–––

# AWS Advanced

**RDS | EC2 database program**

(Create a MySQL instance using RDS and access it using a custom program from an EC2 instance using an appropriate role)

Learning Outcomes

* Get an EC2 instance using the 7 step workflow & SSH in it
* Create a MySQL database using RDS
* Write the custom application and deploy on EC2
* Able to apply roles to EC2 instances

Final Goal

AWS (N Virginia)

IAM

Cache

Laptop

RDS

(MySQL)

Browser

SSH

EC2

DB

App

# What is needed?

1. AWS Account Credentials
2. EC2 Instances (Linux)
3. Shell script environment (any text editor of your choice)
4. Full access to - EC2, RDS, IAM, Elasticache

How to do it? - 1

1. Ensure your region is set to "N Virginia"
2. Create a private MySQL RDS
   1. Create a "Dev/Test" MySQL instance
   2. Select "Non publicly" available DB
   3. DB instance = "flipbasket", UID/PWD = root/password
   4. AZ = 2a, create a new SG
   5. IMP When the RDS instance is created check the SG to ensure that the "inbound" rule does not have any specific IP address, instead use the VPC CIDR block of the EC2 instance. Otherwise we will not be able to connect from the EC2 instance
3. Create 1 EC2 instance using the 7 step workflow
   1. Use the usual Ubuntu 20.04 LTS AMI in AZ1
   2. Download a new PEM file and SSH to the instance
4. IAM
   1. Attach the pre-created role “labinstanceprofile” to the instance OR use an IAM role using

the permission “RDSFullAccess”

# How to do it? - 2

1. SSH to the EC2 instance and fire the following commands
   1. sudo apt update
   2. sudo apt install mysql-client
2. Connect to the RDS instance & create the data
   1. wget https://d6opu47qoi4ee.cloudfront.net/employees.sql
   2. mysql -h [RDS end point] -u admin -p

You should get the MySQL prompt after entering the password

* 1. create database employees;
  2. use employees;
  3. source employees.sql
  4. describe employees;
  5. Note the column names in a text editor from the output of the above command
  6. Close the mysql terminal

# How to do it? - 3

* Follow the steps for the Python program connecting to the RDS instance from EC2
  + sudo apt install python2-minimal
  + sudo apt install python3-pip
  + sudo pip3 install boto3
  + sudo pip3 install mysql-connector-python
  + sudo pip3 install pymysql

In case the pip install fails execute the following 3 lines -

export LC\_ALL="en\_US.UTF-8" export LC\_CTYPE="en\_US.UTF-8" sudo dpkg-reconfigure locales

* Use the rds.py as the template and update the hostname to the endpoint of your RDS instance

and run the program

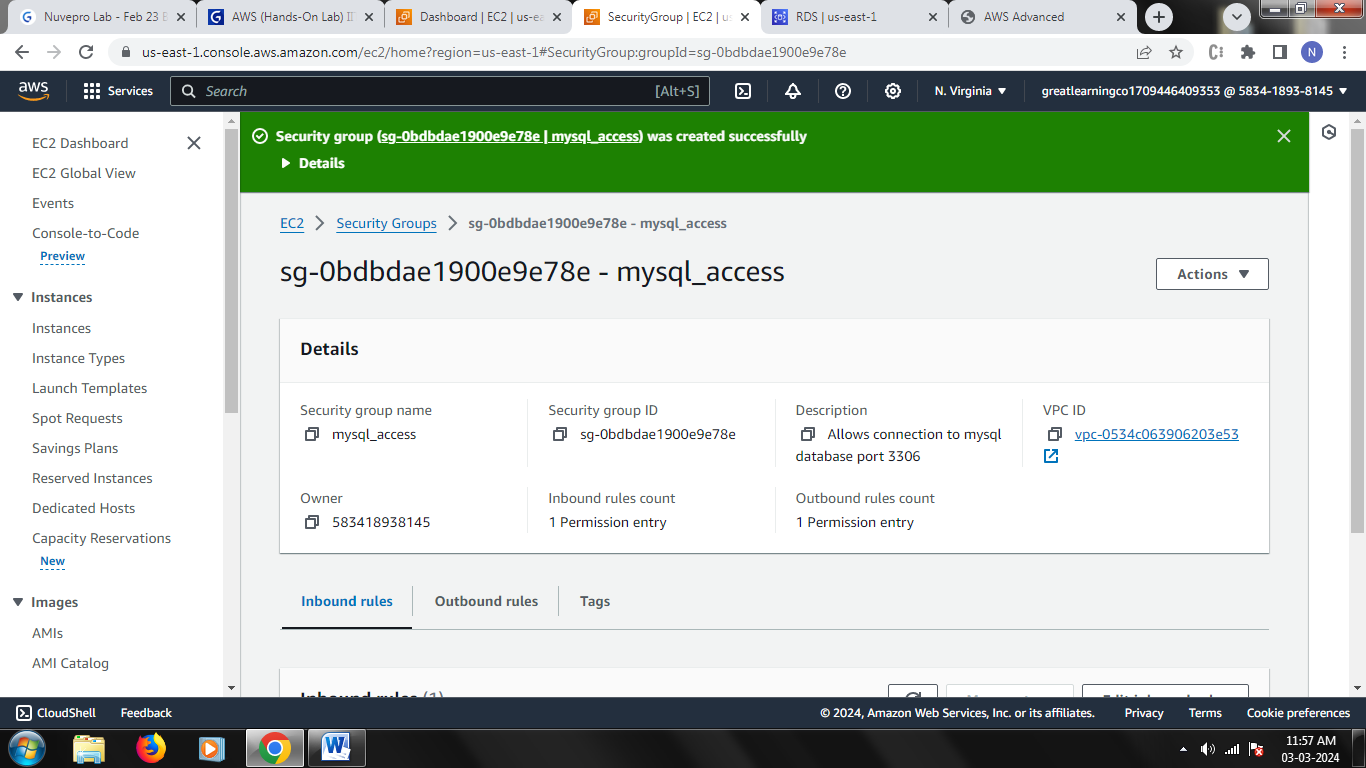
* + wget https://d6opu47qoi4ee.cloudfront.net/rds.py
  + Modify the python program in all the places marked with "TBD"
  + python3 rds.py
  + In case you get an error for the mysql connector just re-execute the sudo pip install command

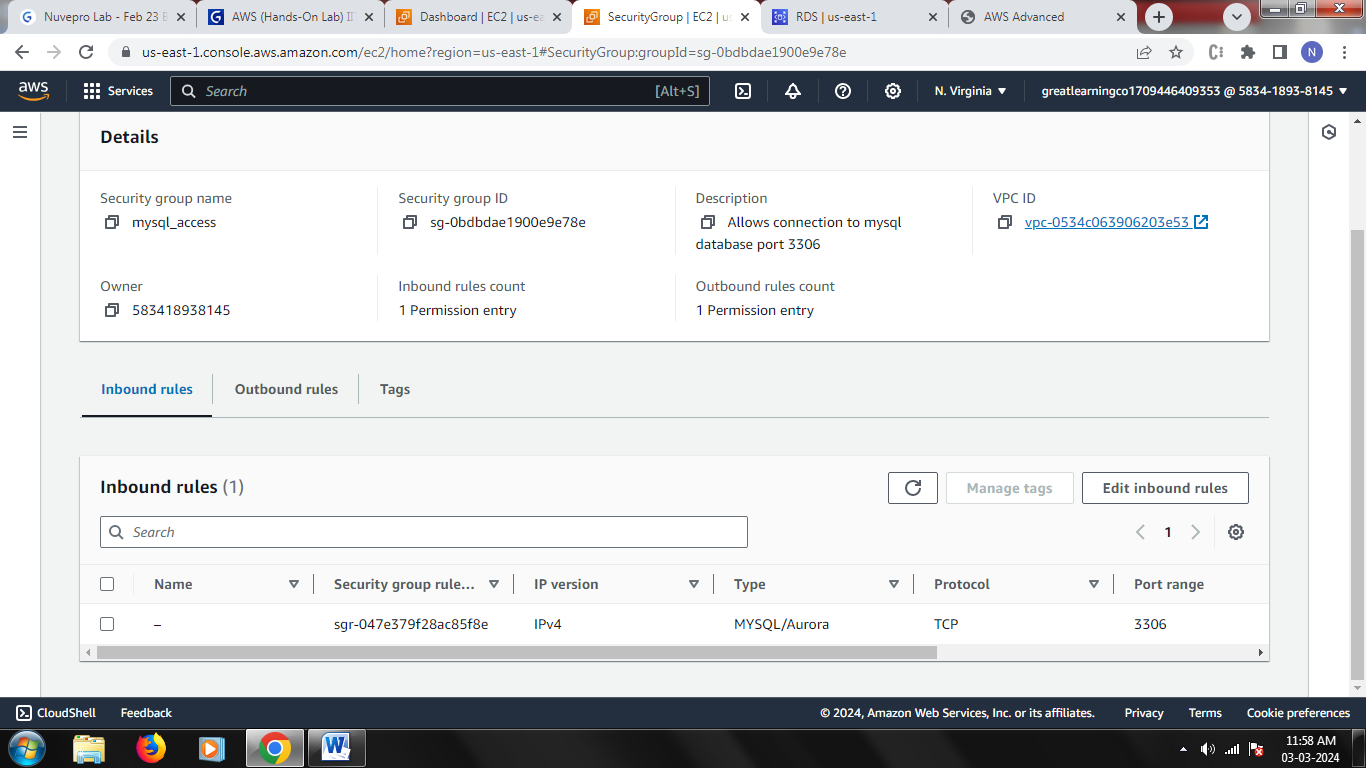
## Q: state your observations when you remove the role and rerun the program (Hint - RDS SG)

How to do it? – 4 (Advanced \*)

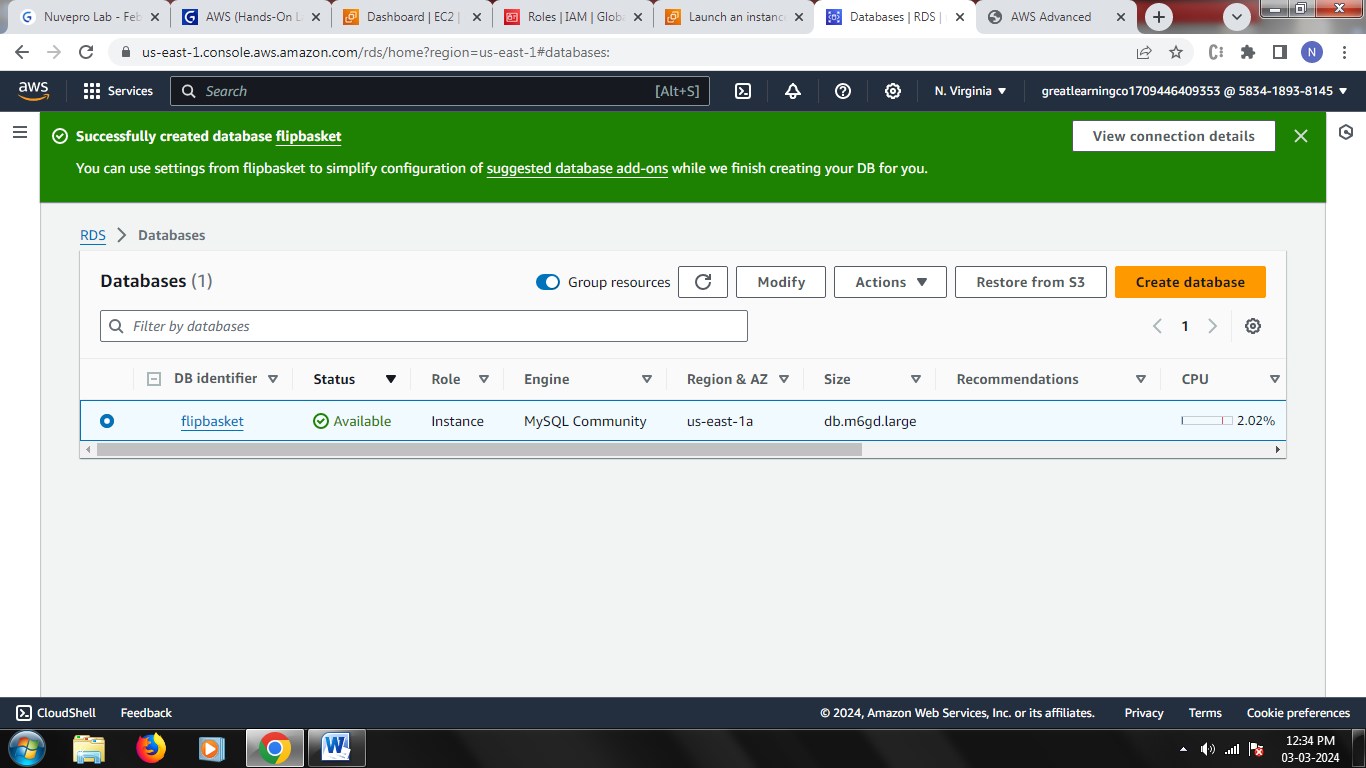
* Once you have updated the program to connect with the database, do the following
  + Create a Redis cluster (self managed / Elasticache)
  + Modify the program to include another method "getOrder(...)"
  + Add a class that represents the order
  + This method needs to check if the order exists in the cache, if found then get from cache and return else go to the DB, put in cache and return
    - Use the python modules - redis & pickle

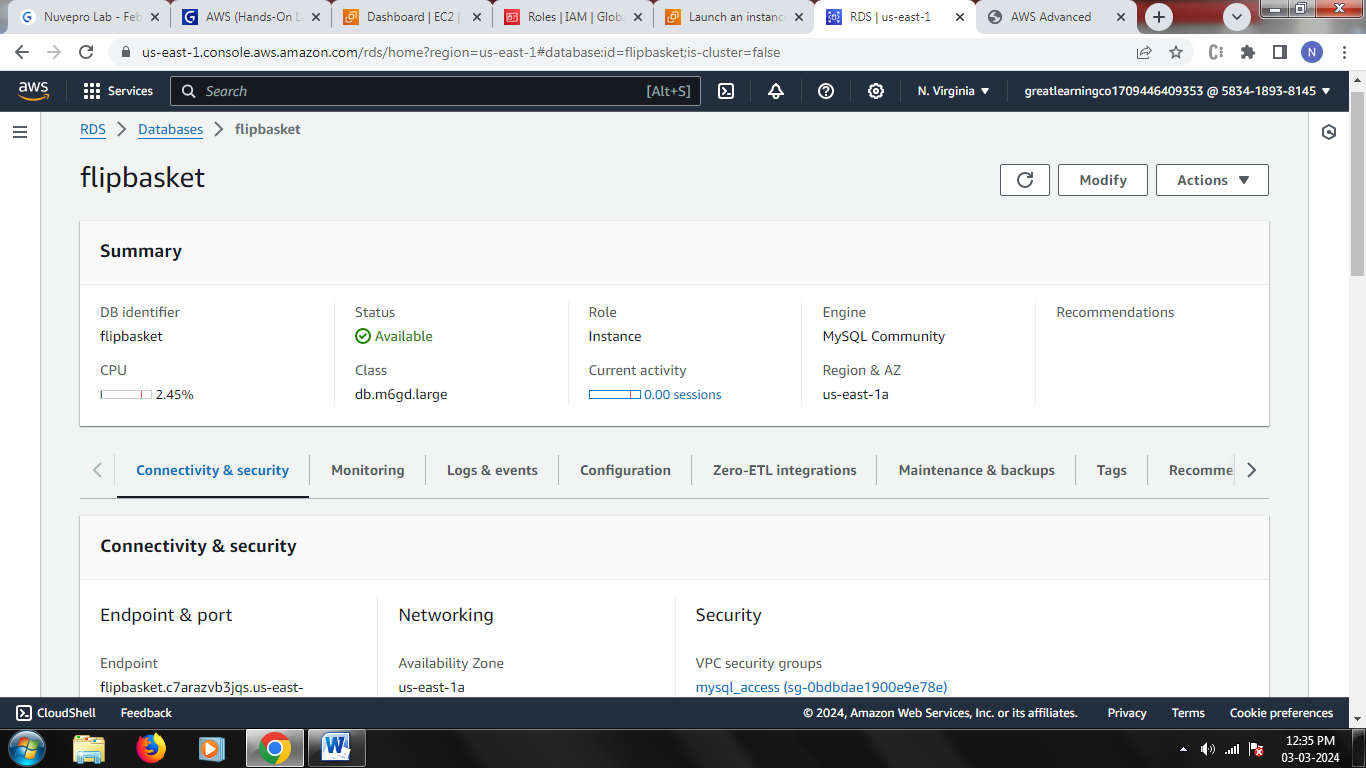
Creating an Security group for RDS



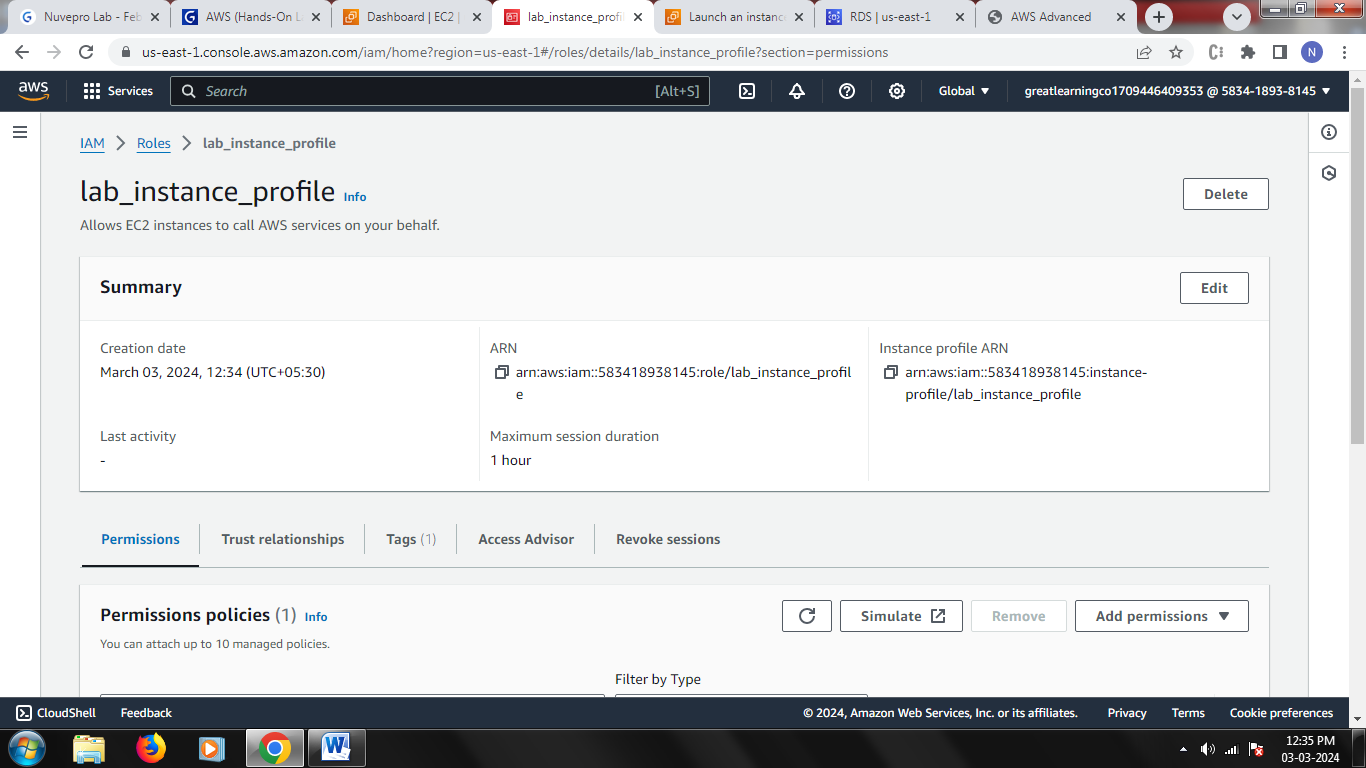


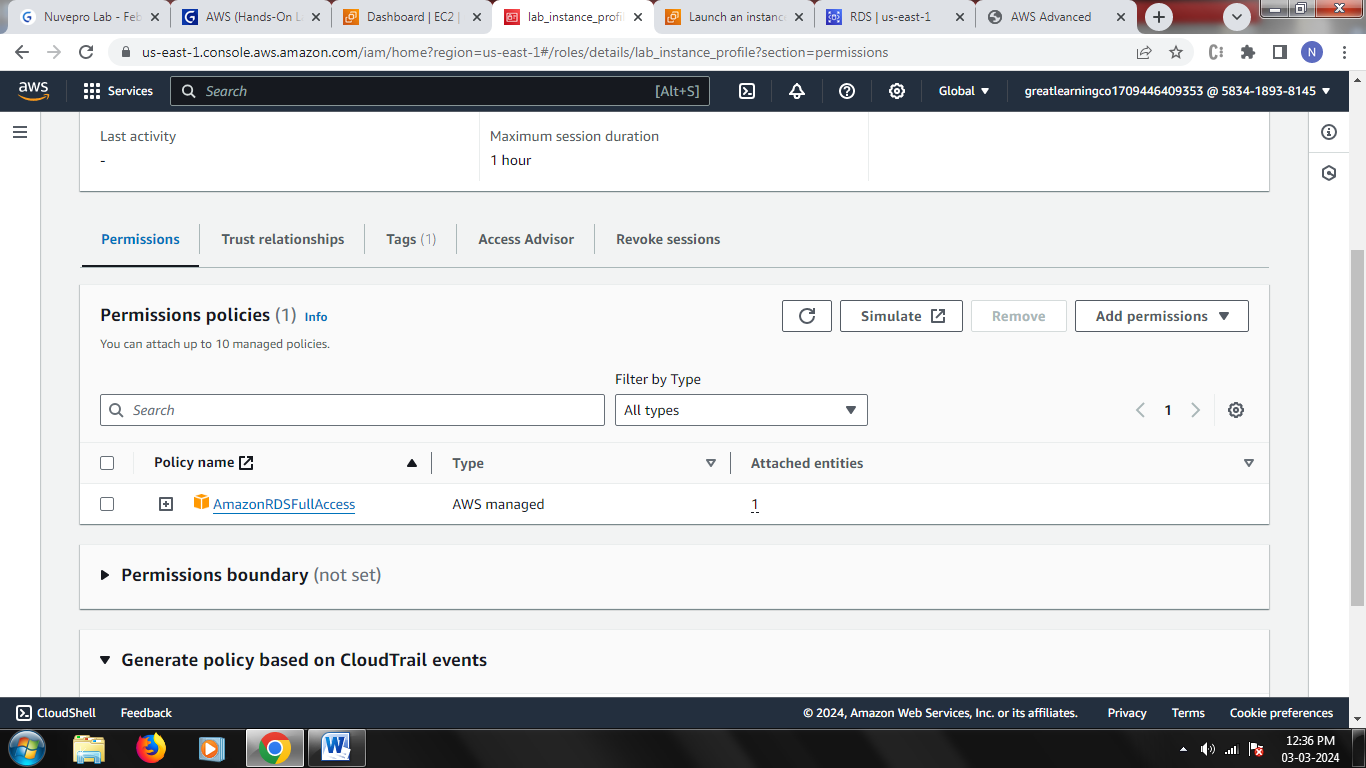
Creating an RDS database



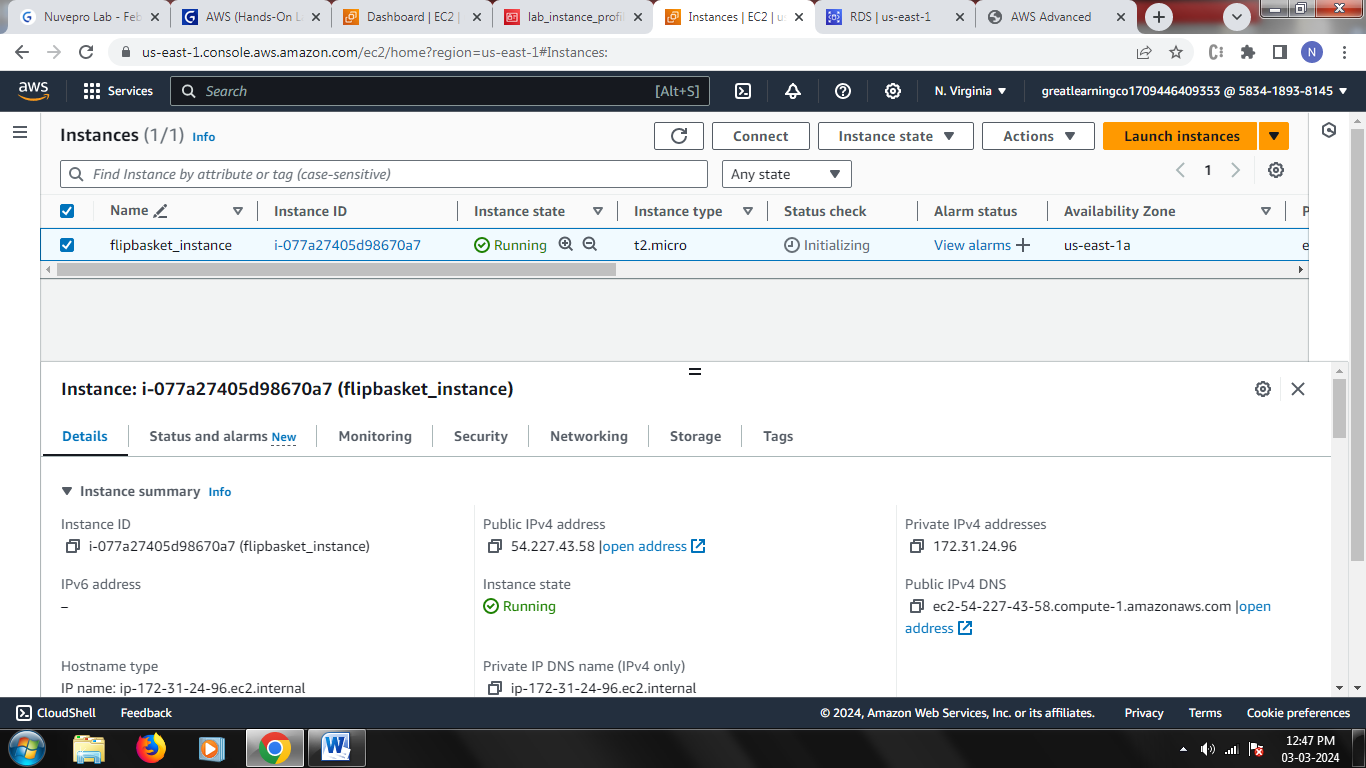


Creating a role with RDS Full Access

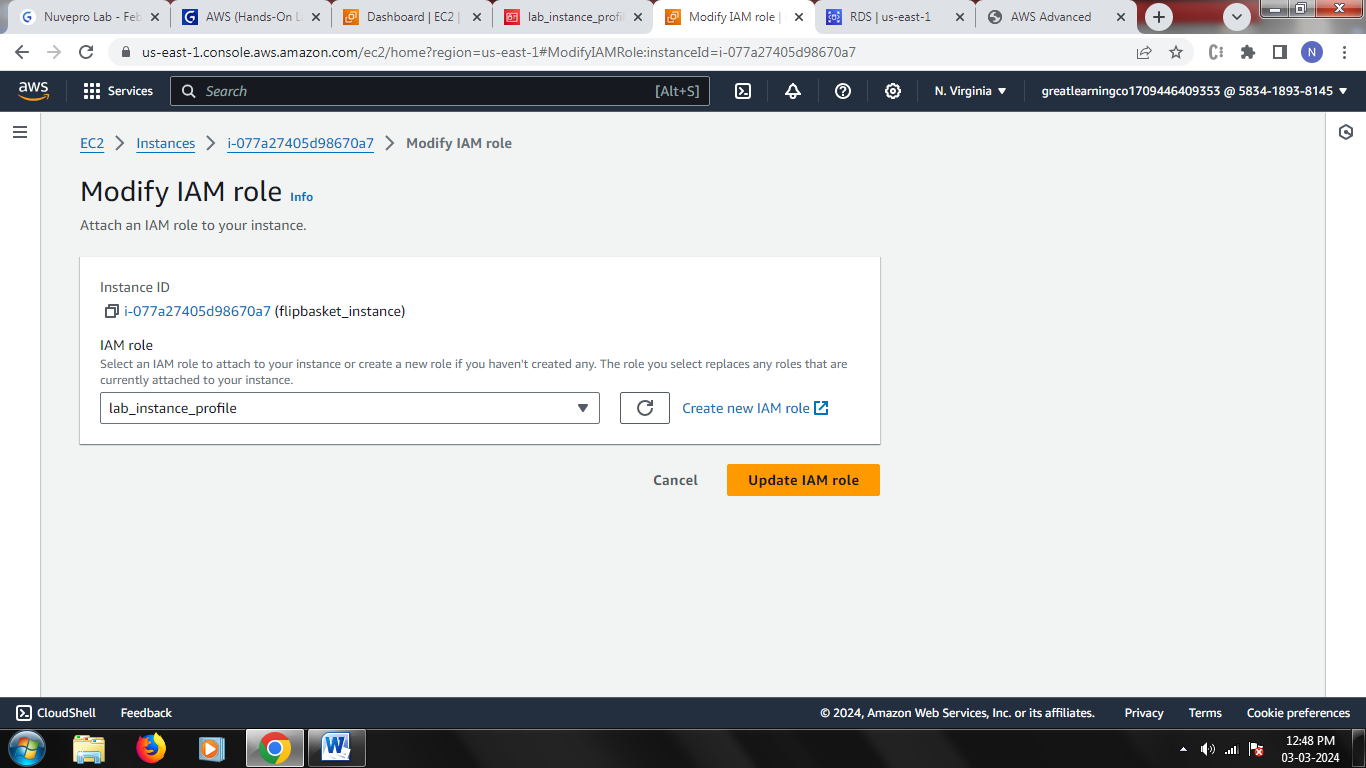


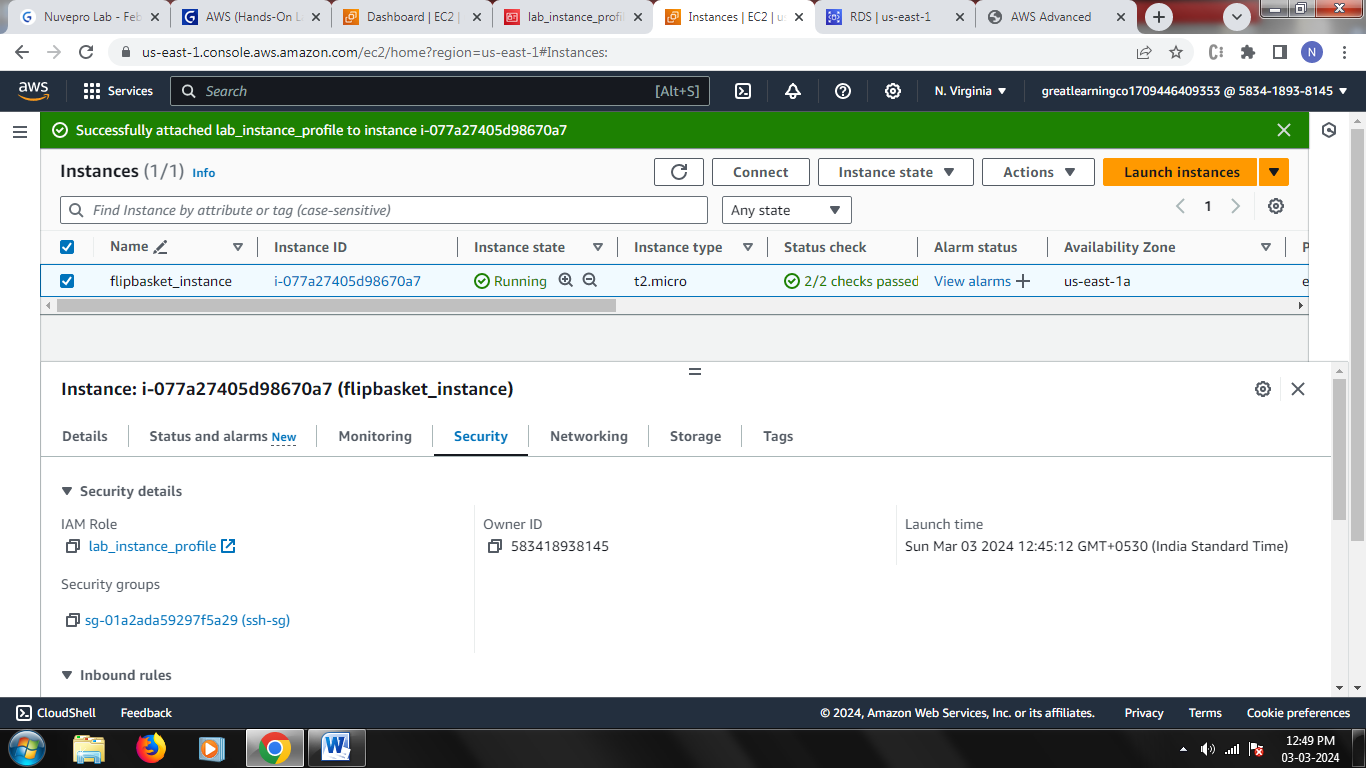


Creating an EC2 instance

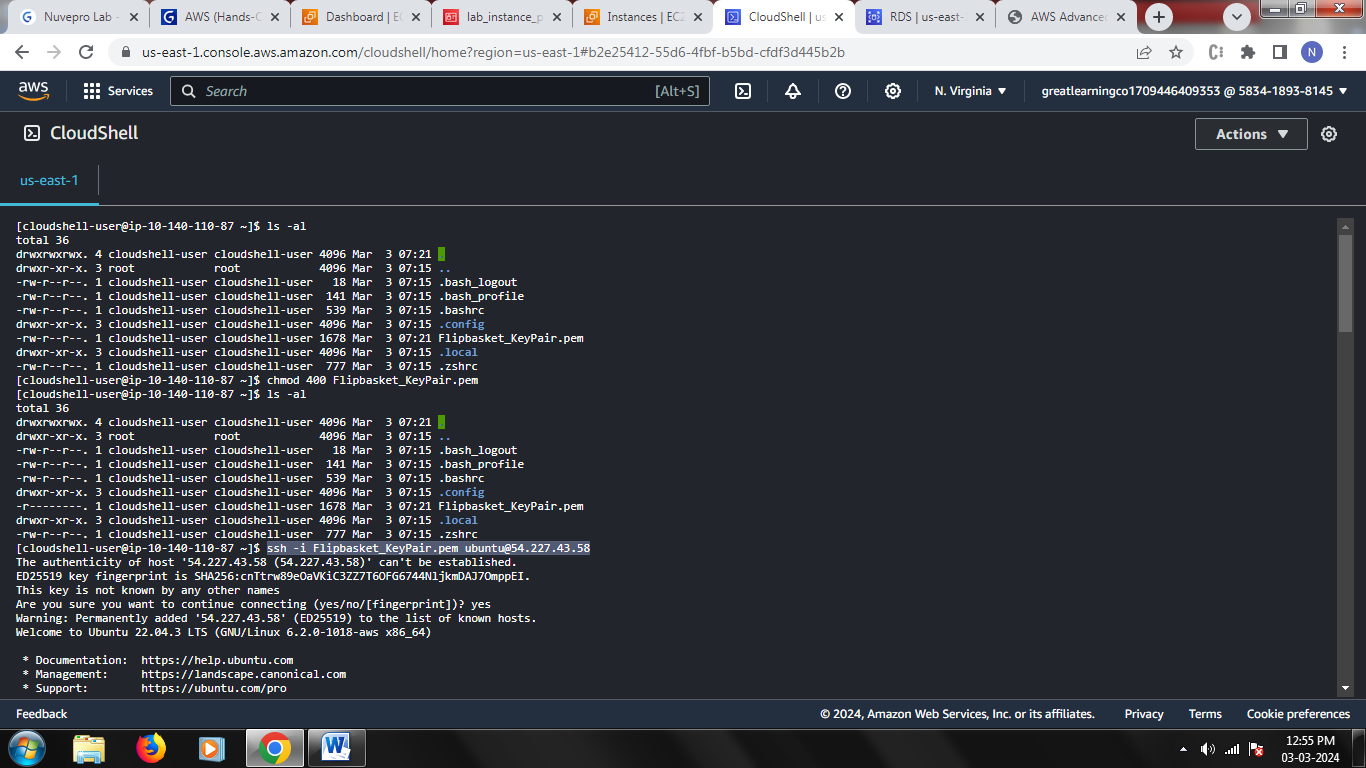


Attaching the above created role with EC2 instance

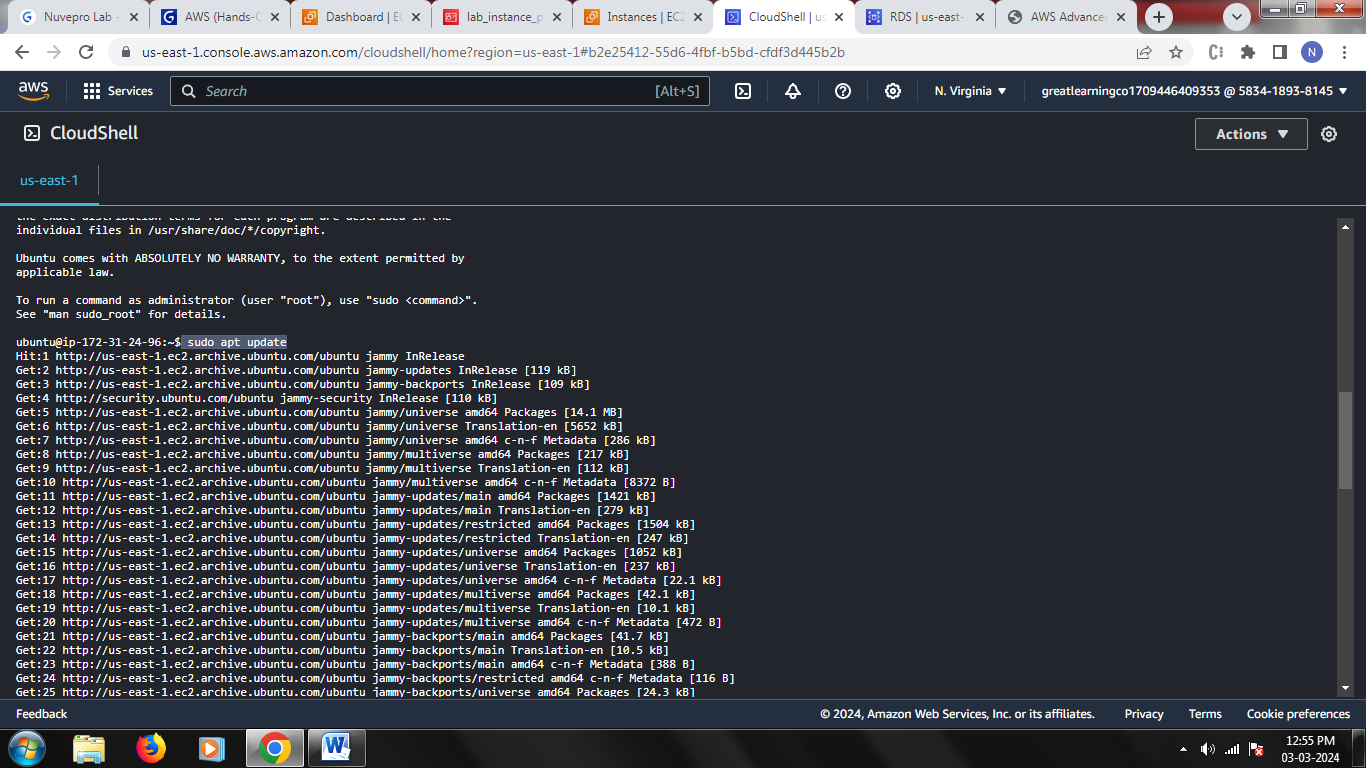




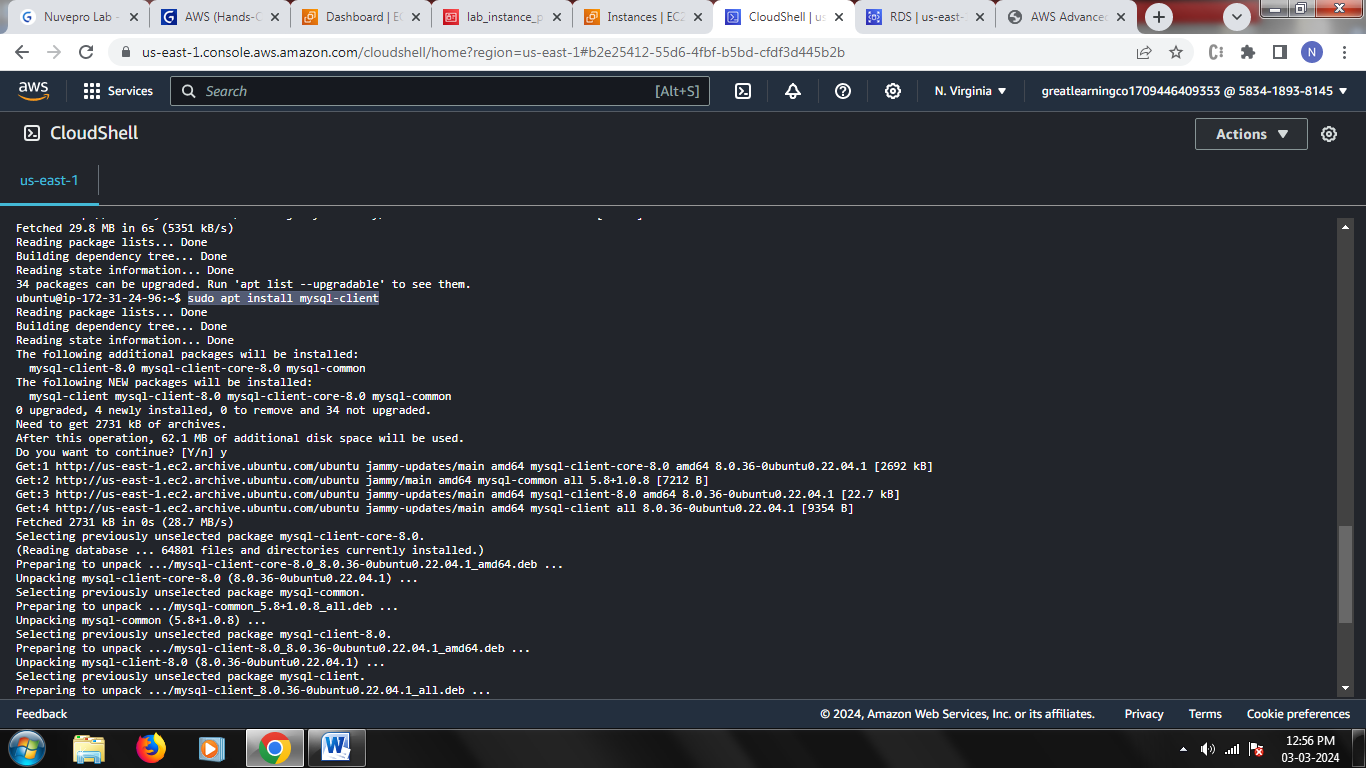
Upload newly generated pem file and SSH into your EC2 instance



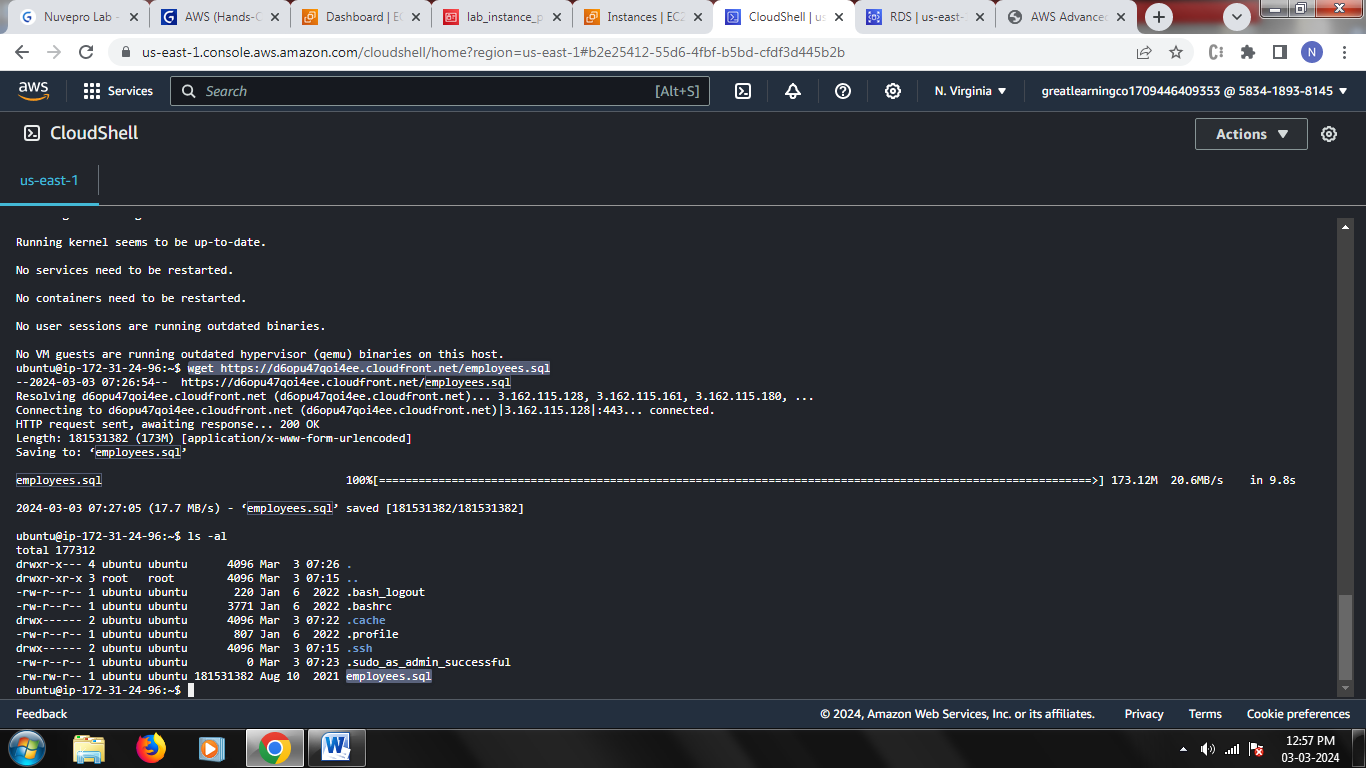
Update all files



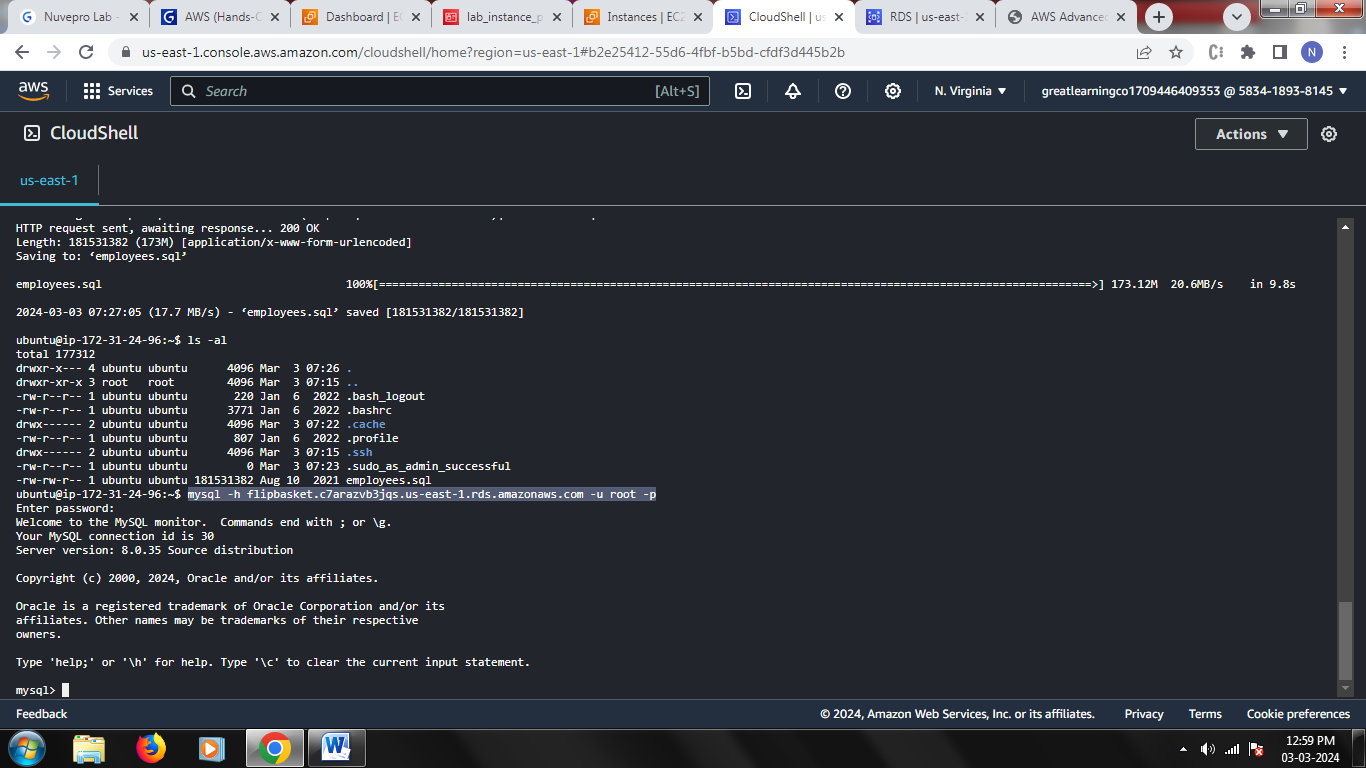
Install mysql-client



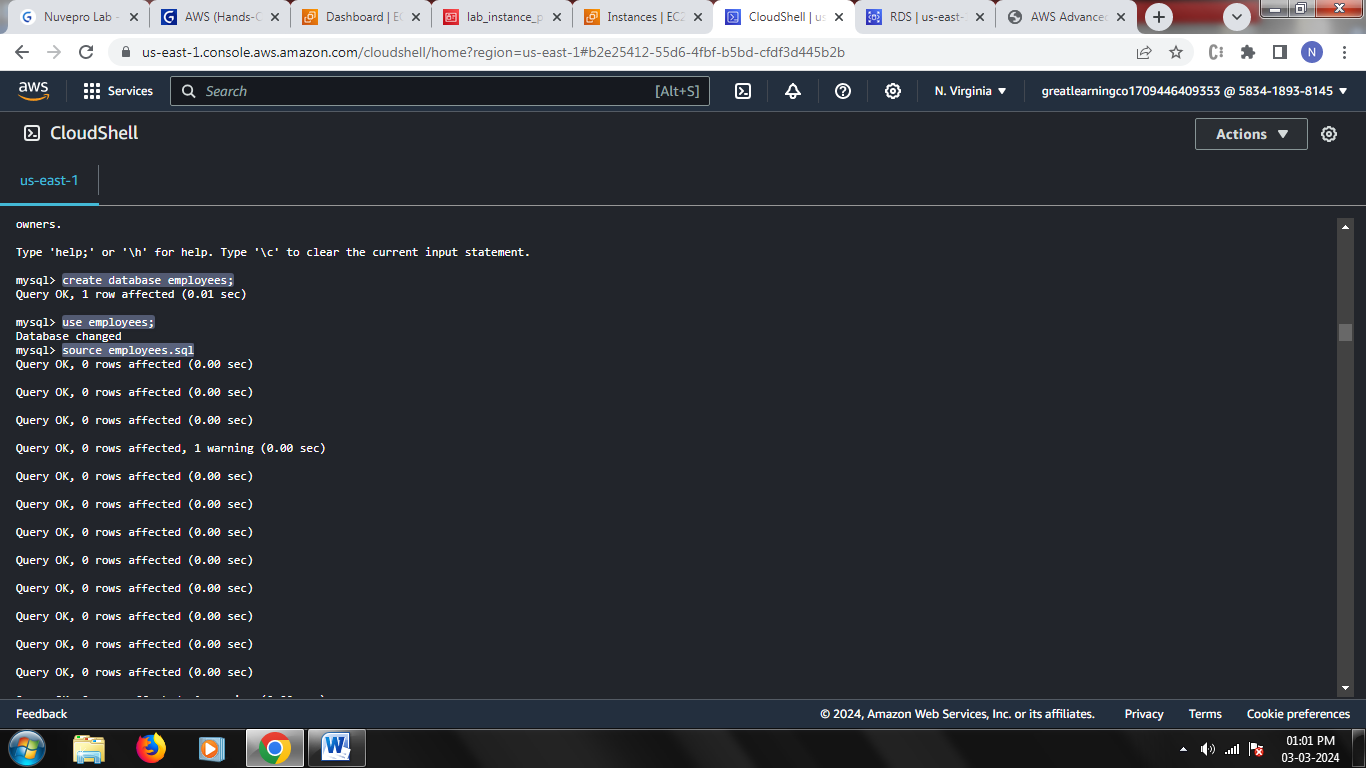
Get employees.sql file

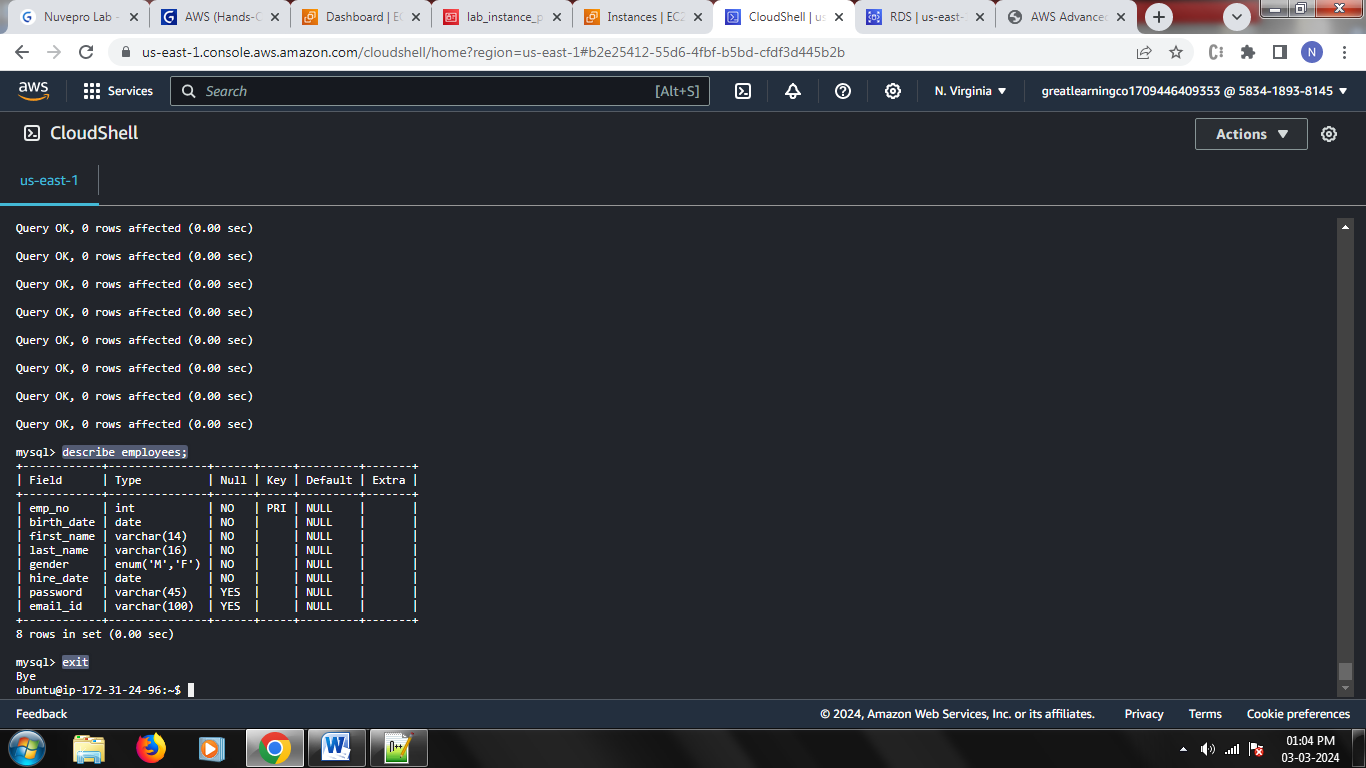


Login into mysql

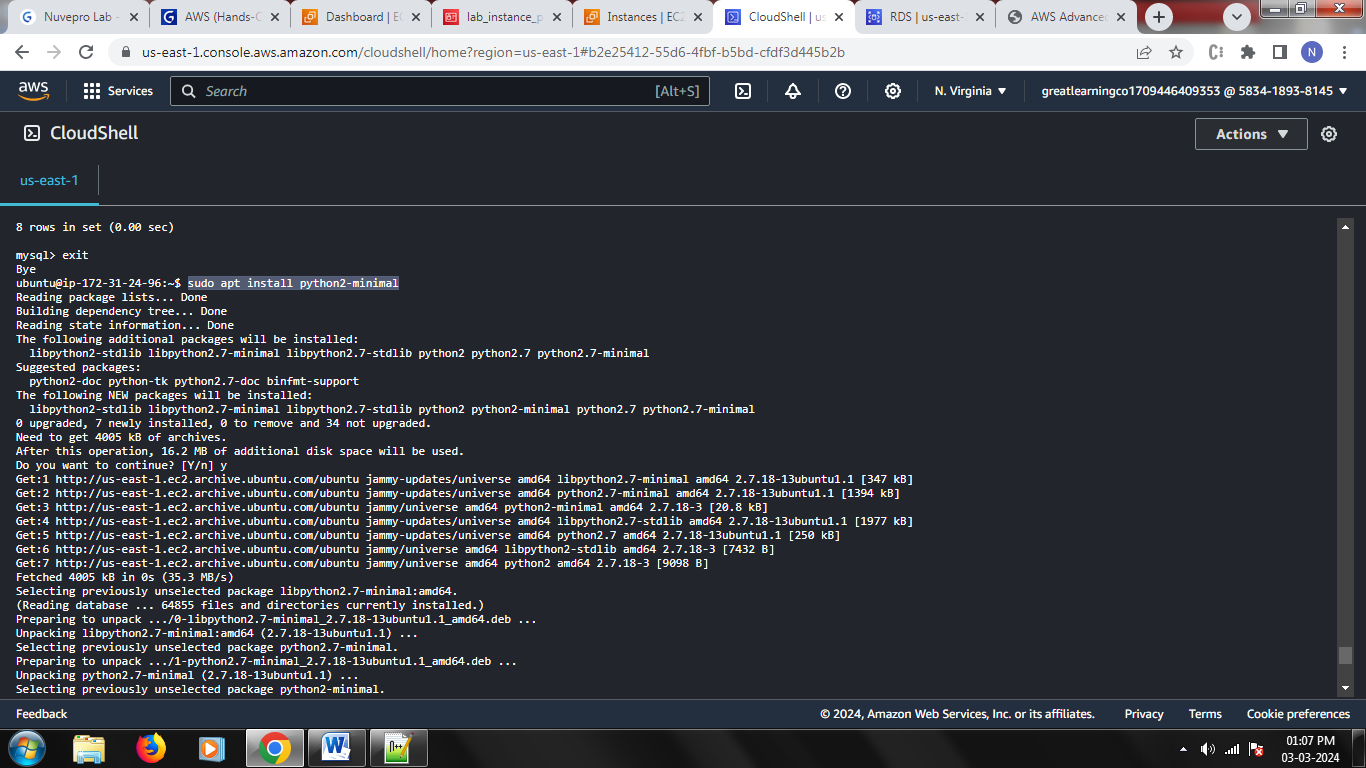


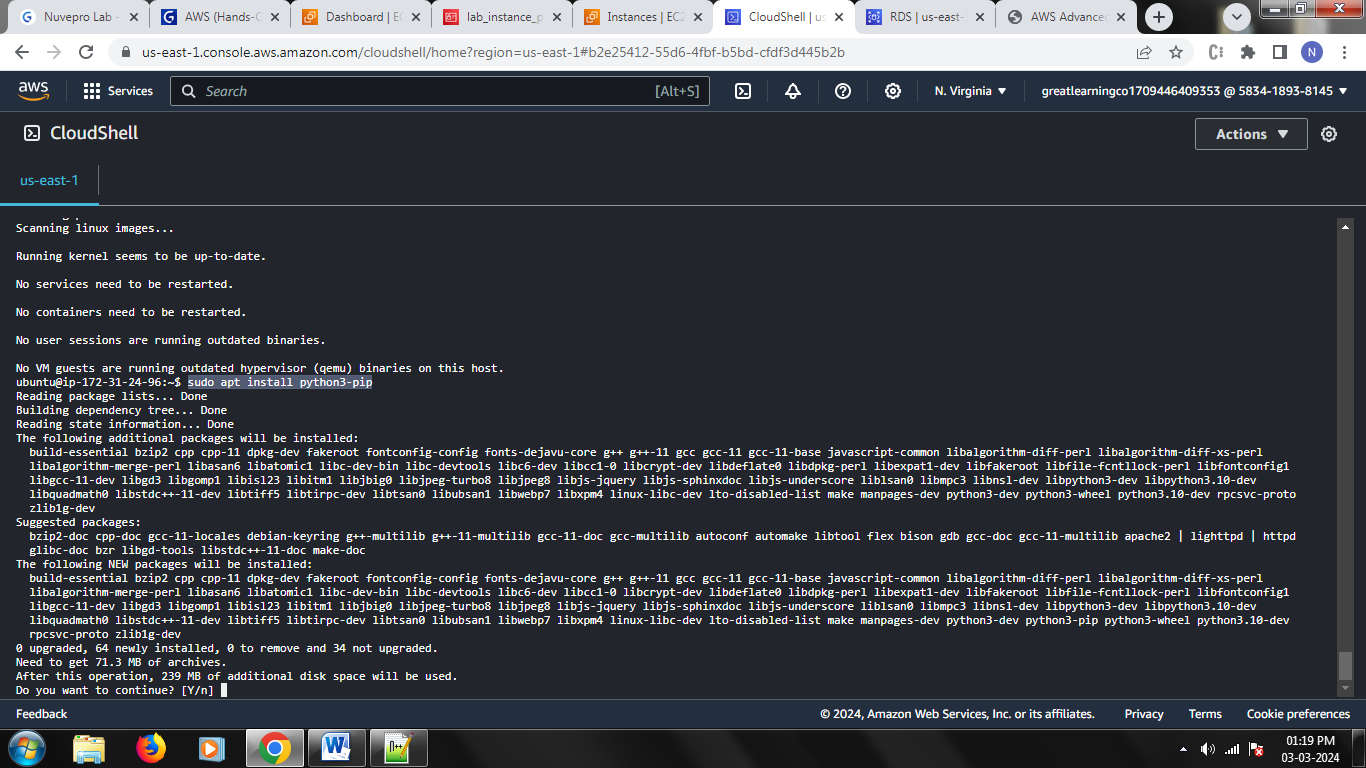
Execute all database commands and create the data

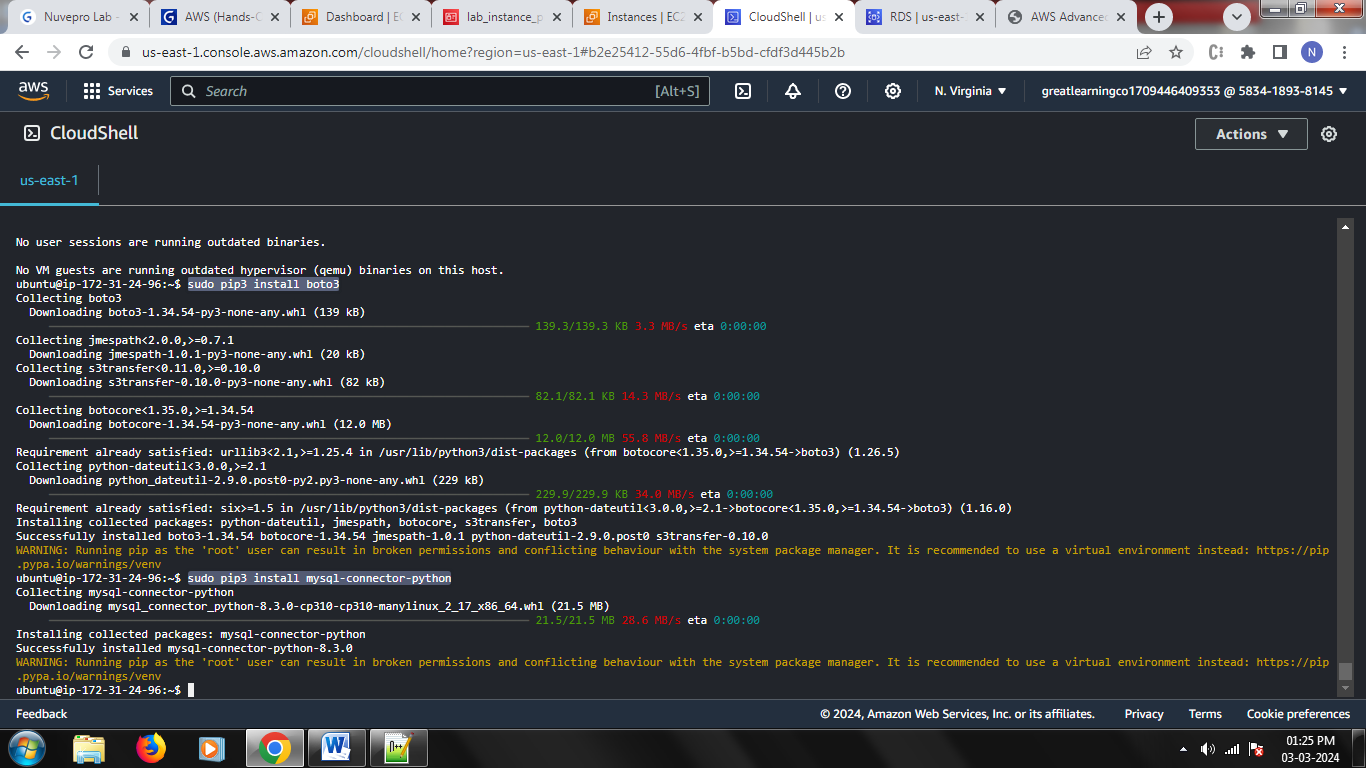


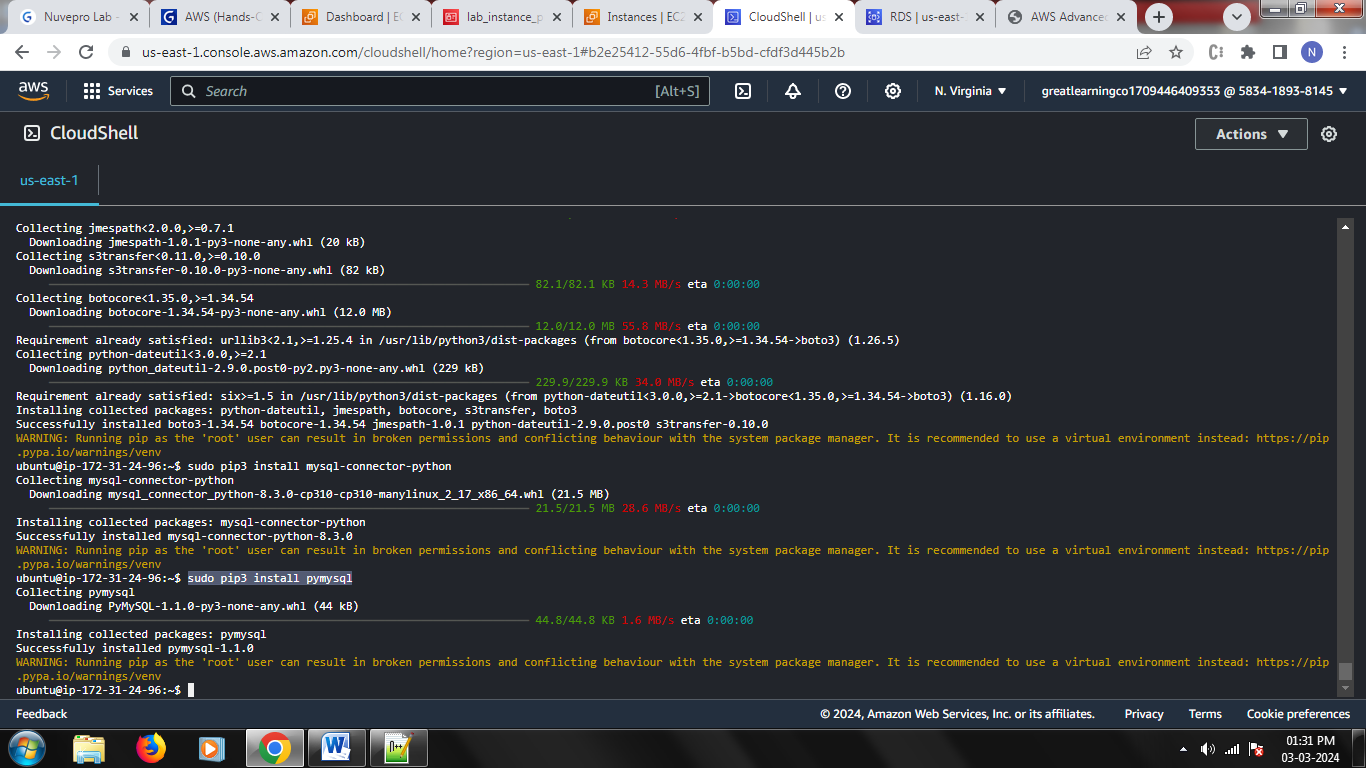


Install all python libraries

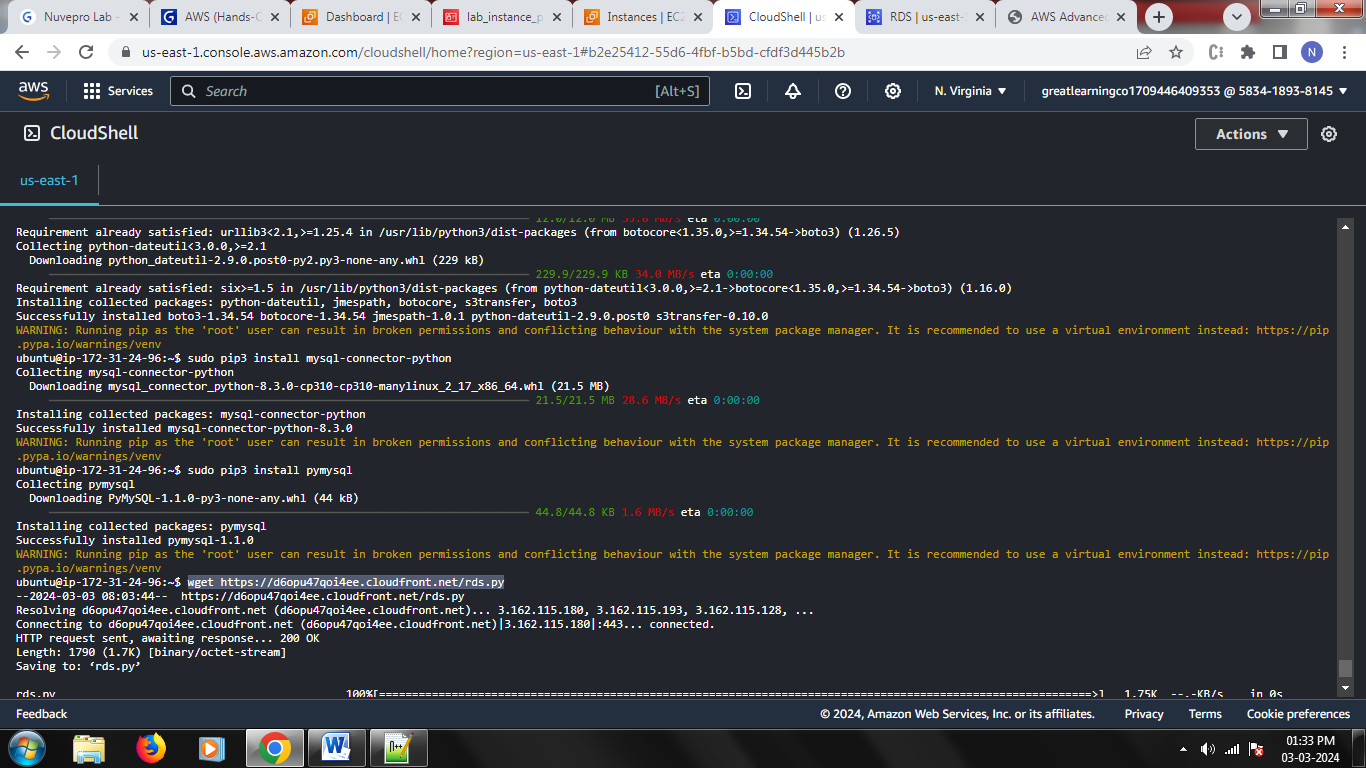




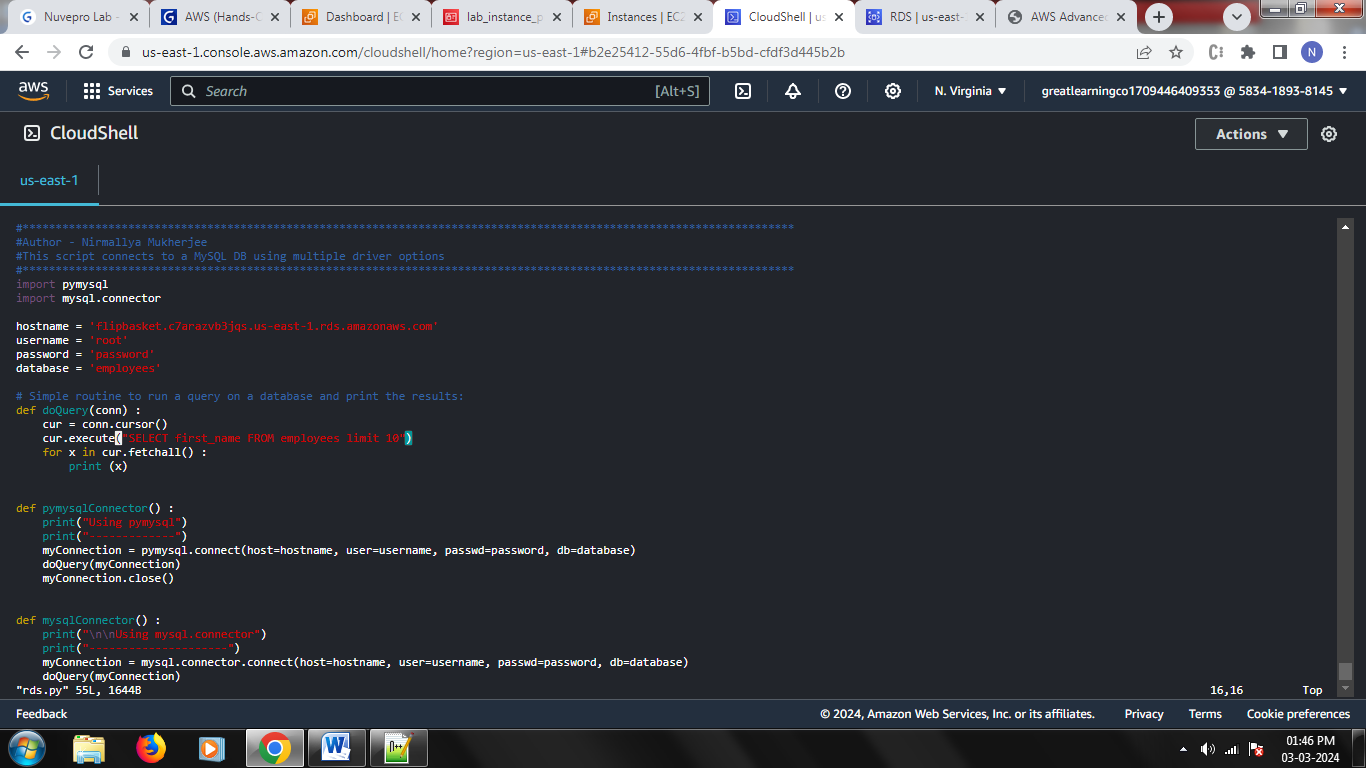




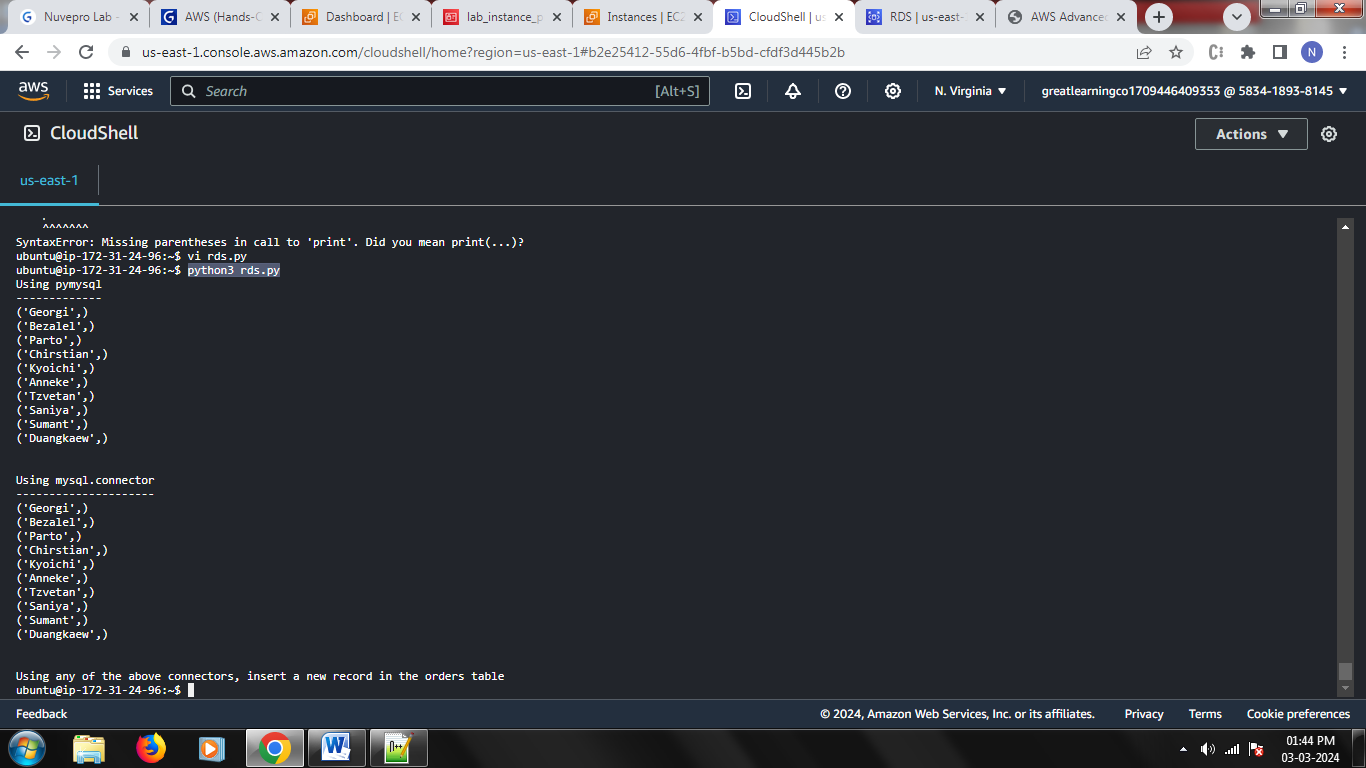
Get the python program



Update the python program to connect with your RDS



Run the python program and get fetch the data from your RDS



## Q: state your observations when you remove the role and rerun the program (Hint - RDS SG)

## If you have an RDS instance with a security group that allows incoming traffic on port 3306 from VPC CIDR block of your EC2 instance, and you have an EC2 instance that can connect to the RDS instance using SSH, you should be able to access the RDS instance without explicitly assigning an IAM role with the AmazonRDSFullAccess policy. Hence even if you remove the role from EC2 instance, the python program will run fine.