Scalable ETL Workflow using Airflow and Kubernetes

Fitra Aditya
Lead Backend Engineer at HappyFresh
fitra.aditya@happyfresh.com

November 7th 2020

About Me

Fitra Aditya

- Lead backend engineer at HappyFresh
- WebRTC enthusiast

https://www.linkedin.com/in/fitraaditya/

fitraditya@gmail.com

Agenda

1. Workflow Overview
2. Manage Workflow using Airflow
3. Airflow Architecture
4. Running Airflow on Kubernetes
5. Summary

OICNDI

Workflow Overview

Workflow

The sequence of industrial, administrative, or other processes through which a piece of work passes from initiation to completion.

A workflow consists of an orchestrated and repeatable pattern of activity, enabled by the systematic organization of resources into processes that transform materials, provide services, or process information.

Simple Case

Home construction workflow

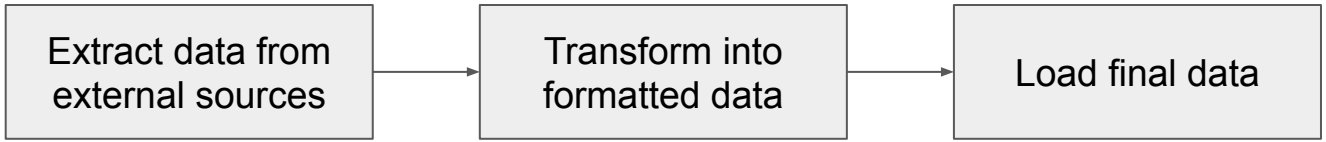
1. Land clearing
2. Foundation
3. Framing
4. Roof and Siding
5. Plumbing and Electrical
6. Painting
7. Finishing

Workflow in Software Engineering

- ETL process
- ML pipelines
- Automated testing orchestrations
- Etc

Sample Use Case

ETL (Extract, Transform, Load) Process



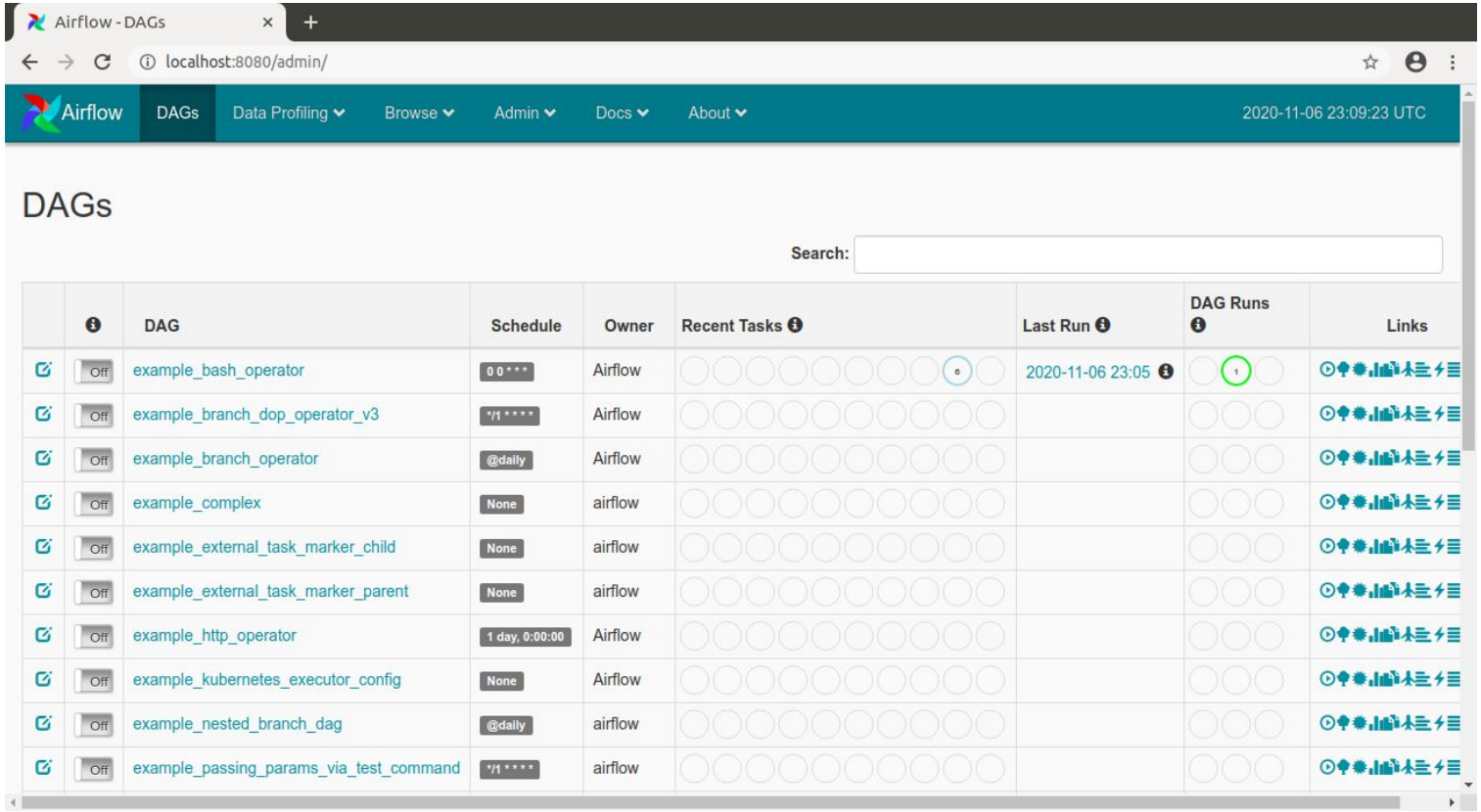
Simple Implementation

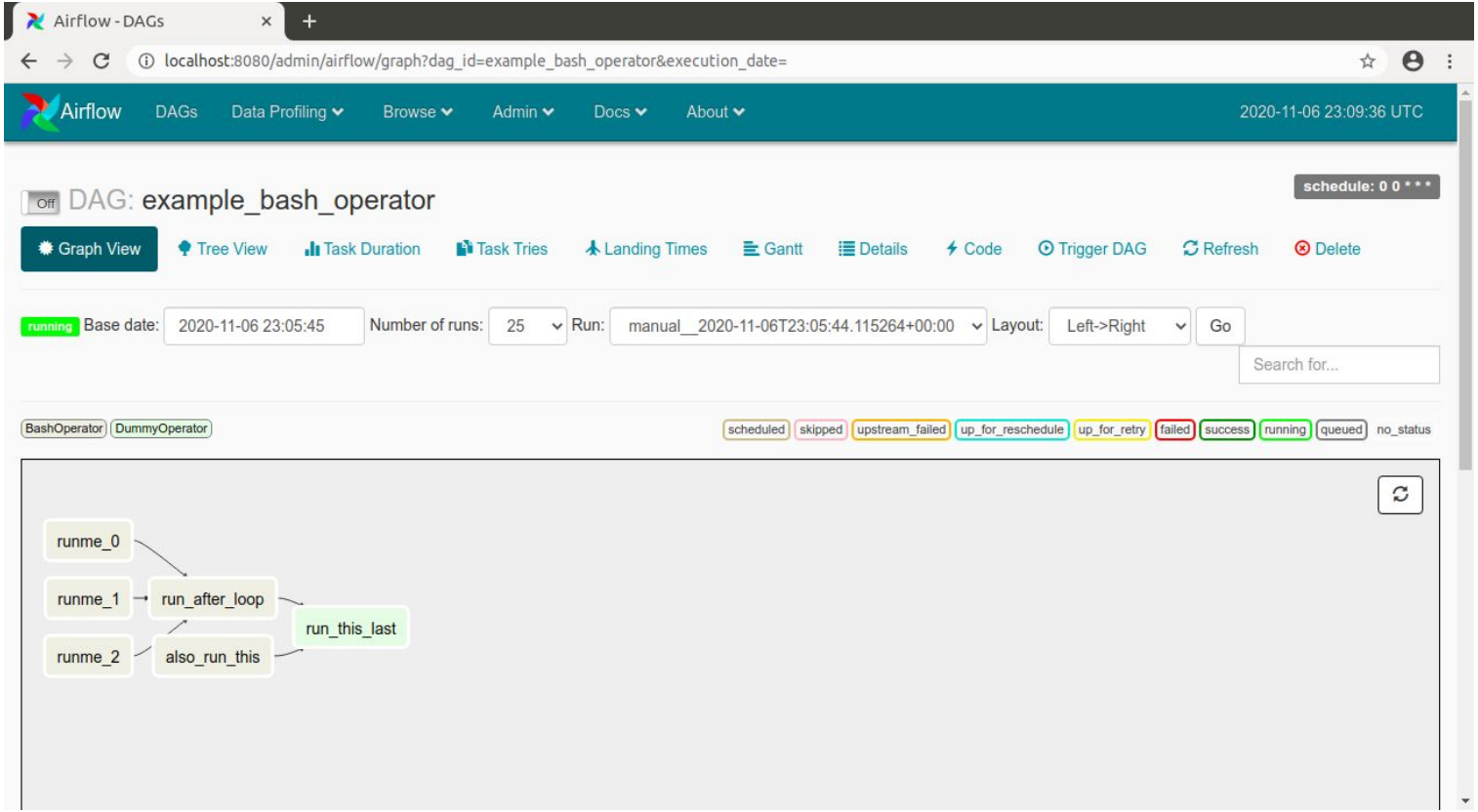
OICNDI

Manage Workflow using Airflow

Airflow

- Airflow is a platform to programmatically author, schedule and monitor workflows.
- Written in python
- Provides monitoring tools like web interface and alert


Airflow - DAGs
localhost:8080/admin/
Airflow
DAGs
Data Profiling
Browse
Admin
Docs
About
2020-11-06 23:09:23 UTC
DAGs
Search:
DAG
Schedule
Owner
Recent Tasks
Last Run
DAG Runs
Links
Off
example_bash_operator
0 0 * * *
Airflow
2020-11-06 23:05
example_branch_dop_operator_v3
*/1 * * * *
Airflow
example_branch_operator
@daily
Airflow
example_complex
None
airflow
example_external_task_marker_child
None
airflow
example_external_task_marker_parent
None
airflow
example_http_operator
1 day, 0:00:00
Airflow
example_kubernetes_executor_config
None
Airflow
example_nested_branch_dag
@daily
airflow
example_passing_params_via_test_command
*/1 * * * *
airflow


Airflow - DAGs
localhost:8080/admin/airflow/graph?dag_id=example_bash_operator&execution_date=
Airflow
DAGs
Data Profiling
Browse
Admin
Docs
About
2020-11-06 23:09:36 UTC
DAG: example_bash_operator
schedule: 0 0 * * *
Graph View
Tree View
Task Duration
Task Tries
Landing Times
Gantt
Details
Code
Trigger DAG
Refresh
Delete
running
Base date:
2020-11-06 23:05:45
Number of runs:
25
Run:
manual__2020-11-06T23:05:44.115264+00:00
Layout:
Left->Right
Go
Search for...
BashOperator
DummyOperator
scheduled
skipped
upstream_failed
up_for_reschedule
up_for_retry
failed
success
running
queued
no_status
runme_0
runme_1
runme_2
run_after_loop
also_run_this
run_this_last

Task Definition

```
from airflow.operators.bash_operator import BashOperator

t1 = BashOperator(
    task_id='print_user',
    bash_command='whoami',
    dag=dag)

t2 = BashOperator(
    task_id='print_date',
    bash_command='date',
    retries=3,
    dag=dag)

t1 >> t2
```

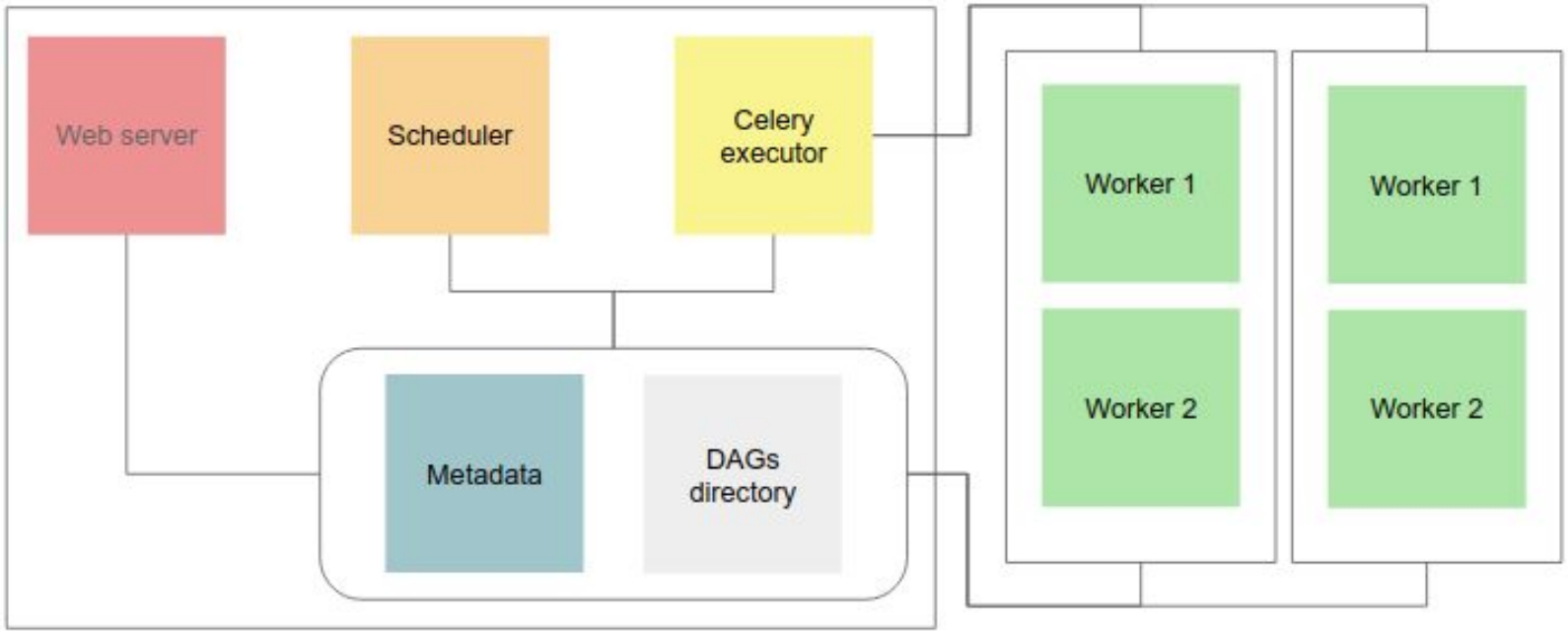
OICNDI

Airflow Architecture

Airflow Executor

- LocalExecutor
- CeleryExecutor
- KubernetesExecutor

Airflow - Multiple Instances



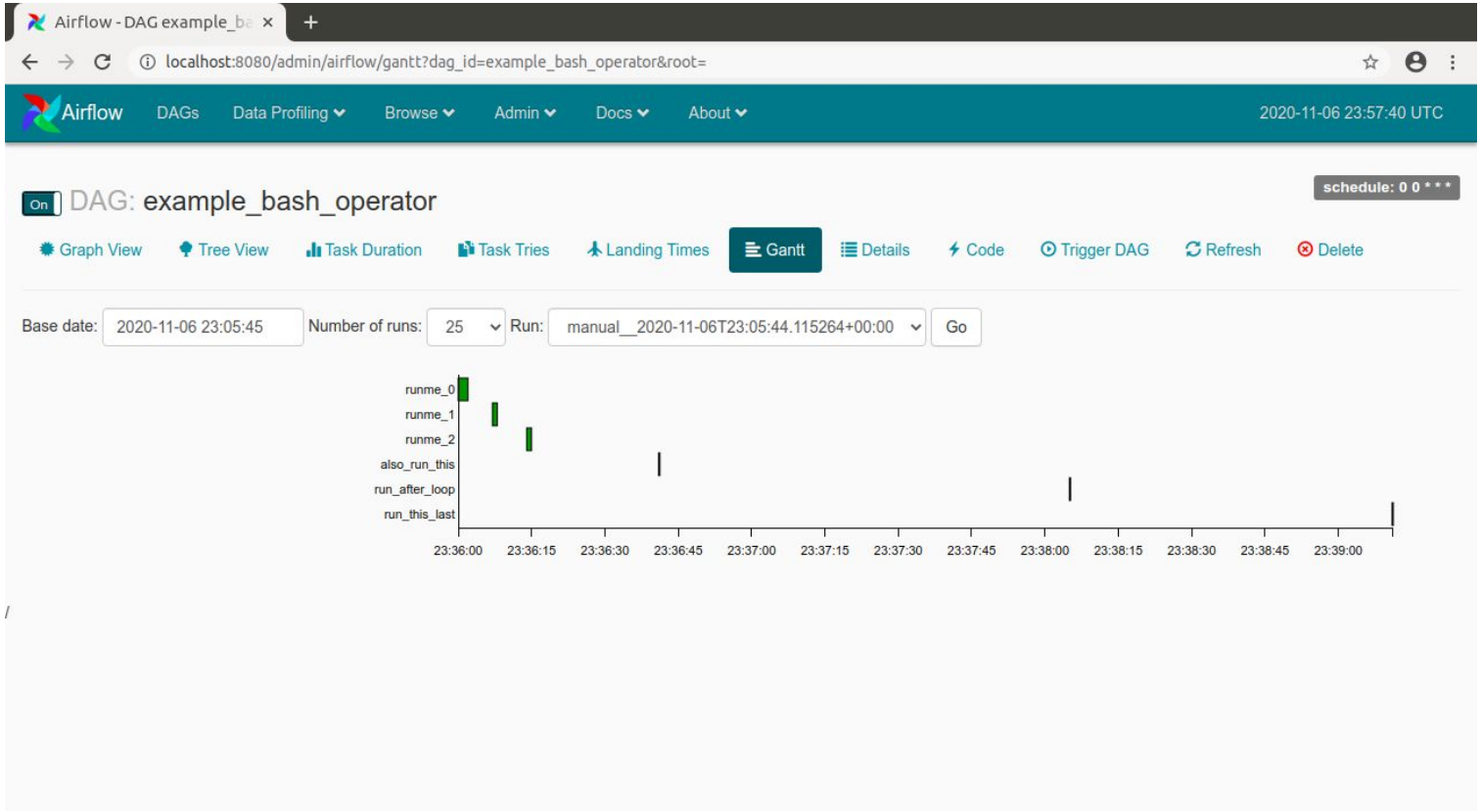
Airflow - Multiple Instances

Pros

- Easy setup

Cons

- Lack of flexibility


Airflow - DAG example_ba
localhost:8080/admin/airflow/gantt?dag_id=example_bash_operator&root=
Airflow
DAGs
Data Profiling
Browse
Admin
Docs
About
2020-11-06 23:57:40 UTC
On
DAG: example_bash_operator
schedule: 0 0 * * *
Graph View
Tree View
Task Duration
Task Tries
Landing Times
Gantt
Details
Code
Trigger DAG
Refresh
Delete
Base date: 2020-11-06 23:05:45
Number of runs: 25
Run: manual__2020-11-06T23:05:44.115264+00:00
Go
runme_0
runme_1
runme_2
also_run_this
run_after_loop
run_this_last
23:36:00
23:36:15
23:36:30
23:36:45
23:37:00
23:37:15
23:37:30
23:37:45
23:38:00
23:38:15
23:38:30
23:38:45
23:39:00

OICNDI

Running Airflow on Kubernetes

Airflow Kubernetes Executor

- High level of elasticity
- Task-level pod configuration
- Fault tolerance

Task Definition

```
start_task = PythonOperator(
    task_id="start_task", python_callable=print_stuff, dag=dag
)
one_task = PythonOperator(
  task_id="one_task", python_callable=print_stuff, dag=dag,
  executor_config={"KubernetesExecutor": {"image": "airflow:latest"}}
)
two_task = PythonOperator(
  task_id="two_task", python_callable=use_airflow_binary, dag=dag,
  executor_config={"KubernetesExecutor": {"image": "airflow:latest"}}
)
```

Airflow Kubernetes Executor

```
[airflow@localhost kubernetes]$ kubectl get pods
NAME                                  READY   STATUS    RESTARTS   AGE
airflow-7bdfd78b9c-4n5w5              2/2     Running   0          18h
busybox-c8f8564d4-5pps7               1/1     Running   0          18h
postgres-airflow-79886555bd-422c4     1/1     Running   0          22h
```

Airflow Kubernetes Executor

```
[airflow@localhost ~]$ kubectl get --watch pods
NAME                                                READY   STATUS              RESTARTS   AGE
airflow-7bdfd78b9c-4n5w5                            2/2     Running             10         24h
busybox-c8f8564d4-5pps7                             1/1     Running             0          24h
postgres-airflow-79886555bd-422c4                   1/1     Running             0          28h
examplekubernetesexecutorstarttask-421b2c62c0d648feb41f95e4ed7fed5e   0/1     Pending             0          0s
examplekubernetesexecutorstarttask-421b2c62c0d648feb41f95e4ed7fed5e   0/1     Pending             0          1s
examplekubernetesexecutorstarttask-421b2c62c0d648feb41f95e4ed7fed5e   0/1     ContainerCreating   0          1s
examplekubernetesexecutorstarttask-421b2c62c0d648feb41f95e4ed7fed5e   1/1     Running             0          16s
examplekubernetesexecutorstarttask-421b2c62c0d648feb41f95e4ed7fed5e   0/1     Completed           0          2m16s
examplekubernetesexecutorstarttask-421b2c62c0d648feb41f95e4ed7fed5e   0/1     Terminating         0          2m18s
examplekubernetesexecutorstarttask-421b2c62c0d648feb41f95e4ed7fed5e   0/1     Terminating         0          2m18s
examplekubernetesexecutortwotask-a03683ccd95540b3af07df55a7994192     0/1     Pending             0          0s
examplekubernetesexecutortwotask-a03683ccd95540b3af07df55a7994192     0/1     Pending             0          0s
examplekubernetesexecutortwotask-a03683ccd95540b3af07df55a7994192     0/1     ContainerCreating   0          2s
examplekubernetesexecutorthreetask-dfa28e1a719e4c839b867f0d5d3942ea   0/1     Pending             0          1s
examplekubernetesexecutorthreetask-dfa28e1a719e4c839b867f0d5d3942ea   0/1     Pending             0          1s
examplekubernetesexecutorthreetask-dfa28e1a719e4c839b867f0d5d3942ea   0/1     ContainerCreating   0          2s
examplekubernetesexecutoronetask-0206ade847e9491dbb380af97c19c130     0/1     Pending             0          1s
examplekubernetesexecutoronetask-0206ade847e9491dbb380af97c19c130     0/1     Pending             0          3s
examplekubernetesexecutoronetask-0206ade847e9491dbb380af97c19c130     0/1     ContainerCreating   0          3s
```

OICNDI

Summary

Summary

Kubernetes Executor offers a new way to execute Airflow tasks, making them flexible and resilient of failures.

Terima Kasih

Sponsored by:

vmware boer technology ONF

OICNDI

November 7th 2020