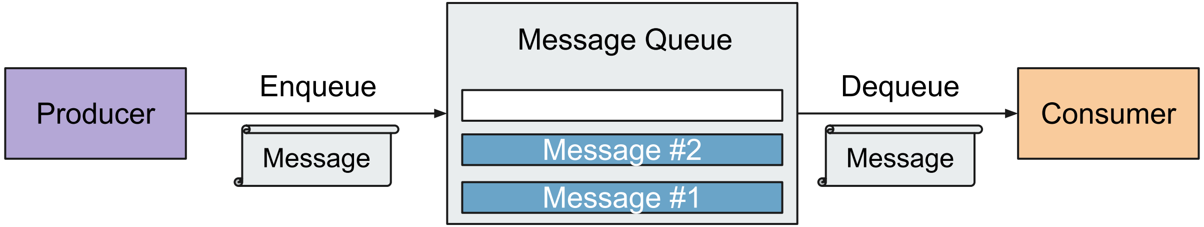
Celery Executor with PostgreSQL and RabbitMQ

Distributed mode

What is RabbitMQ

- RabbitMQ is an open source message queuing software.
- Allow to create queues where applications can be connected to in order to consume messages from these queues.
- Messages (data) placed onto the queue are stored until the consumer (a tierce application) retrieves them.
- A basic architecture of a message queue would be:
  - Client applications called producers, create messages and deliver them to the message queue (broker).
  - Other applications called consumers, connect to the queue and subscribe to the messages to process them.



www.marclamberti.com

What is a Celery Executor

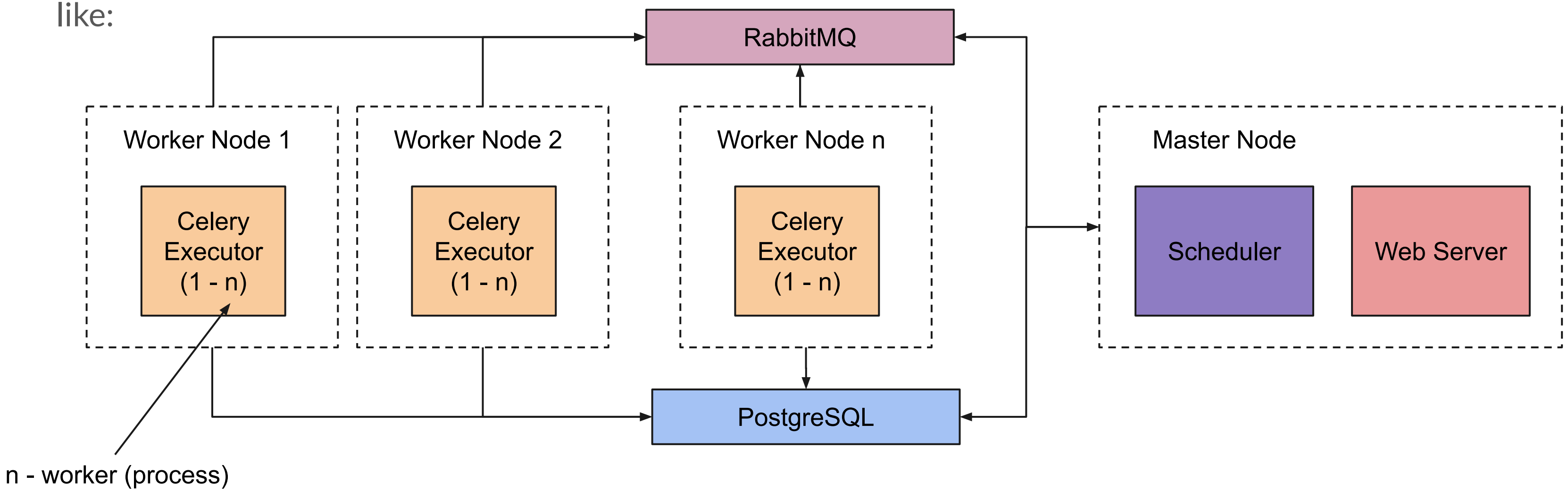
- Celery Executor is recommended for production use of Airflow.
- It allows distributing the execution of task instances to multiple worker nodes.
- Celery is a Python Task-Queue system that handle distribution of tasks on workers across threads or network nodes.
- The tasks need to be pushed into a broker like RabbitMQ, and Celery workers will pop them and schedule task executions.

Celery Executor + RabbitMQ + PostgreSQL

- As we have seen previously, PostgreSQL is a database allowing multiple concurrent clients to connect in both read and write modes
- Celery Executors allow to interact with Celery backend in order to distribute and execute task instances on multiple worker nodes giving a way to high availability and horizontal scaling.
- Celery needs to use a broker in order to pull out from the worker nodes the task instances to execute and that’s why we need to use RabbitMQ.

Schema

Here is an example of a basic view of what the architecture of Apache Airflow in distributed mode looks like:



RabbitMQ
Worker Node 1
Celery Executor (1 - n)
Worker Node 2
Celery Executor (1 - n)
Worker Node n
Celery Executor (1 - n)
Master Node
Scheduler
Web Server
PostgreSQL
n - worker (process)