# From newbie to Data Engineer

Nikita Sharma | Greenhorn Data Science student @ UTS

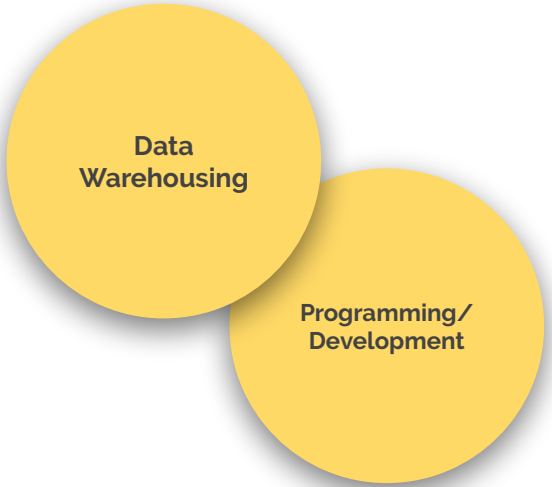
*You want to be a data engineer*
*but don't know where to start.*

* **Newbie content warning !**

Q1

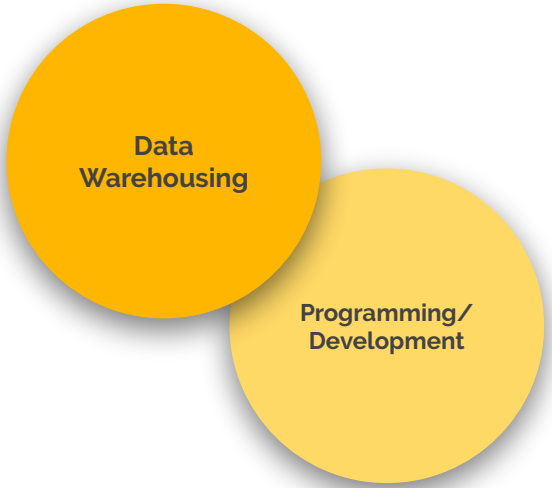
# What makes me a Data Engineer ?

# The union of skills



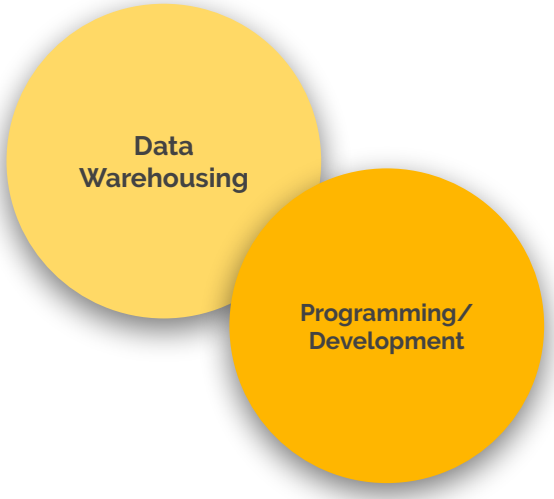
# The union of skills

Data modeling, Relations of datasets, Big Data pipelines, Custom ETLs etc



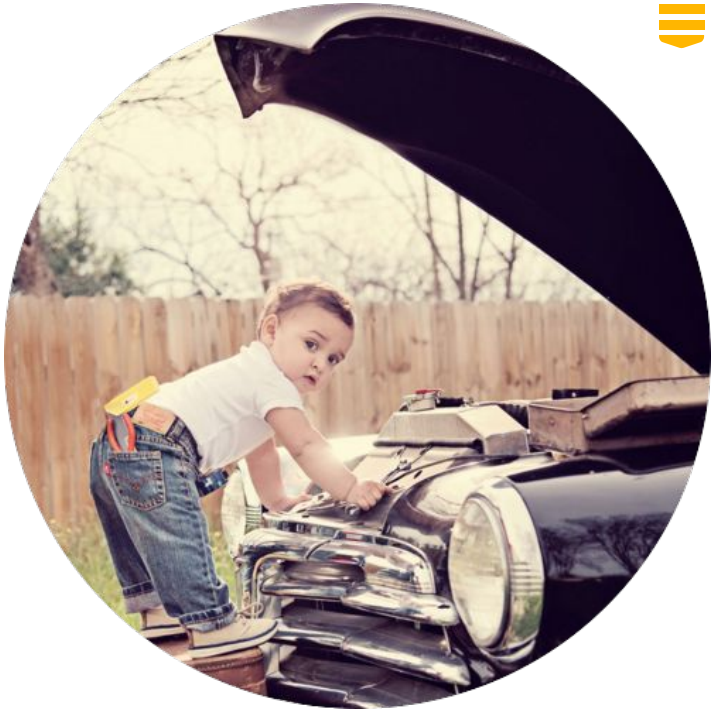
# The union of skills

Data Platforms, Frameworks, Self serving tools etc



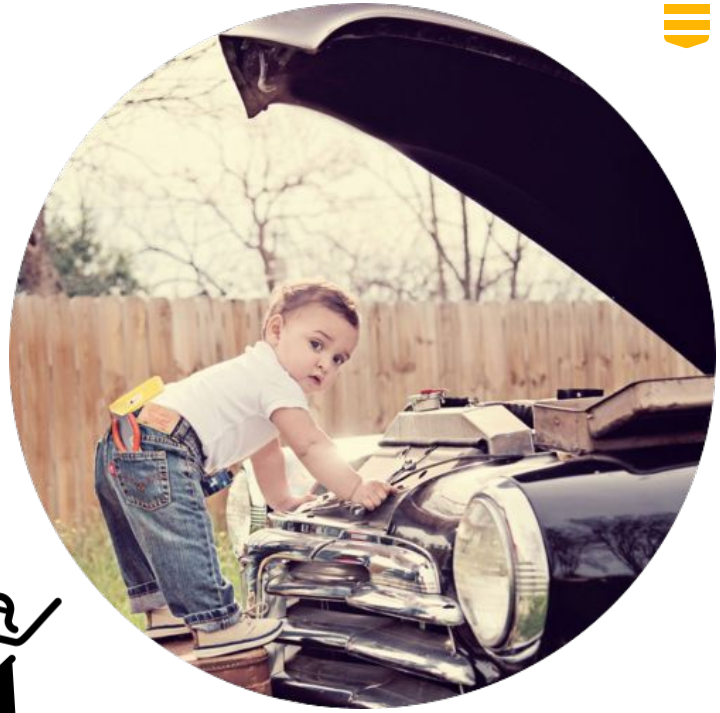
# Most importantly
# the mindset

- Data scale excites you
- You are open to tools (& concepts)
- Want to make data access easier and optimized



# Most importantly
# the mindset

- Data scale excites you
- You are open to tools (& concepts)
- Want to make data access easier and optimized



Untold mission of a Data Engineer

Q2

# ok, Where do I start

# SQL

When in doubt start with the simplest step,
but start.

# Why SQL ?

**Simple**

Easy to get started with SQL. Tons of tutorials online, many other portals to try out SQL problems online.

**Common Language**

SQL is a common business language. Easier to write and share ETL's in SQL.

**Big Data Scale**

All modern Big Data Engines are supporting SQL or SQL-like formats. This skill is certainly not going to get wasted.

# How did I start

**Concepts**

Read SQL web tutorials

**Challenges**

Use HackerRank and other online competitive mediums to develop skills.

**Practice**

Installed MySQL on my local box and worked on some SQL scenarios.

* all the links added to last slide

Q3

# What programming language to start with

# Python

I started with my strongest suit, you can chose the one that's your strongest.

**Popular options:** Scala, Java, Python.

Q4

# How to Big Data ?

# Concepts before tools

**What's happening**

Most important and most difficult step.

- Medium.com
- Netflix / Cloudera/ Hortonworks blogs
- Tons of others

**Optimizations**

- How to organize data
- nicer file formats
- data partitioning
- etc

**Tools**

Commonly used tools.
Tradeoffs in tools.
Awareness of tools.

- Hive
- Spark
- Flink
- Presto
- New one emerging everyday

Q5

# Can I get some data ?

# Data is everywhere

**Public data**

- Kaggle
- Google dataset search
- Crime/Climate open datasets
- Data.gov.au, etc

**Social**

- Twitter tweets
- Facebook api
- Google search api
- Web crawling
- etc

**Generate**

I generated my custom data in batches and saved on S3 in regular intervals

- Apache log format
- Json format

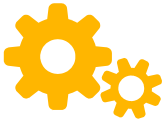
Q6

# What data tools to learn

# Too many tools
# is not necessarily bad

- Optimize on your current skills
- Leverage the tools that support your skills
- Learn about the pitfalls/gotchas on the tools

# My choices



**Hive**

- Supports SQL
- Big Data Scale
- Huge Community
- Has been around for a while (more stackoverflow questions/answers)

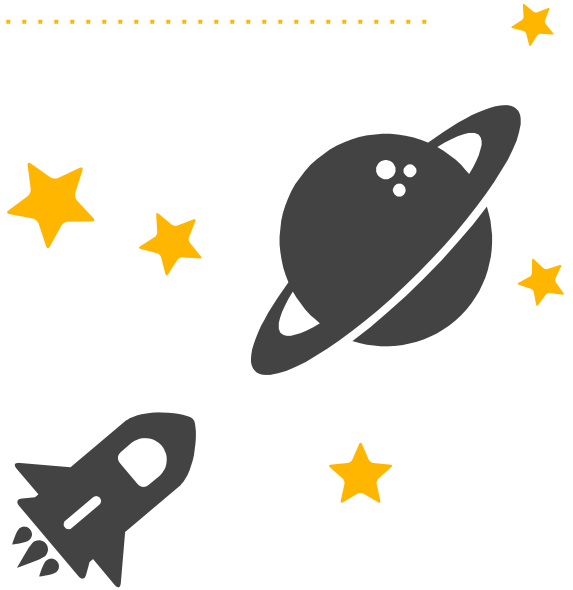
**Spark**

- Supports Hive SQL
- Supports Python
- Big Data Scale
- Growing Community
- Has been around
- Faster than Hive for lot of use cases

Q7

# What cloud infrastructure to work on

# Absolutely anything



- All these cloud offerings are very mature
- AWS / Azure / Google Cloud / Databricks
- You will find your way once you start
- I chose Amazon AWS

Q8

# Em, What is a data lake

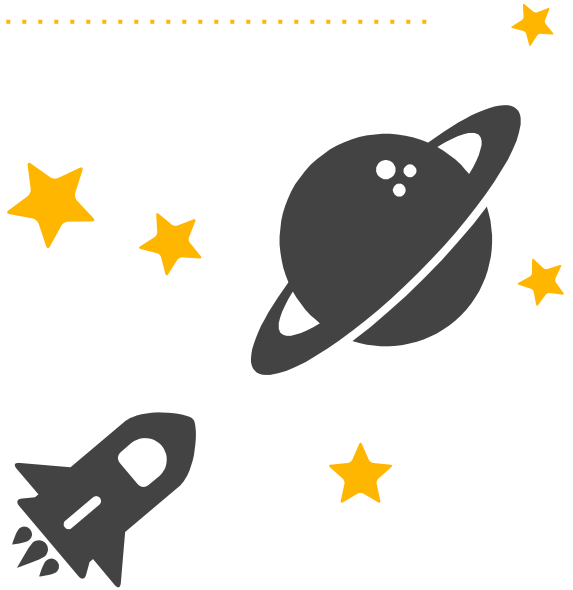
# Data lake

- A fancy word for centralized data
- A home for all your Multi-format, Structured and Unstructured data
- All downstream systems get data from here and do further data processing/ organizing
- **Popular options:** HDFS / S3 / Google FS / Databricks FS

Q9

# Can I haz real world use case

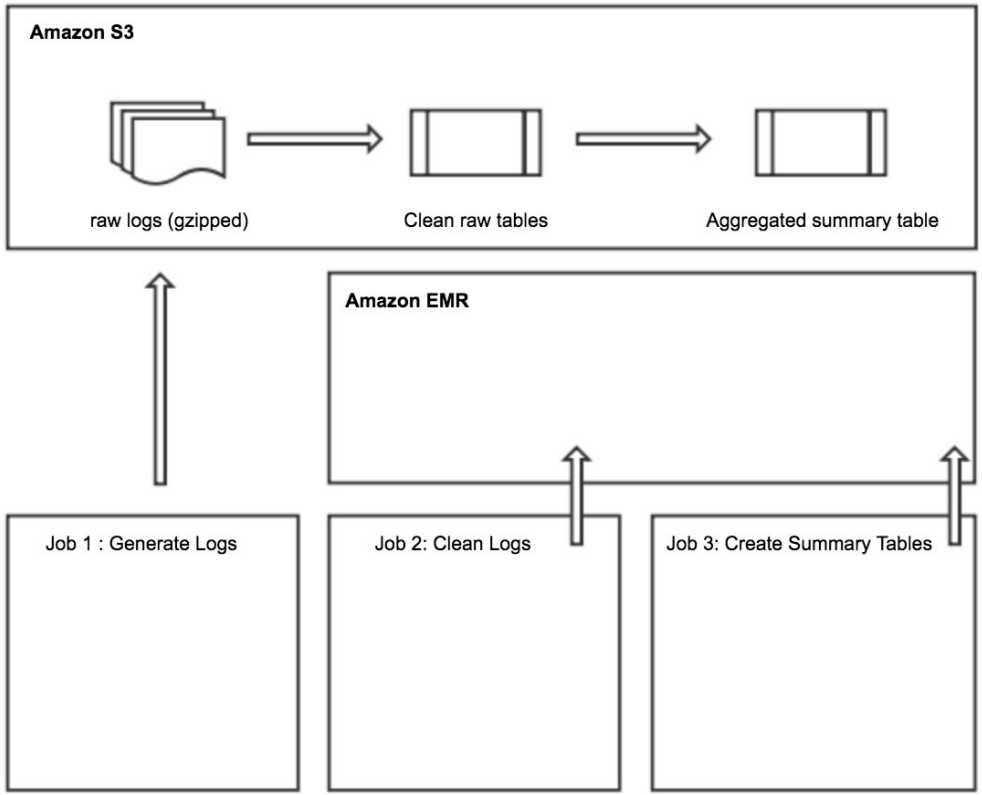
# Create an ETL pipeline



...to replicate a common industry use case.

Doesn't have to be really fancy in the 1st attempt.

- Generate zipped log data manually in hourly batches
- Create Hive SQL ETL for cleaning and aggregations
- SSH to EMR cluster, and run Hive Script
- Save all data in our Data Lake
- Run all jobs daily, over single day partition. Update all tables with new data for that day
- All job triggers are manual at this point

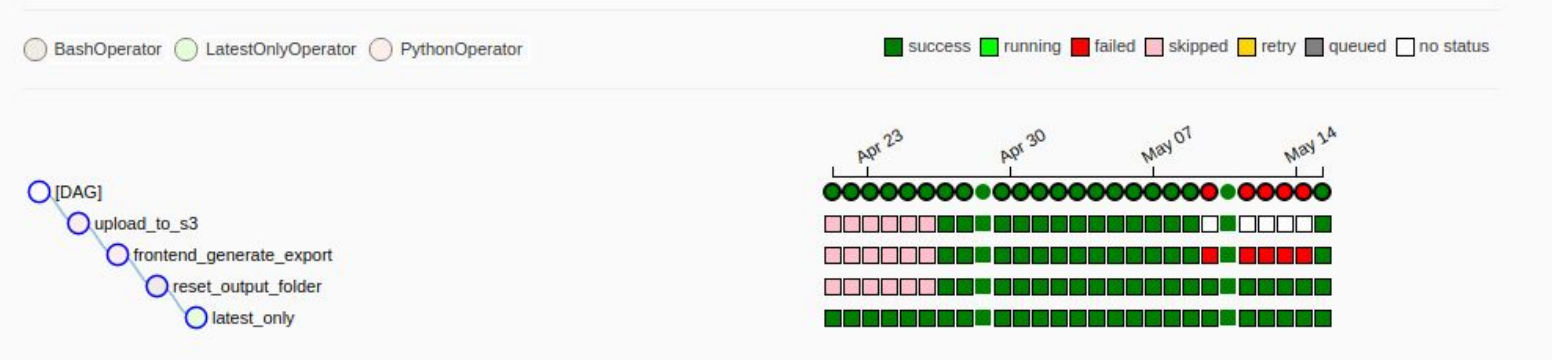


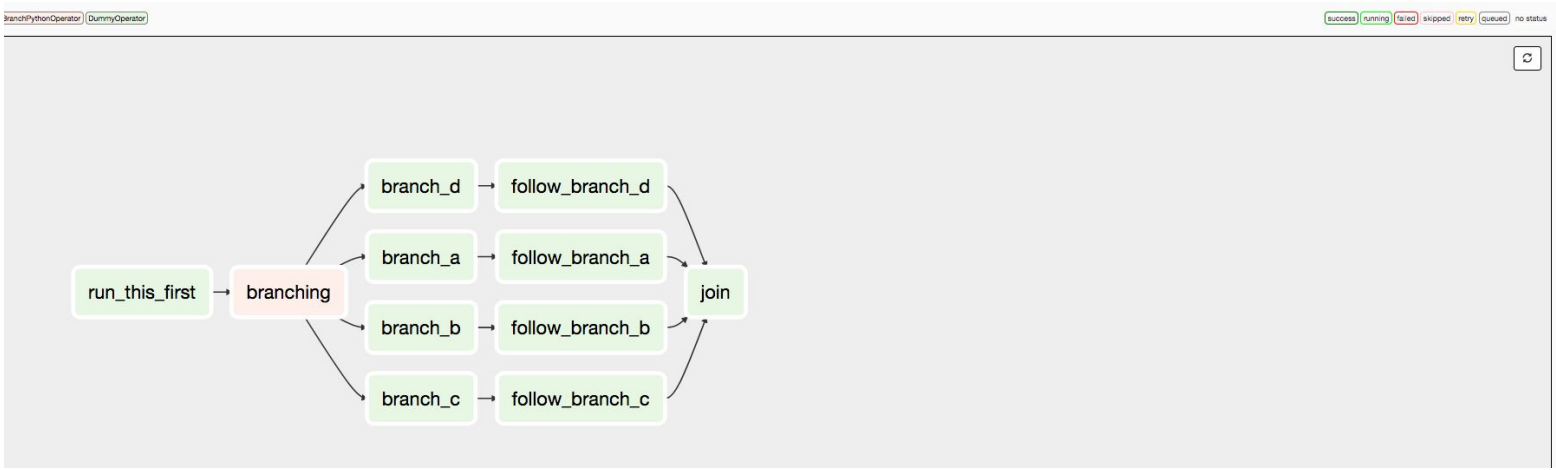
Q10

# Productionizing my casual project

# Airflow scheduler

- Fits naturally with our python skills
- Provides great visibility on job progress
- Has easy to use failure-retry and re-trigger capabilities
- Open source, Awesome community and bunch of pre-created plugins - Email, MySQL, Postgres, Slack, Hive etc

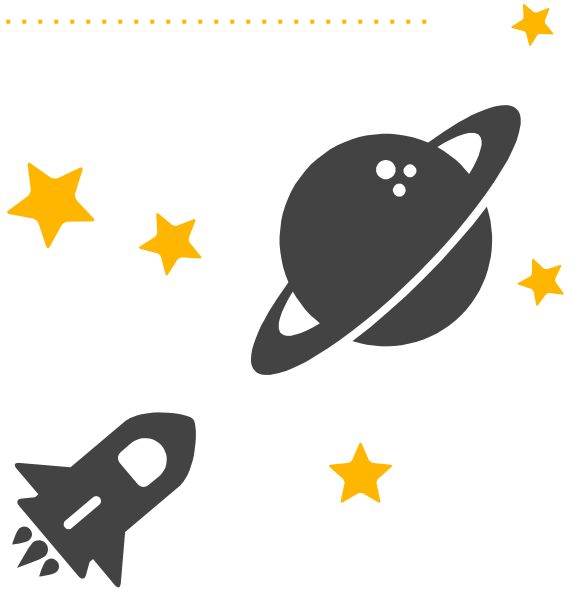

BashOperator
LatestOnlyOperator
PythonOperator
success
running
failed
skipped
retry
queued
no status
[DAG]
upload_to_s3
frontend_generate_export
reset_output_folder
latest_only
Apr 23
Apr 30
May 07
May 14


run_this_first
branching
branch_d
branch_a
branch_b
branch_c
follow_branch_d
follow_branch_a
follow_branch_b
follow_branch_c
join

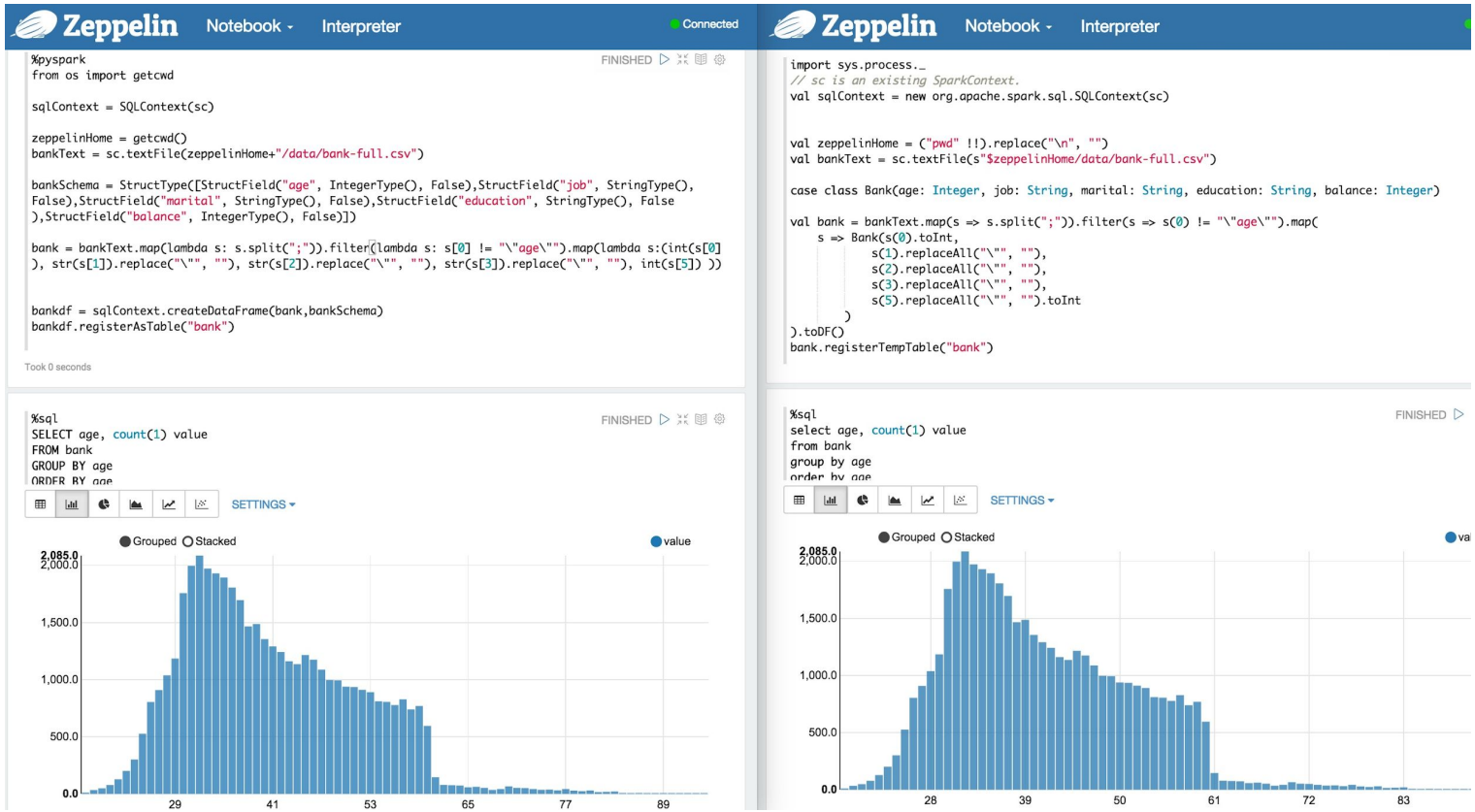
Q11

# Make data accessible to end users

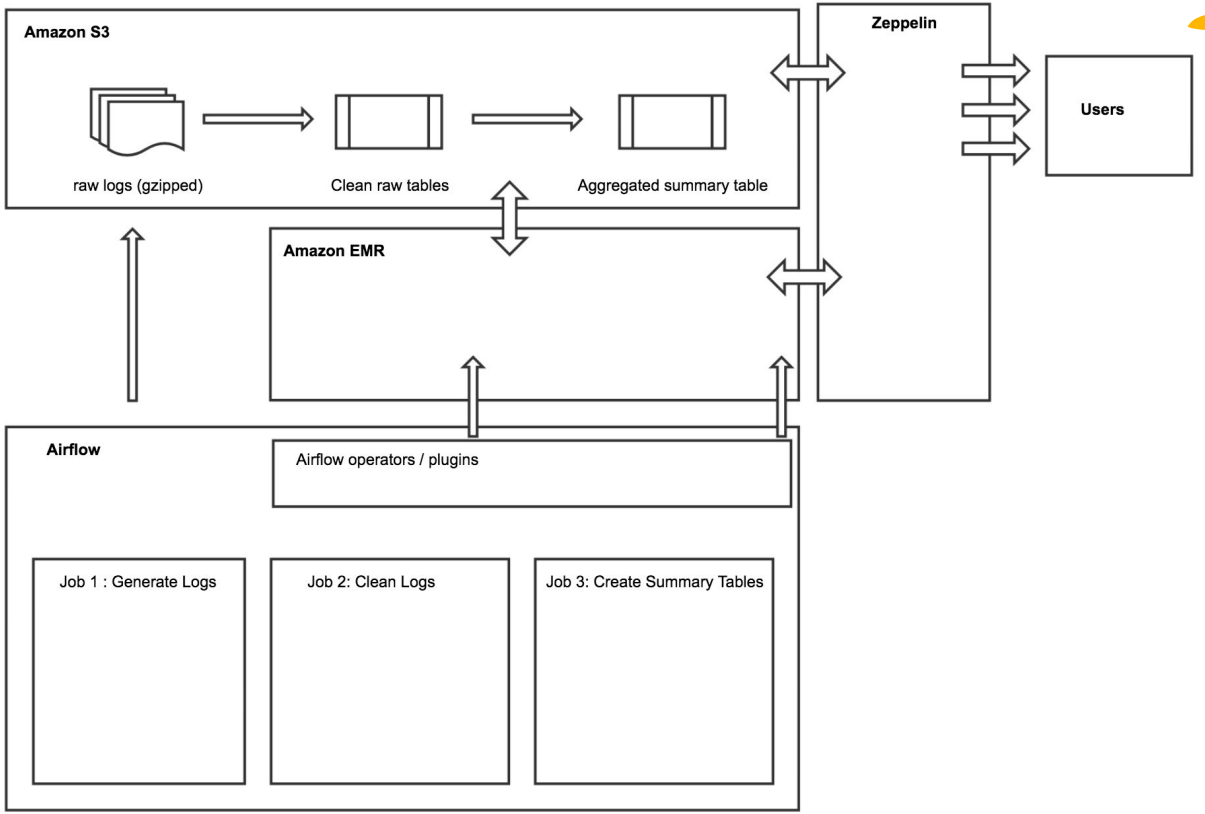
# Apache Zeppelin



- Works on top of Apache Spark
- Supports SQL, Python and scala
- Fits well with our current skills
- Easy to use web interface for beginners
- Easy to install on EMR cluster


Zeppelin
Notebook
Interpreter
Connected
%pyspark
from os import getcwd
sqlContext = SQLContext(sc)
zeppelinHome = getcwd()
bankText = sc.textFile(zeppelinHome+"/data/bank-full.csv")
bankSchema = StructType([StructField("age", IntegerType(), False),StructField("job", StringType(), False),StructField("marital", StringType(), False),StructField("education", StringType(), False),StructField("balance", IntegerType(), False)])
bank = bankText.map(lambda s: s.split(";")).filter(lambda s: s[0] != "\"age\"").map(lambda s:(int(s[0]), str(s[1]).replace("\"", ""), str(s[2]).replace("\"", ""), str(s[3]).replace("\"", ""), int(s[5])))
bankdf = sqlContext.createDataFrame(bank,bankSchema)
bankdf.registerAsTable("bank")
FINISHED
Took 0 seconds
%sql
SELECT age, count(1) value
FROM bank
GROUP BY age
ORDER BY age
FINISHED
SETTINGS
Grouped
Stacked
value
2,085.0
2,000.0
1,500.0
1,000.0
500.0
0.0
29
41
53
65
77
89
Zeppelin
Notebook
Interpreter
import sys.process._
// sc is an existing SparkContext.
val sqlContext = new org.apache.spark.sql.SQLContext(sc)
val zeppelinHome = ("pwd" !!).replace("\n", "")
val bankText = sc.textFile(s"$zeppelinHome/data/bank-full.csv")
case class Bank(age: Integer, job: String, marital: String, education: String, balance: Integer)
val bank = bankText.map(s => s.split(";")).filter(s => s(0) != "\"age\"").map(
s => Bank(s(0).toInt,
s(1).replaceAll("\"", ""),
s(2).replaceAll("\"", ""),
s(3).replaceAll("\"", ""),
s(5).replaceAll("\"", "").toInt
)
).toDF()
bank.registerTempTable("bank")
%sql
select age, count(1) value
from bank
group by age
order by age
FINISHED
SETTINGS
Grouped
Stacked
va
2,085.0
2,000.0
1,500.0
1,000.0
500.0
0.0
28
39
50
61
72
83

# Voila,
# Updated project



Q12

Last question I promise,

# Tips on staying updated ?

# Go to meetups

Talk to the practitioners about their data challenges

# Read blogs

Learn about new tools and techniques adapted by professionals

# Write blogs

Throw out your ideas on some specific data challenge and get feedback from readers/experts

# Alright, lets summarize our plan

# Just start ! Take the leap !

Learn SQL & Python


Learn Hive and Spark


Get started with any cloud service provider


Create an ETL Project


Top-up with tools

Visit meetups and share your learnings

# All the Links (as promised)

- Real world Big Data processing project
- MySQL Starter Guide and SQL pathway guide
- Generate synthetic log data
- Create Hive ETL Project
- Productionize ETL project with Apache Airflow
- Create EMR Cluster with Spark/Hive
- Access Hive tables via Apache Zeppelin
- All my Medium blogs in one place
- All the code on Github: here and here

# Thanks!

Those were my questions,
Time for yours !