Name :- Yogesh Dilip Khairnar **College:-B.K.Birla College**

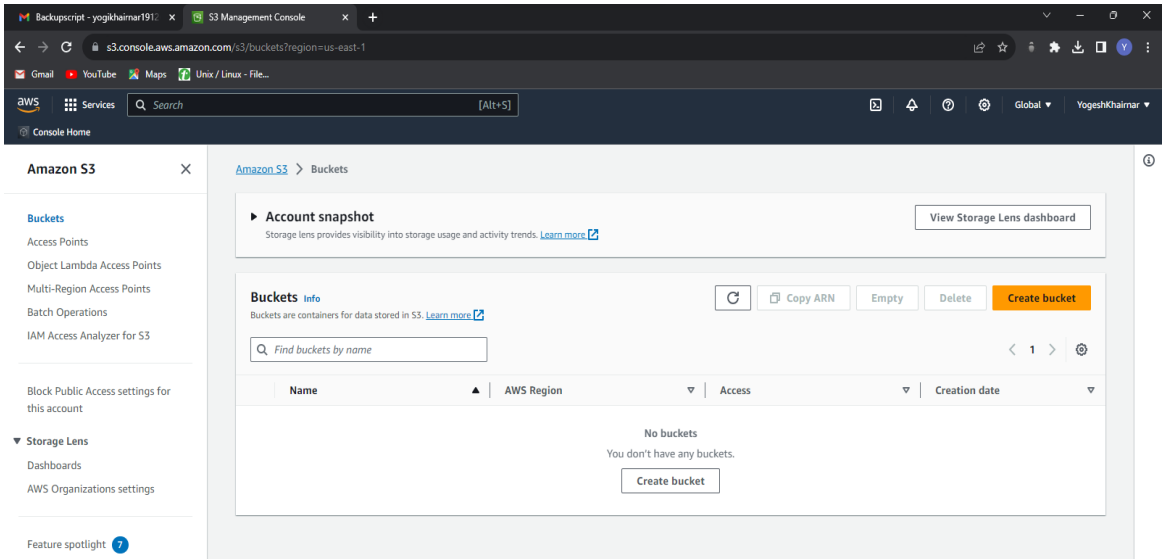
IAC PROJECT(Domain:-Cloud Computing).

Automate MongoDB Backup.

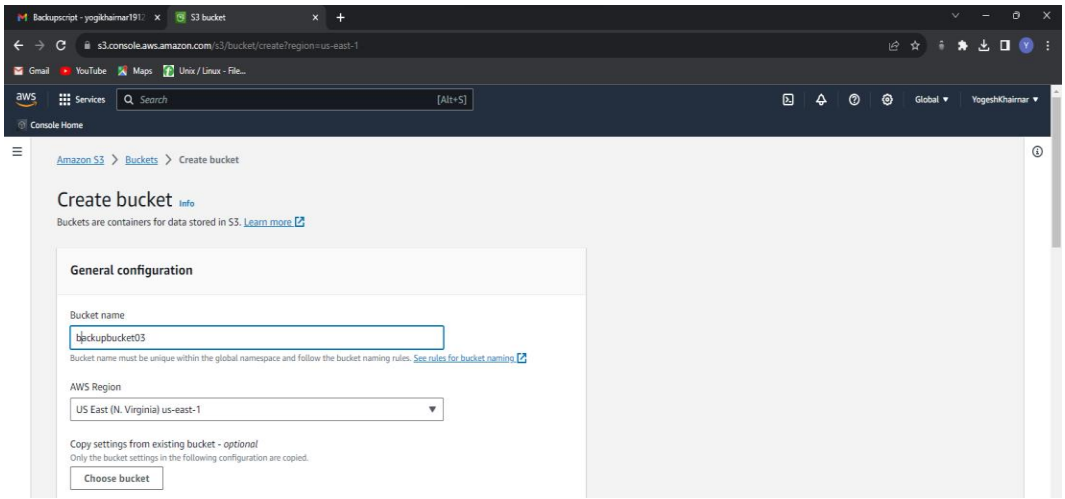
Requirements:-AWS Account,MongoDB.
AWS Services:-EC2,S3,IAM(Role).

Steps:-

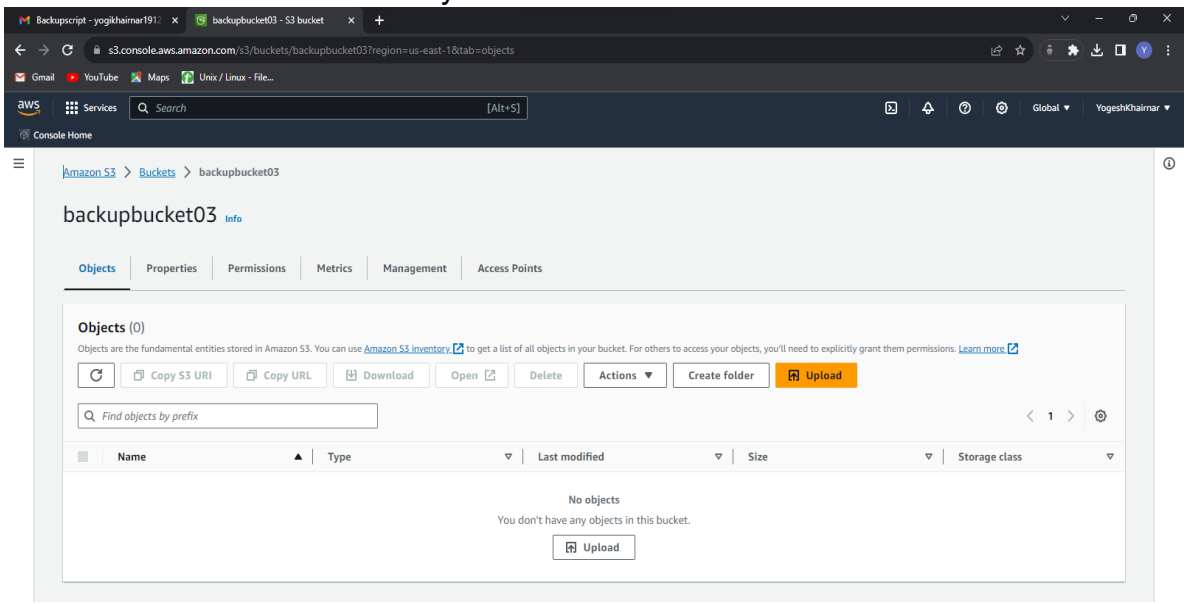
1. S3:-Create a bucket and inside that bucket create a backup folder.
   Login to aws account>Go to management console>search for s3>Bucket>Create bucket.



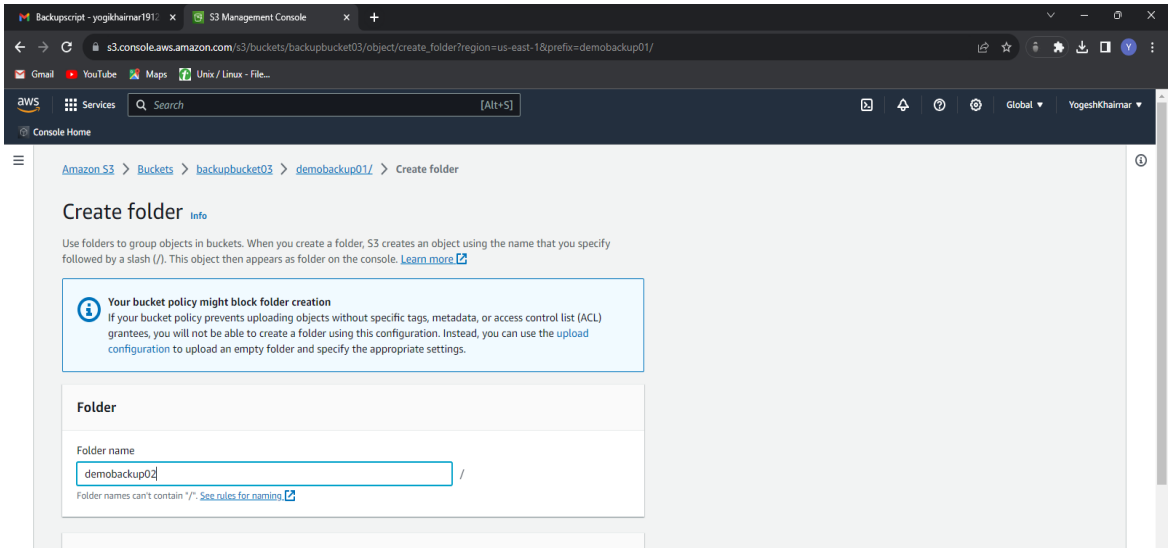
Give name of the bucket>Create bucket.



Now click on the bucket which you have created>Create Folder.



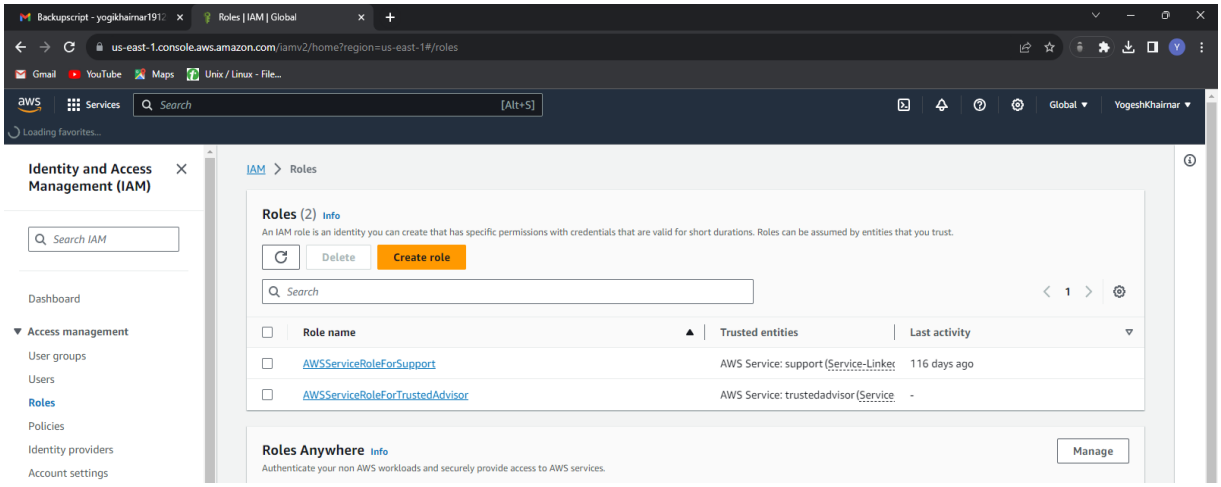
Give the name of folder>Create.



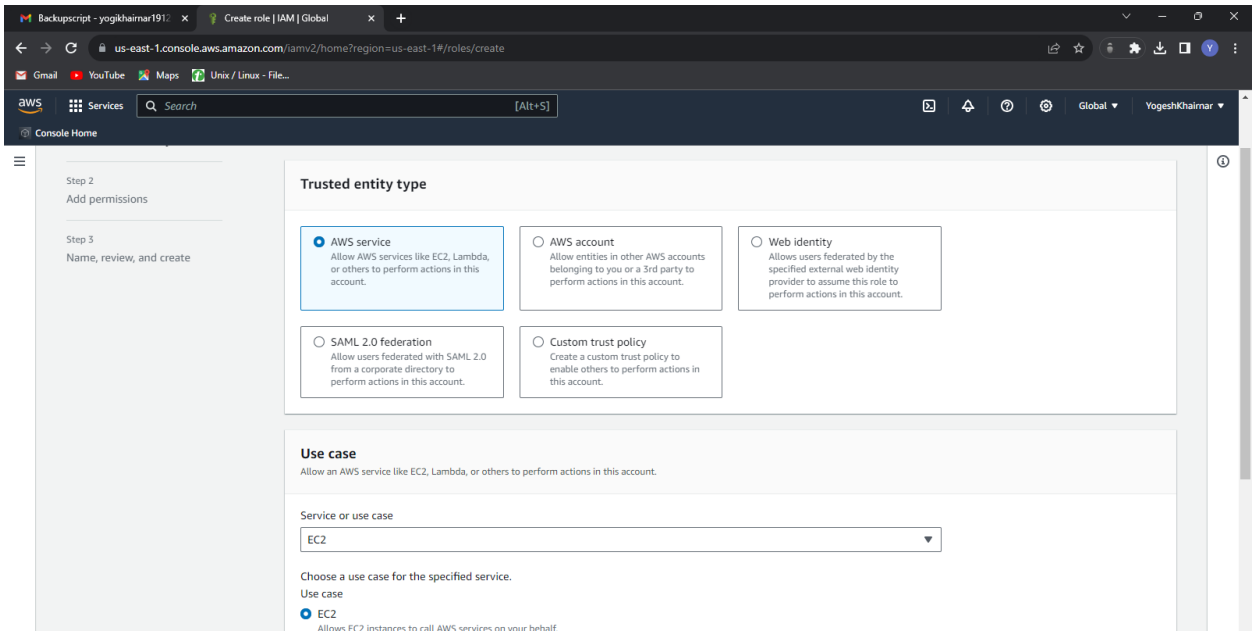
Name :- Yogesh Dilip Khairnar **College:-B.K.Birla College**

2. IAM(Role):-Create a role.

Search for IAM>Role>Create role.

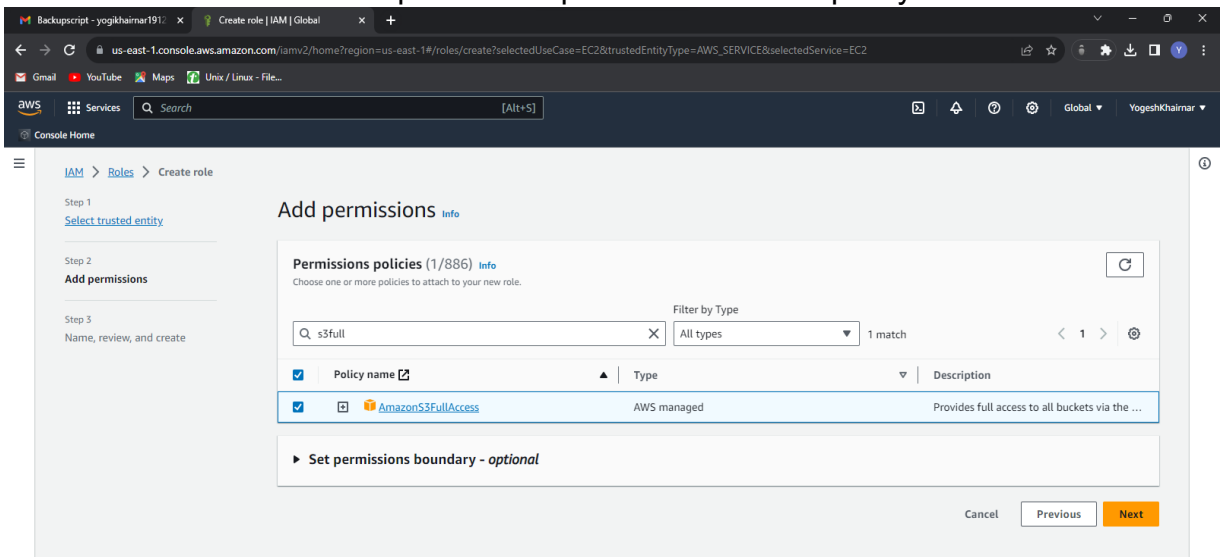


Choose use case as EC2 and keep rest of settings as it is(default)>Next.

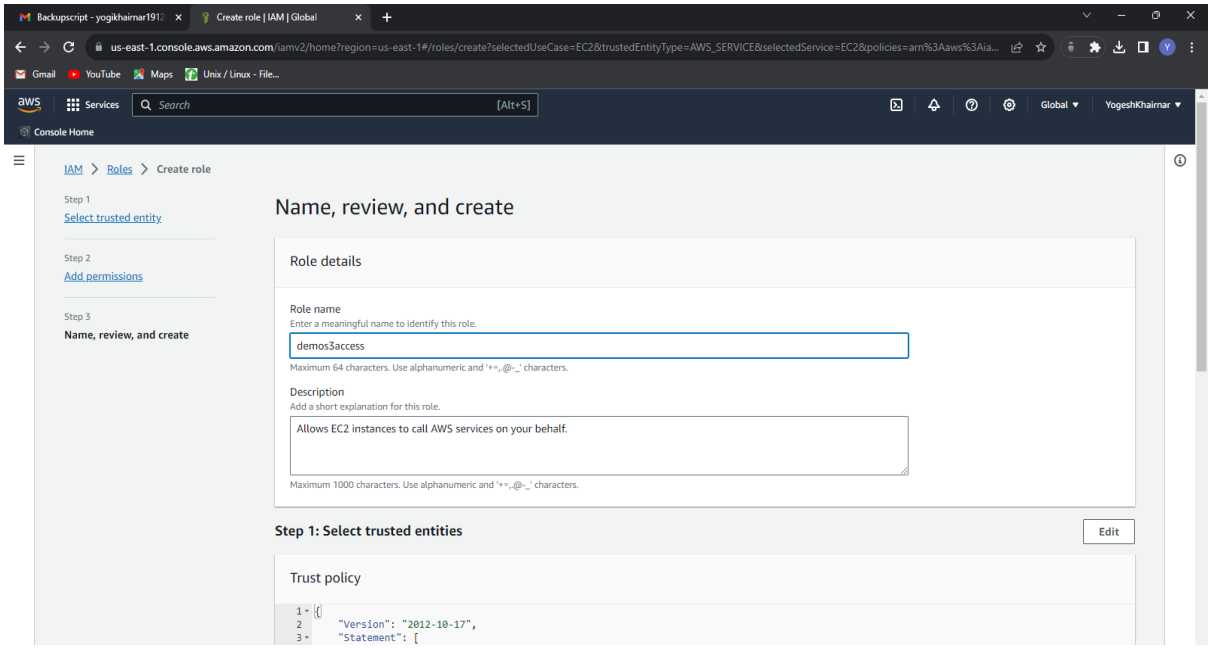


Name :- Yogesh Dilip Khairnar **College:-B.K.Birla College**

Search for S3FullAccess in permission policies select that policy>Next.

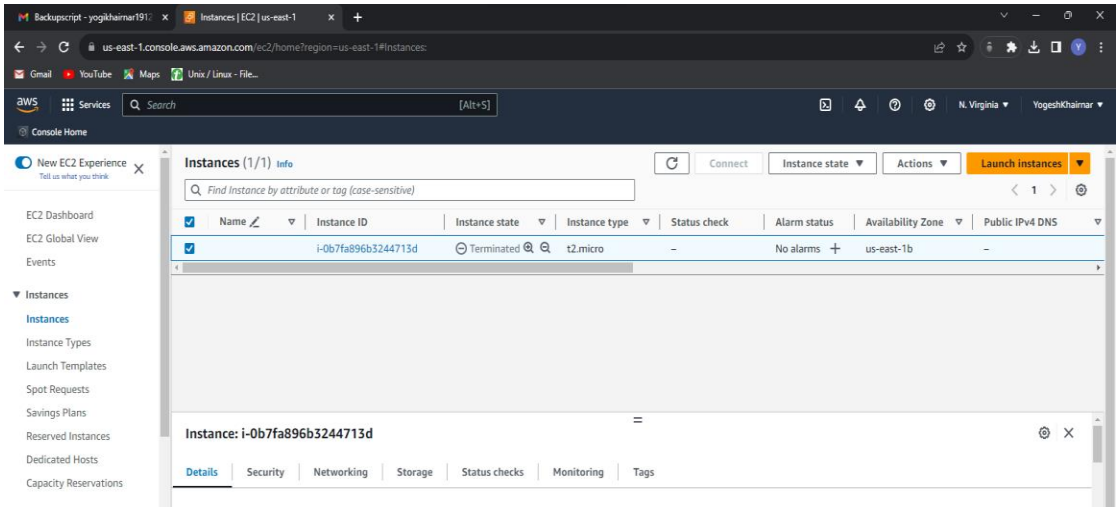


Give the name of role>Create role.

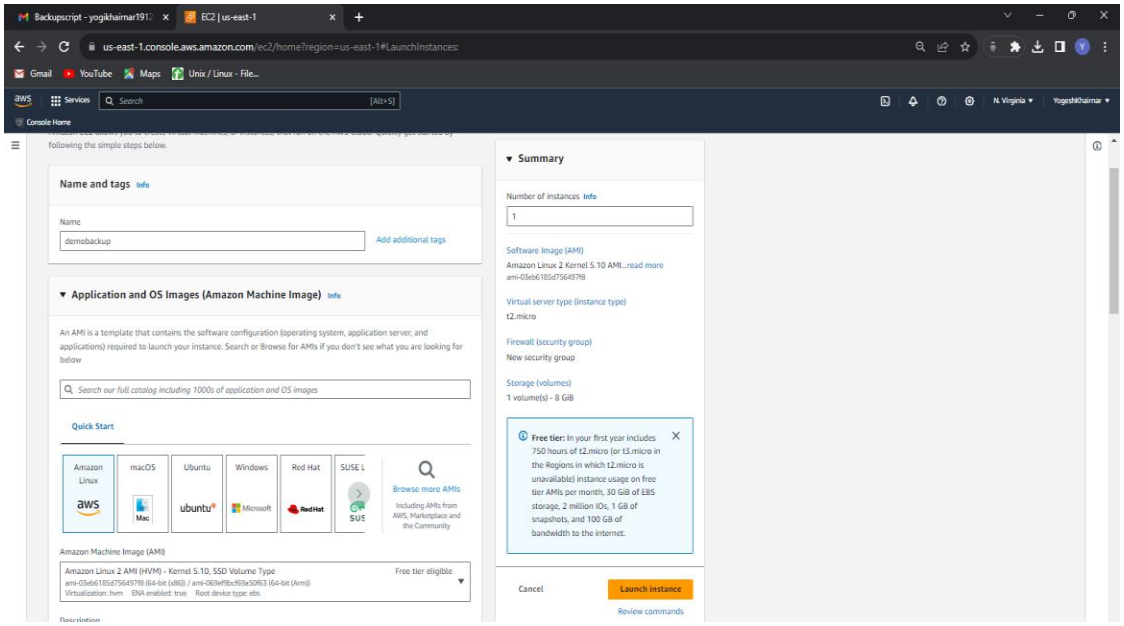


3. EC2:-Launch EC2 Instance.

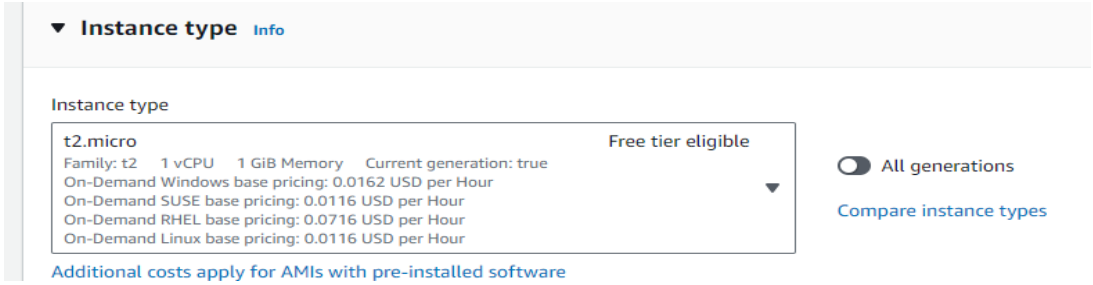
Search for EC2>Instance>Launch Instance.



Give the name of instance>Choose AMI as Amazon Linux2

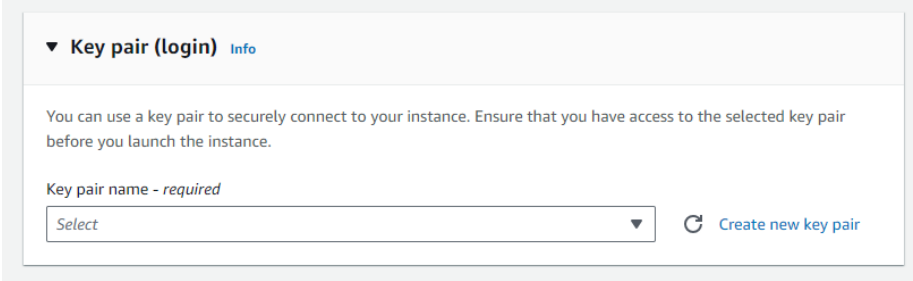


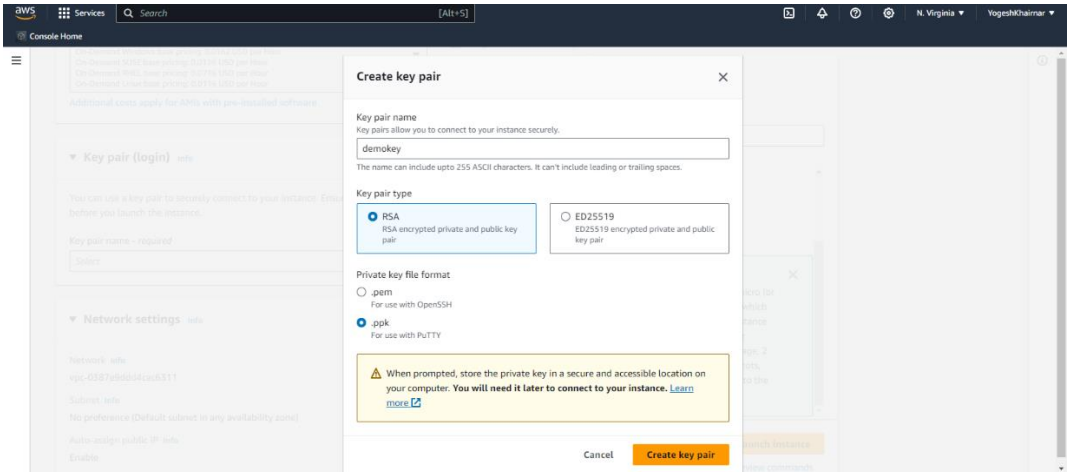
Select Instance type as t2.micro.



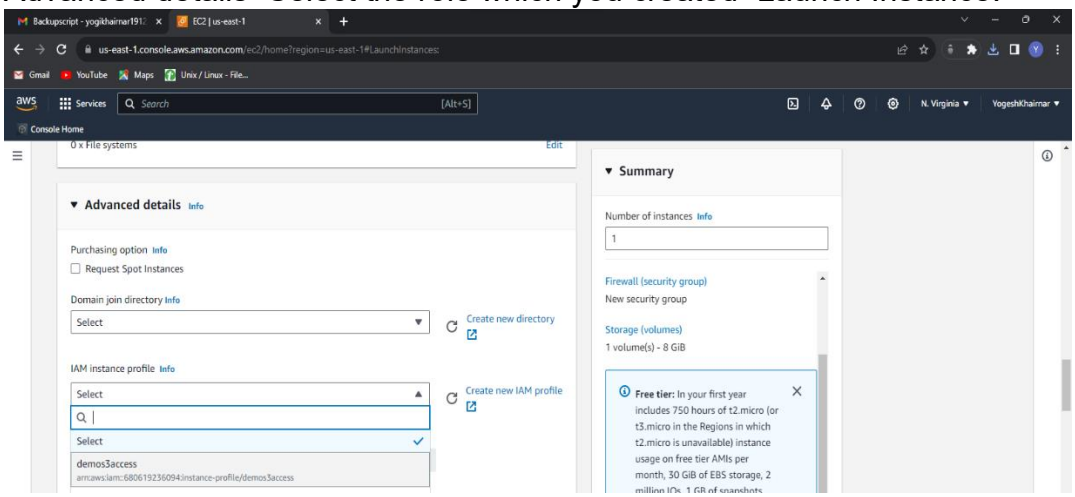
Name :- Yogesh Dilip Khairnar **College:-B.K.Birla College**

Create new key pair>Give key pair name>select key pair type as.ppk>Create.

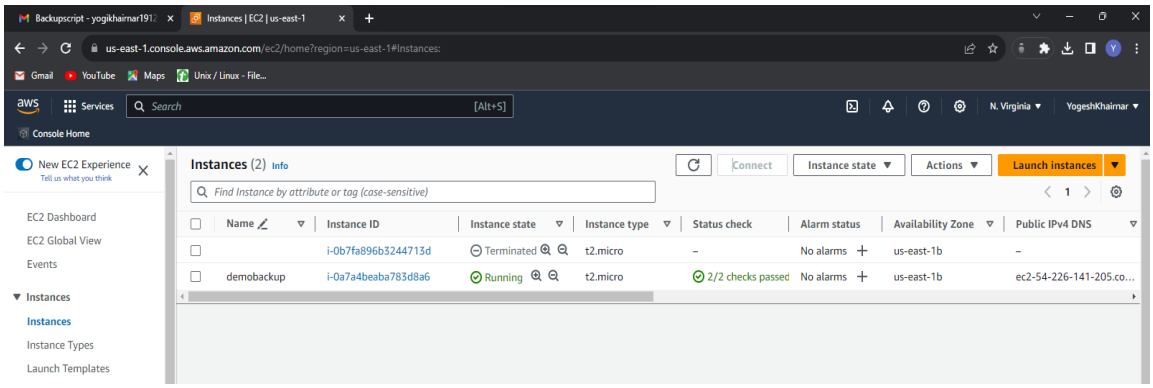




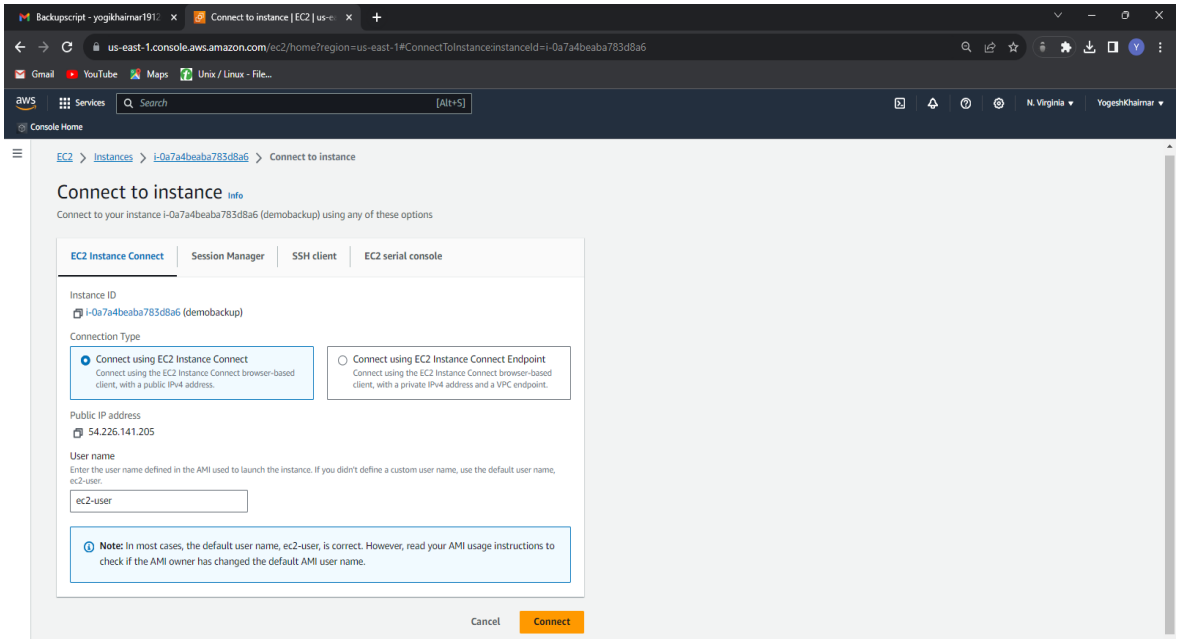
Advanced details>Select the role which you created>Launch Instance.



Wait until the status check pass gets 2/2 once status check gets passed then connect it.



EC2 Instance connect>Connect.

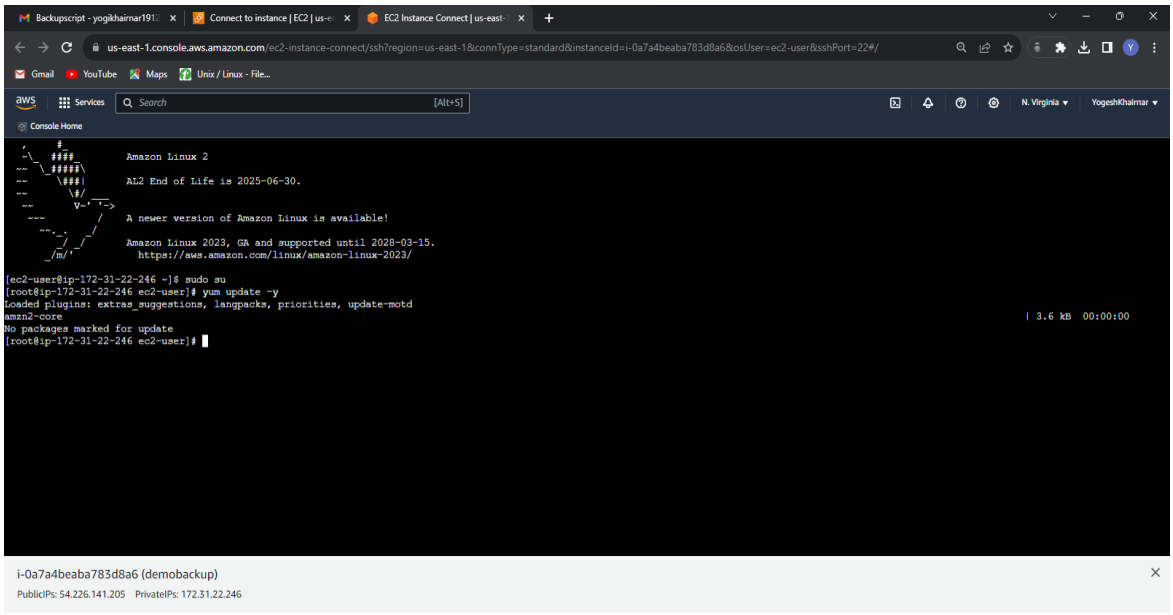


4. Installation of MongoDB and creating a database.

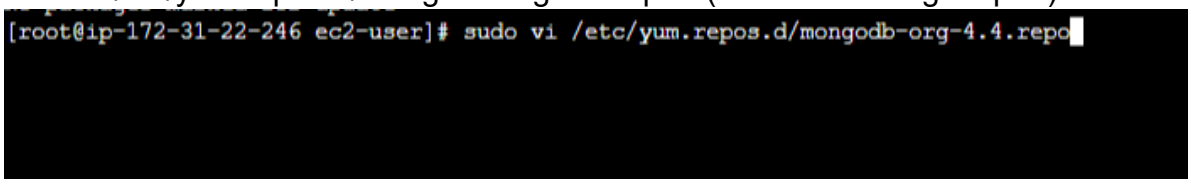
Linux terminal will get open when you connect the instance>write the following commands

sudo su (Enter to root user)

yum update -y (Will update the new dependency if available)



sudo vi /etc/yum.repos.d/mongodb-org-4.4.repo (Vim editor will get open)



In vim editor press **I** write the following

[mongodb-org-4.4]

name=MongoDB Repository

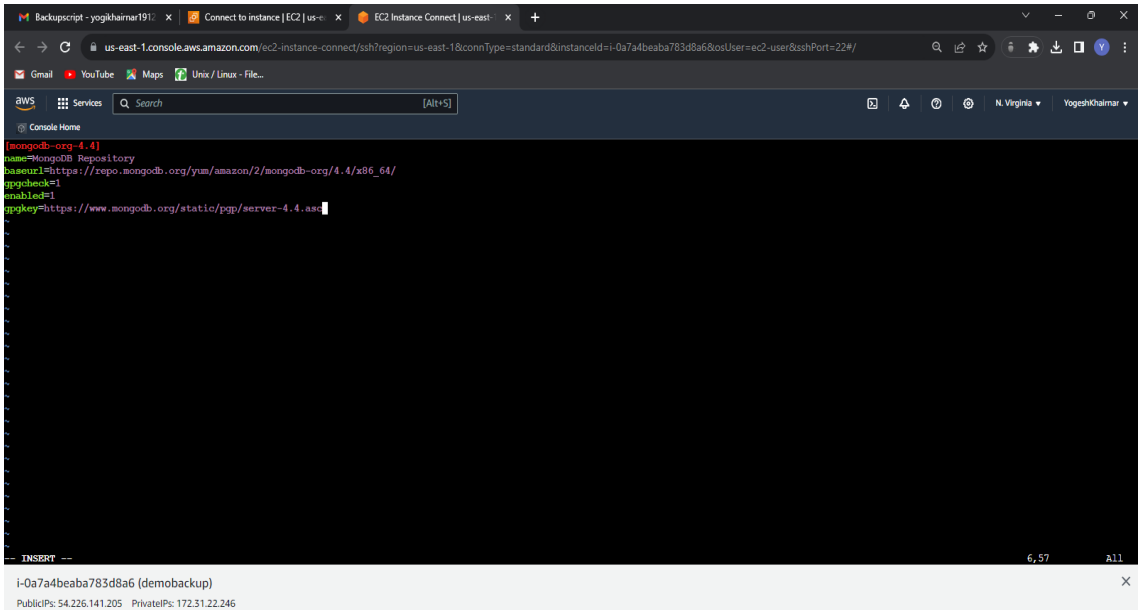
baseurl=https://repo.mongodb.org/yum/amazon/2/mongodb-org/4.4/x86_64/

gpgcheck=1

enabled=1

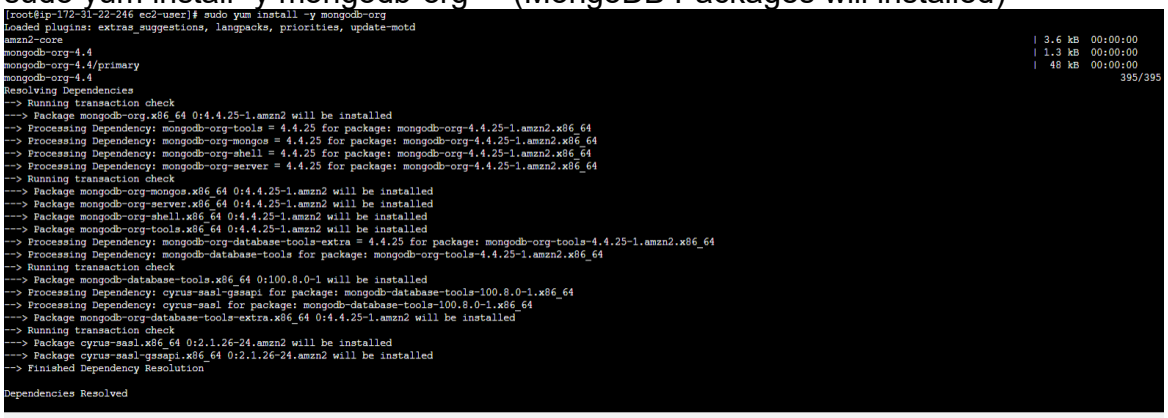
gpgkey=https://www.mongodb.org/static/pgp/server-4.4.asc

Name :- Yogesh Dilip Khairnar **College:-B.K.Birla College**



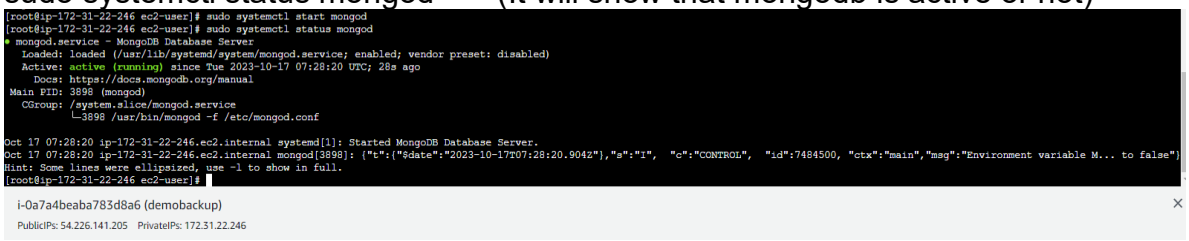
After writing save it by pressing **Esc- shift + : - wq**

sudo yum install -y mongodb-org (MongoDB Packages will installed)



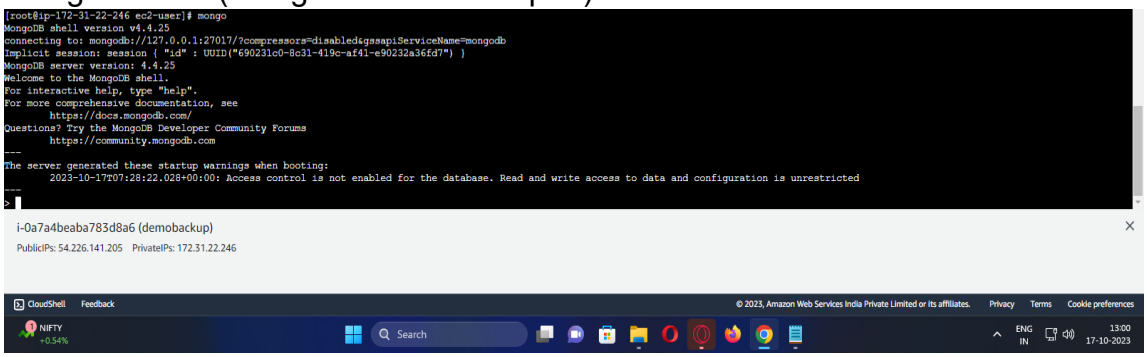
sudo systemctl start mongod (MongoDB will start)

sudo systemctl status mongod (It will show that mongodb is active or not)



Now go to Mongo terminal and create a database.

mongo (mongo terminal will open)

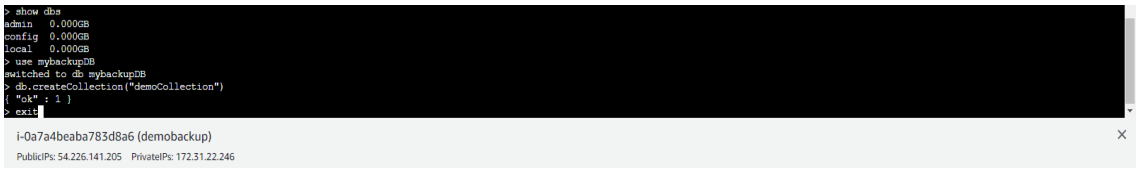


show dbs (It will show the database which are present)

use mybackupDB (Database will be created,you can use other name instead of mybackupDB)

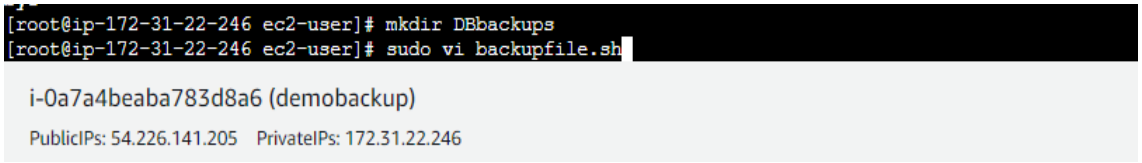
db.createCollection("demoCollection") (Collection of database will be created)

exit



mkdir DBbackups (Directory will be created,you can replace DBbackups it by other name)

sudo vi backupfile.sh (vim editor will open and .sh file will be created)



Editor will open press **I** write the following script

#!/bin/bash

Set the backup folder path

folder_name="/home/ec2-user/**your directory name which you created**/"

Set the backup file name components

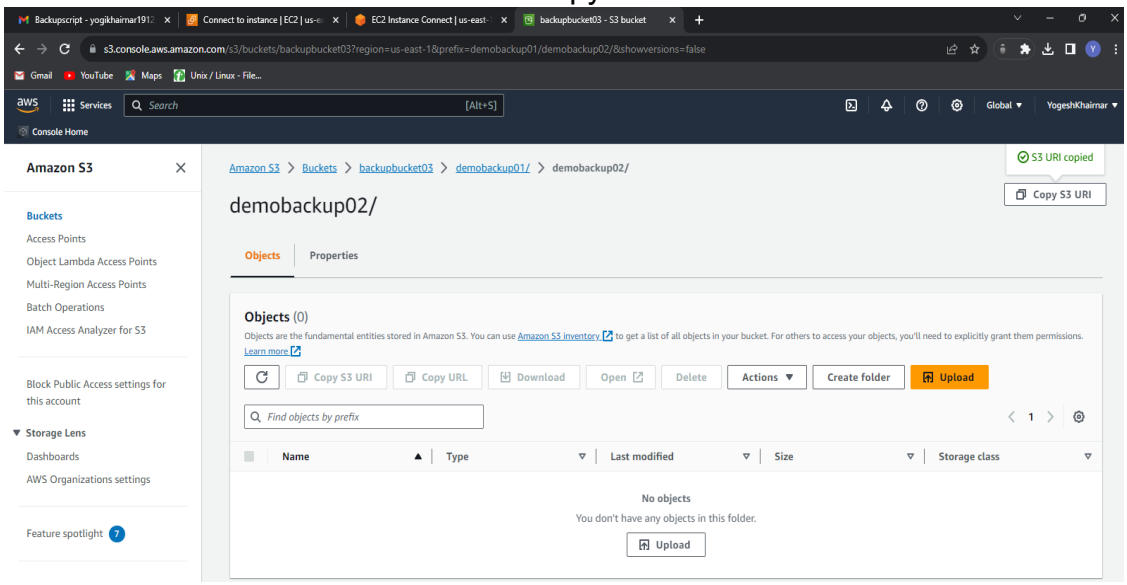
file="backupStorage"

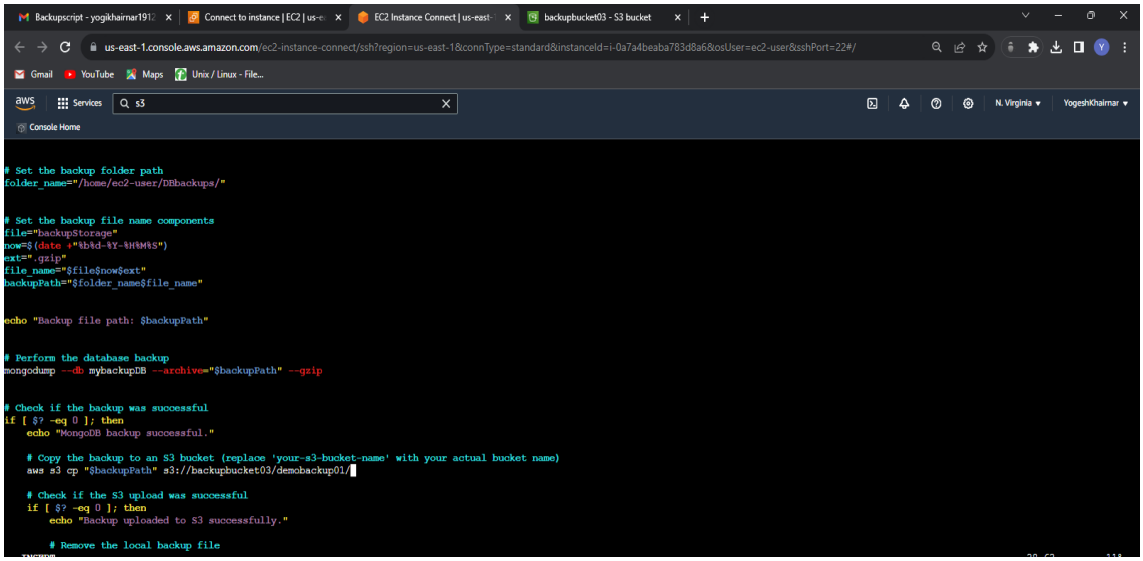
now=$(date +"%b%d-%Y-%H%M%S")

```
ext=".gzip"
file_name="$file$now$ext"
backupPath="$folder_name$file_name"
echo "Backup file path: $backupPath
# Perform the database backup
mongodump --db give your database name --archive="$backupPath" --gzip
# Check if the backup was successful
if [ $? -eq 0 ]; then
   echo "MongoDB backup successful."
# Copy the backup to an S3 bucket (replace 'your-s3-bucket-name' with your actual bucket name)
   aws s3 cp "$backupPath" s3://your-s3-path
 # Check if the S3 upload was successful
   if [ $? -eq 0 ]; then
      echo "Backup uploaded to S3 successfully.
# Remove the local backup file
      rm "$backupPath"
      echo "Local backup file removed."
   else
      echo "Error: Failed to upload the backup to S3."
   fi
else
   echo "Error: MongoDB backup failed."
fi
```

Name :- Yogesh Dilip Khairnar **College:-B.K.Birla College**

for S3 Path-Go to s3>bucket >folder>copy s3url





```
# Set the backup folder path
folder_name="/home/ec2-user/DBbackups/"

# Set the backup file name components
file="backupStorage"
now=$(date +"%b%d-%Y-%H%M%S")
ext=".gzip"
file_name="$file$now$ext"
backupPath="$folder_name$file_name"

echo "Backup file path: $backupPath"

# Perform the database backup
mongodump --db mybackupDB --archive="$backupPath" --gzip

# Check if the backup was successful
if [ $? -eq 0 ]; then
    echo "MongoDB backup successful."

    # Copy the backup to an S3 bucket (replace 'your-s3-bucket-name' with your actual bucket name)
    aws s3 cp "$backupPath" s3://backupbucket03/demobackup01/

    # Check if the S3 upload was successful
    if [ $? -eq 0 ]; then
        echo "Backup uploaded to S3 successfully."

        # Remove the local backup file
```

Save the script by pressing **Esc – shift + : - wq**

chmod 700 backupfile.sh (It will give all permission to file)

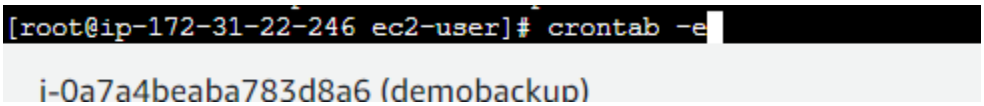
./backupfile.sh (It will execute the file)

```
[root@ip-172-31-22-246 ec2-user]# sudo chmod 700 backupfile.sh
[root@ip-172-31-22-246 ec2-user]# sudo ./backupfile.sh
Backup file path: /home/ec2-user/DBbackups/backupStorageOct17-2023-074700.gzip
2023-10-17T07:47:00.653+0000    writing mybackupDB.demoCollection to archive '/home/ec2-user/DBbackups/backupStorageOct17-2023-074700.gzip'
2023-10-17T07:47:00.657+0000    done dumping mybackupDB.demoCollection (0 documents)
MongoDB backup successful.
upload failed: DBbackups/backupStorageOct17-2023-074700.gzip to s3://backupbucket03/demobackup01/backupStorageOct17-2023-074700.gzip Unable to locate credentials
Error: Failed to upload the backup to S3.
[root@ip-172-31-22-246 ec2-user]#
```

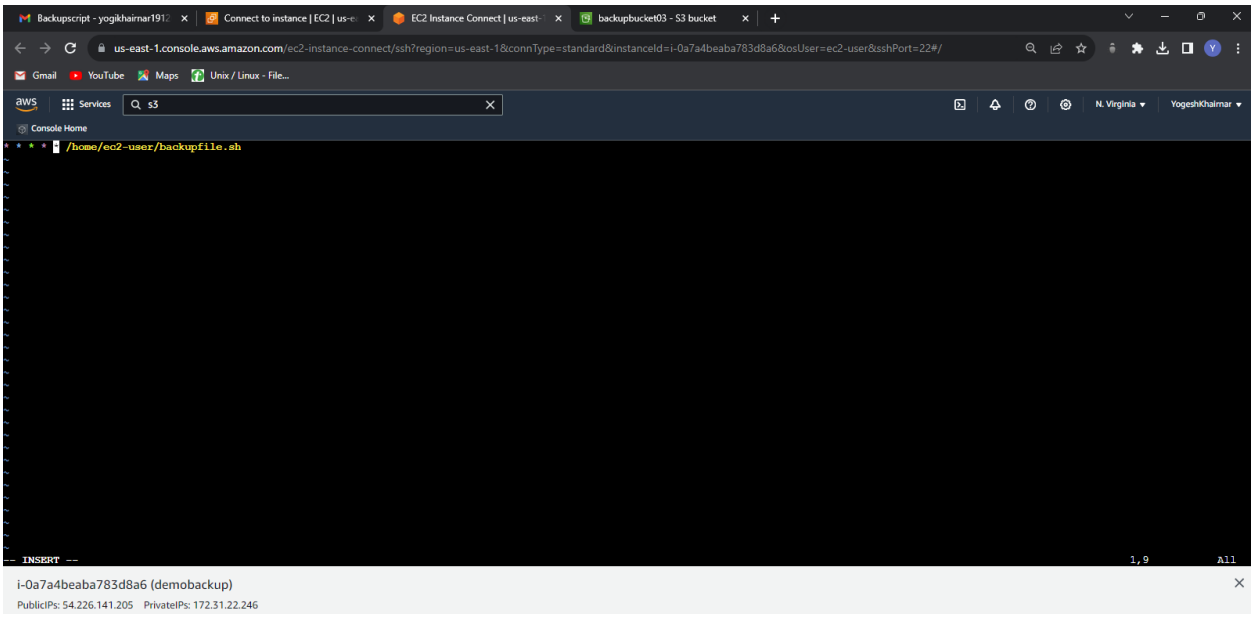
Now we have to schedule the file so that the backup file will be automatically backuped in every one minute

Write the following command

crontab -e



* * * * * /home/ec2-user/**give your backupfile name** (i.e .sh file)



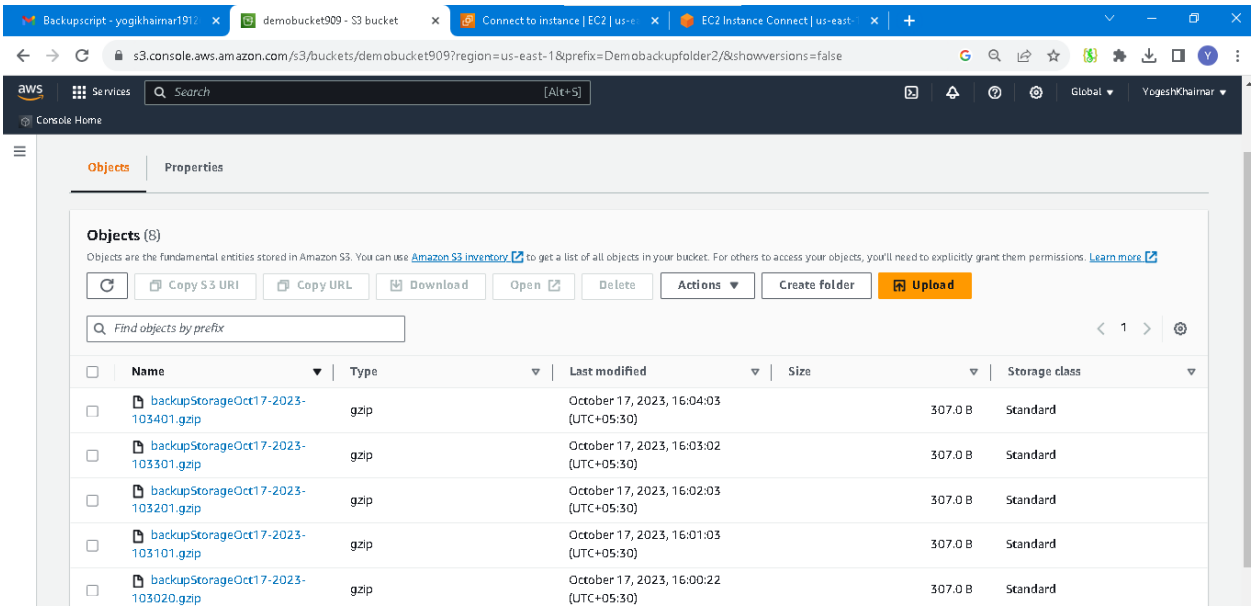
Save it by pressing **Esc – shift + : - wq**

Again execute the file

./backupfile.sh

Name :- Yogesh Dilip Khairnar **College:-B.K.Birla College**

Go to s3>bucket>folder you will see that the files have been backuped in every 1 min



Hence this can be one of the solution to automate backup of MongoDB database.