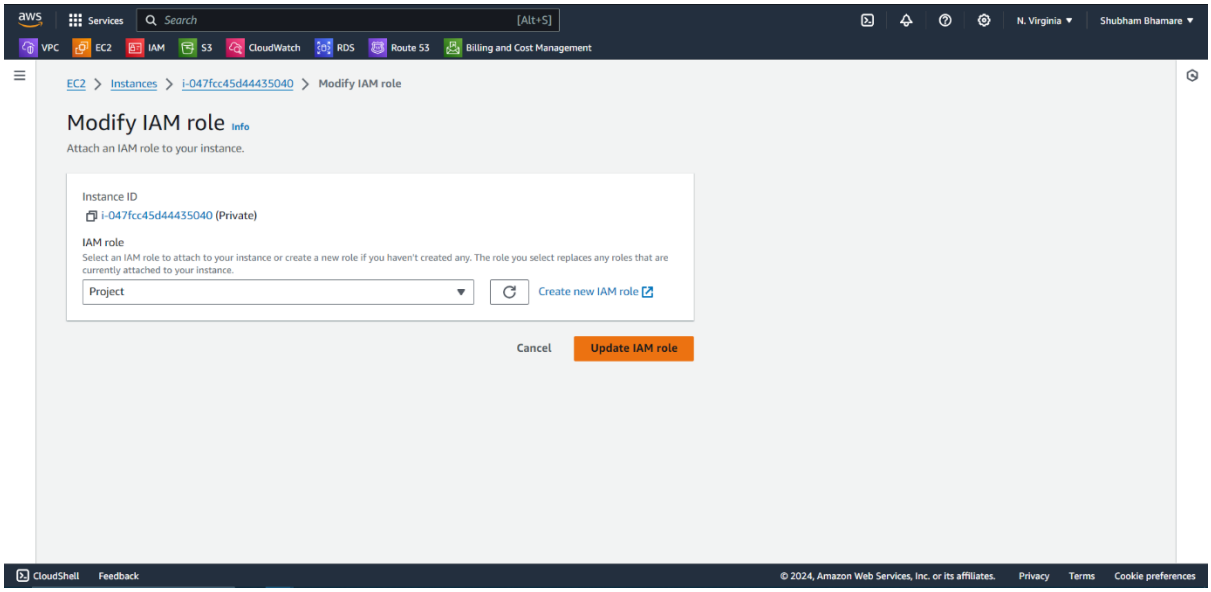
Step 82: Provide name to IAM role and create role.



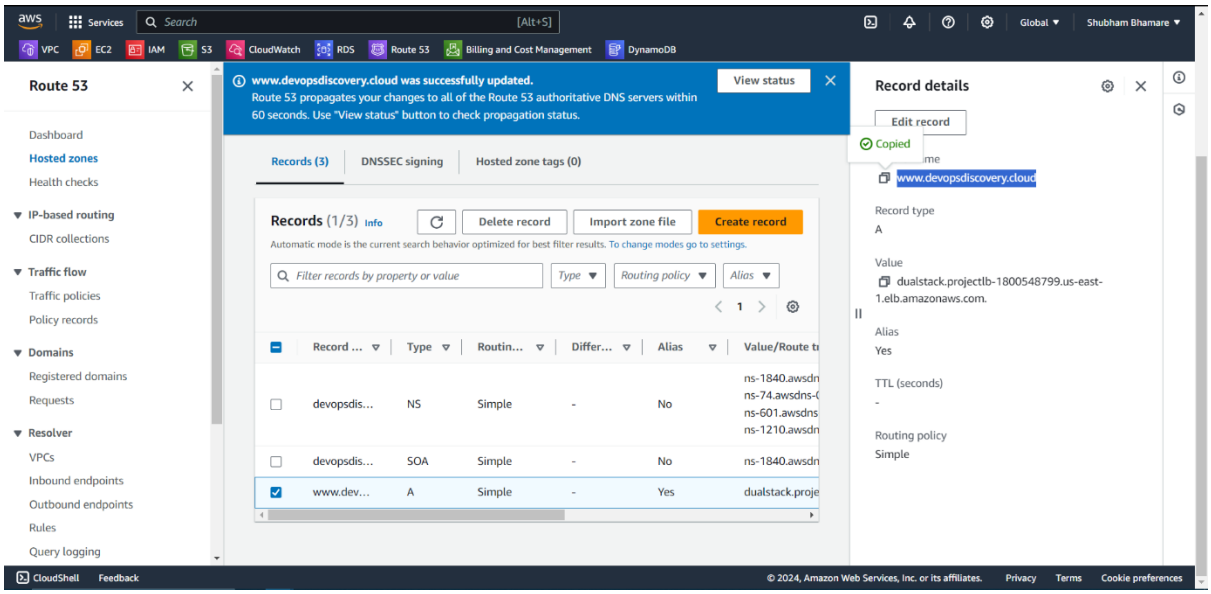
Step 83: Now attach that role role to private EC2 instance.



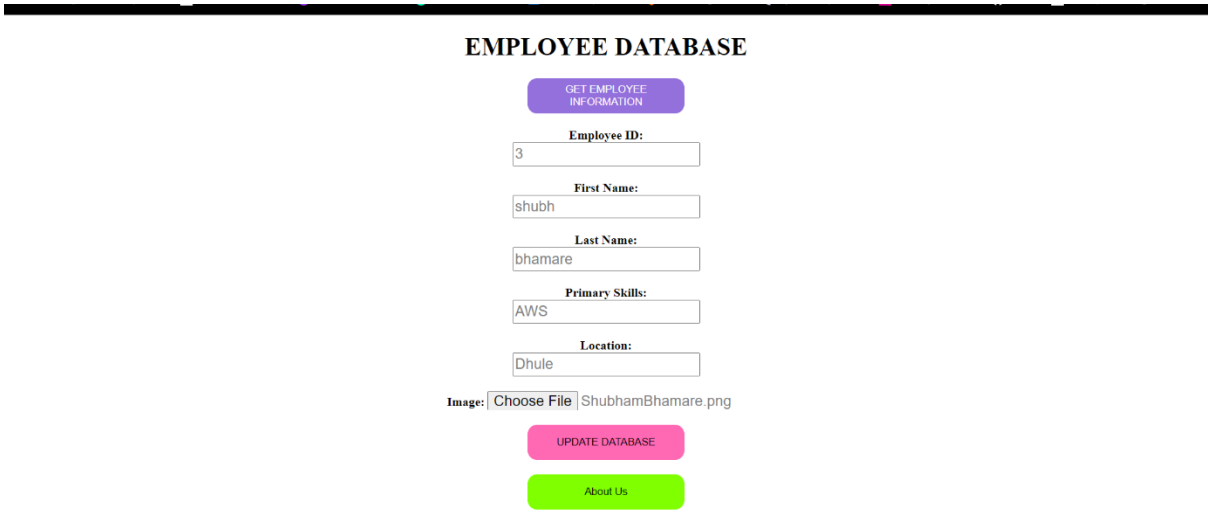
Step 84: Update IAM role.



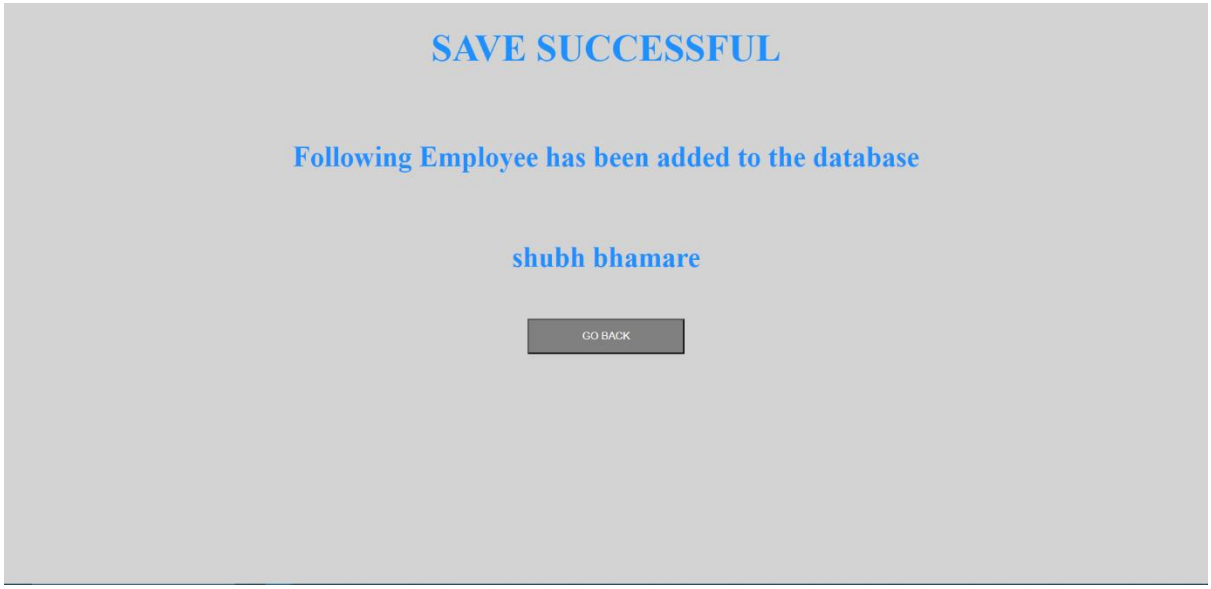
Step 85:Now copy your domain name with subdomain.



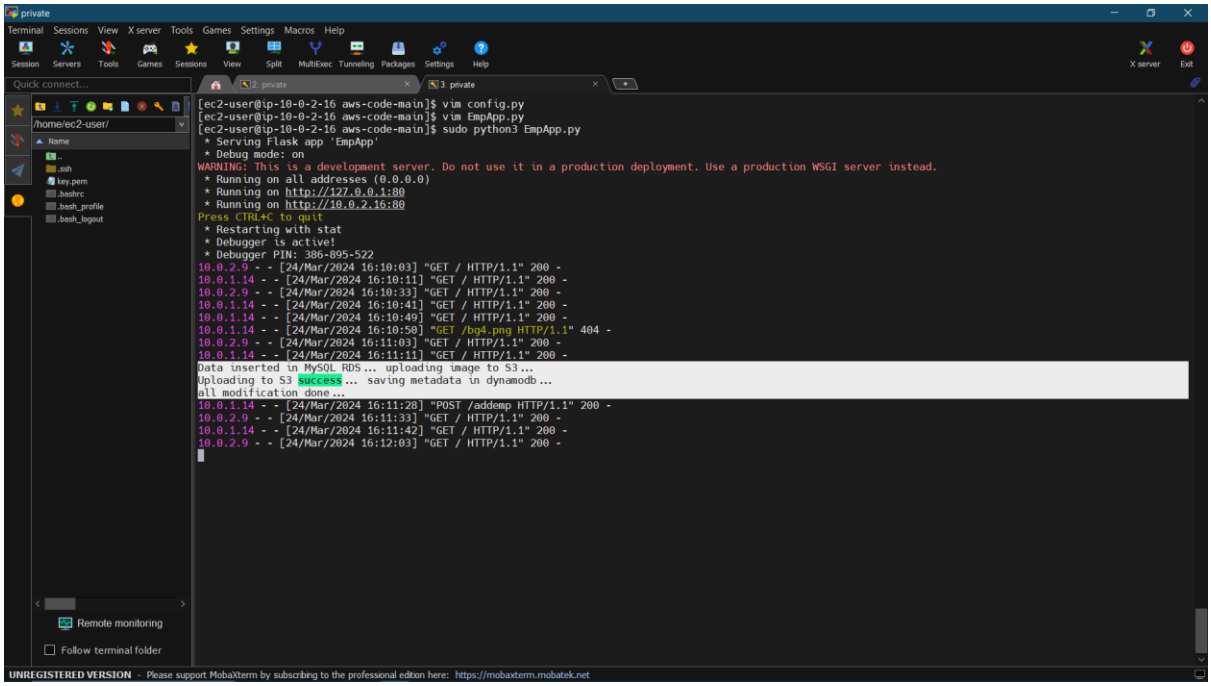
Step 86: Hit that URL on browser and fill all information, Upload image and click on “update database”.



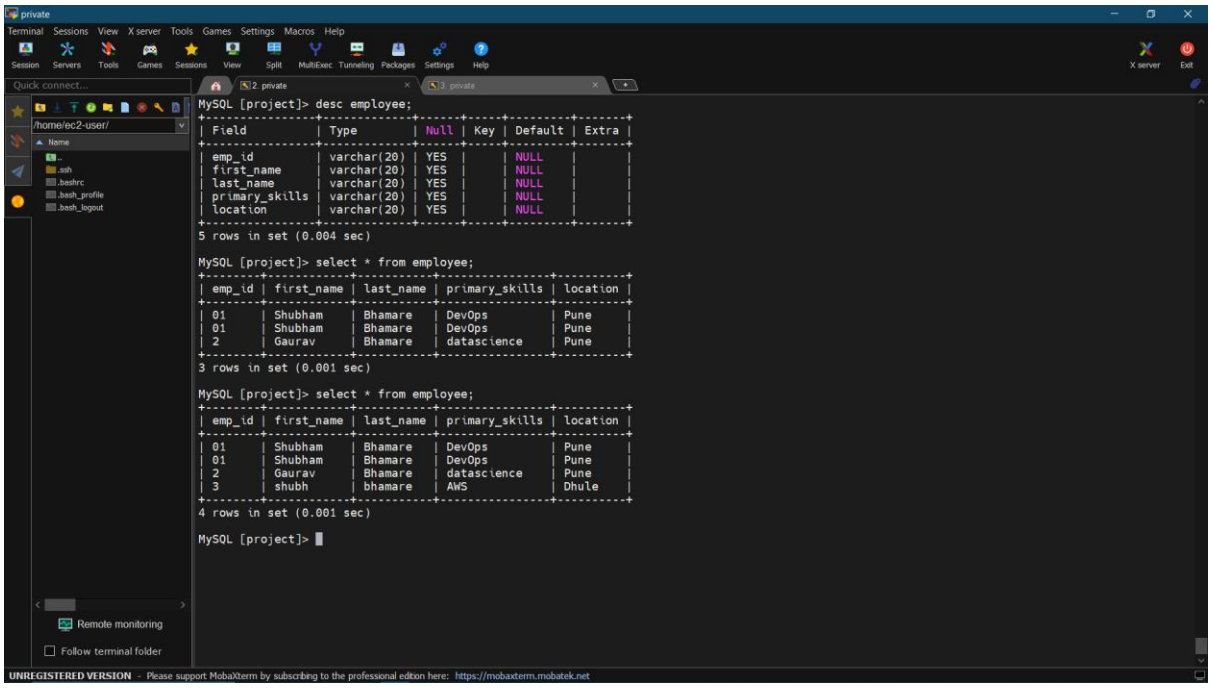
Step 87: Information is added to database successfully.



Step 88: As we can see on private instance that “Data inserted in MySQL RDS , Uploading image to s3, upload successful, saving metadata in DynamoDB, All modification done”.



Step 89: Run command “select * from <table_name>;” to check if the data stored into RDS Database.



Step 90: Check into your S3 bucket whet images is uploaded into your bucket or not.