1. Create AWS free tier accounts

2. Login to root user for admin

3. Create new IAM account for a child account (movie analytics)

4. Add permission to IAM account in the root user

5. Now go to EC2 instance, select ubuntu os and t2.micro, create a key pair and select allow https from insternet.( t2.small and above for airflow)

6. Create key pair and save it to ssh folder in local system

7. Go to ec2 instance which is created and go to security groups and edit outbound rules. Add new port 8080 and select anywhere to access the airflow.

-----------------------------------------------------------------------------------------------------------------------

Connect Ec2 to visual code

1. Click the >< icon, connect to host

2. Click Configure SSH hosts

3. Click the config file to edit

3. Add the below code:

Host instance\_name

HostName public\_ipv4\_address\_from\_ec2\_instance

IdentityFile "C:\Users\dudey\.ssh\weather\_extract\_airflow\_keypair.pem" # make sure key is stored in ssh folder

User ubuntu

4. open git bash and type this command chmod 400 weather\_extract\_airflow\_keypair.pem

5. Now Connect to host and select your instance name.

------------------------------------------------------------------------------------------------------------------------

Ubuntu commands for installing airflow

7. Launch the ec2 instance and type

1. sudo apt update

2. sudo apt install python3-pip

3. sudo apt install python3.10-venv (for virtual env)

4. python3 -m venve movie ( to create a movie venv)

5. ls to get directories

6. source airflow venv/bin/activate ( to activate the venv)

7. sudo pip install pandas

8. sudo pip install s3fs ( to connect s3 bucket)

9. sudo pip install apache-airflow

10.airflow standalone

11. Enter the creds for once.

------------------------------------------------------------------------------------------------------------------------

Airflow DAG and S3 Bucket Creation.

8. Create the dags folder and create weather.py

9. Import the necessary librarbies

10. Creates httpsensor and create the connection in airflow. Mention the httpsid

11. Go to ec2 instance actions, mofify iam role and create new one add s3full access and ec2full access , give a name

12. Add this role to ec2 instance.

13. Create s3 bucket

14. sudo pip install awscii

15. sudo pip install s3fs

16. copy aws credentials key and store csv

17. aws configure

18. add the key and pass, ignore region and hit enter

19. aws get-session-token

20. Update the values of key , secret key and token in the code

21. Now run the dag.

----------------------------------------------------------------------------------------------------------------------

To load S3 csv file to AWS RDS(PostgresSQL DB)

Create a RDS Data Base:

1. Open the aws rds window

2. creat a db with a name (weatherinfo)

3. Select postgres as db engine

4. creat the master username and password and store the info for logging into db

5. Select default db instance class - db.t3.micro free tier

6. Enable public access to Yes

7. Create new VPC security group - add a name ( aws\_rds\_weather) and create the db

8. Click on security group and inbound rules, make sure the port is 0.0.0.0/0 to accept the data from anywhere

A screenshot of a computer

Description automatically generated

9. Install the postgresql to ec2 instance (<https://www.postgresql.org/download/linux/ubuntu/>)

10. # Create the file repository configuration: run this cmd in ec2 instance

sudo sh -c 'echo "deb https://apt.postgresql.org/pub/repos/apt $(lsb\_release -cs)-pgdg main" > /etc/apt/sources.list.d/pgdg.list'

11. # Import the repository signing key:

wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo apt-key add –

12. sudo apt-get update

13. sudo apt-get -y install postgresql

To connect ec2 install to the rds db which we have created.

<https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_PostgreSQL.S3Import.html>

1. Get the host from rds instance in aws rds which is : weatherinfo.cemo9n5kovnk.us-west-2.rds.amazonaws.com
2. psql -h weatherinfo.cemo9n5kovnk.us-west-2.rds.amazonaws.com -p 5432 -U postgres -W
3. enter the password
4. \list for viewing the dbs in postgres
5. CREATE EXTENSION aws\_s3 CASCADE; (this is an extension used to transfer s3 data to postgres)
6. \q to come out of postgres

Create the import access to s3 for rds

<https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_PostgreSQL.S3Import.html>

Creating a policy for giving s3 bucket import access to rds.

Weatherbucketdataeng – create a s3 bucket with this name

aws iam create-policy \

--policy-name postgresS3Policy \

--policy-document '{"Version": "2012-10-17", "Statement": [{"Sid": "s3import", "Action": ["s3:GetObject", "s3:ListBucket"], "Effect": "Allow", "Resource": ["arn:aws:s3::: weatherbucketdataeng ", "arn:aws:s3::: weatherbucketdataeng /\*"]}]}'

8.

Create a role for s3

aws iam create-role \

--role-name postgresql-S3-Role \

--assume-role-policy-document '{"Version": "2012-10-17", "Statement": [{"Effect": "Allow", "Principal": {"Service": "rds.amazonaws.com"}, "Action": "sts:AssumeRole"}]}'

9.

Attacch role to the policy.

Get the arn from the policy created in iam

A screenshot of a computer

Description automatically generated

aws iam attach-role-policy \

--policy- arn:aws:iam::129104151580:policy/postgresS3Policy \

--role-name postgresql-S3-Role

Check the rds instance > manage iam roles, u can see the role attached to the rds.

A screenshot of a computer

Description automatically generated

10.

Get arn from the role created to add the role to dbinstance – weatherinfo

A screenshot of a computer

Description automatically generated

aws rds add-role-to-db-instance \

--db-instance-identifier weatherinfo \

--feature-name s3Import \

--role-arn arn:aws:iam::129104151580:role/postgresql-S3-Role \

--region us-west-2