Airflow Install and manual(Python3.6)

Description:

Airflow is a platform to programmatically author, schedule and monitor workflows.

Use Airflow to author workflows as Directed Acyclic Graphs (DAGs) of tasks. The Airflow scheduler executes your tasks on an array of workers while following the specified dependencies. Rich command line utilities make performing complex surgeries on DAGs a snap. The rich user interface makes it easy to visualize pipelines running in production, monitor progress, and troubleshoot issues when needed.

When workflows are defined as code, they become more maintainable, versionable, testable, and collaborative.
https://airflow.apache.org/

Airflow product environment installation

Prerequisites:

You'll need to be logged in as root or user with sudo access to be able to install packages on your Ubuntu system.

1. Ububtu 18.04
2. Python 3.6
3. pip latest version

Installation

- □ Install prerequisites packages
- □ Install MySQL (MySQL 5.7 later is supported)
- □ Install Airflow
- □ Install redis
- □ Setting CeleryExecutor
- □ Setting smtp
- □ Start Ailflow
- □ Running airflow scheduler as a daemon process

STEP 1 :INSTALL PREREQUISITES PACKAGES

```
$ sudo apt update
$ sudo apt install python3-pip
$ sudo apt install build-essential zlib1g-dev libncurses5-dev libgdbm-dev libnss3-dev
$ sudo apt-get install autoconf libtool pkg-config python-opengl python-pil python-pyr
$ sudo apt-get install python-pip python-dev libmysqlclient-dev
$ sudo apt-get install libmysqlclient-dev
$ sudo apt-get install python-setuptools
```

STEP2 :INSTALL MYSQL (MYSQL 5.7 LATER IS SUPPORTED)

https://www.digitalocean.com/community/tutorials/how-to-install-mysql-on-ubuntu-18-04

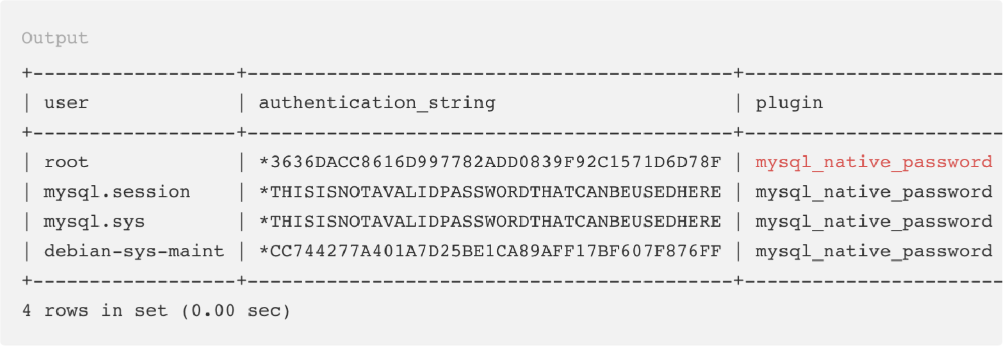
1. Install MySQL

```
$ sudo apt update
$ sudo apt install mysql-server
$ sudo mysql_secure_installation
```

2. (Optional) Adjusting User Authentication and Privileges

```
$ sudo mysql

mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'your_pa
mysql> FLUSH PRIVILEGES;
mysql> SELECT user,authentication_string,plugin,host FROM mysql.user;
mysql > exit
```

Be sure to change your_password to a strong password of your choosing, and note that this command will change the root password.



```
Output
+------------------+-------------------------------------------+----------------------
| user             | authentication_string                     | plugin
+------------------+-------------------------------------------+----------------------
| root             | *3636DACC8616D997782ADD0839F92C1571D6D78F | mysql_native_password
| mysql.session    | *THISISNOTAVALIDPASSWORDTHATCANBEUSEDHERE | mysql_native_password
| mysql.sys        | *THISISNOTAVALIDPASSWORDTHATCANBEUSEDHERE | mysql_native_password
| debian-sys-maint | *CC744277A401A7D25BE1CA89AFF17BF607F876FF | mysql_native_password
+------------------+-------------------------------------------+----------------------
4 rows in set (0.00 sec)
```

1. Create airflow database

```
$ mysql -u root -p
mysql> create database airflow;
```

3. Create user airflow

```
mysql> create user airflow@'localhost' identified by 'airflow';
mysql> grant all on airflow.* to airflow@'localhost';
mysql> flush privileges;
```

4. Edit my.cnf file

```
$ mysql --help | grep my.cnf
Output:
/etc/my.cnf,/etc/mysql/my.cnf

# edit my.cnf
$ sudo vim /etc/mysql/my.cnf

[mysqld]
max_allowed_packet=500M
sql_mode=ANSI
explicit_defaults_for_timestamp=true
```

5. restart MySQL

```
$ sudo service mysql restart
```

6. Check MySQL active(running)

```
$ sudo service mysql status
```

STEP 3 :INSTALL AIRFLOW

1. Install pip3 and Dependencies packages

```
$ sudo apt install python3-pip
$ sudo apt-get install python3-venv
$ sudo pip3 install --upgrade virtualenv
```

2. Set Up Airflow Default Home

```
$ mkdir -p ~/airflow
$ cd airflow
$ python3 -m venv lib
$ source lib/bin/activate

# activate your virtualenv terminal
(lib)$ pip3 install --upgrade setuptools pip

# if Cache entry deserialization failed, entry ignored
# run rm -rf ~/.cache/pip

(lib)$ export AIRFLOW_HOME=~/airflow
(lib)$ export SLUGIFY_USES_TEXT_UNIDECODE=yes
```

3. Installing Airflow

```
(lib)$ pip3 install apache-airflow==1.10.6
```

If error messages
ERROR: flask-appbuilder 1.13.1 has requirement marshmallow<2.20,>=2.18.0, but you'll have marshmallow 3.2.2 which is incompatible.
Solve:

```
(lib)$ pip install -U marshmallow==2.18
(lib)$ pip3 install apache-airflow==1.10.6
```

4. Install Extra Packages

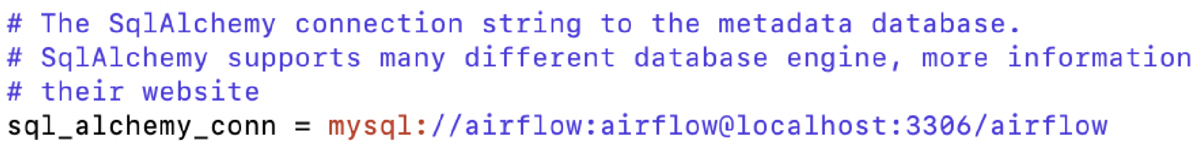
```
(lib)$ pip3 install "apache-airflow[postgres]"
(lib)$ pip3 install "apache-airflow[mysql]"
(lib)$ pip3 install "apache-airflow[celery]"
(lib)$ pip3 install 'apache-airflow[redis]'
```

5. Run command to initialize database

```
(lib)$ airflow initdb
```

6. Modify the configuration in AIRFLOW_HOME/airflow.cfg

```
(lib)$ cd ~/airflow
(lib)$ vim airflow.cfg

# airflow.cfg
#sql_alchemy_conn = sqlite:////home/ubuntu/airflow/airflow.db
sql_alchemy_conn = mysql://airflow:airflow@localhost:3306/airflow
```



```
# The SqlAlchemy connection string to the metadata database.
# SqlAlchemy supports many different database engine, more information
# their website
sql_alchemy_conn = mysql://airflow:airflow@localhost:3306/airflow
```

format:mysql://帳號:密码@ip:port/db

7. Restart initialize database

```
(lib)$ airflow initdb
```

8. test airflow start

```
(lib)$ airflow webserver
```

http://ip:8080/

9. Create dags (Workflow) folder

```
$ cd ~/airflow
$ mkdir dags
```

STEP4 :INSTALL REDIS

1. Run the apt commands below. (Open another terminal)

```
$ sudo apt update
$ sudo apt upgrade
```

2. Install the Redis-server package from the official Ubuntu repository using the apt command below.

```
$ sudo apt install redis-server
```

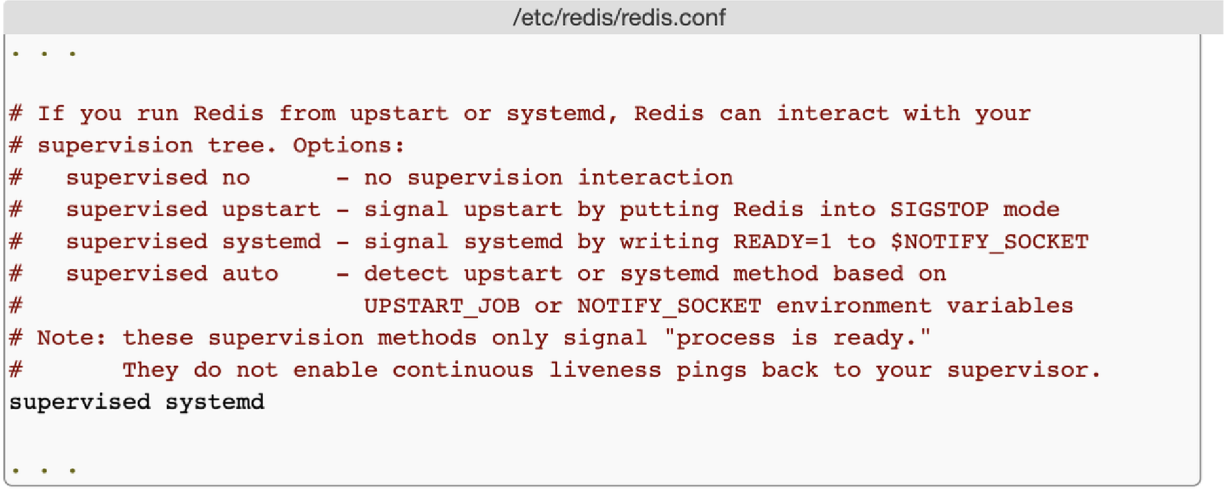
3. Go to the '/etc/redis' directory and edit the configuration file 'redis.conf' using vim editor.

```
$ cd /etc/redis/
$ sudo vim redis.conf
```

4. Change the 'bind' address with the localhost IP address for this example.

```
bind 127.0.0.1 ::1
```

4. We need to set up how the redis service will run on the server. Since we're using the Ubuntu server and systemd, so we need to change the 'supervised' line configuration to 'systemd'.



```
. . .

# If you run Redis from upstart or systemd, Redis can interact with your
# supervision tree. Options:
#   supervised no      - no supervision interaction
#   supervised upstart - signal upstart by putting Redis into SIGSTOP mode
#   supervised systemd - signal systemd by writing READY=1 to $NOTIFY_SOCKET
#   supervised auto    - detect upstart or systemd method based on
#                        UPSTART_JOB or NOTIFY_SOCKET environment variables
# Note: these supervision methods only signal "process is ready."
#       They do not enable continuous liveness pings back to your supervisor.
supervised systemd

. . .
```

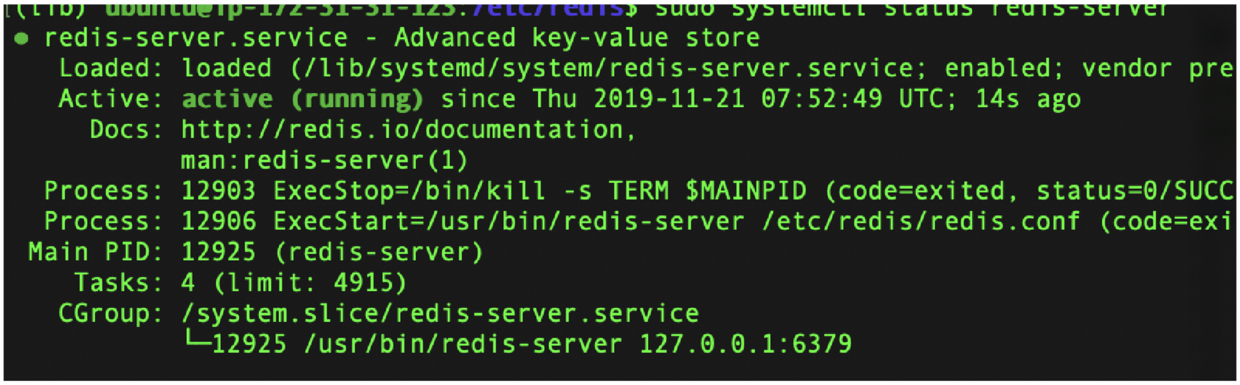
```
#supervised no
supervised systemd
```

5. Restart the redis service

```
$ sudo systemctl restart redis-server
```

6. make sure there is no error and then check its status.

```
$ sudo systemctl status redis-server
```



```
(lib) ubuntu@ip-172-31-31-123:/etc/redis$ sudo systemctl status redis-server
● redis-server.service - Advanced key-value store
   Loaded: loaded (/lib/systemd/system/redis-server.service; enabled; vendor pre
   Active: active (running) since Thu 2019-11-21 07:52:49 UTC; 14s ago
     Docs: http://redis.io/documentation,
           man:redis-server(1)
  Process: 12903 ExecStop=/bin/kill -s TERM $MAINPID (code=exited, status=0/SUCC
  Process: 12906 ExecStart=/usr/bin/redis-server /etc/redis/redis.conf (code=exi
 Main PID: 12925 (redis-server)
    Tasks: 4 (limit: 4915)
   CGroup: /system.slice/redis-server.service
           └─12925 /usr/bin/redis-server 127.0.0.1:6379
```

SEPT5 :SETTING CELERYEXECUTOR

1. For the Redis support you have to install additional dependencies. (On activate your virtualenv terminal)

   ```
   (lib)$ pip3 install celery

   # if meassages Cache entry deserialization failed, entry ignored run below comman
   run rm -rf ~/.cache/pip

   (lib)$ pip3 install redis
   (lib)$ pip3 install "celery[redis]"
   ```

2. Modify the configuration in AIRFLOW_HOME/airflow.cfg

   ```
   (lib)$ cd ~/airflow
   (lib)$ vim airflow.cfg

   # airflow.cfg
   # Setting Executor
   executor = CeleryExecutor

   # Setting broker url
   broker_url = redis://127.0.0.1:6379/1

   # result_backend
   result_backend = db+mysql://airflow:airflow@localhost:3306/airflow
   ```

detail :http://docs.celeryproject.org/en/latest/getting-started/brokers/redis.html

STEP 6 :SETTING SMTP

Modify the configuration in AIRFLOW_HOME/airflow.cfg

```
(lib)$ cd ~/airflow
(lib)$ vim airflow.cfg
```

```
# airflow.cfg
[smtp]
smtp_host = smtp.gmail.com

# Uncomment and set the user/pass settings if you want to use SMTP AUTH
smtp_user = tdcetl@gmail.com
smtp_password = vetypicqxfvxbmrk
smtp_port = 465
smtp_mail_from = tdcetl@gmail.com
```

STEP 7: START AILFLOW

1. Restart initialize database (On activate your virtualenv terminal)

```
(lib)$ airflow initdb
```

2. Open another terminal & activate your virtualenv

```
(lib)$ airflow worker
```

Your worker should be listening to redis and shouldn't see any errors

3. Then open another terminal + activate your virtualenv and launch the scheduler:

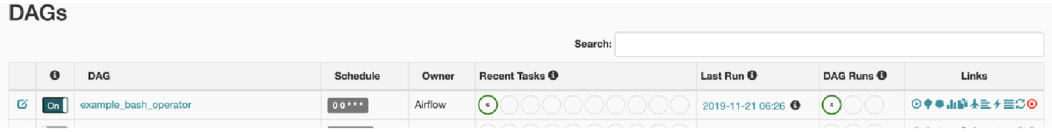
```
$ cd ~/airflow
$ source lib/bin/activate

(lib)$ airflow scheduler
```

4. Then open another terminal + activate your virtualenv and launch the scheduler:

```
$ cd ~/airflow
$ source lib/bin/activate
(lib)$ airflow webserver -p 8080
```

URL:http://ip:8080/

5. Success:You should be able to activate the dags a nd see the task running :



DAGs

Search:

		DAG	Schedule	Owner	Recent Tasks	Last Run	DAG Runs	Links
	On	example_bash_operator	0 0 * * *	Airflow	6	2019-11-21 06:26	4	

6. Airflwo Background Process

```
$ cd ~/airflow
$ source lib/bin/activate

(lib)$ nohup airflow scheduler
(lib)$ nohup airflow webserver
(lib)$ nohup airflow worker
```

https://blog.gtwang.org/linux/linux-nohup-command-tutorial/

STEP 8: RUNNING AIRFLOW SCHEDULER AS A DAEMON PROCESS

https://medium.com/@shahbaz.ali03/run-apache-airflow-as-a-service-on-ubuntu-18-04-server-b637c03f4722
https://cloud.tencent.com/developer/article/1035943
https://stackoverflow.com/questions/39073443/how-do-i-restart-airflow-webserver
https://medium.com/@shahbaz.ali03/run-apache-airflow-as-a-service-on-ubuntu-18-04-server-b637c03f4722

Illustrate importation Configuration

1. clean database

```
(lib)$ airflow resetdb
```

2. shut dowm airflwo

```
ps -ef|grep -Ei '(airflow-webserver)'| grep master | awk '{print $2}'|xargs -i kill {}
```

每次修改完airflow.cfg都要重啟動airflow
每次新增DAG也需要重啟airflow，不然找不到新的DAG

3. FAQ

https://airflow.apache.org/faq.html

4. 修改時區

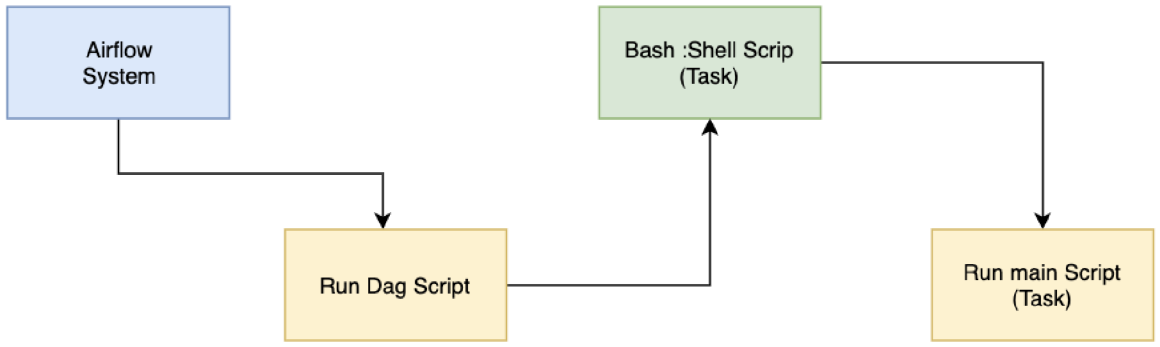
https://blog.csdn.net/Crazy_Hope/article/details/83688986
https://juejin.im/post/5d02a5bbe51d45778f076d26

5. 參數說明

https://cloud.tencent.com/developer/article/1035943

6. Celery 說明

https://zhuanlan.zhihu.com/p/22304455
https://www.sicara.ai/blog/2019-04-08-apache-airflow-celery-workers
https://www.astronomer.io/guides/airflow-executors-explained/
https://stlong0521.github.io/20161023%20-%20Airflow.html


Airflow System
Run Dag Script
Bash :Shell Scrip (Task)
Run main Script (Task)