# János Dani

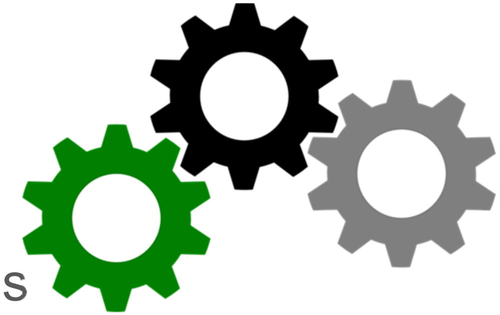
CEU / OSA helpdesk

| | |
|---|---|
| GitHub: | /danijanos |
| Instagram: | /dan1jan1 |

# Using Airflow as an Orchestrator for microservices

Archiving at OSA

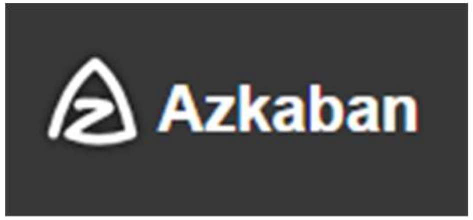
# What are microservices?

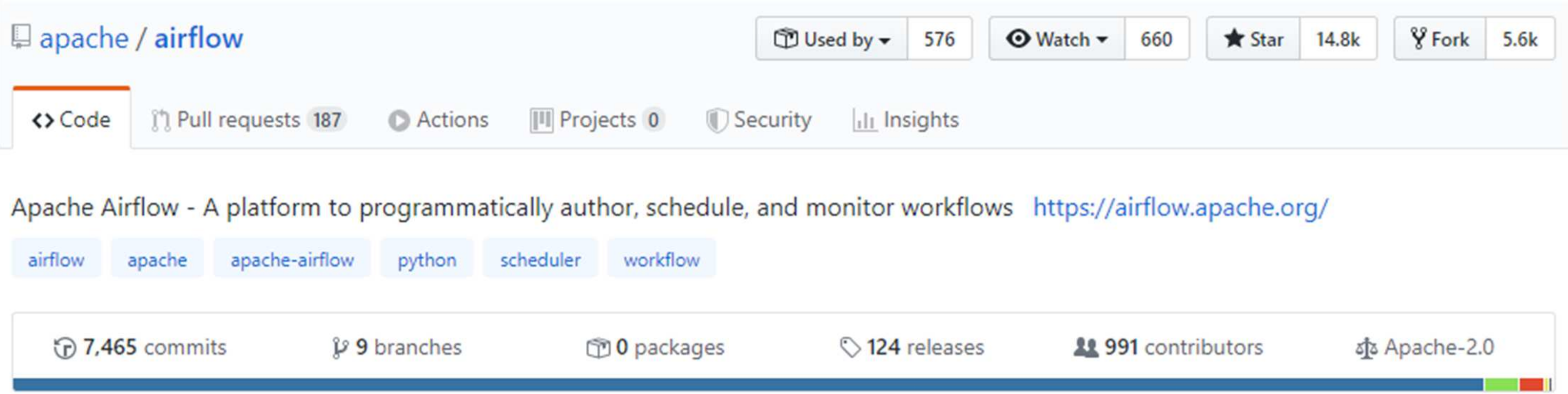


- Architectural style
- Structures an application as a collection of services
- Highly maintainable and testable
- Loosely coupled
- Independently deployable
- Communication through transactions
- Could be scaled separately and independently

# Workflow tools

- There is quite a lot...
- They are for various purposes
- Why is one better or favourable than the other?





apache / airflow

Apache Airflow - A platform to programmatically author, schedule, and monitor workflows https://airflow.apache.org/

airflow apache apache-airflow python scheduler workflow

7,465 commits 9 branches 0 packages 124 releases 991 contributors Apache-2.0

- ○ Email notification
- ○ Charts
- ○ Error checking
- Has a good documentation / community behind
- [Open source]

# Apache Airflow

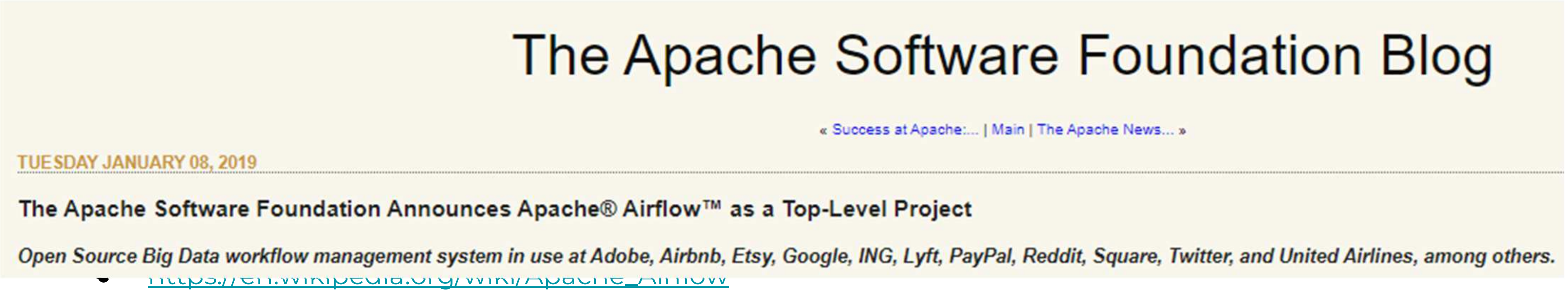
# About Airflow

Started at Airbnb in October 2014

Written in Python

Becoming an Apache Incubator project in March 2016

Top-Level Software Foundation project in January 2019.



The Apache Software Foundation Blog

« Success at Apache:... | Main | The Apache News... »

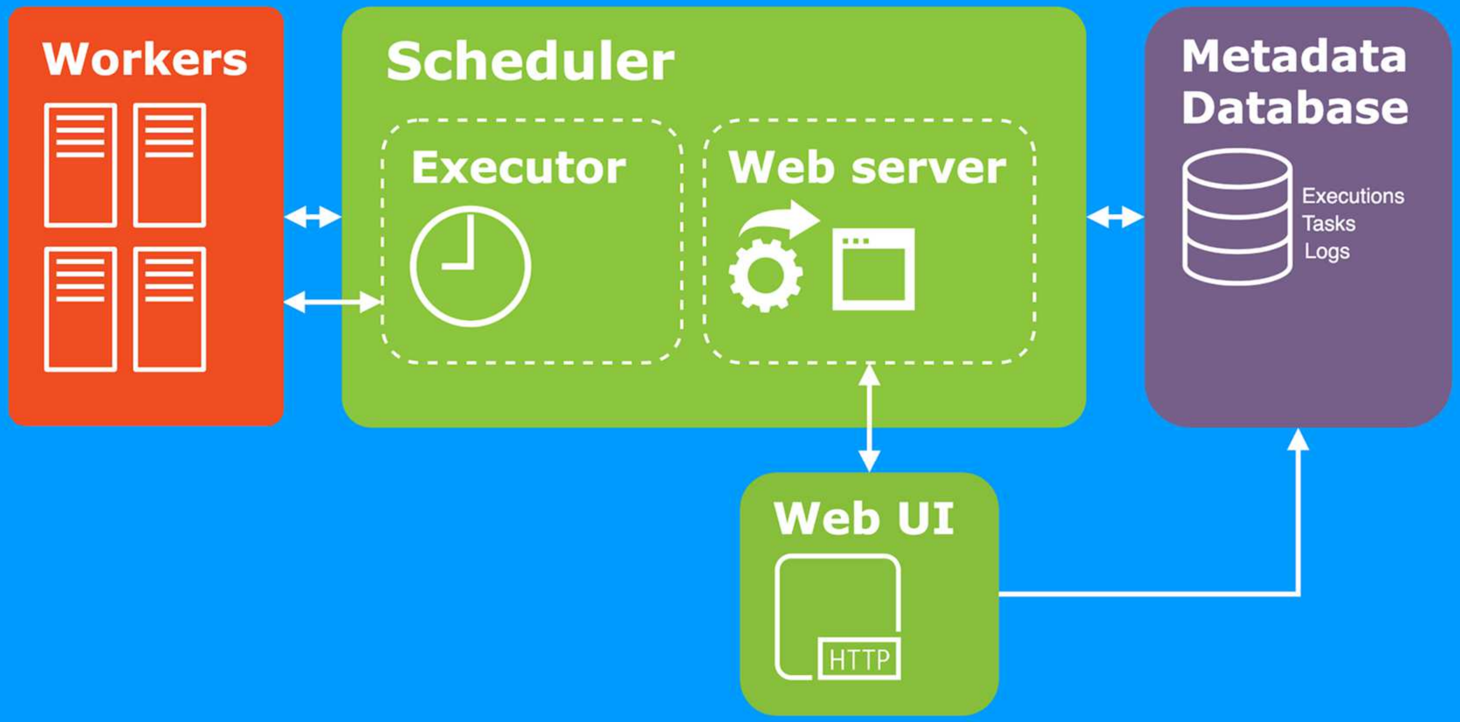
TUESDAY JANUARY 08, 2019

**The Apache Software Foundation Announces Apache® Airflow™ as a Top-Level Project**

*Open Source Big Data workflow management system in use at Adobe, Airbnb, Etsy, Google, ING, Lyft, PayPal, Reddit, Square, Twitter, and United Airlines, among others.*

- https://en.wikipedia.org/wiki/Apache_Airflow
- https://airflow.apache.org/

# Airflow's Architecture



Workers

Scheduler

Executor

Web server

Metadata Database

Executions
Tasks
Logs

Web UI

HTTP

# The scheduler

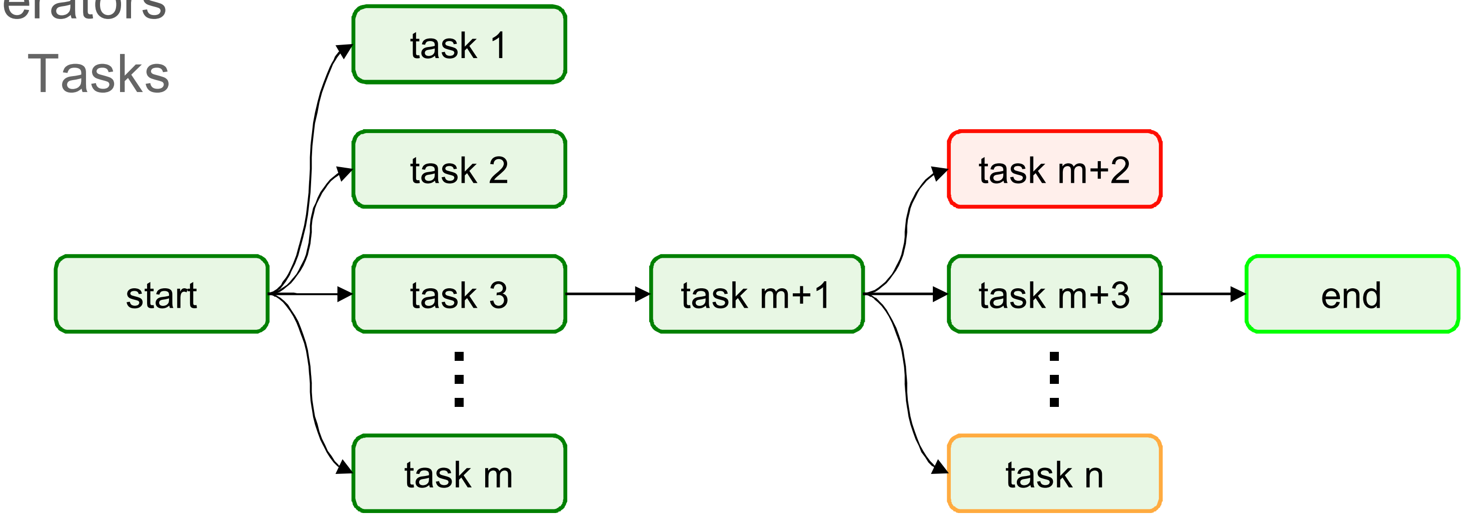
- Executes (triggers) tasks on an array of workers
- Monitors all tasks
- Executor types:
    - Sequential executor
    - Local
    - Celery (to scale tasks on different worker nodes)

source:

- https://airflow.apache.org/docs/stable/scheduler.html

# Basic conceptual building blocks

- Dags (Pipelines)
- Operators
  - Tasks



# Dag

- Holds a series of tasks connected with dependencies
- Made for avoiding cyclical dependencies between tasks
- Dags are identified by their IDs

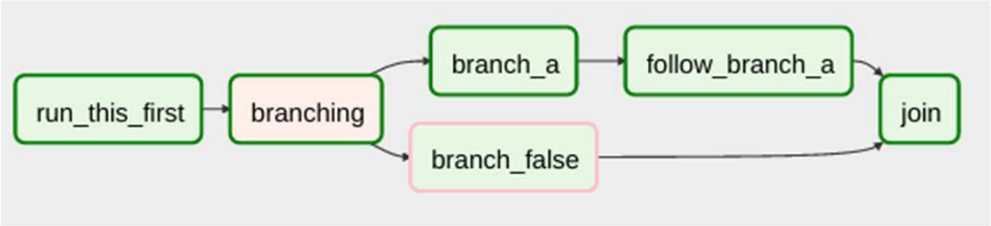
```
default_args = {
    'owner': 'airflow',
    'depends_on_past': False,
    'start_date': datetime(2018, 1, 1),
    'email': ['bonej@ceu.edu', 'danij@ceu.edu'],
    'email_on_failure': True,
    'email_on_retry': False,
    'retries': 1,
    'retry_delay': timedelta(minutes=5)
}
```

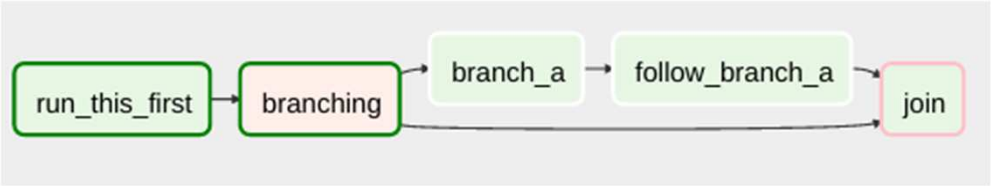
```
# DAGS
osa_av_workflow = DAG(
    dag_id='osa-av-workflow',
    description='Main DAG for the AV preservation workflow',
    default_args=default_args,
    schedule_interval=None,
    catchup=False)
```

# Operators

```
create_master_checksums = PythonOperator(
    task_id='create_master_checksums',
    python_callable=create_checksums,
    dag=osa_av_workflow,
    op_kwargs={
        'directory': 'Preservation',
        'file_extension': MASTER_FILE_EXTENSION
      })
```

- BashOperator - executes a bash command
- PythonOperator - calls a Python function
- EmailOperator - sends an email
- SimpleHttpOperator - sends an HTTP request
- BranchOperator
- …
- Write your our own operators!





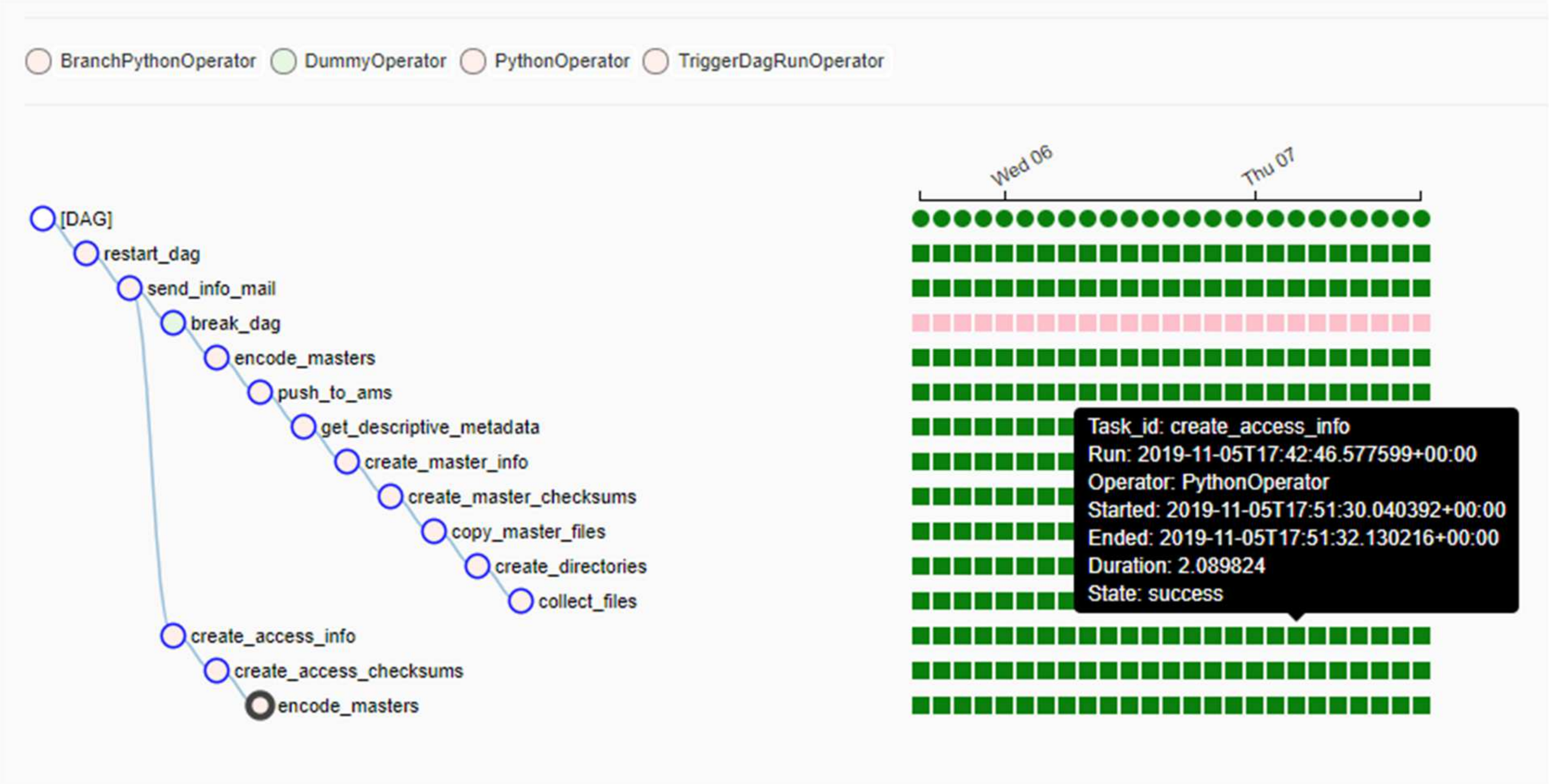
source: https://airflow.apache.org/concepts.html#operators

# Tasks

- Tasks are instantiated operators.
- Connected with dependencies
- Can be parallelized

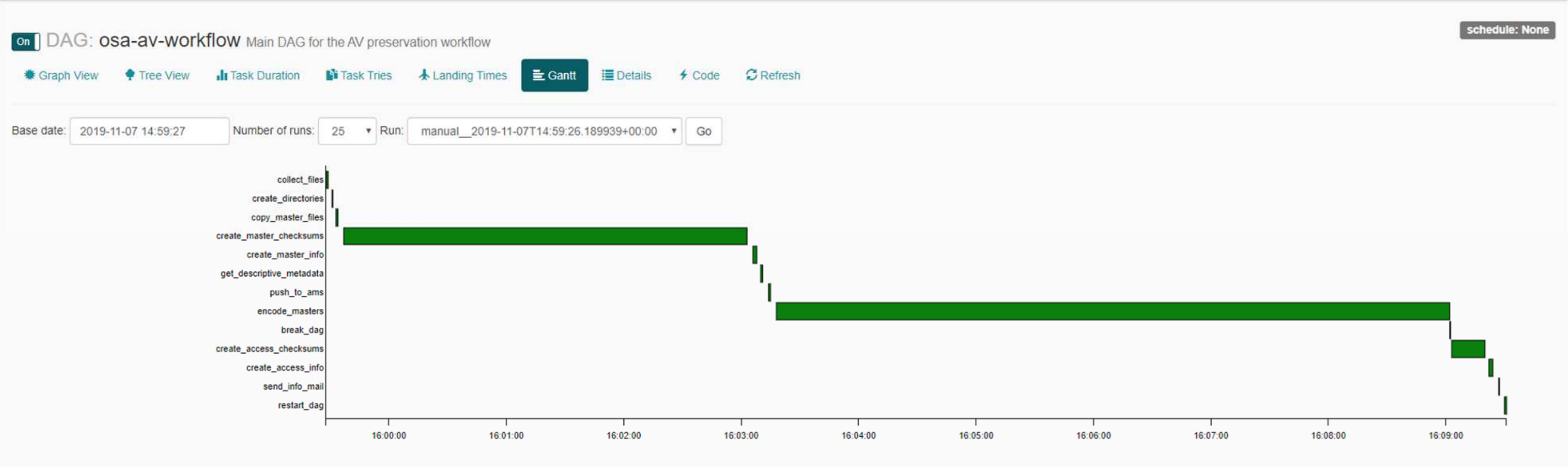
```
op1.dag = dag
op1.set_downstream(op2)
# is the same as:
dag >> op1 >> op2

op1 >> op2
# is the same as:
op1.set_downstream(op2)

op2 << op1
# is the same as:
op2.set_upstream(op1)
```

# Statuses



```
Task_id: create_access_info
Run: 2019-11-05T17:42:46.577599+00:00
Operator: PythonOperator
Started: 2019-11-05T17:51:30.040392+00:00
Ended: 2019-11-05T17:51:32.130216+00:00
Duration: 2.089824
State: success
```

# Airflow Web UI - Gantt chart

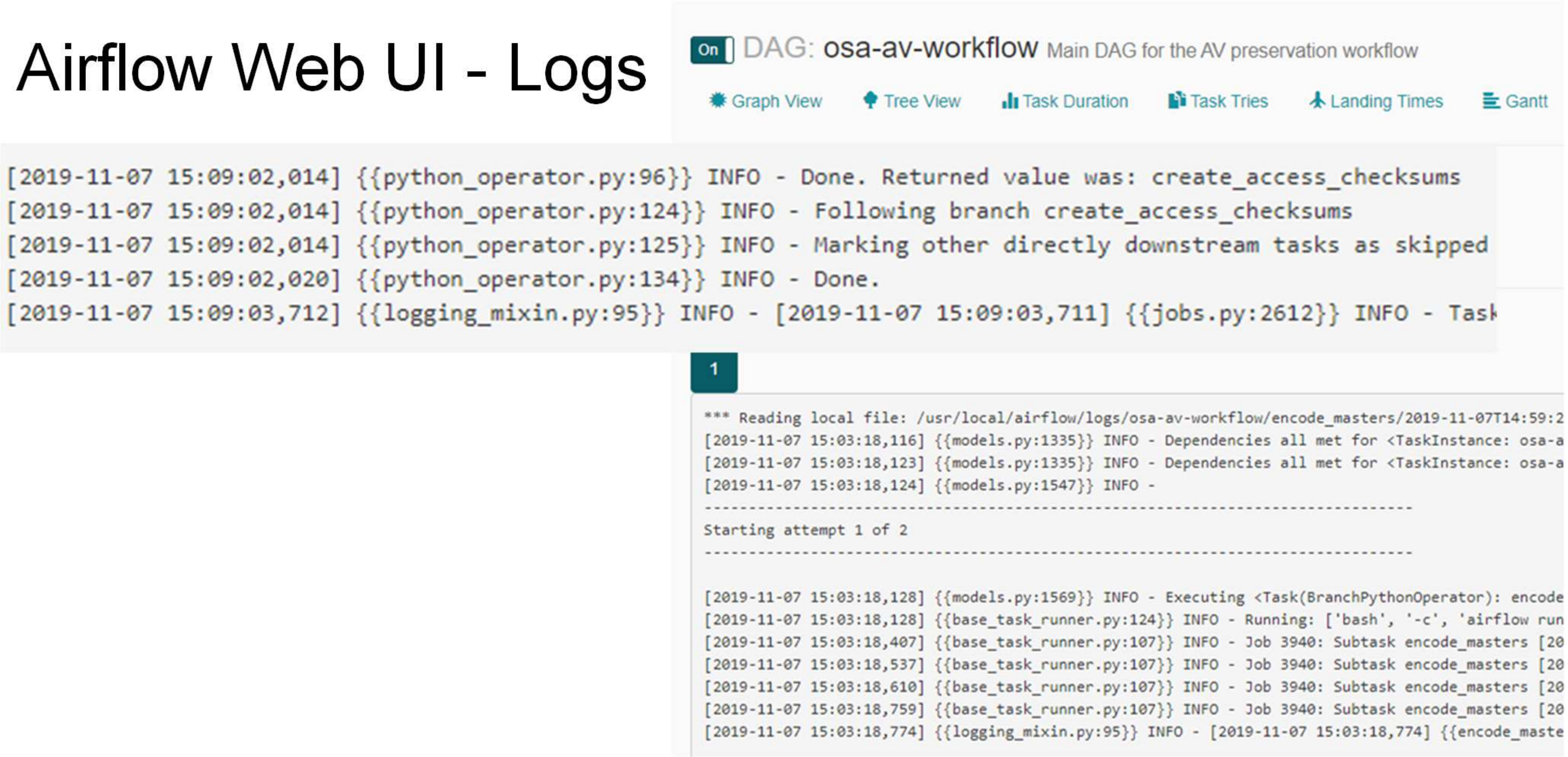


DAG: osa-av-workflow Main DAG for the AV preservation workflow

schedule: None

Base date: 2019-11-07 14:59:27 Number of runs: 25 Run: manual__2019-11-07T14:59:26.189939+00:00

# Airflow Web UI - Logs



```
[2019-11-07 15:09:02,014] {{python_operator.py:96}} INFO - Done. Returned value was: create_access_checksums
[2019-11-07 15:09:02,014] {{python_operator.py:124}} INFO - Following branch create_access_checksums
[2019-11-07 15:09:02,014] {{python_operator.py:125}} INFO - Marking other directly downstream tasks as skipped
[2019-11-07 15:09:02,020] {{python_operator.py:134}} INFO - Done.
[2019-11-07 15:09:03,712] {{logging_mixin.py:95}} INFO - [2019-11-07 15:09:03,711] {{jobs.py:2612}} INFO - Task
```

```
*** Reading local file: /usr/local/airflow/logs/osa-av-workflow/encode_masters/2019-11-07T14:59:2
[2019-11-07 15:03:18,116] {{models.py:1335}} INFO - Dependencies all met for <TaskInstance: osa-a
[2019-11-07 15:03:18,123] {{models.py:1335}} INFO - Dependencies all met for <TaskInstance: osa-a
[2019-11-07 15:03:18,124] {{models.py:1547}} INFO -
--------------------------------------------------------------------------------
Starting attempt 1 of 2
--------------------------------------------------------------------------------

[2019-11-07 15:03:18,128] {{models.py:1569}} INFO - Executing <Task(BranchPythonOperator): encode
[2019-11-07 15:03:18,128] {{base_task_runner.py:124}} INFO - Running: ['bash', '-c', 'airflow run
[2019-11-07 15:03:18,407] {{base_task_runner.py:107}} INFO - Job 3940: Subtask encode_masters [20
[2019-11-07 15:03:18,537] {{base_task_runner.py:107}} INFO - Job 3940: Subtask encode_masters [20
[2019-11-07 15:03:18,610] {{base_task_runner.py:107}} INFO - Job 3940: Subtask encode_masters [20
[2019-11-07 15:03:18,759] {{base_task_runner.py:107}} INFO - Job 3940: Subtask encode_masters [20
[2019-11-07 15:03:18,774] {{logging_mixin.py:95}} INFO - [2019-11-07 15:03:18,774] {{encode_maste
```

# Links

- https://airflow.apache.org/
- https://hub.docker.com/r/apache/airflow
- http://michal.karzynski.pl/blog/2017/03/19/developing-workflows-with-apache-airflow/