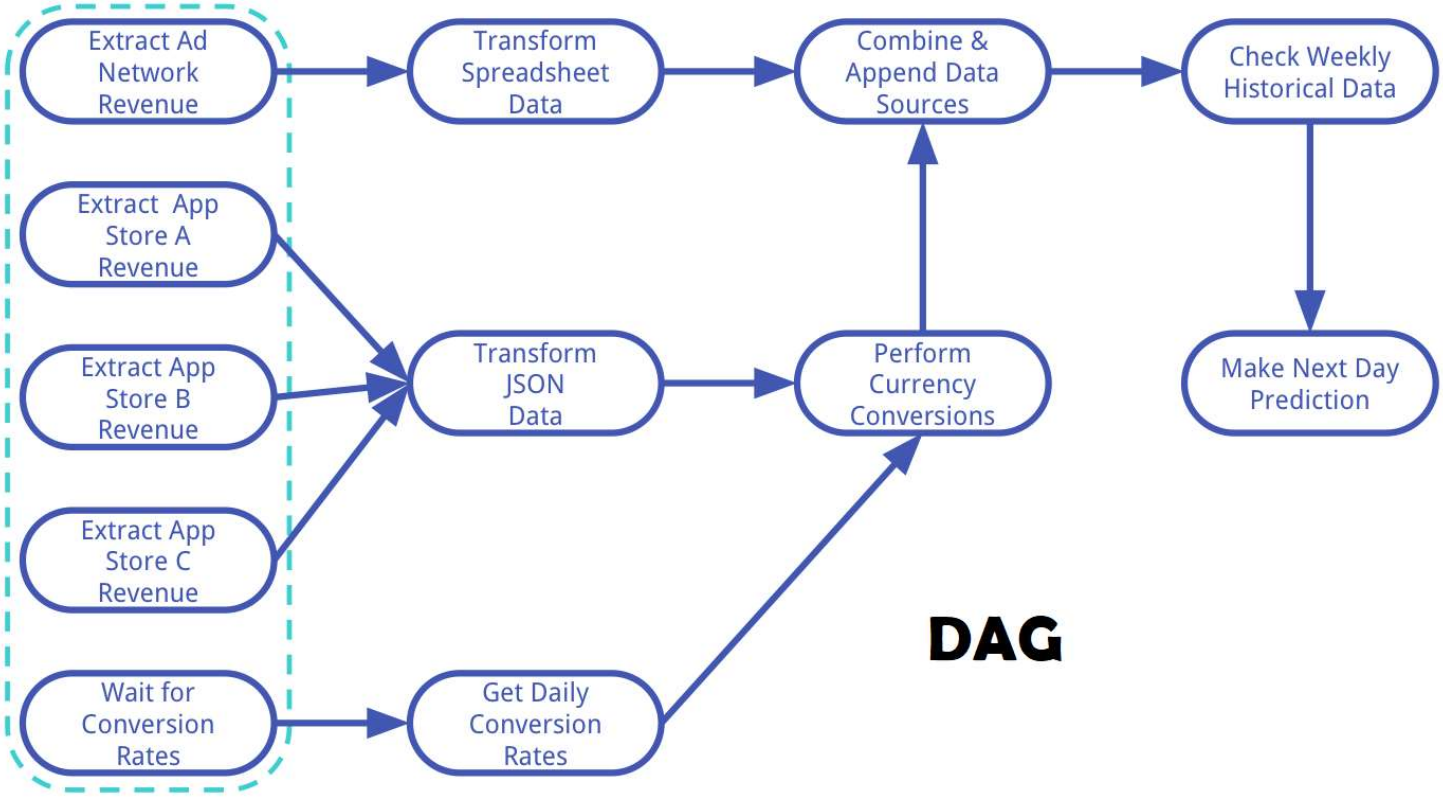
# Apache Airflow

## نرم افزار مدیریت
## جریان‌های کار

# آنچه خواهیم دید

۱. آشنایی با مفاهیم پایه جریان‌های کار
۲. روش سنتی زمان‌بندی کارها (کران جابز)
۳. مفاهیم پایه ایرفلو و معماری آن
۴. بررسی محیط گرافیکی ایرفلو
۵. کارگاه عملی / کار با کران‌جابز
۶. کارگاه عملی / کار با ایرفلو

# مدیریت جریان کار



# ویژگیهای مورد نیاز WMS

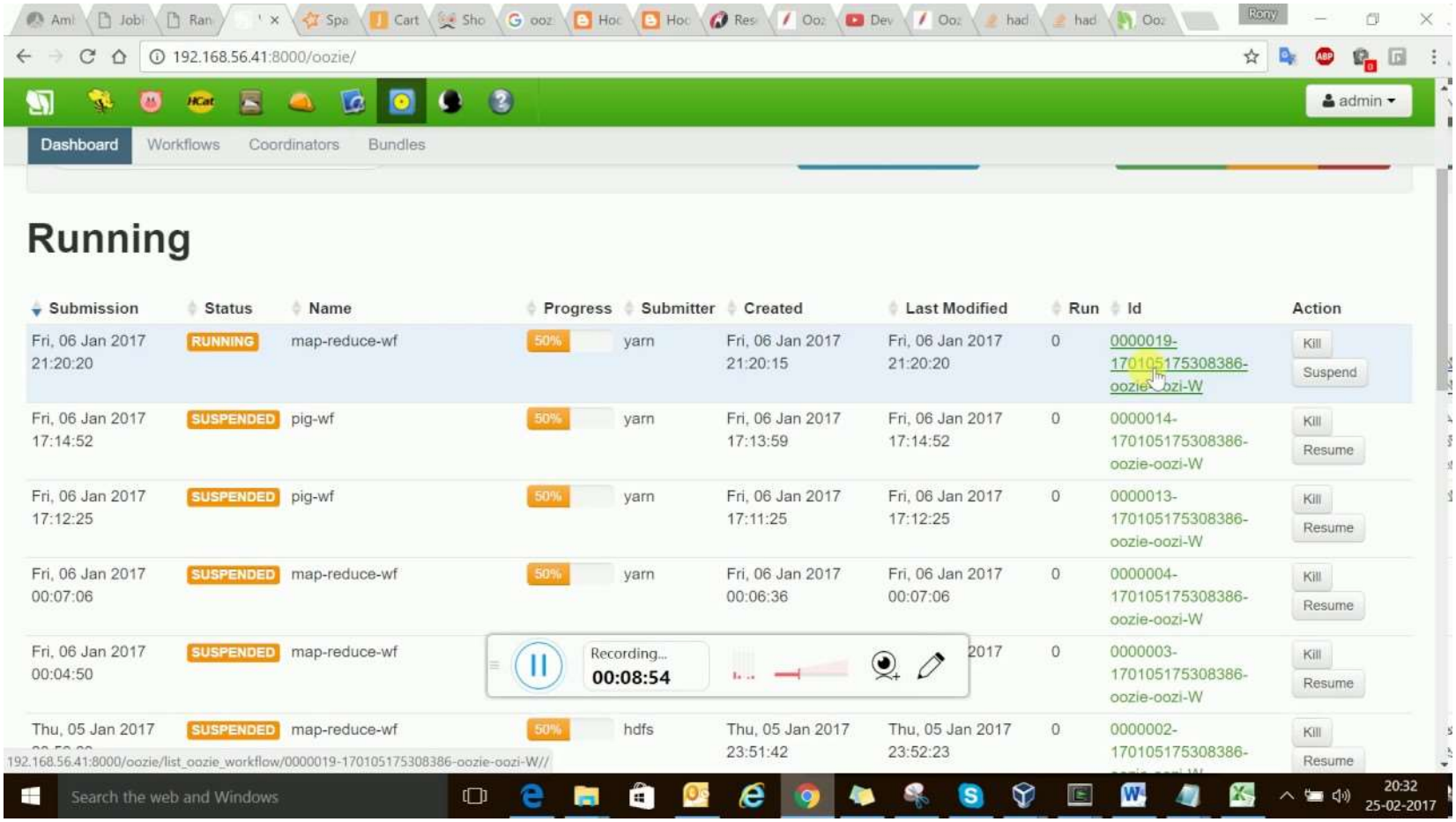
- محیط گرافیکی (در کنار خط فرمان قدرتمند)
- انعطاف در تعریف خطوط پردازش داده
- SLA
- Backfill/Catch Up
- Metrics/Logs
- Alerting/ Tunable Retries
- قابل توسعه
- مقیاس پذیر / اجرای موازی تسک‌ها
- مجموعه غنی از عملگرها و توابع آماده
- جامعه کاربری فعال
- زمان بند

# بازیگران اصلی

| Name | Originated at | Workflows defined in | Written in | Scheduling | Backfilling | User interface[2] |
|---|---|---|---|---|---|---|
| Airflow | Airbnb | Python | Python | Yes | Yes | Yes |
| Argo | Applatix | YAML | Go | 3rd party[3] | | Yes |
| Azkaban | LinkedIn | YAML | Java | Yes | No | Yes |
| Conductor | Netflix | JSON | Java | No | | Yes |
| Luigi | Spotify | Python | Python | No | Yes | Yes |
| Make | | Custom DSL | C | No | No | No |
| Metaflow | Netflix | Python | Python | No | | No |
| Oozie | | XML | Java | Yes | Yes | Yes |

**Source :**
Data Pipelines With Apache Airflow

# Apache Oozie



# Linkedin Azkaban

Azkaban Local
${version}
My Local Azkaban

Projects | Scheduling | Executing | History | Flow Trigger Schedule | azkaban

Flow flow_trigger

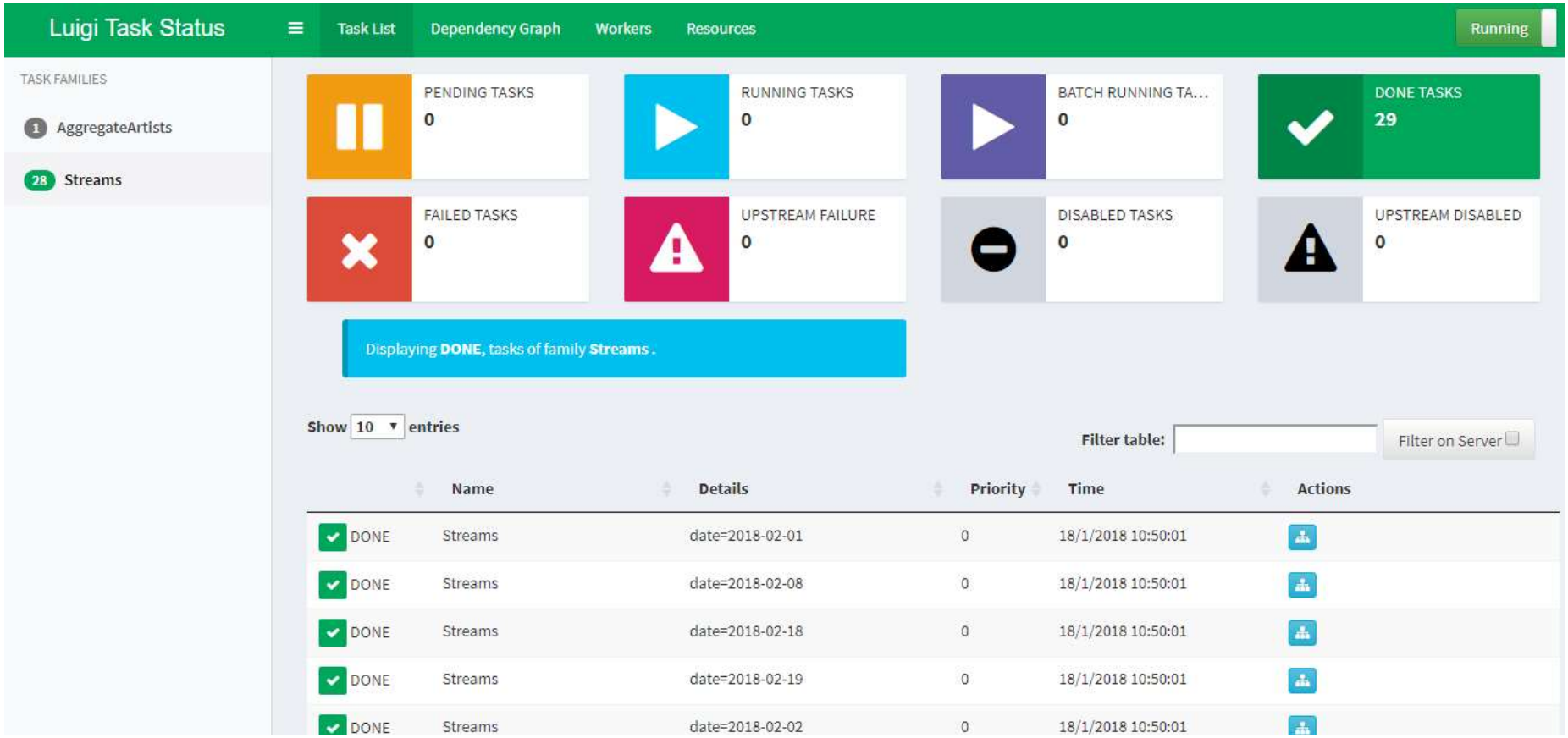
Schedule / Execute Flow

Project ET / Flow flow_trigger

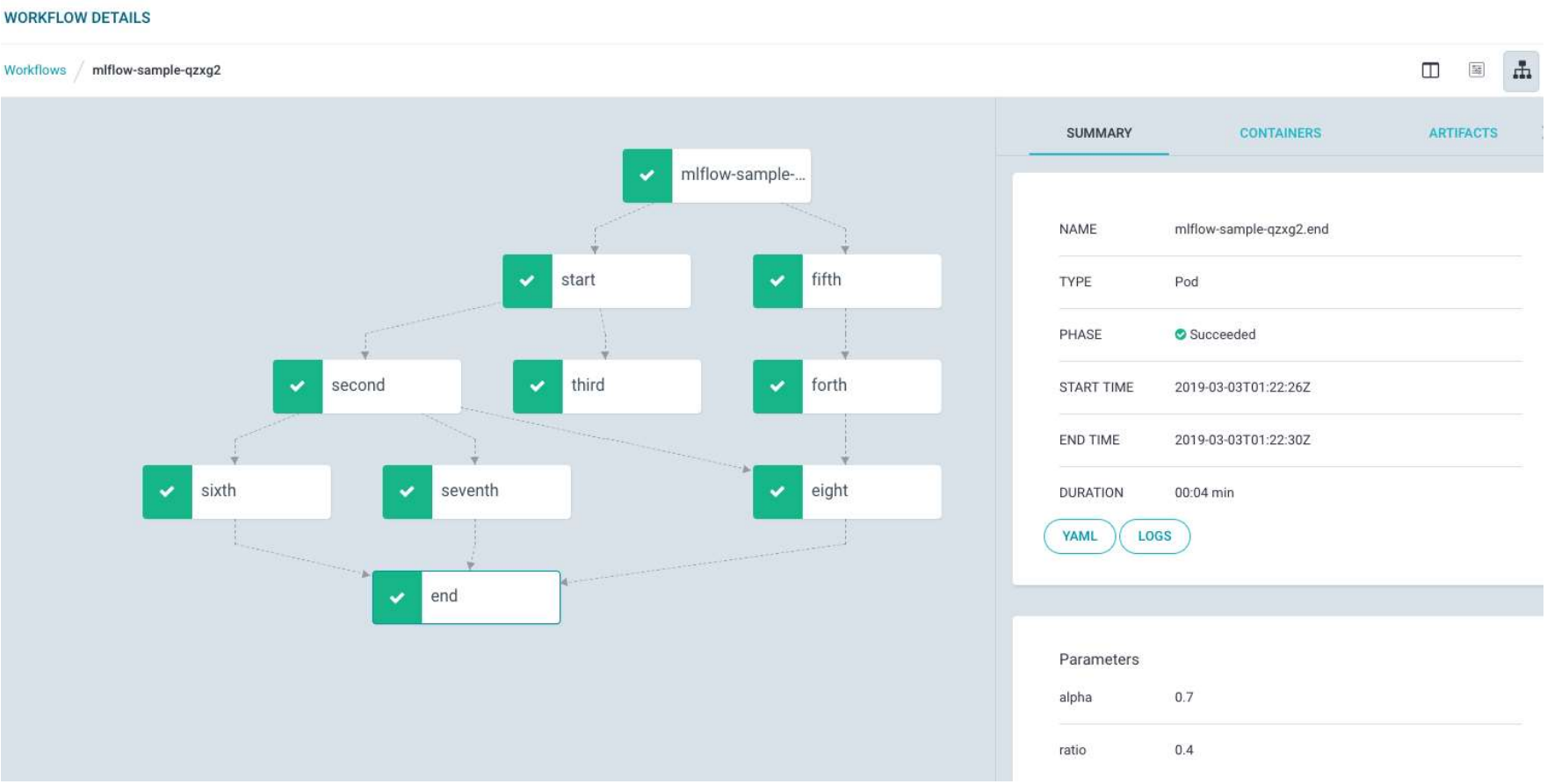
Graph | Executions | Flow Triggers | Summary

| Flow Trigger Instance Id | Submitting user | Start Time | End Time | Elapsed | Status | Action |
|---|---|---|---|---|---|---|
| acd4e5b1-21cd-41d1-b4e9-ddd1e4ddd9c7 | azkaban | 2018-07-31 17:22 00s | - | 29 sec | RUNNING | |
| 340392ba-263f-4cba-b578-32130bbcdd4d | azkaban | 2018-07-31 17:21 25s | - | 1m 3s | RUNNING | |
| 39b6c9fe-5528-491e-8b16-bfc3681d596f | azkaban | 2018-07-30 17:28 00s | - | 23h 54m 29s | RUNNING | |
| 296e76a1-6b6c-4660-9d89-8e2acd101a79 | azkaban | 2018-07-30 17:26 47s | 2018-07-30 17:28 10s | 1m 23s | SUCCEEDED | |
| 6b43f3e3-4722-4d1f-b692-a634c6a8fa41 | azkaban | 2018-07-30 11:56 00s | 2018-07-30 17:28 10s | 5h 32m 10s | SUCCEEDED | |
| 0aa7d799-7db1-4bd0-895d-6c4d26d4a286 | azkaban | 2018-07-30 11:54 00s | 2018-07-30 17:28 10s | 5h 34m 10s | SUCCEEDED | |

# Luigi


Luigi Task Status
Task List
Dependency Graph
Workers
Resources
Running
TASK FAMILIES
1 AggregateArtists
28 Streams
PENDING TASKS 0
RUNNING TASKS 0
BATCH RUNNING TA... 0
DONE TASKS 29
FAILED TASKS 0
UPSTREAM FAILURE 0
DISABLED TASKS 0
UPSTREAM DISABLED 0
Displaying DONE, tasks of family Streams .
Show 10 entries
Filter table:
Filter on Server
Name
Details
Priority
Time
Actions
DONE Streams date=2018-02-01 0 18/1/2018 10:50:01
DONE Streams date=2018-02-08 0 18/1/2018 10:50:01
DONE Streams date=2018-02-18 0 18/1/2018 10:50:01
DONE Streams date=2018-02-19 0 18/1/2018 10:50:01
DONE Streams date=2018-02-02 0 18/1/2018 10:50:01

# Argo


WORKFLOW DETAILS
Workflows
mlflow-sample-qzxg2
mlflow-sample-...
start
fifth
second
third
forth
sixth
seventh
eight
end
SUMMARY
CONTAINERS
ARTIFACTS
NAME
mlflow-sample-qzxg2.end
TYPE
Pod
PHASE
Succeeded
START TIME
2019-03-03T01:22:26Z
END TIME
2019-03-03T01:22:30Z
DURATION
00:04 min
YAML
LOGS
Parameters
alpha
0.7
ratio
0.4

# مفاهیم پایه ایرفلو

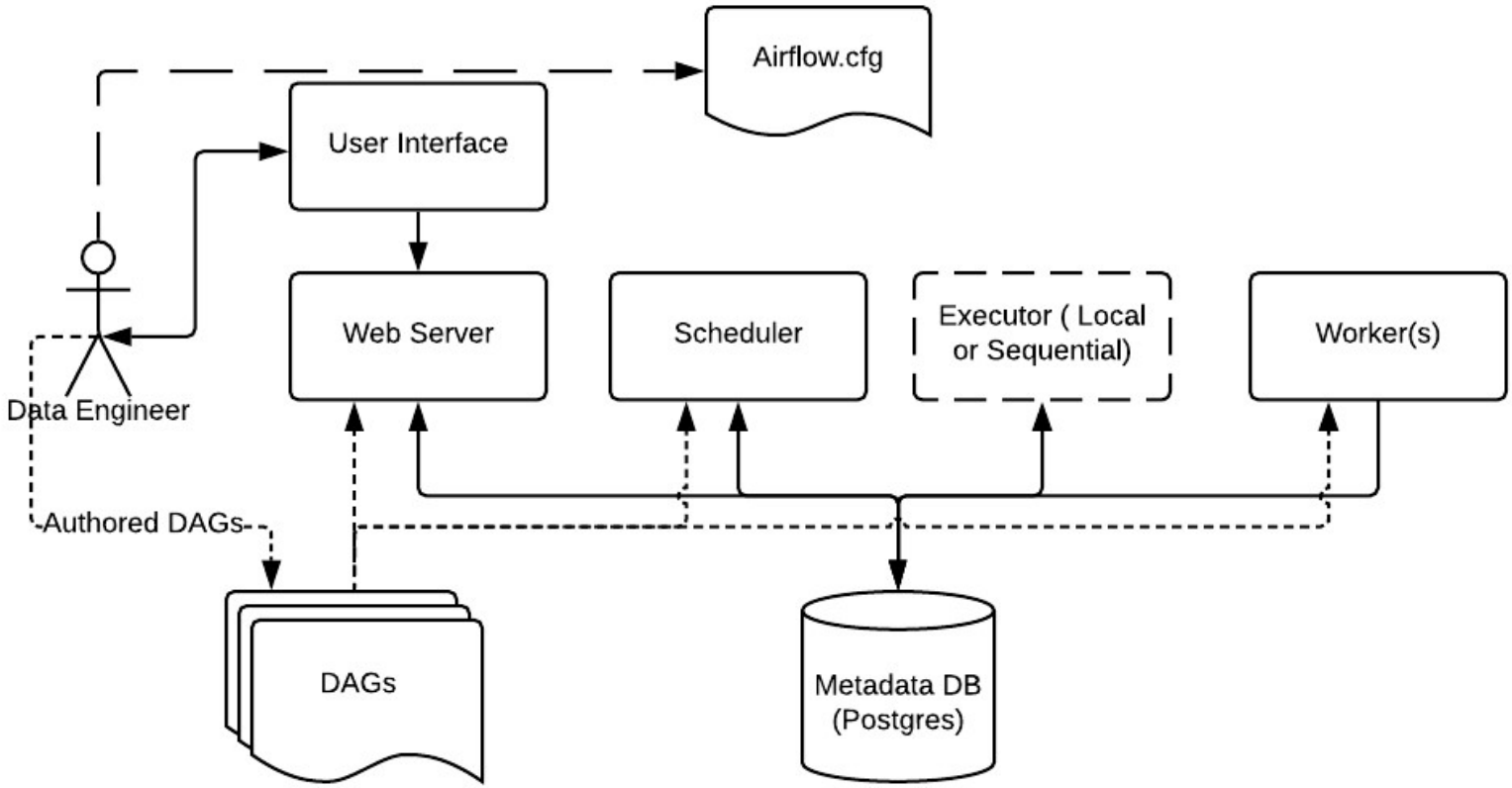
- **DAG**: graph of operator usages (=tasks)

- **Operator**: "Transformation" step
  - **Sensor**: Operator which polls with frequency / timeout (e.g. LocalFileSensor)
  - **Executor**: Trigger operation (e.g. HiveOperator, BashOperator, PigOperator, ...)

- **Task**: Usage of Operator in DAG
  - **Task Instance**: run of a Task at a point in time

- **Hook**: Interface to external System (JDBCHook, HTTPHook, ...)

# معماری ایرفلو



# انواع زیرساخت‌های اجرای تسک‌ها

DASK

Apache Airflow

Celery

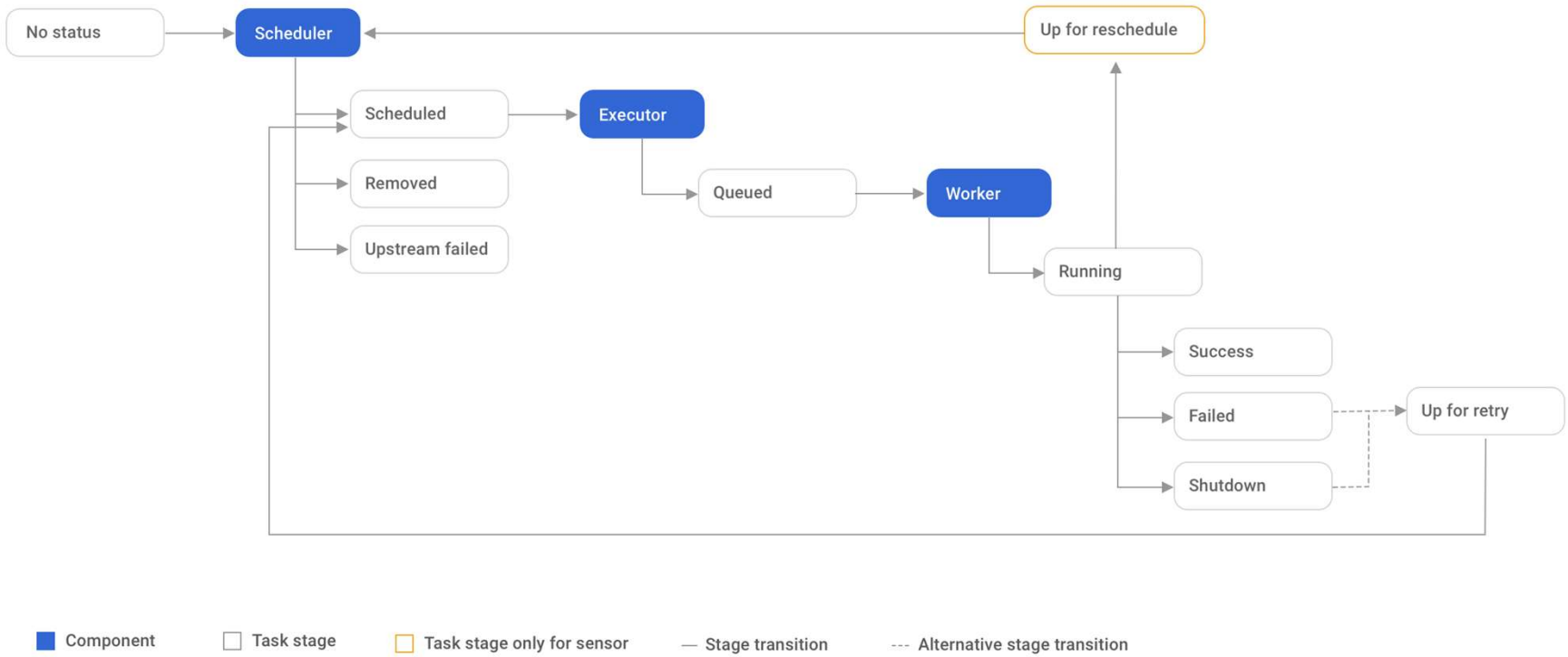
LocalExecutor

SequentialExecutor

DebugExecutor

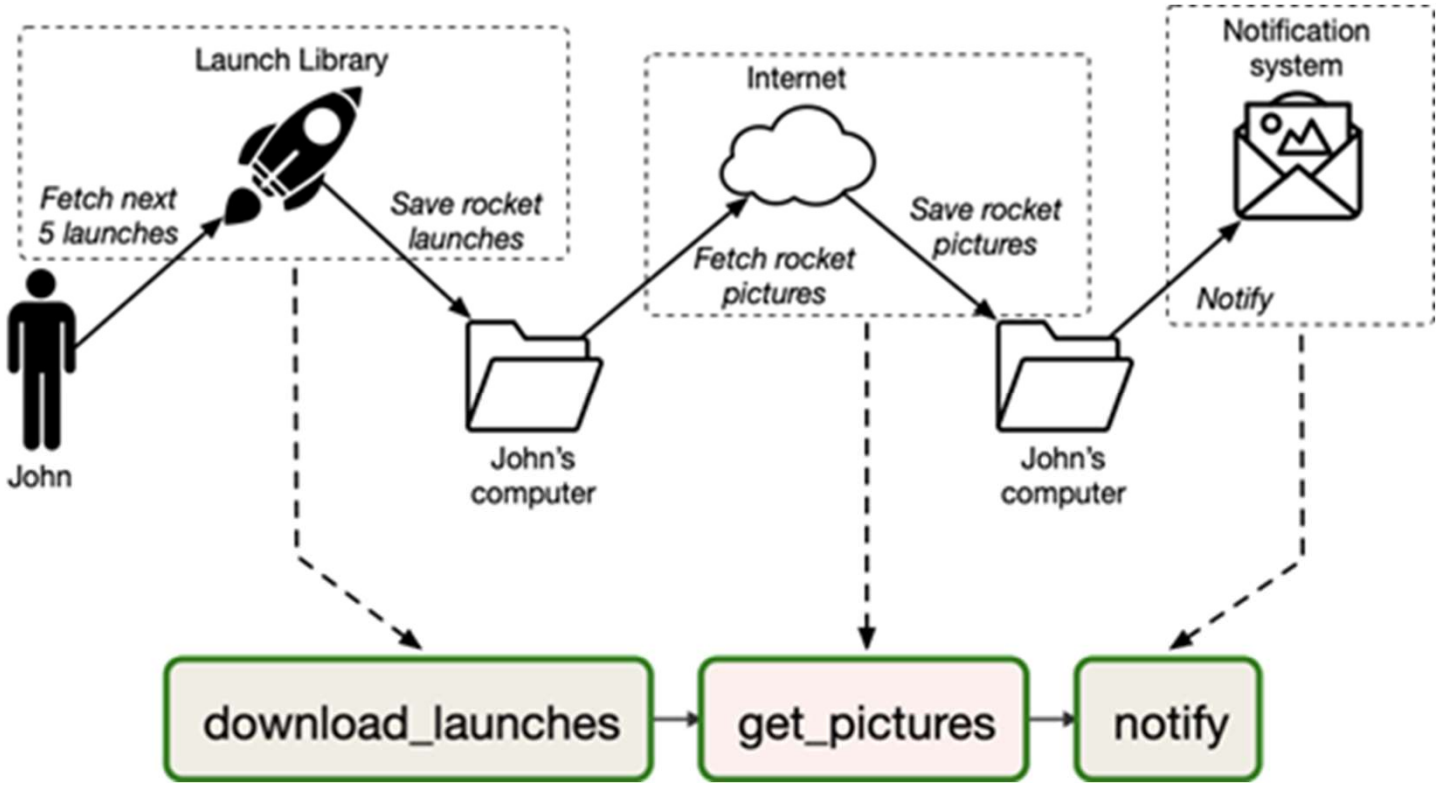
Apache MESOS

# چرخه حیات یک تسک




success running failed skipped upstream_failed up_for_reschedule up_for_retry queued no_status

# بررسی یک مثال



# تعریف جریان کار

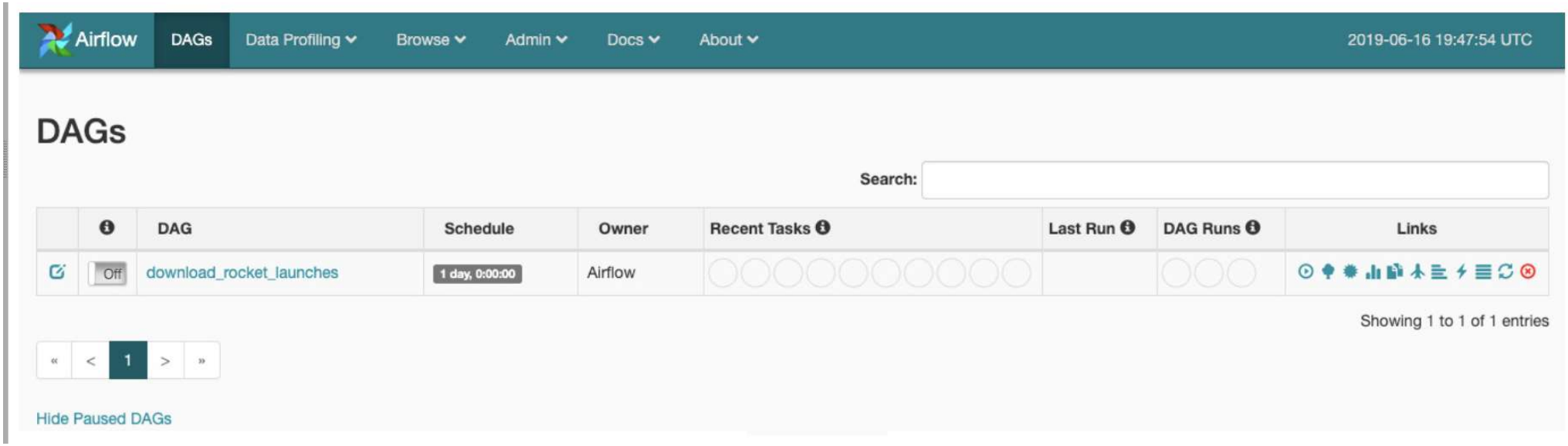
Listing 2.2 DAG for downloading and processing rocket launch data

```
1  import json
2  import pathlib
3
4  import airflow
5  import requests
6  from airflow import DAG
7  from airflow.operators.bash_operator import BashOperator
8  from airflow.operators.python_operator import PythonOperator
9
10 dag = DAG(                                                          A
11     dag_id="download_rocket_launches",                              B
12     start_date=airflow.utils.dates.days_ago(14),                    C
13     schedule_interval=None,                                         D
14 )
15
16 download_launches = BashOperator(                                   E
17     task_id="download_launches",                                    F
18     bash_command="curl -o /tmp/launches.json 'https://launchlibrary.net/1.4/launch
19     dag=dag,
20 )
```

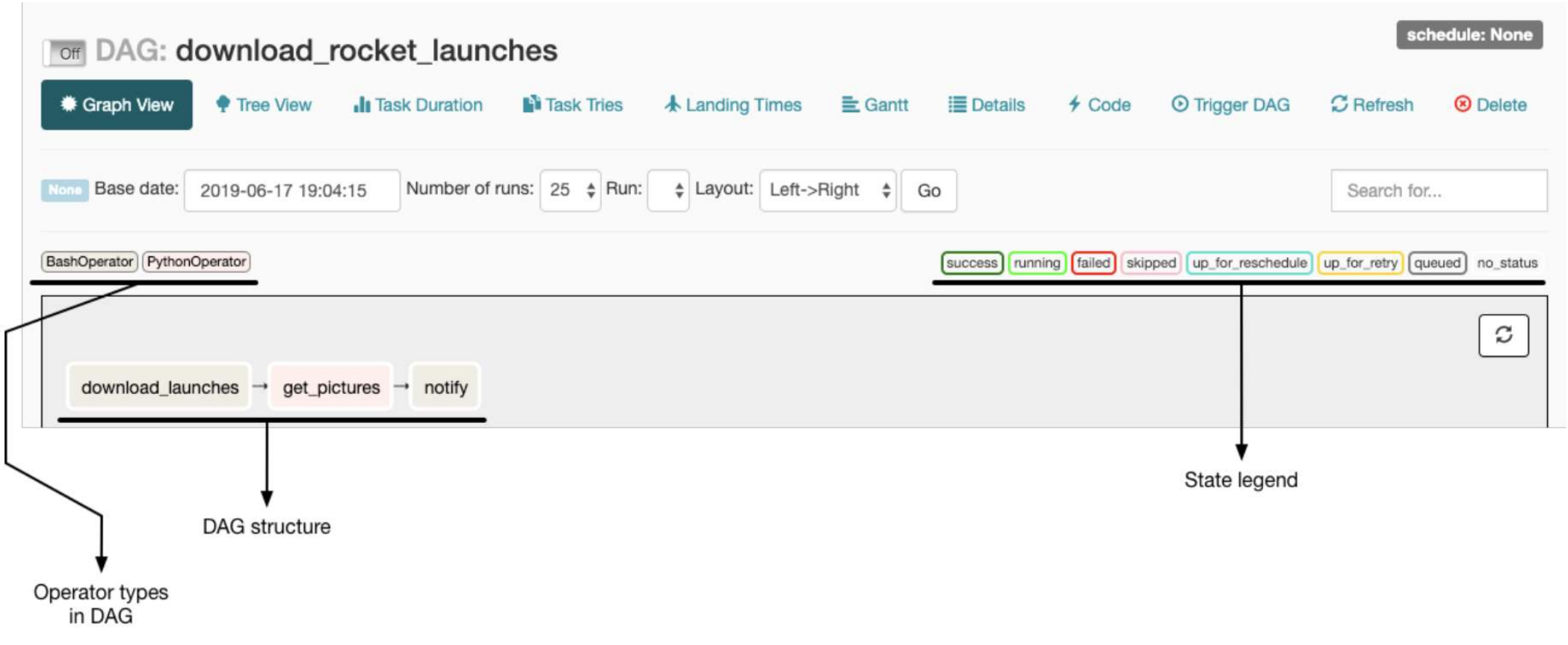
# تعریف جریان کار

```
get_pictures = PythonOperator(
    task_id="get_pictures",
    python_callable=_get_pictures,
    dag=dag,
)

notify = BashOperator(
    task_id="notify",
    bash_command='echo "There are now $(ls /tmp/images/ | wc -l) images."',
    dag=dag,
)

download_launches >> get_pictures >> notify
```

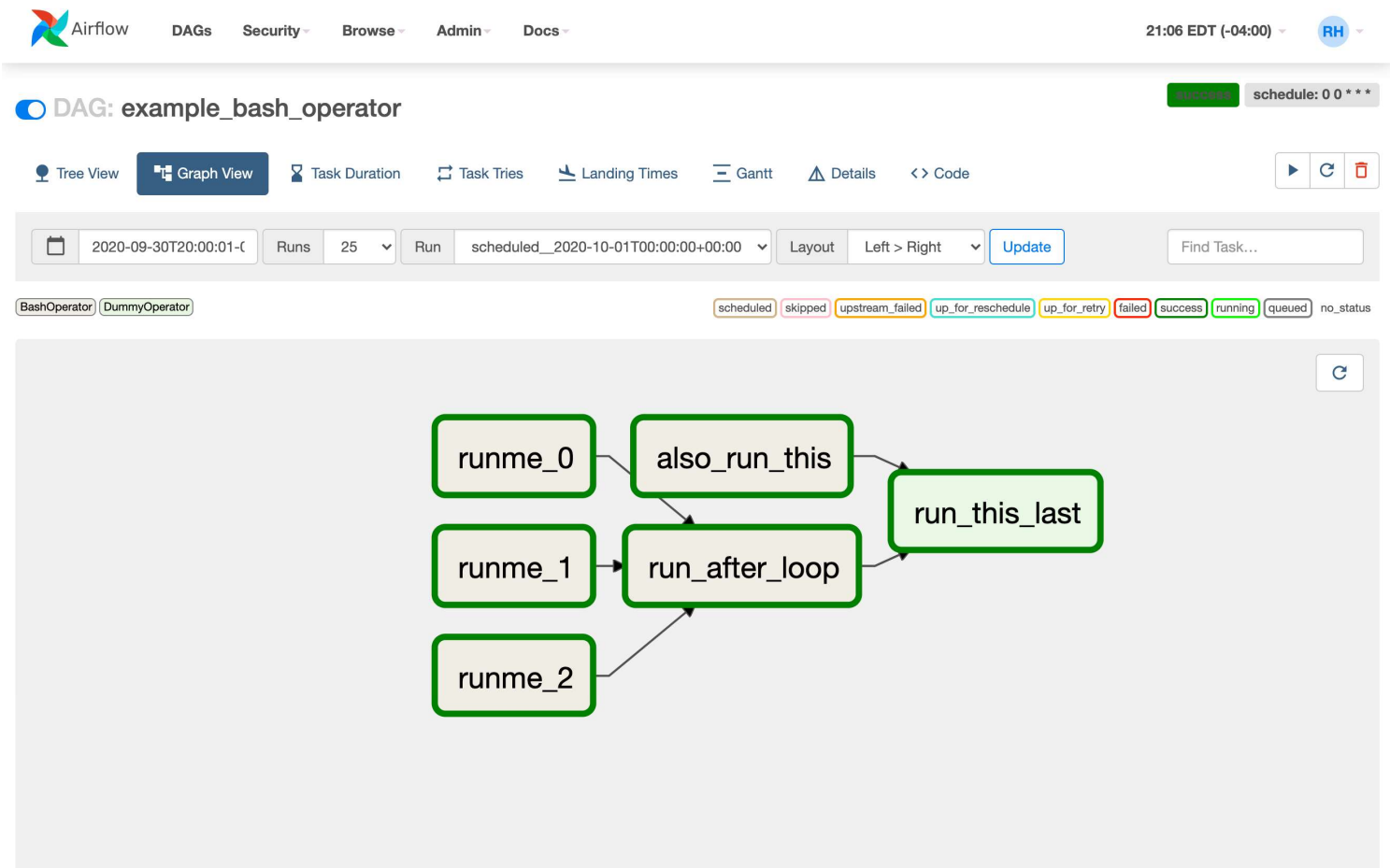
# بررسی محیط ایرفلو



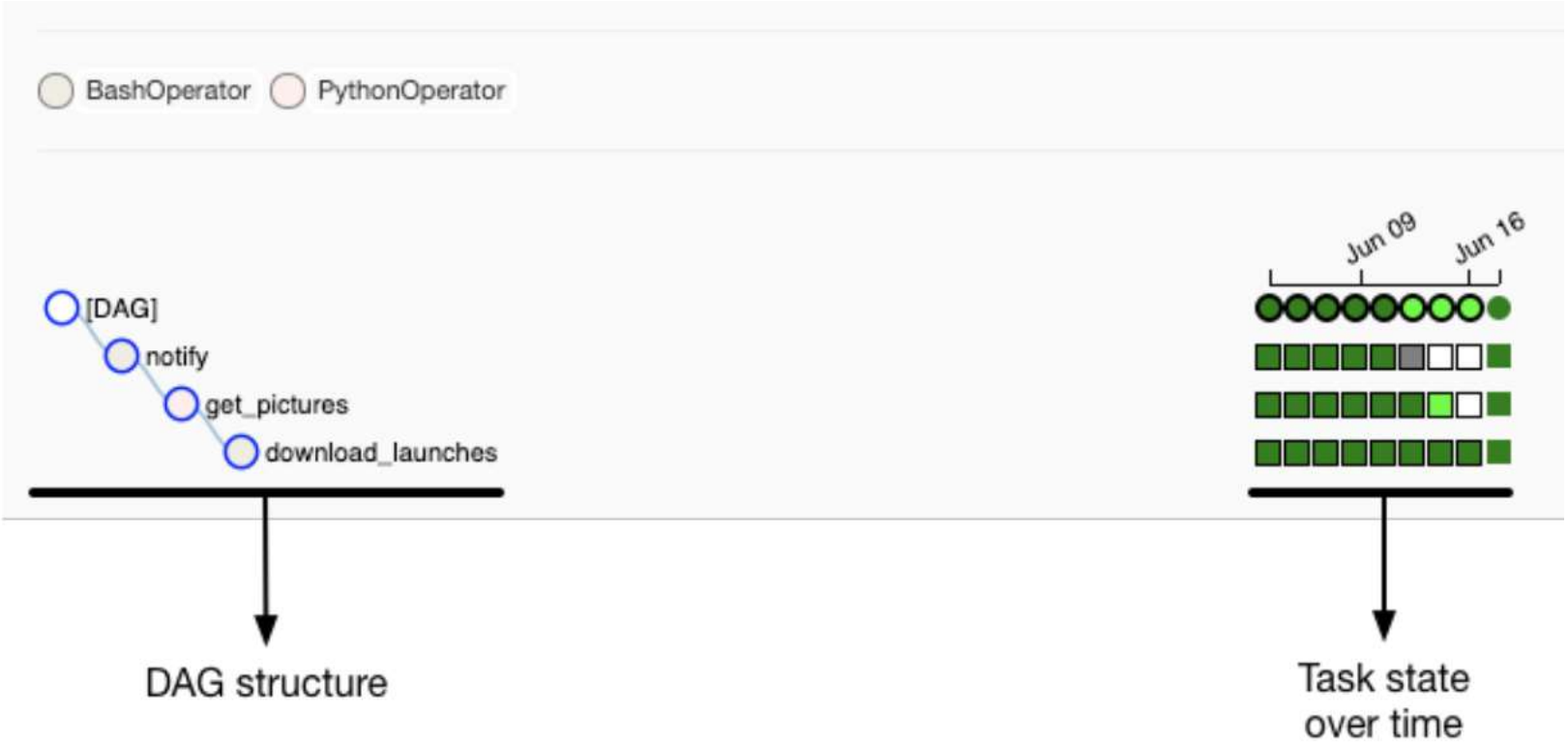
# Graph View


Off
DAG: download_rocket_launches
schedule: None
Graph View
Tree View
Task Duration
Task Tries
Landing Times
Gantt
Details
Code
Trigger DAG
Refresh
Delete
None
Base date:
2019-06-17 19:04:15
Number of runs:
25
Run:
Layout:
Left->Right
Go
Search for...
BashOperator
PythonOperator
success
running
failed
skipped
up_for_reschedule
up_for_retry
queued
no_status
download_launches
get_pictures
notify
State legend
DAG structure
Operator types
in DAG

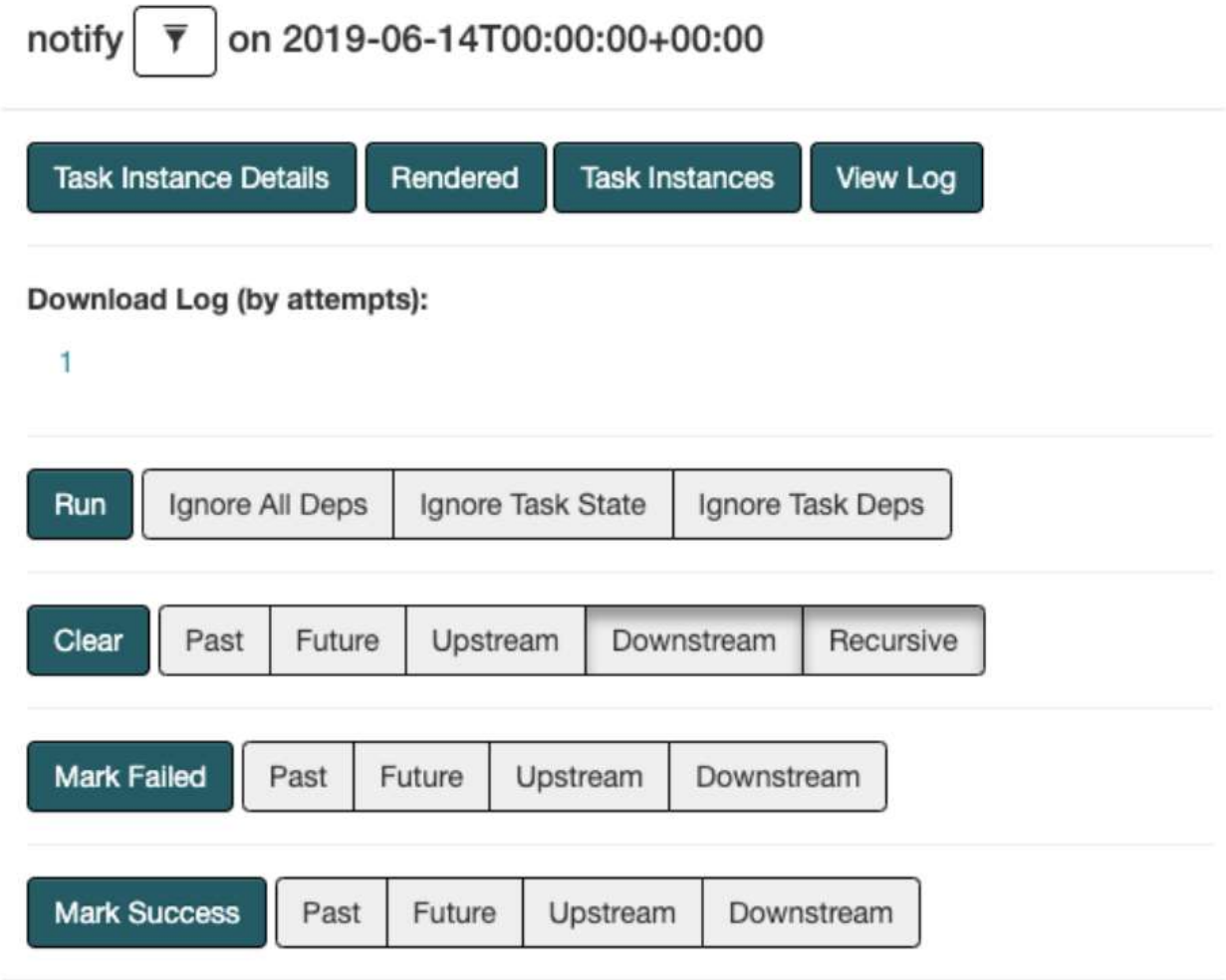
# Graph View


Airflow
DAGs
Security
Browse
Admin
Docs
21:06 EDT (-04:00)
RH
DAG: example_bash_operator
schedule: 0 0 * * *
Tree View
Graph View
Task Duration
Task Tries
Landing Times
Gantt
Details
Code
2020-09-30T20:00:01-(
Runs
25
Run
scheduled__2020-10-01T00:00:00+00:00
Layout
Left > Right
Update
Find Task...
BashOperator
DummyOperator
scheduled
skipped
upstream_failed
up_for_reschedule
up_for_retry
failed
success
running
queued
no_status
runme_0
also_run_this
run_this_last
runme_1
run_after_loop
runme_2

# Tree View


BashOperator
PythonOperator
[DAG]
notify
get_pictures
download_launches
Jun 09
Jun 16
DAG structure
Task state
over time

# Task Details



# Task Details

```
*** Reading local file: /root/airflow/logs/download_rocket_launches/notify/2019-06-18T19:06:28.102026+00:00/1.log
[2019-06-18 19:06:58,698] {__init__.py:1139} INFO - Dependencies all met for <TaskInstance: download_rocket_launches.notify 2019-06-18T
[2019-06-18 19:06:58,705] {__init__.py:1139} INFO - Dependencies all met for <TaskInstance: download_rocket_launches.notify 2019-06-18T
[2019-06-18 19:06:58,705] {__init__.py:1353} INFO -
--------------------------------------------------------------------------------
[2019-06-18 19:06:58,705] {__init__.py:1354} INFO - Starting attempt 1 of 1
[2019-06-18 19:06:58,705] {__init__.py:1355} INFO -
--------------------------------------------------------------------------------
[2019-06-18 19:06:58,715] {__init__.py:1374} INFO - Executing <Task(BashOperator): notify> on 2019-06-18T19:06:28.102026+00:00
[2019-06-18 19:06:58,716] {base_task_runner.py:119} INFO - Running: ['airflow', 'run', 'download_rocket_launches', 'notify', '2019-06-1
[2019-06-18 19:06:59,871] {base_task_runner.py:101} INFO - Job 85: Subtask notify [2019-06-18 19:06:59,871] {__init__.py:51} INFO - Us
[2019-06-18 19:07:00,126] {base_task_runner.py:101} INFO - Job 85: Subtask notify [2019-06-18 19:07:00,126] {__init__.py:305} INFO - F
[2019-06-18 19:07:00,153] {base_task_runner.py:101} INFO - Job 85: Subtask notify [2019-06-18 19:07:00,152] {cli.py:517} INFO - Running
[2019-06-18 19:07:00,165] {bash_operator.py:81} INFO - Tmp dir root location:
 /tmp
[2019-06-18 19:07:00,165] {bash_operator.py:90} INFO - Exporting the following env vars:
AIRFLOW_CTX_DAG_ID=download_rocket_launches
AIRFLOW_CTX_TASK_ID=notify
AIRFLOW_CTX_EXECUTION_DATE=2019-06-18T19:06:28.102026+00:00
AIRFLOW_CTX_DAG_RUN_ID=manual__2019-06-18T19:06:28.102026+00:00
[2019-06-18 19:07:00,165] {bash_operator.py:104} INFO - Temporary script location: /tmp/airflowtmptdhnydwi/notify3obku1dp
[2019-06-18 19:07:00,165] {bash_operator.py:114} INFO - Running command: echo "There are now $(ls /tmp/images/ | wc -l) images."
[2019-06-18 19:07:00,173] {bash_operator.py:123} INFO - Output:
[2019-06-18 19:07:00,177] {bash_operator.py:127} INFO - There are now 5 images.
[2019-06-18 19:07:00,177] {bash_operator.py:131} INFO - Command exited with return code 0
[2019-06-18 19:07:03,692] {logging_mixin.py:95} INFO - [2019-06-18 19:07:03,692] {jobs.py:2562} INFO - Task exited with return code 0
```

# زمان‌بندی

| preset | meaning | cron |
|---|---|---|
| None | Don't schedule, use for exclusively "externally triggered" DAGs | |
| @once | Schedule once and only once | |
| @hourly | Run once an hour at the beginning of the hour | 0 * * * * |
| @daily | Run once a day at midnight | 0 0 * * * |
| @weekly | Run once a week at midnight on Sunday morning | 0 0 * * 0 |
| @monthly | Run once a month at midnight of the first day of the month | 0 0 1 * * |
| @yearly | Run once a year at midnight of January 1 | 0 0 1 1 * |

# زمان‌بندی – کران جابز

```
# ┌───────── minute (0 - 59)
# │ ┌─────── hour (0 - 23)
# │ │ ┌───── day of the month (1 - 31)
# │ │ │ ┌─── month (1 - 12)
# │ │ │ │ ┌─ day of the week (0 - 6) (Sunday to Saturday;
# │ │ │ │ │      7 is also Sunday on some systems)
# * * * * *
```

copy

```
0 2 * * * /bin/sh backup.sh
* * * * *  /scripts/script.sh
*/10 * * * * /scripts/monitor.sh
0 */4 * * * /scripts/script.sh
0 4,17 * * sun,mon /scripts/script.sh
* * * * *  sleep 30; /scripts/script.sh
```

# متغیرها

Airflow | DAGs | Data Profiling | Browse | Admin | Docs

## Variables

List (9) | Create | Add Filter | With selected | Search

| | Key | Val |
|---|---|---|
| | secret_password | ******** |
| | not_so_hidden | test value |
| | secret | ******** |
| | password | ******** |
| | passwd | ******** |
| | api_key | ******** |
| | apikey | ******** |
| | authorization | ******** |
| | access_token | ******** |

# XCom

On DAG: my_dag

Graph View | Tree View | Task Duration | Task Tries | Landing Times | Gantt | Details | Code | Refresh | Delete

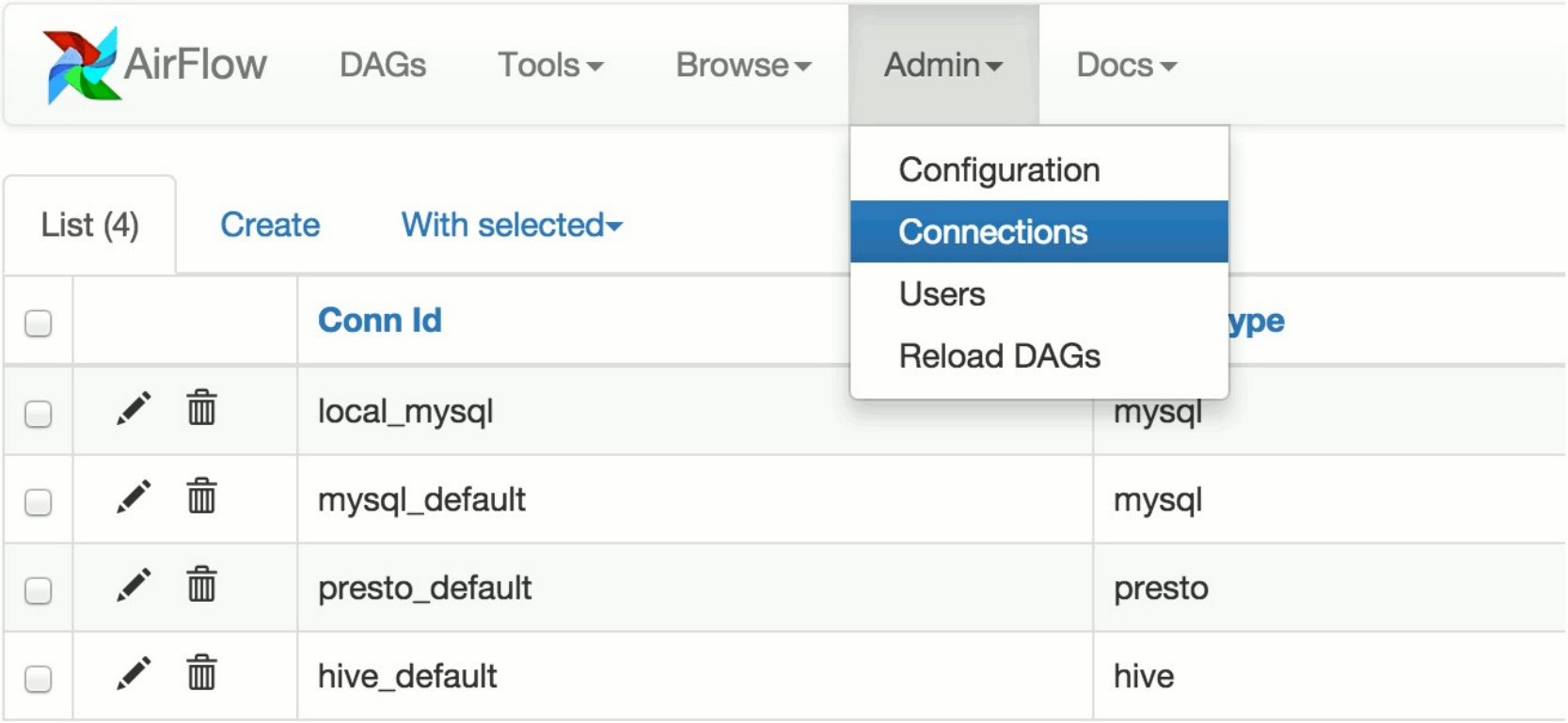
Task Instance: my_task 2019-06-07 00:00:00

Task Instance Details | Rendered Template | Log | XCom

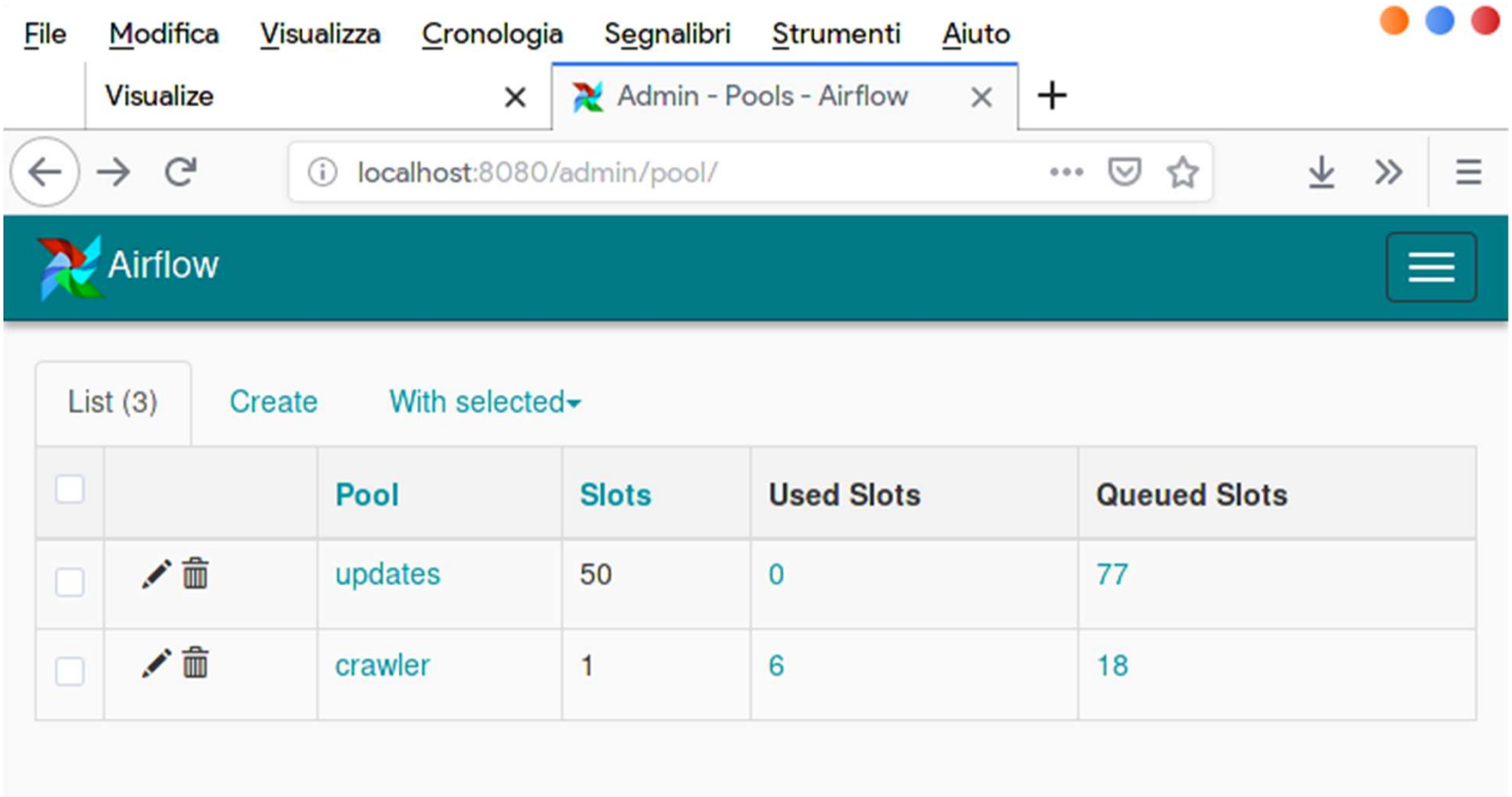
## XCom

| Key | Value |
|---|---|
| start | 2019-06-16T17:59:35.260928 |
| end | 2019-06-16T17:59:47.587656 |
| queries | ['foo', 'bar'] |
| counts | [1] |

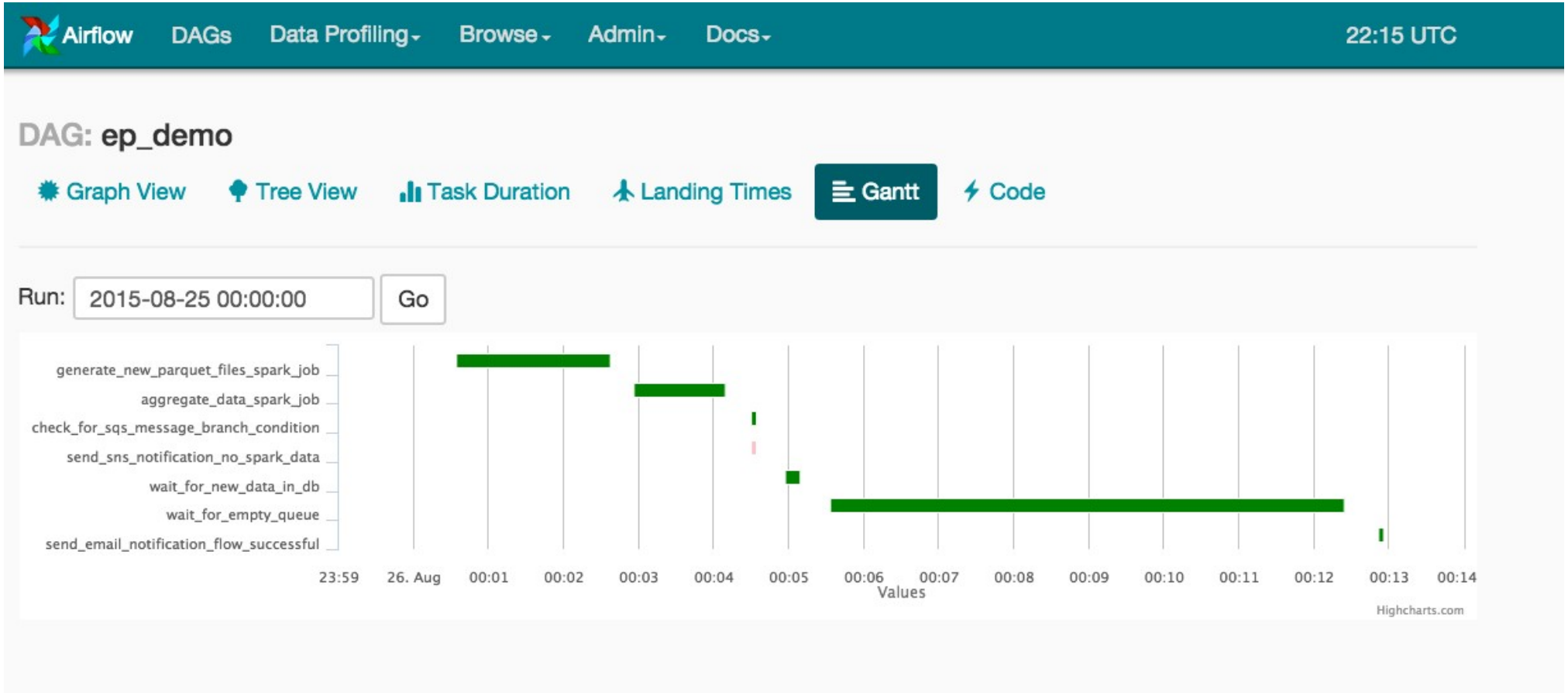
# Connections



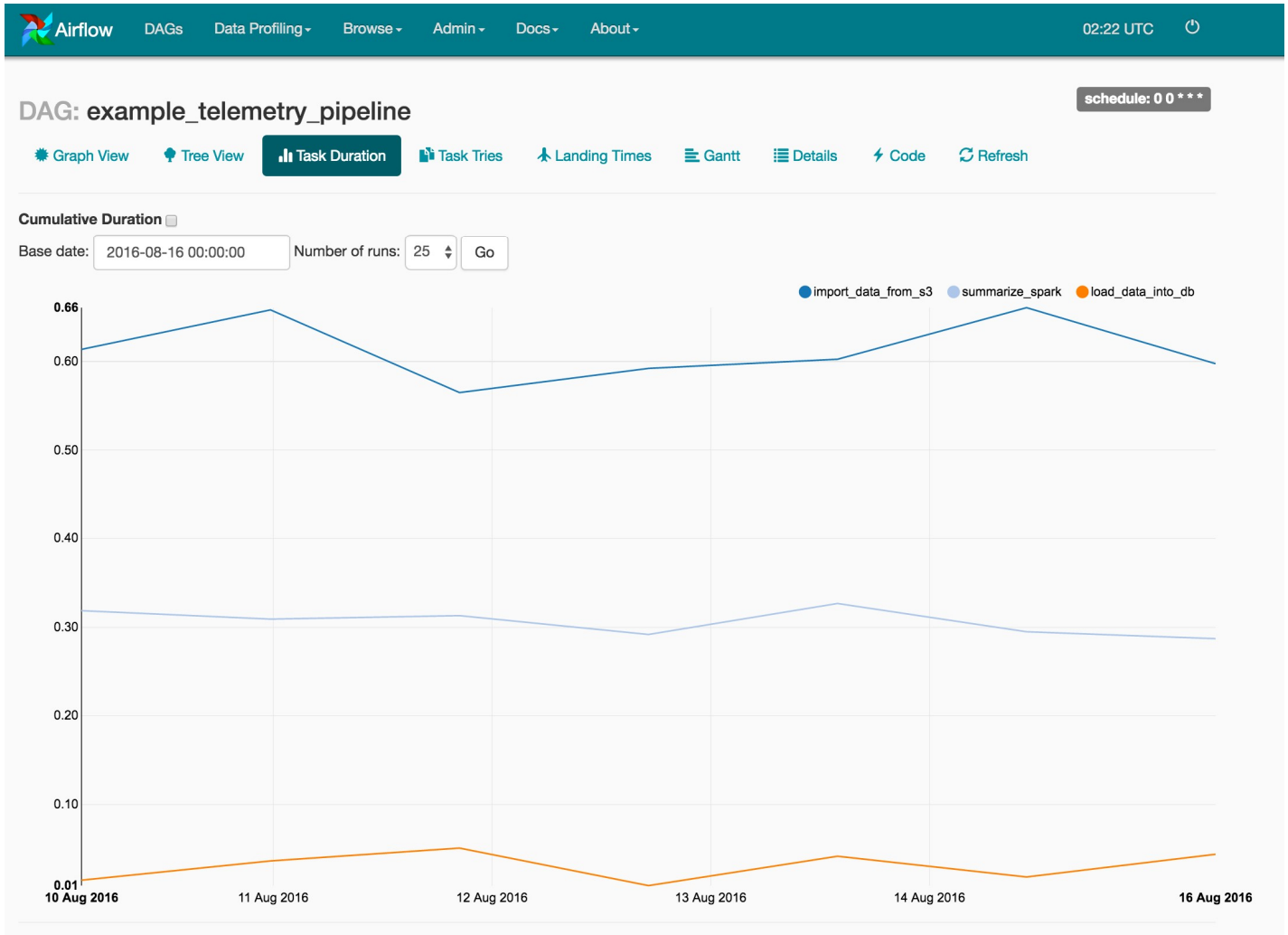
# Pools


File Modifica Visualizza Cronologia Segnalibri Strumenti Aiuto
Visualize
Admin - Pools - Airflow
localhost:8080/admin/pool/
Airflow
List (3)
Create
With selected
Pool
Slots
Used Slots
Queued Slots
updates
50
0
77
crawler
1
6
18

# Gantt Chart


Airflow
DAGs
Data Profiling
Browse
Admin
Docs
22:15 UTC
DAG: ep_demo
Graph View
Tree View
Task Duration
Landing Times
Gantt
Code
Run:
2015-08-25 00:00:00
Go
generate_new_parquet_files_spark_job
aggregate_data_spark_job
check_for_sqs_message_branch_condition
send_sns_notification_no_spark_data
wait_for_new_data_in_db
wait_for_empty_queue
send_email_notification_flow_successful
23:59
26. Aug
00:01
00:02
00:03
00:04
00:05
00:06
00:07
00:08
00:09
00:10
00:11
00:12
00:13
00:14
Values
Highcharts.com

# Task Duration Diagram



# مثال عملی

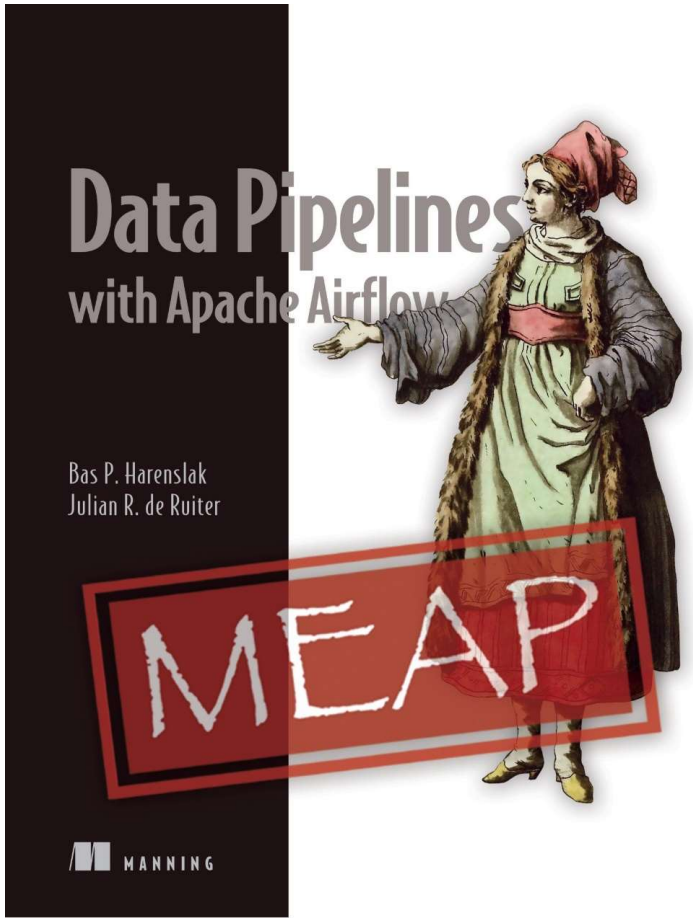
- جریان کار
  - دانلود فایل اکسل داد و ستد روزانه بورس ایران
  - تبدیل فایل به CSV
  - حذف فایل اکسل
  - تبدیل فایل CSV به پارکت
  - حذف فایل CSV

- کران جابز
- ایرفلو

# نصب ایرفلو

```
# airflow needs a home, ~/airflow is the default,
# but you can lay foundation somewhere else if you prefer
# (optional)
export AIRFLOW_HOME=~/airflow

# install from pypi using pip
pip install apache-airflow

# initialize the database
airflow initdb

# start the web server, default port is 8080
airflow webserver -p 8080

# start the scheduler
airflow scheduler

# visit localhost:8080 in the browser and enable the example dag in the home page
```

# منابع




- https://airflow.apache.org/docs/
- https://github.com/jghoman/awesome-apache-airflow


دانلود توئیت‌ها
تجمیع فایلها به صورت ساعتی
تبدیل فایلها به پارکت