How to create the YouTube API?

- Go to https://console.cloud.google.com/
- Sign into your Google Account
- Create Project
- Go to Library -> select the required YouTube API from the list (we are using YouTube Data API v3)
- Enable API
- Create API key -> select Credentials > Create Credentials > API Key

YouTube Documentation:

https://developers.google.com/youtube/v3

Prerequisites:

Python 2.7 or Python 3.5+

The pip package management tool

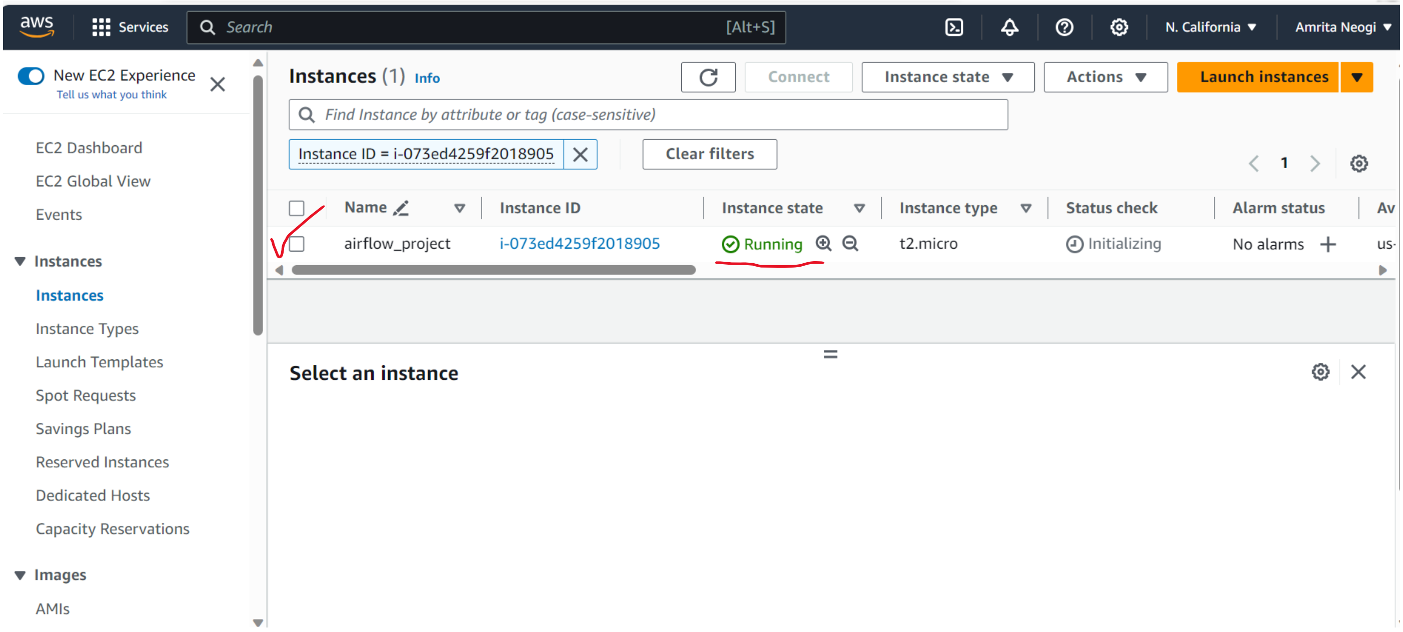
The Google APIs Client Library for Python:

pip install --upgrade google-api-python-client

- Execute the Python script youtube_etl.py to extract the YouTube data.

- Create EC2 instant that is the online machine and deploy Airflow on that.

  - Login to AWS account
  - Select EC2 > Instances > Launch Instances > provide Name and tags (e.g. airflow_project)
  - Choose Application and OS Image → Ubuntu
  - Choose Instance Type → t3_medium (additional charge required, else go with t2_micro it is free)
  - Create key-pair to access the EC2 instance:
    airflow_ec2_key →key will be downloaded → keep the key in the same folder
  - Allow SSH, HTTP, HTTPS traffic from internet
  - Launch Instance

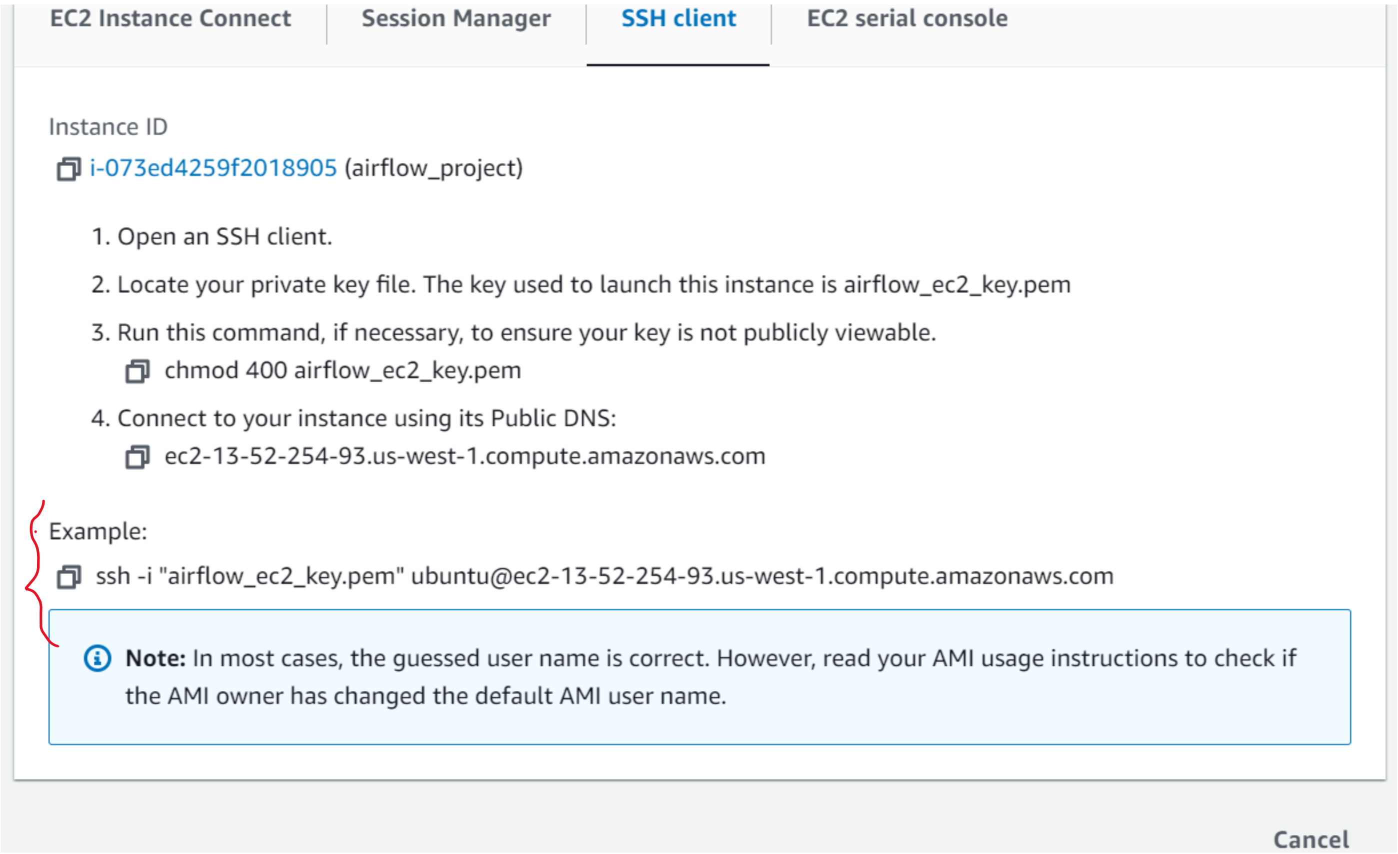


Instance ID = i-073ed4259f2018905

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Av
airflow_project	i-073ed4259f2018905	Running	t2.micro	Initializing	No alarms	us-

- Connect to Airflow:

- Select the Instance > Click on ‘Connect’ > SSH client > copy the autogenerated example



Instance ID

i-073ed4259f2018905 (airflow_project)

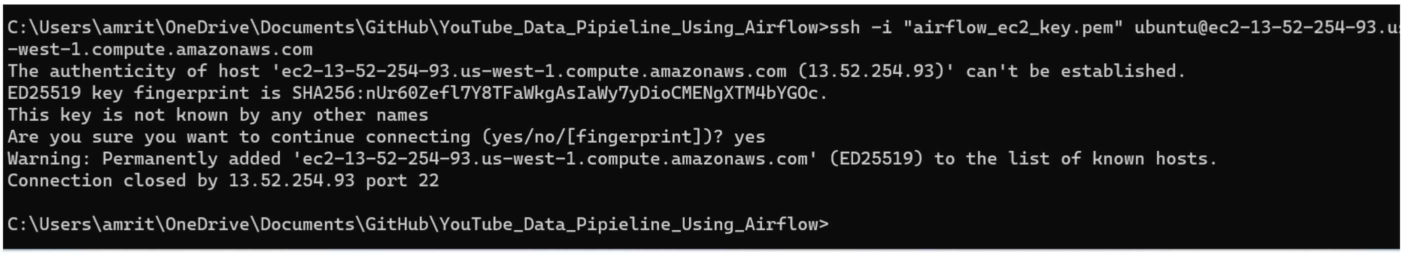
1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is airflow_ec2_key.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
   chmod 400 airflow_ec2_key.pem
4. Connect to your instance using its Public DNS:
   ec2-13-52-254-93.us-west-1.compute.amazonaws.com

Example:

ssh -i "airflow_ec2_key.pem" ubuntu@ec2-13-52-254-93.us-west-1.compute.amazonaws.com

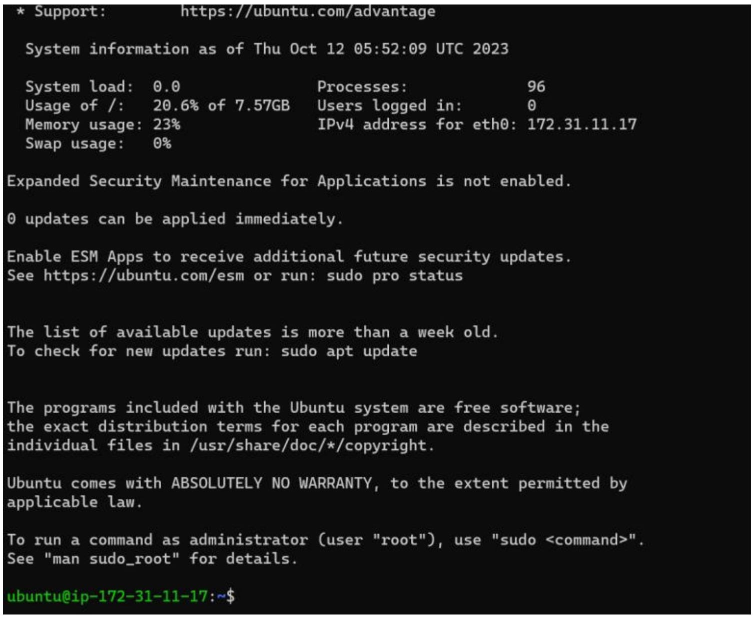
Note: In most cases, the guessed user name is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.

- Open cmd and past the key, make sure to be in the same folder



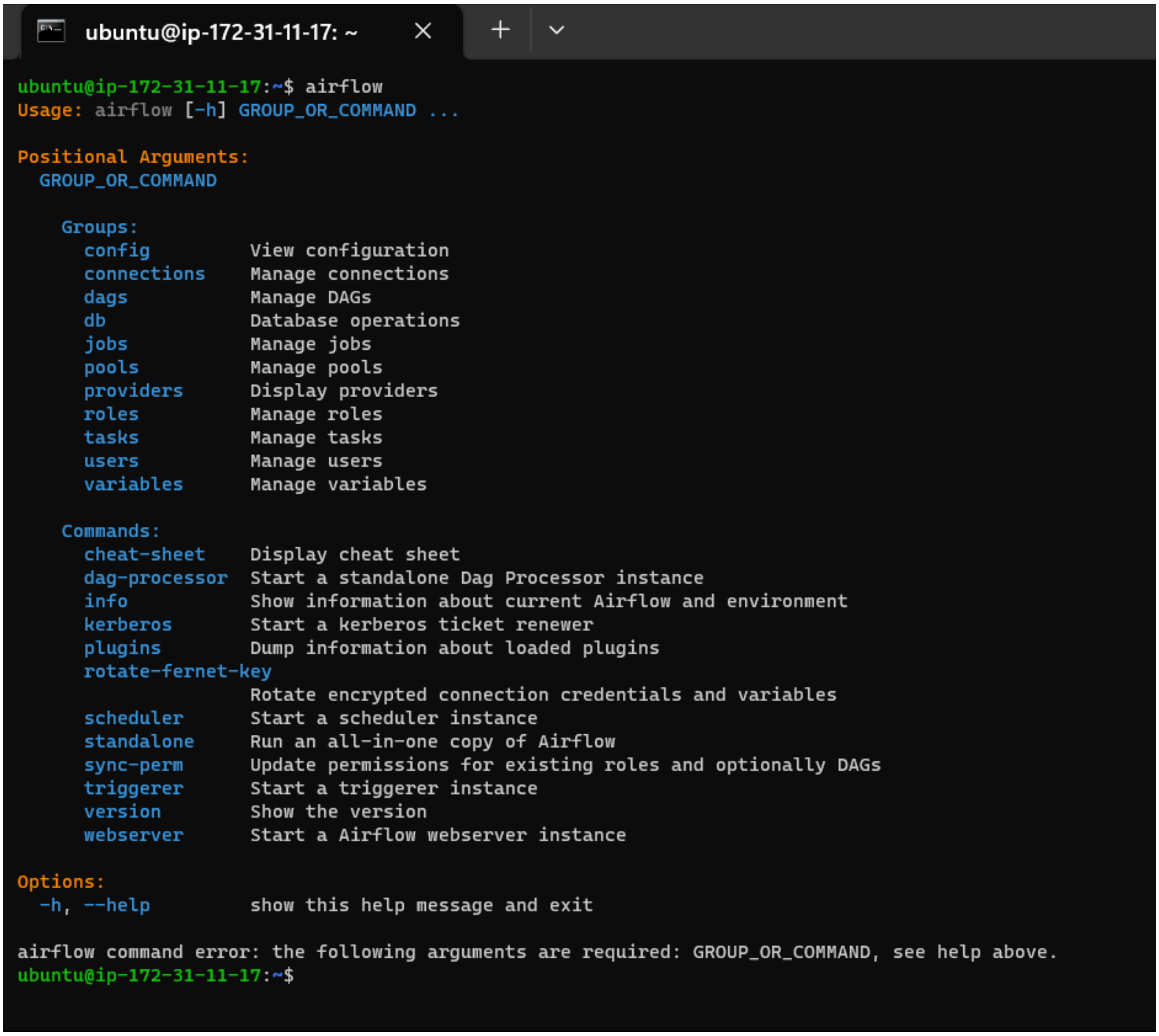
```
C:\Users\amrit\OneDrive\Documents\GitHub\YouTube_Data_Pipieline_Using_Airflow>ssh -i "airflow_ec2_key.pem" ubuntu@ec2-13-52-254-93.u
-west-1.compute.amazonaws.com
The authenticity of host 'ec2-13-52-254-93.us-west-1.compute.amazonaws.com (13.52.254.93)' can't be established.
ED25519 key fingerprint is SHA256:nUr60Zefl7Y8TFaWkgAsIaWy7yDioCMENgXTM4bYGOc.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-13-52-254-93.us-west-1.compute.amazonaws.com' (ED25519) to the list of known hosts.
Connection closed by 13.52.254.93 port 22

C:\Users\amrit\OneDrive\Documents\GitHub\YouTube_Data_Pipieline_Using_Airflow>
```

Ubuntu console will open: (run command twice if the ubuntu console is not up the first time)



```
 * Support:        https://ubuntu.com/advantage

  System information as of Thu Oct 12 05:52:09 UTC 2023

  System load:  0.0               Processes:             96
  Usage of /:   20.6% of 7.57GB   Users logged in:       0
  Memory usage: 23%               IPv4 address for eth0: 172.31.11.17
  Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The list of available updates is more than a week old.
To check for new updates run: sudo apt update


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-11-17:~$
```

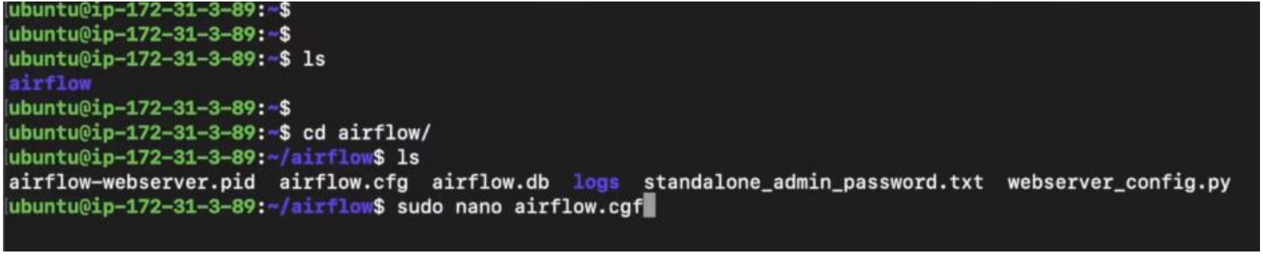
- Install the following SSH commands in the ubuntu console:
  sudo apt-get update
  sudo apt install python3-pip
  sudo pip install apache-airflow
  sudo pip install pandas
  sudo pip install s3fs
  sudo pip install tweepy→ for twitter data
  sudo pip install --upgrade google-api-python-client → for YouTube
- Check if everything is installed properly write 'airflow' in the ubuntu console:



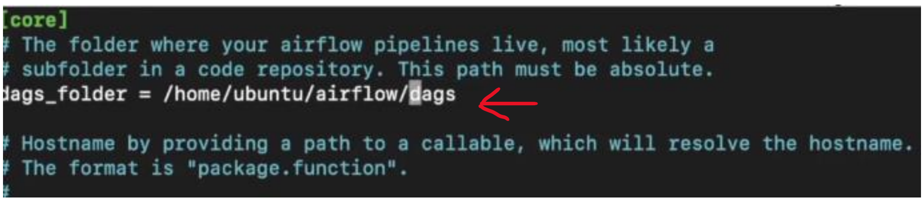
```
ubuntu@ip-172-31-11-17:~$ airflow
Usage: airflow [-h] GROUP_OR_COMMAND ...

Positional Arguments:
  GROUP_OR_COMMAND

    Groups:
      config         View configuration
      connections    Manage connections
      dags           Manage DAGs
      db             Database operations
      jobs           Manage jobs
      pools          Manage pools
      providers      Display providers
      roles          Manage roles
      tasks          Manage tasks
      users          Manage users
      variables      Manage variables

    Commands:
      cheat-sheet    Display cheat sheet
      dag-processor  Start a standalone Dag Processor instance
      info           Show information about current Airflow and environment
      kerberos       Start a kerberos ticket renewer
      plugins        Dump information about loaded plugins
      rotate-fernet-key
                     Rotate encrypted connection credentials and variables
      scheduler      Start a scheduler instance
      standalone     Run an all-in-one copy of Airflow
      sync-perm      Update permissions for existing roles and optionally DAGs
      triggerer      Start a triggerer instance
      version        Show the version
      webserver      Start a Airflow webserver instance

Options:
  -h, --help         show this help message and exit

airflow command error: the following arguments are required: GROUP_OR_COMMAND, see help above.
ubuntu@ip-172-31-11-17:~$
```

- Connect to the airflow server:
  airflow standalone

➢ Create a 'youTube_dag.py' file (refer file for the code)→ import the ETL function from 'youTube_etl.py' file
➢ Create a S3 bucket (S3 buckets | S3 | Global (amazon.com))
➢ Save the file we created in the S3 bucket in the 'youTube_etl.py' file
  e.g., channel_stat.to_csv('s3://amrita-neogi-yt-bucket/youtube_stat.csv')
➢ Connect to the airflow again →make changes to 'airflow.cfg'



```
ubuntu@ip-172-31-3-89:~$
ubuntu@ip-172-31-3-89:~$
ubuntu@ip-172-31-3-89:~$ ls
airflow
ubuntu@ip-172-31-3-89:~$
ubuntu@ip-172-31-3-89:~$ cd airflow/
ubuntu@ip-172-31-3-89:~/airflow$ ls
airflow-webserver.pid  airflow.cfg  airflow.db  logs  standalone_admin_password.txt  webserver_config.py
ubuntu@ip-172-31-3-89:~/airflow$ sudo nano airflow.cgf
```



```
[core]
# The folder where your airflow pipelines live, most likely a
# subfolder in a code repository. This path must be absolute.
dags_folder = /home/ubuntu/airflow/dags

# Hostname by providing a path to a callable, which will resolve the hostname.
# The format is "package.function".
#
```

Change 'dags' to 'youTube_dag' or anyother preferred name

- ➢ Create a new folder inside S3 copy the etl.py code from local folder to the EC2 machine
  - mkdir youTube_dag
  - cd youTube_dag
  - sudo nano youTube_dag.py
  - copy code from 'youTube_dag.py' and paste here
  - ctrl + X to save the file
  - repeate same process for etl file

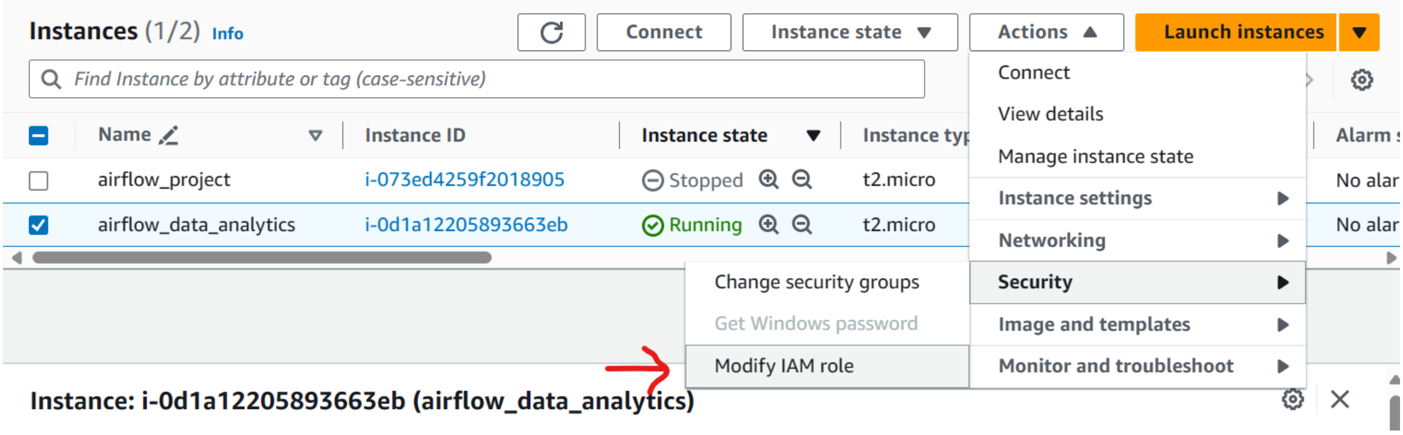
    sudo nano youTube_etl.py

    copy code from 'youTube_etl.py' and paste here

    ctrl + X to save the file

** to stop airflow server → ctrl + C

- ➢ Make sure to have permission from the EC2 to write on the S3 bucket.



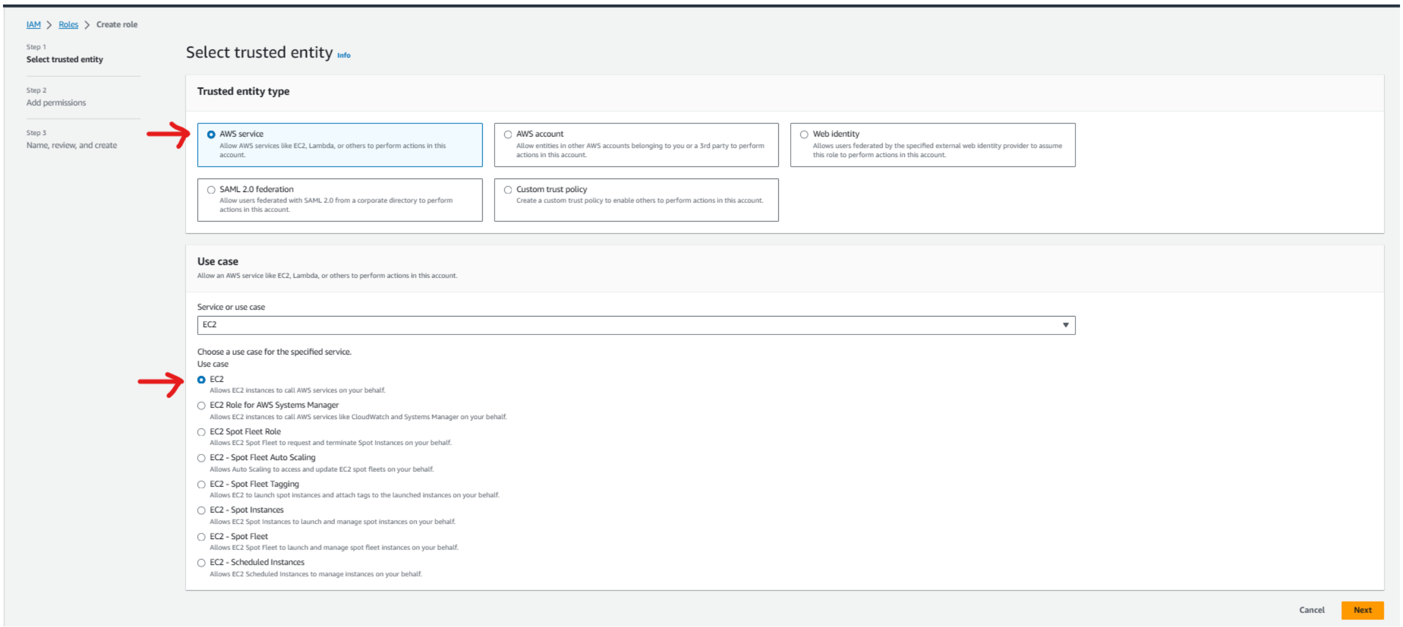
Instances (1/2) Info

Find Instance by attribute or tag (case-sensitive)

	Name	Instance ID	Instance state	Instance typ	Alarm s
□	airflow_project	i-073ed4259f2018905	⊖ Stopped	t2.micro	No alar
☑	airflow_data_analytics	i-0d1a12205893663eb	✓ Running	t2.micro	No alar

Instance: i-0d1a12205893663eb (airflow_data_analytics)

- Create IAM role



IAM > Roles > Create role

Step 1
Select trusted entity

Step 2
Add permissions

Step 3
Name, review, and create

Select trusted entity Info

Trusted entity type

● AWS service
Allow AWS services like EC2, Lambda, or others to perform actions in this account.

○ AWS account
Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.

○ Web identity
Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.

○ SAML 2.0 federation
Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.

○ Custom trust policy
Create a custom trust policy to enable others to perform actions in this account.

Use case
Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

Service or use case
EC2 ▼

Choose a use case for the specified service.
Use case

● EC2
Allows EC2 instances to call AWS services on your behalf.

○ EC2 Role for AWS Systems Manager
Allows EC2 instances to call AWS services like CloudWatch and Systems Manager on your behalf.

○ EC2 Spot Fleet Role
Allows EC2 Spot Fleet to request and terminate Spot Instances on your behalf.

○ EC2 - Spot Fleet Auto Scaling
Allows Auto Scaling to access and update EC2 spot fleets on your behalf.

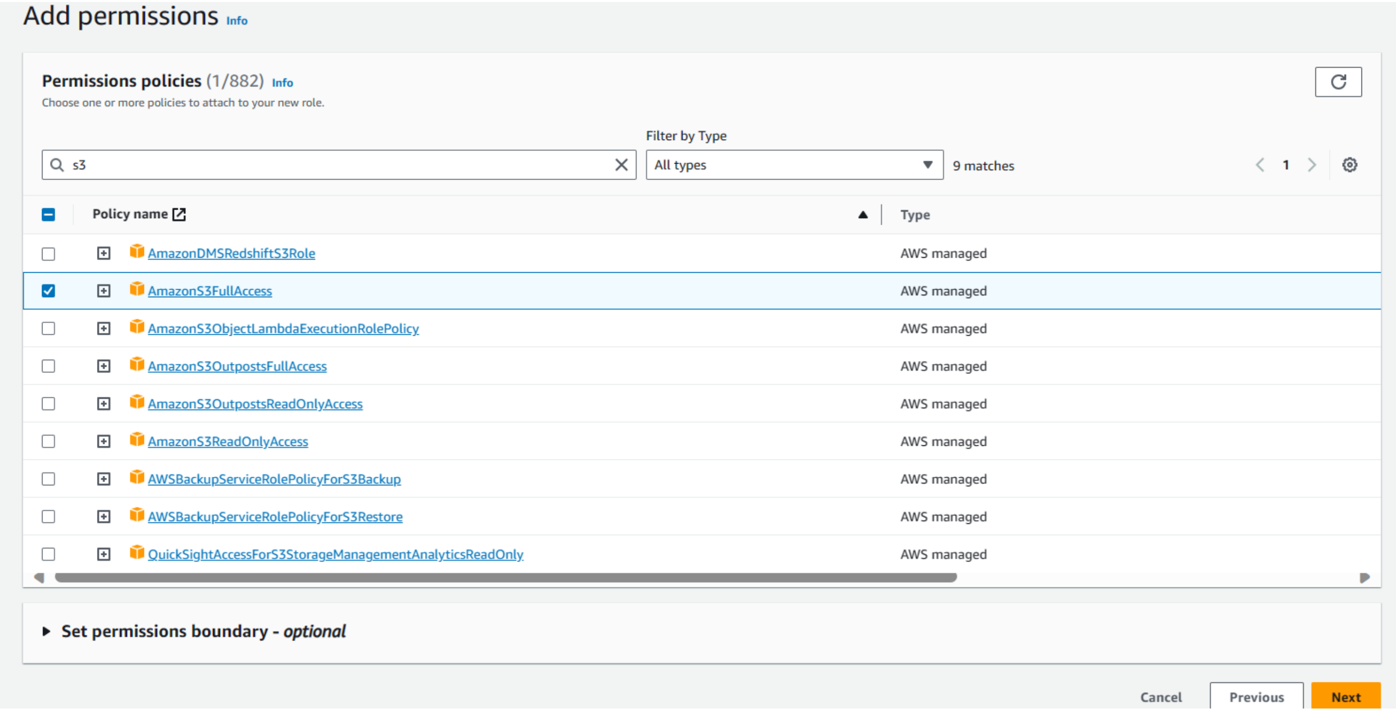
○ EC2 - Spot Fleet Tagging
Allows EC2 to launch spot instances and attach tags to the launched instances on your behalf.

○ EC2 - Spot Instances
Allows EC2 Spot Instances to launch and manage spot instances on your behalf.

○ EC2 - Spot Fleet
Allows EC2 Spot Fleet to launch and manage spot fleet instances on your behalf.

○ EC2 - Scheduled Instances
Allows EC2 Scheduled Instances to manage instances on your behalf.

Give S3 and EC2 full access:



Add permissions Info

Permissions policies (1/882) Info

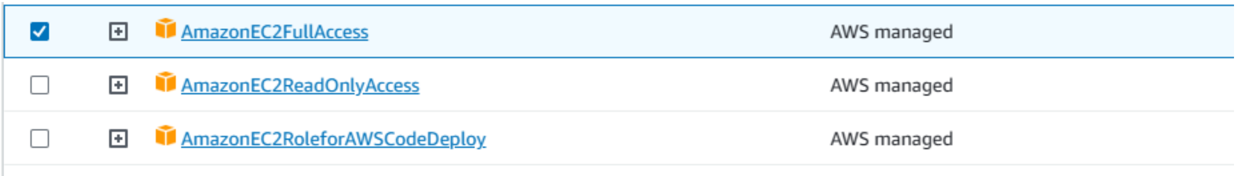
Choose one or more policies to attach to your new role.

Filter by Type

s3 | All types | 9 matches

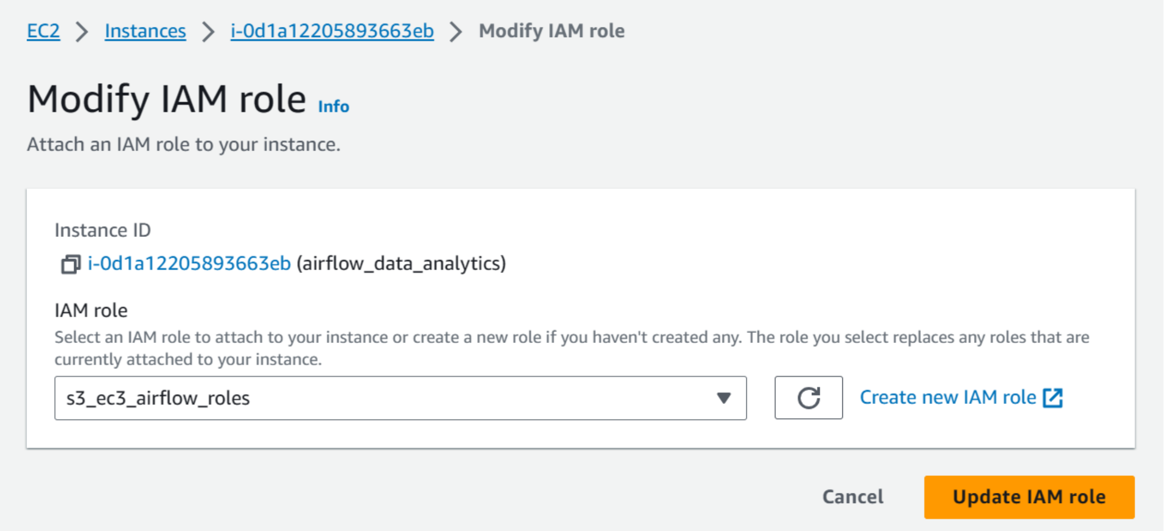
	Policy name	Type
□	AmazonDMSRedshiftS3Role	AWS managed
☑	AmazonS3FullAccess	AWS managed
□	AmazonS3ObjectLambdaExecutionRolePolicy	AWS managed
□	AmazonS3OutpostsFullAccess	AWS managed
□	AmazonS3OutpostsReadOnlyAccess	AWS managed
□	AmazonS3ReadOnlyAccess	AWS managed
□	AWSBackupServiceRolePolicyForS3Backup	AWS managed
□	AWSBackupServiceRolePolicyForS3Restore	AWS managed
□	QuickSightAccessForS3StorageManagementAnalyticsReadOnly	AWS managed

▸ **Set permissions boundary - *optional***



☑	AmazonEC2FullAccess	AWS managed
□	AmazonEC2ReadOnlyAccess	AWS managed
□	AmazonEC2RoleforAWSCodeDeploy	AWS managed

- Create role – ‘s3_ec2_airflow_role’
- Update IAM role



EC2 > Instances > i-0d1a12205893663eb > Modify IAM role

Modify IAM role Info

Attach an IAM role to your instance.

Instance ID

i-0d1a12205893663eb (airflow_data_analytics)

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.

s3_ec3_airflow_roles