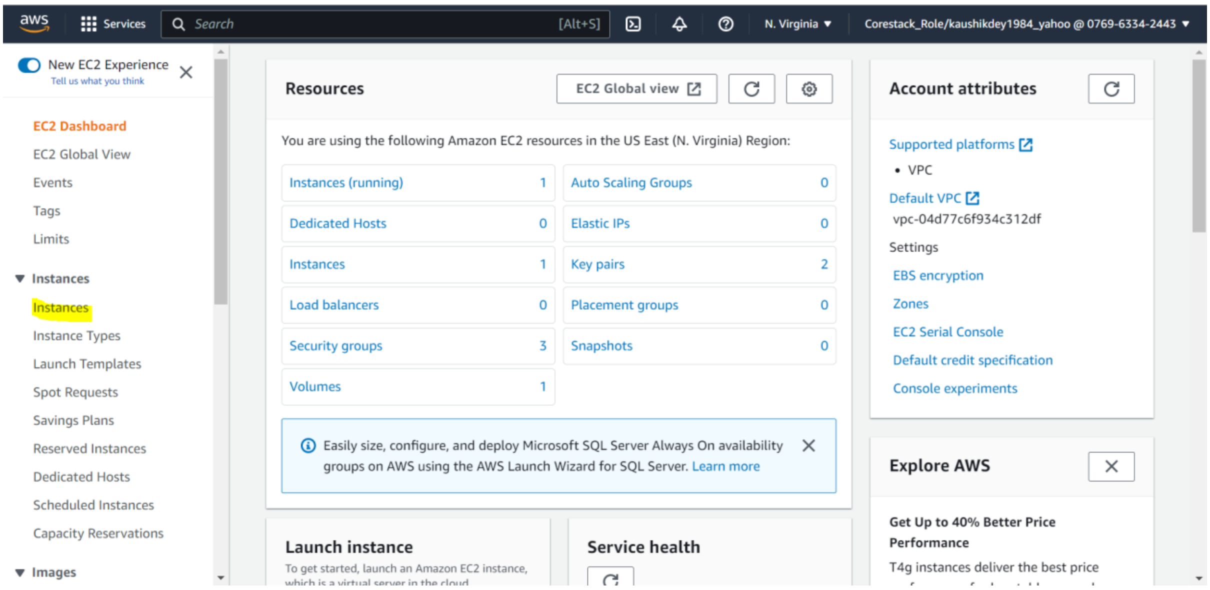
Twitter Data Pipeline using Airflow

Kaushik Dey
Kaushikdey1984@yahoo.com

1. Up & Running Ubuntu server via EC2 Instance and install the Airflow, pandas, s3fs.

Step 1

First, we must create one ubuntu instance. So, we must take ec2 service from Aws. So, we must follow some steps.



You are using the following Amazon EC2 resources in the US East (N. Virginia) Region:

Instances (running)	1	Auto Scaling Groups	0
Dedicated Hosts	0	Elastic IPs	0
Instances	1	Key pairs	2
Load balancers	0	Placement groups	0
Security groups	3	Snapshots	0
Volumes	1		

Easily size, configure, and deploy Microsoft SQL Server Always On availability groups on AWS using the AWS Launch Wizard for SQL Server. Learn more

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

Supported platforms
- VPC

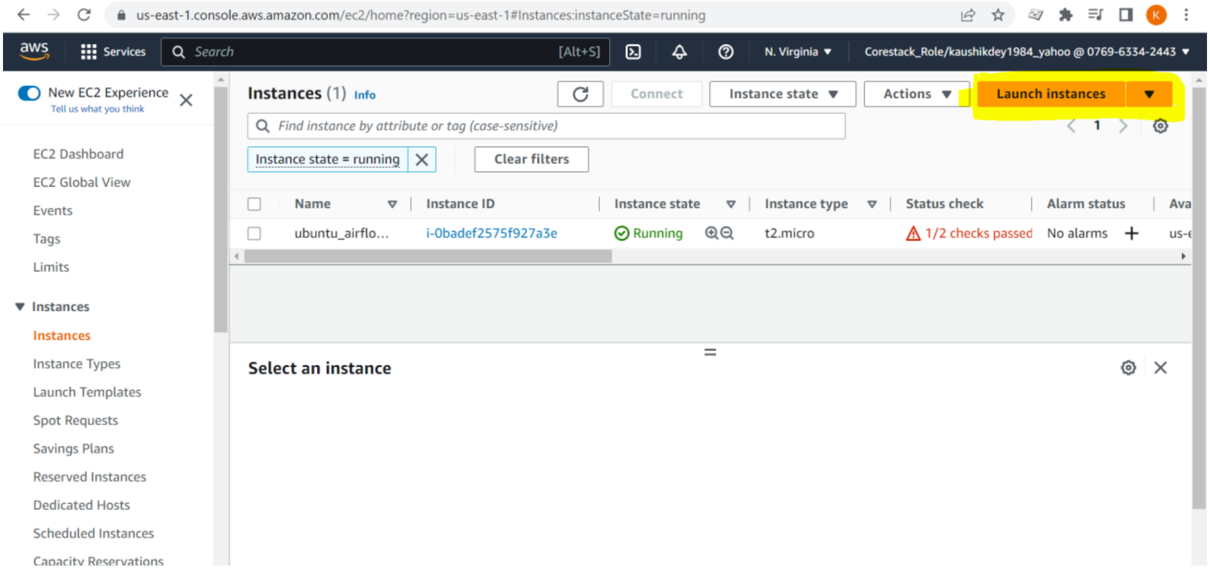
Default VPC
vpc-04d77c6f934c312df

Settings
EBS encryption
Zones
EC2 Serial Console
Default credit specification
Console experiments

Get Up to 40% Better Price Performance

T4g instances deliver the best price

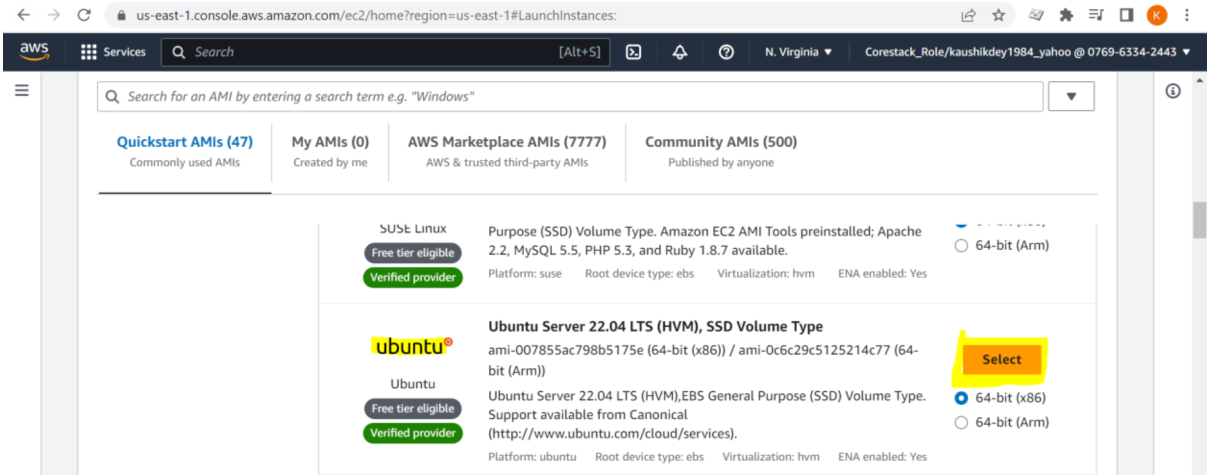
Step 2



□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Ava
□	ubuntu_airflo...	i-0badef2575f927a3e	Running	t2.micro	1/2 checks passed	No alarms	us-e

Step 3

Now we must Browse AMI from the catalogue. The screenshot is given below. (Ubuntu Server 22.04 LTS)



us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LaunchInstances:

Search for an AMI by entering a search term e.g. "Windows"

Quickstart AMIs (47) Commonly used AMIs | My AMIs (0) Created by me | AWS Marketplace AMIs (7777) AWS & trusted third-party AMIs | Community AMIs (500) Published by anyone

SUSE Linux
Free tier eligible
Verified provider

Purpose (SSD) Volume Type. Amazon EC2 AMI Tools preinstalled; Apache 2.2, MySQL 5.5, PHP 5.3, and Ruby 1.8.7 available.

Platform: suse Root device type: ebs Virtualization: hvm ENA enabled: Yes

○ 64-bit (Arm)

ubuntu
Ubuntu
Free tier eligible
Verified provider

Ubuntu Server 22.04 LTS (HVM), SSD Volume Type

ami-007855ac798b5175e (64-bit (x86)) / ami-0c6c29c5125214c77 (64-bit (Arm))

Ubuntu Server 22.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).

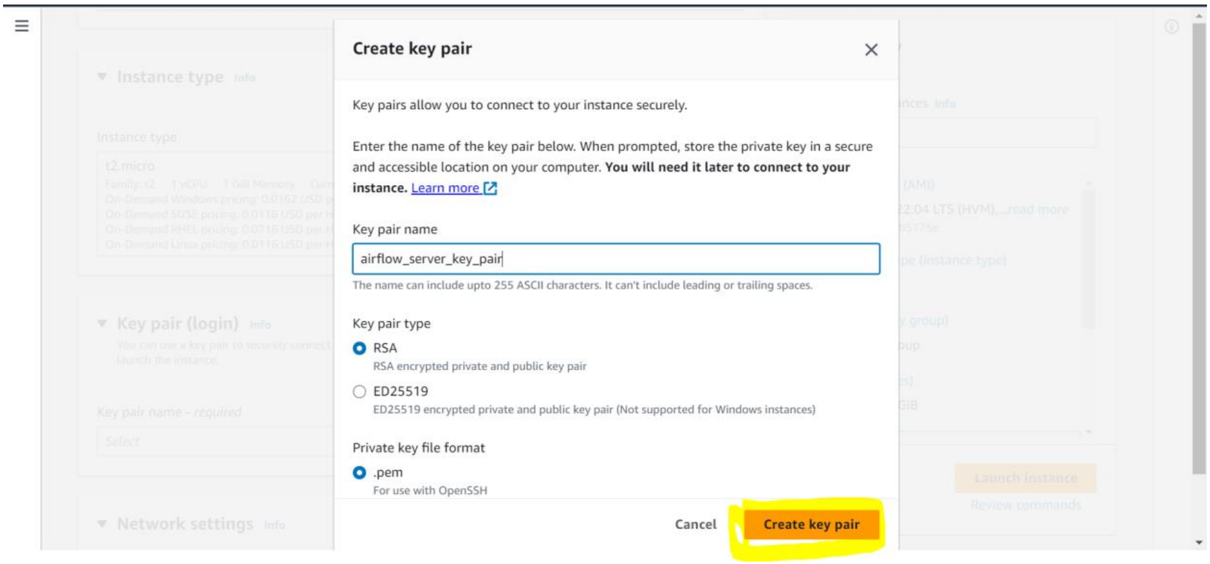
Platform: ubuntu Root device type: ebs Virtualization: hvm ENA enabled: Yes

Select

● 64-bit (x86)
○ 64-bit (Arm)

Step 4

Now we must create new key pair for this instance.



Create key pair ✕

Key pairs allow you to connect to your instance securely.

Enter the name of the key pair below. When prompted, store the private key in a secure and accessible location on your computer. **You will need it later to connect to your instance.** Learn more

Key pair name

airflow_server_key_pair

The name can include upto 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type

● RSA
RSA encrypted private and public key pair

○ ED25519
ED25519 encrypted private and public key pair (Not supported for Windows instances)

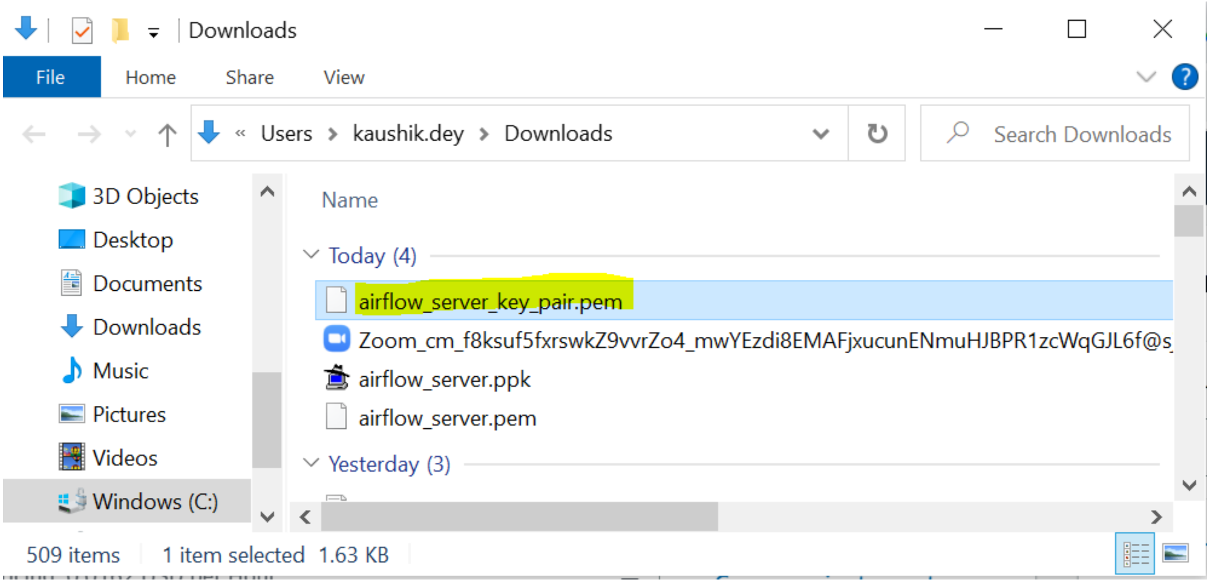
Private key file format

● .pem
For use with OpenSSH

Cancel | Create key pair

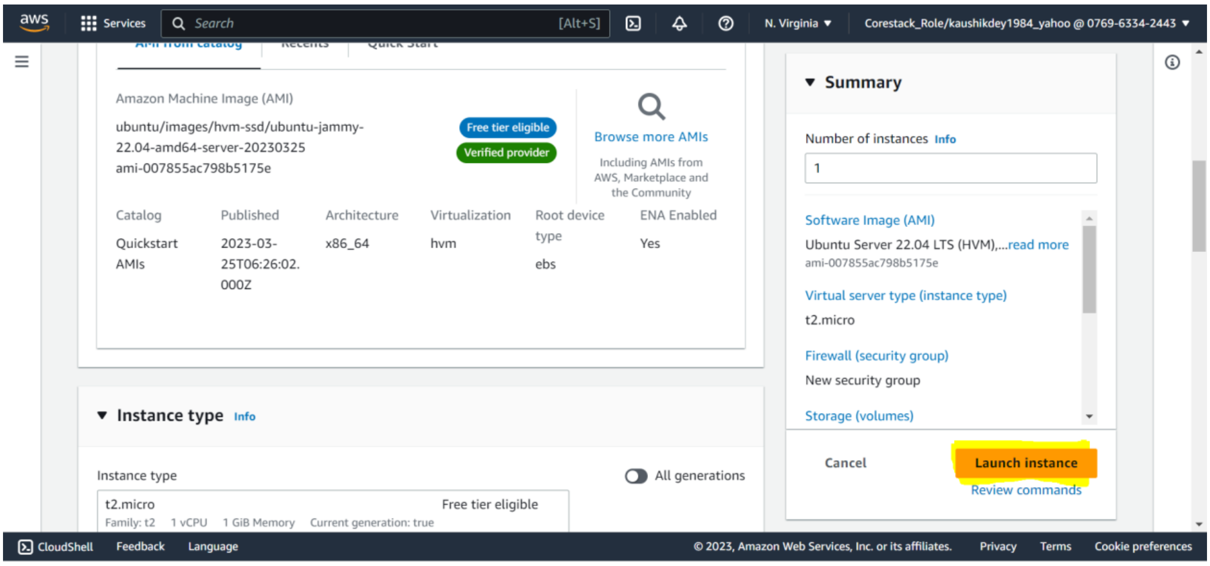
Step 5

Now we can download the keypair (airflow_server_key_pair.pem file). The rest of the settings will be as it is.



Step 6

Now we must launch our instance.



Amazon Machine Image (AMI)

ubuntu/images/hvm-ssd/ubuntu-jammy-22.04-amd64-server-20230325
ami-007855ac798b5175e

Catalog	Published	Architecture	Virtualization	Root device type	ENA Enabled
Quickstart AMIs	2023-03-25T06:26:02.000Z	x86_64	hvm	ebs	Yes

▾ Instance type Info

Instance type

t2.micro Free tier eligible
Family: t2 1 vCPU 1 GiB Memory Current generation: true

▾ Summary

Number of instances Info
1

Software Image (AMI)
Ubuntu Server 22.04 LTS (HVM),...read more
ami-007855ac798b5175e

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

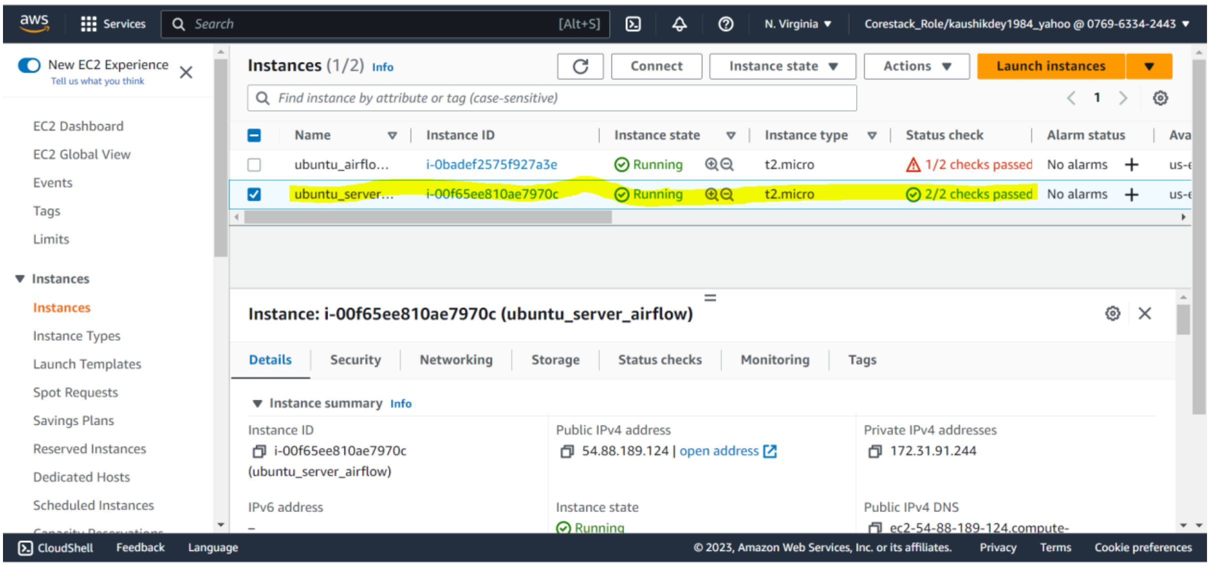
Storage (volumes)

Step 7

Now we can see that our instance is created and up and running . so we can understand it from its status running. Now couple of things we have to highlighted.

Public IP Address: ec2-54-88-189-124.compute-1.amazonaws.com

Private IP Address : 172.31.91.244



Instances (1/2) Info

Find instance by attribute or tag (case-sensitive)

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Ava
□	ubuntu_airflo…	i-0badef2575f927a3e	Running	t2.micro	1/2 checks passed	No alarms	us-e
☑	ubuntu_server…	i-00f65ee810ae7970c	Running	t2.micro	2/2 checks passed	No alarms	us-e

Instance: i-00f65ee810ae7970c (ubuntu_server_airflow)

Details | Security | Networking | Storage | Status checks | Monitoring | Tags

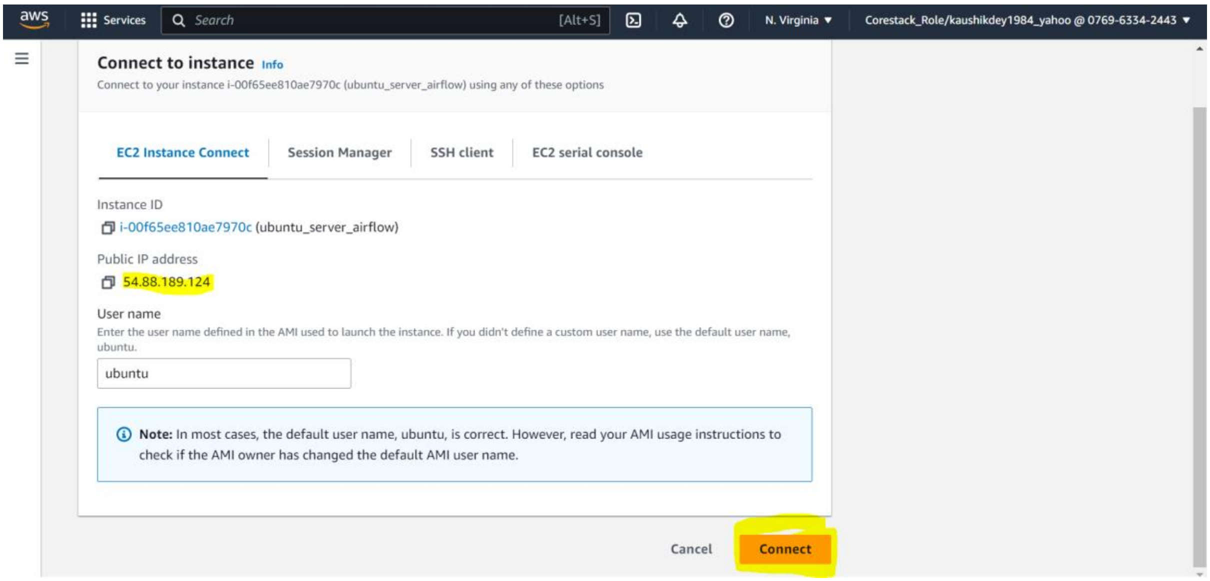
▼ Instance summary Info

Instance ID	Public IPv4 address	Private IPv4 addresses
i-00f65ee810ae7970c (ubuntu_server_airflow)	54.88.189.124 | open address	172.31.91.244
IPv6 address	**Instance state**	**Public IPv4 DNS**
–	Running	ec2-54-88-189-124.compute-

Step 8

Now we have to connect our instances via browser or via ssh client.

Browser Based Access (Not recommended)



Connect to instance Info

Connect to your instance i-00f65ee810ae7970c (ubuntu_server_airflow) using any of these options

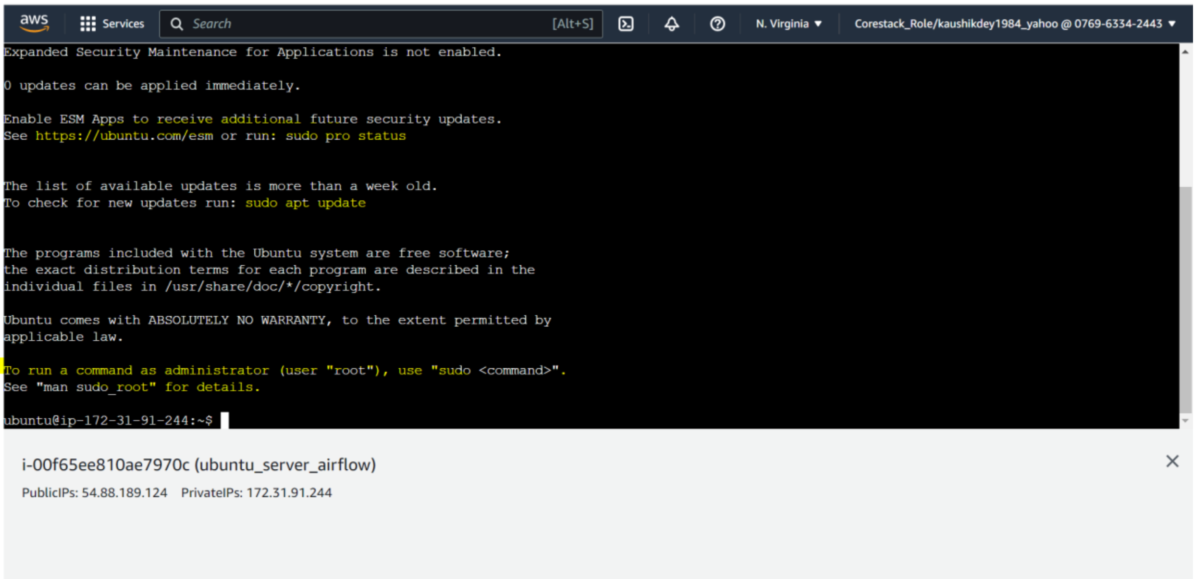
Instance ID
i-00f65ee810ae7970c (ubuntu_server_airflow)

Public IP address
54.88.189.124

User name
Enter the user name defined in the AMI used to launch the instance. If you didn't define a custom user name, use the default user name, ubuntu.

ubuntu

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



```
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-172-31-91-244:~$
```

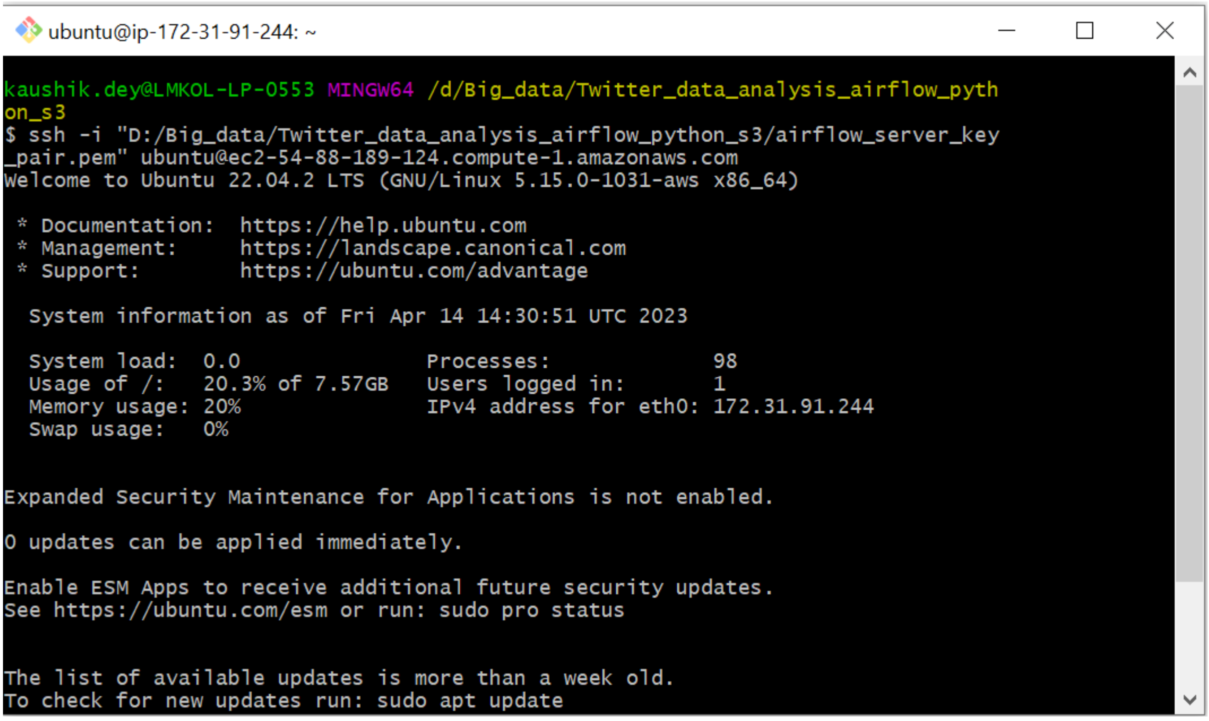
i-00f65ee810ae7970c (ubuntu_server_airflow)

PublicIPs: 54.88.189.124 PrivateIPs: 172.31.91.244

SSH Client Based Access (Recommended)

The command we must run in windows git bash shell, internally it runs Linux command. Just run this command.

ssh -i "D:/Big_data/Twitter_data_analysis_airflow_python_s3/airflow_server_key_pair.pem" ubuntu@ec2-54-88-189-124.compute-1.amazonaws.com



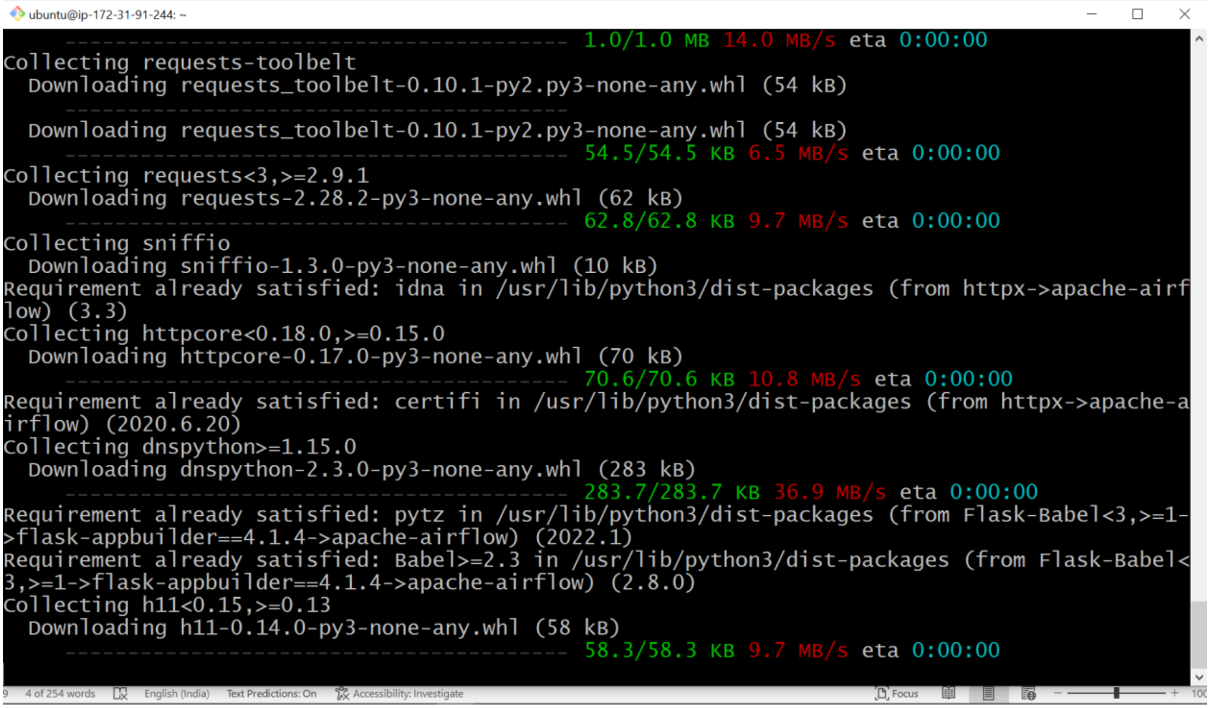
```
kaushik.dey@LMKOL-LP-0553 MINGW64 /d/Big_data/Twitter_data_analysis_airflow_pyth
on_s3
$ ssh -i "D:/Big_data/Twitter_data_analysis_airflow_python_s3/airflow_server_key
_pair.pem" ubuntu@ec2-54-88-189-124.compute-1.amazonaws.com
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-1031-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Fri Apr 14 14:30:51 UTC 2023

  System load:  0.0               Processes:             98
  Usage of /:   20.3% of 7.57GB   Users logged in:       1
  Memory usage: 20%               IPv4 address for eth0: 172.31.91.244
  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
```

Step 9

Now we must install couple of ubuntu commands to establish airflow running.

- ✓ 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 (twitter api)

Installation done following this way



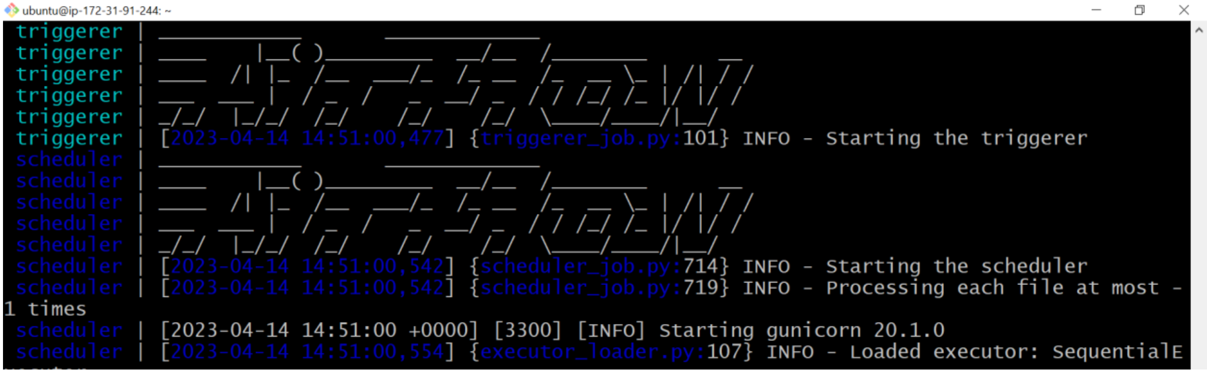
```
     ---------------------------------------- 1.0/1.0 MB 14.0 MB/s eta 0:00:00
Collecting requests-toolbelt
  Downloading requests_toolbelt-0.10.1-py2.py3-none-any.whl (54 kB)
     ----------------------------------------

  Downloading requests_toolbelt-0.10.1-py2.py3-none-any.whl (54 kB)
     ---------------------------------------- 54.5/54.5 KB 6.5 MB/s eta 0:00:00
Collecting requests<3,>=2.9.1
  Downloading requests-2.28.2-py3-none-any.whl (62 kB)
     ---------------------------------------- 62.8/62.8 KB 9.7 MB/s eta 0:00:00
Collecting sniffio
  Downloading sniffio-1.3.0-py3-none-any.whl (10 kB)
Requirement already satisfied: idna in /usr/lib/python3/dist-packages (from httpx->apache-airf
low) (3.3)
Collecting httpcore<0.18.0,>=0.15.0
  Downloading httpcore-0.17.0-py3-none-any.whl (70 kB)
     ---------------------------------------- 70.6/70.6 KB 10.8 MB/s eta 0:00:00
Requirement already satisfied: certifi in /usr/lib/python3/dist-packages (from httpx->apache-a
irflow) (2020.6.20)
Collecting dnspython>=1.15.0
  Downloading dnspython-2.3.0-py3-none-any.whl (283 kB)
     ---------------------------------------- 283.7/283.7 KB 36.9 MB/s eta 0:00:00
Requirement already satisfied: pytz in /usr/lib/python3/dist-packages (from Flask-Babel<3,>=1-
>flask-appbuilder==4.1.4->apache-airflow) (2022.1)
Requirement already satisfied: Babel>=2.3 in /usr/lib/python3/dist-packages (from Flask-Babel<
3,>=1->flask-appbuilder==4.1.4->apache-airflow) (2.8.0)
Collecting h11<0.15,>=0.13
  Downloading h11-0.14.0-py3-none-any.whl (58 kB)
     ---------------------------------------- 58.3/58.3 KB 9.7 MB/s eta 0:00:00
```

Step 10

Now we have to up & running airflow in browser from ubuntu system. So, following two commands are there.

Airflow & airflow standalone (not recommended for production, its used for development) only two things we must know.

```
Login with
username: admin
password: sfkPzwUKBNsyMbG9
```



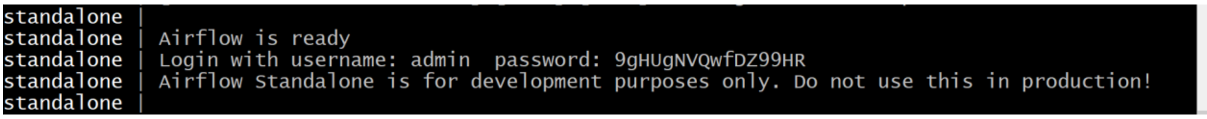
ubuntu@ip-172-31-91-244: ~

```
 triggerer | ____________       _____________
 triggerer | ____    |__( )_________  __/__  /________      __
 triggerer | ____  /| |_  /__  ___/_  /_ __  /_  __ \_ | /| / /
 triggerer | ___  ___ |  / _  /   _  __/ _  / / /_/ /_ |/ |/ /
 triggerer | _/_/  |_/_/  /_/    /_/    /_/  \____/____/|__/
 triggerer | [2023-04-14 14:51:00,477] {triggerer_job.py:101} INFO - Starting the triggerer
 scheduler | ____________       _____________
 scheduler | ____    |__( )_________  __/__  /________      __
 scheduler | ____  /| |_  /__  ___/_  /_ __  /_  __ \_ | /| / /
 scheduler | ___  ___ |  / _  /   _  __/ _  / / /_/ /_ |/ |/ /
 scheduler | _/_/  |_/_/  /_/    /_/    /_/  \____/____/|__/
 scheduler | [2023-04-14 14:51:00,542] {scheduler_job.py:714} INFO - Starting the scheduler
 scheduler | [2023-04-14 14:51:00,542] {scheduler_job.py:719} INFO - Processing each file at most -
1 times
 scheduler | [2023-04-14 14:51:00 +0000] [3300] [INFO] Starting gunicorn 20.1.0
 scheduler | [2023-04-14 14:51:00,554] {executor_loader.py:107} INFO - Loaded executor: SequentialE
```

Only one thing should follow with this message.

Step 11

Now our airflow is ready with that following screen-shots.



```
standalone |
standalone | Airflow is ready
standalone | Login with username: admin  password: 9gHUgNVQwfDZ99HR
standalone | Airflow Standalone is for development purposes only. Do not use this in production!
standalone |
```

Step 12

Now the url of this airflow ui is as following.

The port 8080 is for airflow default port.

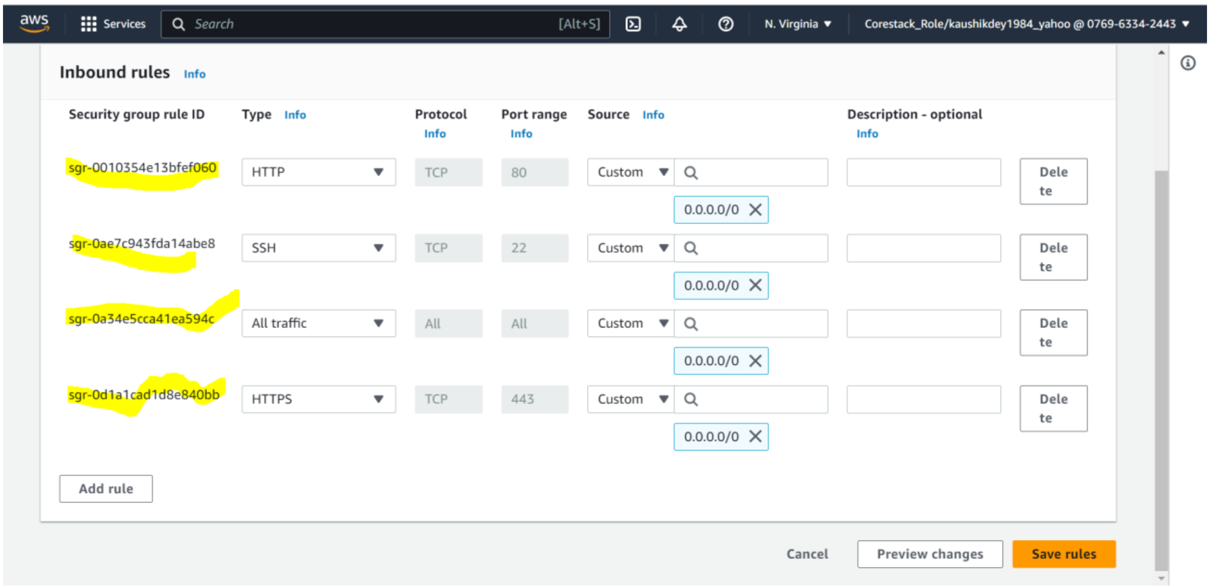
our ec2 public Ip is 54-88-189-124.

Public ipv4 DNS is http://ec2-54-88-189-124.compute-1.amazonaws.com

Airflow url is http://ec2-54-88-189-124.compute-1.amazonaws.com:8080/

Step 13

Now we have to open the port from security group. We have to all traffic , ssh, http and https , the following screenshot is given below.



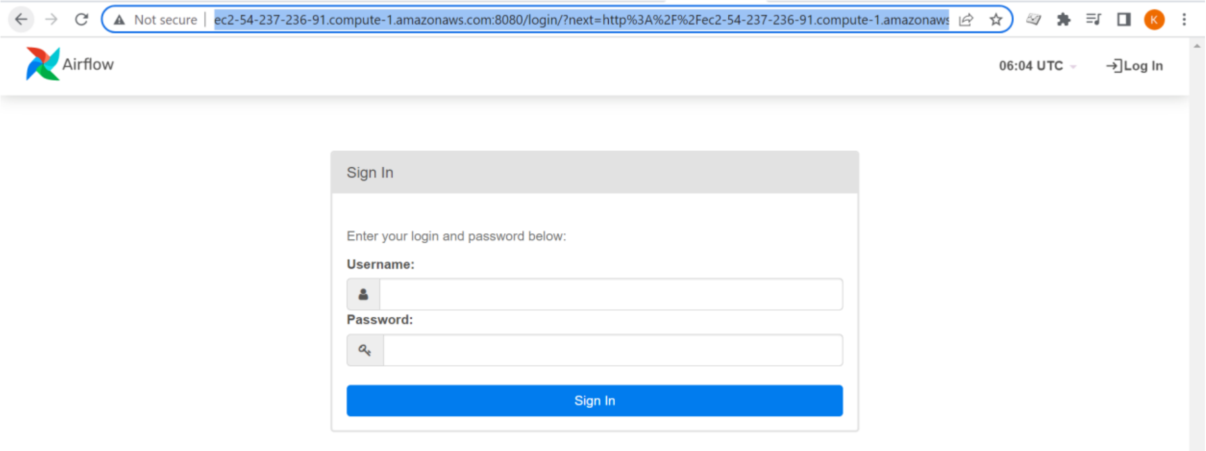
Inbound rules Info

Security group rule ID	Type	Protocol	Port range	Source		Description - optional
sgr-0010354e13bfef060	HTTP	TCP	80	Custom	0.0.0.0/0	
sgr-0ae7c943fda14abe8	SSH	TCP	22	Custom	0.0.0.0/0	
sgr-0a34e5cca41ea594c	All traffic	All	All	Custom	0.0.0.0/0	
sgr-0d1a1cad1d8e840bb	HTTPS	TCP	443	Custom	0.0.0.0/0	

Step 13

Airflow login

URL : http://ec2-54-237-236-91.compute-1.amazonaws.com:8080/login/?next=http%3A%2F%2Fec2-54-237-236-91.compute-1.amazonaws.com%3A8080%2Fhome



Not secure
ec2-54-237-236-91.compute-1.amazonaws.com:8080/login/?next=http%3A%2F%2Fec2-54-237-236-91.compute-1.amazonaws
Airflow
06:04 UTC
Log In

Sign In

Enter your login and password below:

Username:

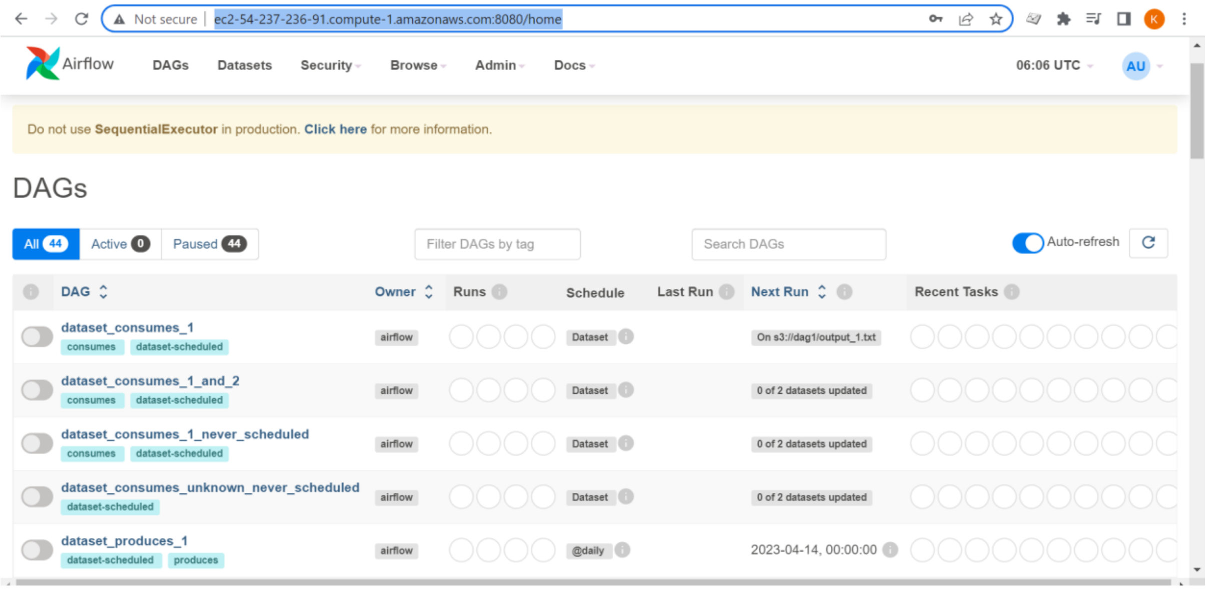
Password:

Username: admin

Password: 98gKZFDKxSAaHnC7

Step 14

DAG LINK : http://ec2-54-237-236-91.compute-1.amazonaws.com:8080/home



Not secure | ec2-54-237-236-91.compute-1.amazonaws.com:8080/home

Airflow DAGs Datasets Security Browse Admin Docs 06:06 UTC AU

Do not use **SequentialExecutor** in production. **Click here** for more information.

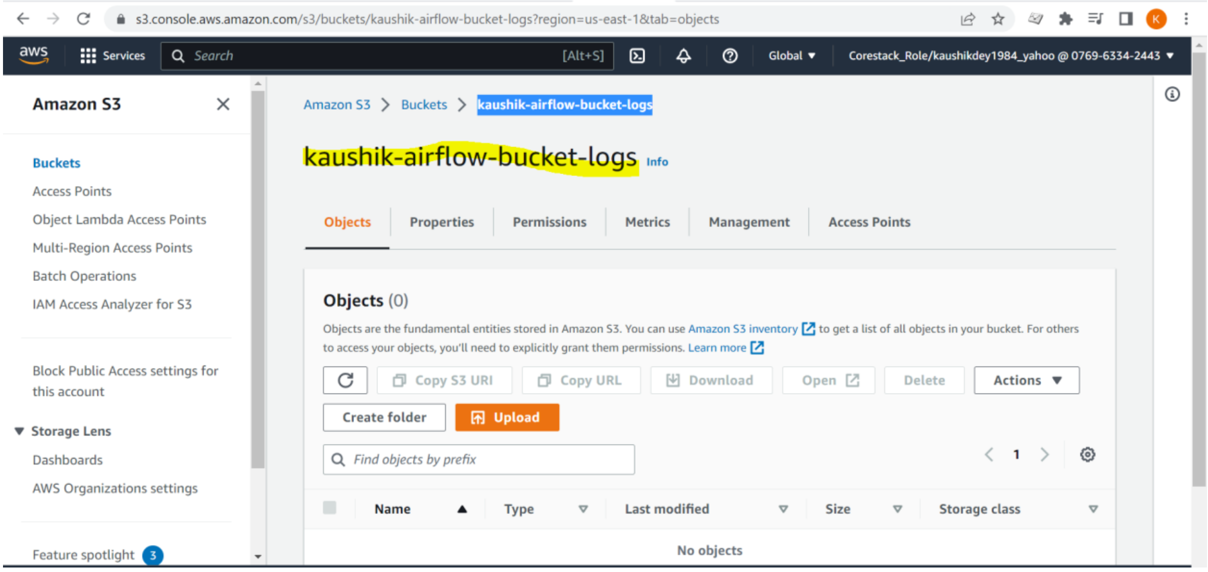
DAGs

All 44 | Active 0 | Paused 44 | Filter DAGs by tag | Search DAGs | Auto-refresh

DAG	Owner	Runs	Schedule	Last Run	Next Run	Recent Tasks
dataset_consumes_1 consumes dataset-scheduled	airflow		Dataset		On s3://dag1/output_1.txt	
dataset_consumes_1_and_2 consumes dataset-scheduled	airflow		Dataset		0 of 2 datasets updated	
dataset_consumes_1_never_scheduled consumes dataset-scheduled	airflow		Dataset		0 of 2 datasets updated	
dataset_consumes_unknown_never_scheduled dataset-scheduled	airflow		Dataset		0 of 2 datasets updated	
dataset_produces_1 dataset-scheduled produces	airflow		@daily		2023-04-14, 00:00:00	

Step 15

S3 bucket creation



s3.console.aws.amazon.com/s3/buckets/kaushik-airflow-bucket-logs?region=us-east-1&tab=objects

aws
Services
Search [Alt+S]
Global
Corestack_Role/kaushikdey1984_yahoo @ 0769-6334-2443
Amazon S3
Buckets
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
IAM Access Analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
AWS Organizations settings
Feature spotlight 3

Amazon S3 > Buckets > kaushik-airflow-bucket-logs

kaushik-airflow-bucket-logs Info

Objects | Properties | Permissions | Metrics | Management | Access Points

Objects (0)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

Name	Type	Last modified	Size	Storage class

No objects

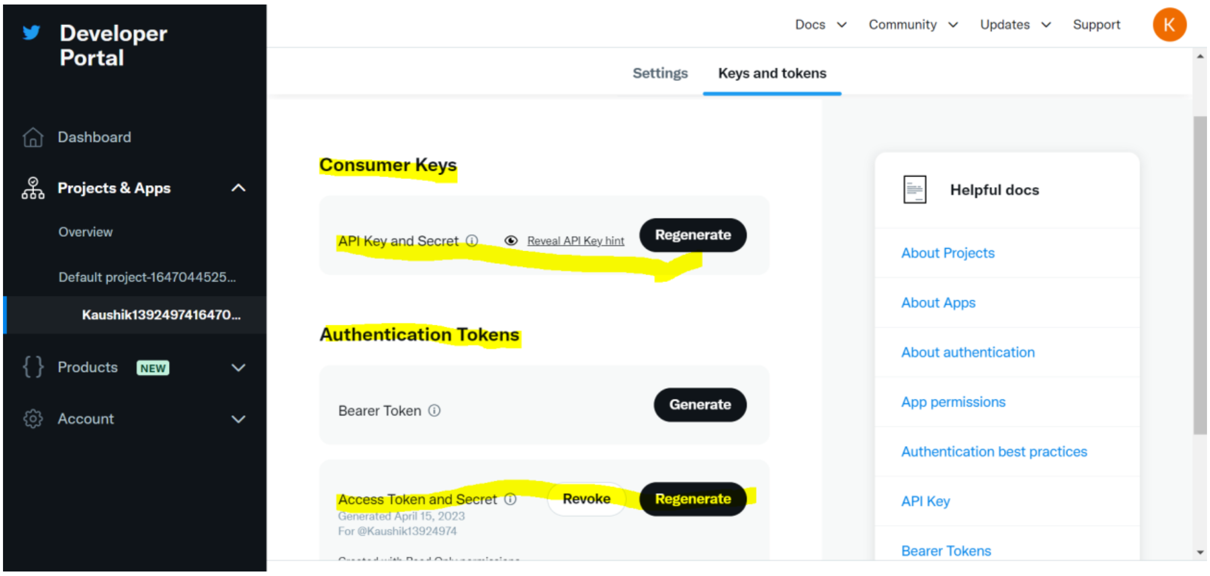
Step 16

Now we have to login in twitter account.

URL
https://developer.twitter.com/en/portal/projects/1647044527285477377/apps/26931143/keys

you must generate access key & secret key (

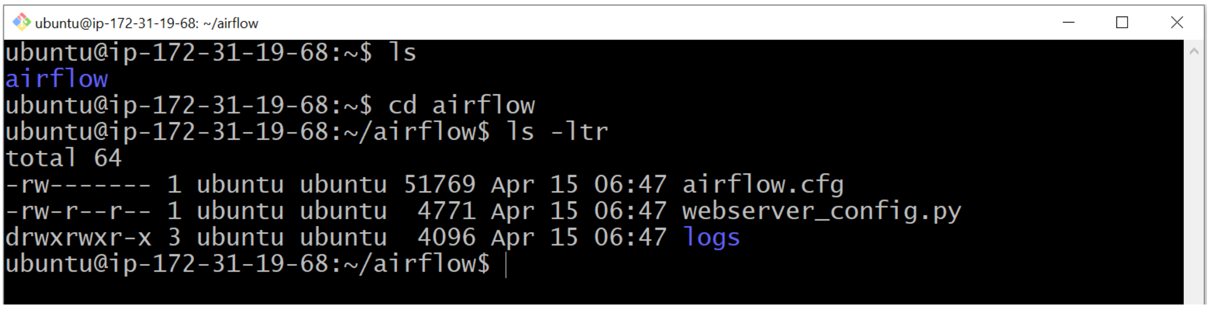
access key, access secret, consumer_key && consumer_secret)



Step 17

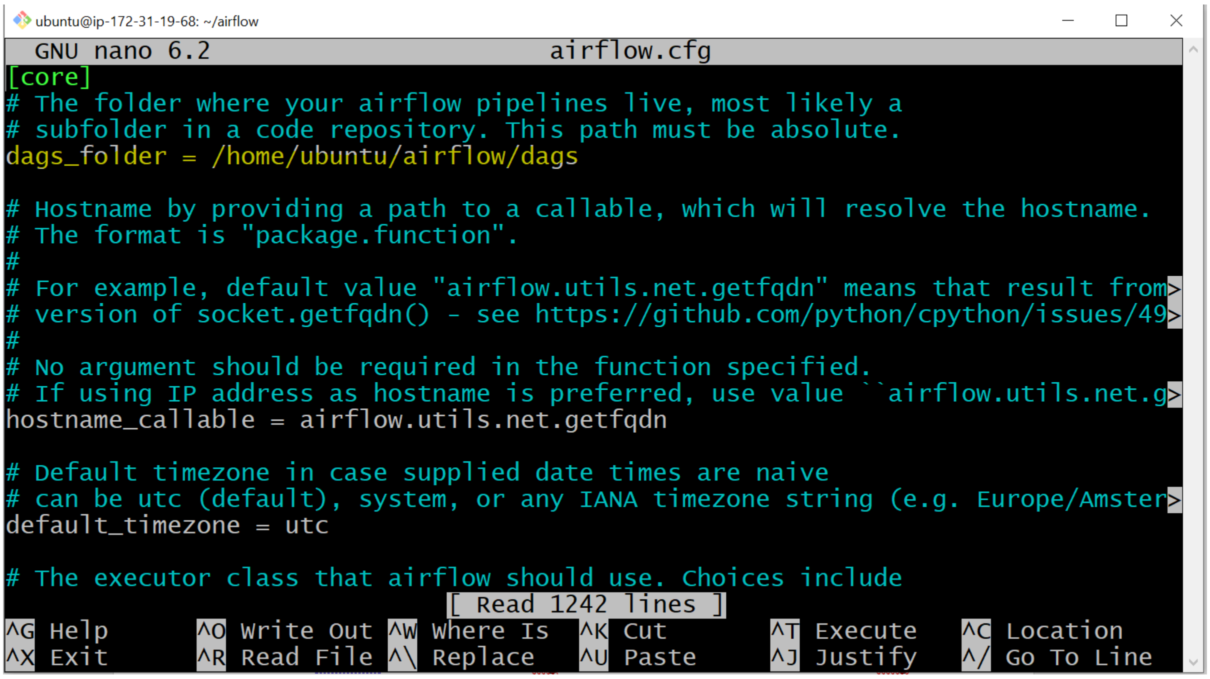
We have to create twitter_etl.py and twitter_dag.py files and upload inside the airflow. So some settings need to be changes .

A. First need to check airflow folder structure.



```
ubuntu@ip-172-31-19-68:~$ ls
airflow
ubuntu@ip-172-31-19-68:~$ cd airflow
ubuntu@ip-172-31-19-68:~/airflow$ ls -ltr
total 64
-rw------- 1 ubuntu ubuntu 51769 Apr 15 06:47 airflow.cfg
-rw-r--r-- 1 ubuntu ubuntu  4771 Apr 15 06:47 webserver_config.py
drwxrwxr-x 3 ubuntu ubuntu  4096 Apr 15 06:47 logs
ubuntu@ip-172-31-19-68:~/airflow$
```

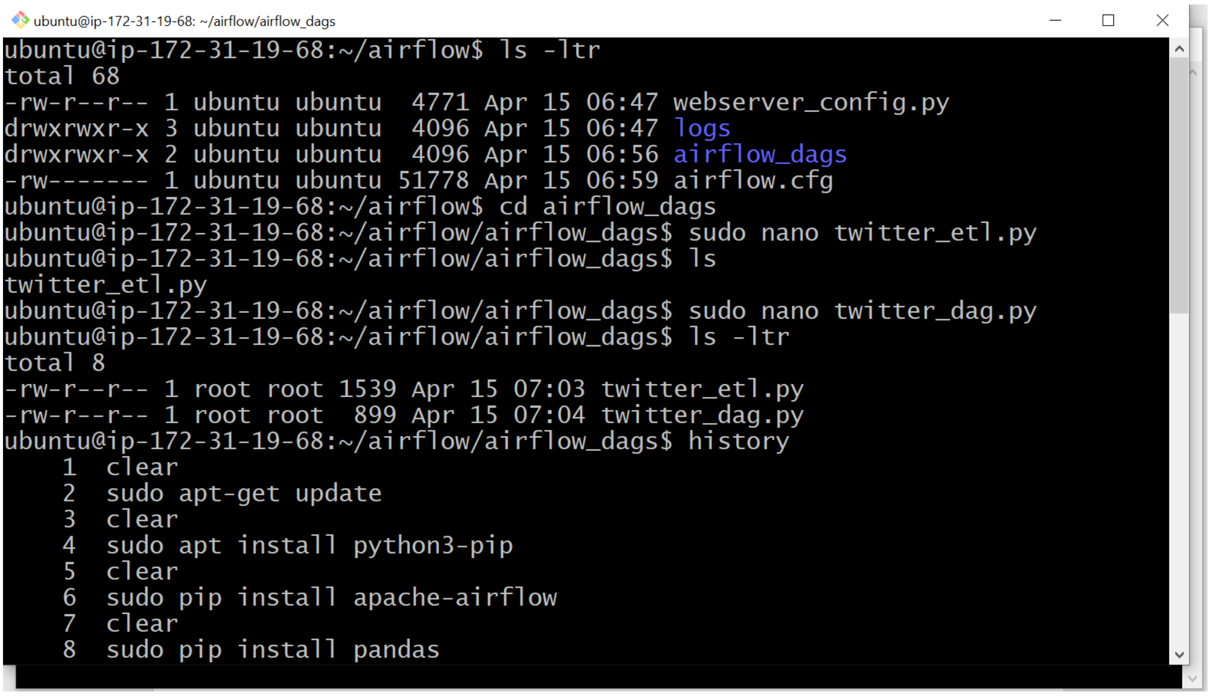
B. We must edit the airflow .cfg file in order to add our own airflow dag folder and create the ETL. We must change the dag folder name.



```
  GNU nano 6.2                     airflow.cfg
[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".
#
# For example, default value "airflow.utils.net.getfqdn" means that result from>
# version of socket.getfqdn() - see https://github.com/python/cpython/issues/49>
#
# No argument should be required in the function specified.
# If using IP address as hostname is preferred, use value ``airflow.utils.net.g>
hostname_callable = airflow.utils.net.getfqdn

# Default timezone in case supplied date times are naive
# can be utc (default), system, or any IANA timezone string (e.g. Europe/Amster>
default_timezone = utc

# The executor class that airflow should use. Choices include
                              [ Read 1242 lines ]
^G Help        ^O Write Out ^W Where Is   ^K Cut         ^T Execute     ^C Location
^X Exit        ^R Read File ^\ Replace    ^U Paste       ^J Justify     ^/ Go To Line
```

Need to add airflow_dags folder inside airflow.cfg file.

dags_folder = /home/ubuntu/airflow/airflow_dags .



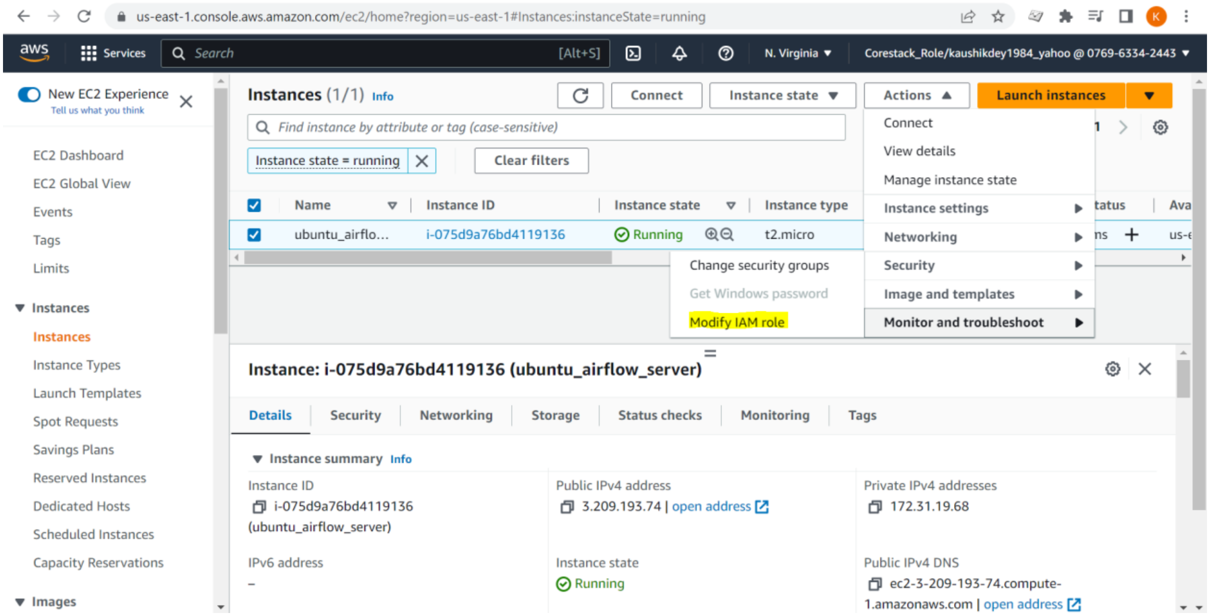
```
ubuntu@ip-172-31-19-68:~/airflow$ ls -ltr
total 68
-rw-r--r-- 1 ubuntu ubuntu  4771 Apr 15 06:47 webserver_config.py
drwxrwxr-x 3 ubuntu ubuntu  4096 Apr 15 06:47 logs
drwxrwxr-x 2 ubuntu ubuntu  4096 Apr 15 06:56 airflow_dags
-rw------- 1 ubuntu ubuntu 51778 Apr 15 06:59 airflow.cfg
ubuntu@ip-172-31-19-68:~/airflow$ cd airflow_dags
ubuntu@ip-172-31-19-68:~/airflow/airflow_dags$ sudo nano twitter_etl.py
ubuntu@ip-172-31-19-68:~/airflow/airflow_dags$ ls
twitter_etl.py
ubuntu@ip-172-31-19-68:~/airflow/airflow_dags$ sudo nano twitter_dag.py
ubuntu@ip-172-31-19-68:~/airflow/airflow_dags$ ls -ltr
total 8
-rw-r--r-- 1 root root 1539 Apr 15 07:03 twitter_etl.py
-rw-r--r-- 1 root root  899 Apr 15 07:04 twitter_dag.py
ubuntu@ip-172-31-19-68:~/airflow/airflow_dags$ history
    1  clear
    2  sudo apt-get update
    3  clear
    4  sudo apt install python3-pip
    5  clear
    6  sudo pip install apache-airflow
    7  clear
    8  sudo pip install pandas
```

Step 18

Now restart the airflow server to take effect the new dag changes. The following screenshot is given below.

Step 19

To access the s3 bucket inside EC2 instance we have to add modify IAM role. The screenshot is given below.



us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Instances:instanceState=running

Corestack_Role/kaushikdey1984_yahoo @ 0769-6334-2443

Instances (1/1)

Instance state = running

Name	Instance ID	Instance state	Instance type
ubuntu_airflo...	i-075d9a76bd4119136	Running	t2.micro

Instance: i-075d9a76bd4119136 (ubuntu_airflow_server)

Instance summary

Instance ID: i-075d9a76bd4119136 (ubuntu_airflow_server)

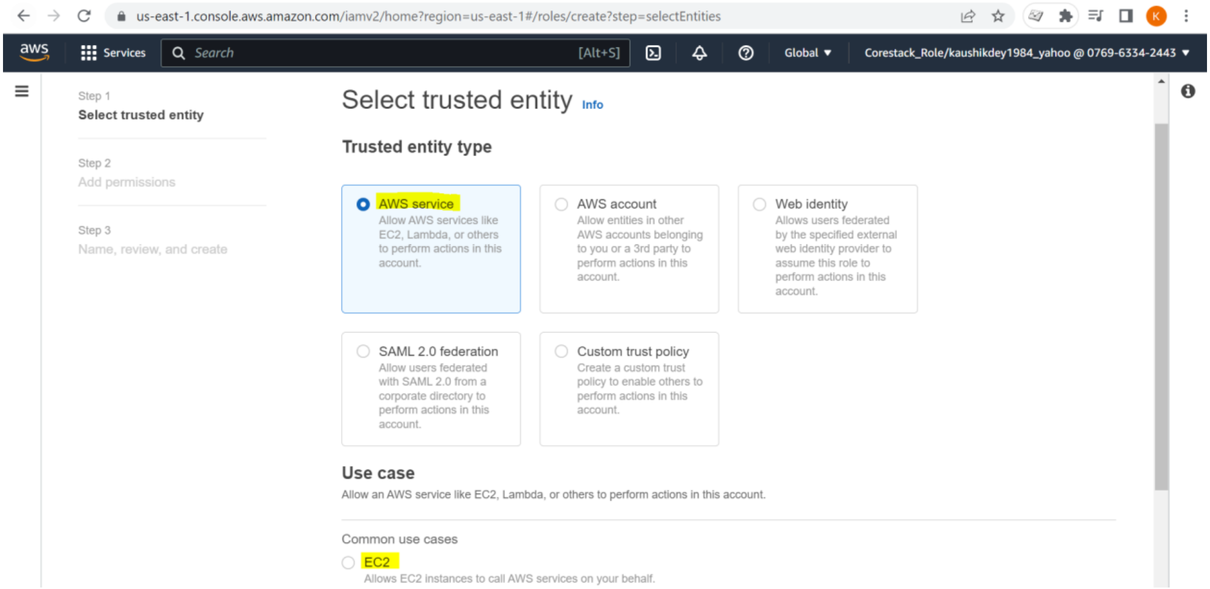
Public IPv4 address: 3.209.193.74 | open address

Private IPv4 addresses: 172.31.19.68

IPv6 address: –

Instance state: Running

Public IPv4 DNS: ec2-3-209-193-74.compute-1.amazonaws.com | open address



us-east-1.console.aws.amazon.com/iamv2/home?region=us-east-1#/roles/create?step=selectEntities

Corestack_Role/kaushikdey1984_yahoo @ 0769-6334-2443

Select trusted entity

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.

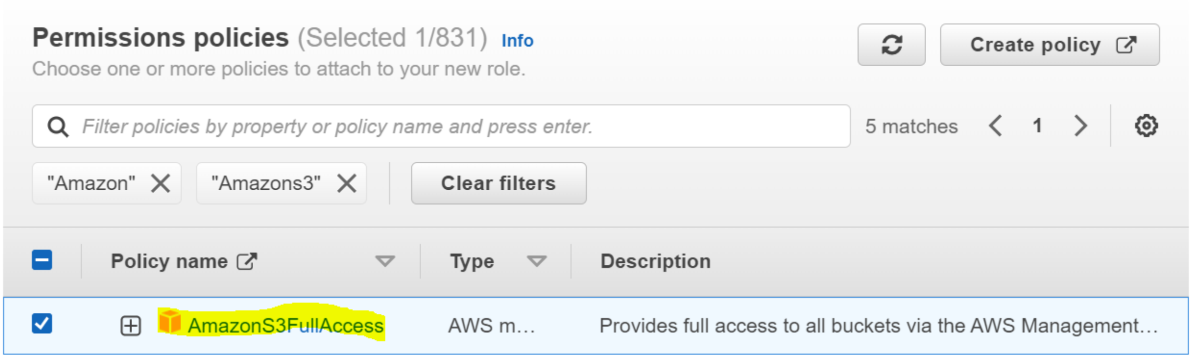
Use case

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

Common use cases

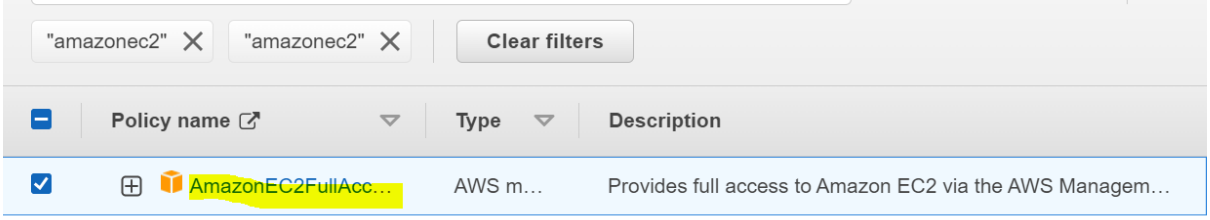
○ EC2
Allows EC2 instances to call AWS services on your behalf.

We must provide two access. (AmazonS3FullAccess && AmazonEc2FullAccess)


Permissions policies (Selected 1/831) Info

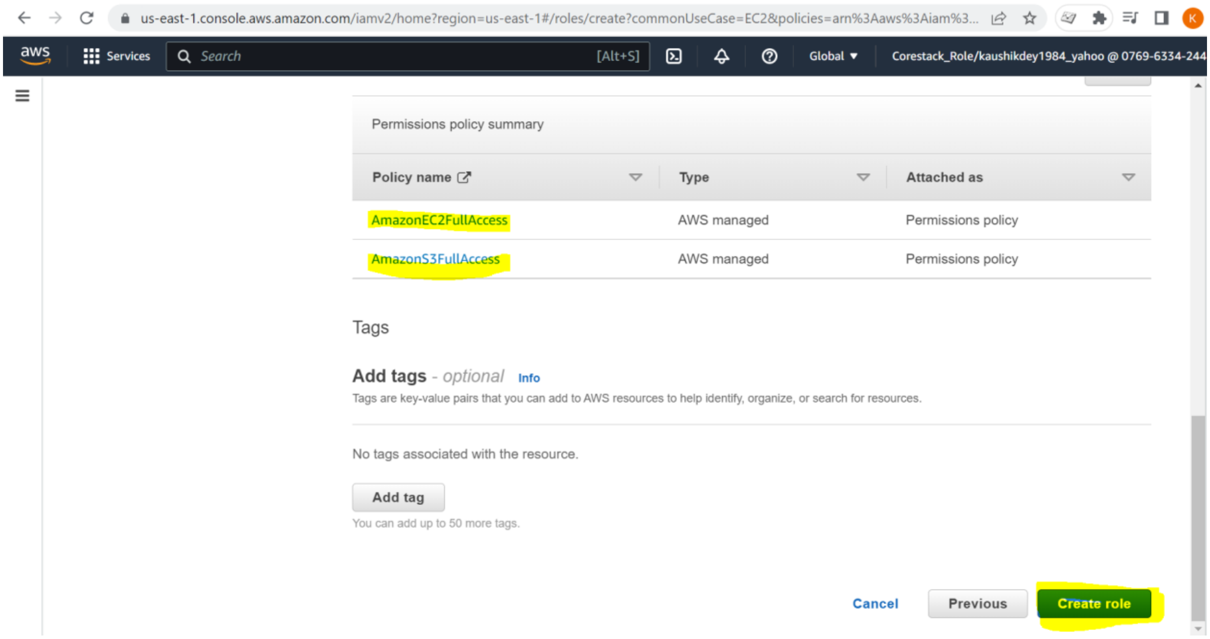
Choose one or more policies to attach to your new role.

	Policy name	Type	Description
☑	AmazonS3FullAccess	AWS m…	Provides full access to all buckets via the AWS Management…



	Policy name	Type	Description
☑	AmazonEC2FullAcc…	AWS m…	Provides full access to Amazon EC2 via the AWS Managem…

Now we have to create one role , the screenshot is given below.


us-east-1.console.aws.amazon.com/iamv2/home?region=us-east-1#/roles/create?commonUseCase=EC2&policies=arn%3Aaws%3Aiam%3...

Permissions policy summary

Policy name	Type	Attached as
AmazonEC2FullAccess	AWS managed	Permissions policy
AmazonS3FullAccess	AWS managed	Permissions policy

Tags

Add tags - *optional* Info

Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.

No tags associated with the resource.

You can add up to 50 more tags.