traveloka

Cronjob on Steroids - Running Your Scheduled Program with Apache Airflow

Rezha Julio
Data Engineer

27 July 2019

Who am I ?

traveloka

- Education
  - (2014) B.Sc in Chemistry - ITB

- Working Experience
  - (2014-2016) Full Stack Developer - Polatic
  - (2016-Present) Data Engineer - Traveloka

Contact me at rezha@python.or.id for Python ID related

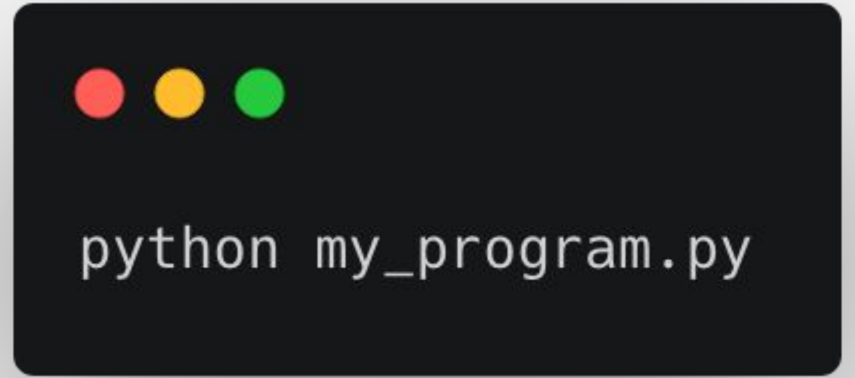
Table of Contents

traveloka

Table of Contents

traveloka

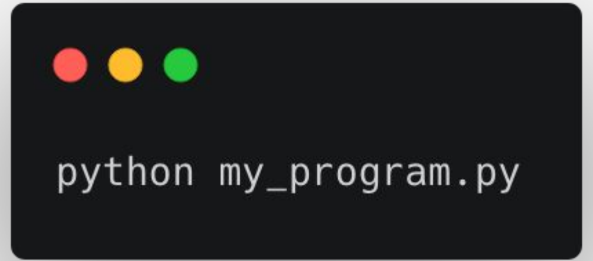
The Problems



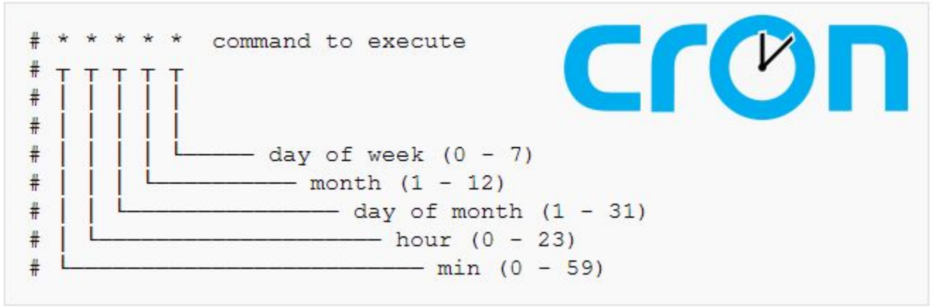
```
python my_program.py
```

Our Program is Simple

traveloka

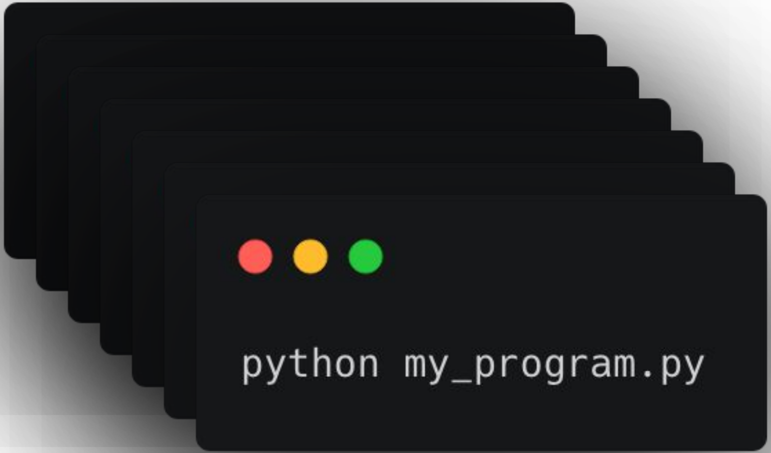


```
python my_program.py
```

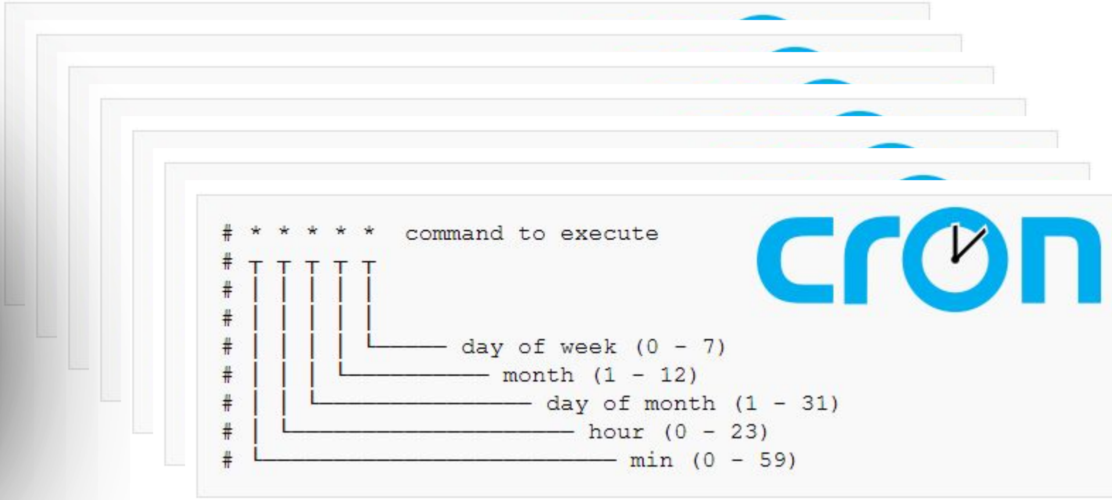


```
# * * * * *  command to execute
# ┬ ┬ ┬ ┬ ┬
# │ │ │ │ │
# │ │ │ │ │
# │ │ │ │ └───── day of week (0 - 7)
# │ │ │ └────────── month (1 - 12)
# │ │ └─────────────── day of month (1 - 31)
# │ └──────────────────── hour (0 - 23)
# └───────────────────────── min (0 - 59)
```

“Simple”



```
python my_program.py
```



```
# * * * * *  command to execute
# ┬ ┬ ┬ ┬ ┬
# │ │ │ │ │
# │ │ │ │ │
# │ │ │ │ └───── day of week (0 - 7)
# │ │ │ └────────── month (1 - 12)
# │ │ └─────────────── day of month (1 - 31)
# │ └──────────────────── hour (0 - 23)
# └───────────────────────── min (0 - 59)
```

The Problems

- Retry ?
- Error reportings ? Notifications ?
- Monitoring ?
- SLA ?
- Maintainability ?
- Scalability ?

Enter Airflow

traveloka

What is Airflow

traveloka

An open source platform to author, orchestrate and monitor processes

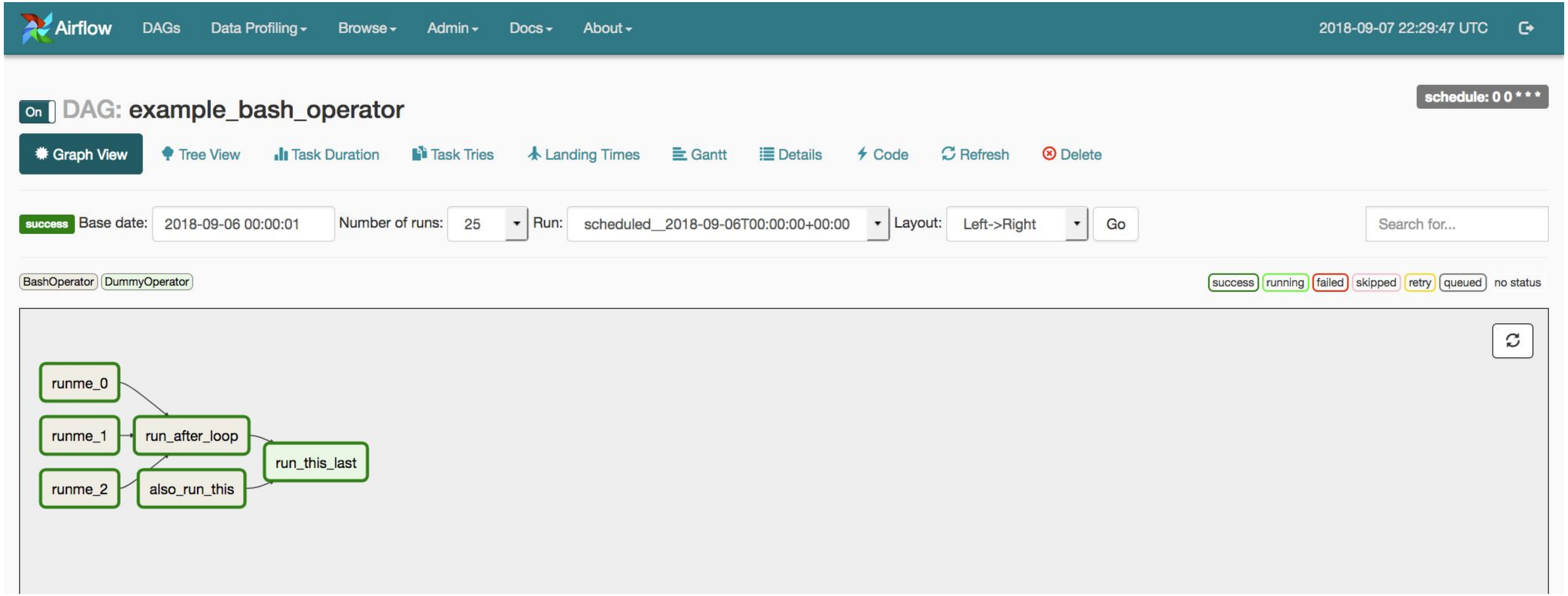
- It orchestrates tasks in a complex networks of job dependencies
- It’s Python all the way down
- It’s expressive and dynamic, workflows are defined in code
- Feature rich web interface
- Worker Process can be scaled vertically and horizontally
- Extensible

Concepts

- **Workflows** are called DAGs for *Directed Acyclic Graph*.
- **Tasks** : Workflows are composed of tasks called Operators.
- **Operators** can do pretty much anything that can be run on the Airflow machine.
- Operators classified in 3 categories : **Sensors**, **Operators**, **Transfers**.
  - BashOperator - executes a bash command
  - PythonOperator - calls an arbitrary Python function
  - EmailOperator - sends an email
  - SimpleHttpOperator - sends an HTTP request
  - MySqlOperator, SqliteOperator, PostgresOperator, MsSqlOperator, OracleOperator, JdbcOperator, etc. - executes a SQL command
  - Sensor - waits for a certain time, file, database row, S3 key, etc…

```python
dag = DAG(
    'tutorial',
    default_args=default_args,
    description='A simple tutorial DAG',
    schedule_interval=timedelta(days=1))
```

```
t1 = BashOperator(
    task_id='print_date',
    bash_command='date',
    dag=dag)
```



Airflow DAGs Data Profiling Browse Admin Docs About 2018-09-07 22:29:47 UTC

On DAG: example_bash_operator

schedule: 0 0 * * *

Graph View | Tree View | Task Duration | Task Tries | Landing Times | Gantt | Details | Code | Refresh | Delete

success Base date: 2018-09-06 00:00:01 Number of runs: 25 Run: scheduled__2018-09-06T00:00:00+00:00 Layout: Left->Right Go

Search for...

BashOperator DummyOperator

success running failed skipped retry queued no status

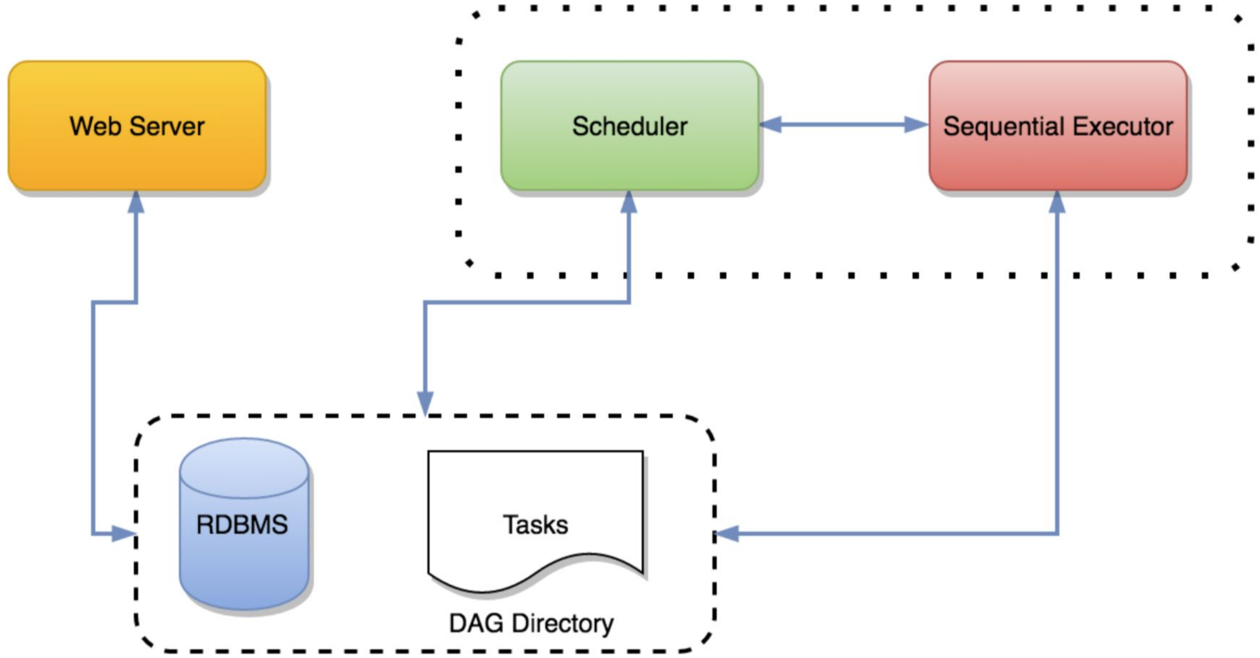
Table of Contents

traveloka

Architecture

Sequence Executor

- One CPU
- Using SQLite
- Not Recommended for production

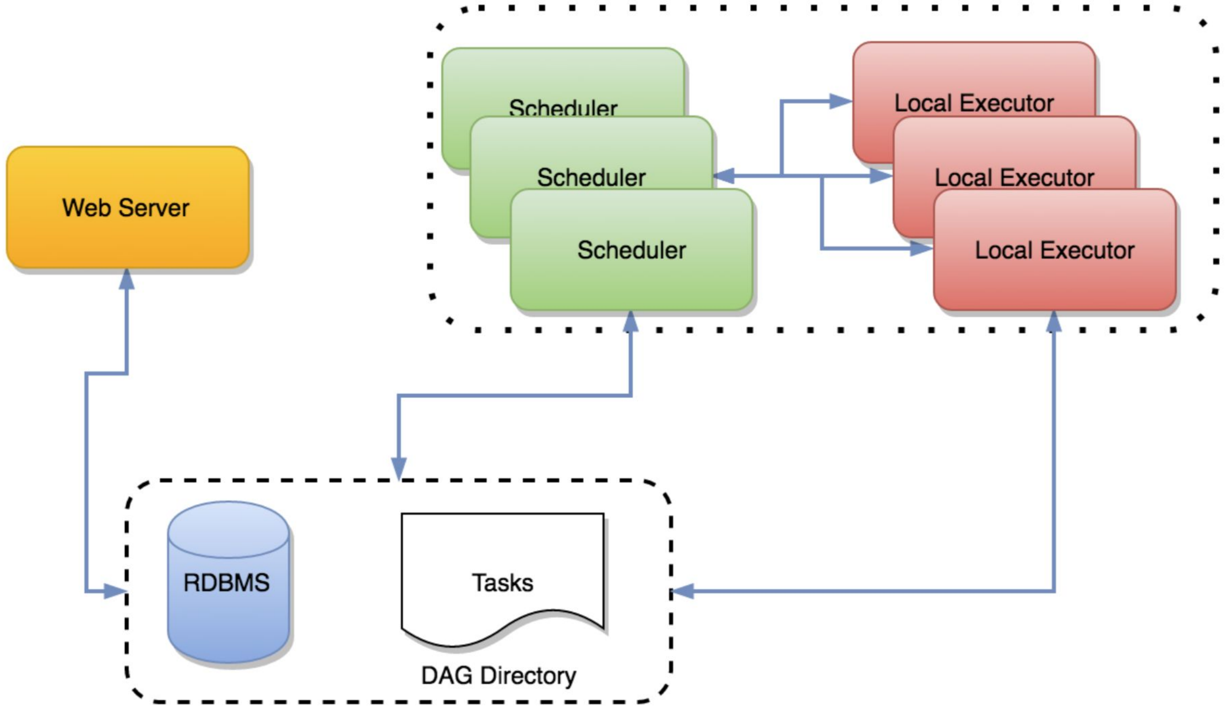


Architecture

traveloka

Local Executor

- Scales vertically
- Runs in threads allowing tasks parallelism
- Suitable for production usually when there's not so many DAGs

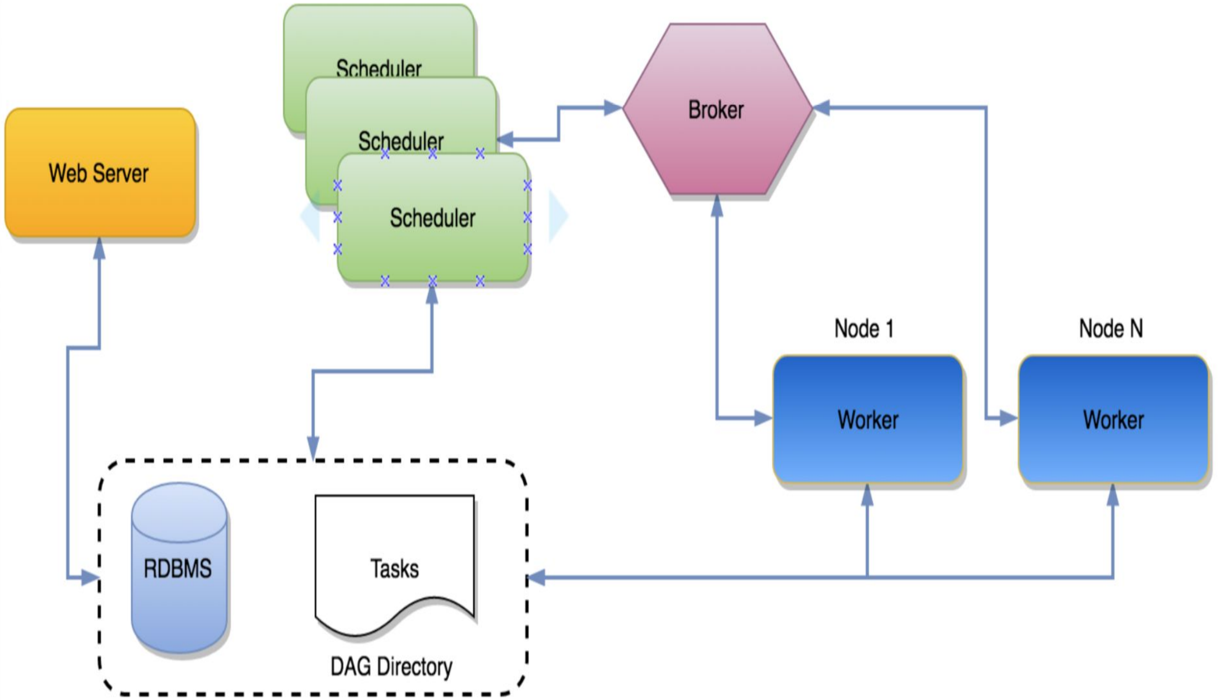


Architecture

traveloka

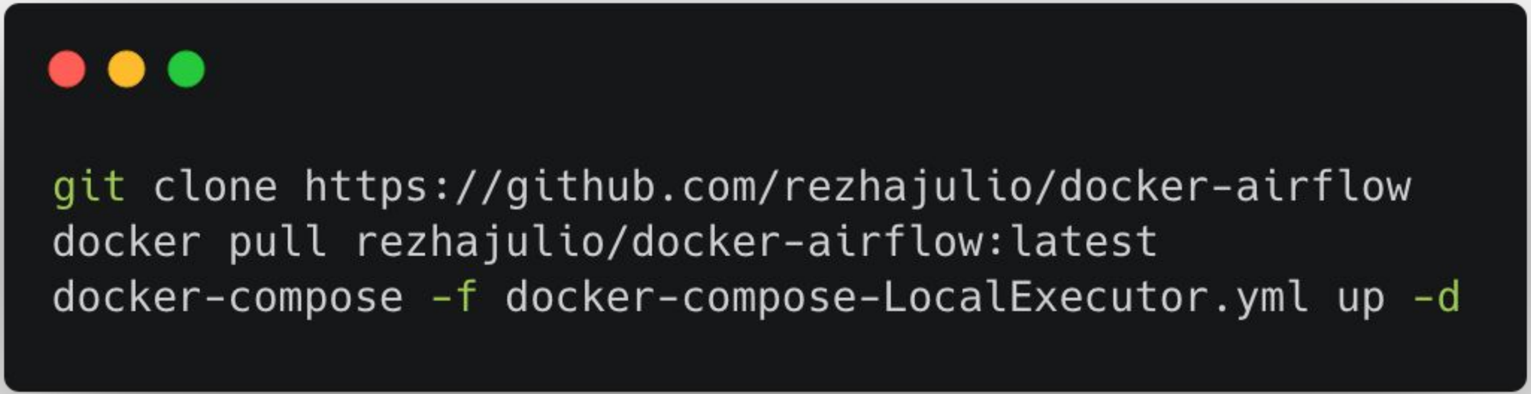
Celery Executor

- Scales a lot
- Each executor resides in one node
- Requires Celery to manage nodes and Redis or RabbitMQ for communication



Deploy

traveloka



```
git clone https://github.com/rezhajulio/docker-airflow
docker pull rezhajulio/docker-airflow:latest
docker-compose -f docker-compose-LocalExecutor.yml up -d
```

Table of Contents

traveloka

Demo

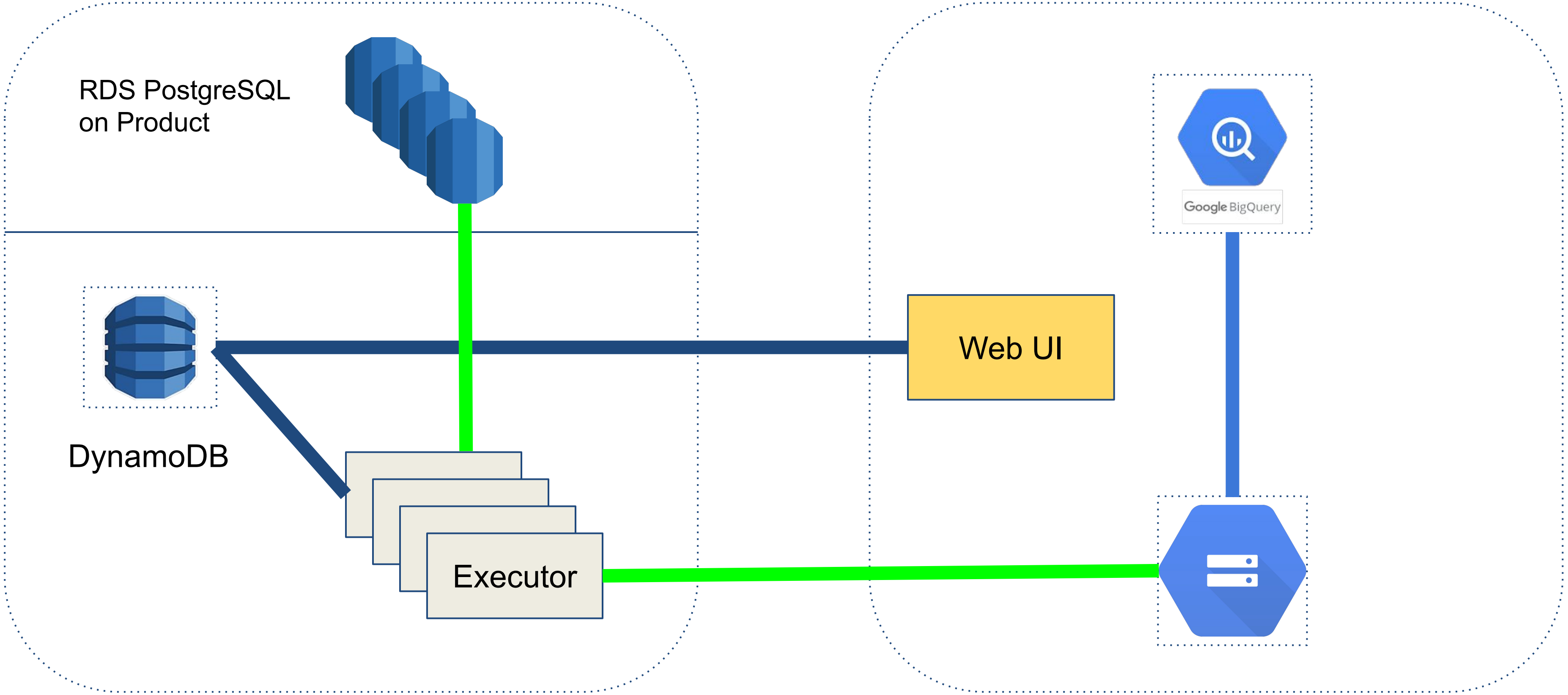
Talk is Cheap, Show Me the Code

Airflow @Traveloka scale

- We Run **500+ DAGs** with **~50k task** a day
- DAG running with **daily, 6 hourly** and **hourly** granularities
- **100+ Data Engineer + Analyst** authored or contributed to DAGs directly
- Using Celery Executor with RabbitMQ as backend

Use Case

traveloka



Use Case

traveloka

Config table

…

…

status : “TESTING”, “RELEASED”, …

granularity: “DAILY”, “HOURLY”, …

is_running: False, True

. . .

DAG

- Testing Pipeline

  status: “TESTING”

- Production Pipeline

  status: “RELEASED”

  granularity: “DAILY”, “HOURLY”, …

  is_running: True

Best Practice

- Enable the email feature and EmailOperator/SlackOperator for monitoring. Checkout the SLA feature to know when your jobs are not completing on time.

- The scheduler is still the weakest link as it is a single point of failure. Make a monitor for scheduler.

- As the number of jobs you run on Airflow increases, so does the load on the Airflow database.

- Try to make you tasks idempotent. Airflow will then be able to handle retrying for you in case of failure.

Resource

traveloka

- **Website:** https://airflow.apache.org/
- **Github:** https://github.com/apache/airflow
- **Chat:** https://apache-airflow-slack.herokuapp.com/
- **Mail list:** https://lists.apache.org/list.html?dev@airflow.apache.org

traveloka

Thanks!

traveloka

We Are Hiring!

rezha.pradana@traveloka.com