TECHTONIC
upshift your technology quotient

PRAMATI

Apache Airflow

For Data Engineering

- Sampath Kumar, Principle Engineer

Learning and Development
2019

Agenda

Goal of this session is to provide an overview of Airflow capabilities & how to use it for Data Engineering

- Introduction (DE, Airflow)
- Key Concepts (Key Words, Demo)
- Architecture (System Design)
- Features (Challenges, Loggins, Analytics)
- Challenges & Recommendations
- Q & A

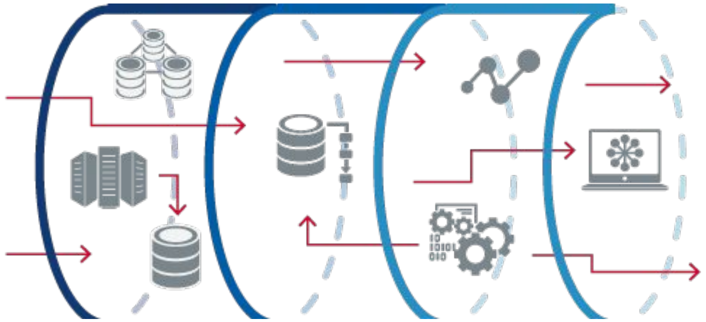
PRAMATI

TECH TONIC upshift your technology quotient

DATA ENGINEERING

Data Engineering is the **aspect of data science** that focuses on **practical applications of data collection and analysis**. **Data Engineers** are tasked with

- **Designing, building, testing, integrating, managing, and optimizing data** from a variety of sources
- **Build the infrastructure** and architecture that enable data generation
- Primary focus is to build **free-flowing data pipelines** by combining a variety of big data technologies that enable real-time analytics
- Data engineers also write complex queries to ensure that data is easily accessible

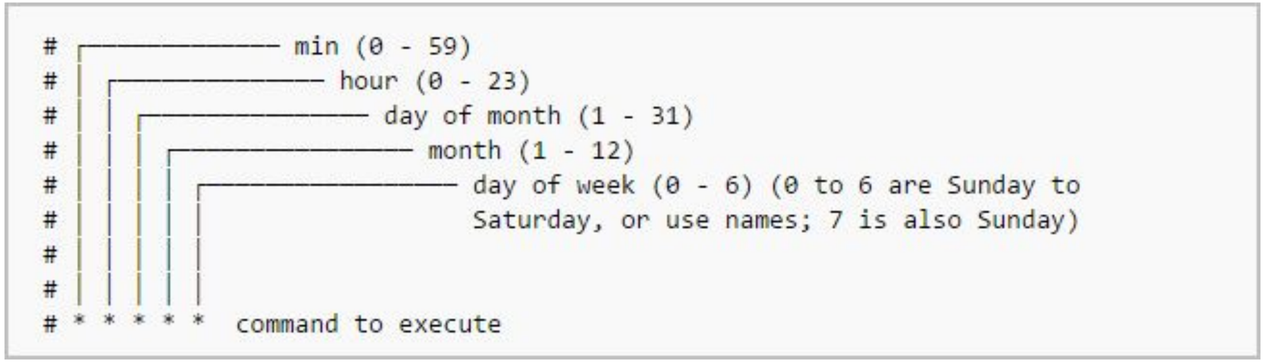


AIRFLOW

- Open-source workflow automation and scheduling system that can be used to author and manage your data pipelines.
- It started at Airbnb in October 2014 as a solution to manage the company's increasing complex workflows.
- License: Apache License 2.0
- Written in: Python
- Operating system: Microsoft Windows, macOS, Linux
- Stable release: 1.10.5 / August 30, 2019; 3 months ago

CRON

CRON: Derived from work CRONOS(means time), is a software for unix-like systems to schedule jobs based on time.



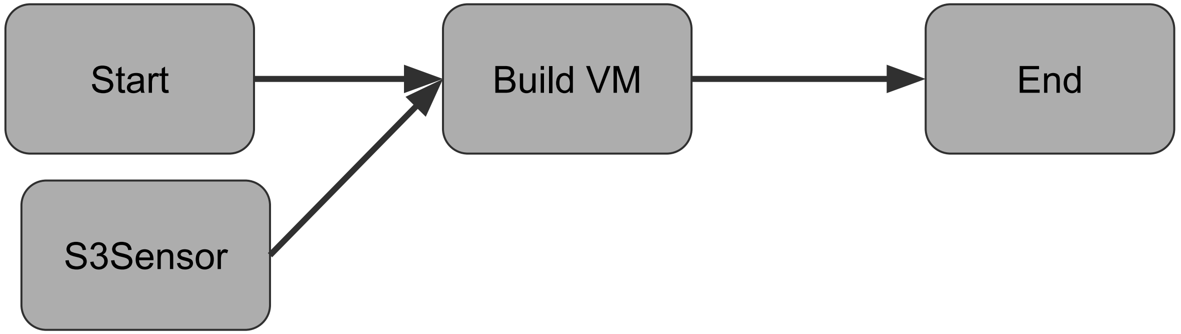
```
# ┌───────────── min (0 - 59)
# │ ┌────────────── hour (0 - 23)
# │ │ ┌─────────────── day of month (1 - 31)
# │ │ │ ┌──────────────── month (1 - 12)
# │ │ │ │ ┌───────────────── day of week (0 - 6) (0 to 6 are Sunday to
# │ │ │ │ │                  Saturday, or use names; 7 is also Sunday)
# │ │ │ │ │
# │ │ │ │ │
# * * * * *  command to execute
```

Airflow is cron on steroids: it allows you to schedule tasks to run, run them in a particular order, and monitor / manage all of your tasks.

KEY CONCEPTS

- DAGS
  - Directed acyclic graphs that represent tasks workflow
- Task
  - Operators - Bash, Python, SSH, Http, MySql, SparkSubmit, Sensors(s3), Docker, Hive, Slack,..
- Hooks
  - To store credentials for services - AWS, GCP, DataBase, Email,..
- Vars & XCom
  - For sharing any global values or inter-task communication

Sample - DAG

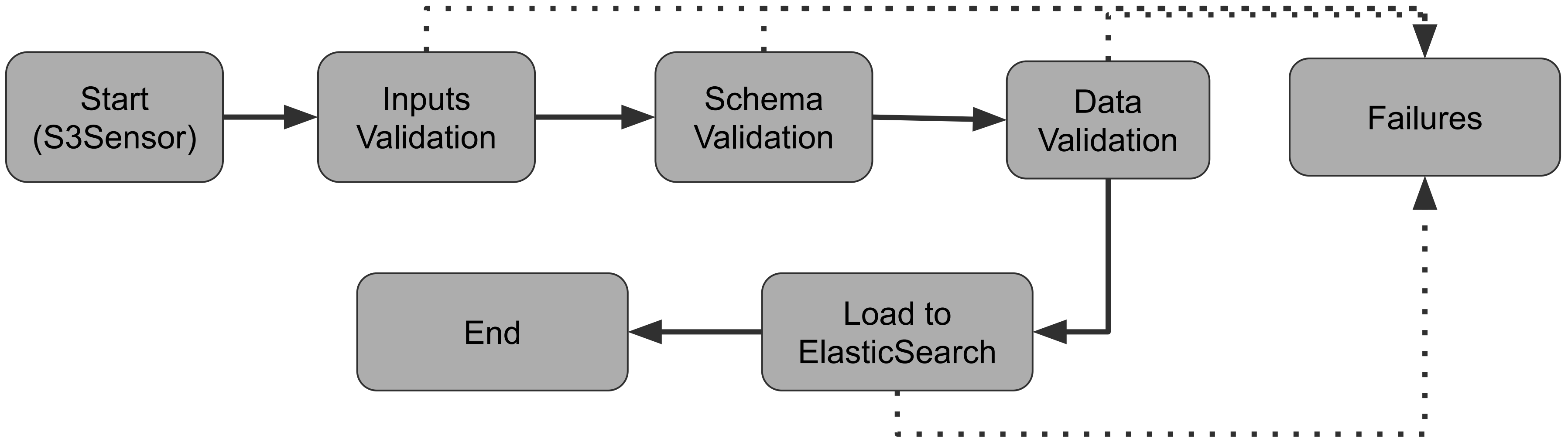


Goal: As and when user files come into AWS S3 bucket, start a high-spec docker VM and process the job. Prior to start and post process, job status to be notified.

Solutions required to be cost efficient and need to auto-scale if required.

DEMO - UI

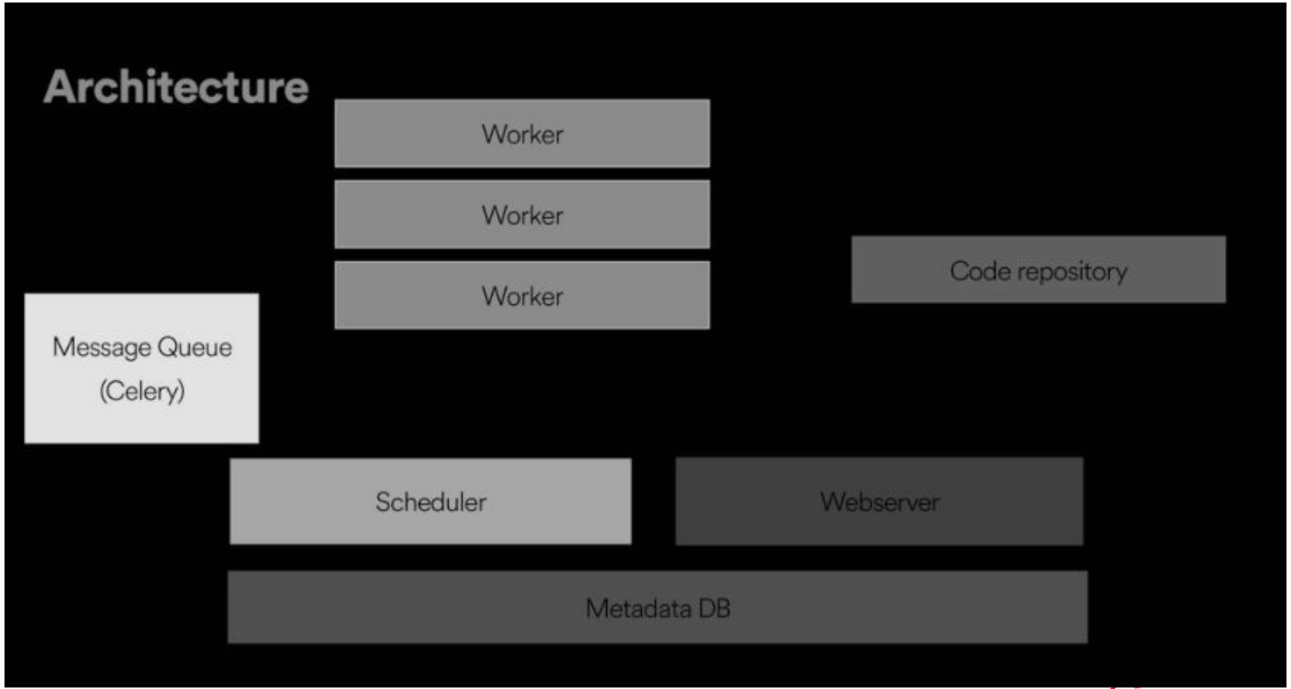
Sample - DAG



A data engineering pipeline, where an S3 sensor is used to identify the arrival of input file, following by several validation checks and then load into ElasticSearch which will be used for serving clients.

ARCHITECTURE

- MetaDB
- Message Broker
- Airflow Webserver
- Airflow Scheduler
- Airflow Workers

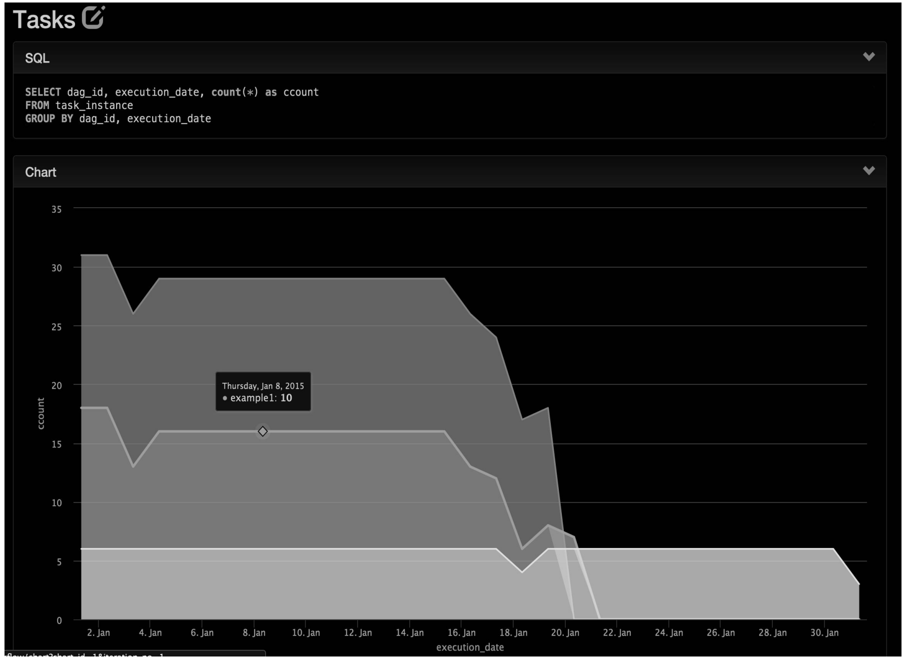


FEATURES

- Airflow Workers are Horizontally scalable
- Airflow Messaging Broker - Celery
- Airflow Integrations - GCP, Azure, AWS, Qubole & Databricks
- Hooks, Connections & Pools - Environment(dev/test/prod) friendly
- DAG - Dynamic sub dags & Branching

ANALYTICS

- Part of being productive with data is having the right weapons to profile the data you are working with.
- Airflow provides a simple query interface to write SQL and get results quickly, and a charting application letting you visualize data.



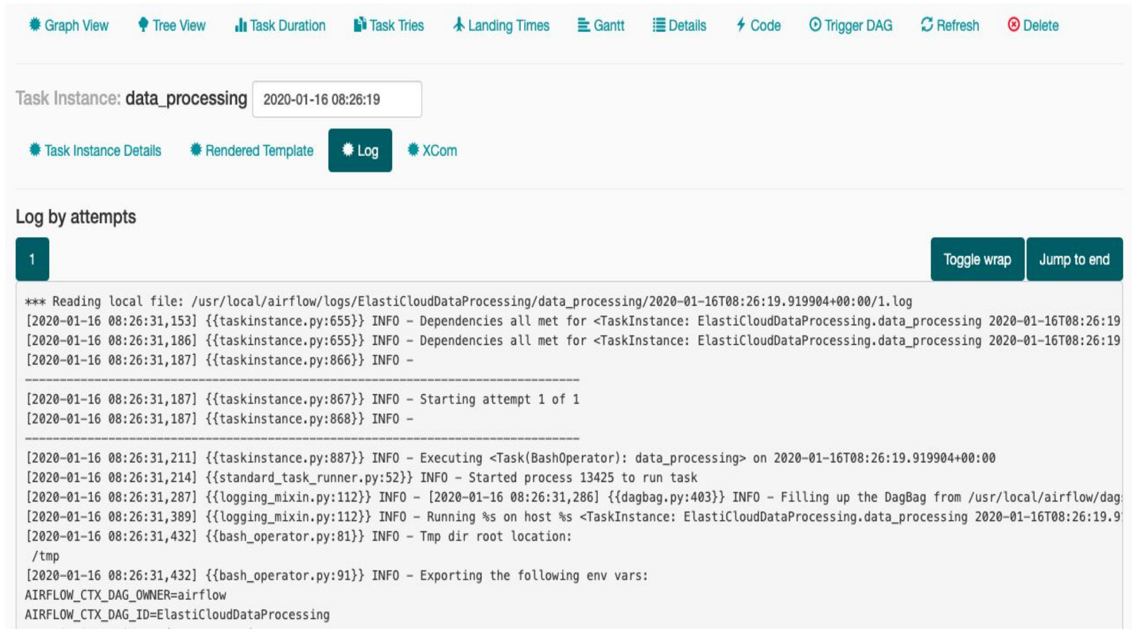
```
SELECT dag_id, execution_date, count(*) as ccount
FROM task_instance
GROUP BY dag_id, execution_date
```

PRAMATI

TECHTONIC
upshift your technology quotient

LOGGING

- Logging is visible from UI



Task Instance: data_processing 2020-01-16 08:26:19

Log by attempts

```
*** Reading local file: /usr/local/airflow/logs/ElastiCloudDataProcessing/data_processing/2020-01-16T08:26:19.919904+00:00/1.log
[2020-01-16 08:26:31,153] {{taskinstance.py:655}} INFO - Dependencies all met for <TaskInstance: ElastiCloudDataProcessing.data_processing 2020-01-16T08:26:19
[2020-01-16 08:26:31,186] {{taskinstance.py:655}} INFO - Dependencies all met for <TaskInstance: ElastiCloudDataProcessing.data_processing 2020-01-16T08:26:19
[2020-01-16 08:26:31,187] {{taskinstance.py:866}} INFO -
--------------------------------------------------------------------------------
[2020-01-16 08:26:31,187] {{taskinstance.py:867}} INFO - Starting attempt 1 of 1
[2020-01-16 08:26:31,187] {{taskinstance.py:868}} INFO -
--------------------------------------------------------------------------------
[2020-01-16 08:26:31,211] {{taskinstance.py:887}} INFO - Executing <Task(BashOperator): data_processing> on 2020-01-16T08:26:19.919904+00:00
[2020-01-16 08:26:31,214] {{standard_task_runner.py:52}} INFO - Started process 13425 to run task
[2020-01-16 08:26:31,287] {{logging_mixin.py:112}} INFO - [2020-01-16 08:26:31,286] {{dagbag.py:403}} INFO - Filling up the DagBag from /usr/local/airflow/dag
[2020-01-16 08:26:31,389] {{logging_mixin.py:112}} INFO - Running %s on host %s <TaskInstance: ElastiCloudDataProcessing.data_processing 2020-01-16T08:26:19.9
[2020-01-16 08:26:31,432] {{bash_operator.py:81}} INFO - Tmp dir root location:
 /tmp
[2020-01-16 08:26:31,432] {{bash_operator.py:91}} INFO - Exporting the following env vars:
AIRFLOW_CTX_DAG_OWNER=airflow
AIRFLOW_CTX_DAG_ID=ElastiCloudDataProcessing
```

PRAMATI

TECHTONIC upshift your technology quotient

CHALLENGES

- Airflow is not Apache NIFI or Apache Spark
  - is not - data routing or data transformation system.
- Airflow Workers
  - requires identical access(network, authentication & authorisation)
  - requires similar hardware capabilities
- Airflow Webserver & Scheduler
  - not scalable (Work-around - supervisor, docker health checks)

PRAMATI

TECH TONIC
upshift your technology quotient

RECOMMENDATIONS

Airflow is a best suited where

- Agility is important
- Portability
- Segregation b/w compute and workflow mgmt
- Scalability on demand
- Pool/Connection management
- Job Analytics
- Logging visibility

Q&A

Open for discussions,..

REFERENCES

- Roles of Data Engineer - https://inlovewithcode.wordpress.com/2019/05/15/roles-of-data-engineer-required-skill/
- System Design - https://inlovewithcode.wordpress.com/system-design/
- Airflow Tutorials - https://airflow-tutorial.readthedocs.io/en/latest/airflow-intro.html
- Airflow Documentation - https://airflow.apache.org/docs/stable/
- Airflow Dockers - https://towardsdatascience.com/getting-started-with-airflow-using-docker-cd8b44dbff98

Thank you

PRAMATI

TECHTONIC
upshift your technology quotient