



Apache Airflow Workshop

Throw away your cron jobs

April 2021

Agenda

- About us
- Intro to Apache Airflow
- Build your own first Airflow pipeline
 - Note: please, ensure that you have Python 3.6, Docker and Docker Compose installed

Our team



Andrey Elistratov

Data Engineer
aelistratov@griddynamics.com



Sevak Avetisyan

Senior Data Engineer
savetisyan@griddynamics.com



Vladimir Baev

Delivery Manager
vbaev@griddynamics.com

Founded in **2006**

We are publicly traded company

[NASDAQ:GDYN](#)

Specializations

- ✓ You will have your **mentor** in your location
- ✓ We build personal **development plan** for every engineer
- ✓ **Big Community** where you can grow as engineer and as mentor for interns we train
- ✓ We are **leading** [community](#) in cutting edge technologies
- ✓ **GridU** - education center with over **100** hard and soft skills trainings

You join Grid Dynamics
Project is only part of the journey

50+ projects in various domains
We have flexible rotation process

Growth of every engineer
Is our main motto

Low level of bureaucracy and no time tracking
A lot of our managers grew from engineers

Projects

Our **domains** Retail, Tech, Media, Fintech
Communication directly with customers
Business trips and **relocation** to EU / US
Industry **best** practices and technologies
All projects in **clouds**

We do not support **legacy**

Grid Dynamics is known for architecting and delivering some of the largest digital transformation programs in the retail, brands, CPG, manufacturing, technology, and financial sectors. [Read more](#)

Fortune 500

Our customers are large and **famous**



Grid Big Data Community

180+

Big Data engineers in 10 offices
in 5 countries

40+

Ongoing Big Data projects

10+

Years of developing
in-house Big Data expertise

Technology Stack

Big Data projects

- Architect and build analytical data platforms
- Optimize performance, build access layers, and add data governance
- Migrate to native or hybrid cloud platforms without disruption
- Gather and process data with quality automation and AI/ML
- Complexity of some projects delivered include:
 - 5PB data in the platform
 - 300,000 events/sec data streams

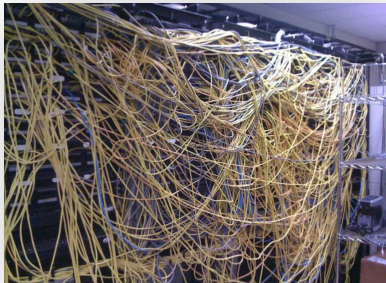
Technology Stack

- **Languages:** Scala, Python, Java
- **Platforms:** Spark, Kafka, Beam, Flink, NiFi, BigQuery
- **Clouds:** Google Cloud Platform, Amazon AWS, Microsoft Azure
- **NoSQL:** Cassandra, Redis, HBase
- Airflow, Talend, Snowflake

Introduction to Apache Airflow

Orchestration tools

- Way of connecting application's components to provide appropriate scheduling and interaction
- Examples
 - Sequential processing of image by multiple separate logical units (resize, apply filters, export)
 - Sequence of ETL jobs, connected by inputs/outputs



Orchestration tools: major players

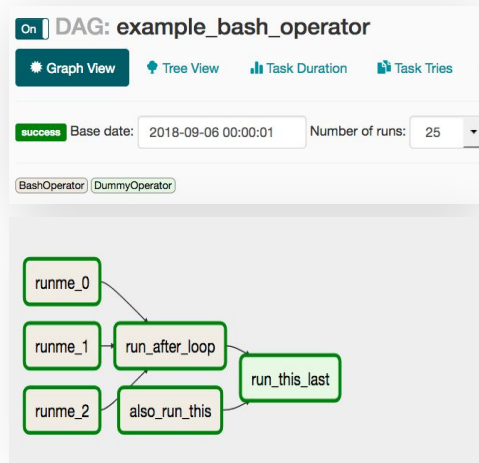


Apache Airflow: Overview

“Platform to programmatically author, schedule, and monitor workflows”



- Vocabulary
 - DAGs (Directed acyclic graphs) = pipelines
 - Tasks, Task instances
 - Operators, Sensors
 - DAG run
 - master, worker nodes
- Features
 - Batch-oriented
 - Define pipelines (DAGs) as Python code
 - Dynamic DAGs support
 - Extensible
 - Parameterizing with Jinja templates
 - Scalable
 - Rich UI



Apache Airflow: Overview

DAGs

Search:

	i	DAG	Schedule	Owner	Recent Tasks i	Last Run i	DAG Runs i	Links
	<input checked="" type="checkbox"/>	example_bash_operator	0 0 ***	airflow	6	2018-09-06 00:00 i	5	
	<input checked="" type="checkbox"/>	example_branch_dop_operator_v3	*/* ****	airflow	3 1 1 5	2018-09-05 00:56 i	54 3	
	<input checked="" type="checkbox"/>	example_branch_operator	@daily	airflow	5	2018-09-06 00:00 i	2	
	<input checked="" type="checkbox"/>	example_xcom	@once	airflow	3	2018-09-05 00:00 i	1	
	<input checked="" type="checkbox"/>	latest_only	4:00:00	Airflow	2	2018-09-07 16:00 i	35	

on DAG: example_branch_dop_operator_v3

Graph View ☒ Tree View Task Duration Task Tries Landing Times Gantt Details Code

Base date: 2018-09-05 01:04:00 Number of runs: 25 Go

BranchPythonOperator DummyOperator






run_after_loop on 2018-09-08T00:00:00+00:00

Close

Simple DAG

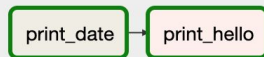
```
1 dag = DAG(
2     'simple_dag_example',
3     default_args=default_args,
4     description='A simple tutorial DAG',
5     schedule_interval='*/5 * * * *',
6 )
7
8 t1 = BashOperator(
9     task_id='print_date',
10    bash_command='date',
11    dag=dag
12 )
13
14 def print_hello():
15     print('Hello!')
16
17 t2 = PythonOperator(
18     task_id='print_hello',
19     python_callable=print_hello,
20     dag=dag
21 )
22
23 t1 >> t2
```

☐ Off DAG: simple_dag_example

 Graph View  Tree View  Task Duration

success Base date: 2019-12-01 00:05:01 Number of tasks: 2

BashOperator **PythonOperator**



On

DAG: simple_dag_example

A simple tutorial DAG

schedule: */5 *

Graph View

Tree View

Task Duration

Task Tries

Landing Times

Gantt

Details

Code

Trigger DAG

Refresh

Delete

Base date:

2019-12-01 00:05:00

Number of runs:

25

Go

☐ BashOperator
 ☐ PythonOperator

success

running

failed

skipped

up_for_reschedule

up_for_retry

queued

no_status

[DAG]

print_hello

print_date

print_date

print_hello

print_date

print_hello

print_date

print_hello

Log by attempts

1

Toggle wrap

Jump to end

```

[2019-12-01 00:05:10,229] {{taskinstance.py:620}} INFO - Dependencies all met for <TaskInstance: simple_dag_example.print_hello 2019-12-01T00:00:00+00:00 [q
[2019-12-01 00:05:10,259] {{taskinstance.py:620}} INFO - Dependencies all met for <TaskInstance: simple_dag_example.print_hello 2019-12-01T00:00:00+00:00 [q
[2019-12-01 00:05:10,260] {{taskinstance.py:838}} INFO -

[2019-12-01 00:05:10,260] {{taskinstance.py:839}} INFO - Starting attempt 1 of 2
[2019-12-01 00:05:10,260] {{taskinstance.py:840}} INFO -

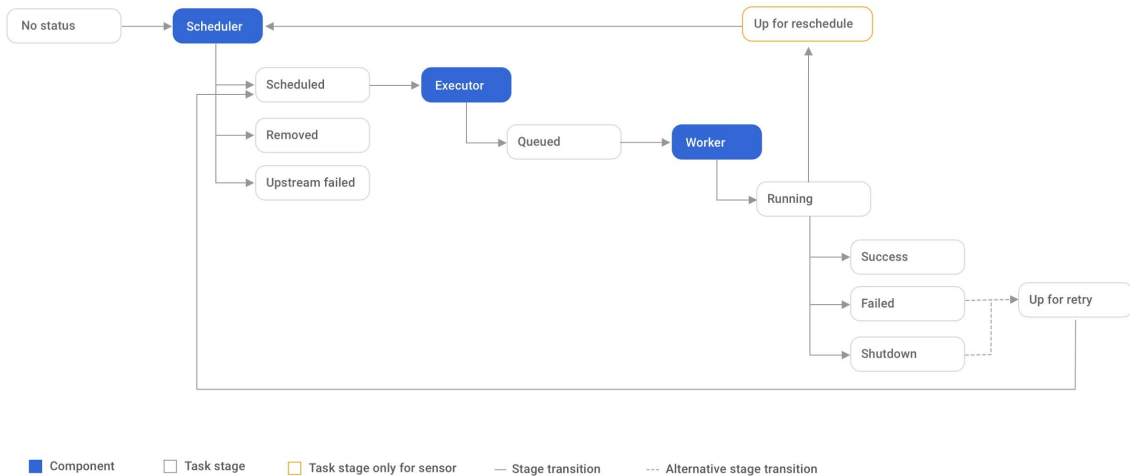
[2019-12-01 00:05:10,281] {{taskinstance.py:859}} INFO - Executing <Task(PythonOperator): print_hello> on 2019-12-01T00:00:00+00:00
[2019-12-01 00:05:10,281] {{base_task_runner.py:133}} INFO - Running: ['airflow', 'run', 'simple_dag_example', 'print_hello', '2019-12-01T00:00:00+00:00', '.
[2019-12-01 00:05:11,840] {{base_task_runner.py:115}} INFO - Job 26: Subtask print_hello [2019-12-01 00:05:11,839] {{settings.py:213}} INFO - settings.confi
[2019-12-01 00:05:11,911] {{base_task_runner.py:115}} INFO - Job 26: Subtask print_hello /usr/local/lib/python3.7/site-packages/psycopg2/_init_
[2019-12-01 00:05:11,911] {{base_task_runner.py:115}} INFO - Job 26: Subtask print_hello
[2019-12-01 00:05:12,570] {{base_task_runner.py:115}} INFO - Job 26: Subtask print_hello [2019-12-01 00:05:12,569] {{_init_.py:51}} INFO - Using executor
[2019-12-01 00:05:14,202] {{base_task_runner.py:115}} INFO - Job 26: Subtask print_hello [2019-12-01 00:05:14,202] {{dagbag.py:90}} INFO - Filling up the Dag
[2019-12-01 00:05:14,276] {{base_task_runner.py:115}} INFO - Job 26: Subtask print_hello [2019-12-01 00:05:14,275] {{cli.py:516}} INFO - Running <TaskInstan
[2019-12-01 00:05:14,312] {{python_operator.py:105}} INFO - Exporting the following env vars:
AIRFLOW_CTX_DAG_ID=simple_dag_example
AIRFLOW_CTX_TASK_ID=print_hello
AIRFLOW_CTX_EXECUTION_DATE=2019-12-01T00:00:00+00:00
AIRFLOW_CTX_DAG_RUN_ID=scheduled__2019-12-01T00:00:00+00:00
[2019-12-01 00:05:14,313] {{logging_mixin.py:95}} INFO - Hello!
[2019-12-01 00:05:14,313] {{python_operator.py:114}} INFO - Done. Returned value was: None
[2019-12-01 00:05:15,197] {{logging_mixin.py:95}} INFO - [ [34m2019-12-01 00:05:15,196 [0m] [ [34mlocal_task.job.py: [0m105]] INFO [0m - Task exited with ret

```

Apache Airflow: Concepts

- Operators and Sensors
 - BashOperator
 - PythonOperator
 - PostgresOperator
 - SparkSubmitOperator
 - BaseBranchOperator, TriggerDagRunOperator, SubDagOperator
 - S3PrefixSensor
- Pools, Queues
- Hooks (HDFSHook, SlackHook), Connections, Variables, XComs, etc

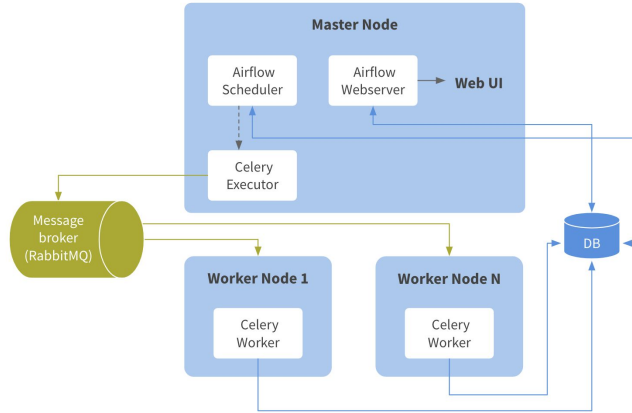
Infrastructure Task Lifecycle



Infrastructure

Airflow setup (Celery Executors)

- Master node has:
 - Airflow scheduler
 - Webserver
- Several Airflow workers (CeleryExecutors)
- Python virtualenvs
 - Airflow runs scheduler and executes tasks under virtualenvs
 - scheduler env
 - workers env
 - PythonVirtualenvOperator
- Migrations
 - Airflow versions
 - Python versions



Infrastructure

Airflow Integration

- Executors
 - SequentialExecutor
 - LocalExecutor
 - CeleryExecutor
 - Kubernetes Executor
- Cloud integrations (Operators, Hooks)
 - Azure
 - AWS
 - GCP
- Command Line Interface
- REST API (experimental)
- Airflow DataBase

Practical part

Let's launch a rocket 🚀

- Our goals for today
 - Learn something cool about Airflow
 - Build your first DAG
 - Have fun

Let's launch a rocket 🚀

- Our goals for today
 - Learn something cool about Airflow
 - Build your first DAG
 - Have fun
- Implementation:

Wait for
approval

Let's launch a rocket 🚀

- Our goals for today
 - Learn something cool about Airflow
 - Build your first DAG
 - Have fun
- Implementation:

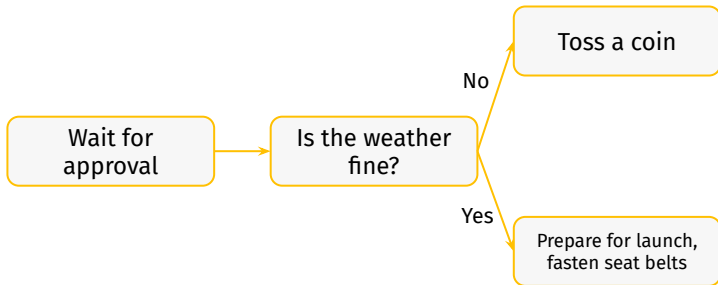
Wait for
approval



Is the weather
fine?

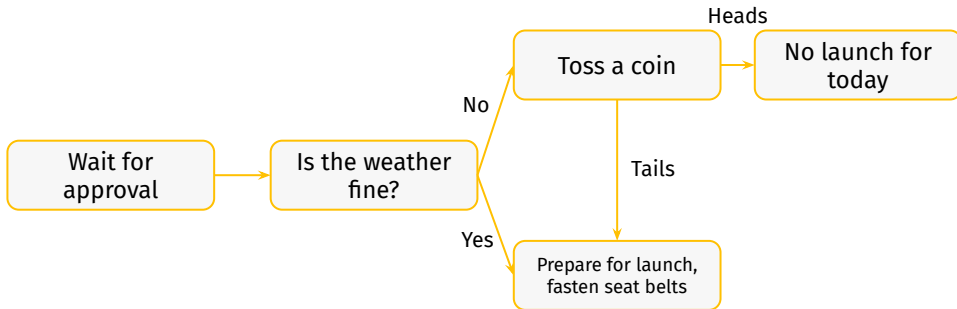
Let's launch a rocket 🚀

- Our goals for today
 - Learn something cool about Airflow
 - Build your first DAG
 - Have fun
- Implementation:



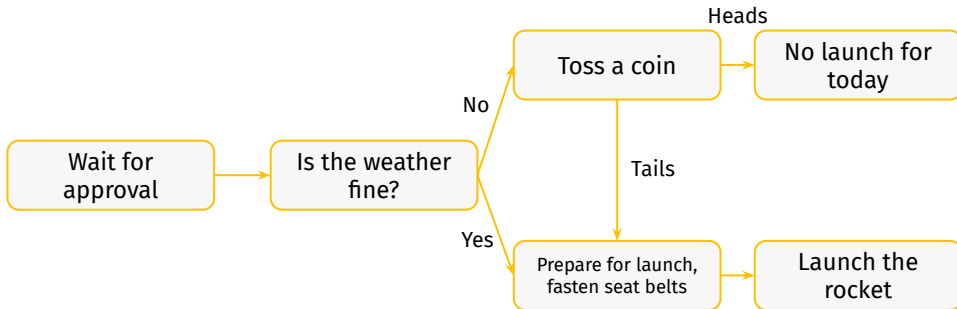
Let's launch a rocket 🚀

- Our goals for today
 - Learn something cool about Airflow
 - Build your first DAG
 - Have fun
- Implementation:



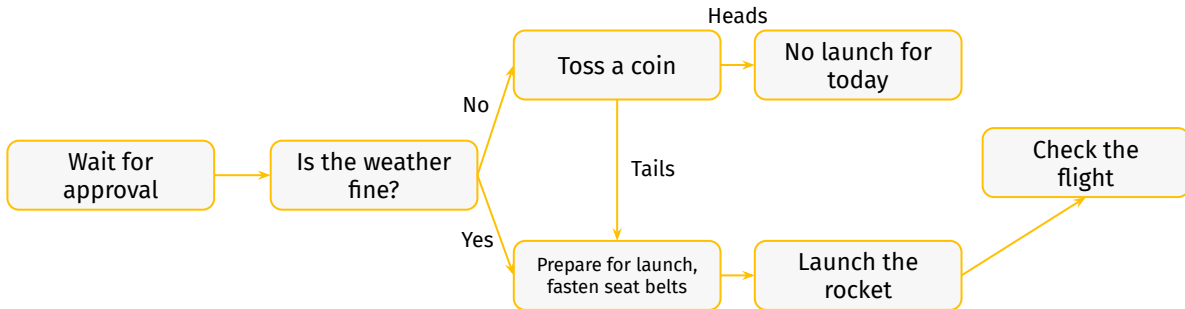
Let's launch a rocket 🚀

- Our goals for today
 - Learn something cool about Airflow
 - Build your first DAG
 - Have fun
- Implementation:



Let's launch a rocket 🚀

- Our goals for today
 - Learn something cool about Airflow
 - Build your first DAG
 - Have fun
- Implementation:



Environment setup

- Prerequisites
 - Python 3.6
 - Docker
 - Docker Compose
- Setup Airflow
 - Download workshop [repo](#)
 - Execute steps from README.md
 - Check that example DAG works

It's coding time!

Resources

- https://github.com/vladimirbaev/grid_dynamics_apache_airflow_workshop_2021
- <https://airflow.apache.org/>
- <https://github.com/apache/airflow>
- <https://stackoverflow.com/questions/tagged/airflow/>
- Common Pitfalls:
<https://cwiki.apache.org/confluence/pages/viewpage.action?pageId=62694614>
- Official Slack workspace: <https://apache-airflow.slack.com/>
- Russian Telegram community (1200 members): <https://t.me/ruairflow>



Thank
you!

Grid Dynamics International, Inc.

500 Executive Parkway,
Suite 520 / San Ramon, CA
+1 650-523-5000
info@griddynamics.com
www.griddynamics.com