AWS Big Data Demystified #1.2
Big Data Architecture
Lessons Learned

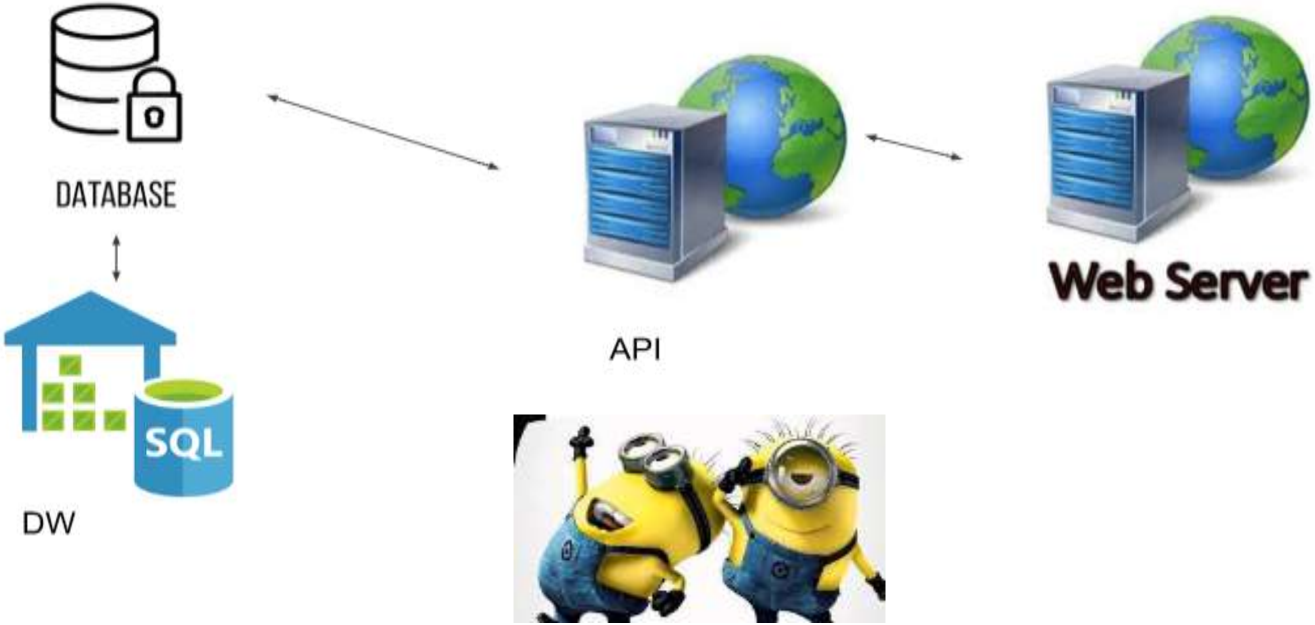
Omid Vahdaty, Big Data Ninja

Disclaimer

- I am **not** the **best**, I simply **love** what I do VERY much.
- You are more than welcome to **challenge** me or anything I have to say as I could be wrong.
- This Lecture has evolves over time, this is the 3nd **iteration**.
- Feel Free to **send me comments**

A long time ago
in a galaxy far, far away....

In the Past(web,api, ops db, data warehouse)



Then came Big Data...

Then came the cloud...


Then came the invoice ...


It keeps growing….
INFRASTRUCTURE
ANALYTICS
APPLICATIONS – ENTERPRISE
APPLICATIONS – INDUSTRY
DATA SOURCES & APIS
DATA RESOURCES

TODAY'S BIG DATA APPLICATION STACK PaaS and DC...

redis
amazon REDSHIFT
presto
MySQL workbench
DynamoDB
MESOS
kubernetes
Apache mahout
amazon RDS
APACHE kafka
Amazon S3
Amazon Athena
Thrift
docker
Apache Flink
R
Amazon Kinesis
FLUME
hadoop YARN
Spark SQL
LIVY
APACHE PIG
Apache Zeppelin
APACHE SHIRO
Kinesis Streams
Kinesis Firehose
Kinesis Analytics
hadoop HDFS
HIVE
OOZIE
HUE
PHOENIX
mxnet
TensorFlow
TEZ
sqoop
Airflow

MY BIG DATA APPLICATION STACK “demystified”...

redis

amazon REDSHIFT

presto

MySQL workbench

DynamoDB

amazon RDS

Amazon S3

Amazon Athena

Thrift

R

LIVY

Spark SQL

hadoop YARN

APACHE SHIRO

Apache Zeppelin

Amazon Kinesis

Kinesis Streams

Kinesis Firehose

Kinesis Analytics

hadoop HDFS

HIVE

OOZIE

HUE

TEZ

MY AWS BIG DATA APPLICATION STACK
With some "Myst".

redis

amazon REDSHIFT

presto

MySQL workbench

DynamoDB

amazon RDS

Amazon S3

Amazon Athena

Amazon Kinesis

Kinesis Streams

Kinesis Firehose

Kinesis Analytics

amazon EMR

Jargon

Big data?

Architecture?

Considerations?

Challenges?

How to get started?

Big Jargon & Basics concepts u should know

https://amazon-aws-big-data-demystified.ninja/2019/02/18/big-data-jargon-faqs-and-everything-you-wanted-to-know-and-didnt-ask-about-big-data/

- What is Big Data?
- scale out / up ?
- structured/semi structured/unstructured data ?
- ACID?
- OLAP VS OLTP? == Analytics VS operational
- DIY = Do it Yourself
- PaaS = Platform as a service

Big Data =
When your data outgrows your infrastructure ability to process

- Volume (x TB processing per day)
- Velocity (x GB/s)
- Variety (JSON, CSV, events etc)
- Veracity (how much of the data is accurate?)

Challenges creating big data architecture?

- What is the business use case ? How fast do u need the insights?
  - 15 min - 24 hours delay and above → use batch
  - Less than 15 min?
    - Might be batch - depends data source is files or events?.
    - **Streaming**?
      - Sub seconds delay?
      - Sub minute delay?
    - Streaming with **in flight** analytics ?
  - How complex is the **compute jobs?** Aggregations? joins?

Challenges creating big data architecture?

- What is the **Velocity**?
    - Under 100K events per second? Not a problem
    - Over 1M events per second? Costly. But doable.
    - Over 1B events per seconds? Not trivial at all.
- **Volume** ?
    - ~1TB a day ? Not a problem
    - Over ? it depends.
    - Over a petabyte? Well.... It depends.
- **Veracity** (how are you going to handle different data sources?)
    - Structured (CSV)
    - Semi structured (JSON,XML)
    - Unstructured (pictures, movies etc)

Challenges creating big data architecture?

- Performance targets?
- Costs targets?
- Security restrictions?
- Regulation restriction? privacy?
- Which technology to choose?
- Datacenter or cloud?
- Latency?
- Throughput?
- Concurrency?
- Security Access patterns?
- Pass? Max 7 technologies
- Iaas? Max 4 technologies

Cloud Architecture rules of thumb...

- **Decouple** :
  - Store
  - Process
  - Store
  - Process
  - insight...
- **Rule of thumb: max 3 technologies in dc, 7 tech max in cloud**
  - Do use more b/c: maintenance
  - Training time
  - complexity/simplicity

How to get started on big data architecture?
Get answers to the below:

1. What is the business use case?
   a. Volume? velocity? Variety? vercity/
   b. Did you map all of data sources?
2. Where should we build a data platform?
   a. What is the product? → requirements.
   b. Cloud? datacenter?
3. Architecture?
   a. DIY? Paas? , Pay as you go? Or fixed? Decoupled?
   b. Fast? Cheap? simple?
4. Did you Communicate you plans?
5. Did you map all known challenges?

Business Use Case?

Use Case 1: Analyzing browsing history

- Data Collection: browsing history from an ISP
- Product - derives user intent and interest for marketing purposes.
- Challenges
  - Velocity: 1 TB per day
  - History of: 3M
  - Remote DC
  - Enterprise grade security
  - Privacy

Use Case 2: Insights from location based data

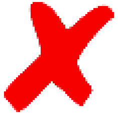
- Data collection: from a Mobile operator
- Products:
  - derives user intent and interest for marketing purposes.
  - derive location based intent for marketing purposes.
- Challenges
  - **Velocity: 4GB/s …**
  - Scalability: Rate was expected double every year...
  - Remote DC
  - Enterprise grade security
  - **Privacy**
  - **Edge analytics**

Use Case 3: Analyzing location based events.

- Data collection: streaming
- Product: building location based audiences
- Challenges: **<u>minimizing DevOps work on maintenance</u>**

Getting started

(notice the markings on upper right corner)



I didn't choose this technology

I choose this technology

So what is the product?

- Big data platform that
  - **ingests** data from multiple sources (cloud and DC)
  - **Analyzes** the data
  - **Generates** insights :
    - Smart **Segments** (online marketing)
    - Smart **reports** (for marketer)
    - **Audience** analysis (for agencies)
- Customers?
  - Marketers
  - Publishers
  - Agencies

Where to build the data platform?

We choose AWS because

- After a long **competitive analysis** we choose AWS because, it seems to have all the relevant features For all our big data products and wallas publisher products
- The project was challenging enough, without adding the complexity of a **learning curve** (learning new cloud). We already knew how to work with AWS
- Of course, there business aspects as well.

My Big Data product does:

- Data **Ingestion**
  - **Online**
    - messaging
    - Streaming
  - **Offline**
    - Batch
    - Performance aspects
- Data **Transformation** (Hive)
  - JSON, CSV, TXT, PARQUET, Binary
- Data **Modeling** - (R, ML, AI, DEEP, SPARK)
- Data **Visualization** (choose your poison)
- **PII** regulation + **GPDR** regulation
- And: **Performance**... **Cost... Security... Simple**... **Cloud best practices**...

Big Data Generic Architecture

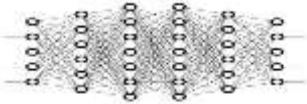
Amazon S3

Parquet

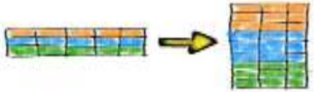
Text,
RAW

Data **Presentation**

Data **Modeling** (joins/agg/ML/R)



Data **Transformation** (**row** to **colunar** + cleansing)



amazon webservices™

Data **Ingestion**
(file based ETL from remote DC)

Data Ingestion

A layer in your big data architecture designed to do one thing : ingest data via Batch or Streaming, I.e move (only) data from point A to point B. from source data to the next layer in the architecture (decoupled).

Big Data Generic Architecture | Data Ingestion

Amazon S3

Data Presentation

Data Modeling

Data Transformation

Data Ingestion

Java

Amazon Kinesis

Kinesis Streams

Kinesis Firehose

Kinesis Analytics

Batch Data collection considerations

- Every hour , about 30GB compressed CSV file
- Why s3
  - Multi part upload
  - S3 CLI
  - S3 SDK
  - (tip : gzip!)
- Why ETL Client - needs to run at remote DC
- Why NOT your **own ETL client**
  - Involves code →
    - **Bugs?**
    - **maintenance**
  - **Don't analyze data at Edge , cant go back in time.**
- Why **Not Streaming**?
  - **less accurate**
  - **Expensive**

S3 Considerations

- Security
  - at rest: server side S3-Managed Keys (**SSE-S3**)
  - at transit: **SSL** / VPN
  - **Hardening**: **user, IP ACL, write permission only.**
- Upload
  - AWS **s3 cli**
  - **Multi part** upload
  - **Aborting Incomplete Multipart Uploads Using a Bucket Lifecycle Policy**
  - Consider S3 CLI **Sync** command instead of **CP**

Sqoop - ETL

- Open source , **part** of **EMR**
- **HDFS** to **RDMS** and back. Via **JDBC**.
- E.g BiDirectional **ETL** from **RDS** to **HDFS**
- **Unlikely use case: ETL from customer source operational DB.**

Flume & Kafka

- Opens source project for streaming & messaging
- Popular
- Generic
- Good practice for many use cases. (a meetup by it self)
- Highly durable, scalable, extension etc.
- **Downside : DIY, Non trivial to get started**

Data Transfer Options

- Direct Connect (4GB/s?)
- For all other use case
    - S3 multipart upload
    - Compression
    - Security
        - Data at motion
        - Data at rest

AWS Direct Connect

Quick intro to Stream ingestion

- Kinesis Client Library (code)
- AWS lambda (code)
- EMR (managed hadoop)
- Third party (DIY)
    - **Spark streaming** (latency min =1 sec) , near real time, with lot of libraries.
    - **Storm** - Most real time (sub millisec), java code based.
    - **Flink** (similar to spark)

Kinesis family of products

- **Kinesis Stream** - **collect@source** and near real time proce
  - **Near real time**
  - **High throughput**
  - Low cost
  - Easy administration - set desired level of capacity
  - Delivery to : s3,redshift, Dynamo, ...
  - Ingress 1mb, egress 2mbs. Upto 1000 Transaction per second.
  - Not managed!
- **Kinesis Analytics** - **in flight analytics.**
- **Kin. Firehose** - Park you **data @ destination.**

Kinesis Firehose - for Data parking

- **Not for fast lane** - **no in flight analytics**
- Ingest , transform and load to:
  - Kinesis
  - S3
  - Redshift
  - elastic search
- Managed Service

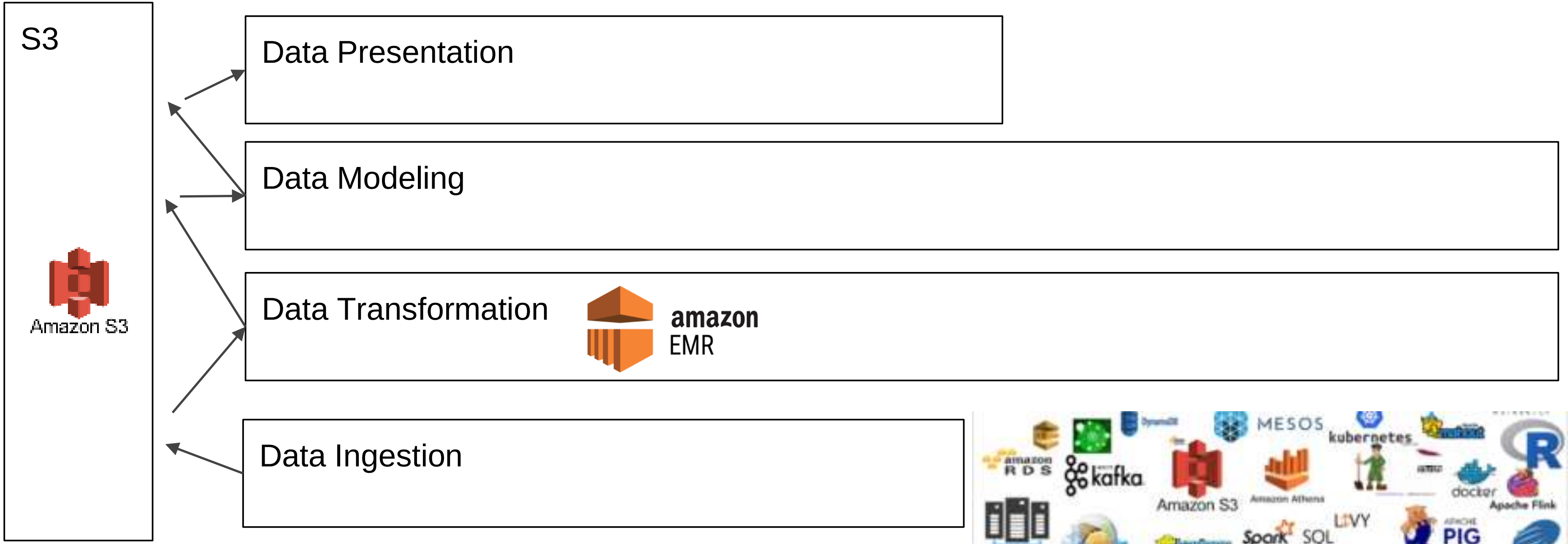
Comparison of Kinesis products

- Streams
  - **Sub 1 sec processing latency**
  - Choice of stream processor (generic)
  - For smaller events
- Firehose
  - **Zero admin**
  - 4 targets built in (redshift, s3, search, etc)
  - **Buffering 60 sec minimum.**
  - For larger “events”

Data Transformation

A layer in your big data architecture designed to : Transform and Cleanse data (row data to columnar data and convert data types, Fix bugs in data)

Big Data Generic Architecture | Transformation



EMR ecosystem

- Hive
- Pig
- Hue
- Spark
- Oozie
- Presto
- Ganglia
- Zookeeper (hbase)
- zeppelin

EMR Architecture

amazon EMR

- Master node
- Core nodes - like data nodes (with storage: HDFS)
- Task nodes - (extends compute)
- Does Not have Standby Master node
- Best for transient cluster (goes up and down every night)

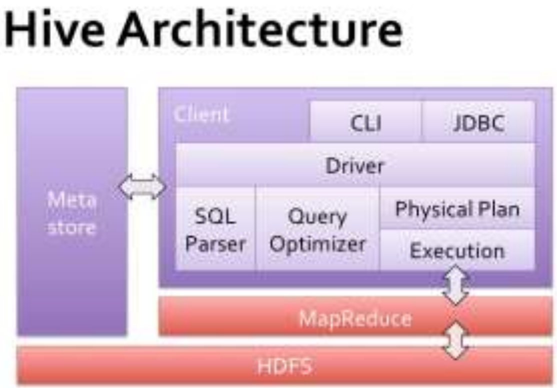
EMR lesson learned...

amazon EMR

- **Bigger** instance type is good architecture
- Use **spot** instances - for the tasks only.
- Don't always use **TEZ (MR? Spark?)**
- Make sure your choose instance with **network optimized**
- **Resize** cluster is not recommended
- **Bootstrap** to automate cluster upon provisioning
- Use **Steps** to automate steps on running cluster
- Use **Glue** to share Hive MetaStore
- Good Cost reduction article on EMR

So use EMR for ...

- Most dominant
  - **Hive**
  - **Spark**
  - **Presto**
- And many more....
- Good for:
  - Data **transformation**
  - Data **modeling**
  - **Batch**
  - **Machine learning**



Hive

- SQL over hadoop.
- Engine: spark, tez, MR
- JDBC / ODBC
- Not good when need to shuffle.
- Not peta scale.
- SerDe json, parquet,regex,text etc.
- Dynamic partitions
- Insert overwrite
- Data Transformation
- Convert to Columnar

Presto

- SQL over hadoop
- Not good always for join on 2 large tables.
- Limited by memory
- Not fault tolerant like hive.
- **Optimized for ad hoc queries**
- No insert overwrite
- No dynamic partitions.
- **Has some connectors : redshift and more**
- **https://amazon-aws-big-data-demystified.ninja/2018/07/02/aws-emr-presto-demystified-everything-you-wanted-to-know-about-presto/**

Pig

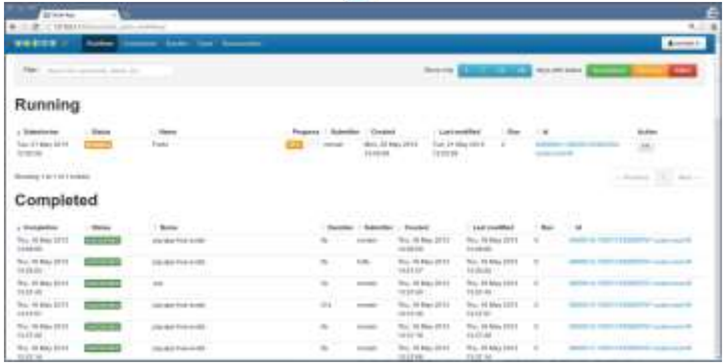
- Distributed Shell scripting
- Generating SQL like operations.
- Engine: MR, Tez
- S3, DynamoDB access
- Use Case: for data science who don't know SQL, for system people, for those who want to avoid java/scala
- Fair fight compared to hive in term of performance only
- **Good for unstructured files ETL : file to file , and use sqoop.**

Hue

- Hadoop user experience
- Logs in real time and failures.
- Multiple users
- Native access to S3.
- File browser to HDFS.
- Manipulate metascore
- Job Browser
- Query editor
- Hbase browser
- Sqoop editor, oozier editor, Pig Editor

Orchestration

- EMR Oozie
  - Opens source workflow
    - Workflow: graph of action
    - Coordinator: scheduler jobs
  - Support: hive, sqoop , spark etc.
- Other options: AirFlow, Knime, Luigi, Azkaban,AWS Data Pipeline



Big Data Generic Architecture | Transformation

S3

Amazon S3

Data Visualization

Data Modeling

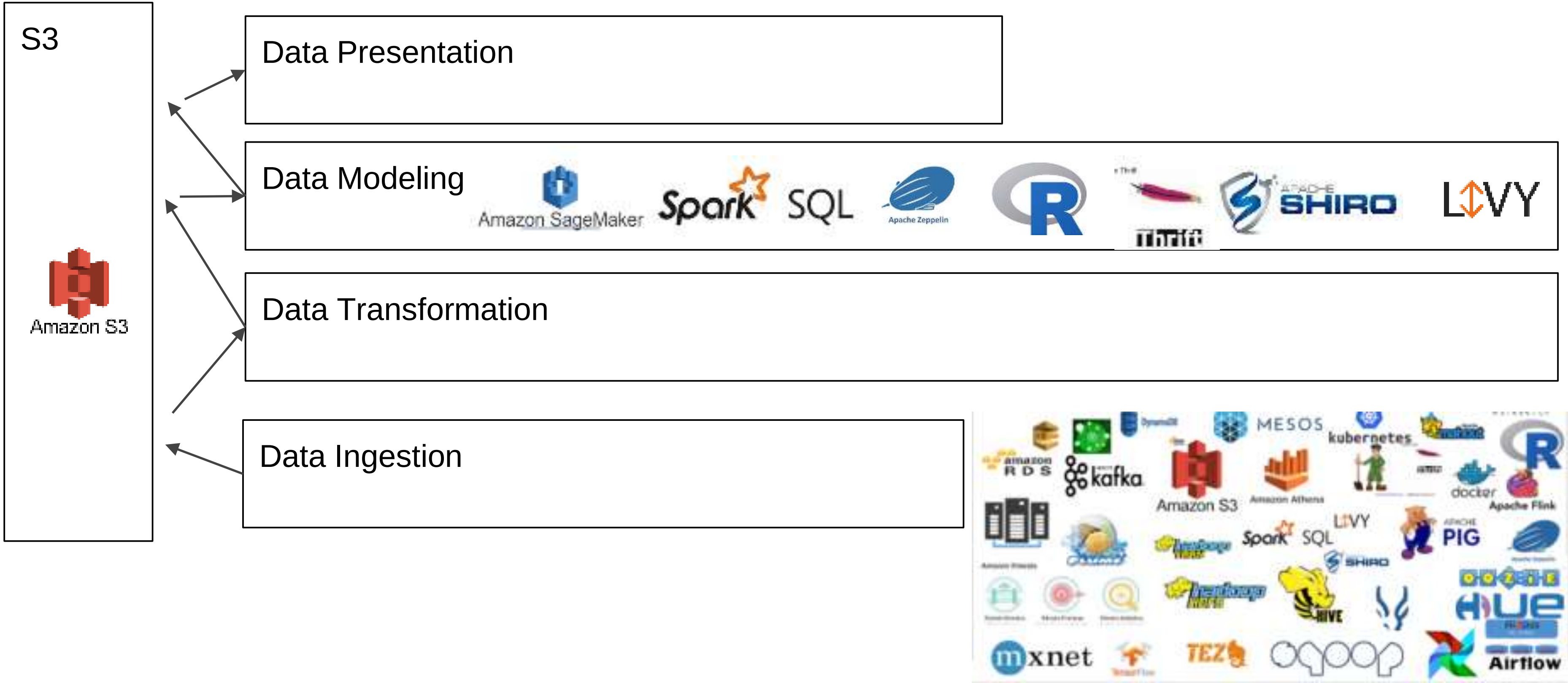
Data Transformation

Data Ingestion

Data Modeling

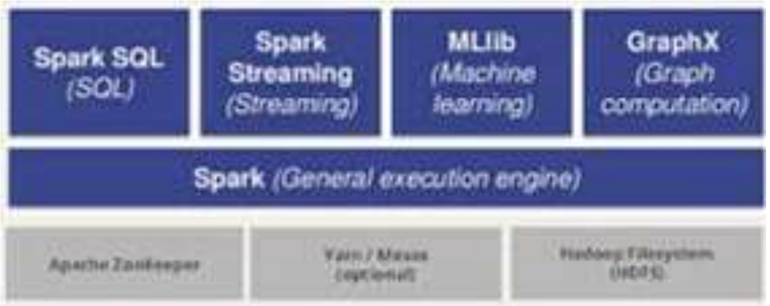
A layer in your big data architecture designed to Model data: Joins, Aggregations, nightly jobs, Machine learning

Big Data Generic Architecture | Modeling



Spark

- In memory
- X10 to X100 times faster from hive
- Good optimizer for distribution
- Rich API
- Spark SQL
- Spark Streaming
- Spark ML (ML lib)
- Spark GraphX (DB graphs)
- SparkR



Spark Streaming

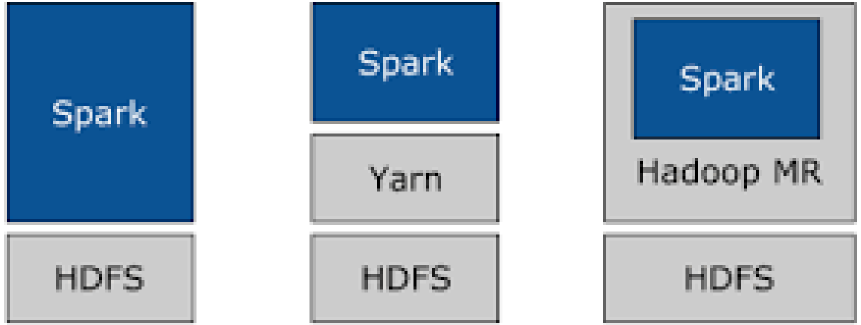
- Near real time (1 sec latency)
- like batch of 1sec windows
- Streaming jobs with API
- **DIY = Not relevant to us...**

Spark ML

- Classification
- Regression
- Collaborative filtering
- Clustering
- Decomposition
- Code: java, scala, python, sparkR

Spark flavours

- Standalone
- With yarn
- With mesos



(a) Standalone (b) Over Yarn (c) Spark in MR (SIMR)

Spark Downside

- **Compute intensive**
- Performance gain over mapreduce is not guaranteed.
- Streaming processing is actually batch with very small window.

Spark SQL

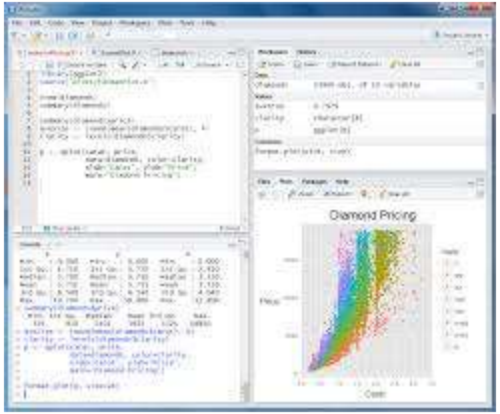
- Same syntax as hive
- Optional JDBC via thrift
- Non trivial learning curve
- Upto X10 faster than hive.
- Works well with Zeppelin (out of the box)
- Does not replaces Hive
- Spark not always faster than hive
- insert overwrite -

Apache Zeppelin

- Notebook - visualizer
- Built in spark integration
- Interactive data analytics
- Easy collaboration.
- Uses SQL
- works on top of Hive/ Spark SQL
- Inside EMR.
- Uses in the background:
    - **Shiro**
    - **Livy**

R + spark R

- Open source package for statistical computing.
- Works with EMR
- “Matlab” equivalent
- Works with spark
- Not for developer :) for statistician
- **R is single threaded - use spark R to distribute.**
- Not everything works perfect.



Redshift

- OLAP, not OLTP→ analytics , not transaction
- Fully SQL
- Fully ACID
- No indexing
- Fully managed
- Petabyte Scale
- MPP
- Can create slow queue for queries
    - which are long lasting.
- DO NOT USE FOR transformation.
- Good for : DW, Complex Joins.

Redshift spectrum

- Extension of Redshift, use external table on S3.
- Require redshift cluster.
- Not possible for CTAS to s3, complex data structure, joins.
- Good for
  - Read only Queries
  - Aggregations on Exabyte.

EMR vs Redshift

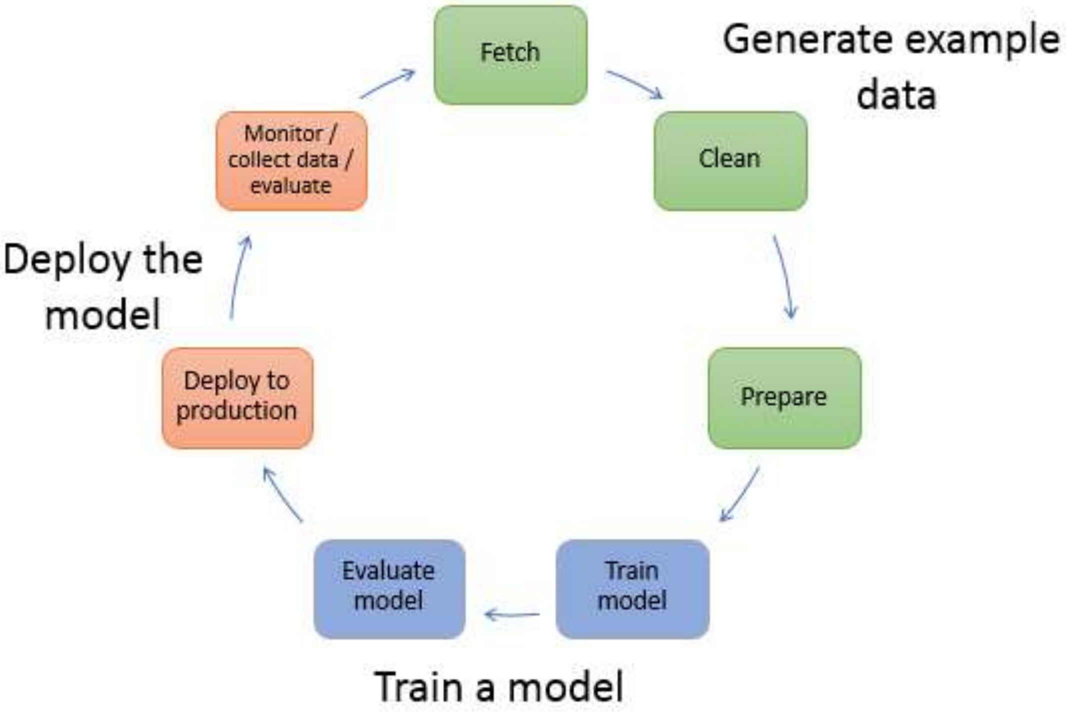
- How much data loaded and unloaded?
- Which operations need to performed?
- Recycling data? → EMR
- History to be analyzed again and again ? → emr
- What the data needs to end up? BI?
- Use spectrum in some use cases. (aggregations)?
- Raw data? S3.
- When to use emr and when redshift?

Hive VS. Redshift

- Amount of concurrency ? low → hive, high → redshift
- Access to customers? Redshift? athena?
- Transformation, Unstructured , batch, ETL → hive.
- Peta scale ? redshift
- Complex joins → Redshift

Sage Maker

- Web notebook (jupiter based) for data science
- Connects to all your data sources (s3,athena etc)
- Help you manage the entire lifecycle machine learning
- Managed Service
- Used to create a ML to predict cookie gender



AWS Glue

Shared meta store

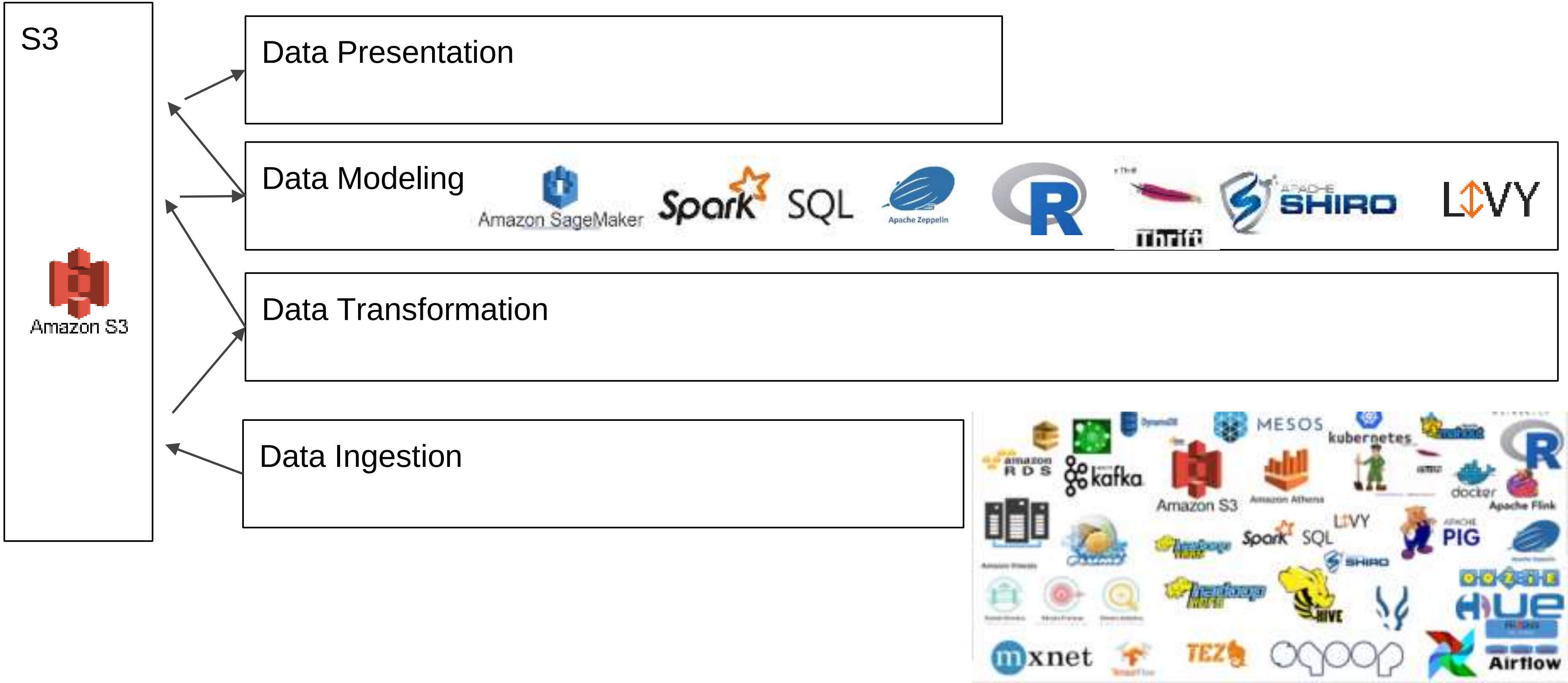
Helps with some data transformation (managed service)

Automatic Schema discovery

AWD RDS (aurora, postgres, mysql)

- We used RDS aurora as Operational DB
- We did not use it for big data analytics although it supports upto 64Tb
- It is row based.
- The syntax is missing analytical functions

Big Data Generic Architecture | Modeling



Data Presentation

Used ONLY for presenting data for operational applications or BI, Use managed service to ensure HA.

Big Data Generic Architecture | Presentation

S3

Amazon S3

Data Visualization

Amazon Athena

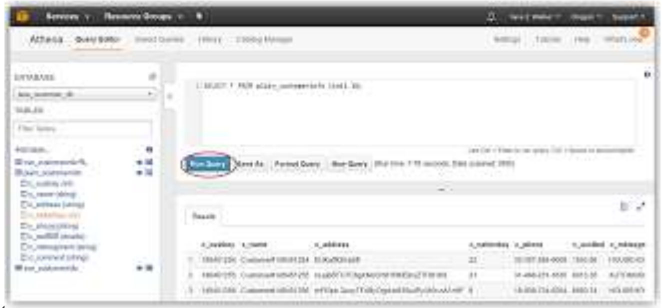
Qlik Sense

Data Modeling

Data Transformation

Data Collection

Athena



Amazon Athena

- Presto SQL
- In memory
- Hive metastore for DDL functionality
    - Complex data types
    - Multiple formats
    - Partitions
- Now supports CTAS (No inserts are supported)
- Good for:
    - Read only SQL,
    - Ad hoc query,
    - low cost,
    - Managed
- [Good cost reduction article on athena]

Visualize

- QuickSight
- Managed Visualizer, simple, cheap

Summery

Take Away Message & Action Items

Big Data Generic Architecture | Summary

S3

Amazon S3

Data Presentation

Amazon Athena

Qlik Sense

Data Modeling

Amazon SageMaker

Spark SQL

Apache Zeppelin

Thrift

APACHE SHIRO

LIVY

Data Transformation

HIVE

hadoop YARN

hadoop HDFS

TEZ

HUE

OOZIE

Data Ingestion

Java

Amazon Kinesis

Kinesis Streams

Kinesis Firehose

Kinesis Analytics

Summary: Lesson learned

- **Decouple, Decouple, Decouple**
- **Productivity** of Data Science and Data engineering
  - Common language of both teams IS SQL!
  - Minimize the life cycle from dev to production of ETL and ML jobs
- **Minimize** the amount DB's used
  - Different **syntax** (presto/hive/redshift)
  - Different **data types**
  - **Minimize ETLS via External Tables+Glue!**
- Not always **Streaming** is justified (what is the business use case? PaaS?)
- **Spark SQL**
  - Sometimes **faster** than **redshift**
  - Sometimes **slower** than **hive**
  - Learning curve is **non trivial**

Summery: Common Q&A

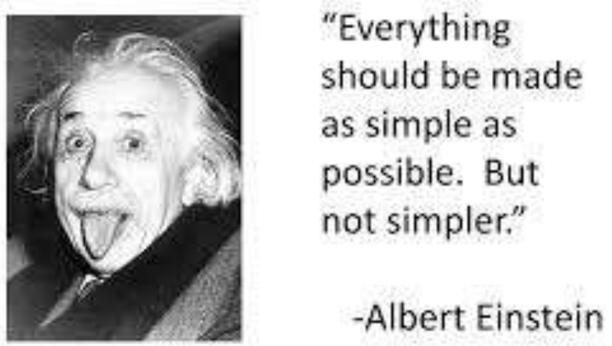
1. Can this architecture be done on another cloud?
2. Redshift VS EMR ?
3. Athena VS Redshift?
4. Cost reduction on EMR?
5. Cost Reduction on Athena?
6. Exporting data from Google Analytics into AWS?

Lesson learned: Big Data Architecture ?

Faster!

Cheaper!

Simpler!



How to get started | Call for Action

Lectures: AWS Big Data Demystified lectures #1 until #4

[AWS Big Data Demystified Meetup]

[Big Data Demystified meetup]

Stay in touch...

- Omid Vahdaty
- +972-54-2384178
- https://big-data-demystified.ninja/
- Join our meetups subscribe to youtube channels
    - https://www.meetup.com/AWS-Big-Data-Demystified/
    - https://www.meetup.com/Big-Data-Demystified/
    - Big Data Demystified YouTube
    - AWS Big Data Demystified YouTube
    - WhatsApp group