Setup Document

Set up Ubuntu in EC2

- Go to AWS EC2 and create an instance.
- Choose Ubuntu Server 18.04 and t3a.medium.
- Generate key pair.
- Launch it.
- Go to security group
- Add inbound rules
    - Type:Custom TCP, Port range:8888, Source:0.0.0.0/0, Description:Jupyter
    - Type:Custom TCP, Port range:8080, Source:0.0.0.0/0, Description:Airflow
    - Type:MYSQL/Aurora, Port range:3306, Source:own security group, Description:MySQL database
- Add outbound rules
    - Type:All traffic, Port range:All, Destination:0.0.0.0/0

Use Ubuntu in EC2

- Open PowerShell.
- Go to the directory which contains the generated key pair.
- $ ssh -i “KEYPAIR_PEM_NAME” ubuntu@[Public DNS (IPv4)]

Set up Jupyter Lab

- $ sudo apt update
- $ sudo apt upgrade
- $ sudo apt install python3-pip
- $ pip3 --version
- $ sudo pip3 install --upgrade pip
- $ pip3 --version
- $ sudo apt install python3-dev
- $ sudo -H pip3 install jupyter
- $ jupyter notebook
    - Just check whether the notebook works, and quit by Ctrl + C.
- $ sudo -H pip3 install jupyterlab
- $ jupyter lab
    - Just checking, Ctrl + C.
- $ jupyter notebook --generate-config
- $ ls -l -a
    - Check .jupyter directory
- $ jupyter notebook password

- $ cd .jupyter/
- $ openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout mykey.key -out mycert.pem
    - It asks to type your information. For example, type . for all except common name is your name.
- $ cat jupyter_notebook_config.json
    - Copy password inside.
- $ pwd
    - Use this directory in the following. /home/ubuntu/.jupyter
- $ vim jupyter_notebook_config.py
    - Remove # from the following code and modify
    - c.NotebookApp.certfile = u'/home/ubuntu/.jupyter/mycert.pem'
    - c.NotebookApp.keyfile = u'/home/ubuntu/.jupyter/mykey.key'
    - c.NotebookApp.ip = '*'
    - c.NotebookApp.password = u'[PASSWORD IN JSON]'
    - c.NotebookApp.open_browser = False
    - c.NotebookApp.port = 8888
    - c.NotebookApp.notebook_dir = '/home/ubuntu'
- $ jupyter lab
- Go to EC2 instance page and copy IPv4 Public IP
- Open Google Chrome, and type in search window, https://[IPv4 Public IP]:8888/

Set up Airflow

- $ sudo -H pip3 install apache-airflow
- $ airflow info
- $ airflow initdb
- $ airflow webserver -p 8080
- Open Google Chrome, and type in search window, http://[IPv4 Public IP]:8080/

Use Airflow

- $ cd airflow/
- $ mkdir dags
- Make python file which contains dag and tasks info
- $ airflow test DAG_ID TASK_ID YYYY-MM-DD
    - Test whether each task works before running in schedule.
- $ airflow scheduler
- Go to Airflow UI, and enable DAG

Keep running Jupyter and Airflow

- $ screen --version
- $ screen -S jupyter
- $ jupyter lab
    - In a new terminal
- Ctrl + A, then D
    - Detached from the screen
- $ screen -S airflow-webserver
- $ airflow webserver -p 8080
- Ctrl + A, then D
- $ screen -S airflow-scheduler
- $ airflow scheduler
- Ctrl + A, then D
- $ screen -ls

Set up RDS

- Easy create
- MySQL
- DB instance size: Free tier
- DB instance identifier: jupyter-database
- Master username: admin
- Master password: [YOUR_PASSWORD]
- Security group: same as EC2
- Public accessibility: Yes
- Enable IAM DB authentication

Connect RDS from EC2

- $ sudo apt install mysql-client-core-5.7
- $ mysql -h [END_POINT] -P 3306 -u admin -p
- Enter password: [YOUR_PASSWORD] from easy create RDS

Set up MySQL database and table

- mysql> SHOW DATABASES;
- mysql> USE jupyterdb;
- mysql> SHOW TABLES;
- mysql> CREATE DATABASE jupyterdb;

- mysql> SHOW DATABASES;
- mysql> USE jupyterdb;
    - Default database switch to jupyterdb
- mysql> SELECT DATABASE();
- mysql>
    - CREATE TABLE jupyterdb….

Resource

- Ubuntu
    - https://www.digitalocean.com/community/tutorials/how-to-upgrade-to-ubuntu-16-04-lts
- Jupyter
    - https://www.digitalocean.com/community/tutorials/how-to-set-up-jupyter-notebook-with-python-3-on-ubuntu-18-04
- Linux command
    - Screen command
        - https://linuxize.com/post/how-to-use-linux-screen/