链滴

Airflow 初探

作者：OldPanda
原文链接：https://ld246.com/article/1544509325691
来源网站：链滴

这篇其实转自我自己的博客，简单记录了折腾 Airflow 的过程。发到这里，希望能帮到更多的同样初接触 Airflow 的朋友。

距离上回写小作文过了多半年，这几个月来发生了一些事情，最大的就是这个月初我换了工作，从 Pal Alto 换到了 Mountain View ，附近吃的喝的玩的较之以前有了很大的提升。但总归主业是过来干的，上班大概三周了，很开心刚入职就让我研究开发一个新项目，其核心就是Airflow ，一个有向无图任务（ Directed Acyclic Graph – DAG）的调度工具，看了不少文档博客，踩了大大小小的坑，是成功的把它运行到了服务器上，下一步就可以在这个基础上开发一些东西。既然告一段落，那么应写点文字，以 Airflow 为例，简单描述如何把一个程序作为一个服务运行在 Linux 机器上。

安装

我创建了一个用户 airflow 专门负责 Airflow 的运行，即无论是安装运行 Airflow ，还是修改 Airflow 的配置，都通过该用户来进行，为了进展顺利，给这个用户开了绿灯，授予 root 权限。安装 Airflow 这一步其实是最简单的，官网有详细的说明。我的环境是 Python 3.6.6 ， Airflow 的版本是 1.10.0 为了避免与已有的包冲突，我将其安装在一个 virtualenv 中，在 /home/airflow 下执行如下命令

```
virtualenv venv -p `which python3`
source venv/bin/activate
pip install apache-airflow[postgres,crypto,gcp_api]==1.10.0
```

方括号中的是可选的依赖，在这里我用 PostgreSQL 作为 Airflow metadata 的数据库（默认是 SQLi e ），并且想要加密我的各种链接参数如密码，同时想要与谷歌云服务进行交互，所以安装这三个。户可以根据自己的实际情况选择不同的依赖，详细说明可以参考官方文档。

插一句题外话， 如果想给自己开发的 Python 包添加可选依赖的话（方括号），可以通过定义setup.py extra_require来实现，具体可参考这里。

配置

因为我们希望把 Airflow 作为一个服务运行起来，便于以后的继续开发及维护，而不是运行一次给人看效果就拉倒，所以我采用了systemd来管理 Airflow 进程的运行。

关于 systemd 的配置， Airflow 的文档上有个简要介绍，具体来说，在我的配置中，我将环境变量 A RFLOW_HOME 设置为 /etc/airflow ，将 AIRFLOW_CONFIG 设置为 /etc/airflow/airflow.cfg ，样，在我的文件 /etc/sysconfig/airflow 中只有这两行环境变量

```
AIRFLOW\_CONFIG=/etc/airflow/airflow.cfg
AIRFLOW\_HOME=/etc/airflow
```

为了将 Airflow 能顺利运行起来，有两个必需的服务，一个 webserver ，用于显示 web UI ，一个 sc eduler ，用于执行 DAG 中的任务，好在 Airflow 已经提供给了我们这两个服务的示例文件：airflow webserver.service 和 airflow-scheduler.service ，唯一需要修改的一行就是 ExecStart ，因为我们要在虚拟环境中运行 Airflow ，最终这两个文件分别如下所示

- airflow-webserver.service

```
#
# Licensed to the Apache Software Foundation (ASF) under one
# or more contributor license agreements. See the NOTICE file
# distributed with this work for additional information
# regarding copyright ownership. The ASF licenses this file
# to you under the Apache License, Version 2.0  (the
```

```
# "License"); you may not use this file except in compliance
# with the License. You may obtain a copy of the License at
#
# [http://www.apache.org/licenses/LICENSE\-](http://www.apache.org/licenses/LICENSE-)2.0
#
# Unless required by applicable law or agreed to in writing,
# software distributed under the License is distributed on an
# "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
# KIND, either express or implied. See the License for the
# specific language governing permissions and limitations
# under the License.

[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=airflow
Group=airflow
Type=simple
ExecStart=/bin/bash -c 'source /home/airflow/venv/bin/activate ; airflow webserver --pid /run
airflow/webserver.pid'
Restart=on-failure
RestartSec=5s
PrivateTmp=true

[Install]
WantedBy=multi-user.target
```

- airflow-scheduler.service

```
#
# Licensed to the Apache Software Foundation (ASF) under one
# or more contributor license agreements. See the NOTICE file
# distributed with this work for additional information
# regarding copyright ownership. The ASF licenses this file
# to you under the Apache License, Version 2.0  (the
# "License"); you may not use this file except in compliance
# with the License. You may obtain a copy of the License at
#
# [http://www.apache.org/licenses/LICENSE\-](http://www.apache.org/licenses/LICENSE-)2.0
#
# Unless required by applicable law or agreed to in writing,
# software distributed under the License is distributed on an
# "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
# KIND, either express or implied. See the License for the
# specific language governing permissions and limitations
# under the License.

[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=airflow
Group=airflow
Type=simple
ExecStart=/bin/bash -c 'source /home/airflow/venv/bin/activate ; airflow scheculer'
Restart=always
RestartSec=5s

[Install]
WantedBy=multi-user.target
```

具体 bash 的位置因系统而异，需要注意的一点就是必须用绝对路径来执行。然后将两者置于 /etc/systemd/system 下。还有一个文件是不可缺少的 airflow.conf ，这个直接抄下来放在 /etc/systemd 里。

这样 systemd 的部分算是完成了，但还不算完，我们还需要一个 airflow.cfg 来告诉 Airflow 如何配。每个用户的具体情况不一样，我就不一一赘述了，这里只提几个比较重要的。

- sql_alchemy_conn = postgresql+psycopg2://<user>:<password>@<host>:<port> 我们在产环境采用 PostgreSQL 作为 metadata 数据库
- load_examples = False 示例 DAG 自己在开发测试的时候是很好的参考，但明显在生产环境中用到它们，所以关掉
- fernet_key = <some base64 string> 这个肯定得有，要不然 Airflow 会把各种链接的敏感参数存明文，生成方法可以参考这里
- executor = LocalExecutor LocalExecutor 可以最大程度的利用单机的并行能力，即运行多个进程同时执行不同的任务，对于目前的需求来说是足够了，以后还可以考虑使用 redis + celery 的方式进横向扩展

运行

首先要初始化数据库，这个得手动搞，还是以 airflow 的身份运行

```
source ~/venv/bin/activate
export AIRFLOW_HOME=/etc/airflow
airflow initdb
```

然后就可以用 systemd 来控制 Airflow 的启停了

```
sudo systemctl [start|stop|restart|status] airflow-webserver
sudo systemctl [start|stop|restart|status] airflow-scheduler
```

每次希望新加入一个 DAG 时，只需要把 Python 文件放到 /etc/airflow/dags 里即可。

参考

- https://airflow.readthedocs.io/en/latest/
- https://github.com/apache/incubator-airflow
- https://wecode.wepay.com/posts/airflow-wepay
- https://medium.com/@vando/airflow-inside-a-virtual-enviroment-and-integrated-with-syst

md-3b6427bd6430

- https://robinhood.engineering/why-robinhood-uses-airflow-aed13a9a90c8