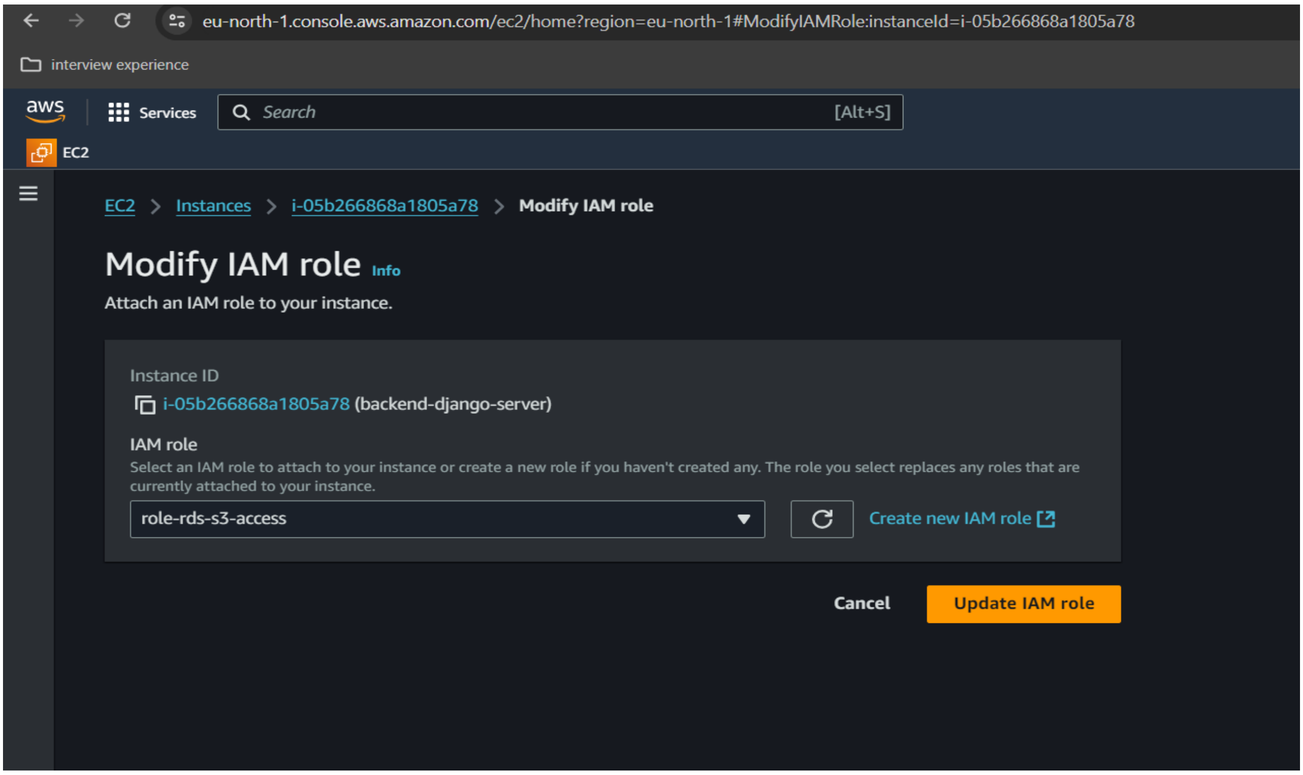
Automating Daily RDS Backups to an S3 Bucket from an EC2 Instance

1. Create an IAM Role with Required Permissions

1. **In the AWS Management Console**:
   - Navigate to **IAM** > **Roles** > **Create Role**.
2. **Select Trusted Entity**:
   - Choose **AWS Service** > **EC2** > **Next**.
3. **Attach Policies**:
   - Attach AmazonS3FullAccess to allow the EC2 instance to upload backups to your S3 bucket.
   - Optionally, attach AmazonRDSReadOnlyAccess if you require RDS read access for more tasks.
4. **Name the Role**:
   - Name the role (e.g., EC2-RDSBackupRole) and complete the setup.
5. **Attach Role to EC2 Instance**:
   - Go to **EC2 Console** > **Instances**, select your instance, go to **Actions** > **Security** > **Modify IAM Role**, and attach the newly created IAM role.



eu-north-1.console.aws.amazon.com/ec2/home?region=eu-north-1#ModifyIAMRole:instanceId=i-05b266868a1805a78

EC2 > Instances > i-05b266868a1805a78 > Modify IAM role

Modify IAM role

Attach an IAM role to your instance.

Instance ID

i-05b266868a1805a78 (backend-django-server)

IAM role

Select an IAM role to attach to your instance or create a new role if you haven't created any. The role you select replaces any roles that are currently attached to your instance.

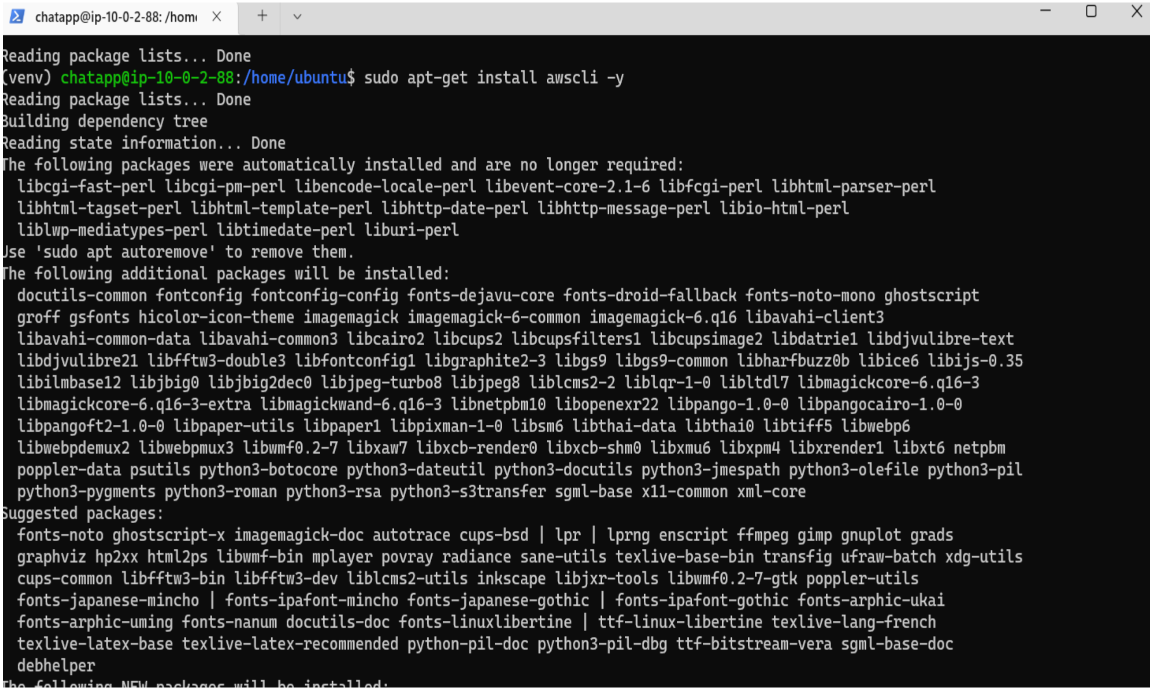
role-rds-s3-access

2. Install the AWS CLI on the EC2 Instance

1. **SSH into Your EC2 Instance**:
    - Access your instance via SSH.
2. **Install the AWS CLI**:

```
sudo apt-get update

sudo apt-get install awscli -y
```



```
Reading package lists... Done
(venv) chatapp@ip-10-0-2-88:/home/ubuntu$ sudo apt-get install awscli -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libcgi-fast-perl libcgi-pm-perl libencode-locale-perl libevent-core-2.1-6 libfcgi-perl libhtml-parser-perl
  libhtml-tagset-perl libhtml-template-perl libhttp-date-perl libhttp-message-perl libio-html-perl
  liblwp-mediatypes-perl libtimedate-perl liburi-perl
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  docutils-common fontconfig fontconfig-config fonts-dejavu-core fonts-droid-fallback fonts-noto-mono ghostscript
  groff gsfonts hicolor-icon-theme imagemagick imagemagick-6-common imagemagick-6.q16 libavahi-client3
  libavahi-common-data libavahi-common3 libcairo2 libcups2 libcupsfilters1 libcupsimage2 libdatrie1 libdjvulibre-text
  libdjvulibre21 libfftw3-double3 libfontconfig1 libgraphite2-3 libgs9 libgs9-common libharfbuzz0b libice6 libijs-0.35
  libilmbase12 libjbig0 libjbig2dec0 libjpeg-turbo8 libjpeg8 liblcms2-2 liblqr-1-0 libltdl7 libmagickcore-6.q16-3
  libmagickcore-6.q16-3-extra libmagickwand-6.q16-3 libnetpbm10 libopenexr22 libpango-1.0-0 libpangocairo-1.0-0
  libpangoft2-1.0-0 libpaper-utils libpaper1 libpixman-1-0 libsm6 libthai-data libthai0 libtiff5 libwebp6
  libwebpdemux2 libwebpmux3 libwmf0.2-7 libxaw7 libxcb-render0 libxcb-shm0 libxmu6 libxpm4 libxrender1 libxt6 netpbm
  poppler-data psutils python3-botocore python3-dateutil python3-docutils python3-jmespath python3-olefile python3-pil
  python3-pygments python3-roman python3-rsa python3-s3transfer sgml-base x11-common xml-core
Suggested packages:
  fonts-noto ghostscript-x imagemagick-doc autotrace cups-bsd | lpr | lprng enscript ffmpeg gimp gnuplot grads
  graphviz hp2xx html2ps libwmf-bin mplayer povray radiance sane-utils texlive-base-bin transfig ufraw-batch xdg-utils
  cups-common libfftw3-bin libfftw3-dev liblcms2-utils inkscape libjxr-tools libwmf0.2-7-gtk poppler-utils
  fonts-japanese-mincho | fonts-ipafont-mincho fonts-japanese-gothic | fonts-ipafont-gothic fonts-arphic-ukai
  fonts-arphic-uming fonts-nanum docutils-doc fonts-linuxlibertine | ttf-linux-libertine texlive-lang-french
  texlive-latex-base texlive-latex-recommended python-pil-doc python3-pil-dbg ttf-bitstream-vera sgml-base-doc
  debhelper
The following NEW packages will be installed:
```

3. Create the Backup Script

1. **Navigate to Your Project Directory**:
    - For example:

```
cd /new_chatapp/venv
```

2. **Create the Script File**:
    - Open a new file with nano:

```
nano /new_chatapp/venv/backup_script.sh
```

3. Add the Following Code to the Script:

```
#!/bin/bash


# Load environment variables
export DB_NAME="rdsdb"
export DB_USER="admin"
export DB_PASSWORD="Admin0987"
export DB_HOST="django-application-db.c7c60i4sgkr5.eu-north-1.rds.amazonaws.com"
export DB_PORT="3306"
export S3_BUCKET="aayush-s3-bucket1233"  # Your S3 bucket name


TIMESTAMP=$(date +"%Y%m%d%H%M")
LOG_FILE="/new_chatapp/venv/backup.log"


# Backup RDS database
mysqldump -h "$DB_HOST" -u "$DB_USER" -p"$DB_PASSWORD" --port="$DB_PORT" "$DB_NAME" >
"/tmp/db_backup_$TIMESTAMP.sql"


# Check if mysqldump was successful
if [ $? -eq 0 ]; then
  echo "$(date +'%Y-%m-%d %H:%M:%S') - Database backup successful." >> "$LOG_FILE"


  # Upload backup to S3
  aws s3 cp "/tmp/db_backup_$TIMESTAMP.sql" "s3://$S3_BUCKET/db_backup_$TIMESTAMP.sql"


  # Check if upload was successful
  if [ $? -eq 0 ]; then
    echo "$(date +'%Y-%m-%d %H:%M:%S') - Backup uploaded to S3 successfully." >> "$LOG_FILE"
  else
    echo "$(date +'%Y-%m-%d %H:%M:%S') - Failed to upload backup to S3." >> "$LOG_FILE"
  fi
else
```

```
    echo "$(date +'%Y-%m-%d %H:%M:%S') - Database backup failed." >> "$LOG_FILE"
fi


# Optionally, remove local backup
rm "/tmp/db_backup_$TIMESTAMP.sql"
```

4. Save and Close:

- Press Ctrl + X, then Y, and Enter to save and exit.

5. Make the Script Executable:

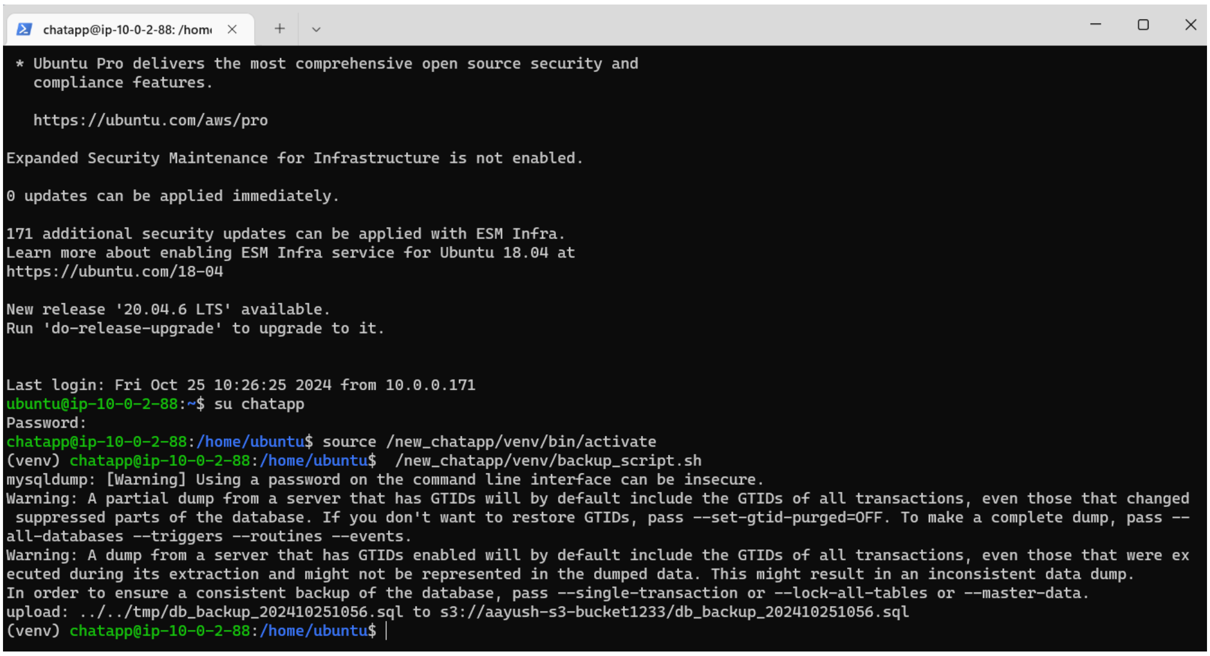
```
chmod +x /new_chatapp/venv/backup_script.sh
```

---

4. Run the Script Manually to Test

```
/new_chatapp/venv/backup_script.sh
```

Check /new_chatapp/venv/backup.log to ensure the backup and upload are successful.



```
 * Ubuntu Pro delivers the most comprehensive open source security and
   compliance features.

   https://ubuntu.com/aws/pro

Expanded Security Maintenance for Infrastructure is not enabled.

0 updates can be applied immediately.

171 additional security updates can be applied with ESM Infra.
Learn more about enabling ESM Infra service for Ubuntu 18.04 at
https://ubuntu.com/18-04

New release '20.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


Last login: Fri Oct 25 10:26:25 2024 from 10.0.0.171
ubuntu@ip-10-0-2-88:~$ su chatapp
Password:
chatapp@ip-10-0-2-88:/home/ubuntu$ source /new_chatapp/venv/bin/activate
(venv) chatapp@ip-10-0-2-88:/home/ubuntu$  /new_chatapp/venv/backup_script.sh
mysqldump: [Warning] Using a password on the command line interface can be insecure.
Warning: A partial dump from a server that has GTIDs will by default include the GTIDs of all transactions, even those that changed
 suppressed parts of the database. If you don't want to restore GTIDs, pass --set-gtid-purged=OFF. To make a complete dump, pass --
all-databases --triggers --routines --events.
Warning: A dump from a server that has GTIDs enabled will by default include the GTIDs of all transactions, even those that were ex
ecuted during its extraction and might not be represented in the dumped data. This might result in an inconsistent data dump.
In order to ensure a consistent backup of the database, pass --single-transaction or --lock-all-tables or --master-data.
upload: ../../tmp/db_backup_202410251056.sql to s3://aayush-s3-bucket1233/db_backup_202410251056.sql
(venv) chatapp@ip-10-0-2-88:/home/ubuntu$ |
```

---

5. Schedule the Script Using Cron

1. **Open Crontab**:

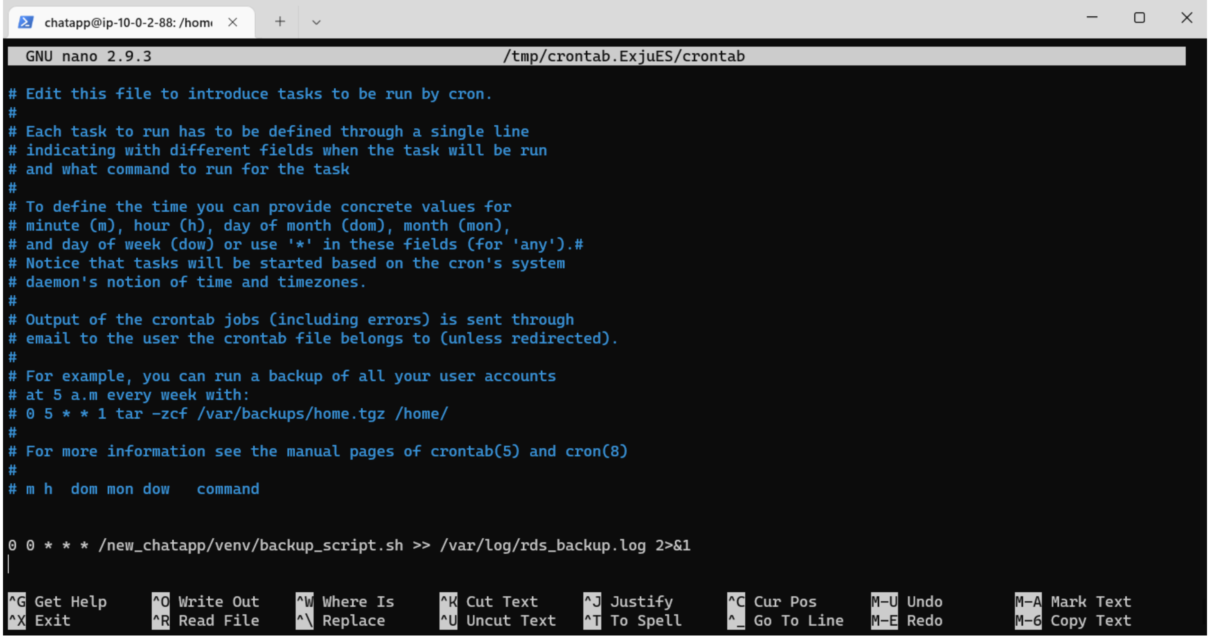
```
crontab -e
```

2. **Add a Cron Job to Run Daily at Midnight**:

0 0 * * * /new_chatapp/venv/backup_script.sh >> /new_chatapp/venv/backup.log 2>&1

3. **Save and Exit Crontab**.

This will execute the script every day at midnight and log output to backup.log.



```
  GNU nano 2.9.3                                   /tmp/crontab.ExjuES/crontab

# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command


0 0 * * * /new_chatapp/venv/backup_script.sh >> /var/log/rds_backup.log 2>&1


^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify     ^C Cur Pos     M-U Undo       M-A Mark Text
^X Exit        ^R Read File   ^\ Replace     ^U Uncut Text  ^T To Spell    ^_ Go To Line  M-E Redo       M-6 Copy Text
```

Goals

- **Automate RDS Backup**: Ensure daily backup of RDS data to secure S3 storage.
- **Minimize Manual Intervention**: Using IAM roles and automation to streamline access and backup operations.
- **Logging and Monitoring**: Keep logs of each backup operation with timestamps for transparency and troubleshooting.