玩转ApsaraDB HBase内嵌Spark系列(2)--通过Airflow 快速实现作业编排及定时调度

前言

当业务场景数据处理复杂时，需要多个作业组成一个工作流，周期性的去调度运行，作业失败需要有状态及重试等功能。目前有几种方案：

- 方案一 crontab：使用crontab去定时调用提交作业到LivyServer的脚本，作业依赖、作业重试、报警等处理比较麻烦；
- 方案二 自己开发：自己写一个作业管理系统对接LivyServer管理，该方式可以完成业务对于作业编排和定时调度的需求，但是需要投入开发成本；
- 方案三 airflow：使用开源的作业编排、调度和监控workflow的平台Airflow；
- 方案三 官方调度：使用 ApsaraDB HBase内嵌Spark的工作台，目前还在开发中，尽情期待...

本篇主要介绍如何使用开源airflow来搭建ApsaraDB HBase内嵌Spark的作业编排调度系统。

Airflow服务搭建

环境准备

1. 准备一台搭建Airflow的ECS：该ECS需要和云HBase的Spark集群在同一个VPC网络中。

Airflow源码安装

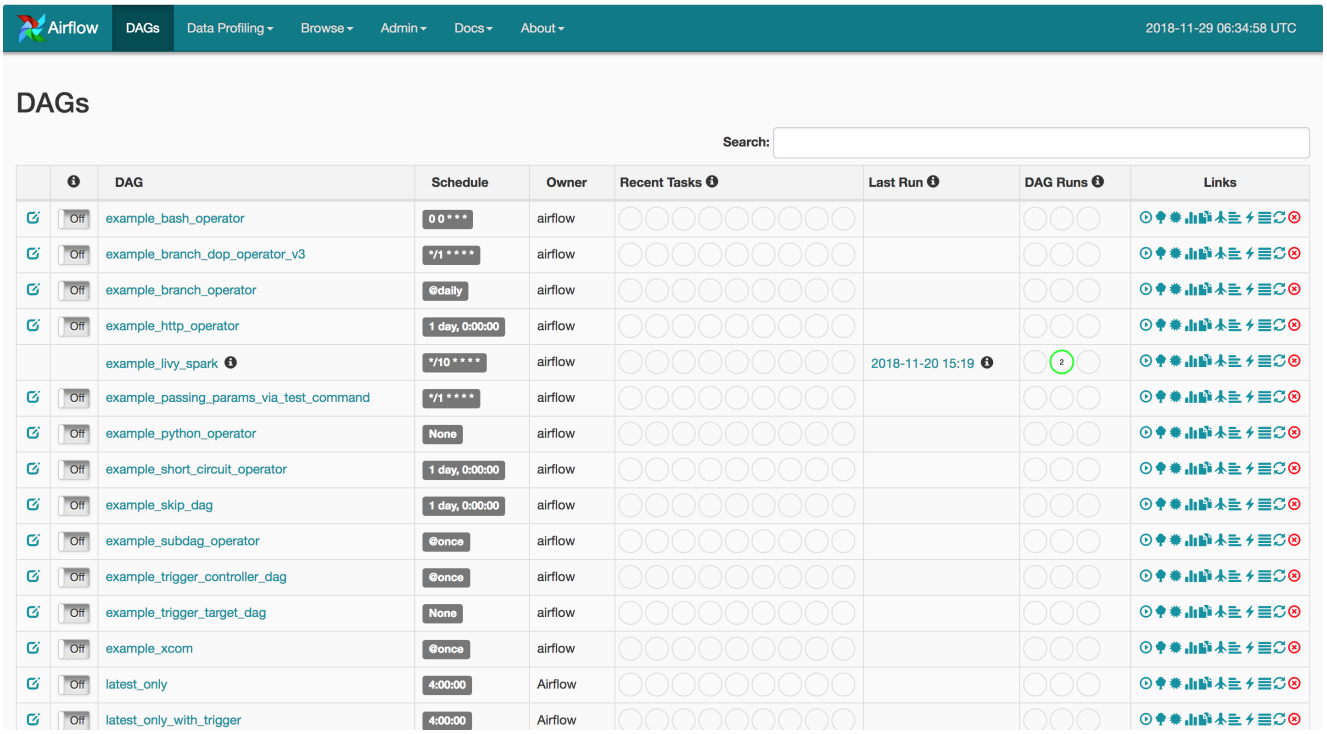
1. 下载airflow 从https://github.com/apache/incubator-airflow/releases 下载最新的release版本
2. 准备 ApsaraDB HBase内嵌Spark的airflow插件livy_spark_operator.py

- 从aliyun-apsaradb-hbase-demo下载livy_spark_operator.py插件
- 把下载的插件livy_spark_operator.py拷贝到airflow项目的./airflow/contrib/operators/目录

1. 安装airflow

- 指定airflow安装到哪个目录：export AIRFLOW_HOME = ~/airflow
- 在airflow源码根目录执行命令安装： python setup.py install

1. 初始化airflow airflow initdb
2. 启动airflow airflow webserver -p 8080，如果后台运行可以使用 airflow webserver -p 8080 -D True
3. 查看airflow页面 通过ecsip:8080在浏览器访问：



开发Spark的作业编排

1. 参考Spark文档通过LivyServer提交单个作业进行调试跑通
2. 开发spark作业编排

- 开发作业编排实例，源码下载

```
from airflow import utils
from airflow import DAG
from datetime import datetime, timedelta
from airflow.contrib.operators.livy_spark_operator import LivySubmitRunOperat
or

now = datetime.now()
now_to_the_hour = (now - timedelta(0, 0, 0, 0, 0, 3)).replace(minute=0, seco
nd=0, microsecond=0)
START_DATE = now_to_the_hour
DAG_NAME = 'livy_operator_test1'

default_args = {
```

```
    'owner': 'test',
    'depends_on_past': True,
    'start_date': utils.dates.days_ago(2)
}
dag = DAG(DAG_NAME, schedule_interval='*/10 * * * *', default_args=default_ar
gs)
json = {"file": "/spark/pi.py"}
livy_conn = "http://ap-xxx-b.rds.aliyuncs.com:8998"
job1 = LivySubmitRunOperator(task_id='livy_python_task1', json=json, livy_con
n=livy_conn, dag=dag)
job2 = LivySubmitRunOperator(task_id='livy_python_task2', json=json, livy_con
n=livy_conn, dag=dag)
job2.set_upstream(job1)
```

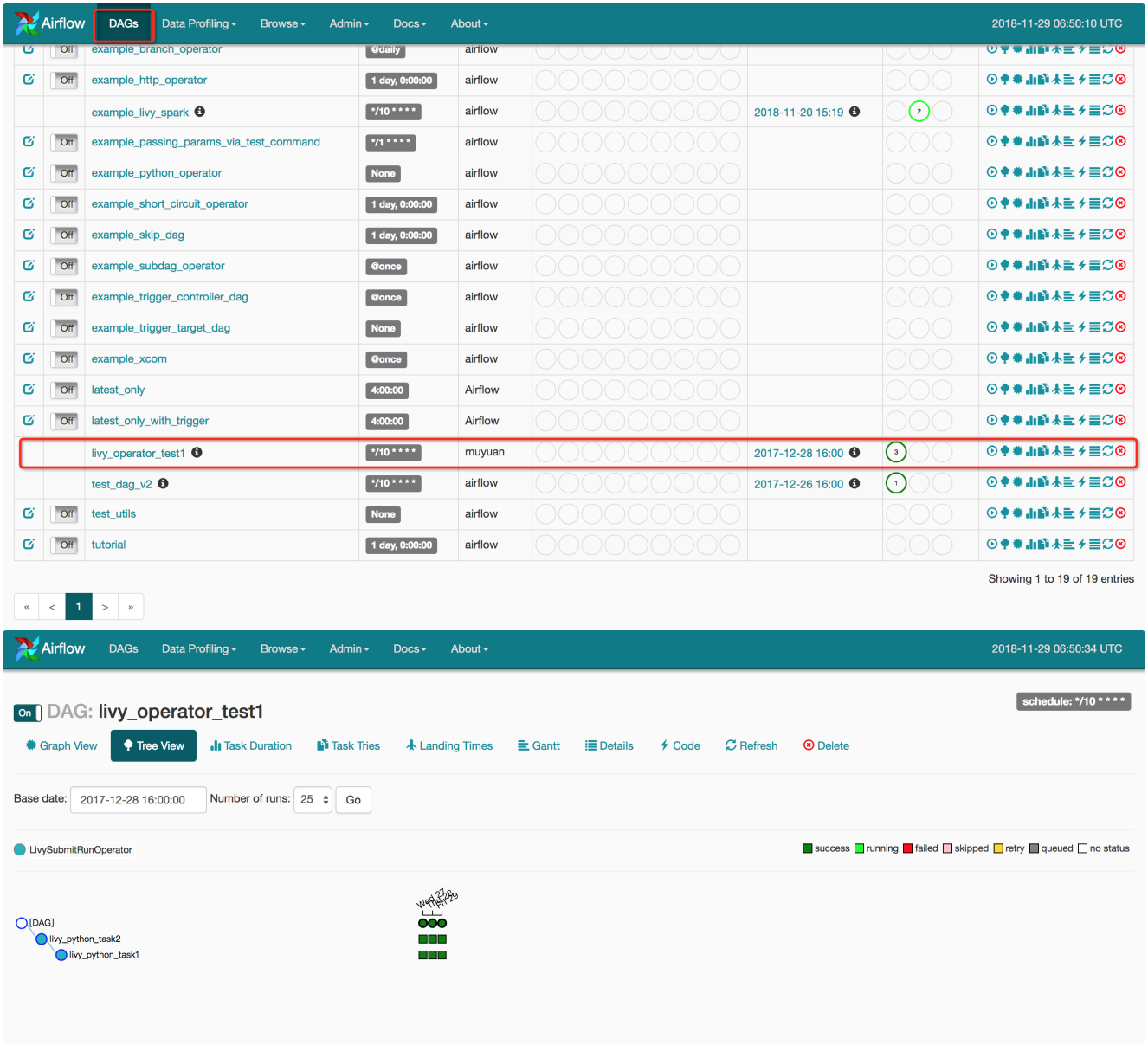
- 样例说明
  - 引入Spark的插件：from airflow.contrib.operators.livy_spark_operator import LivySubmitRunOperator
  - 设置工作流名称：DAG_NAME = 'livy_operator_test1'
  - 设置工作流调度周期：dag = DAG(DAG_NAME, schedule_interval='*/10 * * * *', default_args=default_args)
  - 作业提交的json：json = {"file": "/spark/pi.py"}
  - 获取对应Spark集群的LivyServer地址:ivy_conn = "http://ap-xxx-master1-001.spark.9b78df04-b.rds.aliyuncs.com:8998"
  - 定义两个作业：job1、job2
  - 构建作业依赖：job2.set_upstream(job1)

1. 部署调度工作流

- 将上面的工作流样例文件copy到AIRFLOW_HOME/dags/目录
- 使用airflow list_dags 查看对应的工作流是否安装成功

1. 工作流调试

- 调试单个task：airflow test livy_operator_test1 livy_python_task1 2017-07-01
- 调试整个工作流：airflow backfill livy_operator_test1 -s 2017-12-27
- 日志查看：在AIRFLOW_HOME/logs/查看日志
- 在页面查看工作流状态



相关资料

- airflow插件及样例：https://github.com/aliyun/aliyun-apsaradb-hbase-demo/tree/master/spark/airflow
- airflow官方文档：https://airflow.apache.org/
- airflow报警对接钉钉机器人：http://yangcongchufang.com/airflow/airflow-dingding-bot-plugin.html
- airflow实践：https://sanyuesha.com/2017/11/13/airflow/