Airflow

Airflow is an orchestrator, not a processing framework, process your gigabytes of data outside of Airflow (i.e. You have a Spark cluster, you use an operator to execute a Spark job, the data is processed in Spark).

// Airflow is an orchestrator allowing you to execute your tasks in the right time, right way at the right order.

**Benefits:**

-- Dynamic - Python

-- Highly scalable - using Kubernetes too

-- UI - Monitor DP, interact tasks

-- Extensible - No need to wait for inbuilt plugin, we can build/ customize our instance

**Core components:**

-- Web Server : Flask Server

-- Scheduler: Deamon incharge of scheduling DP (Heart of Airflow)

-- Metastore: DB usually Postgres, responsible to store metadata, tasks and so on

-- Executor: How tasks are executed - Using Kubernetes tasks, executes

-- Worker: The process/sub-process executes the tasks

An Executor defines how your tasks are execute whereas a worker is a process executing your task

The scheduler schedules your tasks, the web server serves the UI, the database stores the metadata of Airflow.

**Core Component:**

**DAG:**

A DAG is a data pipeline

1->2->3 (flow should not be looped)

**OPERATOR:** - an Operator is a task.

A wrapper around the class to do operations, connecting a database will require an operator that passes data.

3 types:

-- Action Operator: Executing - Functions/commands (Bash Operator/ Python Operator)

-- Transfer Operator: Allow to trasfer data from S->D Presto data to Mysql

-- Sensor Operator: Wait for something to perform -> Example: File sensor waits for some files to be in the folder to perform

**Task/Task Instance:**

-- As soon as a task is triggered in DP, that task becomes a task instance.

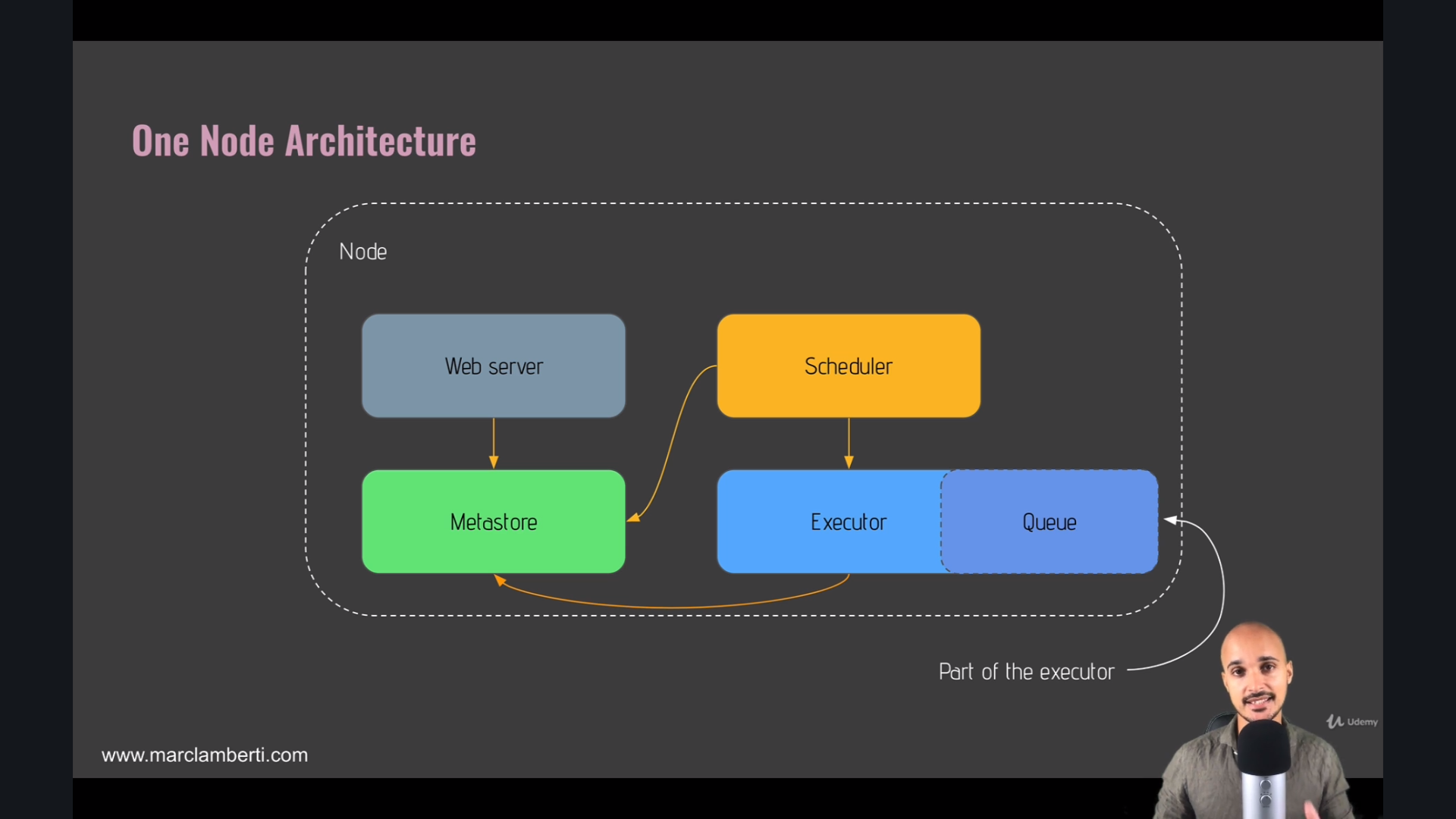
-- A executor also becomes task instance

**WORKFLOW:**

All put together is called a workflow

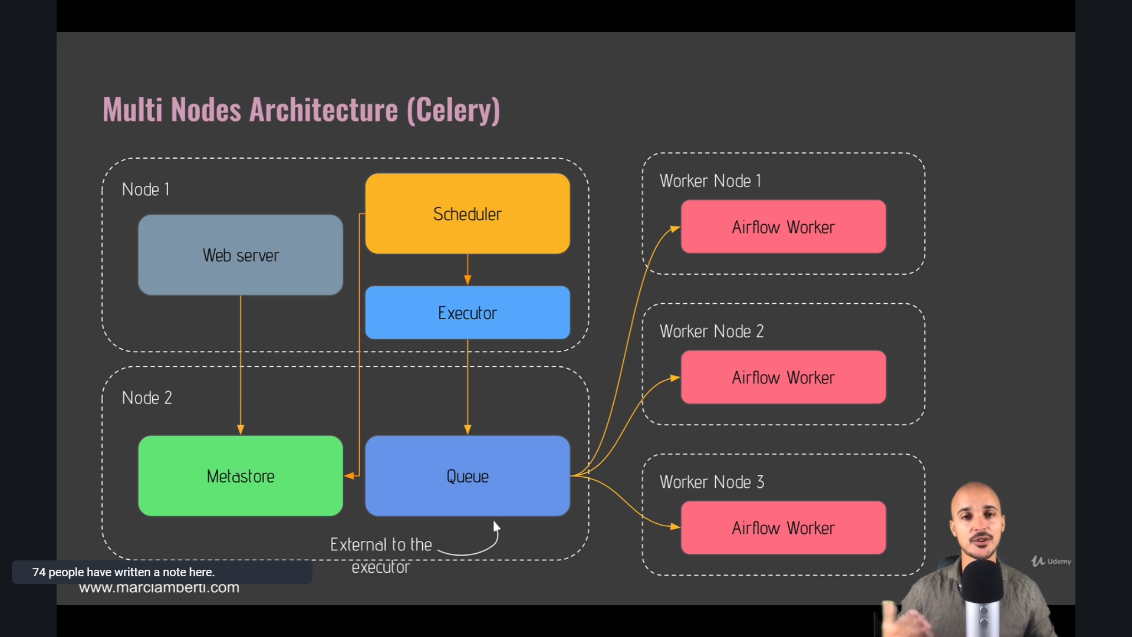
**## ARCHITECTURE**

Single Node:

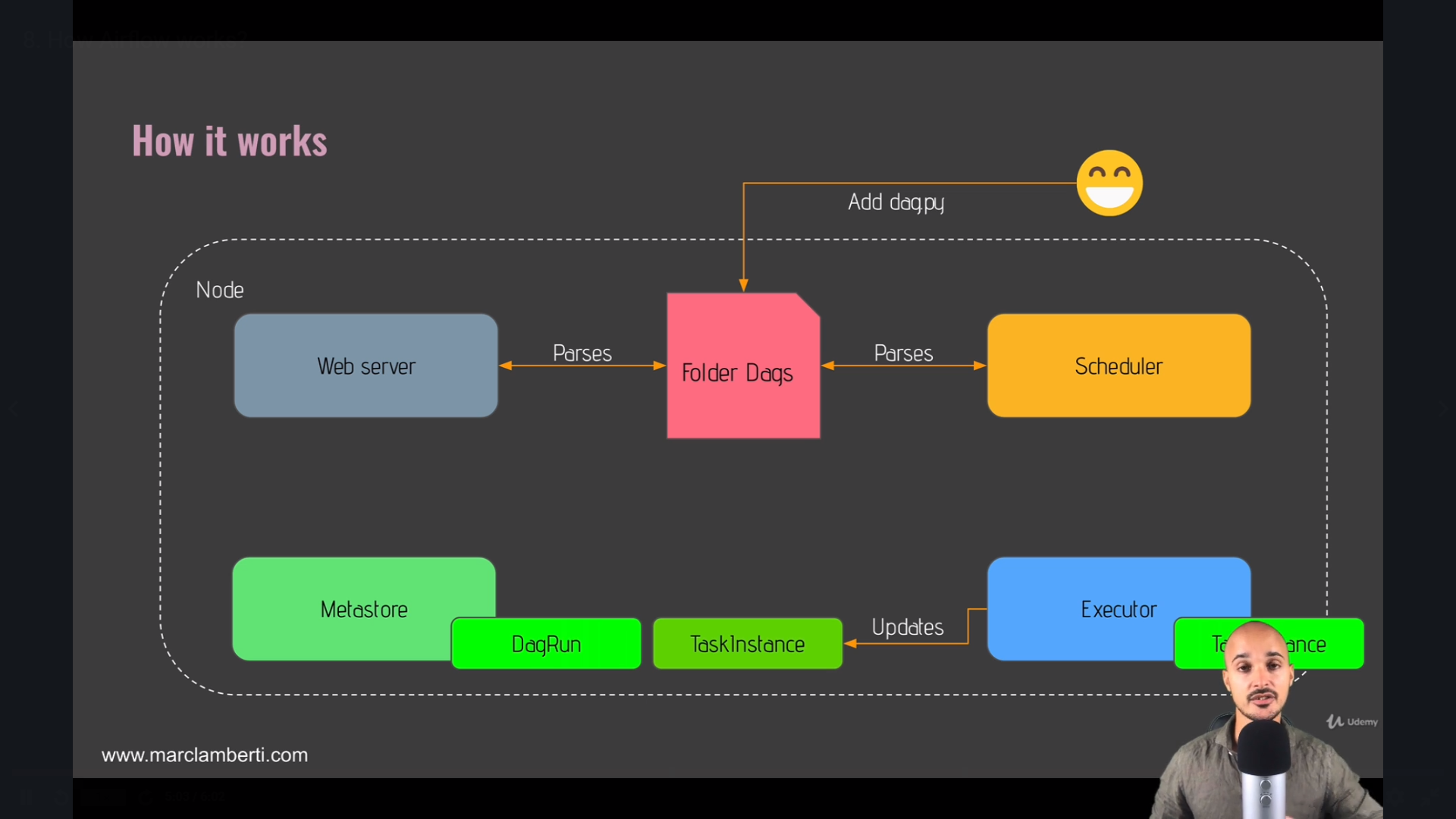


webserver // Scheduler // Metastore // Executor

Multi Node:

* Queue will be in a separate node and not in the Executor itself
* The worker nodes will fetch the tasks from the queue and executes them

New DP – we will add to the **folder dags**



Scheduler and Webserver parses the Folder dags if we put new files/dag files in the Folder Dags.

* DAG Run object is created when ready by the Scheduler (DP is triggered ) It is an initialization of the DAG. It creates a status Running in the metastore.
* When Task is ready for your DP the Task instance object is created
* Task Instance will be sent to the Executor by the Scheduler
* Executor runs the task instance and it keeps updating the status of Task instance Objects to the metastore
* If Successful, the DAG Run Object status is set to Completed
* UI shows the status

airflow db init is the first command to execute to initialise Airflow

If a task fails, check the logs by clicking on the task from the UI and "Logs"

The Gantt view is super useful to sport bottlenecks and tasks are too long to execute