

一：任务

1: 任务描述

在搭建起来的平台上，写一个airflow的脚本，脚本中包含两个task；

t1: 从ftp服务器上下载一个csv文件,ftp服务器的用户名、密码需要放到airflow的变量里

t2: 写一个简单的python_operator，处理这个csv文件，把csv文件的内容print出来，这里需要关注csv路径信息如何传递

2: 任务环境

- 本机Windows10系统
- 虚拟机Ubuntu16
- airflow框架搭在虚拟机linux系统上
- FTP服务器架在Windows上

3: 任务完成流程

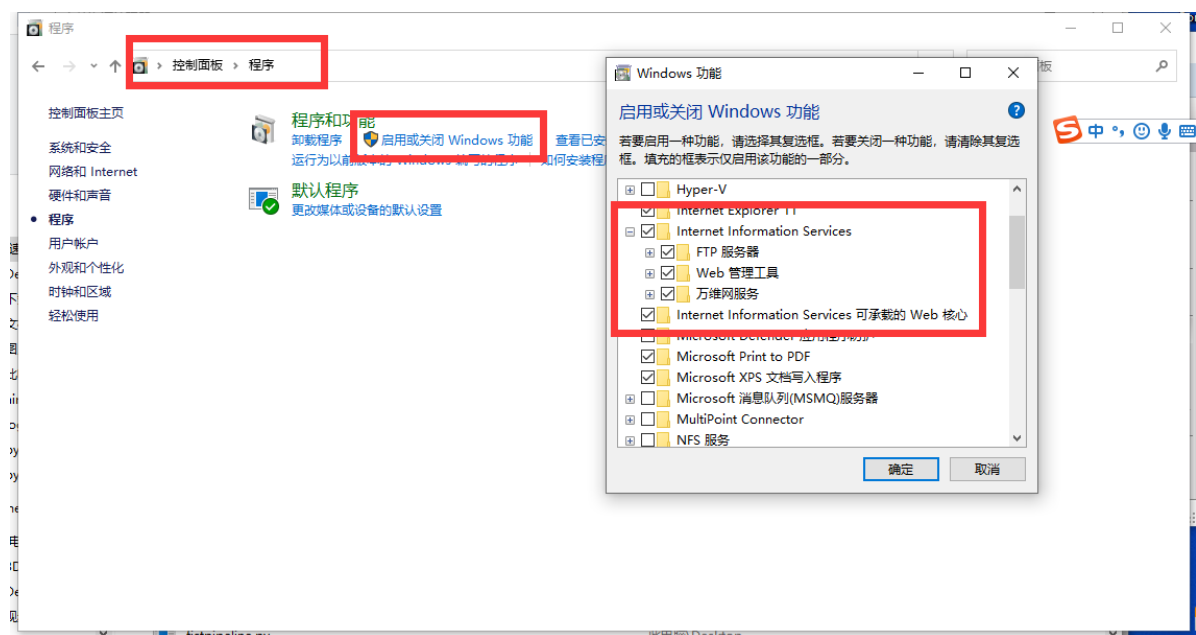
- Windows，搭建一个ftp服务器，在服务器上放一个写有 陈欢、test、csv 的data.csv文件用于测试；
- 编写airflow脚本，两个task，确定依赖关系
- 通过web运行airflow脚本，查看文件和日志，确定文件下载成功，输出结果正确；

二：过程

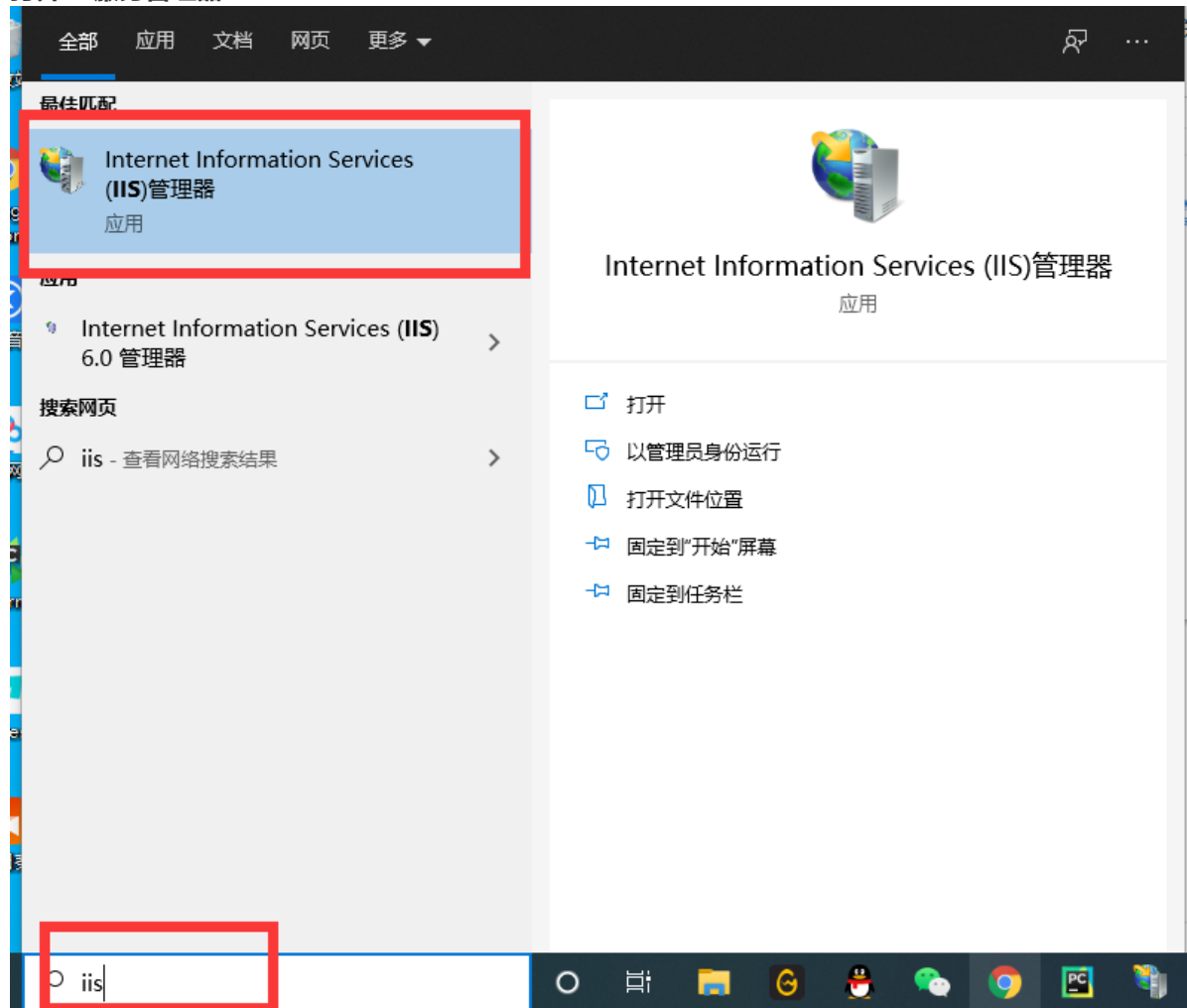
1: FTP服务器

Windows系统IIS服务

控制面板---->程序---->启用或关闭Windows功能---->把框起来的IIs服务都打开，这里我不确定开哪些，所以我都打开了。

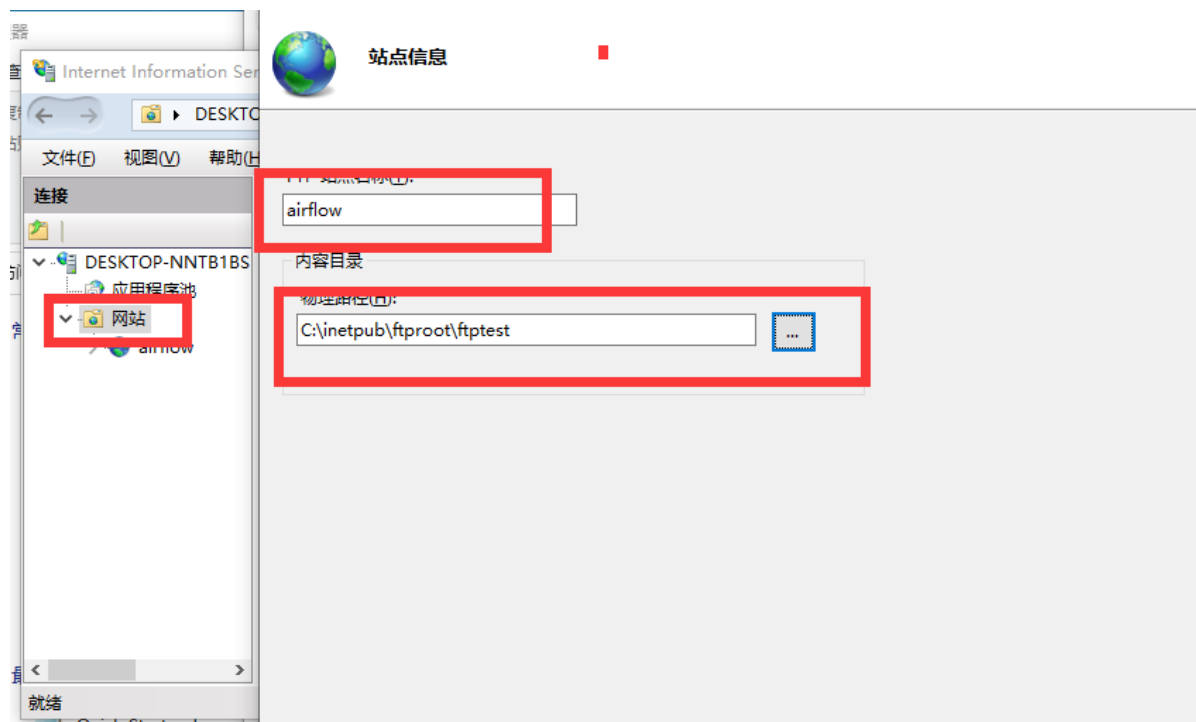


打开IIS服务管理器



配置ftp服务器

右击网站新建ftp站点，物理路径就是访问时的默认目录，可以自己指定，服务器搭建完成后把文件放在这里；



IP地址一定要是自己本机的IP，ipconfig查看本机IP然后再选择

添加 FTP 站点

绑定和 SSL 设置

绑定

IP 地址(A): 192.168.1.108 端口(O): 21

☐ 启用虚拟主机名(E):

虚拟主机(示例: ftp.contoso.com)(H):

☒ 自动启动 FTP 站点(I)

SSL

☒ 无 SSL(L)

☐ 允许 SSL(W)

☐ 需要 SSL(R)

SSL 证书(C): 未选定 选择(S)... 查看(V)...

上一页(P) 下一步(N) 完成(F) 取消

后续的访问设置，我是默认所有用户均可访问；但通过linux的ftp、lftp等命令仍需要用户名密码登录；可以自己新建一个Windows用户给其访问权限。后续的任务也可以通过这个用户来完成。

放个文件

我在ftp服务器上放了一个data.csv文件用于测试，内容如下：

	A	B	C
1	陈欢	test	csv
2			
3			

2: airflow脚本编写

#导入依赖库

```
import airflow
from airflow import DAG
```

#导入特定的执行器

```
from airflow.operators.bash_operator import BashOperator
from airflow.operators.python_operator import PythonOperator
from datetime import timedelta
import socket
import csv
import os
from ftplib import FTP
```

#these args will get passed on to each operator

#you can override them on a per-task basis during operator initialization

#显式地将一组参数传递给每个任务的构造函数作为默认参数

```
default_args = {
    'owner': 'airflow',
    'depends_on_past': False,
    'start_date': airflow.utils.dates.days_ago(2),
    'email': ['huan.chen@kylg.org'],
    'email_on_failure': False,
    'email_on_retry': False,
    'retries': 1,
    'retry_delay': timedelta(minutes=5),
    'user_name': 'chenhuan',
    'password': '213012',
}
```

#我们需要一个DAG对象来嵌入我们的任务。 这里我们需要传递一个定义dag_id的字符串，它用作DAG的唯一标识符。 我们还传递我们刚刚定义的默认参数字典，并为DAG定义1天的schedule_interval。

```
dag = DAG(
    'print_csvdata',
    default_args=default_args,
    description='A simple tutorial DAG',
    schedule_interval=timedelta(days=1))
```

#task1获取csv的Python函数，这里我是作为简单的测试默认给出了IP地址和端口，可以像用户名密码一样设置为默认参数，方便后续更改。path路径是airflow容器内的路径，可以自己指定，对于在容器内执行的Python脚本来说也就是本地文件了；下载完成后可以进入容器相应的路径查看是否存在该文件。

```
def get_csv():
    ftp = FTP()
    try:
        ftp.connect(host='192.168.1.108', port=21) # ftp connect函数的作用
        print("*****已经成功连接'%s'服务器FTP服务!!!")
        ftp.login(dag.default_args['user_name'], dag.default_args['password'])
        print(ftp.getwelcome()) # 显示ftp服务器欢迎信息
        bufsize = 1024
        filename = "data.csv"
        path = '/usr/local/spark/resources/data/data.csv'
        file_handler = open(path, 'wb').write # 以写模式在本地打开文件
        ftp.retrbinary('RETR %s' % filename, file_handler, bufsize) # 接收服务器上
        文件并写入本地文件
        ftp.quit()
    except (socket.error, socket.gaierror) as e:
        print(e)
        exit()
```

#task1执行get_csv函数

```
t1 = PythonOperator(
    task_id='get_csv',
    python_callable=get_csv,
    dag=dag)
```

#task2打印csv内容的Python函数，路径这里，我直接给出了，这样写死了不好，后面再慢慢学，看有啥好方法。

```
def print_data():
    path = '/usr/local/spark/resources/data/data.csv'
    with open(path) as f:
        rows = csv.reader(f)
        for row in rows:
            for text in row:
                print(text)
```

#task2执行print_data函数

```
t2 = PythonOperator(
    task_id='print_data',
    depends_on_past=False,
    python_callable=print_data,
    dag=dag)
```

#确定依赖关系，只有下载了csv文件之后才能进行读取打印

```
t2.set_upstream(t1)
```

3: 拷贝脚本

完成脚本编写之后需要把脚本提交到airflow，可以进入运行起来的容器用airflow命令刷新运行，也可以直接通过web可视化界面运行；

添加脚本

脚本copy到airflow对应路径：

```
# airflow LocalExecutor
airflow-webserver:
    image: docker-airflow-spark:1.10.7_3.0.1
    restart: always
    networks:
        - default_net
    depends_on:
        - postgres
    environment:
        - LOAD_EX=n
        - EXECUTOR=Local
    volumes:
        - ../dags:/usr/local/airflow/dags #DAG folder
        - ../spark/app:/usr/local/spark/app #Spark Scripts (Must be the same path as the host)
        - ../spark/resources:/usr/local/spark/resources #Resources folder (Must be the same path as the host)
    ports:
        - "8282:8282"
    command: webserver
```

上图是docker-compose.yml文件airflow-webserver部分的配置截图，可以看到这里做了宿主机和容器两者间的文件映射，格式为（**宿主机路径：容器内部路径**），都可自行更改；这是docker的数据卷技术，保证两个路径下拥有相同的文件副本，一个文件夹下发生的任何改变，如文件的增删查改，都会相应的同步到另一个文件。可用于数据保存等场景，在宿主机保留一个数据副本，免得删掉容器之后里边的数据也随之消失。

通过上述理解，我们可以不用进入容器直接把编写好的dags放在宿主机的airflow/dags/文件夹下，不管容器是否启动，这一步都有效；才发现我firstpipeline写错了。。。。

```
apue@ntsl: ~/airflow-spark/dags
apue@ntsl:~/airflow-spark$ ls
dags doc docker notebooks README.md spark
apue@ntsl:~/airflow-spark$ cd dags
apue@ntsl:~/airflow-spark/dags$ ls
airflow-create.txt fistpipeline.py __pycache__ spark-hello-world-module.py
apue@ntsl:~/airflow-spark/dags$
```

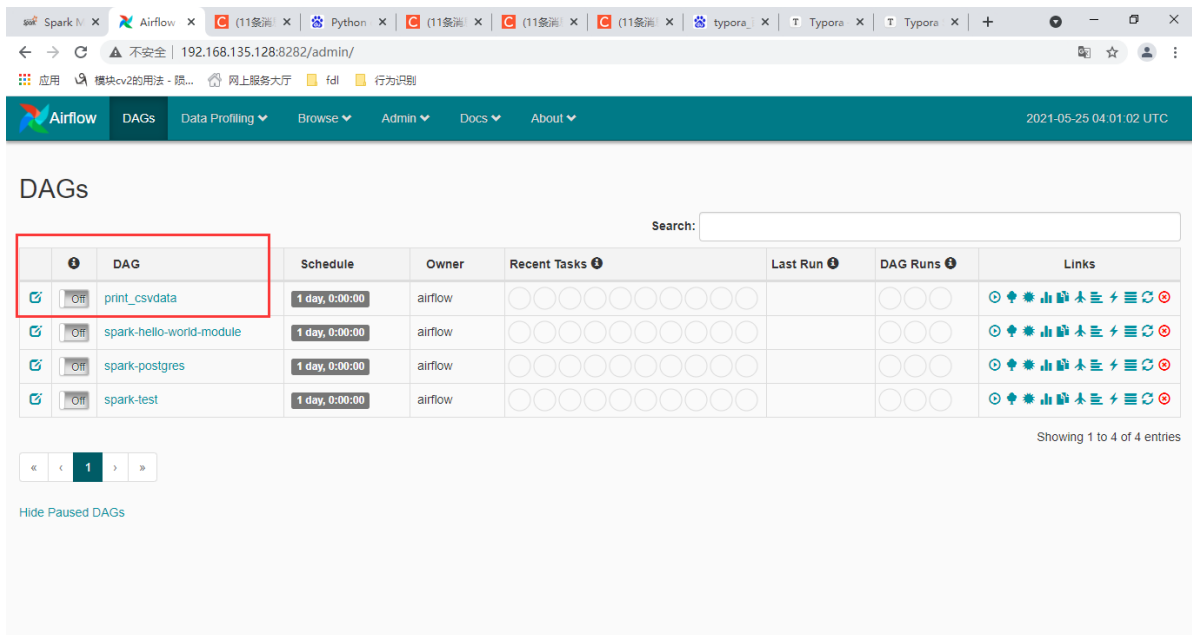
把容器启动起来进入容器看下是不是有这个东西：

```
apue@ntsl: ~
apue@ntsl:~$ sudo docker exec -it ebb24affe106 /bin/bash
[sudo] password for apue:
airflow@ebb24affe106:~$ pwd
/usr/local/airflow
airflow@ebb24affe106:~$ ls
airflow dags logs
airflow@ebb24affe106:~$ cd dags
airflow@ebb24affe106:~/dags$ ls
airflow-create.txt __pycache__ spark-postgres.py
fistpipeline.py spark-hello-world-module.py spark-test.py
airflow@ebb24affe106:~/dags$
```

usr/local/airflow/dags/文件夹下找到了这个写错名字脚本。

web查看任务

等待刷新，去web查看任务是否出现：



The screenshot shows the Airflow web interface with the 'DAGs' tab selected. A table lists the DAGs, with the first one, 'print_csvdata', highlighted by a red box. The table has columns for DAG, Schedule, Owner, Recent Tasks, Last Run, DAG Runs, and Links.

DAG	Schedule	Owner	Recent Tasks	Last Run	DAG Runs	Links
print_csvdata	1 day, 0:00:00	airflow				
spark-hello-world-module	1 day, 0:00:00	airflow				
spark-postgres	1 day, 0:00:00	airflow				
spark-test	1 day, 0:00:00	airflow				

找到了自己设置的DAG id: print_data;

一个坑

这里不知道是我自己虚拟机卡还是还是咋回事，这个web上dag脚本刷新很慢；经常我把脚本修改了一下，从web上删除这个print_data，再重新加载这个页面运行，它运行的不是我修改过的脚本，而是之前那个未修改的版本；这一度让我认为我改过的脚本也是错误的，后边看了运行日志才发现这个问题；所以每次我脚本有修改都要restart一下docker。

4: airflow运行

查看文件

运行前看看容器里我之前设置的路径下是否有个data.csv

```
apue@ntsl: ~/airflow-spark/docker
apue@ntsl:~/airflow-spark/docker$ sudo docker ps
CONTAINER ID        IMAGE               COMMAND
apue@ntsl: ~
airflow@ebb24affe106:/usr/local$ cd spark/
airflow@ebb24affe106:/usr/local/spark$ cd resources/
airflow@ebb24affe106:/usr/local/spark/resources$ cd data/
airflow@ebb24affe106:/usr/local/spark/resources/data$ ls
airflow.cfg  movies.csv  ratings.csv
airflow@ebb24affe106:/usr/local/spark/resources/data$
```

有一些csv文件，但没有我们的data.csv

运行脚本

接下来就是运行脚本了：web点击运行起来就可以了，去日志看一下运行结果；

task1

	DAG	Schedule	Owner	Recent Tasks	Last Run	DAG Runs	Links
On	print_csvdata	1 day, 0:00:00	airflow	2	2021-05-24 00:00	2	🔍 📊 📄 🔗 🔄 🛑
On	spark-hello-world-module	1 day, 0:00:00	airflow				🔍 📊 📄 🔗 🔄 🛑
On	spark-postgres	1 day, 0:00:00	airflow				🔍 📊 📄 🔗 🔄 🛑
On	spark-test	1 day, 0:00:00	airflow				🔍 📊 📄 🔗 🔄 🛑

Showing 1 to 4 of 4 entries

很快，脚本运行完成；看下日志

```
Log by attempts
1
Toggle wrap Jump to end

*** Reading local file: /usr/local/airflow/logs/print_csvdata/get_csv/2021-05-24T00:00:00+00:00/1.log
[2021-05-25 04:15:54,898] {{taskinstance.py:655}} INFO - Dependencies all met for <TaskInstance: print_csvdata.get_csv 2021-05-24T00:00:00+00:00 [queued]>
[2021-05-25 04:15:54,917] {{taskinstance.py:655}} INFO - Dependencies all met for <TaskInstance: print_csvdata.get_csv 2021-05-24T00:00:00+00:00 [queued]>
[2021-05-25 04:15:54,917] {{taskinstance.py:866}} INFO -
-----
[2021-05-25 04:15:54,917] {{taskinstance.py:867}} INFO - Starting attempt 1 of 2
[2021-05-25 04:15:54,917] {{taskinstance.py:868}} INFO -
-----
[2021-05-25 04:15:54,932] {{taskinstance.py:887}} INFO - Executing <Task(PythonOperator): get_csv> on 2021-05-24T00:00:00+00:00
[2021-05-25 04:15:54,936] {{standard_task_runner.py:52}} INFO - Started process 11721 to run task
[2021-05-25 04:15:55,000] {{logging_mixin.py:112}} INFO - [2021-05-25 04:15:55,073] {{dagbag.py:403}} INFO - Filling up the DagBag from /usr/local/airflow/dags/fistpipeline.py
[2021-05-25 04:15:55,164] {{logging_mixin.py:112}} INFO - Running %s on host %s <TaskInstance: print_csvdata.get_csv 2021-05-24T00:00:00+00:00 [running]> ebb24affe106
[2021-05-25 04:15:55,370] {{python_operator.py:105}} INFO - Exporting the following env vars:
AIRFLOW_CTX_DAG_EMAIL=huan.chen@kyig.org
AIRFLOW_CTX_DAG_OWNER=airflow
AIRFLOW_CTX_DAG_ID=print_csvdata
AIRFLOW_CTX_TASK_ID=get_csv
AIRFLOW_CTX_EXECUTION_DATE=2021-05-24T00:00:00+00:00
AIRFLOW_CTX_DAG_RUN_ID=scheduled_2021-05-24T00:00:00+00:00
[2021-05-25 04:15:55,414] {{logging_mixin.py:112}} INFO - *****已经成功连接'ks'服务器FTP服务!!!
[2021-05-25 04:15:55,513] {{logging_mixin.py:112}} INFO - 220 Microsoft FTP Service
[2021-05-25 04:15:55,531] {{python_operator.py:114}} INFO - Done, returned value was: None
[2021-05-25 04:16:04,864] {{logging_mixin.py:112}} INFO - [2021-05-25 04:16:04,863] {{local_task_job.py:103}} INFO - Task exited with return code 0
```

忘了写下载完成的提示了，去容器看下data.csv在不在：

```
apue@ntsl: ~/airflow-spark/docker
apue@ntsl:~/airflow-spark/docker$ sudo docker ps
CONTAINER ID        IMAGE               COMMAND
apue@ntsl: ~
airflow@ebb24affe106:/usr/local$ cd spark/
airflow@ebb24affe106:/usr/local/spark$ cd resources/
airflow@ebb24affe106:/usr/local/spark/resources$ cd data/
airflow@ebb24affe106:/usr/local/spark/resources/data$ ls
airflow.cfg  movies.csv  ratings.csv
airflow@ebb24affe106:/usr/local/spark/resources/data$ ls
airflow.cfg  data.csv  movies.csv  ratings.csv
airflow@ebb24affe106:/usr/local/spark/resources/data$
```

和上次的ls输出结果相比，多了data.csv说明下载文件成功了；

task2

看下task2的日志：

```
Log by attempts
1
Toggle wrap Jump to end

*** Reading local file: /usr/local/airflow/logs/print_csvdata/print_data/2021-05-24T00:00:00+00:00/1.log
[2021-05-25 04:16:00,911] {{taskinstance.py:655}} INFO - Dependencies all met for <TaskInstance: print_csvdata.print_data 2021-05-24T00:00:00+00:00 [queued]>
[2021-05-25 04:16:00,978] {{taskinstance.py:655}} INFO - Dependencies all met for <TaskInstance: print_csvdata.print_data 2021-05-24T00:00:00+00:00 [queued]>
[2021-05-25 04:16:00,978] {{taskinstance.py:866}} INFO -
-----
[2021-05-25 04:16:00,978] {{taskinstance.py:867}} INFO - Starting attempt 1 of 2
[2021-05-25 04:16:00,978] {{taskinstance.py:868}} INFO -
-----
[2021-05-25 04:16:01,064] {{taskinstance.py:887}} INFO - Executing <Task(PythonOperator): print_data> on 2021-05-24T00:00:00+00:00
[2021-05-25 04:16:01,122] {{standard_task_runner.py:52}} INFO - Started process 11794 to run task
[2021-05-25 04:16:01,421] {{logging_mixin.py:112}} INFO - [2021-05-25 04:16:01,420] {{dagbag.py:403}} INFO - Filling up the DagBag from /usr/local/airflow/dags/fistpipeline.py
[2021-05-25 04:16:01,506] {{logging_mixin.py:112}} INFO - Running %s on host %s <TaskInstance: print_csvdata.print_data 2021-05-24T00:00:00+00:00 [running]> ebb24affe106
[2021-05-25 04:16:01,595] {{python_operator.py:105}} INFO - Exporting the following env vars:
AIRFLOW_CTX_DAG_EMAIL=huan.chen@kyig.org
AIRFLOW_CTX_DAG_OWNER=airflow
AIRFLOW_CTX_DAG_ID=print_csvdata
AIRFLOW_CTX_TASK_ID=print_data
AIRFLOW_CTX_EXECUTION_DATE=2021-05-24T00:00:00+00:00
AIRFLOW_CTX_DAG_RUN_ID=scheduled_2021-05-24T00:00:00+00:00
[2021-05-25 04:16:01,662] {{logging_mixin.py:112}} INFO - 陈欢
[2021-05-25 04:16:01,662] {{logging_mixin.py:112}} INFO - test
[2021-05-25 04:16:01,662] {{logging_mixin.py:112}} INFO - csv
[2021-05-25 04:16:01,662] {{python_operator.py:114}} INFO - Done, returned value was: None
[2021-05-25 04:16:10,821] {{logging_mixin.py:112}} INFO - [2021-05-25 04:16:10,821] {{local_task_job.py:103}} INFO - Task exited with return code 0
```

成功输出了放在ftp服务器上的data.csv的内容：陈欢， test， csv

三：总结

任务流程基本上跑通了，但是许多细节并不完善；

- airflow运行原理，一些参数的含义，和spark协调的分布式调度等基础知识还需要补充
- airflow基本代码架构能理解，但并不能熟练的编写应用级脚本
- 在实现的过程中每一步走得都很慢，不仅对个过程要有理解，对于上述过程提到的坑等各处的细节还需要不断加强理解，知道为什么。