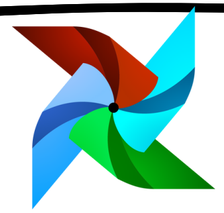
Intelligent Data Analytics



AIRFLOW

Presented by DKN 2002

THE TEAM

Nguyen Vu Khoi

Member

Bui Van Nghia

Member

Dang Minh Duc

Member

CONTENT

SECTION 1

Getting started with Airflow: introduction, core, architecture, how does it work, implement

SECTION 2

Airflow UI: all of main view page to monitor task

SECTION 3

Demo simple data pipeline with Airflow: config Dag, create table, use API, hook

SECTION 4

Scheduling Dag with depend on Dataset trigger

SECTION 5

Executor's categorical and parralel Dag

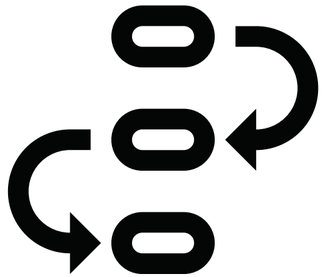
SECTION 6

Group task and Airflow plugins

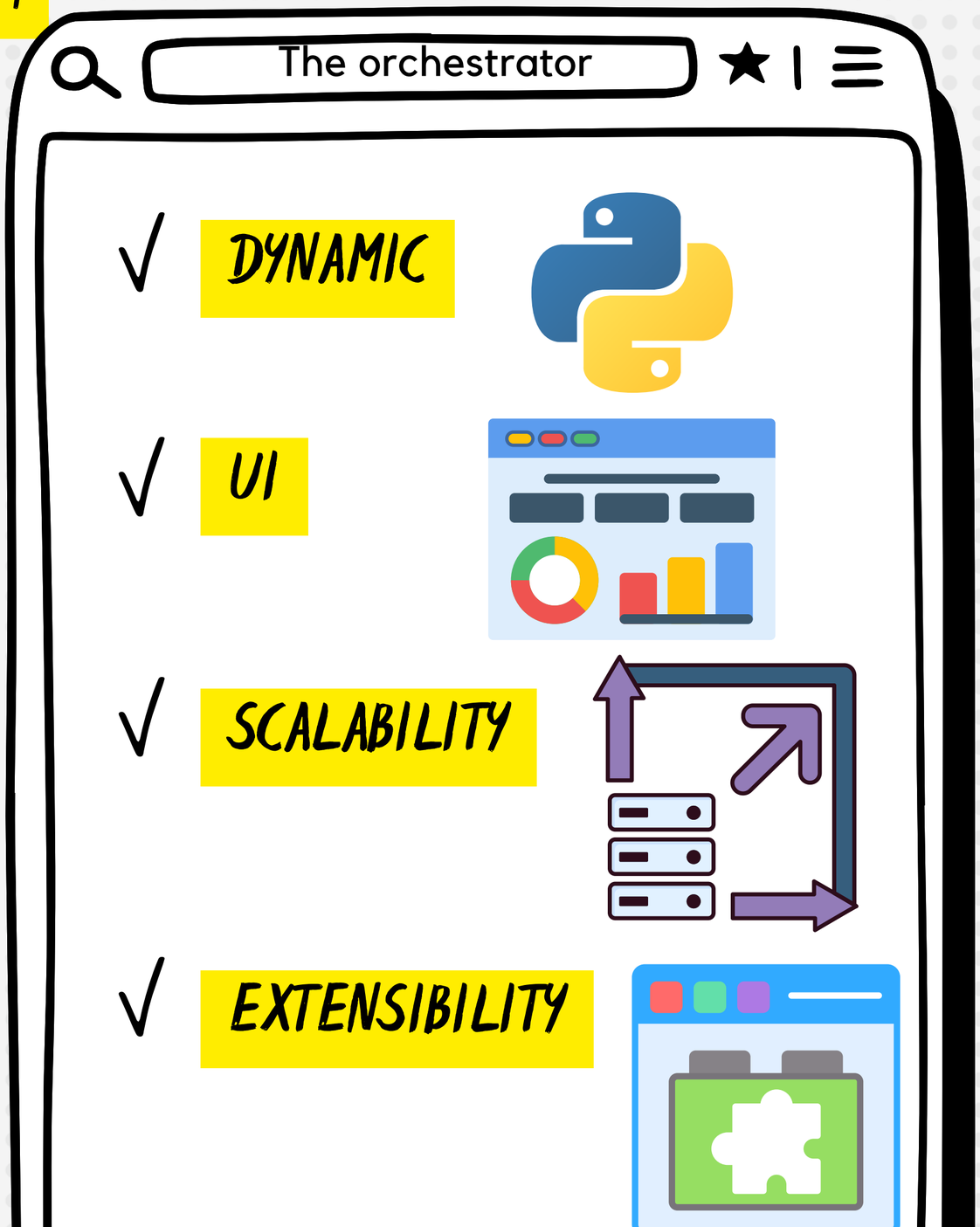
SECTION 1

USE CASE

Data pipeline runs everyday at 10P.M



SECTION 1



Apache Airflow

WHAT IS ?

Apache Airflow is an open source platform to programmatically author, schedule and monitor workflows

SECTION 1

Apache Airflow

CORE COMPONENT

In addition, we have Executor, Queue, Worker,...

WEB SERVER

access User Interface

SCHEDULER

schedule task, pipeline

METASTORE

save metadata and workflows

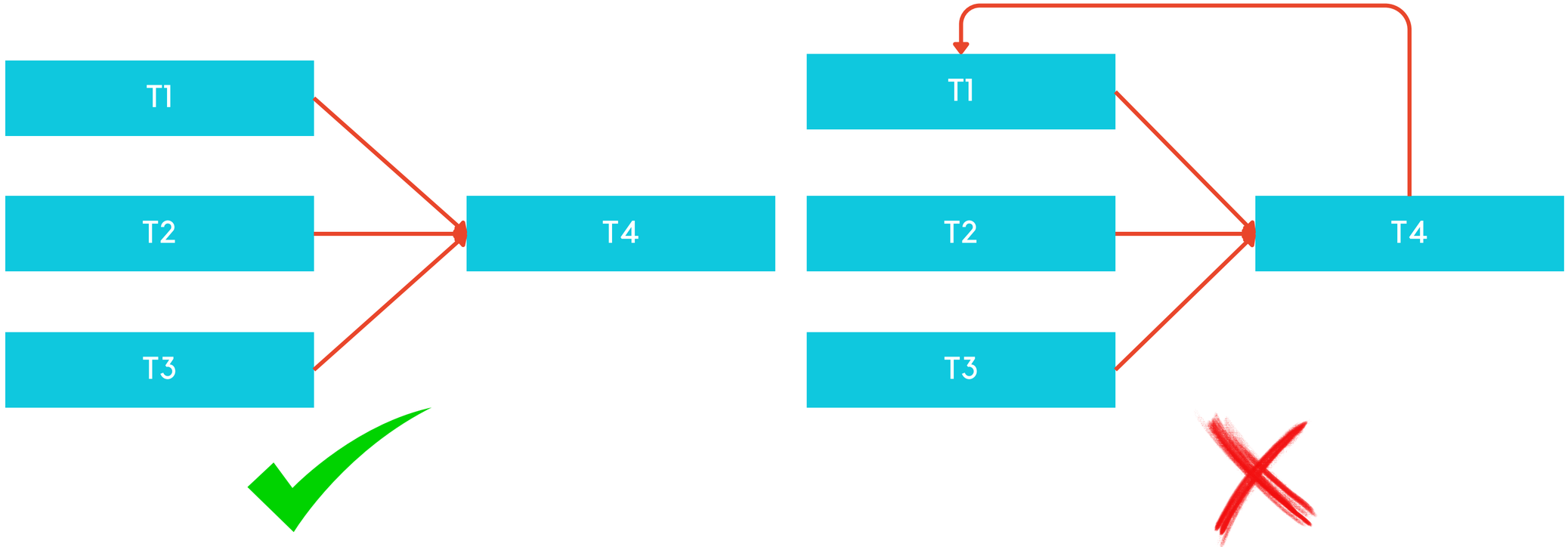
TRIGGERER

trigger specify of task

SECTION 1

Apache Airflow

Directed Acyclic Graph



SECTION 1

Apache Airflow

Operator

Action Operator	BashOperator, PythonOperator
Transfer Operator	ExternalTaskSensor, HttpSensor
Action Operator	S3ToRedshiftTransfer, MySqlToHiveTransfer

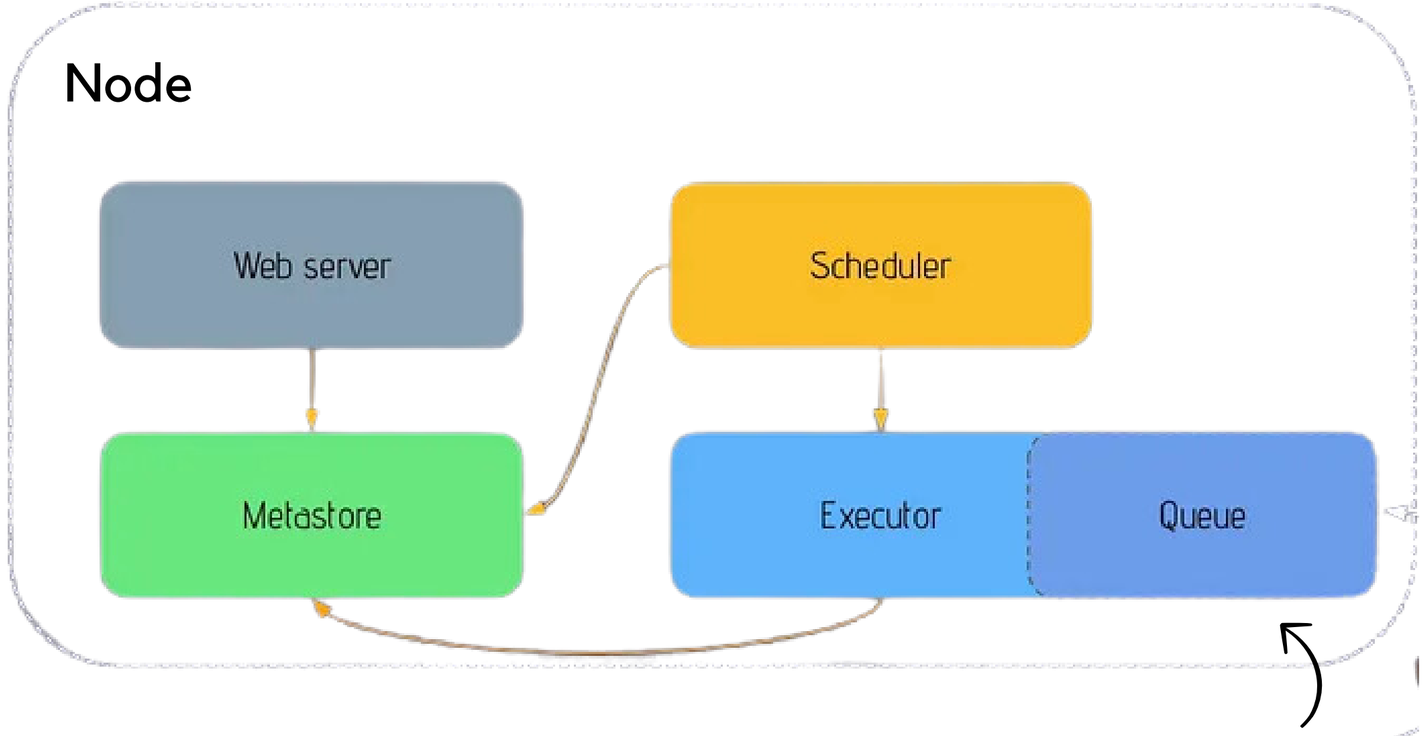
SECTION 1


AIRFLOW IS NOT A DATA STREAMING SOLUTION
NEITHER A DATA PROCESSING FRAMEWORK

SECTION 1

Apache Airflow

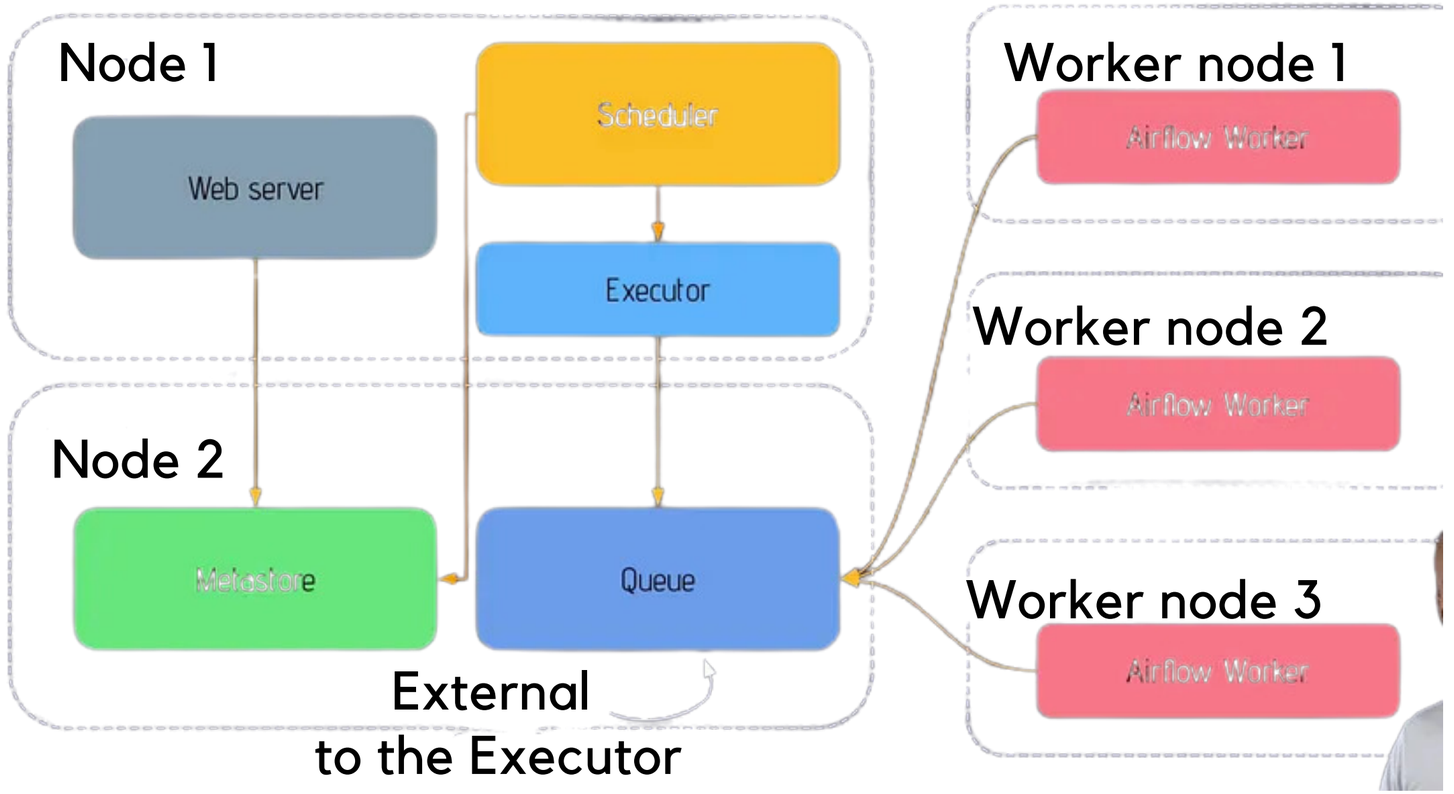
One node



SECTION 1

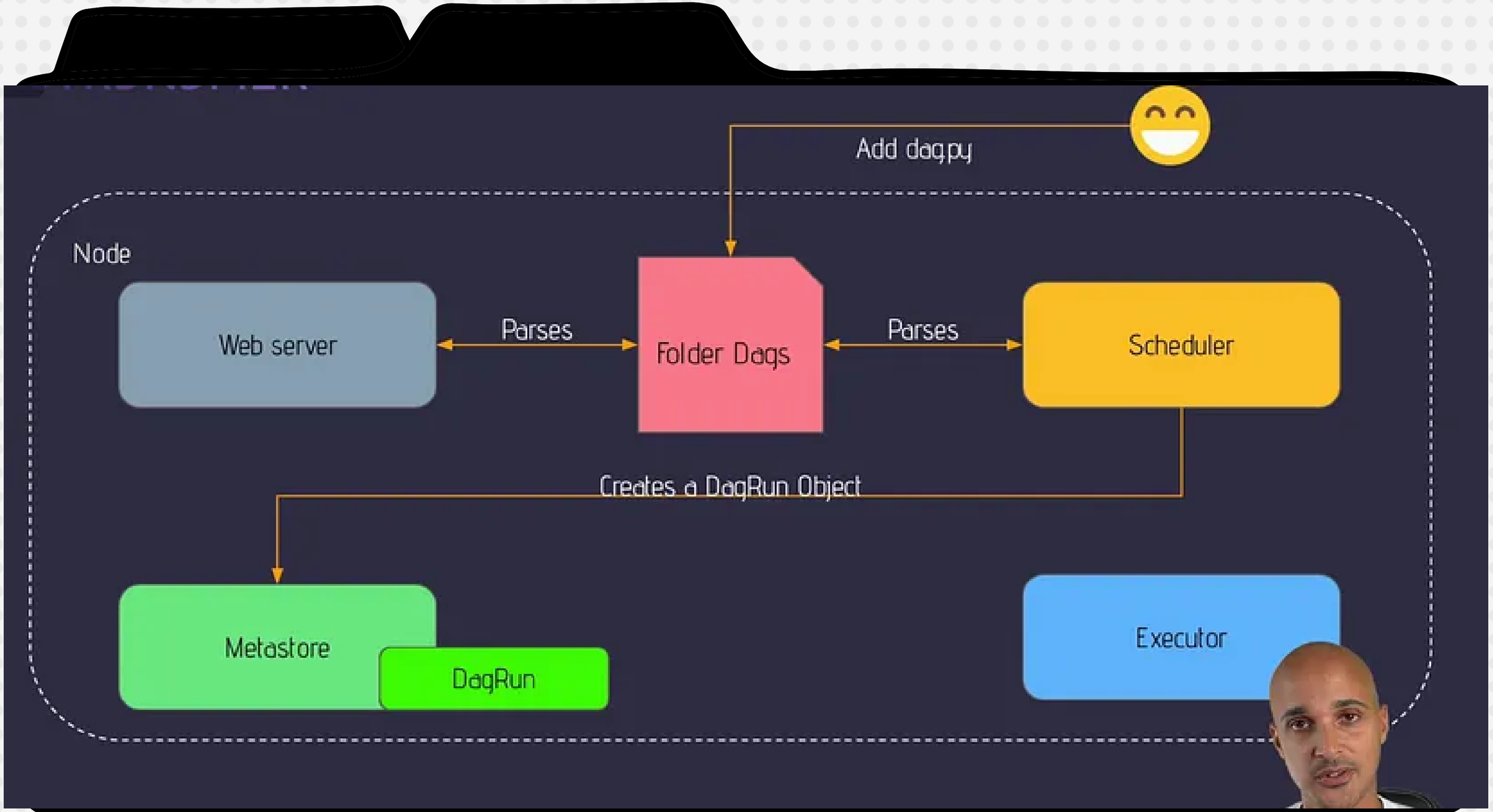
Apache Airflow

Multi node (celery)



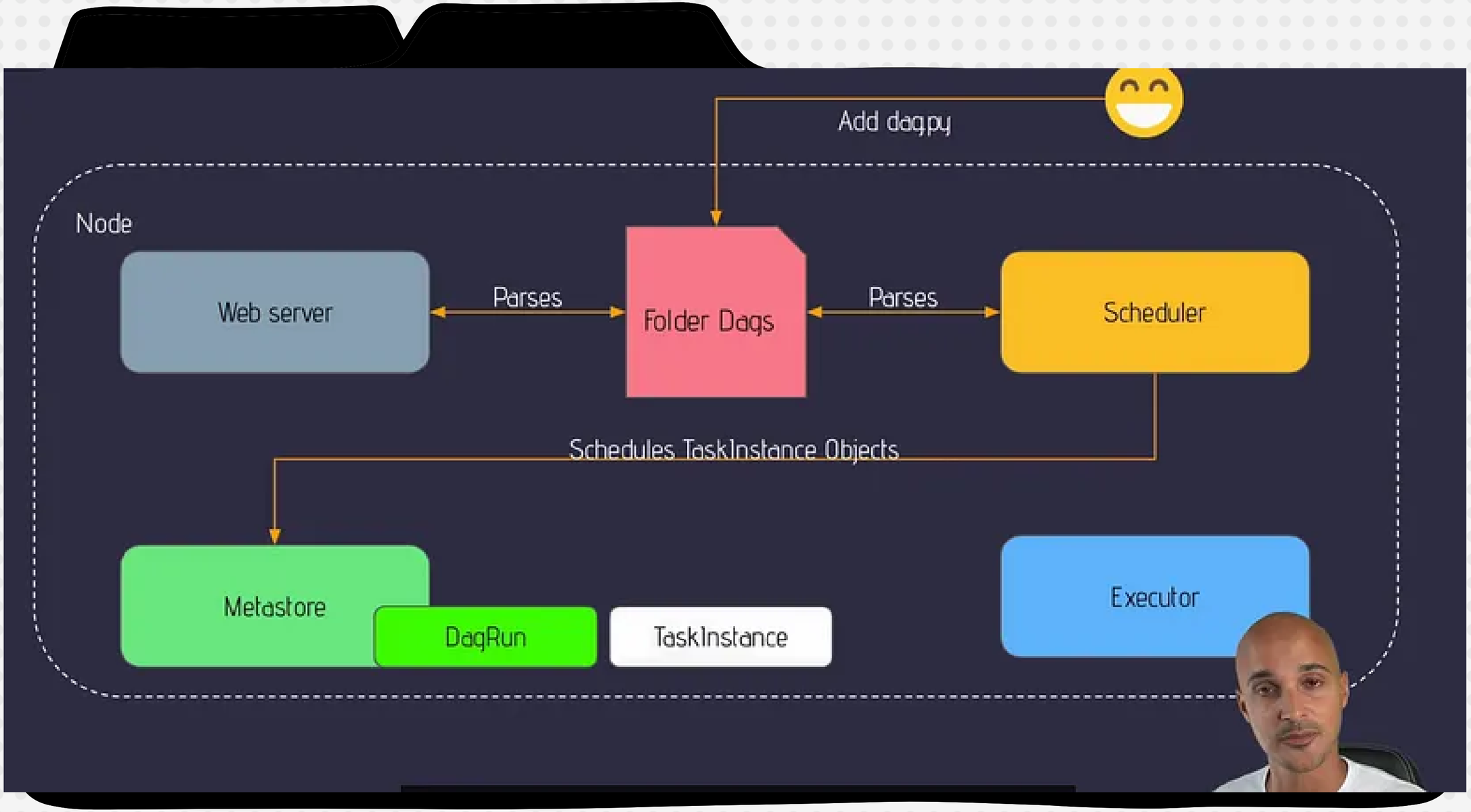
SECTION 1

Apache Airflow



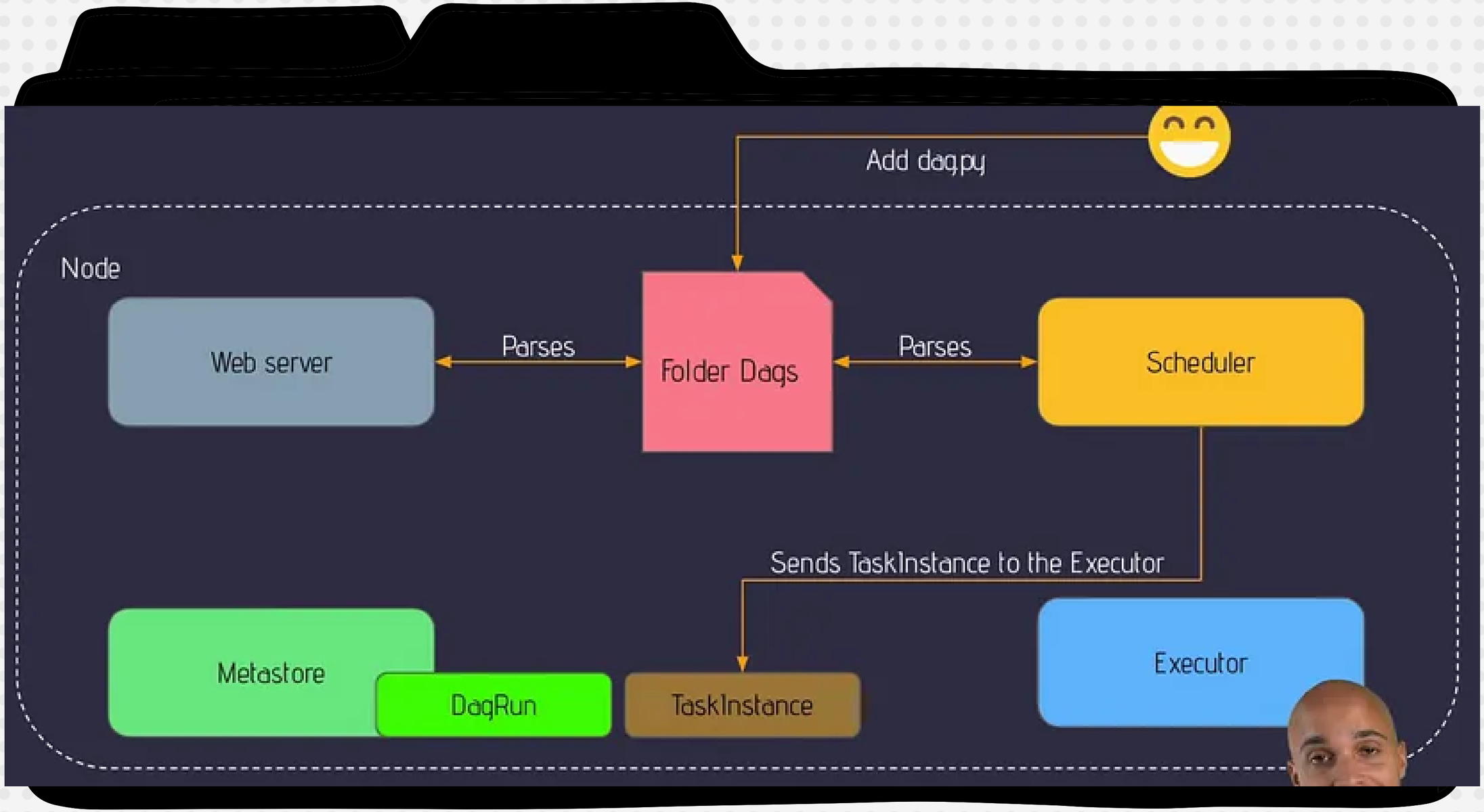
SECTION 1

Apache Airflow



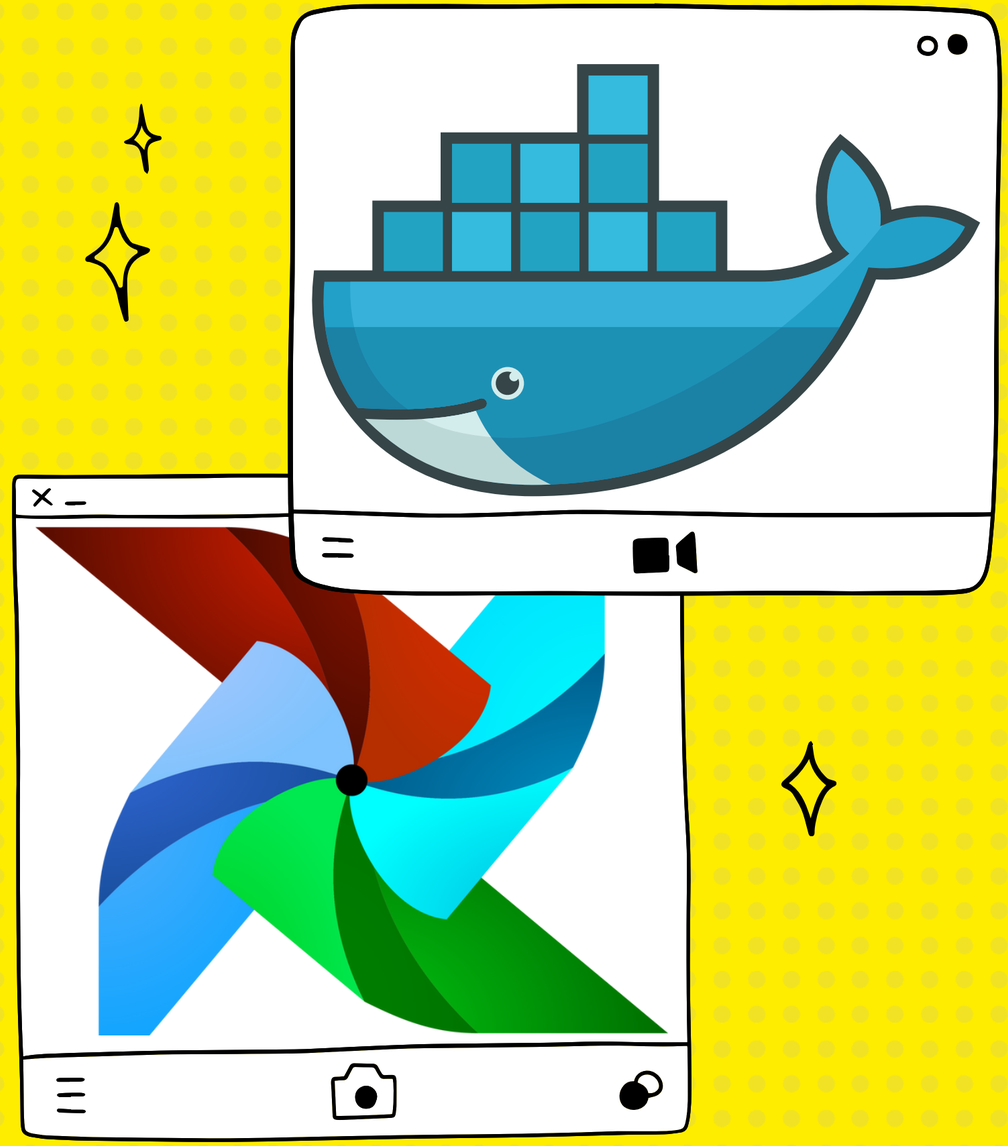
SECTION 1

Apache Airflow



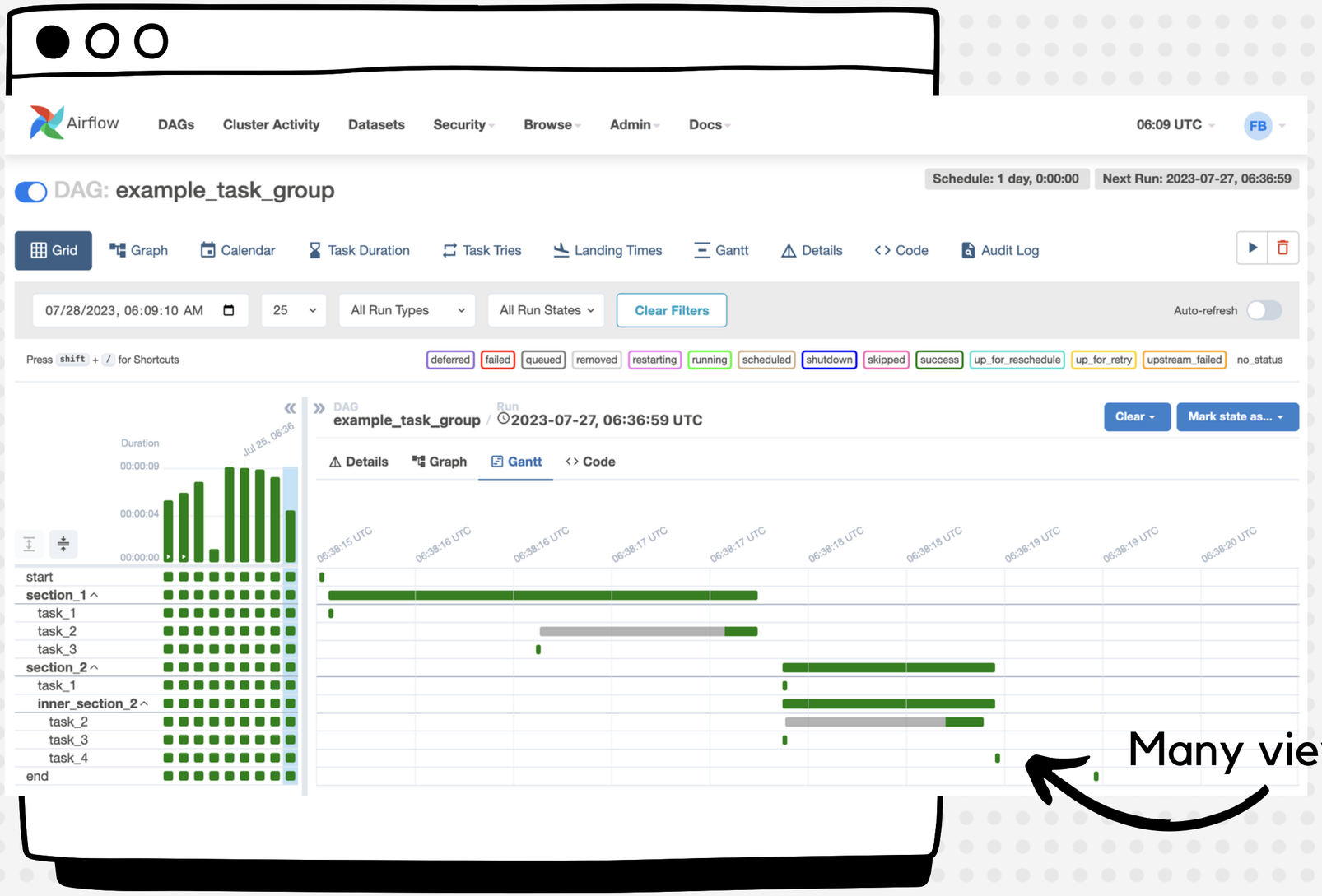
IMPLEMENT

Use Docker to deploy by docker-compose



SECTION 2

USER INTERFACE



Many view tab in Airflow's UI

SECTION 3

Apache Airflow

DAG

parameter

CREATE TABLE TASK

connection, parameter

CREATE IS_API_AVAILABLE

connection, sensor parameter

CREATE EXTRACT_USER

SIMPLE DATAPIPELINE

simple pipeline with create table, call api, extract data, load and transform, store to database

file: user_processing.py

SECTION 3

Apache Airflow

CREATE PROCESS_USER

use pre-defined function, use xcom

CREATE STORE_USER

postgresHook

- docker exec -it material-2_postgres_1 /bin/bash
- psql -Uairflow
- SELECT * FROM users;

SIMPLE DATAPIPELINE

simple pipeline with create table, call api, extract data, load and transform, store to database

file: user_processing.py

SECTION 4

Apache Airflow

SCHEDULING DAG WITH DEPEND ON DATASET TRIGGER

- Dataset view
- consumer.py & producer.py

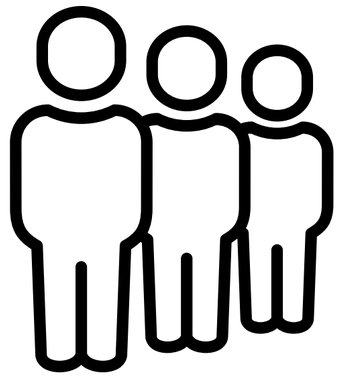
- DAGs can only use Datasets in the same Airflow instance. A DAG cannot wait for a Dataset defined in another Airflow instance.
- Consumer DAGs are triggered every time a task that updates datasets completes successfully. Airflow doesn't check whether the data has been effectively updated.
- You can't combine different schedules like datasets with cron expressions.
- If two tasks update the same dataset, as soon as one is done, that triggers the Consumer DAG immediately without waiting for the second task to complete.
- Airflow monitors datasets only within the context of DAGs and Tasks. If an external tool updates the actual data represented by a Dataset, Airflow has no way of knowing that.

SECTION 5

PARRALEL TASK

- SequentialExecutor
- CeleryExecutor
- parralel_dag.py

- docker-compose down && docker-compose --profile flower up -d
- localhost:5555
- trigger dag
- add new worker, check in worker -> queue -> name
- check task send to new queue



SECTION 6

GROUP TASK

- groups folder
- dowload_tasks()
- transform_tasks()
- group_dag.py

XCOM TASK

- share data with xcoms
- xcom_dag.py
- check xcom view
- trigger rule to specific path of pipeline

SECTION 6

ELASTIC PLUGINS

- ElasticSearch
- docker-compose-es.yaml
- docker-compose -f docker-compose-es.yaml up -d
- docker-compose -f docker-compose-es.yaml p
- localhost:9200
- docker exec -it material-2_elastic_1 /bin/bash
- curl -X GET 'http://elastic:9200'

- how plugins work (lazy loaded)
- create the connection
- elastic_hook.py

- add elastic hook to plugins system
- docker exec -it material-2_airflow-scheduler_1 /bin/bash
- airflow plugins
- add AirflowElasticPlugin
- restart -> check again
- add dag

REFERENCE

1 Udemy: The Complete Hands-On Introduction to Apache Airflow

2 Medium: Introduction Lifecycle of a DAG

THANK YOU