=~=~=~=~=~=~=~=~=~=~=~= AIRFLOW SETUP =~=~=~=~=~=~=~=~=~=~=~=

This setup was done using:

- Fresh installation of ***Linux Ubuntu 18.04.2 LTS (GNU/Linux 4.18.0-15-generic x86\_64)*** under VM environment.

- ***PostgreSQL 10.6***

- ***Python 2.7.15***

- ***Installed under Linux superuser (not root)***

- ***I didn’t use any virtual environment to install Airflow***

THEN, LET’S GET INTO IT!

-- Make sure to update the packages for your fresh Ubuntu.

*ubuntu@ubuntu-VirtualBox:~$ sudo apt-get update*

-= SETUP POSTGRESQL =-

-- Install PostgreSQL

*ubuntu@ubuntu-VirtualBox:~$ sudo apt-get install postgresql postgresql-contrib*

-- Create PostgreSQL user and database for airflow meta db purpose via PostgreSQL command line.

Create database ‘airflow’.

*ubuntu@ubuntu-VirtualBox:~$ sudo -u postgres psql*

*psql (10.6 (Ubuntu 10.6-0ubuntu0.18.04.1))*

*Type "help" for help.*

*postgres=#*

*postgres=# CREATE DATABASE airflow;*

*CREATE DATABASE*

*postgres=#*

-- You can either create new PostgreSQL user to access this airflow database, or you can just use your existing PostgreSQL user. In this case, I was creating new user under name : ***ubuntu***.

*postgres=# CREATE USER ubuntu;*

*CREATE ROLE*

-- And then grant privileges to airflow database for new user above.

*postgres=# GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA public TO ubuntu;*

*GRANT*

-- Reconfig the ***‘pg\_hba.conf’*** to allow connection from airflow.

You can use either nano, vi, any kind of text editor that you’re familiar with.

*ubuntu@ubuntu-VirtualBox:~$ sudo vi /etc/postgresql/10/main/pg\_hba.conf*

Change this line

*# IPv4 local connections:*

*host all all 127.0.0.1/32 md5*

into

*# IPv4 local connections:*

*host all all 0.0.0.0/0 trust*

-- Next, we configure ***‘postgresql.conf’***.

*ubuntu@ubuntu-VirtualBox:~$ sudo vi /etc/postgresql/10/main/postgresql.conf*

Change this line

*#listen\_addresses = ‘localhost’ # what IP address(es) to listen on;*

into

*listen\_addresses = ‘\*’ # what IP address(es) to listen on;*

-- Restart the Postgre service to apply the changes.

*ubuntu@ubuntu-VirtualBox:~$ sudo service postgresql restart*

-= SETUP AIRFLOW =-

-- Install Python 2.7.

*ubuntu@ubuntu-VirtualBox:~$ sudo apt-get install python-setuptools*

-- Install pip.

*ubuntu@ubuntu-VirtualBox:~$ sudo apt-get install python-pip*

**-- Note** : If the default installed pip is not the up-to-date version, you may want to consider update it.

But if you believe that the pip you’re using is the latest version, then you can skip this step.

*ubuntu@ubuntu-VirtualBox:~$ sudo pip install --upgrade pip*

-- Install Airflow

-- Setup the airflow default home directory.

*ubuntu@ubuntu-VirtualBox:~$ export AIRFLOW\_HOME=~/airflow*

-- Next step is to install the system and python packages for airflow.

On some occasions, you will need to install some dependencies before installing Airflow packages.

Some of the examples are:

*sudo apt-get install libmysqlclient-dev  -- (dependency for airflow[mysql] package)*

*sudo apt-get install libssl-dev  -- (dependency for airflow[cryptograph] package)*

*sudo apt-get install libkrb5-dev -- (dependency for airflow[kerbero] package)*

*sudo apt-get install libsasl2-dev -- (dependency for airflow[hive] package)*

**-- Note** : Some people prefer to install the airflow under virtual environment, if this is the case you will need to prepare the virtual environment first, and then install the airflow under it later. Since I didn’t use it, I will explain this part another time.

-- Setup the airflow default home directory.

*ubuntu@ubuntu-VirtualBox:~$ export AIRFLOW\_HOME=~/airflow*

-- Just in case you run into this error message…

*raise RuntimeError("By default one of Airflow's dependencies installs a GPL "*

*RuntimeError: By default one of Airflow's dependencies installs a GPL dependency (unidecode). To avoid this dependency set SLUGIFY\_USES\_TEXT\_UNIDECODE=yes in your environment when you install or upgrade Airflow. To force installing the GPL version set AIRFLOW\_GPL\_UNIDECODE*

…use any of this command below :

*sudo AIRFLOW\_GPL\_UNIDECODE=yes pip install "apache-airflow[postgres, celery, rabbitmq]"*

OR

*sudo SLUGIFY\_USES\_TEXT\_UNIDECODE=yes pip install "apache-airflow[postgres, celery, rabbitmq]"*

-- Start up Airflow database.

*ubuntu@ubuntu-VirtualBox:~$ airflow initdb*

-- The command above will generate airflow.cfg file in Airflow’s home directory we set up earlier.

We need to change some line in that file.

*ubuntu@ubuntu-VirtualBox:~$ sudo vi airflow/airflow.cfg*

**-- Note** : There are 5 executor that you can use, which are SequentialExecutor, LocalExecutor, CeleryExecutor, DaskExecutor, KubernetesExecutor.

But the one I will use is the CeleryExecutor.

-- Change the executor, from ***SequentialExecutor*** to ***CeleryExecutor***

*executor = CeleryExecutor*

-- Pass along the connection info of the postgresql database airflow, by adding this line

*sql\_alchemy\_conn = postgresql+psycopg2://ubuntu@localhost:5432/airflow*

-- Change the broker\_url, since we’re using rabbitmq as the broker

*broker\_url = amqp://guest:guest@localhost:5672//*

-- Don’t forget to add the meta db connection to Celery backend as well

*sql\_alchemy\_conn = db+ postgresql://ubuntu@localhost:5432/airflow*

-- Start up Airflow database again to reload the changes we did to the configuration file.

*ubuntu@ubuntu-VirtualBox:~$ airflow initdb*

-- Install rabbitmq.

*ubuntu@ubuntu-VirtualBox:~$ sudo apt-get install rabbitmq-server*

-- And start the rabbitmq service.

*ubuntu@ubuntu-VirtualBox:~$ sudo service rabbitmq-server start*

Until this step, the airflow webserver, scheduler and worker, should be ready.

You can run these service with these command lines respectively :

*ubuntu@ubuntu-VirtualBox:~$ airflow webserver  
ubuntu@ubuntu-VirtualBox:~$ airflow scheduler  
ubuntu@ubuntu-VirtualBox:~$ airflow worker*

~=~=Set airflow services as daemon service using systemd =~=~

Once you can run all those three manually, then we can register them as system service.

Get the necessary files from this link:

*https://github.com/apache/airflow/tree/master/scripts/systemd*

The files that you will need are: *airflow, airflow-flower.service, airflow-kerberos.service, airflow-scheduler.service, airflow-webserver.service, airflow-worker.service, airflow-conf*.

-- file ***airflow***

-- Copy ***airflow*** file to ***/etc/sysconfig/***, if you don’t have it yet, make a new directory.

On airflow file, you should see these 2 lines:

# AIRFLOW\_CONFIG= 🡨 *point this to your airflow.cfg file*

# AIRFLOW\_HOME= 🡨 *point this to your airflow home directory*

*AIRFLOW\_CONFIG=****/home/ubuntu/airflow/airflow.cfg***

*AIRFLOW\_HOME=****/home/ubuntu/airflow/***

Uncomment or add new line, and fill the blank part based on your configuration.

-- file ***airflow.conf***

-- Copy the ***airflow.conf*** to ***/etc/tmpfiles.d/***

-- Copy all of the file services to ***/lib/systemd/system/***, or wherever all of your system services file located.

Adjust the content of the services file to your configuration.

For example: let’s try to configure the webserver service file.

*[Unit]*

*Description=Airflow webserver daemon*

*After=network.target postgresql.service mysql.service redis.service rabbitmq-server.service*

*Wants=postgresql.service mysql.service redis.service rabbitmq-server.service*

*[Service]*

*EnvironmentFile=/etc/sysconfig/airflow*

*User=****ubuntu*** *🡨 set to your Linux username*

*Group=****ubuntu*** *🡨 set to your Linux username group*

*Type=simple*

*ExecStart=****/usr/local/bin/*** *airflow webserver --pid /run/airflow/webserver.pid*

*Restart=on-failure*

*RestartSec=5s*

*PrivateTmp=true*

*[Install]*

*WantedBy=multi-user.target*

For this line above:

***ExecStart=/usr/bin/*** *airflow webserver --pid /run/airflow/webserver.pid*

It may located on a different place compared to mine, search where is the command “airflow” located.

*ubuntu@ubuntu-VirtualBox:/etc/sysconfig$ which airflow*

*/usr/local/bin/airflow*

-- Copy ***airflow.conf*** to your environment directory.

-- Enable the service that you have configure using the command like this example: ***systemctl enable airflow-webserver***

*ubuntu@ubuntu-VirtualBox:/etc/sysconfig$ sudo systemctl enable airflow-webserver.service*

*Created symlink /etc/systemd/system/multi-user.target.wants/airflow-webserver.service → /lib/systemd/system/airflow-webserver.service.*

-- And then start the service using ***systemctl start airflow-webserver***

*ubuntu@ubuntu-VirtualBox:/etc/sysconfig$ sudo systemctl start airflow-webserver.service*

-- Run ***systemctl status airflow-webserver*** to check the status of the service, whether it is failed or running successfully, it should show up in the log.

*ubuntu@ubuntu-VirtualBox:/etc/sysconfig$ sudo systemctl status airflow-webserver.service*

*● airflow-webserver.service - Airflow webserver daemon*

*Loaded: loaded (/lib/systemd/system/airflow-webserver.service; enabled; vendor preset: enabled)*

*Active: active (running) since Mon 2019-07-15 12:20:10 WIB; 18min ago*

*Main PID: 2043 (/usr/bin/python)*

*Tasks: 7 (limit: 4915)*

*CGroup: /system.slice/airflow-webserver.service*

*├─2043 /usr/bin/python /usr/local/bin/airflow webserver --pid /run/airflow/webserver.pid*

*├─2157 gunicorn: master [airflow-webserver]*

*├─7884 [ready] gunicorn: worker [airflow-webserver]*

*├─8066 [ready] gunicorn: worker [airflow-webserver]*

*├─8319 [ready] gunicorn: worker [airflow-webserver]*

*├─8507 [ready] gunicorn: worker [airflow-webserver]*

*└─8685 [ready] gunicorn: worker [airflow-webserver]*

-- From my own experiences, most of the time the airflow webserver will stop after running for a moment, and it came with an error message like below :

*“Error: /run/airflow doesn't exist. Can't create pidfile.”*

-- To solve this just add these two lines under the [Service] tab in the airflow-webserver.service file

*RuntimeDirectory=airflow*

*RuntimeDirectoryMode=0775*

-- As for the other two process, airflow-scheduler and airflow-worker, just follow the same step like above, you can see the file’s content sample below.

-- airflow-scheduler.service

*[Unit]*

*Description=Airflow scheduler daemon*

*After=network.target postgresql.service mysql.service redis.service rabbitmq-server.service*

*Wants=postgresql.service mysql.service redis.service rabbitmq-server.service*

*[Service]*

*EnvironmentFile=/etc/sysconfig/airflow*

*User=****ubuntu***

*Group=****ubuntu***

*Type=simple*

*ExecStart=****/usr/local/bin/*** *airflow scheduler*

*Restart=always*

*RestartSec=5s*

*[Install]*

*WantedBy=multi-user.target*

-- airflow-worker.service

*[Unit]*

*Description=Airflow celery worker daemon*

*After=network.target postgresql.service mysql.service redis.service rabbitmq-server.service*

*Wants=postgresql.service mysql.service redis.service rabbitmq-server.service*

*[Service]*

*EnvironmentFile=/etc/sysconfig/airflow*

*User=****ubuntu***

*Group=****ubuntu***

*Type=simple*

*ExecStart=****/usr/local/bin/*** *airflow worker*

*Restart=always*

*RestartSec=5s*

*[Install]*

*WantedBy=multi-user.target*

I hope this will help you to setup your Airflow easier.

Good luck!!