

# airflow Variable尝试

## 任务描述

根据老师给出的指导意见，更改上次代码中airflow脚本中链接服务器时获取用户名和密码的方式；在上次的代码中是直接给出了用户名和密码，但这样的方式其实在实际生产时是根本不可行的；我们需要对用户名和登录密码这一类敏感信息提供一定的保护。

所以在改进的方式中，用airflow中的admin下得Variable定义出用户名和密码，相当于给出了全局变量，每个task，每个dags都可以访问；不仅更加安全，而且也更加高效，方便维护。

## 任务过程

首先，在Variable中定义出用户名和密码信息，然后脚本中通过Variable模块获取就可以了

首先，创建信息，我是以json格式给出了用户名和密码：

The screenshot shows the Airflow web interface. The top navigation bar includes 'Airflow', 'DAGs', 'Data Profiling', 'Browse', 'Admin', 'Docs', and 'About'. The 'Admin' menu is highlighted with a red box. Below the navigation bar, the 'Variable [create]' page is displayed. It has two tabs: 'List' and 'Create'. The 'Create' tab is active. The form has two main fields: 'Key' and 'Val'. The 'Key' field is labeled 'login\_info' and the 'Val' field contains the JSON string: `{ "user_name": "chenhuan", "password": "213012" }`. Both the 'Admin' menu and the form fields are highlighted with red boxes.

在脚本中，根据定义获取相应的用户名和密码，修改脚本代码如下：

```

14 # 导入依赖库
15 import airflow
16 from airflow import DAG
17 from airflow.models import Variable
18 # 导入特定的执行器
19 from airflow.operators.bash_operator import BashOperator
20 from airflow.operators.python_operator import PythonOperator
21 from datetime import timedelta
22 import socket
23 import csv
24 import os
25 from ftplib import FTP
26
27
28 # these args will get passed on to each operator
29 # you can override them on a per-task basis during operator initialization
30 # 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
31
32 login_info = Variable.get("login_info", deserialize_json=True)
33
34 default_args = {
35     'owner': 'airflow',
36     'depends_on_past': False,
37     'start_date': airflow.utils.dates.days_ago(2),
38     'email': ['huan.chen@kylg.org'],
39     'email_on_failure': False,
40     'email_on_retry': False,
41     'retries': 1,
42     'retry_delay': timedelta(minutes=5),
43     'user_name': login_info['user_name'],
44     'password': login_info['password'],

```

创建完成之后直接启动运行就可以了

The screenshot shows the Apache Airflow web interface at 192.168.135.128:8282/admin/. The 'DAGs' tab is active, displaying a table of DAGs. The 'Recent Tasks' column for the 'print\_csvdata' DAG is highlighted with a red box, showing a green circle indicating successful completion. The table lists four DAGs: 'print\_csvdata', 'spark-hello-world-module', 'spark-postgres', and 'spark-test'. The 'print\_csvdata' DAG has a 'Last Run' of 2021-05-30 00:00 and 'DAG Runs' of 1. The 'spark-hello-world-module' DAG has a 'Last Run' of 2021-05-31 07:23:54 UTC and 'DAG Runs' of 1. The 'spark-postgres' DAG has a 'Last Run' of 2021-05-31 07:23:54 UTC and 'DAG Runs' of 1. The 'spark-test' DAG has a 'Last Run' of 2021-05-31 07:23:54 UTC and 'DAG Runs' of 1.

DAG	Schedule	Owner	Recent Tasks	Last Run	DAG Runs	Links
print_csvdata	1 day, 0:00:00	airflow	1 (Success)	2021-05-30 00:00	1	View Details
spark-hello-world-module	1 day, 0:00:00	airflow	1 (Success)	2021-05-31 07:23:54 UTC	1	View Details
spark-postgres	1 day, 0:00:00	airflow	1 (Success)	2021-05-31 07:23:54 UTC	1	View Details
spark-test	1 day, 0:00:00	airflow	1 (Success)	2021-05-31 07:23:54 UTC	1	View Details

很快，运行完成了