Airflow in Docker

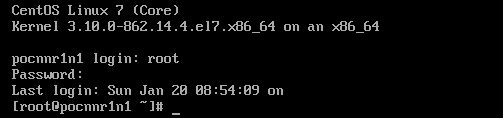
# Environment

Working host: pocnnr1n1

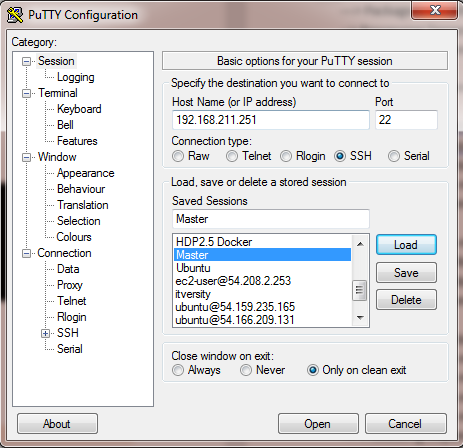
## 1.1 Start the master from VM console

root, X@C

rxie, X@C



## 1.2 SSH to the master node



login as: rxie

rxie@192.168.211.251's password:

Last login: Fri Dec 7 09:59:40 2018 from 192.168.211.1

[rxie@pocnnr1n1 ~]$

# Docker installation[[1]](#footnote-1)

## 2.1 install required packages

[root@pocnnr1n1 ~]# yum install -y yum-utils \

> device-mapper-persistent-data \

> lvm2

Loaded plugins: fastestmirror, langpacks

Repodata is over 2 weeks old. Install yum-cron? Or run: yum makecache fast

base | 3.6 kB 00:00

cloudera-manager | 951 B 00:00

extras | 3.4 kB 00:00

updates | 3.4 kB 00:00

(1/4): base/7/x86\_64/group\_gz | 166 kB 00:00

(2/4): extras/7/x86\_64/primary\_db | 205 kB 00:01

(3/4): updates/7/x86\_64/primary\_db

(4/4): base/7/x86\_64/primary\_db

Determining fastest mirrors

\* base: centos.mirror.vexxhost.com

\* extras: centos.mirror.netelligent.ca

\* updates: centos.mirror.netelligent.ca

Resolving Dependencies

--> Running transaction check

---> Package device-mapper-persistent-data.x86\_64 0:0.6.3-1.el7 will be updated

---> Package device-mapper-persistent-data.x86\_64 0:0.7.3-3.el7 will be an update

---> Package lvm2.x86\_64 7:2.02.166-1.el7 will be updated

---> Package lvm2.x86\_64 7:2.02.177-4.el7 will be an update

--> Processing Dependency: lvm2-libs = 7:2.02.177-4.el7 for package: 7:lvm2-2.02.177-4.el7.x86\_64

--> Processing Dependency: libdevmapper.so.1.02(DM\_1\_02\_141)(64bit) for package: 7:lvm2-2.02.177-4.el7.x86

--> Processing Dependency: libdevmapper.so.1.02(DM\_1\_02\_138)(64bit) for package: 7:lvm2-2.02.177-4.el7.x86

---> Package yum-utils.noarch 0:1.1.31-40.el7 will be updated

---> Package yum-utils.noarch 0:1.1.31-46.el7\_5 will be an update

--> Running transaction check

---> Package device-mapper-libs.x86\_64 7:1.02.135-1.el7 will be updated

--> Processing Dependency: device-mapper-libs = 7:1.02.135-1.el7 for package: 7:device-mapper-1.02.135-1.e

---> Package device-mapper-libs.x86\_64 7:1.02.146-4.el7 will be an update

---> Package lvm2-libs.x86\_64 7:2.02.166-1.el7 will be updated

---> Package lvm2-libs.x86\_64 7:2.02.177-4.el7 will be an update

--> Processing Dependency: device-mapper-event = 7:1.02.146-4.el7 for package: 7:lvm2-libs-2.02.177-4.el7.

--> Running transaction check

---> Package device-mapper.x86\_64 7:1.02.135-1.el7 will be updated

---> Package device-mapper.x86\_64 7:1.02.146-4.el7 will be an update

---> Package device-mapper-event.x86\_64 7:1.02.135-1.el7 will be updated

---> Package device-mapper-event.x86\_64 7:1.02.146-4.el7 will be an update

--> Processing Dependency: device-mapper-event-libs = 7:1.02.146-4.el7 for package: 7:device-mapper-event-

--> Running transaction check

---> Package device-mapper-event-libs.x86\_64 7:1.02.135-1.el7 will be updated

---> Package device-mapper-event-libs.x86\_64 7:1.02.146-4.el7 will be an update

--> Finished Dependency Resolution

Dependencies Resolved

=====================================================================================

Package Arch Version Rep

=====================================================================================

Updating:

device-mapper-persistent-data x86\_64 0.7.3-3.el7 bas

lvm2 x86\_64 7:2.02.177-4.el7 bas

yum-utils noarch 1.1.31-46.el7\_5 upd

Updating for dependencies:

device-mapper x86\_64 7:1.02.146-4.el7 bas

device-mapper-event x86\_64 7:1.02.146-4.el7 bas

device-mapper-event-libs x86\_64 7:1.02.146-4.el7 bas

device-mapper-libs x86\_64 7:1.02.146-4.el7 bas

lvm2-libs x86\_64 7:2.02.177-4.el7 bas

Transaction Summary

==================================================================================

Upgrade 3 Packages (+5 Dependent packages)

Total download size: 3.8 M

Downloading packages:

No Presto metadata available for base

Not downloading deltainfo for updates, MD is 679 k and rpms are 120 k

(1/8): device-mapper-1.02.146-4.el7.x86\_64.rpm

(2/8): device-mapper-libs-1.02.146-4.el7.x86\_64.rpm

(3/8): device-mapper-event-1.02.146-4.el7.x86\_64.rpm

(4/8): device-mapper-event-libs-1.02.146-4.el7.x86\_64.rpm

(5/8): device-mapper-persistent-data-0.7.3-3.el7.x86\_64.rpm

(6/8): yum-utils-1.1.31-46.el7\_5.noarch.rpm

(7/8): lvm2-libs-2.02.177-4.el7.x86\_64.rpm

(8/8): lvm2-2.02.177-4.el7.x86\_64.rpm

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

Total 543 k

Running transaction check

Running transaction test

Transaction test succeeded

Running transaction

Updating : 7:device-mapper-libs-1.02.146-4.el7.x86\_64

Updating : 7:device-mapper-1.02.146-4.el7.x86\_64

Updating : 7:device-mapper-event-libs-1.02.146-4.el7.x86\_64

Updating : 7:device-mapper-event-1.02.146-4.el7.x86\_64

Updating : 7:lvm2-libs-2.02.177-4.el7.x86\_64

Updating : device-mapper-persistent-data-0.7.3-3.el7.x86\_64

Updating : 7:lvm2-2.02.177-4.el7.x86\_64

Updating : yum-utils-1.1.31-46.el7\_5.noarch

Cleanup : yum-utils-1.1.31-40.el7.noarch

Cleanup : 7:lvm2-2.02.166-1.el7.x86\_64

Cleanup : 7:lvm2-libs-2.02.166-1.el7.x86\_64

Cleanup : 7:device-mapper-event-1.02.135-1.el7.x86\_64

Cleanup : 7:device-mapper-event-libs-1.02.135-1.el7.x86\_64

Cleanup : 7:device-mapper-libs-1.02.135-1.el7.x86\_64

Cleanup : 7:device-mapper-1.02.135-1.el7.x86\_64

Cleanup : device-mapper-persistent-data-0.6.3-1.el7.x86\_64

Verifying : device-mapper-persistent-data-0.7.3-3.el7.x86\_64

Verifying : yum-utils-1.1.31-46.el7\_5.noarch

Verifying : 7:device-mapper-1.02.146-4.el7.x86\_64

Verifying : 7:device-mapper-event-libs-1.02.146-4.el7.x86\_64

Verifying : 7:device-mapper-event-1.02.146-4.el7.x86\_64

Verifying : 7:lvm2-libs-2.02.177-4.el7.x86\_64

Verifying : 7:lvm2-2.02.177-4.el7.x86\_64

Verifying : 7:device-mapper-libs-1.02.146-4.el7.x86\_64

Verifying : 7:device-mapper-event-libs-1.02.135-1.el7.x86\_64

Verifying : 7:device-mapper-1.02.135-1.el7.x86\_64

Verifying : 7:lvm2-2.02.166-1.el7.x86\_64

Verifying : yum-utils-1.1.31-40.el7.noarch

Verifying : device-mapper-persistent-data-0.6.3-1.el7.x86\_64

Verifying : 7:device-mapper-libs-1.02.135-1.el7.x86\_64

Verifying : 7:device-mapper-event-1.02.135-1.el7.x86\_64

Verifying : 7:lvm2-libs-2.02.166-1.el7.x86\_64

Updated:

device-mapper-persistent-data.x86\_64 0:0.7.3-3.el7 lvm2.x86\_64 7:2.02.177-4.el7 yum-utils.noar

Dependency Updated:

device-mapper.x86\_64 7:1.02.146-4.el7 device-mapper-event.x86\_64 7:1.02.14

device-mapper-event-libs.x86\_64 7:1.02.146-4.el7 device-mapper-libs.x86\_64 7:1.02.146

lvm2-libs.x86\_64 7:2.02.177-4.el7

Complete!

## 2.2 Setup the stable repository

[root@pocnnr1n1 ~]# yum-config-manager \

> --add-repo \

> https://download.docker.com/linux/centos/docker-ce.repo

Loaded plugins: fastestmirror, langpacks

adding repo from: https://download.docker.com/linux/centos/docker-ce.repo

grabbing file https://download.docker.com/linux/centos/docker-ce.repo to /etc/yum.repos.d/docker-ce.repo

repo saved to /etc/yum.repos.d/docker-ce.repo

## 2.3 Enable edge and test repository

[root@pocnnr1n1 ~]# yum-config-manager --enable docker-ce-edge

Loaded plugins: fastestmirror, langpacks

================================ repo: docker-ce-edge ============================

[docker-ce-edge]

async = True

bandwidth = 0

base\_persistdir = /var/lib/yum/repos/x86\_64/7

baseurl = https://download.docker.com/linux/centos/7/x86\_64/edge

cache = 0

cachedir = /var/cache/yum/x86\_64/7/docker-ce-edge

check\_config\_file\_age = True

compare\_providers\_priority = 80

cost = 1000

deltarpm\_metadata\_percentage = 100

deltarpm\_percentage =

enabled = 1

enablegroups = True

exclude =

failovermethod = priority

ftp\_disable\_epsv = False

gpgcadir = /var/lib/yum/repos/x86\_64/7/docker-ce-edge/gpgcadir

gpgcakey =

gpgcheck = True

gpgdir = /var/lib/yum/repos/x86\_64/7/docker-ce-edge/gpgdir

gpgkey = https://download.docker.com/linux/centos/gpg

hdrdir = /var/cache/yum/x86\_64/7/docker-ce-edge/headers

http\_caching = all

includepkgs =

ip\_resolve =

keepalive = True

keepcache = False

mddownloadpolicy = sqlite

mdpolicy = group:small

mediaid =

metadata\_expire = 21600

metadata\_expire\_filter = read-only:present

metalink =

minrate = 0

mirrorlist =

mirrorlist\_expire = 86400

name = Docker CE Edge - x86\_64

old\_base\_cache\_dir =

password =

persistdir = /var/lib/yum/repos/x86\_64/7/docker-ce-edge

pkgdir = /var/cache/yum/x86\_64/7/docker-ce-edge/packages

proxy = False

proxy\_dict =

proxy\_password =

proxy\_username =

repo\_gpgcheck = False

retries = 10

skip\_if\_unavailable = False

ssl\_check\_cert\_permissions = True

sslcacert =

sslclientcert =

sslclientkey =

sslverify = True

throttle = 0

timeout = 30.0

ui\_id = docker-ce-edge/x86\_64

ui\_repoid\_vars = releasever,

basearch

username =

[root@pocnnr1n1 ~]# yum-config-manager --enable docker-ce-test

Loaded plugins: fastestmirror, langpacks

============================== repo: docker-ce-test =============================

[docker-ce-test]

async = True

bandwidth = 0

base\_persistdir = /var/lib/yum/repos/x86\_64/7

baseurl = https://download.docker.com/linux/centos/7/x86\_64/test

cache = 0

cachedir = /var/cache/yum/x86\_64/7/docker-ce-test

check\_config\_file\_age = True

compare\_providers\_priority = 80

cost = 1000

deltarpm\_metadata\_percentage = 100

deltarpm\_percentage =

enabled = 1

enablegroups = True

exclude =

failovermethod = priority

ftp\_disable\_epsv = False

gpgcadir = /var/lib/yum/repos/x86\_64/7/docker-ce-test/gpgcadir

gpgcakey =

gpgcheck = True

gpgdir = /var/lib/yum/repos/x86\_64/7/docker-ce-test/gpgdir

gpgkey = https://download.docker.com/linux/centos/gpg

hdrdir = /var/cache/yum/x86\_64/7/docker-ce-test/headers

http\_caching = all

includepkgs =

ip\_resolve =

keepalive = True

keepcache = False

mddownloadpolicy = sqlite

mdpolicy = group:small

mediaid =

metadata\_expire = 21600

metadata\_expire\_filter = read-only:present

metalink =

minrate = 0

mirrorlist =

mirrorlist\_expire = 86400

name = Docker CE Test - x86\_64

old\_base\_cache\_dir =

password =

persistdir = /var/lib/yum/repos/x86\_64/7/docker-ce-test

pkgdir = /var/cache/yum/x86\_64/7/docker-ce-test/packages

proxy = False

proxy\_dict =

proxy\_password =

proxy\_username =

repo\_gpgcheck = False

retries = 10

skip\_if\_unavailable = False

ssl\_check\_cert\_permissions = True

sslcacert =

sslclientcert =

sslclientkey =

sslverify = True

throttle = 0

timeout = 30.0

ui\_id = docker-ce-test/x86\_64

ui\_repoid\_vars = releasever,

basearch

username =

## 2.4. Install Docker CE

[root@pocnnr1n1 ~]# yum install docker-ce

Loaded plugins: fastestmirror, langpacks

docker-ce-edge

docker-ce-stable

docker-ce-test

(1/6): docker-ce-stable/x86\_64/updateinfo

(2/6): docker-ce-edge/x86\_64/updateinfo

(3/6): docker-ce-edge/x86\_64/primary\_db

(4/6): docker-ce-test/x86\_64/primary\_db

(5/6): docker-ce-test/x86\_64/updateinfo

(6/6): docker-ce-stable/x86\_64/primary\_db

Loading mirror speeds from cached hostfile

\* base: centos.mirror.vexxhost.com

\* extras: centos.mirror.netelligent.ca

\* updates: centos.mirror.netelligent.ca

Resolving Dependencies

--> Running transaction check

---> Package docker-ce.x86\_64 3:18.09.0-3.el7 will be installed

--> Processing Dependency: container-selinux >= 2.9 for package: 3:docker-ce-18.09.0-3.el7.x86\_64

--> Processing Dependency: containerd.io for package: 3:docker-ce-18.09.0-3.el7.x86\_64

--> Processing Dependency: docker-ce-cli for package: 3:docker-ce-18.09.0-3.el7.x86\_64

--> Running transaction check

---> Package container-selinux.noarch 2:2.68-1.el7 will be installed

--> Processing Dependency: selinux-policy-targeted >= 3.13.1-192 for package: 2:container-selinux-2.68-1.e

--> Processing Dependency: selinux-policy-base >= 3.13.1-192 for package: 2:container-selinux-2.68-1.el7.n

--> Processing Dependency: selinux-policy >= 3.13.1-192 for package: 2:container-selinux-2.68-1.el7.noarch

---> Package containerd.io.x86\_64 0:1.2.0-3.el7 will be installed

---> Package docker-ce-cli.x86\_64 1:18.09.0-3.el7 will be installed

--> Running transaction check

---> Package selinux-policy.noarch 0:3.13.1-102.el7\_3.16 will be updated

---> Package selinux-policy.noarch 0:3.13.1-192.el7\_5.6 will be an update

--> Processing Dependency: policycoreutils >= 2.5-18 for package: selinux-policy-3.13.1-192.el7\_5.6.noarch

---> Package selinux-policy-targeted.noarch 0:3.13.1-102.el7\_3.16 will be updated

---> Package selinux-policy-targeted.noarch 0:3.13.1-192.el7\_5.6 will be an update

--> Running transaction check

---> Package policycoreutils.x86\_64 0:2.5-11.el7\_3 will be updated

--> Processing Dependency: policycoreutils = 2.5-11.el7\_3 for package: policycoreutils-python-2.5-11.el7\_3

---> Package policycoreutils.x86\_64 0:2.5-22.el7 will be an update

--> Processing Dependency: libsepol >= 2.5-8 for package: policycoreutils-2.5-22.el7.x86\_64

--> Processing Dependency: libselinux-utils >= 2.5-12 for package: policycoreutils-2.5-22.el7.x86\_64

--> Running transaction check

---> Package libselinux-utils.x86\_64 0:2.5-6.el7 will be updated

---> Package libselinux-utils.x86\_64 0:2.5-12.el7 will be an update

--> Processing Dependency: libselinux(x86-64) = 2.5-12.el7 for package: libselinux-utils-2.5-12.el7.x86\_64

---> Package libsepol.x86\_64 0:2.5-6.el7 will be updated

--> Processing Dependency: libsepol(x86-64) = 2.5-6.el7 for package: libsepol-devel-2.5-6.el7.x86\_64

---> Package libsepol.x86\_64 0:2.5-8.1.el7 will be an update

---> Package policycoreutils-python.x86\_64 0:2.5-11.el7\_3 will be updated

---> Package policycoreutils-python.x86\_64 0:2.5-22.el7 will be an update

--> Processing Dependency: setools-libs >= 3.3.8-2 for package: policycoreutils-python-2.5-22.el7.x86\_64

--> Processing Dependency: libsemanage-python >= 2.5-9 for package: policycoreutils-python-2.5-22.el7.x86\_

--> Running transaction check

---> Package libselinux.x86\_64 0:2.5-6.el7 will be updated

--> Processing Dependency: libselinux(x86-64) = 2.5-6.el7 for package: libselinux-devel-2.5-6.el7.x86\_64

--> Processing Dependency: libselinux(x86-64) = 2.5-6.el7 for package: libselinux-python-2.5-6.el7.x86\_64

---> Package libselinux.x86\_64 0:2.5-12.el7 will be an update

---> Package libsemanage-python.x86\_64 0:2.5-5.1.el7\_3 will be updated

---> Package libsemanage-python.x86\_64 0:2.5-11.el7 will be an update

--> Processing Dependency: libsemanage = 2.5-11.el7 for package: libsemanage-python-2.5-11.el7.x86\_64

---> Package libsepol-devel.x86\_64 0:2.5-6.el7 will be updated

---> Package libsepol-devel.x86\_64 0:2.5-8.1.el7 will be an update

---> Package setools-libs.x86\_64 0:3.3.8-1.1.el7 will be updated

---> Package setools-libs.x86\_64 0:3.3.8-2.el7 will be an update

--> Running transaction check

---> Package libselinux-devel.x86\_64 0:2.5-6.el7 will be updated

---> Package libselinux-devel.x86\_64 0:2.5-12.el7 will be an update

---> Package libselinux-python.x86\_64 0:2.5-6.el7 will be updated

---> Package libselinux-python.x86\_64 0:2.5-12.el7 will be an update

---> Package libsemanage.x86\_64 0:2.5-5.1.el7\_3 will be updated

---> Package libsemanage.x86\_64 0:2.5-11.el7 will be an update

--> Finished Dependency Resolution

Dependencies Resolved

==========================================================================

Package Arch Version Repositor

==========================================================================

Installing:

docker-ce x86\_64 3:18.09.0-3.el7 docker-ce

Installing for dependencies:

container-selinux noarch 2:2.68-1.el7 extras

containerd.io x86\_64 1.2.0-3.el7 docker-ce

docker-ce-cli x86\_64 1:18.09.0-3.el7 docker-ce

Updating for dependencies:

libselinux x86\_64 2.5-12.el7 base

libselinux-devel x86\_64 2.5-12.el7 base

libselinux-python x86\_64 2.5-12.el7 base

libselinux-utils x86\_64 2.5-12.el7 base

libsemanage x86\_64 2.5-11.el7 base

libsemanage-python x86\_64 2.5-11.el7 base

libsepol x86\_64 2.5-8.1.el7 base

libsepol-devel x86\_64 2.5-8.1.el7 base

policycoreutils x86\_64 2.5-22.el7 base

policycoreutils-python x86\_64 2.5-22.el7 base

selinux-policy noarch 3.13.1-192.el7\_5.6 updates

selinux-policy-targeted noarch 3.13.1-192.el7\_5.6 updates

setools-libs x86\_64 3.3.8-2.el7 base

Transaction Summary

=====================================================================================

Install 1 Package (+ 3 Dependent packages)

Upgrade ( 13 Dependent packages)

Total download size: 65 M

Is this ok [y/d/N]: y

Downloading packages:

No Presto metadata available for base

updates/7/x86\_64/prestodelta

(1/17): container-selinux-2.68-1.el7.noarch.rpm

(17/17): docker-ce-cli-18.09.0-3.el7.x86\_64.rpm

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

Total 1.1 M

Retrieving key from https://download.docker.com/linux/centos/gpg

Importing GPG key 0x621E9F35:

Userid : "Docker Release (CE rpm) <docker@docker.com>"

Fingerprint: 060a 61c5 1b55 8a7f 742b 77aa c52f eb6b 621e 9f35

From : https://download.docker.com/linux/centos/gpg

Is this ok [y/N]: y

Running transaction check

Running transaction test

Transaction test succeeded

Running transaction

Updating : libsepol-2.5-8.1.el7.x86\_64

Verifying : setools-libs-3.3.8-1.1.el7.x86\_64 30/30

Installed:

docker-ce.x86\_64 3:18.09.0-3.el7

Dependency Installed:

container-selinux.noarch 2:2.68-1.el7 containerd.io.x86\_64 0:1.2.0-3.el7

docker-ce-cli.x86\_64 1:18.09.0-3.el7

Dependency Updated:

libselinux.x86\_64 0:2.5-12.el7 libselinux-devel.x86\_64 0:2.5-12.el7

setools-libs.x86\_64 0:3.3.8-2.el7

Complete!

# 3. Access Docker

## 3.1 Start docker

[root@pocnnr1n1 ~]# systemctl start docker

## 3.2 Test docker

[root@pocnnr1n1 ~]# docker run hello-world

Unable to find image 'hello-world:latest' locally

latest: Pulling from library/hello-world

d1725b59e92d: Pull complete

Digest: sha256:0add3ace90ecb4adbf7777e9aacf18357296e799f81cabc9fde470971e499788

Status: Downloaded newer image for hello-world:latest

Hello from Docker!

This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:

1. The Docker client contacted the Docker daemon.

2. The Docker daemon pulled the "hello-world" image from the Docker Hub.

(amd64)

3. The Docker daemon created a new container from that image which runs the

executable that produces the output you are currently reading.

4. The Docker daemon streamed that output to the Docker client, which sent it

to your terminal.

To try something more ambitious, you can run an Ubuntu container with:

$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:

https://hub.docker.com/

For more examples and ideas, visit:

https://docs.docker.com/get-started/

[root@pocnnr1n1 ~]#

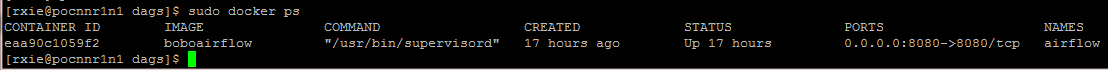
## 3.3 docker and host

[rxie@pocnnr1n1 dags]$ sudo docker ps

[sudo] password for rxie:

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

eaa90c1059f2 boboairflow "/usr/bin/supervisord" 13 hours ago Up 13 hours 0.0.0.0:8080->8080/tcp airflow



### Run docker image[[2]](#footnote-2)

sudo docker run -d --name airflow -p 8080:8080 boboairflow

### Stop docker

sudo docker stop eaa90c1059f2

### Start docker

sudo docker start eaa90c1059f2

## 3.4 Transfer files between host and docker

Copy file from host to docker:

[rxie@pocnnr1n1 dags]$ sudo docker cp test\_sql.py eaa90c1059f2:/app/airflow/dags

Copy file from docker to host:

[rxie@pocnnr1n1 dags]$ sudo docker cp eaa90c1059f2:/app/airflow/dags/test\_sql.py test\_sql\_docker.py

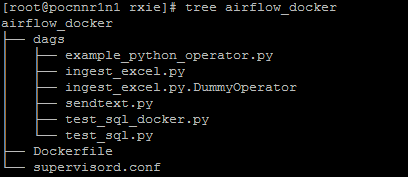
## 3.5 Accessing docker

sudo docker exec -it airflow /bin/bash

# 4. Airflow Image Build

## 4.1 Package

Location: /home/rxie/airflow\_docker



[root@pocnnr1n1 rxie]# tree airflow\_docker

airflow\_docker

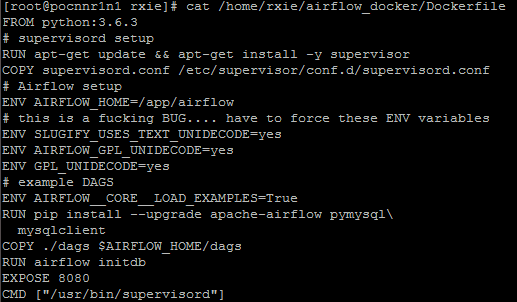
├── dags

│   └── test\_sql.py

├── Dockerfile

└── supervisord.conf

## 4.2 Docker file



[root@pocnnr1n1 dags]# cat ../Dockerfile

FROM python:3.6.3

# supervisord setup

RUN apt-get update && apt-get install -y supervisor

COPY supervisord.conf /etc/supervisor/conf.d/supervisord.conf

# Airflow setup

ENV AIRFLOW\_HOME=/app/airflow

# There are special instructions from Airflow itself, some Env variables must be manually set

ENV SLUGIFY\_USES\_TEXT\_UNIDECODE=yes

ENV AIRFLOW\_GPL\_UNIDECODE=yes

ENV GPL\_UNIDECODE=yes

# example DAGS: in a real prod, it should be disabled

ENV AIRFLOW\_\_CORE\_\_LOAD\_EXAMPLES=True

#install airflow with pip, options: pymysql and mysqlclient

RUN pip install --upgrade apache-airflow pymysql\

Mysqlclient

#copy all dags (only one here: test\_sql.py) under ./dags to airflow in docker image:

COPY ./dags $AIRFLOW\_HOME/dags

#initialize airflow database

RUN airflow initdb

#using port 8080

EXPOSE 8080

CMD ["/usr/bin/supervisord"]

# 5. Dockerized airflow

Airflow\_docker file is saved at: C:\RXIE\Learning\Docker\airflow\_docker (provided by Huang Zhongbin via email: Nov 22, 2018, 9:12 PM)

#transfer the airflow\_docker.tar.gz file to host



and unzip the docker file

tar -xf airflow\_docker.tar.gz -C ~/downloads

#enter the docker folder

cd downloads/airflow\_docker

#build the airflow image with the name boboairflow

sudo docker build -t boboairflow .

Note there is a dot in the end of the command.

[root@pocnnr1n1 airflow\_docker]# docker build - < Dockerfile

Sending build context to Docker daemon 2.56kB

Step 1/13 : FROM python:3.6.3

---> a8f7167de312

Step 2/13 : RUN apt-get update && apt-get install -y supervisor

---> Using cache

---> 7a1801495217

Step 3/13 : COPY supervisord.conf /etc/supervisor/conf.d/supervisord.conf

COPY failed: stat /var/lib/docker/tmp/docker-builder280432540/supervisord.conf: no such file or directory

Here is a bug from the Dockerfile or from docker itself:

If there is no existing /etc/supervisor/conf.d, presumably the step 3 should fail, but instead it just ignores the non-existing folder and move forward without error message.

Fix: manually create the folder and re-run the docker build command

## 5.1 Details of build

**Step 1/13 : FROM python:3.6.3**

---> a8f7167de312

**Step 2/13 : RUN apt-get update && apt-get install -y supervisor**

---> Using cache

---> 7a1801495217

**Step 3/13 : COPY supervisord.conf /etc/supervisor/conf.d/supervisord.conf**

---> Using cache

---> 7abe753f90a2

**Step 4/13 : ENV AIRFLOW\_HOME=/app/airflow**

---> Using cache

---> 534e8d02f9fb

**Step 5/13 : ENV SLUGIFY\_USES\_TEXT\_UNIDECODE=yes**

---> Using cache

---> cecf84f2020c

**Step 6/13 : ENV AIRFLOW\_GPL\_UNIDECODE=yes**

---> Using cache

---> 458be3291f46

**Step 7/13 : ENV GPL\_UNIDECODE=yes**

---> Using cache

---> 6f668880ebb5

**Step 8/13 : ENV AIRFLOW\_\_CORE\_\_LOAD\_EXAMPLES=True**

---> Using cache

---> d9f7919ce18f

**Step 9/13 : RUN pip install --upgrade apache-airflow pymysql mysqlclient**

---> Using cache

---> 18ebfac5f627

**Step 10/13 : COPY ./dags $AIRFLOW\_HOME/dags**

---> Using cache

---> 5d7a30050f41

**Step 11/13 : RUN airflow initdb**

---> Using cache

---> 7d0b324f439f

**Step 12/13 : EXPOSE 8080**

---> Using cache

---> c4e7b82995e1

**Step 13/13 : CMD ["/usr/bin/supervisord"]**

---> Using cache

---> 5115cc30b134

Successfully built 5115cc30b134

Successfully tagged boboairflow:latest

#run the image in the name of airflow at port 8080

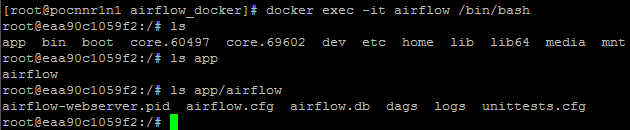
## 5.2 Verfication of the build

sudo docker run -d --name airflow -p 8080:8080 boboairflow

#access docker

sudo docker exec -it airflow /bin/bash

#now the airflow is under app folder



## 5.3 Airflow File Structure

tree /app

/app

└── airflow

├── airflow-webserver.pid

├── airflow.cfg

├── airflow.db

├── dags

│   ├── \_\_pycache\_\_

│   │   └── test\_sql.cpython-36.pyc

│   └── test\_sql.py

├── logs

│   ├── scheduler

│   │   ├── 2018-11-23

│   │   │   └── test\_sql.py.log

│   │   └── latest -> /app/airflow/logs/scheduler/2018-11-23

│   └── tutorial

│   ├── print\_date

│   │   ├── 2018-11-21T00:00:00+00:00

│   │   │   └── 1.log

│   │   └── 2018-11-22T00:00:00+00:00

│   │   └── 1.log

│   ├── sleep

│   │   ├── 2018-11-21T00:00:00+00:00

│   │   │   └── 1.log

│   │   └── 2018-11-22T00:00:00+00:00

│   │   └── 1.log

│   └── templated

│   ├── 2018-11-21T00:00:00+00:00

│   │   └── 1.log

│   └── 2018-11-22T00:00:00+00:00

│   └── 1.log

└── unittests.cfg

## 5.4 Location of airflow

root@eaa90c1059f2:/app/airflow# echo $PATH

/usr/local/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin

root@eaa90c1059f2:/app/airflow# which airflow

/usr/local/bin/airflow

root@eaa90c1059f2:/app/airflow# ls -la /usr/local/bin/

total 216

drwxrwsr-x 1 root staff 75 Nov 25 20:41 .

drwxrwsr-x 1 root staff 28 Dec 9 2017 ..

lrwxrwxrwx 1 root staff 8 Dec 12 2017 2to3 -> 2to3-3.6

-rwxr-xr-x 1 root staff 101 Dec 12 2017 2to3-3.6

drwxr-sr-x 2 root staff 4096 Nov 22 21:27 \_\_pycache\_\_

-rwxr-xr-x 1 root staff 1264 Nov 22 21:27 airflow

# 6. DAG practice

## 6.1 Create database and table in MySQL/MariaDB

MariaDB [hue]> create database airflow;

Query OK, 1 row affected (0.00 sec)

MariaDB [hue]> use airflow;

Database changed

MariaDB [airflow]> show tables;

Empty set (0.00 sec)

MariaDB [airflow]> create table my\_test (id int not null AUTO\_INCREMENT, name varchar(255), primary key(id));

Query OK, 0 rows affected (0.01 sec)

MariaDB [airflow]> show create table my\_test;

| Table | Create Table

| my\_test | CREATE TABLE `my\_test` (

`id` int(11) NOT NULL AUTO\_INCREMENT,

`name` varchar(255) DEFAULT NULL,

PRIMARY KEY (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1 |

1 row in set (0.01 sec)

MariaDB [airflow]> select \* From my\_test;

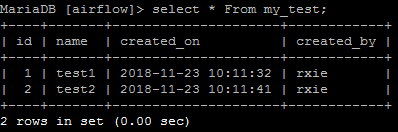
| id | name | created\_on | created\_by |

| 1 | test1 | 2018-11-23 10:11:32 | rxie |

| 2 | test2 | 2018-11-23 10:11:41 | rxie |

insert into my\_test(name, created\_on, created\_by) values('test1', CURRENT\_TIMESTAMP, 'rxie');

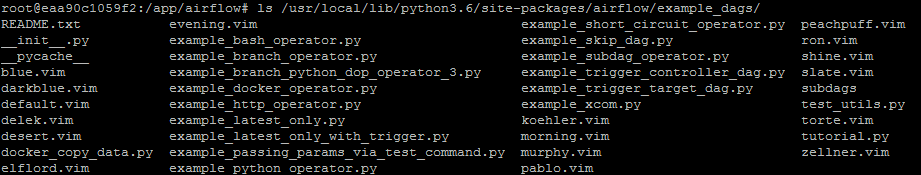
insert into my\_test(name, created\_on, created\_by) values('test2', CURRENT\_TIMESTAMP, 'rxie');



## 6.2 DAG Location

### Example DAGs

/usr/local/lib/python3.6/site-packages/airflow/example\_dags/

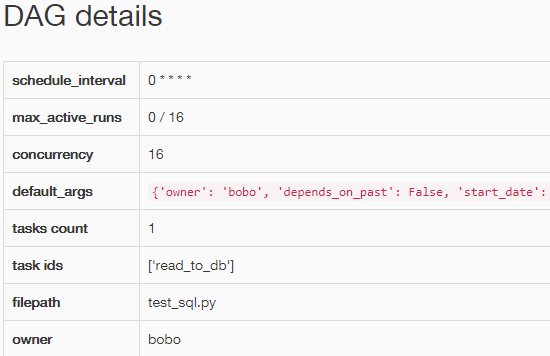


### Local example

/app/airflow/dags/test\_sql.py



### Example dag explain



[rxie@pocnnr1n1 dags]$ cat test\_sql.py

from airflow import DAG

#from airflow.operators.bash\_operator import BashOperator

from datetime import datetime, timedelta

from airflow.hooks import HttpHook, MySqlHook

from airflow.operators import PythonOperator

from airflow.models import DAG

import pymysql

# Main DAG function

def read\_to\_db():

# establish mysql connection

conn = pymysql.connect(host='localhost', port=3306, user='root', passwd='root', db='airflow')

cur = conn.cursor()

cur.execute("INSERT INTO my\_test (name, created\_on, created\_by) values ('airflow in docker', CURRENT\_TIMESTAMP, 'rxie');")

cur.close()

conn.commit()

conn.close()

# Following are defaults which can be overridden later on

default\_args = {

'owner': 'bobo',

'depends\_on\_past': False,

'start\_date': datetime(2018, 11, 21),

'email': ['xie3208080@gmail.com'],

'email\_on\_failure': True,

'email\_on\_retry': True,

'retries': 1,

'retry\_delay': timedelta(minutes=1),

}

dag = DAG('test\_sql', schedule\_interval='0 \* \* \* \* ', default\_args=default\_args)

read\_to\_db = PythonOperator(task\_id='read\_to\_db',

provide\_context=False,

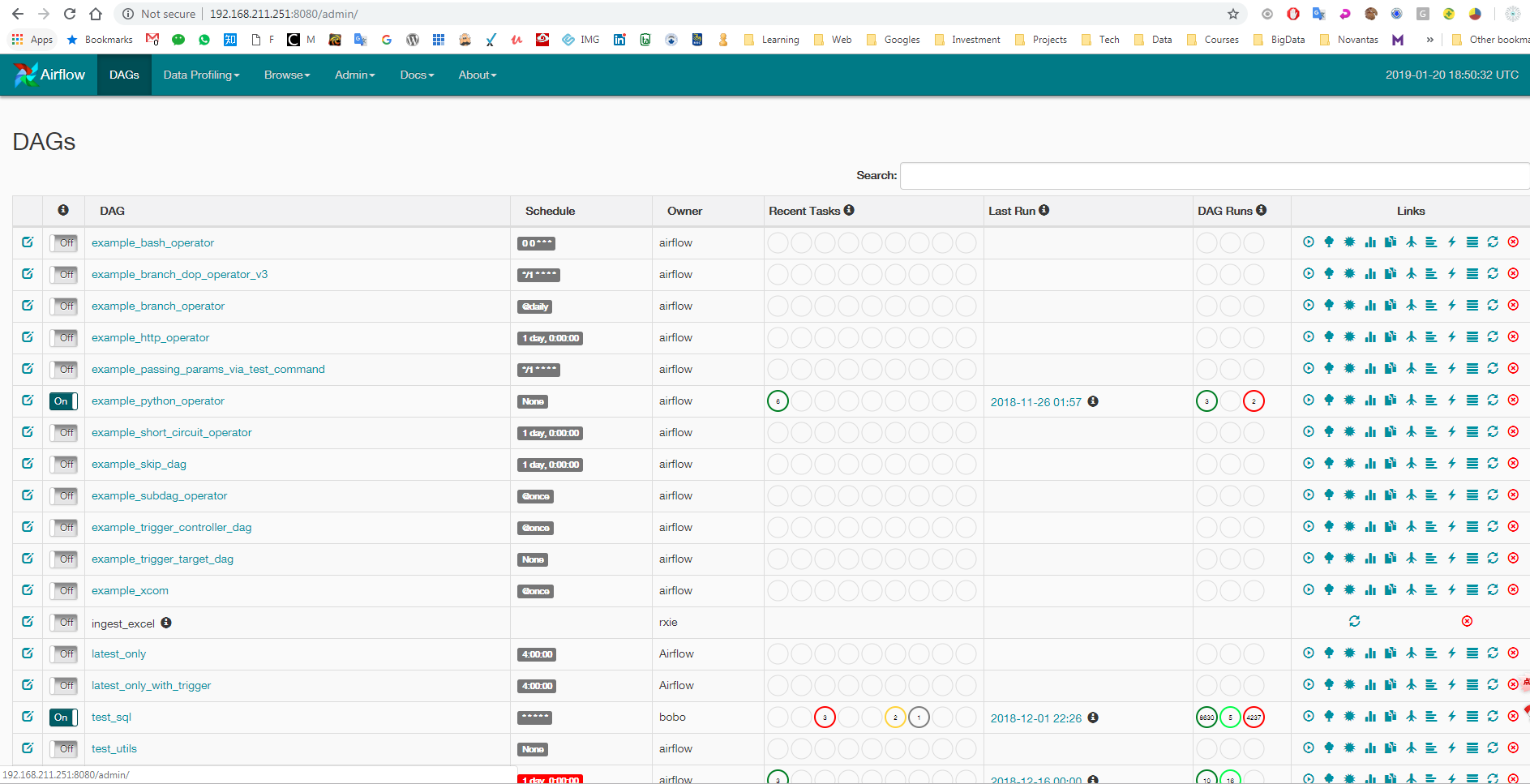
python\_callable=read\_to\_db,

dag=dag)

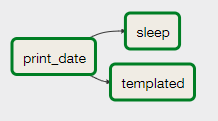
## 6.3 DAG Design Examples

### Open Airflow Website

<http://192.168.211.251:8080>



### Example 1:



t1 = BashOperator(

task\_id='print\_date',

bash\_command='date',

dag=dag,

)

t2 = BashOperator(

task\_id='sleep',

depends\_on\_past=False,

bash\_command='sleep 5',

dag=dag,

)

t3 = BashOperator(

task\_id='templated',

depends\_on\_past=False,

bash\_command=templated\_command,

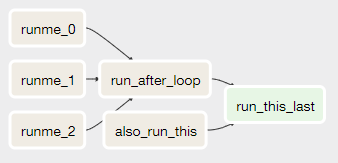
params={'my\_param': 'Parameter I passed in'},

dag=dag,

)

t1 >> [t2, t3]

### Example 2:



run\_this = BashOperator(

task\_id='run\_after\_loop',

bash\_command='echo 1',

dag=dag,

)

run\_this >> run\_this\_last

**for** i **in** range(3):

task = BashOperator(

task\_id='runme\_' + str(i),

bash\_command='echo "{{ task\_instance\_key\_str }}" && sleep 1',

dag=dag,

)

task >> run\_this

*# [START howto\_operator\_bash\_template]*

also\_run\_this = BashOperator(

task\_id='also\_run\_this',

bash\_command='echo "run\_id={{ run\_id }} | dag\_run={{ dag\_run }}"',

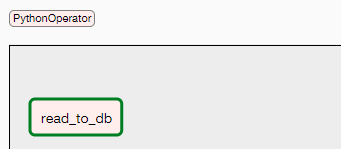
dag=dag,

)

*# [END howto\_operator\_bash\_template]*

also\_run\_this >> run\_this\_last

### Example 3:



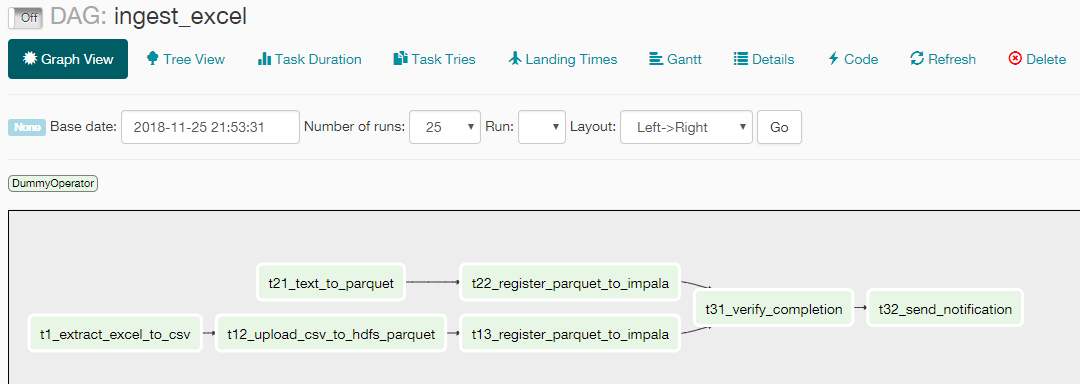
read\_to\_db = PythonOperator(task\_id='read\_to\_db',

provide\_context=False,

python\_callable=read\_to\_db,

dag=dag)

## 6.4 Design a new DAG



## 6.5 Copy DAG from host to docker

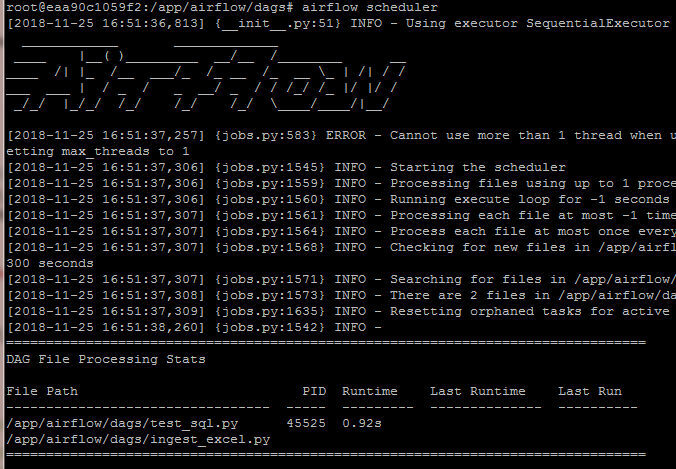
sudo docker cp ingest\_excel.py eaa90c1059f2:/app/airflow/dags

## 6.6 Copy DAG from docker to host

sudo docker cp eaa90c1059f2:/usr/local/lib/python3.6/site-packages/airflow/example\_dags/example\_python\_operator.py example\_python\_operator.py

## 6.7 Activate DAG

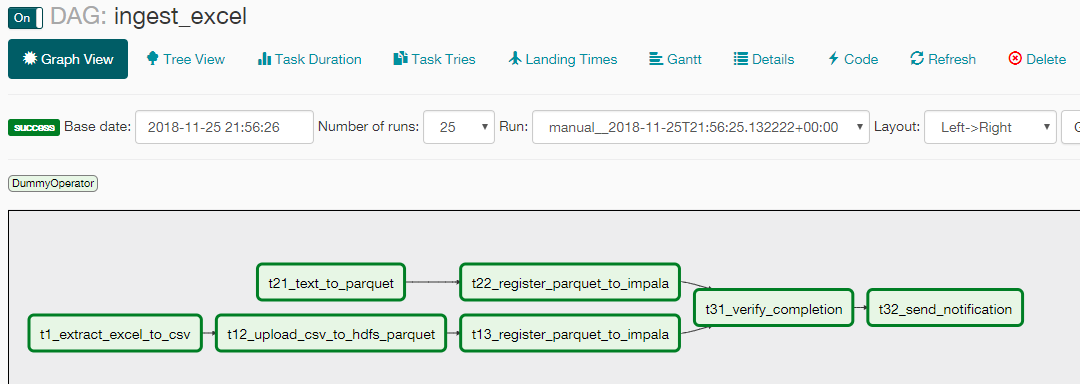
After a new dag is uploaded to the dags folder, run the airflow scheduler once and the new dag will be activated



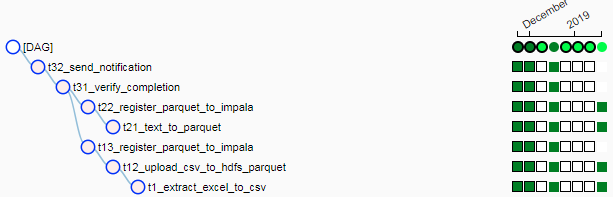
## 6.8 Run DAG

Turn on -> manually run it or wait until scheduled time arrives









**import** **airflow** **from \_\_future\_\_ import print\_function**

**import time**

**from builtins import** **range**

**from datetime import** **timedelta**

**from pprint import** **pprint**

**from** **airflow.models** **import** DAG

**#from** **airflow.operators.bash\_operator** **import** BashOperator

**from** **airflow.operators.python\_operator** **import** PythonOperator

args = {

'owner': 'rxie',

'start\_date': airflow.utils.dates.days\_ago(2),

}

dag = DAG(

dag\_id='ingest\_excel',

default\_args=args,

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

dagrun\_timeout=timedelta(minutes=60),

)

def print\_context(ds, \*\*kwargs):

pprint("kwargs info below:")

pprint(kwargs)

pprint("ds info below:")

print(ds)

return 'Whatever you return gets printed in the logs'

def print\_context(ds, \*\*kwargs):

pprint("DAG info below:")

pprint(kwargs)

print(ds)

return 'Whatever you return gets printed in the logs'

t11\_extract\_excel\_to\_csv = PythonOperator(

task\_id='t1\_extract\_excel\_to\_csv',

provide\_context=True,

python\_callable=print\_context,

op\_kwargs=None,

dag=dag

)

t12\_upload\_csv\_to\_hdfs\_parquet = PythonOperator(

task\_id='t12\_upload\_csv\_to\_hdfs\_parquet',

provide\_context=True,

python\_callable=print\_context,

op\_kwargs=None,

dag=dag,

)

t13\_register\_parquet\_to\_impala = PythonOperator(

task\_id='t13\_register\_parquet\_to\_impala',

provide\_context=True,

python\_callable=print\_context,

op\_kwargs=None,

dag=dag,

)

t21\_text\_to\_parquet = PythonOperator(

task\_id='t21\_text\_to\_parquet',

provide\_context=True,

python\_callable=print\_context,

op\_kwargs=None,

dag=dag,

)

t22\_register\_parquet\_to\_impala = PythonOperator(

task\_id='t22\_register\_parquet\_to\_impala',

provide\_context=True,

python\_callable=print\_context,

op\_kwargs=None,

dag=dag,

)

t31\_verify\_completion = PythonOperator(

task\_id='t31\_verify\_completion',

provide\_context=True,

python\_callable=print\_context,

op\_kwargs=None,

dag=dag,

)

t32\_send\_notification = PythonOperator(

task\_id='t32\_send\_notification',

provide\_context=True,

python\_callable=print\_context,

op\_kwargs=None,

dag=dag,

)

t11\_extract\_excel\_to\_csv >> t12\_upload\_csv\_to\_hdfs\_parquet

t12\_upload\_csv\_to\_hdfs\_parquet >> t13\_register\_parquet\_to\_impala

t21\_text\_to\_parquet >> t22\_register\_parquet\_to\_impala

t13\_register\_parquet\_to\_impala >> t31\_verify\_completion

t22\_register\_parquet\_to\_impala >> t31\_verify\_completion

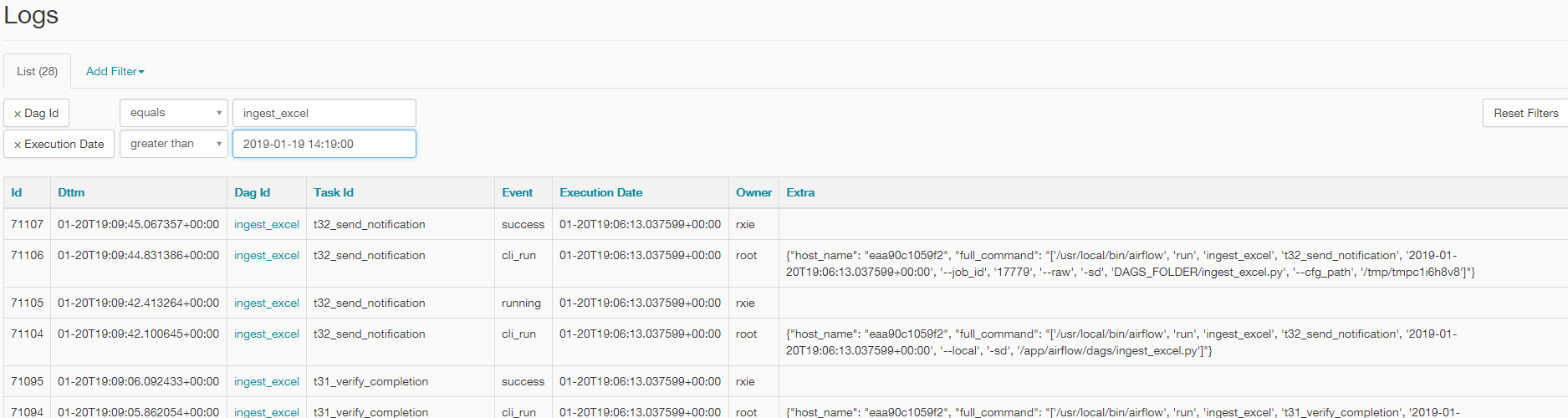
t31\_verify\_completion >> t32\_send\_notification

**#if** \_\_name\_\_ == "\_\_main\_\_":

**#** dag.cli()

## 6.9 Accessing log

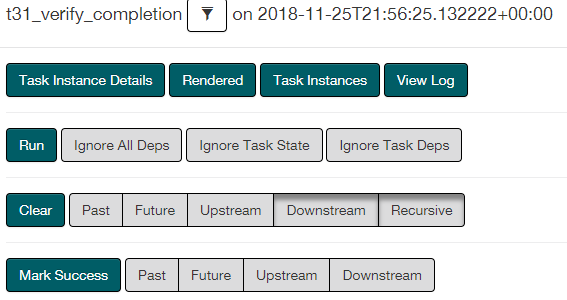
### 6.9.1 From dashboard



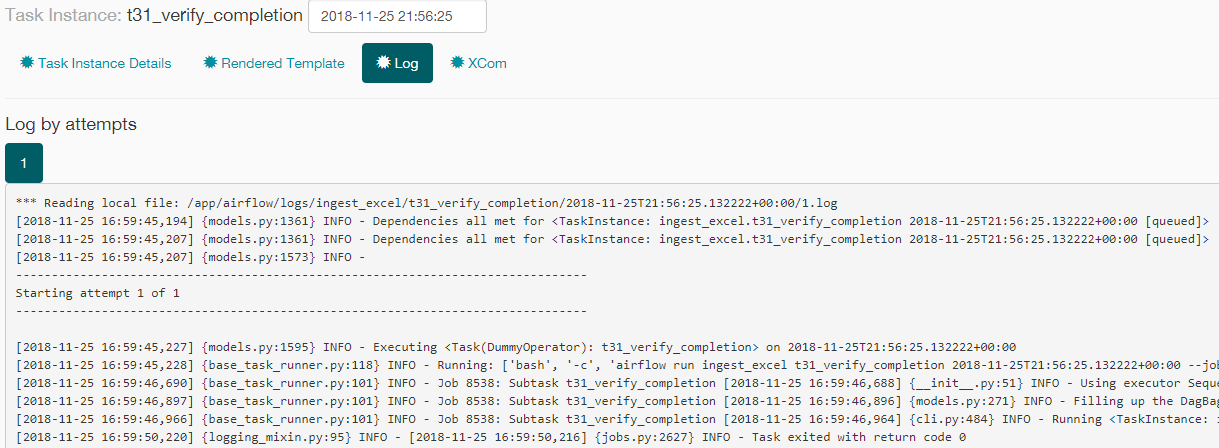
Note the hostname shows the docker’s id eaa90c1059f2

### 6.9.2 From the task

Set cursor on the task, pop up the following:



Click on View Log



More details from the log:

[2019-01-20 14:41:43,118] {logging\_mixin.py:95} INFO - 'kwargs info below:'

[2019-01-20 14:41:43,122] {logging\_mixin.py:95} INFO - {'END\_DATE': '2019-01-20',

'conf': <module 'airflow.configuration' from '/usr/local/lib/python3.6/site-packages/airflow/configuration.py'>,

'dag': <DAG: ingest\_excel>,

'dag\_run': <DagRun ingest\_excel @ 2019-01-20 19:40:24.884481+00:00: manual\_\_2019-01-20T19:40:24.884481+00:00, externally triggered: True>,

'ds\_nodash': '20190120',

'end\_date': '2019-01-20',

'execution\_date': <Pendulum [2019-01-20T19:40:24.884481+00:00]>,

'inlets': [],

'latest\_date': '2019-01-20',

'macros': <module 'airflow.macros' from '/usr/local/lib/python3.6/site-packages/airflow/macros/\_\_init\_\_.py'>,

'next\_ds': '2019-01-21',

'next\_execution\_date': datetime.datetime(2019, 1, 21, 0, 0, tzinfo=<TimezoneInfo [UTC, GMT, +00:00:00, STD]>),

'outlets': [],

'params': {},

'prev\_ds': '2019-01-20',

'prev\_execution\_date': datetime.datetime(2019, 1, 20, 0, 0, tzinfo=<TimezoneInfo [UTC, GMT, +00:00:00, STD]>),

'run\_id': 'manual\_\_2019-01-20T19:40:24.884481+00:00',

'tables': None,

'task': <Task(PythonOperator): t31\_verify\_completion>,

'task\_instance': <TaskInstance: ingest\_excel.t31\_verify\_completion 2019-01-20T19:40:24.884481+00:00 [running]>,

'task\_instance\_key\_str': 'ingest\_excel\_\_t31\_verify\_completion\_\_20190120',

'templates\_dict': None,

'test\_mode': False,

'ti': <TaskInstance: ingest\_excel.t31\_verify\_completion 2019-01-20T19:40:24.884481+00:00 [running]>,

'tomorrow\_ds': '2019-01-21',

'tomorrow\_ds\_nodash': '20190121',

'ts': '2019-01-20T19:40:24.884481+00:00',

'ts\_nodash': '20190120T194024.884481+0000',

'var': {'json': None, 'value': None},

'yesterday\_ds': '2019-01-19',

'yesterday\_ds\_nodash': '20190119'}

[2019-01-20 14:41:43,122] {logging\_mixin.py:95} INFO - 'ds info below:'

[2019-01-20 14:41:43,123] {logging\_mixin.py:95} INFO - 2019-01-20

[2019-01-20 14:41:43,123] {python\_operator.py:96} INFO - Done. Returned value was: Whatever you return gets printed in the logs

[2019-01-20 14:41:44,196] {logging\_mixin.py:95} INFO - [2019-01-20 14:41:44,193] {jobs.py:2627} INFO - Task exited with return code 0

Note the two highlighted messages are actually from the code:

**def** print\_context(ds, \*\*kwargs):

pprint("kwargs info below:")

pprint(kwargs)

pprint("ds info below:")

**print**(ds)

**return** 'Whatever you return gets printed in the logs'

## 6.10 Delete DAG

### Dag id test\_sql is still in DagBag. Remove the DAG file first

root@eaa90c1059f2:/app/airflow/logs/test\_sql/read\_to\_db# curl -X "DELETE" http://127.0.0.1:8080/api/experimental/dags/test\_sql

{

"error": "Dag id test\_sql is still in DagBag. Remove the DAG file first."

}

root@eaa90c1059f2:/app/airflow/logs/test\_sql/read\_to\_db# cd ../../../dags

root@eaa90c1059f2:/app/airflow/dags# ls

\_\_pycache\_\_ test\_sql.py

root@eaa90c1059f2:/app/airflow/dags# mv test\_sql.py test\_sql.py.old

root@eaa90c1059f2:/app/airflow/dags# curl -X "DELETE" http://127.0.0.1:8080/api/experimental/dags/test\_sql

{

"count": 2873,

"message": "Removed 2873 record(s)"

}

root@eaa90c1059f2:/app/airflow/dags#

Now rename it back to test\_sql.py



# Accessing Airflow

# Accessing Web GUI

# Accessing Log

# Airflow User Management

# Airflow DAG creation

Example dag

For the docker’s airflow:

1. Ensure the docker image is running:

sudo docker run -d --name airflow -p 8080:8080 boboairflow

1. Enter the docker

sudo docker exec -it airflow /bin/bash

root@eaa90c1059f2:/app/airflow/dags# ls

\_\_pycache\_\_ test\_sql.py

root@eaa90c1059f2:/app/airflow/dags#

## Dag code

**from** **airflow** **import** DAG

*#from airflow.operators.bash\_operator import BashOperator*

**from** **datetime** **import** datetime, timedelta

**from** **airflow.hooks** **import** HttpHook, MySqlHook

**from** **airflow.operators** **import** PythonOperator

**from** **airflow.models** **import** DAG

**# import pymysql to connect to mysql**

**import** **pymysql**

*# Main DAG function*

**def** read\_to\_db():

*# establish mysql connection*

*# For debugging, this entire script can be verified through command line*

#host is where the mysql resides, note: there must be a user [airflow@172.17.0.2](mailto:airflow@172.17.0.2) where 172.17.0.2 is the docker’s ip address found by: docker inspect -f

# the user [airflow@172.17.0.2](mailto:airflow@172.17.0.2) muse be granted needed privilege on mysql with GRANT ALL TO 'airflow'@'172.17.0.2' IDENTIFIED BY 'airflow'; FLUSH PRIVILEGES;

conn = pymysql.connect(host='192.168.211.251', port=3306, user='airflow', passwd='airflow', db='airflow')

cur = conn.cursor()

cur.execute("INSERT INTO my\_test (name, created\_on, created\_by) values ('airflow in docker', CURRENT\_TIMESTAMP, 'rxie');")

cur.close()

conn.commit()

conn.close()

*# Following are defaults which can be overridden later on*

default\_args = {

'owner': 'bobo',

'depends\_on\_past': False,

'start\_date': datetime(2018, 11, 23),

'email': ['xie3208080@gmail.com'],

'email\_on\_failure': False,

'email\_on\_retry': False,

'retries': 1,

'retry\_delay': timedelta(minutes=1),

}

# the following defines dag name/id and schedule

dag = DAG('test\_sql', schedule\_interval='\* \* \* \* \* ', default\_args=default\_args)

# task name is read\_to\_db, with task\_id of the same wording

# log will be created as per dag per task per run

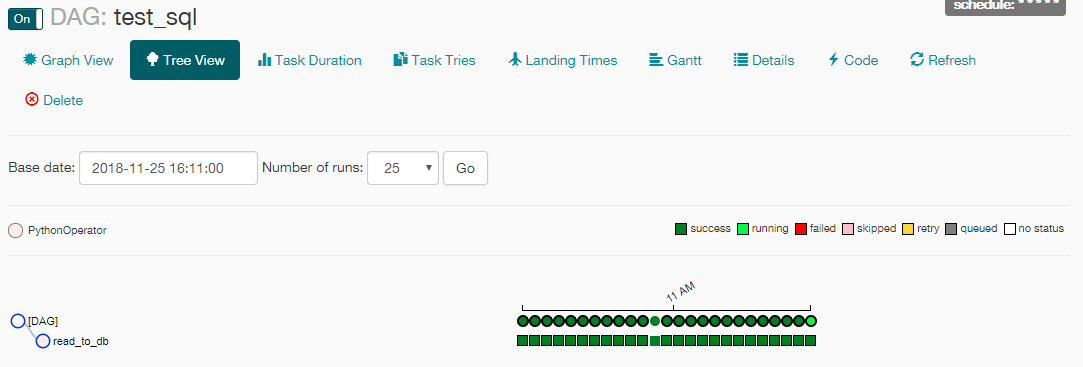
read\_to\_db = PythonOperator(task\_id='read\_to\_db',

provide\_context=False,

python\_callable=read\_to\_db,

dag=dag)

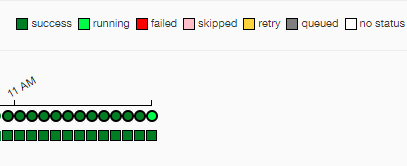
## Running status (Tree View)





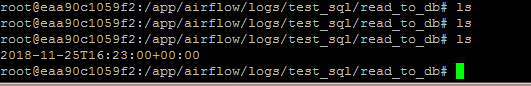


## One is running



## Log is generated

Note: log is per dag per task per run



## Read the log

root@eaa90c1059f2:/app/airflow/logs/test\_sql/read\_to\_db/2018-11-25T16:23:00+00:00# cat 1.log

[2018-11-25 11:24:06,284] {models.py:1361} INFO - Dependencies all met for <TaskInstance: test\_sql.read\_to\_db 2018-11-25T16:23:00+00:00 [queued]>

[2018-11-25 11:24:06,287] {models.py:1361} INFO - Dependencies all met for <TaskInstance: test\_sql.read\_to\_db 2018-11-25T16:23:00+00:00 [queued]>

[2018-11-25 11:24:06,287] {models.py:1573} INFO -

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

Starting attempt 1 of 2

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

[2018-11-25 11:24:06,297] {models.py:1595} INFO - Executing <Task(PythonOperator): read\_to\_db> on 2018-11-25T16:23:00+00:00

[2018-11-25 11:24:06,297] {base\_task\_runner.py:118} INFO - Running: ['bash', '-c', 'airflow run test\_sql read\_to\_db 2018-11-25T16:23:00+00:00 --job\_id 8187 --raw -sd DAGS\_FOLDER/test\_sql.py --cfg\_path /tmp/tmpd96efrdv']

[2018-11-25 11:24:07,222] {base\_task\_runner.py:101} INFO - Job 8187: Subtask read\_to\_db [2018-11-25 11:24:07,221] {\_\_init\_\_.py:51} INFO - Using executor SequentialExecutor

[2018-11-25 11:24:07,454] {base\_task\_runner.py:101} INFO - Job 8187: Subtask read\_to\_db [2018-11-25 11:24:07,453] {models.py:271} INFO - Filling up the DagBag from /app/airflow/dags/test\_sql.py

[2018-11-25 11:24:07,558] {base\_task\_runner.py:101} INFO - Job 8187: Subtask read\_to\_db [2018-11-25 11:24:07,555] {cli.py:484} INFO - Running <TaskInstance: test\_sql.read\_to\_db 2018-11-25T16:23:00+00:00 [running]> on host eaa90c1059f2

[2018-11-25 11:24:07,576] {python\_operator.py:96} INFO - Done. Returned value was: None

[2018-11-25 11:24:07,594] {base\_task\_runner.py:101} INFO - Job 8187: Subtask read\_to\_db /usr/local/lib/python3.6/site-packages/airflow/utils/helpers.py:346: DeprecationWarning: Importing 'HttpHook' directly from 'airflow.hooks' has been deprecated. Please import from 'airflow.hooks.[operator\_module]' instead. Support for direct imports will be dropped entirely in Airflow 2.0.

[2018-11-25 11:24:07,595] {base\_task\_runner.py:101} INFO - Job 8187: Subtask read\_to\_db DeprecationWarning)

[2018-11-25 11:24:07,595] {base\_task\_runner.py:101} INFO - Job 8187: Subtask read\_to\_db /usr/local/lib/python3.6/site-packages/airflow/utils/helpers.py:346: DeprecationWarning: Importing 'MySqlHook' directly from 'airflow.hooks' has been deprecated. Please import from 'airflow.hooks.[operator\_module]' instead. Support for direct imports will be dropped entirely in Airflow 2.0.

[2018-11-25 11:24:07,595] {base\_task\_runner.py:101} INFO - Job 8187: Subtask read\_to\_db DeprecationWarning)

[2018-11-25 11:24:07,595] {base\_task\_runner.py:101} INFO - Job 8187: Subtask read\_to\_db /usr/local/lib/python3.6/site-packages/airflow/utils/helpers.py:346: DeprecationWarning: Importing 'PythonOperator' directly from 'airflow.operators' has been deprecated. Please import from 'airflow.operators.[operator\_module]' instead. Support for direct imports will be dropped entirely in Airflow 2.0.

[2018-11-25 11:24:07,596] {base\_task\_runner.py:101} INFO - Job 8187: Subtask read\_to\_db DeprecationWarning)

[2018-11-25 11:24:11,347] {logging\_mixin.py:95} INFO - [2018-11-25 11:24:11,344] {jobs.py:2627} INFO - Task exited with return code 0

The dag has been running with 23-10 = 13 success after the mysql is fixed.

# Miscelleneous

## 8.1 (Optional) Install Tree

Pip does’t work on installing tree, yum is not available in Docker, installing tree within Docker image, using:

apt-get install tree

## 8.2 Update timezone to local

root@eaa90c1059f2:/app/airflow/dags# dpkg-reconfigure tzdata

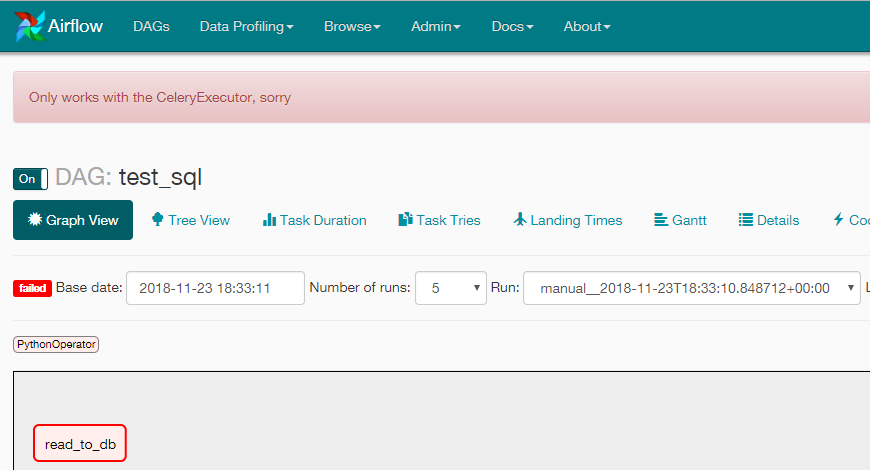
## 8.3 Get docker’s IP from host

[rxie@pocnnr1n1 dags]$ sudo docker inspect -f '{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}' eaa90c1059f2

[sudo] password for rxie:

172.17.0.2

## Run a DAG manually fails



## Setup and send out email

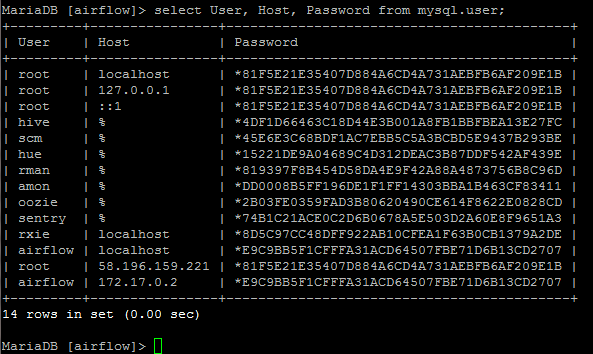
## MYSQL Privielge setup

To set up the right user the right privilege to access mysql from docker, docker’s ip is needed, and the user ‘airflow@ '172.17.0.2' should be granted privilege to the target database.table (in this case the db is airflow.my\_test)

GRANT ALL TO 'airflow'@'172.17.0.2' IDENTIFIED BY 'airflow' ;

FLUSH PRIVILEGES;

Confirmed by: select User, Host, Password from mysql.user;



Python operator

pprint(kwargs) shows the DAG info below in the task’s log

[2018-11-25 20:57:32,468] {logging\_mixin.py:95} INFO - {'END\_DATE': '2018-11-26',

'conf': <module 'airflow.configuration' from '/usr/local/lib/python3.6/site-packages/airflow/configuration.py'>,

'dag': <DAG: example\_python\_operator>,

'dag\_run': <DagRun example\_python\_operator @ 2018-11-26 01:57:23.767021+00:00: manual\_\_2018-11-26T01:57:23.767021+00:00, externally triggered: True>,

'ds\_nodash': '20181126',

'end\_date': '2018-11-26',

'execution\_date': <Pendulum [2018-11-26T01:57:23.767021+00:00]>,

'inlets': [],

'latest\_date': '2018-11-26',

'macros': <module 'airflow.macros' from '/usr/local/lib/python3.6/site-packages/airflow/macros/\_\_init\_\_.py'>,

'next\_ds': None,

'next\_execution\_date': None,

'outlets': [],

'params': {},

'prev\_ds': None,

'prev\_execution\_date': None,

'run\_id': 'manual\_\_2018-11-26T01:57:23.767021+00:00',

'tables': None,

'task': <Task(PythonOperator): print\_the\_context>,

'task\_instance': <TaskInstance: example\_python\_operator.print\_the\_context 2018-11-26T01:57:23.767021+00:00 [running]>,

'task\_instance\_key\_str': 'example\_python\_operator\_\_print\_the\_context\_\_20181126',

'templates\_dict': None,

'test\_mode': False,

'ti': <TaskInstance: example\_python\_operator.print\_the\_context 2018-11-26T01:57:23.767021+00:00 [running]>,

'tomorrow\_ds': '2018-11-27',

'tomorrow\_ds\_nodash': '20181127',

'ts': '2018-11-26T01:57:23.767021+00:00',

'ts\_nodash': '20181126T015723.767021+0000',

'var': {'json': None, 'value': None},

'yesterday\_ds': '2018-11-25',

'yesterday\_ds\_nodash': '20181125'}

1. <https://docs.docker.com/install/linux/docker-ce/centos/#install-from-a-package> [↑](#footnote-ref-1)
2. Refer to the Airflow section of starting new docker and image, and how to get the container id with docker ps [↑](#footnote-ref-2)