Airflow in Ubuntu[[1]](#footnote-1)

# Description

Current Ubuntu has anaconda3 installed already, in the past, there have been many times airflow cannot be installed successfully.

rxie@ubuntu:~$ echo $PATH

/home/rxie/venv/bin:/home/rxie/Downloads/spark//bin:/home/rxie/anaconda3/bin:/usr/local/src/scala/scala-2.10.4/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/usr/lib/jvm/java-8-oracle/bin:/usr/lib/jvm/java-8-oracle/db/bin:/usr/lib/jvm/java-8-oracle/jre/bin:/usr/share/sbt/bin/sbt-launch.jar/bin:/home/rxie/Downloads/spark//bin

Note: this means anaconda3 is the current python3 environment and will be used to install airflow

# Create a virtual environment for airflow

## Install virtualenv

virtualenv is used to create the dedicated environment for airflow. Other method (conda env based on yml) should be similar and should be working too.

conda install virtualenv

## Create an env

Create an env with the name of venv under the current python3 folder

virtualenv -p `which python3` venv

(venv) rxie@ubuntu:~$ which python3

/home/rxie/venv/bin/python3

(venv) rxie@ubuntu:~$ /home/rxie/venv/bin/python3

Python 3.6.3 |Anaconda, Inc.| (default, Oct 13 2017, 12:02:49)

## Activate the env

source venv/bin/activate

## Airflow specific option

export SLUGIFY\_USES\_TEXT\_UNIDECODE=yes

## Install airflow

pip install apache-airflow

## Home folder for Airflow Post-installation

export AIRFLOW\_HOME=`pwd`/airflow\_home

Note: AIRFLOW\_HOME is only for DAGs and plugins.

## Check airflow version

airflow version

Note: after this command, airflow.cfg is created in airflow\_home folder

(venv) rxie@ubuntu:~/airflow\_home$ ls

airflow.cfg airflow.db logs unittests.cfg

## Initialize Airflow DB

(venv) $ airflow initdb

Note: this creates airflow SQLite database, for production environment, mysql is recommended.

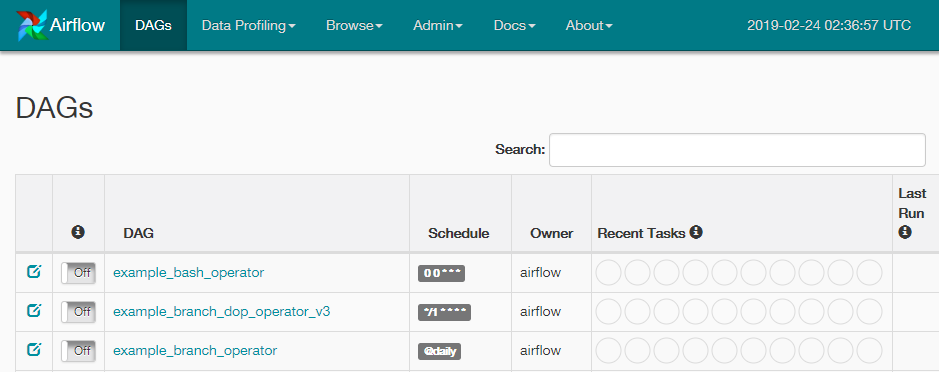
# Start airflow

## Start airflow webserver

(venv)airflow webserver

## Access airflow

<http://192.168.112.141:8080>



# Airflow DAG

## Create DAG

Under airflow\_home, create a dedicated folder for DAGs

Within dags folder, create DAG hello\_world.py

(venv) rxie@ubuntu:~/airflow\_home/dags$ cat hello\_world.py

from datetime import datetime

from airflow import DAG

from airflow.operators.dummy\_operator import DummyOperator

from airflow.operators.python\_operator import PythonOperator

def print\_hello():

return 'Hello world!'

dag = DAG('hello\_world', description='Simple tutorial DAG',

schedule\_interval='0 12 \* \* \*',

start\_date=datetime(2017, 3, 20), catchup=False)

dummy\_operator = DummyOperator(task\_id='dummy\_task', retries=3, dag=dag)

hello\_operator = PythonOperator(task\_id='hello\_task', python\_callable=print\_hello, dag=dag)

dummy\_operator >> hello\_operator

## Run DAG

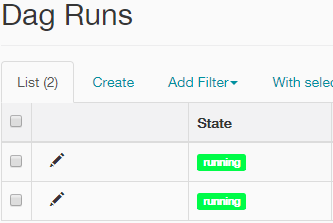
Start airflow scheduler

By starting scheduler, the new DAG will be loaded into the scheduler, where 16 other example dags are also loaded.



QUESTION:

Why it hangs there after triggered?



# MySQL installation[[2]](#footnote-2)

apt-get install mysql-server

root/@M

## Config mysql service to start when host is started

(venv) rxie@ubuntu:~/Downloads$ sudo ufw allow mysql

Rules updated

Rules updated (v6)

(venv) rxie@ubuntu:~/Downloads$ sudo systemctl start mysql

(venv) rxie@ubuntu:~/Downloads$ sudo systemctl enable mysql

Synchronizing state of mysql.service with SysV init with /lib/systemd/systemd-sysv-install...

Executing /lib/systemd/systemd-sysv-install enable mysql

## Connect to mysql

(venv) rxie@ubuntu:~/Downloads$ mysql -u root -p

Enter password:

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 4

Server version: 5.7.25-0ubuntu0.16.04.2 (Ubuntu)

Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its

affiliates. Other names may be trademarks of their respective

owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;

+--------------------+

| Database |

+--------------------+

| information\_schema |

| mysql |

| performance\_schema |

| sys |

+--------------------+

4 rows in set (0.00 sec)

## MySQL Config

Location: /etc/mysql

Detailed configuration:

/usr/sbin/mysqld --help –verbose

……..

Variables (--variable-name=value)

and boolean options {FALSE|TRUE} Value (after reading options)

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

abort-slave-event-count 0

allow-suspicious-udfs FALSE

archive ON

auto-increment-increment 1

auto-increment-offset 1

autocommit TRUE

automatic-sp-privileges TRUE

avoid-temporal-upgrade FALSE

back-log 80

basedir /usr/thread-cache-size 8

thread-handling one-thread-per-connection

thread-stack 196608

time-format %H:%i:%s

tls-version TLSv1,TLSv1.1

tmp-table-size 16777216

tmpdir /tmp

transaction-alloc-block-size 8192

transaction-isolation REPEATABLE-READ

transaction-prealloc-size 4096

transaction-read-only FALSE

transaction-write-set-extraction OFF

updatable-views-with-limit YES

validate-user-plugins TRUE

verbose TRUE

wait-timeout 28800

## MySQL Log

Location: /var/log/mysql

(venv) rxie@ubuntu:~/Downloads$ cd /var/log/mysql

(venv) rxie@ubuntu:/var/log/mysql$ ls

error.log

# PostgreSQL[[3]](#footnote-3)[[4]](#footnote-4)

## Installation:

sudo apt-get update

sudo apt-get install postgresql postgresql-contrib

## Config PostgreSQL service to start when host is started

update-rc.d postgresql enable

## Start PostgreSQL

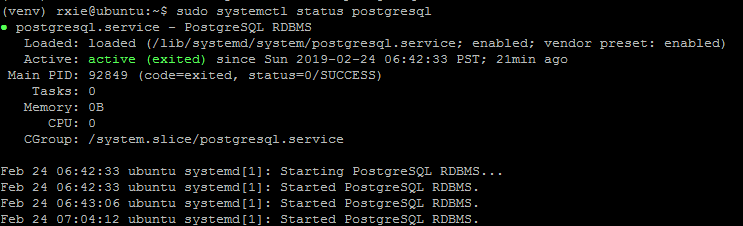
sudo systemctl start postgresql

## Stop PostgreSQL

sudo systemctl start postgresql

## Status PostgreSQL

sudo systemctl status postgresql



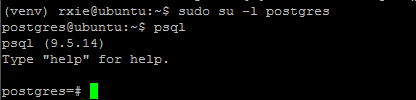
## Access PostgreSQL

### Login in as postgres user

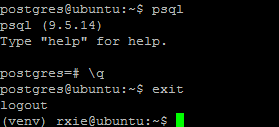
sudo su -l postgres (no password needed)

### run psql as interactive shell

psql



Ctrl+D to exit



### List all users

(venv) rxie@ubuntu:~$ cut -d: -f1 /etc/passwd

root

daemon

bin

sys

sync

games

man

lp

mail

news

uucp

proxy

www-data

backup

list

irc

gnats

nobody

systemd-timesync

systemd-network

systemd-resolve

systemd-bus-proxy

syslog

\_apt

messagebus

uuidd

lightdm

whoopsie

avahi-autoipd

avahi

dnsmasq

colord

speech-dispatcher

hplip

kernoops

pulse

rtkit

saned

usbmux

rxie

sshd

sbt

user1

user2

user3

sid\_project1

sid\_project2

sid\_project3

mysql

postgres

Note: a user postgres is created in linux during postgres installation

## Secure PostgreSQL database

### Set password for Linux user (postgres)

sudo passwd postgres

@P

### Set password for DB administrator (postgres)

su - postgres

psql

postgres=# \password

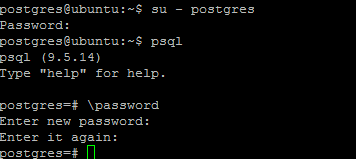
postgres=# \password postgres

postgres=# \password postgres

Enter new password:

Enter it again:

@P



Remove PostgreSQL

sudo apt-get --purge remove postgresql

sudo rm -rf /var/lib/postgresql/

sudo rm -rf /var/log/postgresql/

sudo rm -rf /etc/postgresql/

sudo rm -rf /etc/postgresql-common

## Delete linux user postgres

userdel postgres

## PostgreSQL simple db commands

### Create database

Login as root user postgres

postgres@ubuntu:~$ su - postgres

Password:

postgres@ubuntu:~$ createdb test

postgres@ubuntu:~$ psql test

psql (9.5.14)

Type "help" for help.

### Create user

test=# create user rxie with superuser login password 'password';

CREATE ROLE

### List roles

test=# \du

List of roles

Role name | Attributes | Member of

-----------+------------------------------------------------------------+-----------

postgres | Superuser, Create role, Create DB, Replication, Bypass RLS | {}

rxie | Superuser | {}

test=# psql -h localhost -d test -U rxie

test-# \l

List of databases

Name | Owner | Encoding | Collate | Ctype | Access privileges

-----------+----------+----------+-------------+-------------+-----------------------

postgres | postgres | UTF8 | en\_US.UTF-8 | en\_US.UTF-8 |

template0 | postgres | UTF8 | en\_US.UTF-8 | en\_US.UTF-8 | =c/postgres +

| | | | | postgres=CTc/postgres

template1 | postgres | UTF8 | en\_US.UTF-8 | en\_US.UTF-8 | =c/postgres +

| | | | | postgres=CTc/postgres

test | postgres | UTF8 | en\_US.UTF-8 | en\_US.UTF-8 |

(4 rows)

### Add new user and drop it

test=# create user fake\_user login password 'fake\_user';

CREATE ROLE

test=# \du

List of roles

Role name | Attributes | Member of

-----------+------------------------------------------------------------+-----------

fake\_user | | {}

postgres | Superuser, Create role, Create DB, Replication, Bypass RLS | {}

rxie | Superuser | {}

test=# drop user fake\_user;

DROP ROLE

test=# \du

List of roles

Role name | Attributes | Member of

-----------+------------------------------------------------------------+-----------

postgres | Superuser, Create role, Create DB, Replication, Bypass RLS | {}

rxie | Superuser | {}

## PostgreSQL basic commands

### List all databases

\d

### Change current database

\c dbname

### Show all tables in the current database

\d

### Show all fields for specific table

\d tablename

### QUERY

Same as usual SQL grammar

### Exit

\ q

# Open Exchange Practice

## Sign up with Open Exchange Rates[[5]](#footnote-5)

Email: paslechoix0@gmail.com

API ID: 1c24db0a40454c22986717b7c16e0be5

## Basic API Call

curl -S https://openexchangerates.org/api/latest.json?app\_id=1c24db0a40454c22986717b7c16e0be5 | python -mjson.tool

# Error and troubleshooting

sqlalchemy.exc.OperationalError: (sqlite3.OperationalError) no such table: connection [SQL: 'SELECT connection.conn\_id AS connection\_conn\_id \nFROM connection GROUP BY connection.conn\_id'] (Background on this error at: http://sqlalche.me/e/e3q8)

# Readings:

<https://tech.marksblogg.com/airflow-postgres-redis-forex.html>

<https://tech.marksblogg.com/python-big-data-airflow-jupyter-notebook-hadoop-3-hive-presto.html>

1. <http://michal.karzynski.pl/blog/2017/03/19/developing-workflows-with-apache-airflow/> [↑](#footnote-ref-1)
2. <https://support.rackspace.com/how-to/installing-mysql-server-on-ubuntu/> [↑](#footnote-ref-2)
3. <https://www.godaddy.com/garage/how-to-install-postgresql-on-ubuntu-14-04/> [↑](#footnote-ref-3)
4. <https://www.itzgeek.com/how-tos/linux/ubuntu-how-tos/how-to-install-postgresql-10-on-ubuntu-18-04-lts.html> [↑](#footnote-ref-4)
5. <https://openexchangerates.org> [↑](#footnote-ref-5)