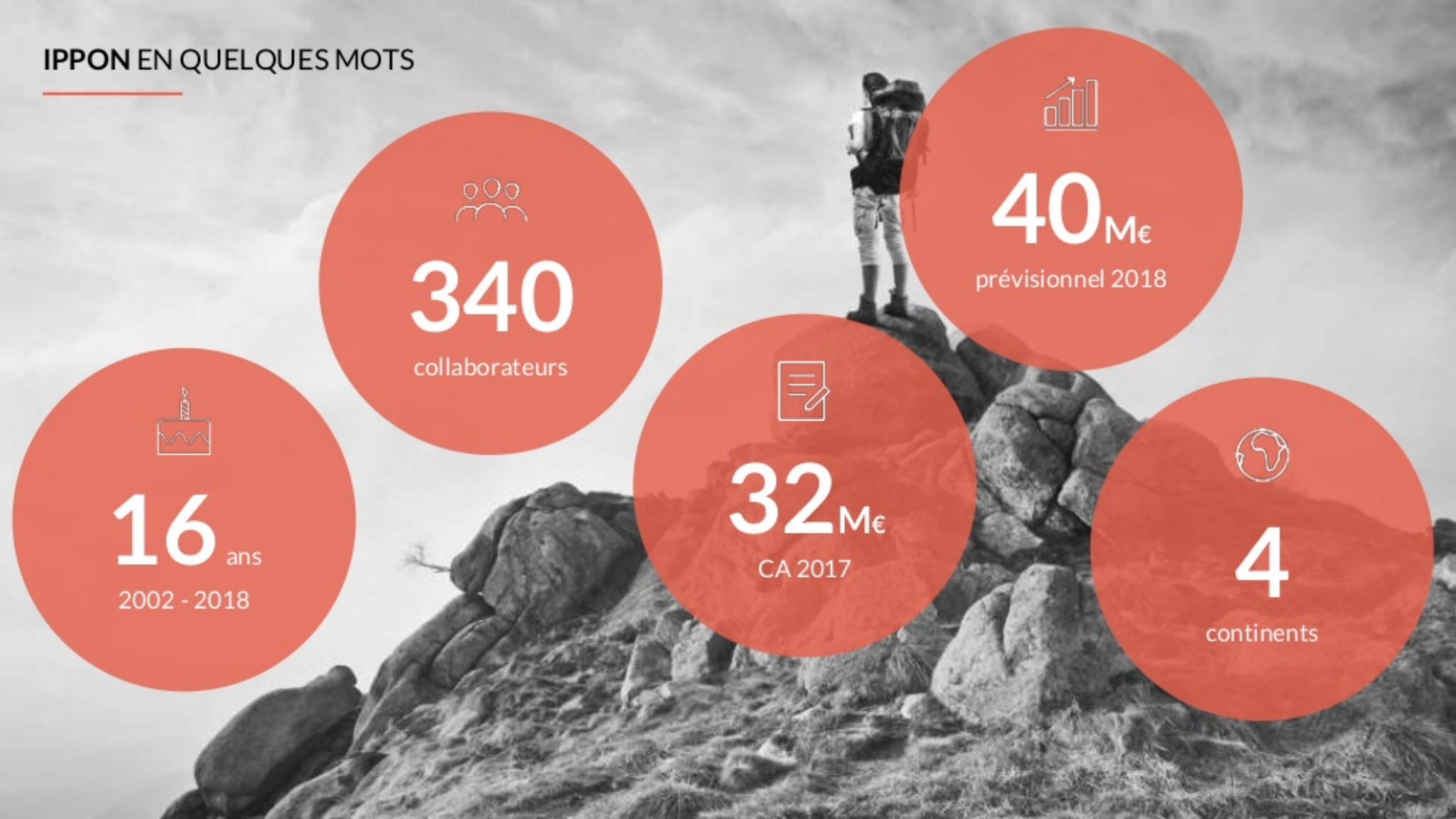
Spark, de l'étincelle à la production

Lucien Fregosi
Nicolas Martin

Introduction / Contextualisation

IPPON EN QUELQUES MOTS



16 ans
2002 - 2018

340
collaborateurs

32M€
CA 2017

40M€
prévisionnel 2018

4
continents


Ippon
Toulouse

Temps fort Data

VIVEZ UN MOIS DE JUIN 100% DATA

05/06 : Paris (REX DataLake AWS avec La Mutuelle Générale)
12/06 : Paris (La data Experience et design de services)
14/06 : Toulouse (Industrialiser Spark)
20/06 : Paris (REX l'IA au service de la validation de documents administratifs)
27/06 : Lyon (Big Data in the Cloud)

Nicolas Martin

Data engineer #Data #Cloud

nmartin@ippon.fr

Lucien Fregosi

Data engineer #Data #Cloud

https://twitter.com/lulufrego

lfregosi@ippon.fr

Mobile: +33 6 45 85 83 97

Certified Solutions Architect - Associate

Certified Data Engineer

Ippon Technologies 2018

Our Manifesto

We want our Jobs go on for a long time

We want to be efficient during development

We want to evolve our Jobs to add new features or refactor

We want to monitor and debug our Jobs

We want to work with clean and qualitative data

We want to put in production often in an automatic way

We want to have good performance

We want to track data transformation

Typical Data Science Work

jupyter convnets Last Checkpoint: 08/29/2017 (unsaved changes)

File Edit View Insert Cell Kernel Help

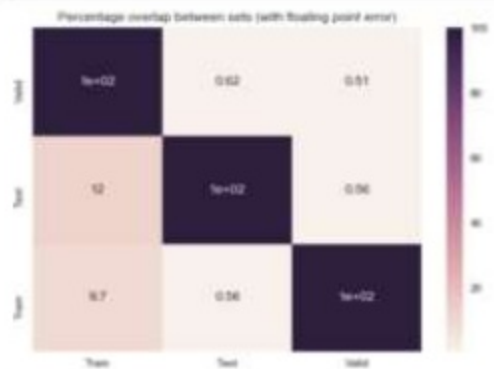
Code CellToolbar

Looking for Collisions in Training, Test and Validation Sets

In [47]:
```
train_hashes = set(hash(i.tostring()) for i in train_dataset)
test_hashes = set(hash(i.tostring()) for i in test_dataset)
valid_hashes = set(hash(i.tostring()) for i in valid_dataset)
```

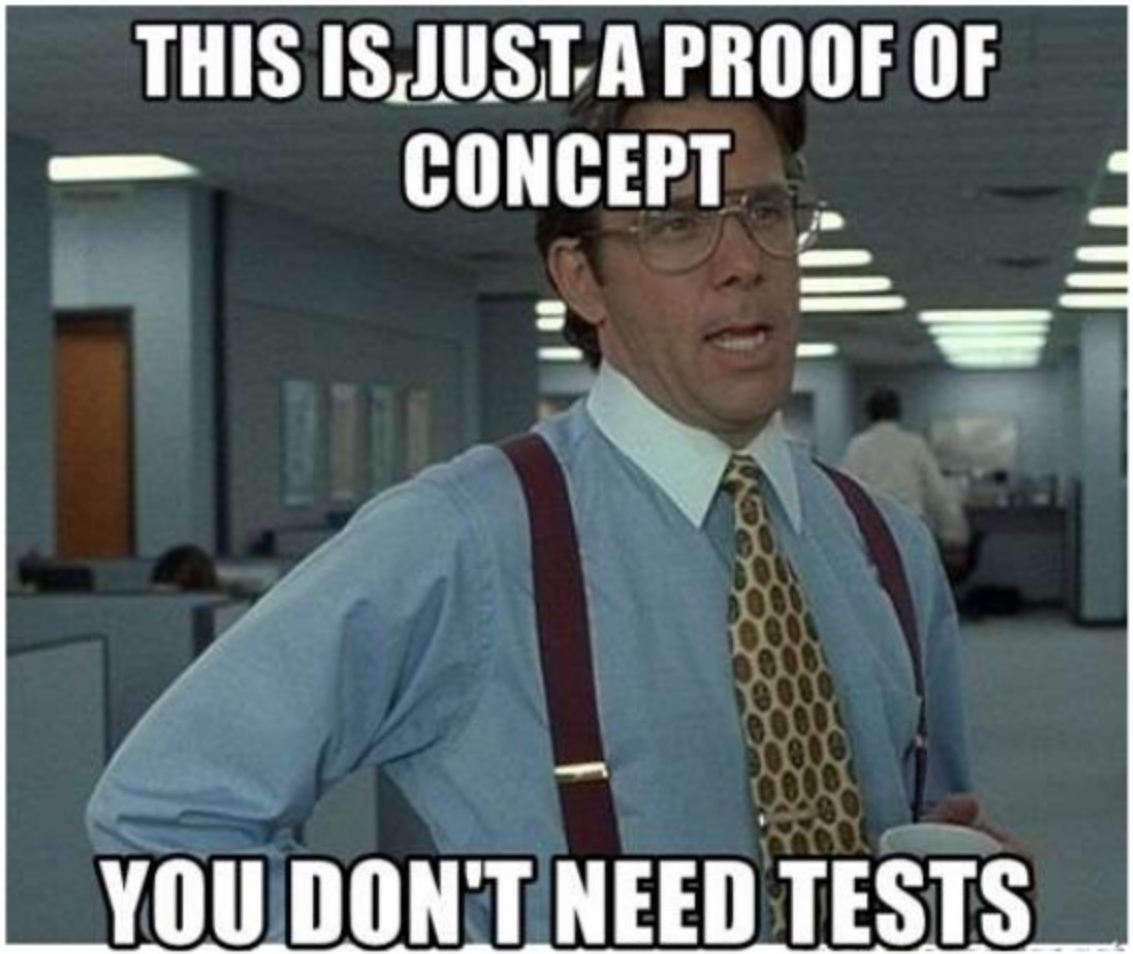
In [66]:
```
with Timer('28x 28 Pixel Photo Set Sizes {} {} {}'.format(
    len(train_dataset), len(test_dataset), len(valid_dataset))):
    train_hashes = set(hash(i.tostring()) for i in train_dataset)
    test_hashes = set(hash(i.tostring()) for i in test_dataset)
    valid_hashes = set(hash(i.tostring()) for i in valid_dataset)

    result = np.ndarray(shape=(3, 3))
    hashes = [train_hashes, test_hashes, valid_hashes]

    for i, first_hashes in enumerate(hashes):
```

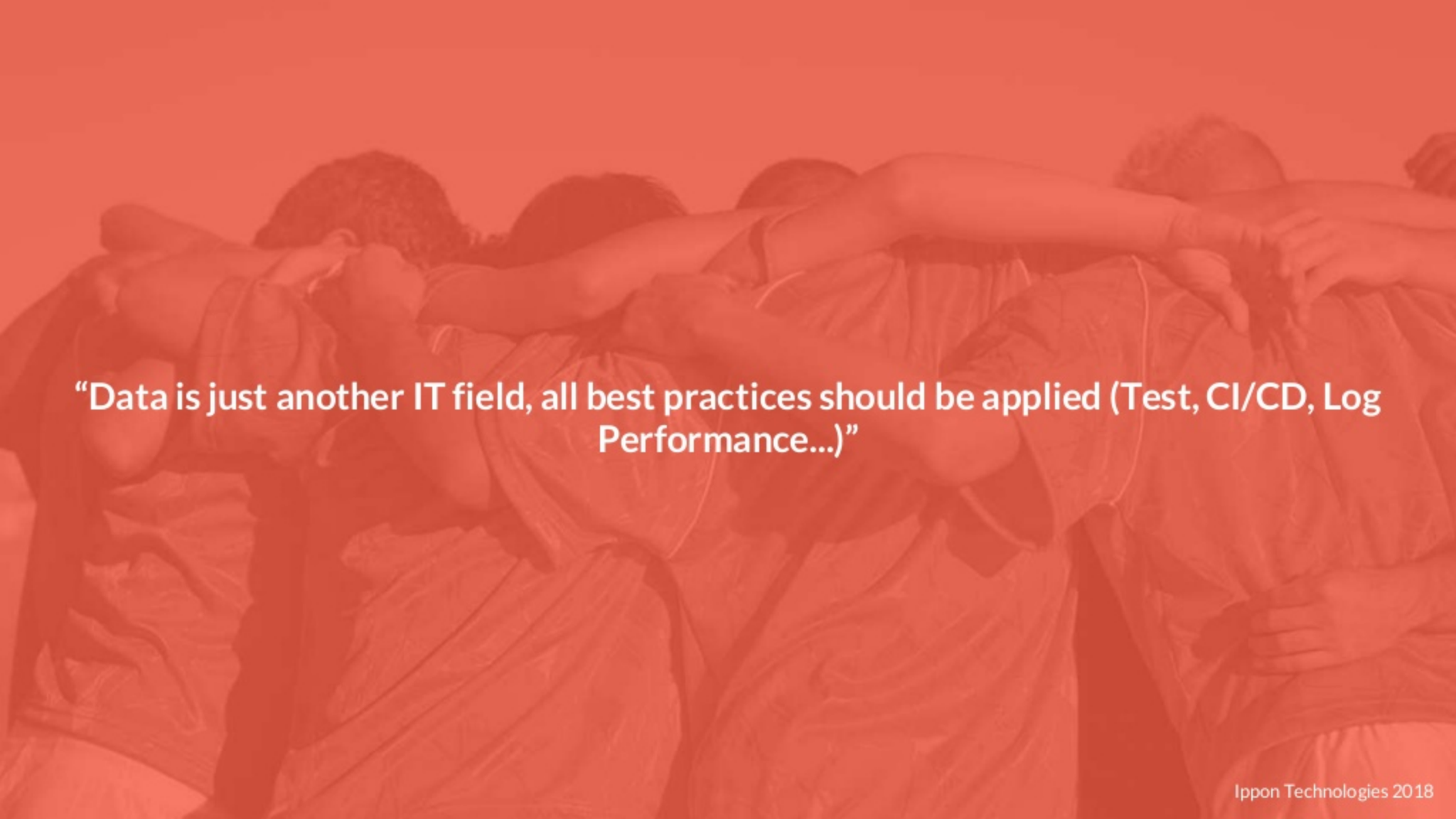
```
ax = sns.heatmap(result, annot=True)
plt.title('Percentage overlap between sets (with floating point error)')
ax.set_yticklabels(['Train', 'Test', 'Valid'])
ax.set_xticklabels(['Train', 'Test', 'Valid'])
plt.show();
```




```
28x 28 Pixel Photo Set Sizes 200000 10000 10000 taken 0.89 secs
```

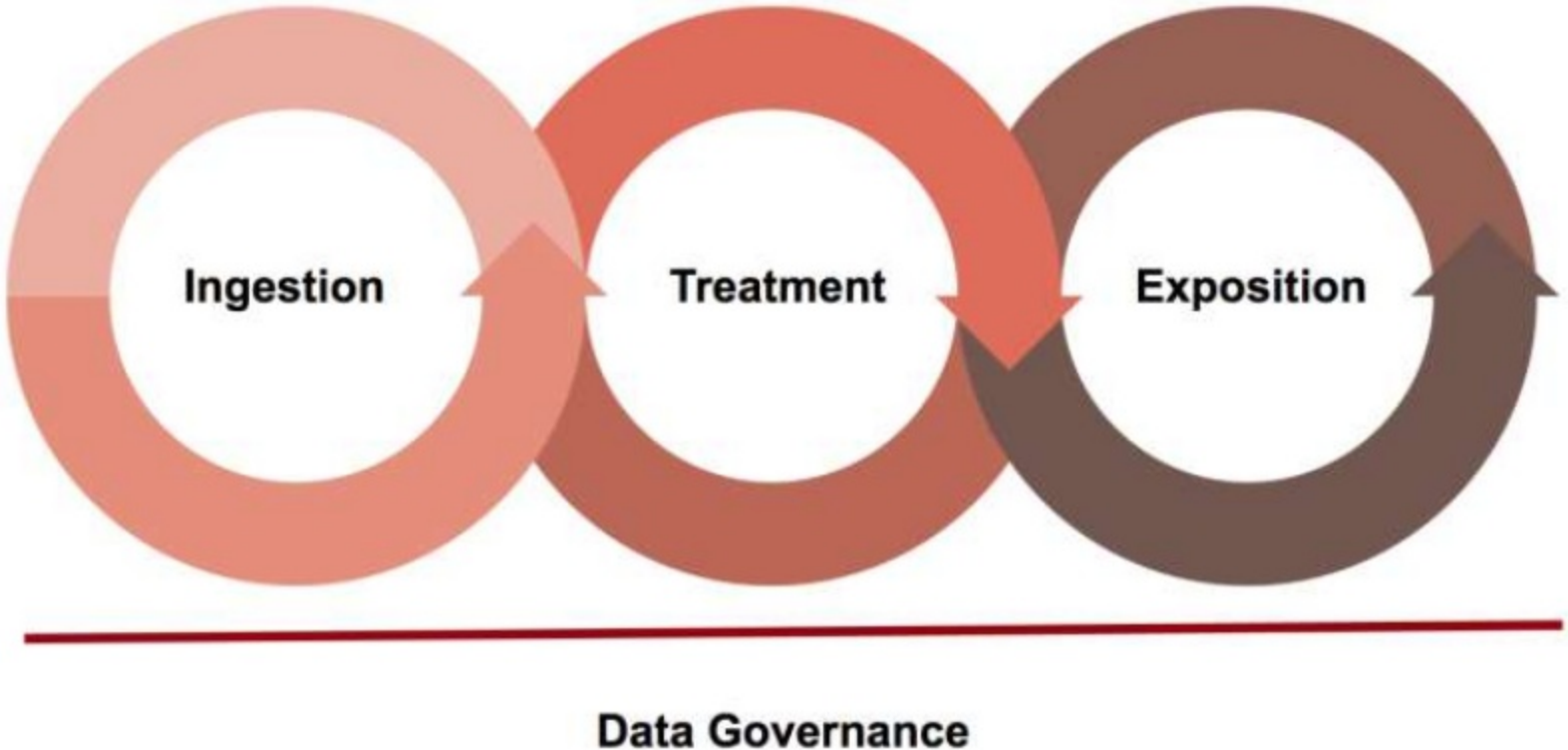
POC is everywhere in Big Data

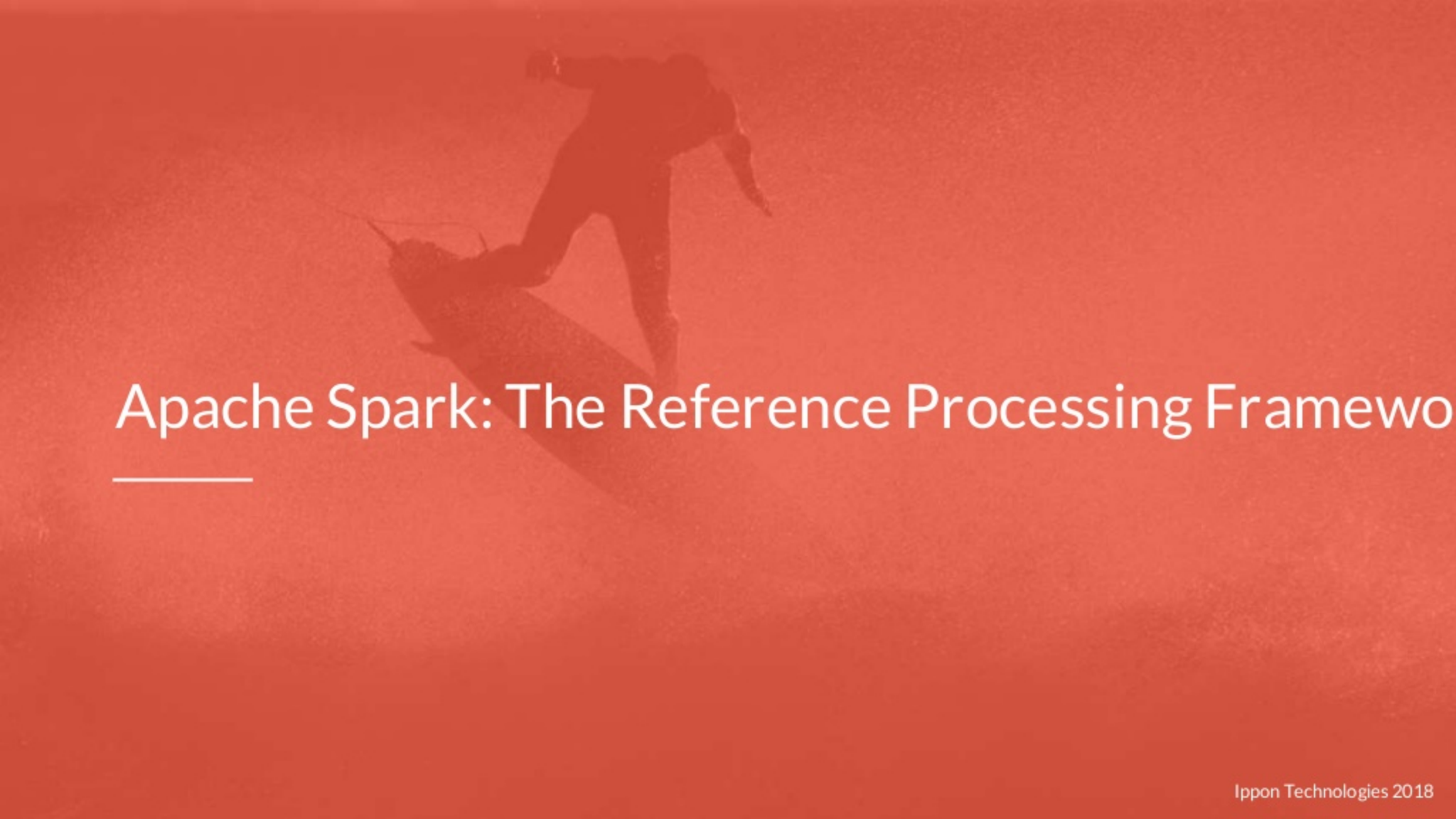



“Data is just another IT field, all best practices should be applied (Test, CI/CD, Log Performance...)”

Data engineer to the rescue!!

Data Journey

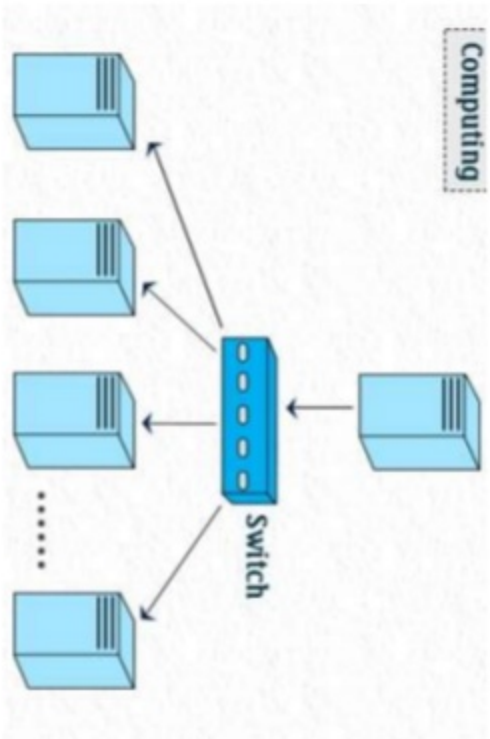


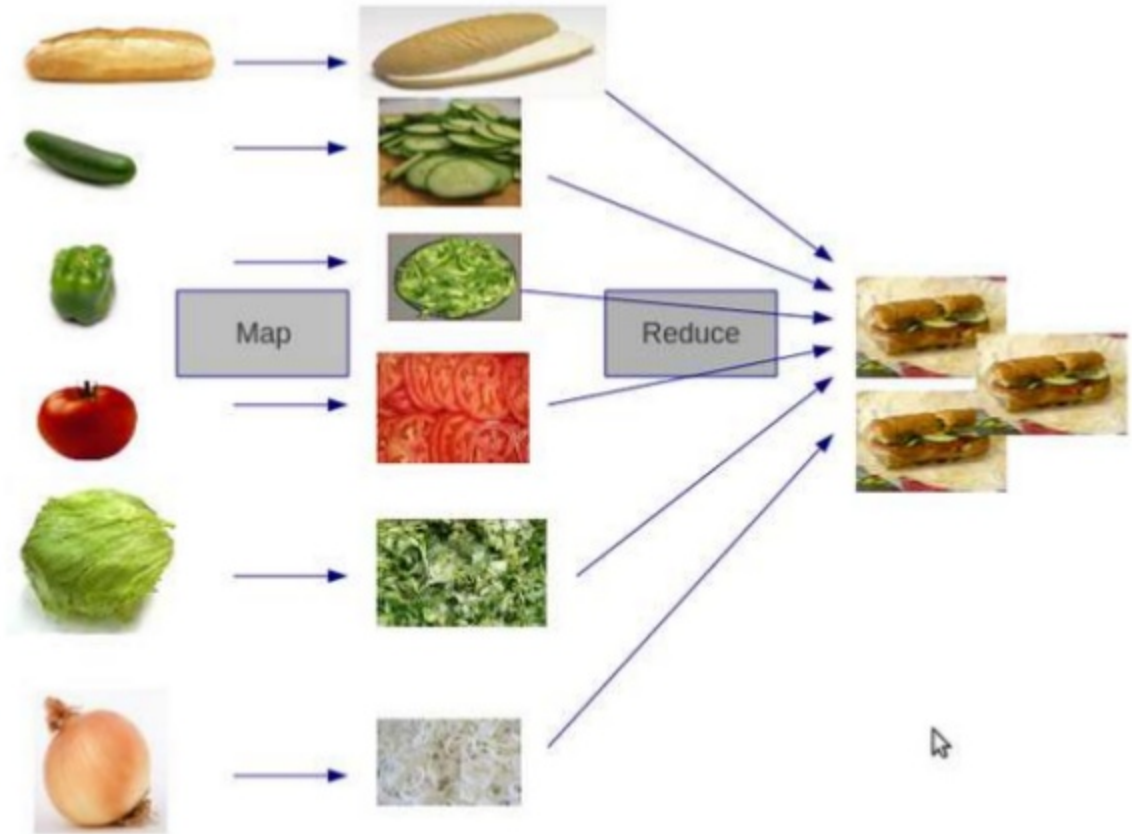


Apache Spark: The Reference Processing Framewo

Ippon Technologies 2018

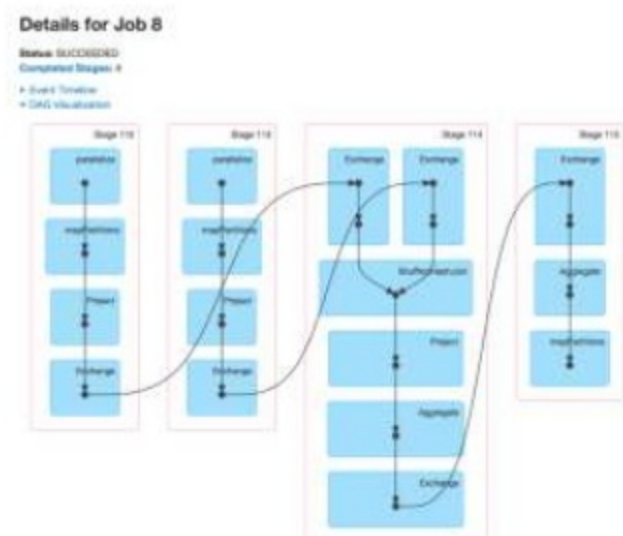
Spark - Distributed computing Framework



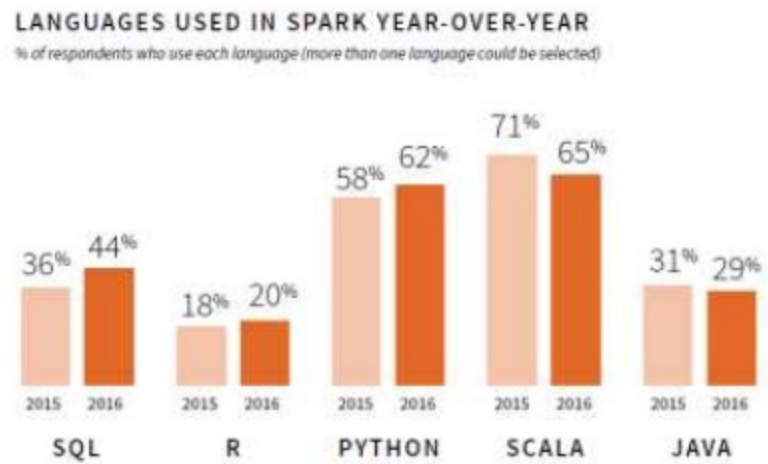


Spark - Key Concepts

Transformations *(lazy)*	Actions
select	show
distinct	count
groupBy	collect
sum	save
orderBy	
filter	
limit	



Spark - Most used languages



Spark - Cluster Manager

kubernetes

Ingestion

Data cleaning

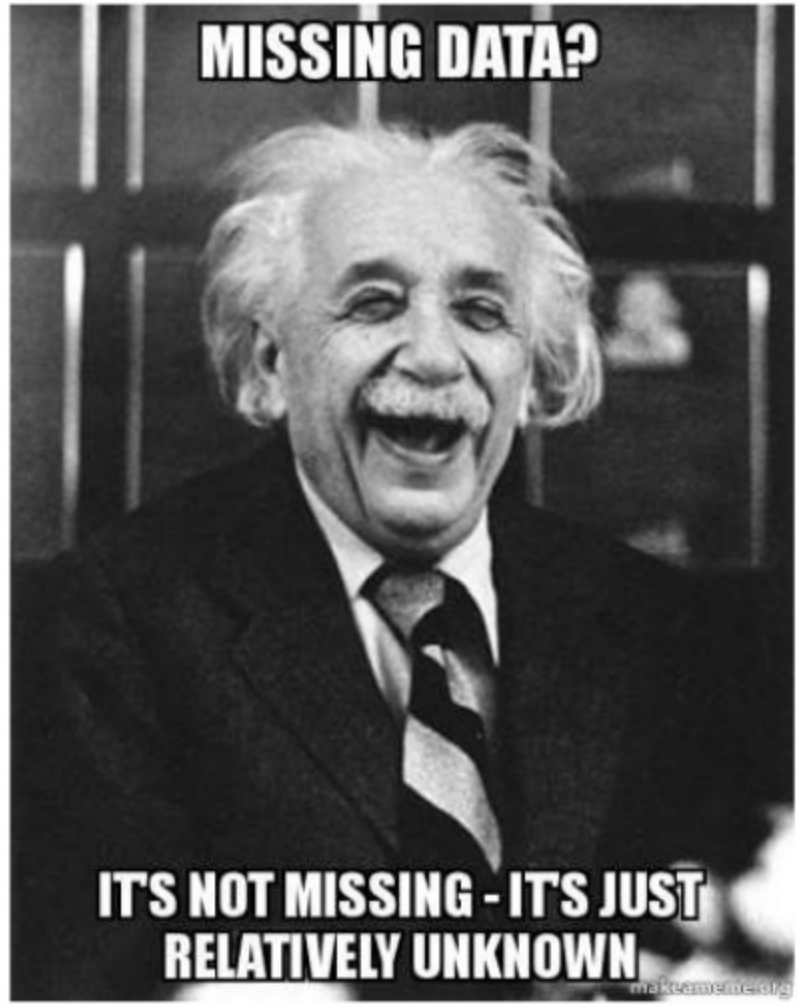
- “Never trust the input“
- Schema validation
- Define a key for deduplication
- Handle missing values
- Data wrangling solutions:

Spark

TRIFACTA

Data checks

- Quality check
  - Provide KPI before and after cleansing
  - Monitor KPI evolution
- Check for null value number
- Check for number of rows
- Depends on your data, check for max/min average..
- Gives you informations for data governance



Processing

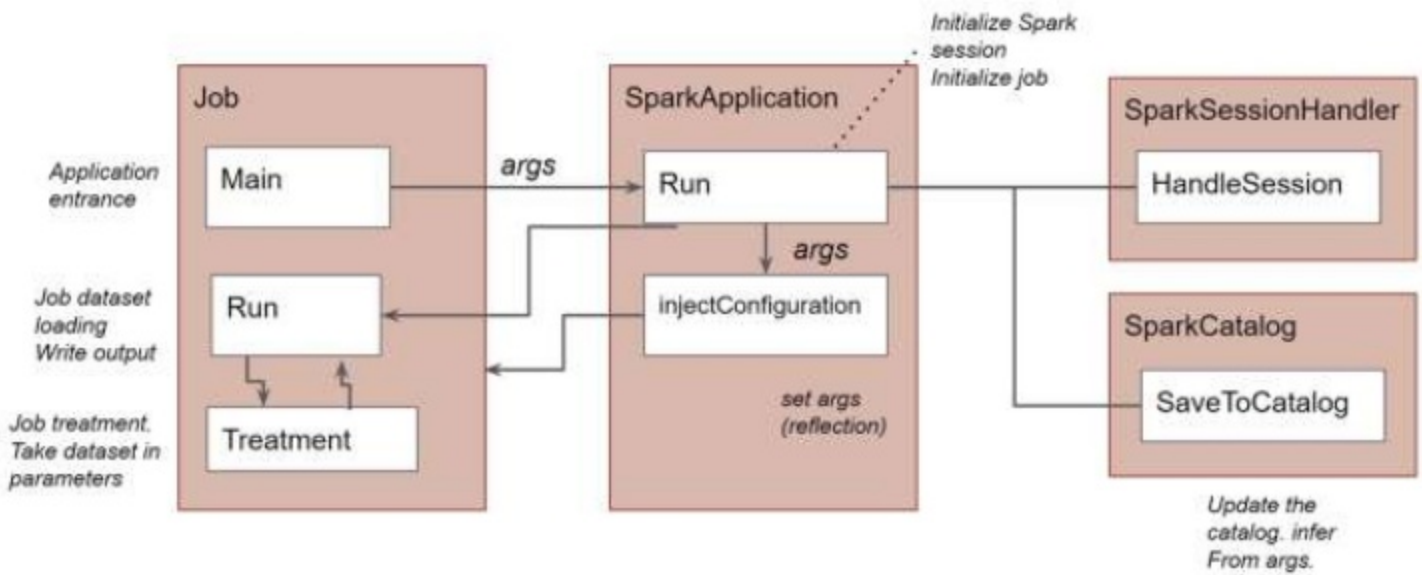
Development environment

- Use a project a professional project structure ***not a script***
  - https://github.com/dbast/spark-scala-template
  - https://github.com/Alexioannides/pyspark-example-project
- Use a dependency manager

- Use a version controller and do code review

Development environment

- Give developers immediate environment
- Hide configuration complexity
- Harmonize development
- Handle session



Testing Spark Code

- How to unit test Spark jobs?
  - Unit test for Scala/Python Function

ScalaTest

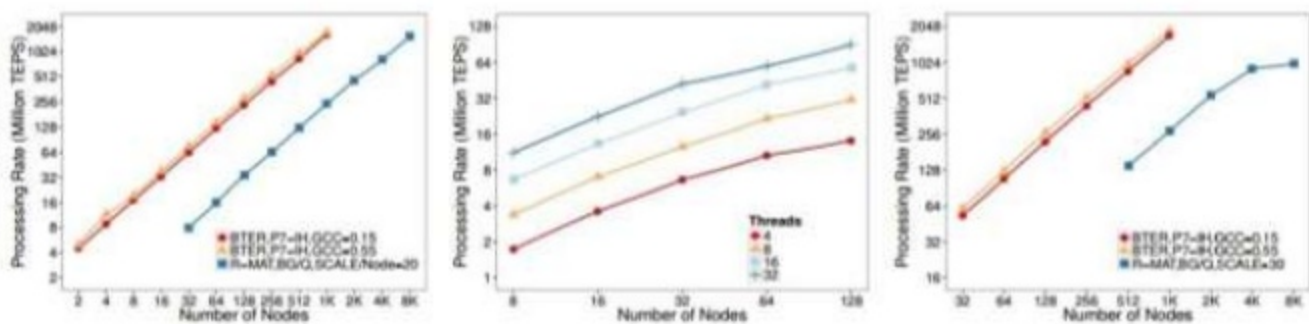
simply productive

- How to do integration tests for Spark jobs?
  - Work with sample data
  - Spark Testing Base https://github.com/holdenk/spark-testing-base

- How do I validate my job efficiency?
  - Check for counters with metrics (bytes read, time, etc..)
  - Spark Validator https://github.com/holdenk/spark-validator

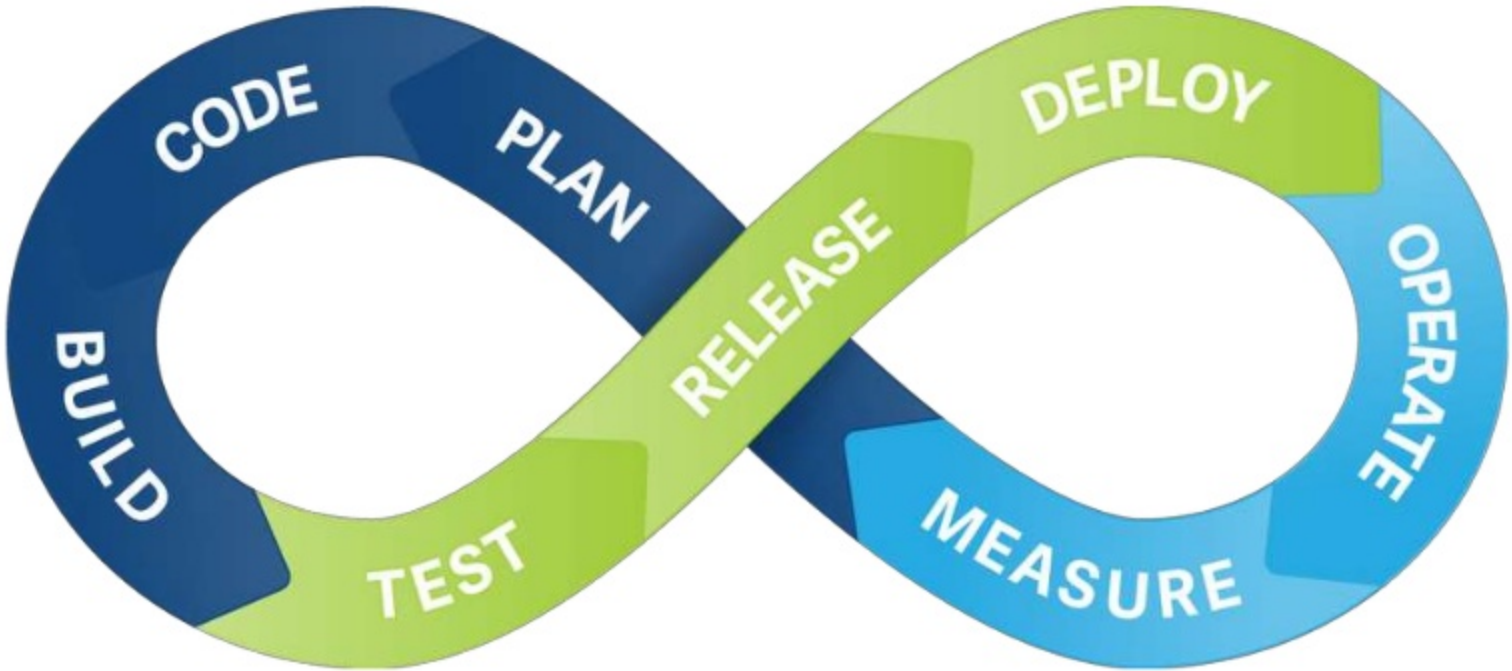
Testing Spark Code

- How perf test Spark jobs?
- Benchmark
  - Set a baseline performance
  - Launch this test every changes made (it has to be short)
- For horizontality scalability test:



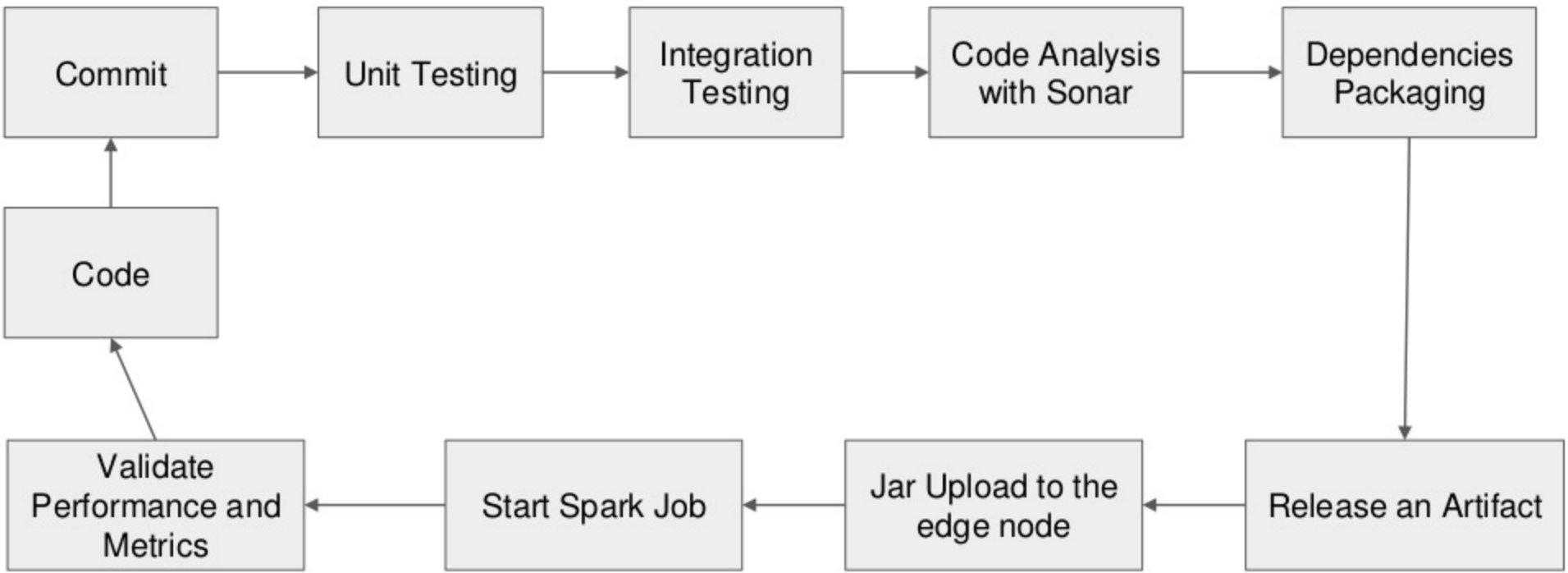
CI / CD

- What is Continuous Integration?



CI / CD

How can we implement Continuous Integration in Spark?



Job configuration

Depends on your Cluster Topology !!

Job Configuration Example

- Class: Main class
- Master: Cluster Manager (Yarn)
- Deploy mode: Client
- Driver-memory: 4go
- Num-executors: 10
- Executor-core: 4
- Executor-memory: 4go
- Garbage Collector: G1GC
- Serializer: Kryo

We can adjust with number of executors and increase/decrease core and memory

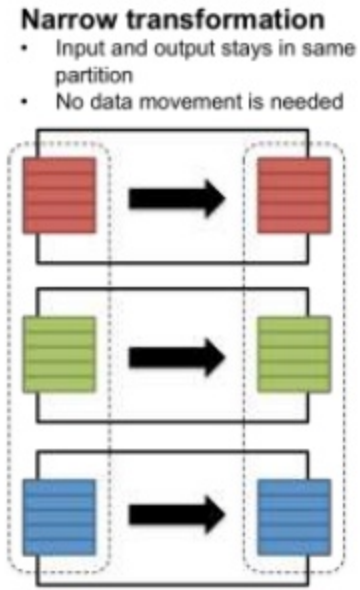
Allocation could be dynamic (for development mainly)

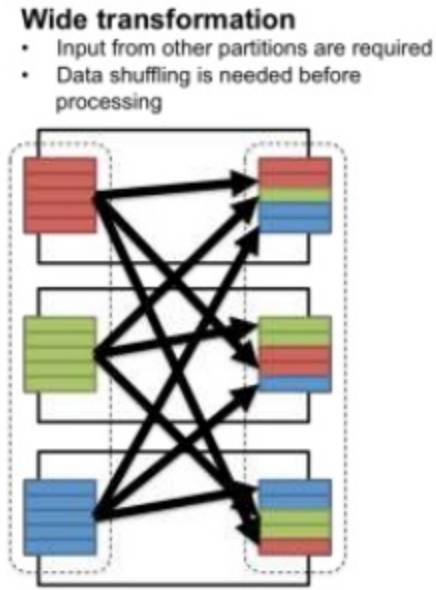
Data knowledge and partitioning

- To do have good performance, you have to know your data
- Know how your dataset is constituted
- Partition should have more or less the same size

Performances

- Shuffle
  - During “byKey” operations
  - During joins
  - Due to repartitionning





- Skew
  - Too big partitions
  - Can be seen when task are longer than others
  - Easily findable in Spark UI

Performances: Cache & Broadcast

- Cache
  - Prevents recalculation of a dataframe in different actions
  - Always chach dataframe if it will be used different times
  - You can cache in memory, on disc or both

- Broadcast
  - Allow diffusing a variable on all executors in read-only
  - Map-side join with lookup table

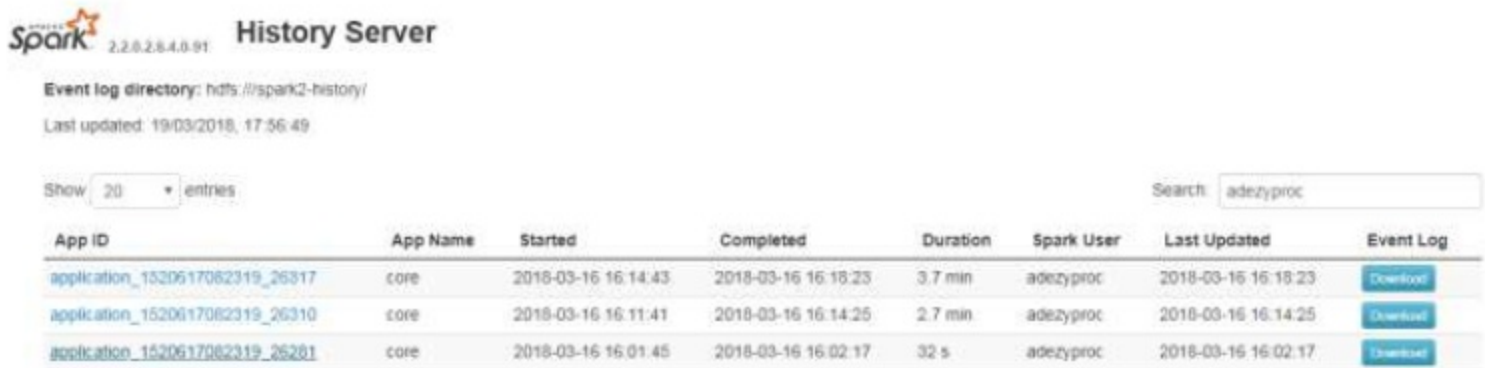
Performances: Files Format & Compression Algorithm

Dataset	Size on Amazon S3	Query Run time	Data Scanned	Cost
Data stored as CSV files	1 TB	236 seconds	1.15 TB	$5.75
Data stored in Apache Parquet format*	**130 GB**	**6.78 seconds**	**2.51 GB**	**$0.01**
Savings / Speedup	**87% less with Parquet**	**34x faster**	**99% less data scanned**	**99.7% savings**

	CSV	ORC		Parquet	
		ZLIB	SNAPPY	GZIP	SNAPPY
data size	11.5 GB	1.1 GB	1.6 GB	1.2 GB	1.7 GB
conversion time from CSV	NA	9 m 10 s	9 m 15 s	10 m 0 s	9 m 23 s
Files	21	42	42	42	42
Time for delayed flights	5 m 57 s	49 s	47 s	1 m 2 s	49 s
Time for fetching a single record	255 s	70 s	62 s	130 s	111 s

Logs The simple way

- Spark History Server -> Applicatif

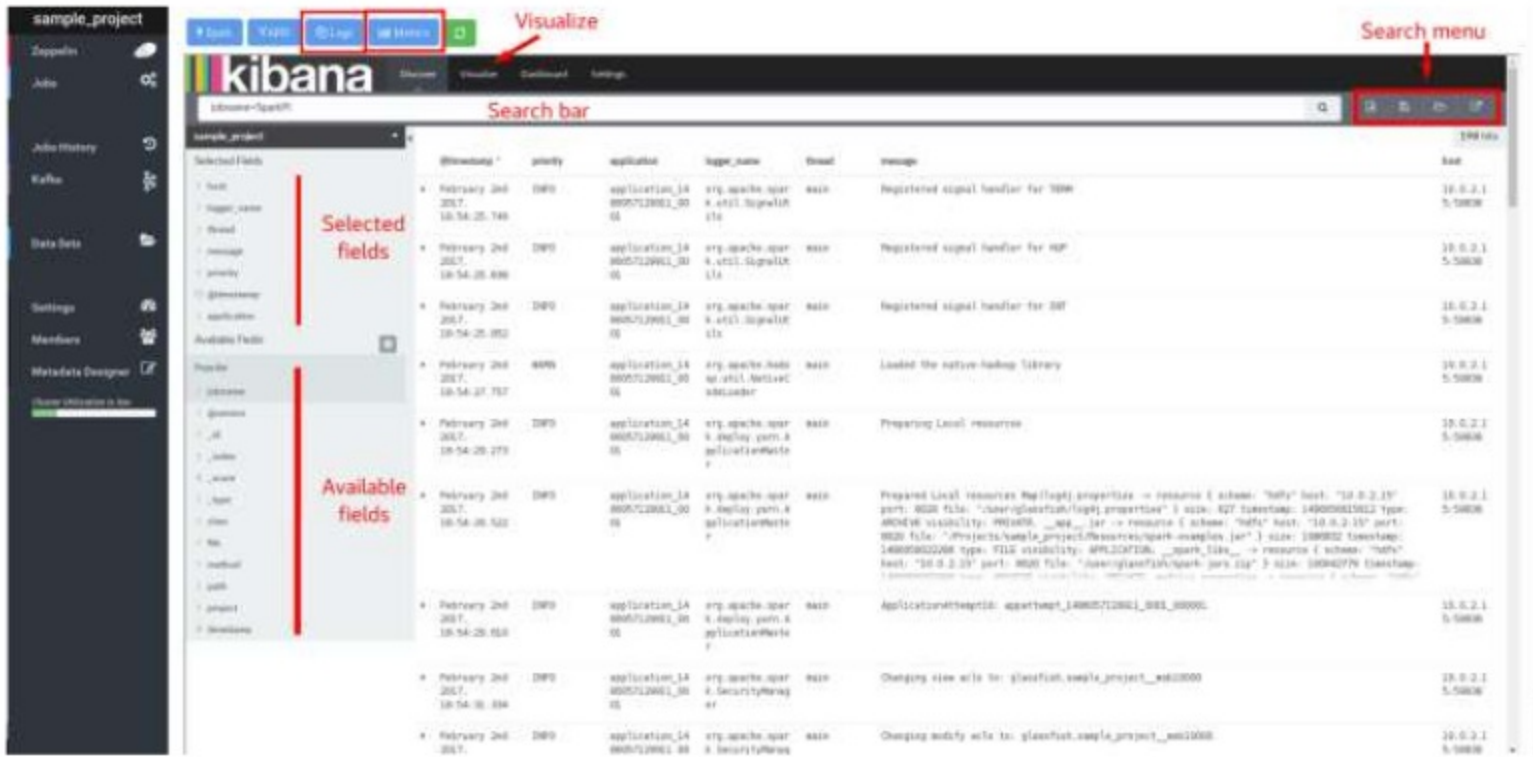


- Log Yarn -> Ressources & Metrics

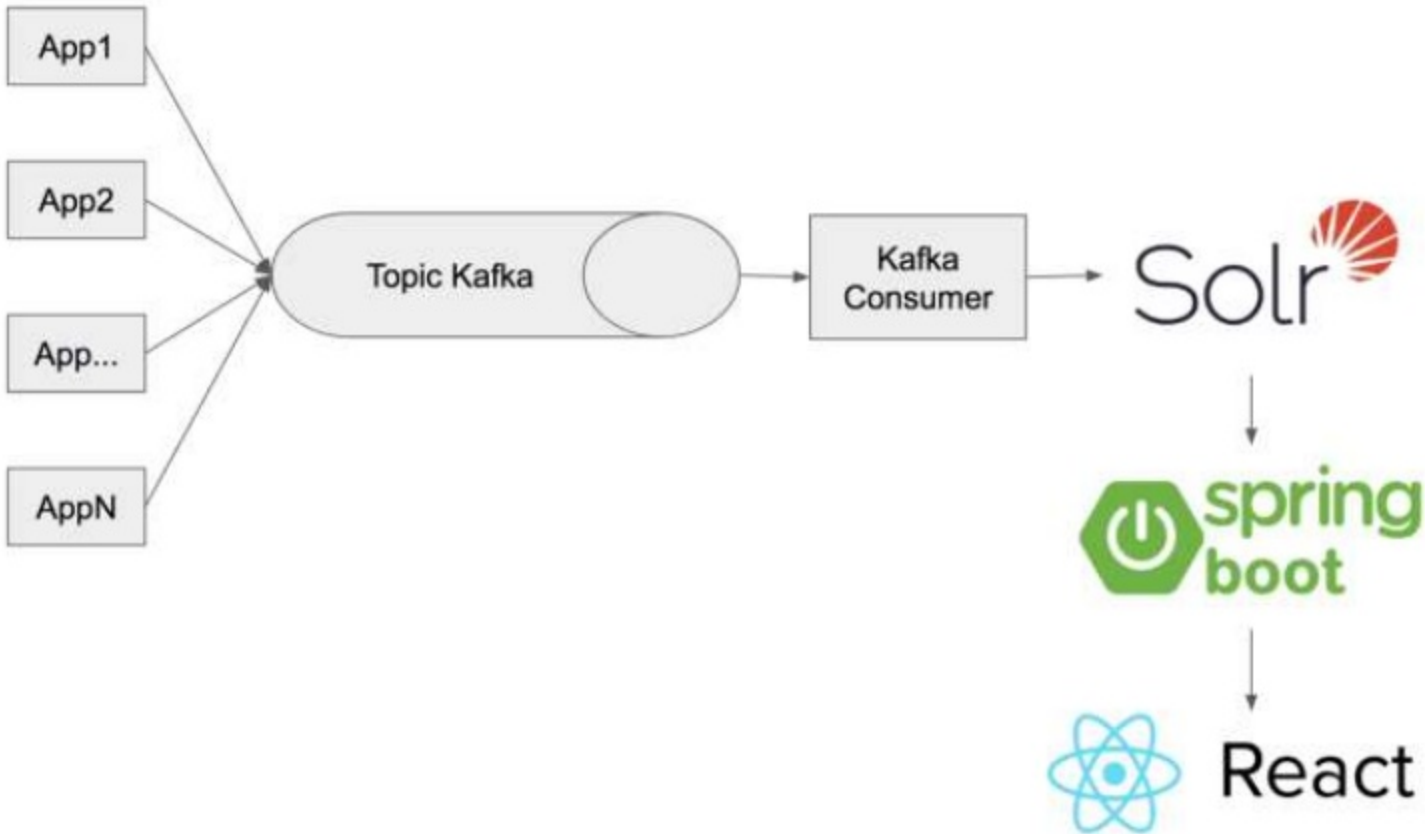
Logs Beyond the UI: ELK or Splunk

As Spark is a distributed computing framework, how to gather logs and use it

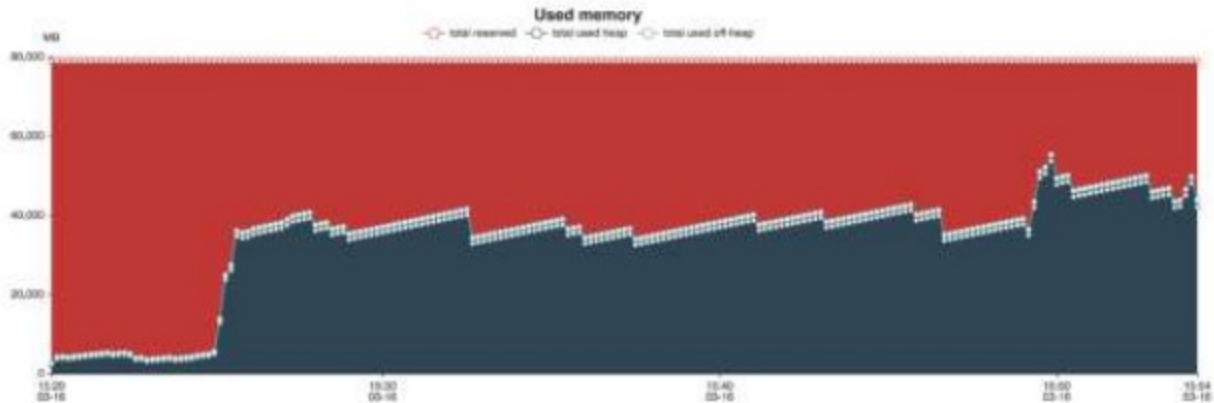
Apache LOG4J™



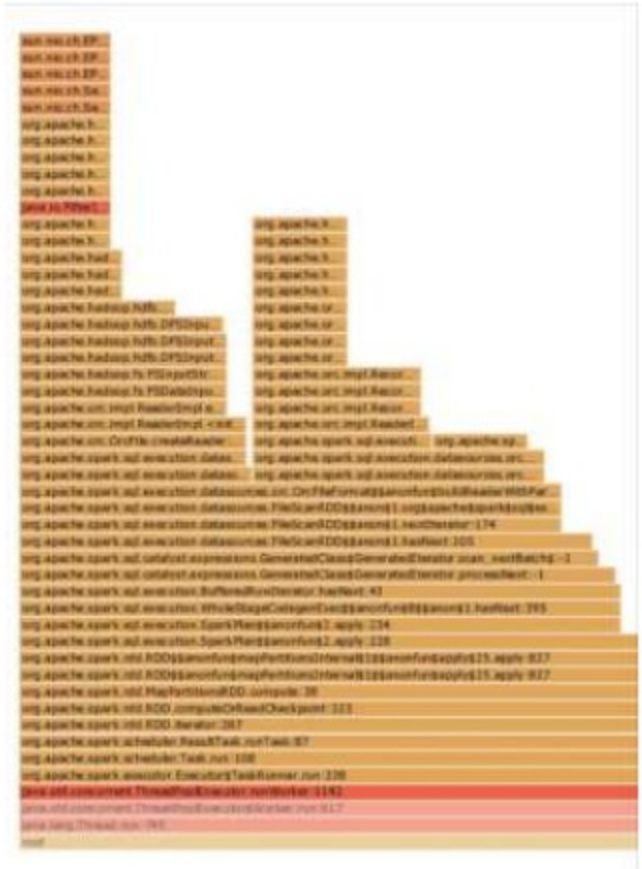
Logs Beyond the UI: Rex Custom Solution



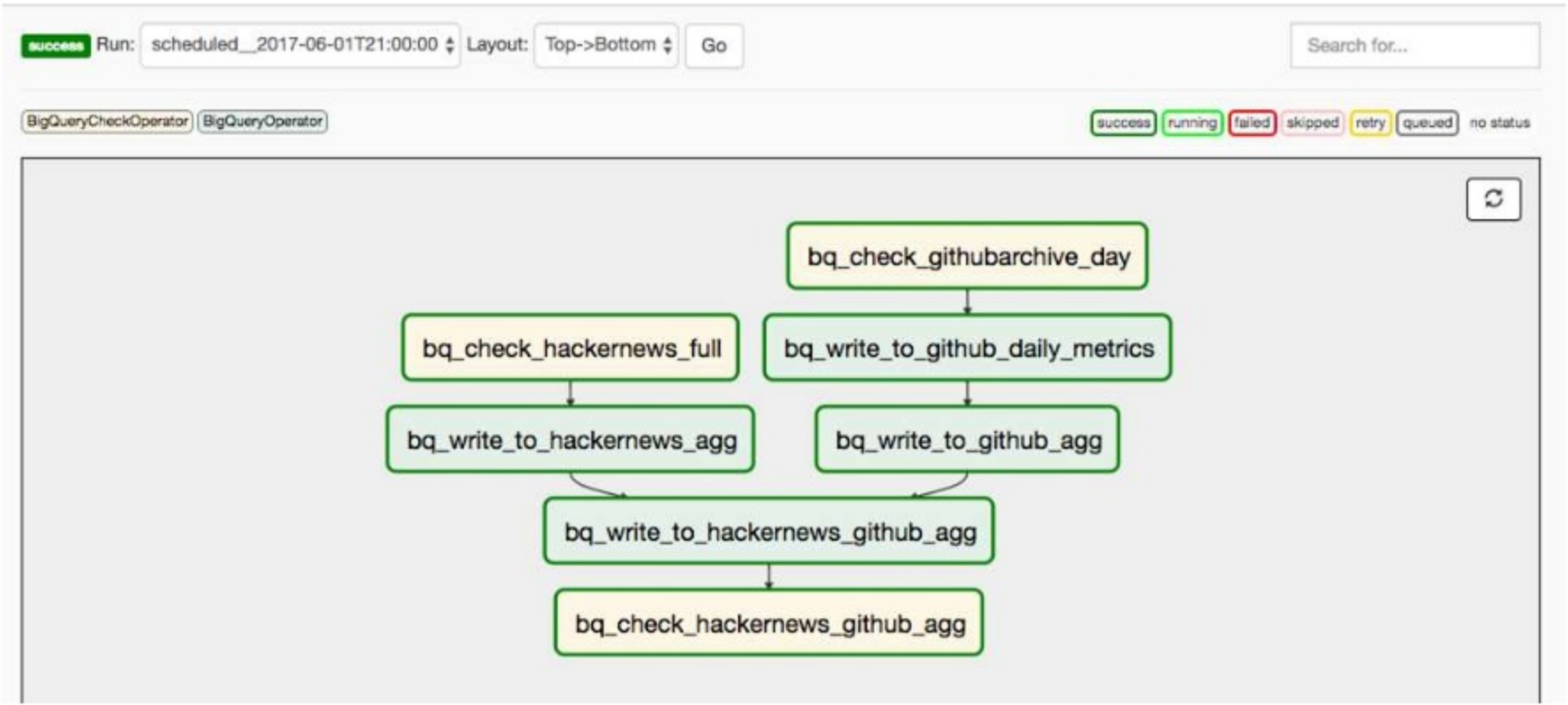
Logs Beyond the UI: Monitor Resources and use flamegraphs



https://github.com/criteo/babar

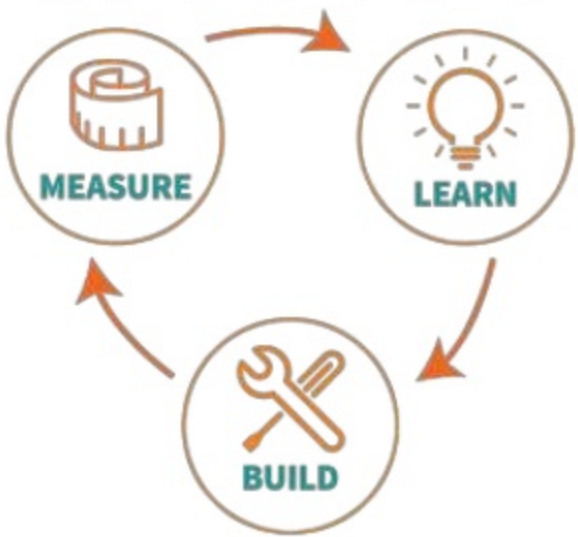


Schedule Spark Jobs & Workflow management



Machine learning

- Can we industrialize Machine learning pipline with spark? (feature creation, coding normalization, algorithms…)
- Pipeline object in Spark
- Can be done like batch processing
- Adapt agile method into Machine Learning



Take Care of Overfitting

Data exposition

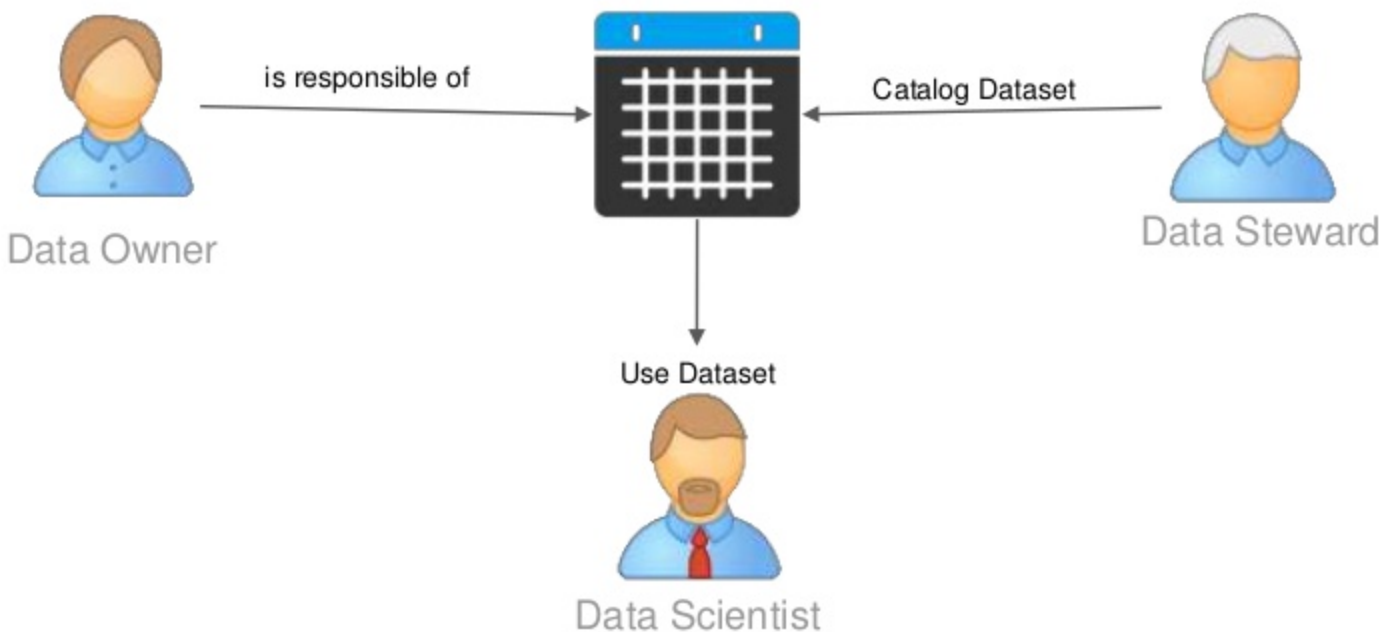
Data exposition

- After data treatment, we want to use this data.
  - For business purpose, we can expose data through a BI Tool
    - Tableau, microstrategy
  - Use it on webapp => Relational database
    - Better performances than datalake. Latency is reduced
  - Export it in csv format for diffusion

Data governance

Data responsibility

- In a company, who is responsible for data?
- How it is related to industrialization?



Data catalog

- What are my datasets in my datalake?
- Three major issues addressed by data catalog
  - What do we know about our information?
  - Where did this data come from and who can use it?
  - Does this data adhere to company policies and rules?
- Schema, last build date, metadatas
- Hive metastore, Collibra, Apache Atlas, Zeenea, Custom app

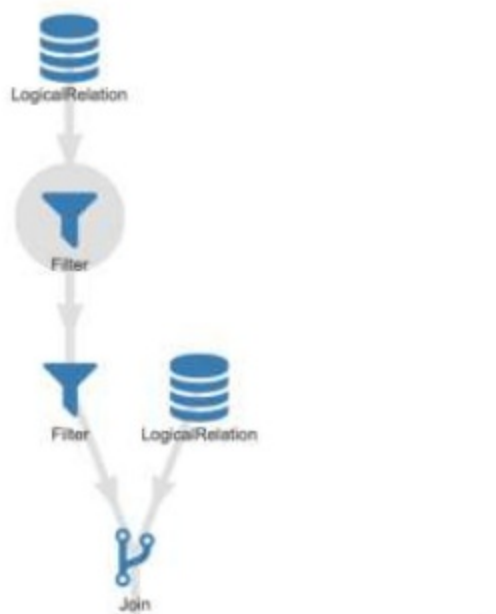
collibra

ZEENEA

Apache Atlas

Data lineage

- Getting informations about how the datasets are created and which operations were achieved to build the dataset.

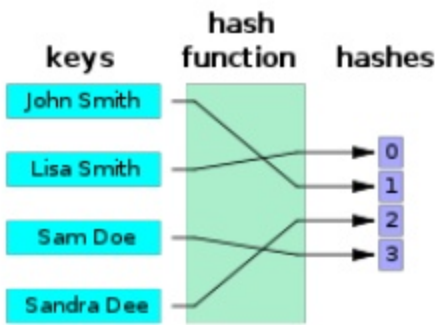


Dataset Automatocs Reloading

- Automatic rebuilding from lineage

RGPD

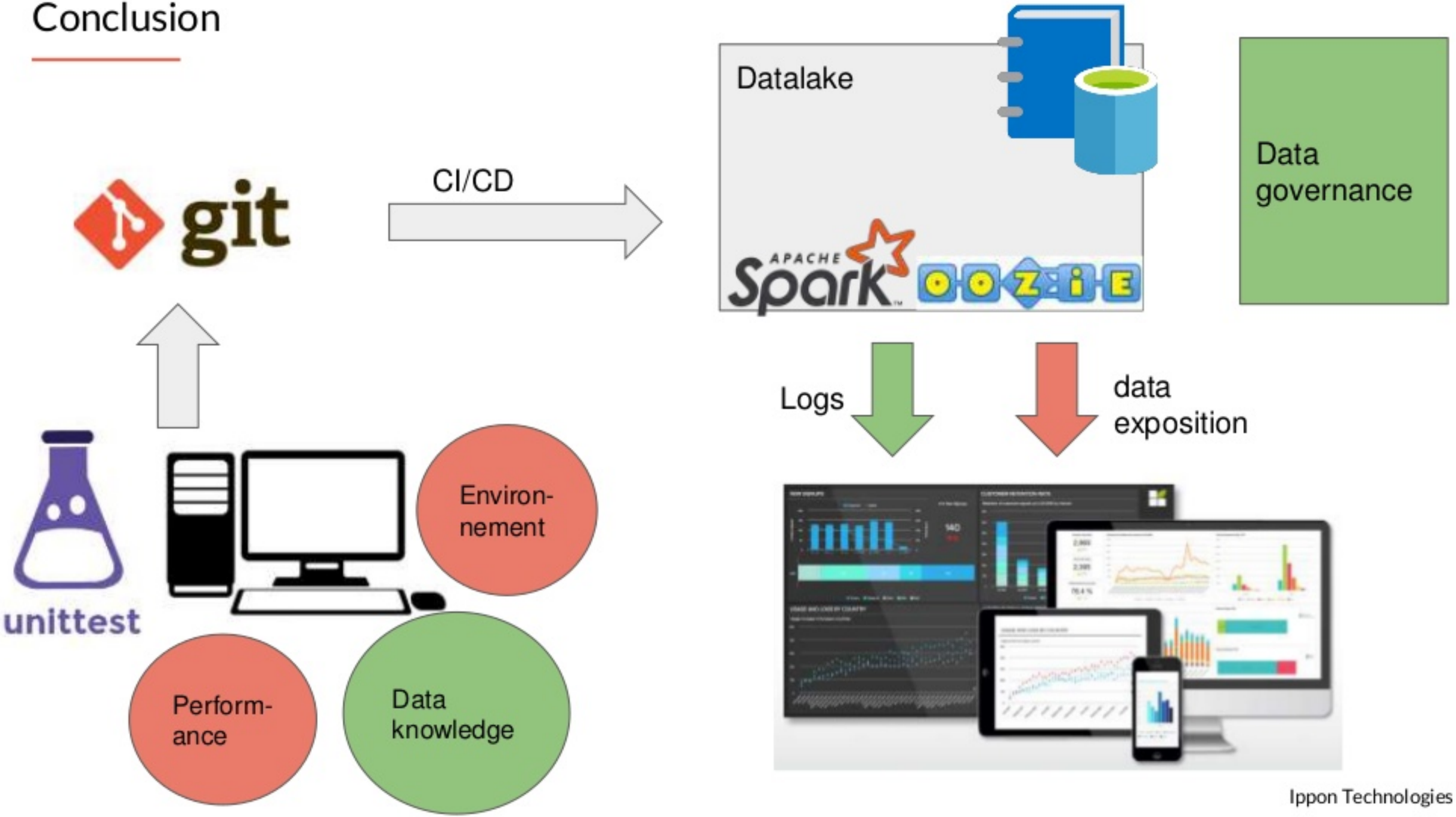
- RGPD is a huge Topic
- Job should anonymize data during the ingestion



- How to delete Data in a distributed immutable environment?

Conclusion

Conclusion


git
CI/CD
Datalake
APACHE Spark
OOZIE
Data governance
Logs
data exposition
unittest
Environ-nement
Perform-ance
Data knowledge

Questions ?

lucien fregosi

@lulufrego

Data engineer @ippontech. In love with distributed systems. Cloud first and ☕ Driven Development (CDD).

Ippon.fr

contact@ippon.fr

+33 1 46 12 48 48

@IpponTech