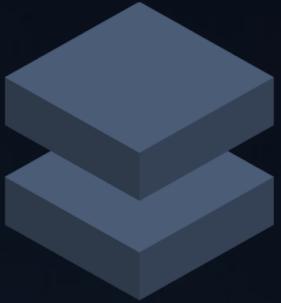




ONE WAY
SOLUTION



One Way Solution

Foundation of Apache Spark

Data Engineering – [Day 1]



LUAN MORENO

CEO & CDO

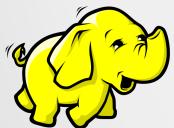
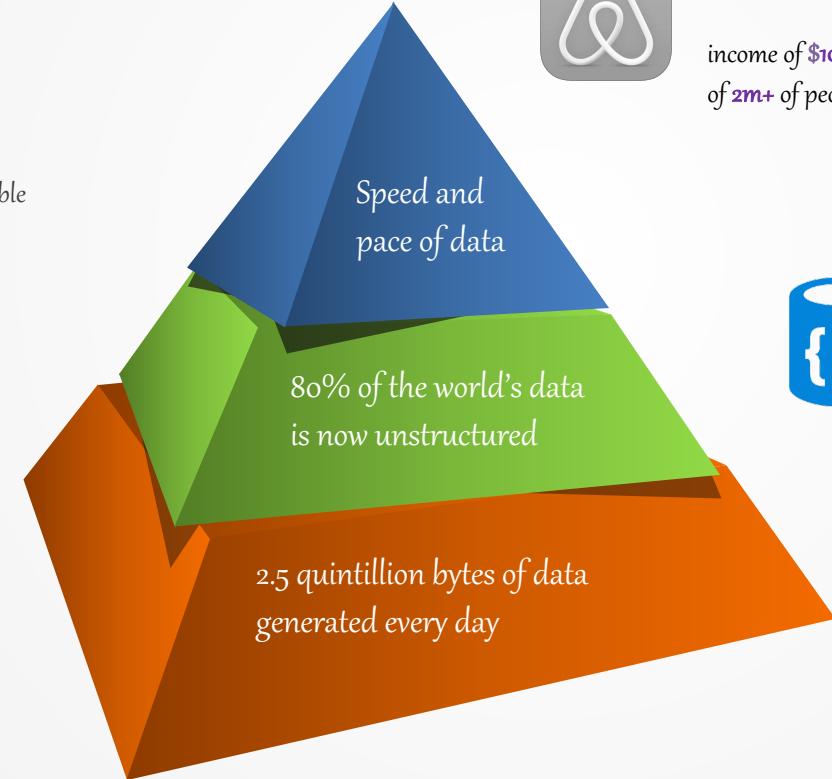
Data Engineer & Data Platform MVP

A wide-angle landscape photograph of a mountain range. The foreground is filled with dark, snow-dusted evergreen forests. In the middle ground, several majestic peaks rise, their slopes covered in patches of white snow. The sky above is a clear, pale blue, suggesting either early morning or late afternoon light.

Do not fear mistakes. You
will know failure.
Continue to reach out.

Benjamin Franklin

Big Data



Apache Hadoop [2006]

quantity of generated & stored data, size of data & value of potential insights



Apache Kafka [2014]

speed of data generation & processing large datasets, ability to ingest data as fast as possible

Batch, Near, **Real-Time**



Netflix, Inc.

137 million users worldwide with consumption of 25% of the world's internet bandwidth



Spotify

191 million users worldwide with more than 30 million of songs available



Airbnb, Inc.

income of \$107 millions with average of 2m+ of people staying in places



Lyft, Inc.

1m+ million of rides per day and 30M+ of users worldwide



NoSQL [2009]

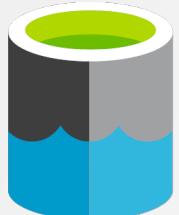
type & nature of data, different data sources & mappings, easier for developers

Key-Value Pairs, Column-Family, **JSON**, Graph

MB | GB | TB | **PB**

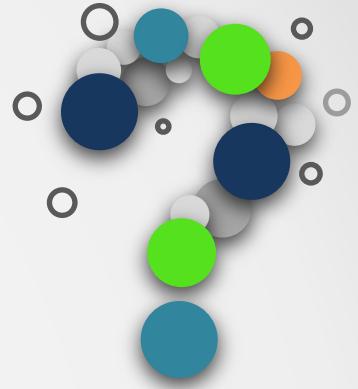
Data Lake

Repository of Raw Data
Without Schema Enforcement



Data Producers

Microservices + Data Stores
Feeding Data



Data Swamp

Without Contextual Metadata
and Data Governance



Data Governance

Management & Understanding of
Huge Amounts of Data



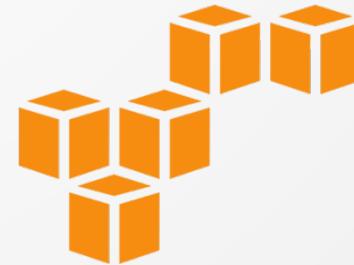
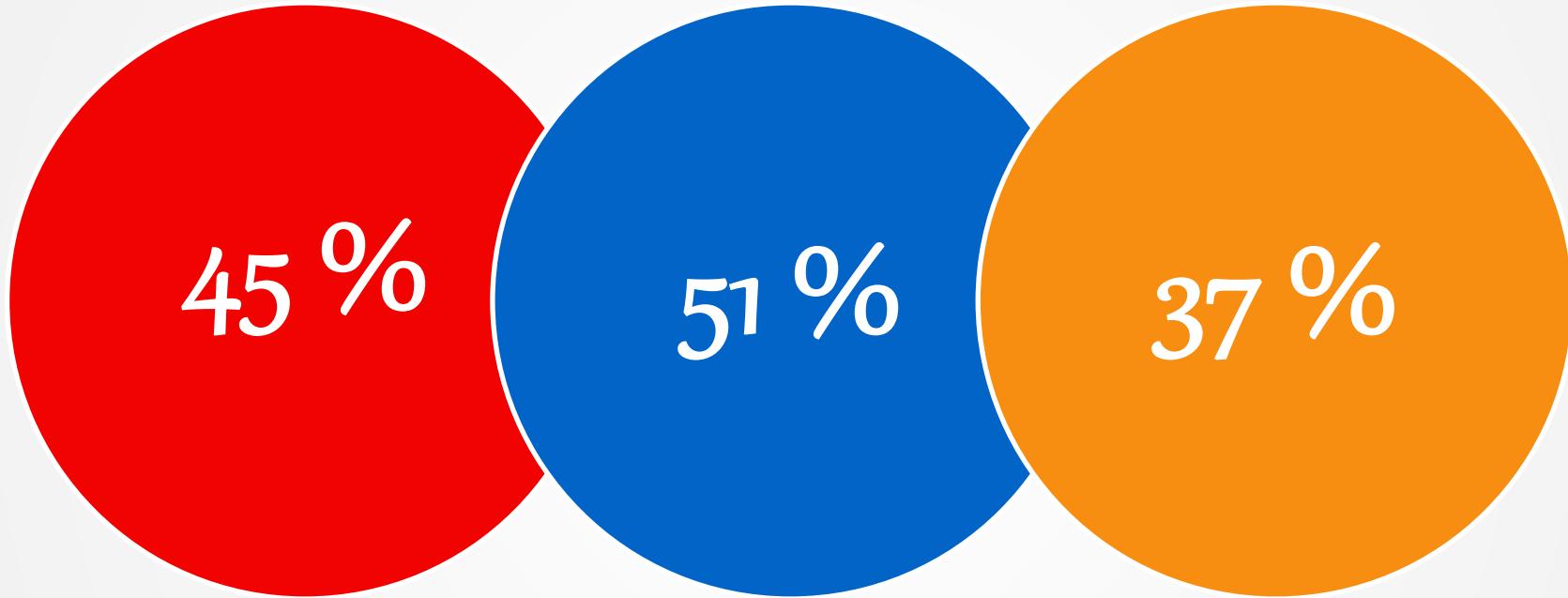
Veracity

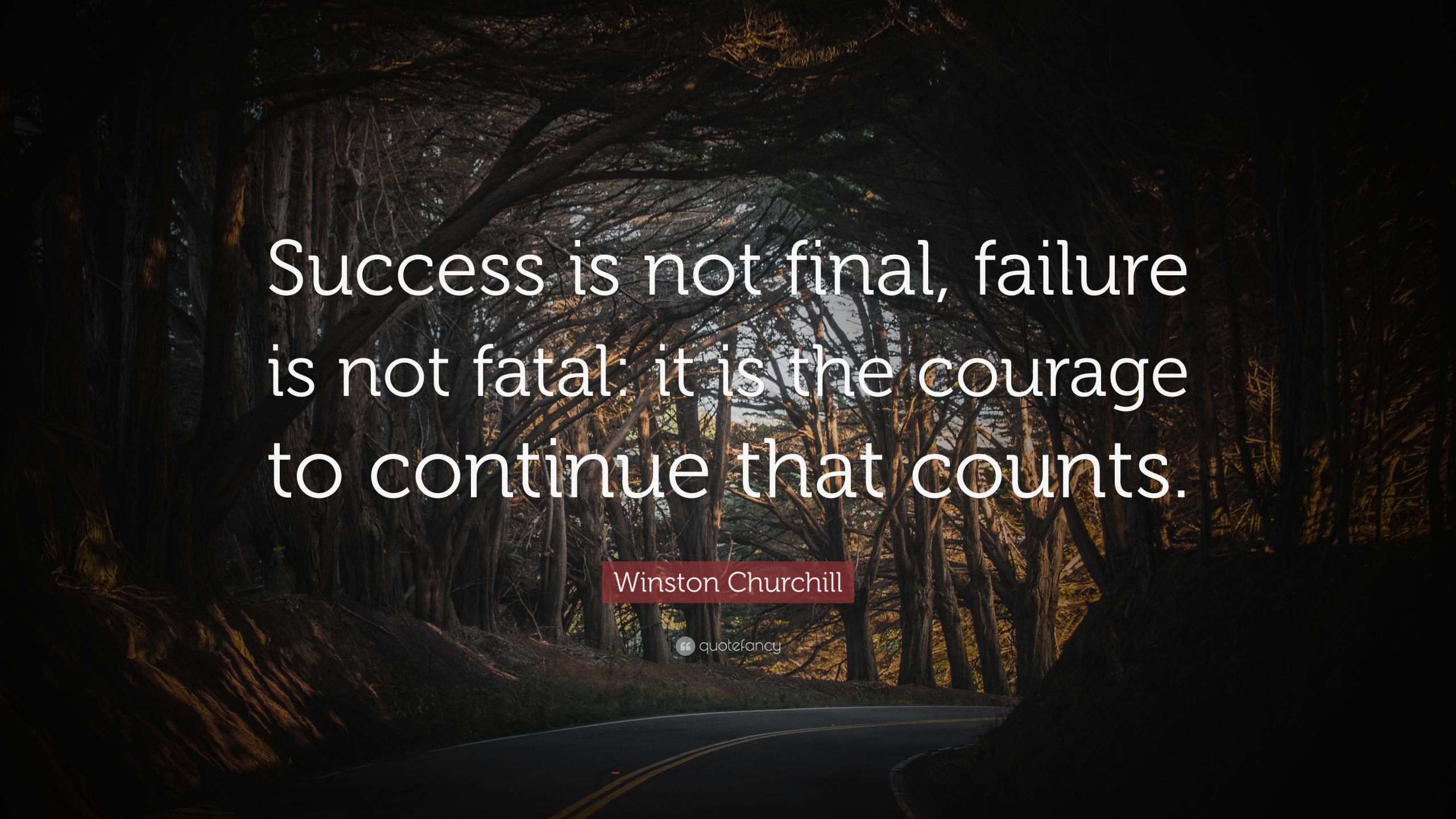


Precision of Data Stored with Processed
Data Quality & Enrichment for Insights



Google Cloud Platform vs. Microsoft Azure vs. Amazon Web Services



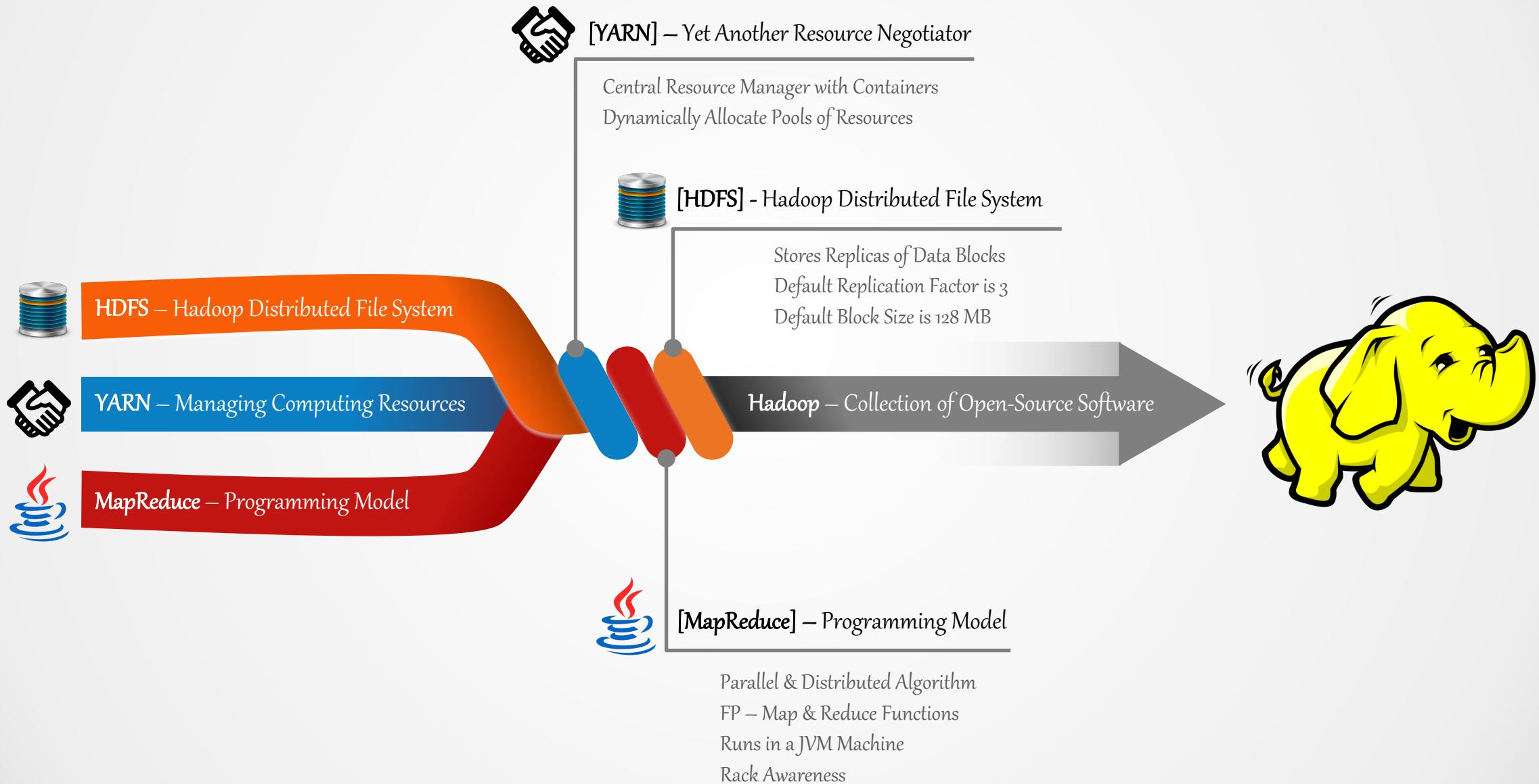


Success is not final, failure
is not fatal: it is the courage
to continue that counts.

Winston Churchill

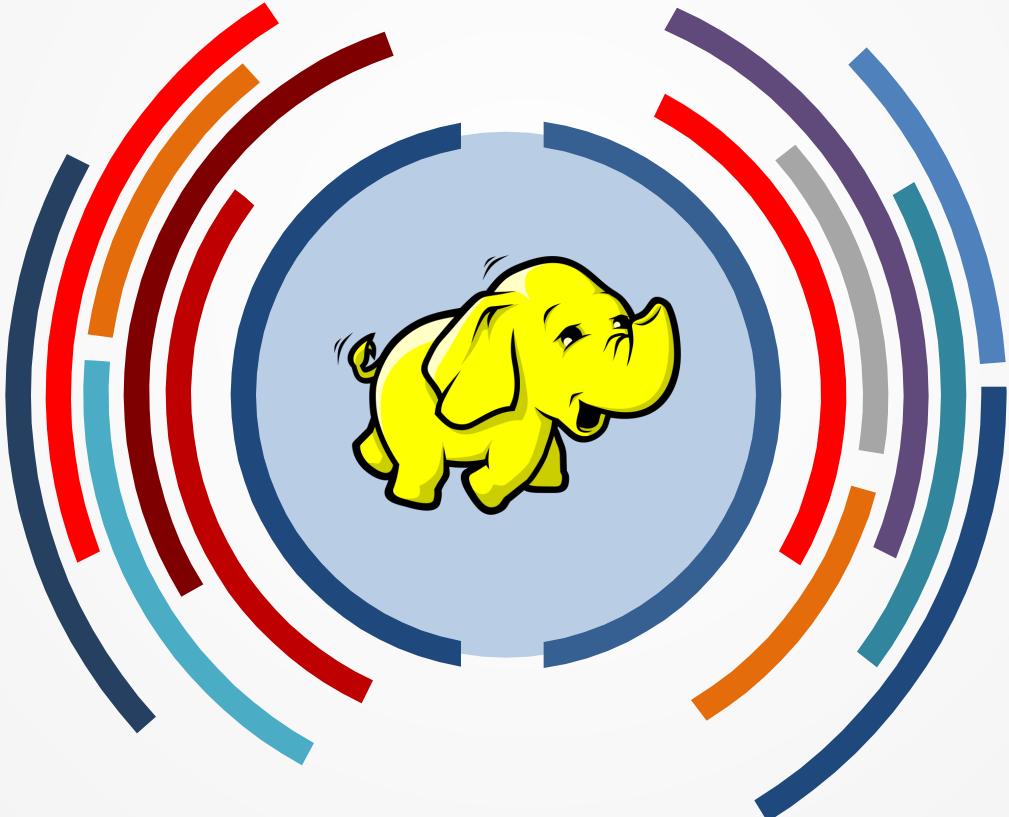


Apache Hadoop [Fundamentals]



Ecosystem of Hadoop Animal Zoo [2006 ~ 2014]

Apache Pig High-Level Platform Pig Latin for ETL Jobs
Apache Hive Data Warehouse with SQL-Like Interface Used By Facebook & Netflix
Apache HBase Non-Relational Distributed Database Used By Netflix & Spotify
Apache Phoenix Massively Parallel & Relational Database Skin of Apache HBase [ACID] OLTP
Apache Zookeeper Distributed Configuration Service Sync & Name Registry



Apache Flume Distributed & Reliable for Collecting & Aggregating Large Amounts of Log Data
Apache Storm Distributed Stream Processing Computation & Acquired by Twitter
Apache Sqoop Command-Line Interface for Transferring Data Between Relational DB's & Hadoop
Apache Oozie Server-Based Workflow Scheduling for Hadoop Jobs
Apache Mahout Scalable ML Focused with Collaborative Filtering Clustering & Classification

History of Apache Hadoop & Apache Spark



Doug Cutting
Started Working on Nutch



Google
Published MapReduce Paper



Google
Published GFS Paper
GFS & MapReduce Support



Nutch



Hadoop



Hadoop



Hadoop

2002

2003

2004

2005

2006

2008

2009

2011

2010

2019

2018

2017

2014

2011

2010



Apache Spark
Spark 3.0.2 Released
Spark 3.1.1 Released
Spark 3.1.2 Released



Apache Spark
Spark 2.4.5 Released
Spark 2.4.6 Released
Spark 3.0.0 Released
Spark 3.0.1 Released



Apache Spark
Spark 2.2.3 Released
Spark 2.3.3 Released
Spark 2.4.1 Released
Spark 2.4.2 Released
Spark 2.4.3 Released
Spark 2.4.4 Released



Apache Hadoop 3.1



Apache Hadoop 2.9
Apache Hadoop 3.0



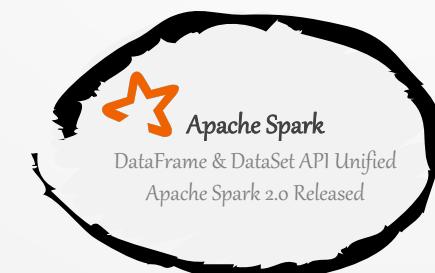
Apache Hadoop 2.3
Apache Hadoop 2.4
Apache Hadoop 2.5
Apache Hadoop 2.6



Facebook, LinkedIn, eBay & IBM
200,000 Lines of Code
42K Hadoop Nodes
Top Prize at Media Guardian Innovation Awards
Rob Beardson & Eric Badleschieler Spin HortonWorks



Yahoo 4,000 Nodes & 70 PB
Facebook 2,300 Clusters & 40 PB
Apache HBase Graduates
Apache Hive Graduates
Apache Pig Graduates
Apache Zookeeper Graduates

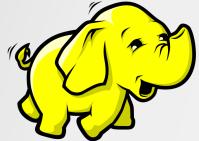


Kappa Architecture
Jay Kreps
Principal Staff Engineer



Lambda Architecture
Nathan Marz
Software Engineer at Twitter

Big Data-as-a-Services [BDaaS]



Hadoop-as-a-Service [HaaS]

Collection of Open-Source Software

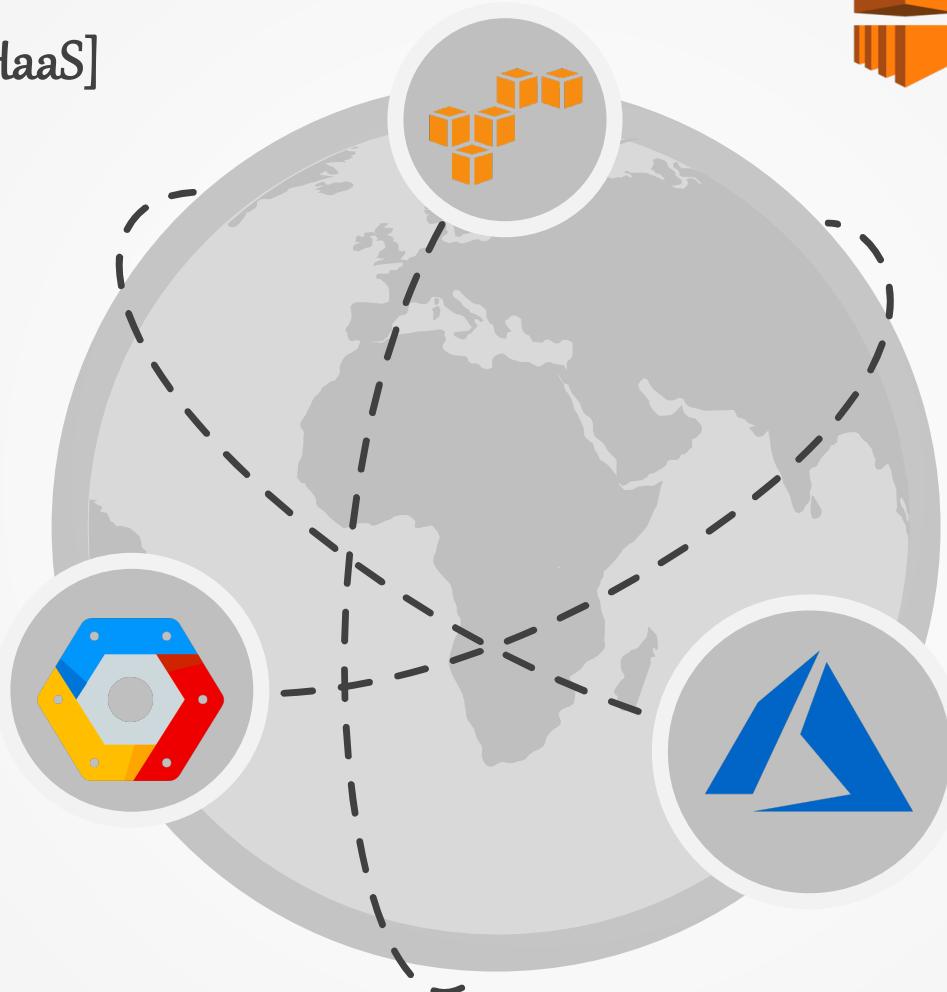
Fully-Managed Cloud Service

- Amazon AWS
- Microsoft Azure
- Google Cloud Platform



Cloud DataProc

Fully-Managed Cloud Service
Cloud Native Apache Hadoop & Apache Spark
Provisioning Time of 90 Seconds



Amazon Elastic MapReduce [EMR]

Easy Run & Scale Big Data Frameworks

Apache Hadoop
Apache Spark
HBase
Presto & Hive



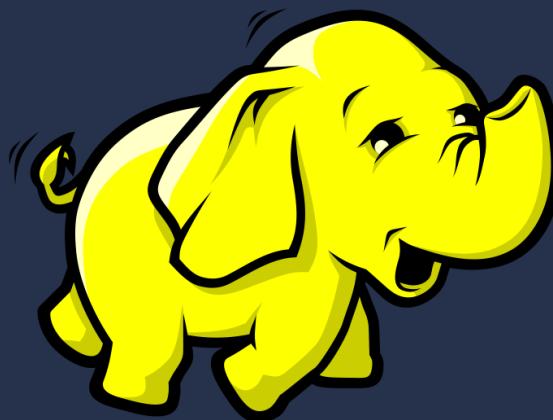
HDInsight

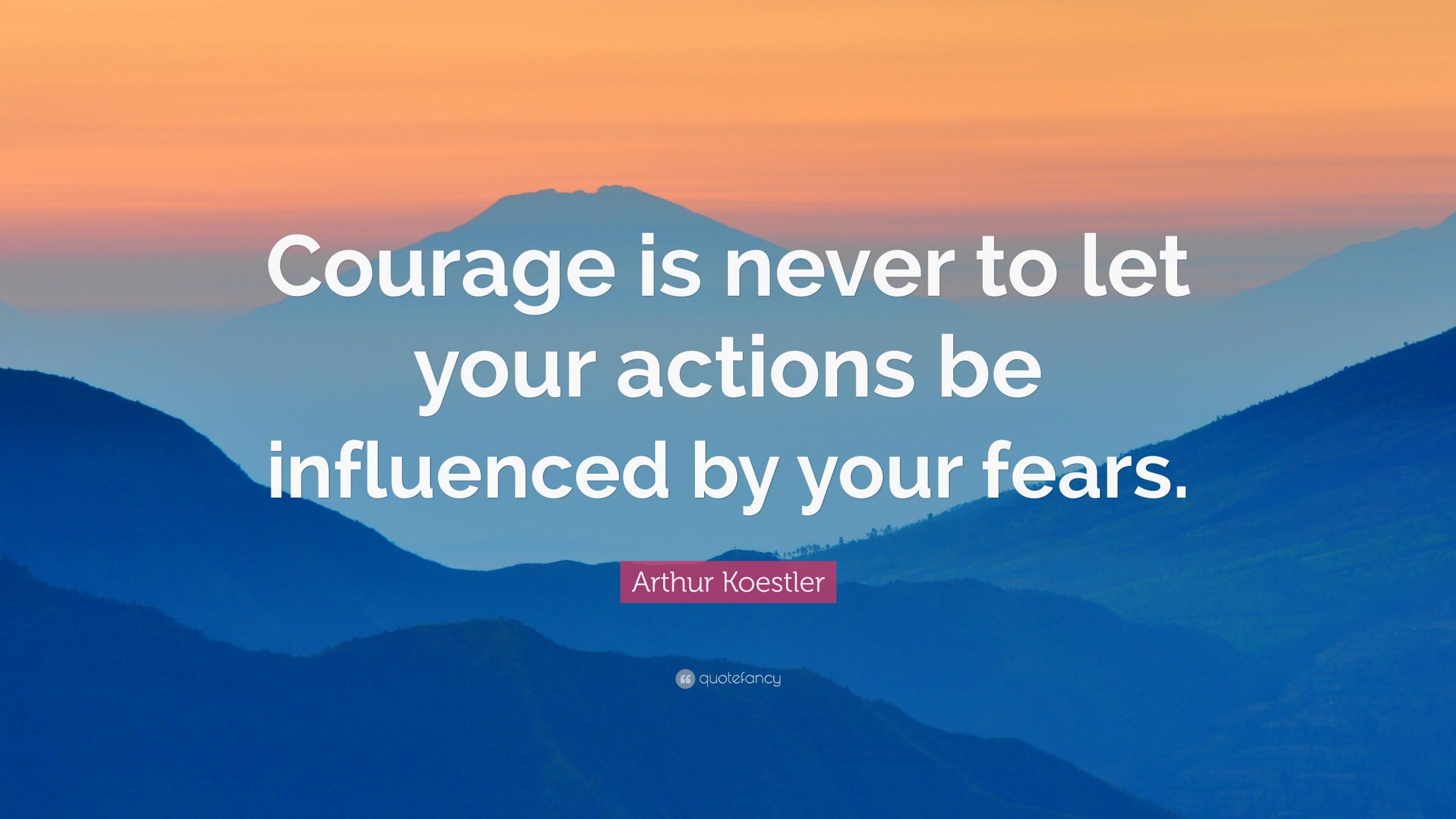
Easy & Cost-Effective for Open-Source Analytics
with Apache Hadoop 3.0

Apache Hadoop
Apache Spark
Apache Kafka
Apache HBase
Apache Hive LLAP
Apache Storm
Machine Learning



Develop Java MapReduce Program for Apache Hadoop [Locally]

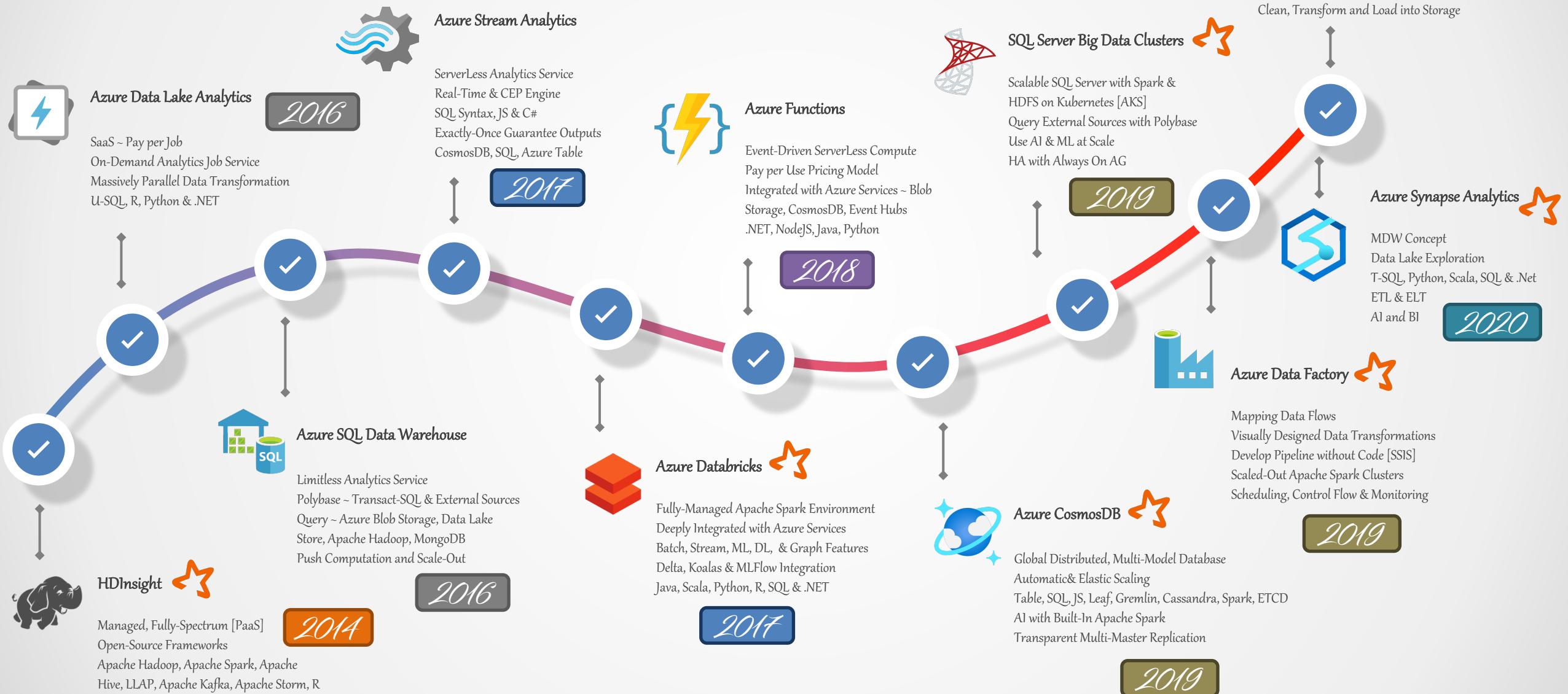


A wide-angle photograph of a mountain range at sunset. The sky is filled with warm, orange and yellow hues near the horizon, which transition into cooler blues and purples higher up. The mountains in the foreground are dark silhouettes, while those in the background are partially illuminated by the setting sun.

Courage is never to let
your actions be
influenced by your fears.

Arthur Koestler

Data Processing Engines [TimeLine] ~ Microsoft Azure



Apache Spark on [Cloud Providers] as a Product



Azure Data Factory Mapping Data Flows

Visually Designed Data Transformations in Azure Data Factory.
Engineers to Develop and Write Logic without Writing Code.
Use Scale-Out Apache Spark Clusters.

Integration with Azure Services for Better Experience. Offers
Latest Data Processing Transformations.

Monitoring, Lineage and Metadata Visualization



AWS Glue

Serverless Data Integration Service for Discover, Prepare and
Combine Data for Analytics, ML & Application Development.
Run Python & Scala Code using Apache Spark Engine

Provides Visual & Code-Based Interfaces for Data Integration
Workflows. Use AWS Glue Data Catalog for Data Governance
and Metadata.

AWS Glue = Data Engineers
AWS Glue DataBrew = Data Scientists
AWS Glue Elastic Views = SQL



Cloud Data Fusion

Fully-Managed, Cloud Native Data Integration at Any Scale.
Visual Point Interface for Code-Free Enablement, ETL & ELT Data
Pipelines.

+150 Pre-Configured Connectors and Transformations
Natively Integrated Best-In-Class Google Cloud Services
End-to-End Data Lineage and RCA
Built with Open-Source CDAP for Pipeline Portability

Uses Apache Spark Cluster for Data Processing ~ Google Cloud
Data Proc

Apache Spark [Fundamentals]



Apache Spark

- Open-Source Distributed Cluster-Computing Framework
- **Implicit Data Parallelism** & Fault Tolerance
- Optimized for Memory Computation
- Written In - Scala
- 100x ~ MapReduce Jobs & 10x – Disk-Based Operations



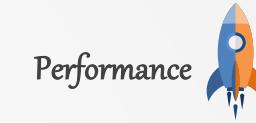
History

- University of California, Berkeley's AMPLab
- Open-Sourced in 2014 – Top-Level Apache Project
- Databricks – New World Record in Large Scale Sorting [2014]
- Ali Ghodsi | Reynold Xin | Matei Zaharia ~ **Databricks**
- 1,000 Contributors in 2015



Key Capabilities

- Unified Stack for Interactive, Streaming & Predictive Analysis
- Batch & Streaming in an Unified Platform
- Designed for Large-Scale Data Processing



Daytona Gray

- 100 TB in **23 Minutes** with 206 EC2 VMs
- 100 TB in **72 Minutes** with 2,100 Nodes



Core APIs



- SQL
- Java
- Scala
- Python
- R

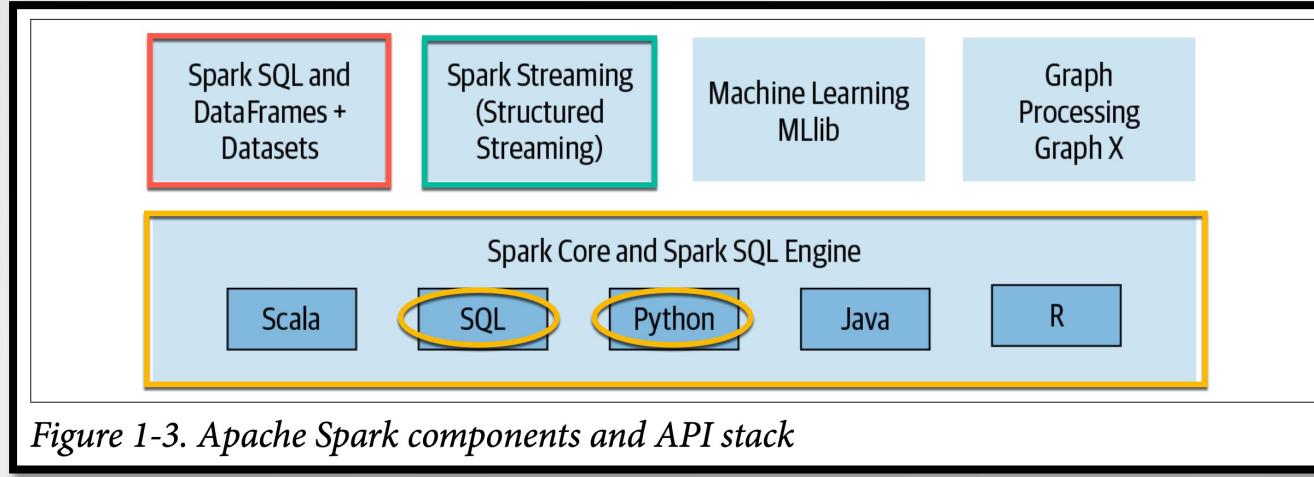


Processing Structures

- RDD – Resilient Distributed DataSet
- Spark Streaming – Processing Data Streams using DStreams
- Spark-SQL, DataSets & DataFrames – Processing Structured Data
- Structured Streaming – Processing Structured Data Streams



Apache Spark [Components]

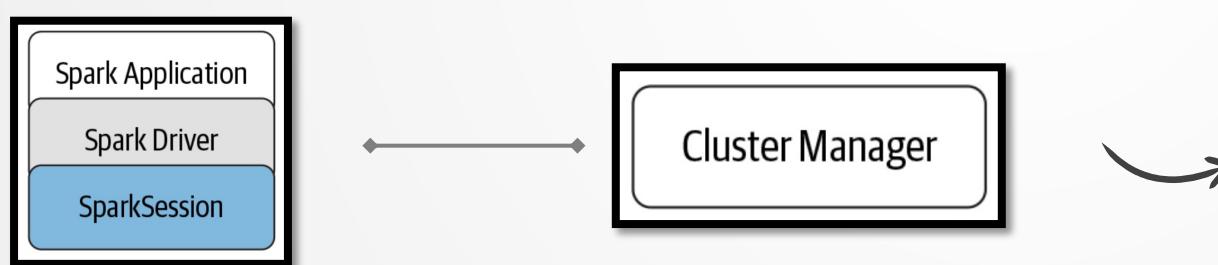


SQL & DataFrames + DataSet

- Read Structured Data
- Formats ~ CSV, Text, JSON, Avro, ORC, Parquet, Delta, Iceberg, Hudi
- Permanent or Temporary Tables In-Memory
- Use SQL Interface for Data Accessing

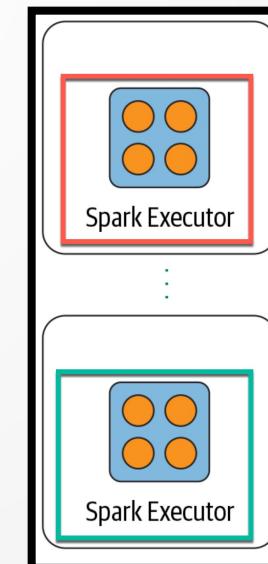
Structured Streaming

- Continuous Streaming Model using Structured Streaming API
- Uses SQL Engine & DataFrames-Based
- Lightweight, Faster and Better Integration



Responsible for Instantiating a Session,
Communicates with Cluster Manager, Request
Resources, Transforms DAG Computation,
Schedule and Distribute Tasks.

Responsible for Managing and Allocating
Resources for Cluster of Nodes.



Runs on Worker Node on Cluster.
Communicate with Driver Program
and Responsible for Executing Tasks.

Apache Spark [Deployment Modes]



Local

Runs on a Single JVM ~ Laptop or Single Node,
Executor and Cluster Manager Runs on Same Host



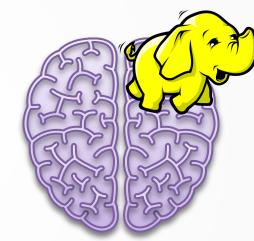
Standalone

Run on Any Node in Cluster. Each Node Launch Own
Executor JVM. Allocated Arbitrarily to Any Host



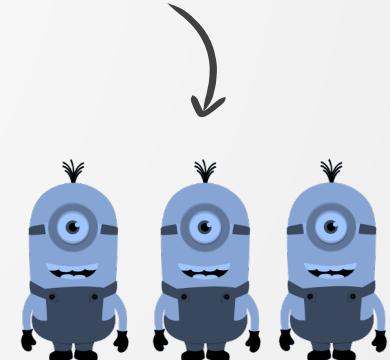
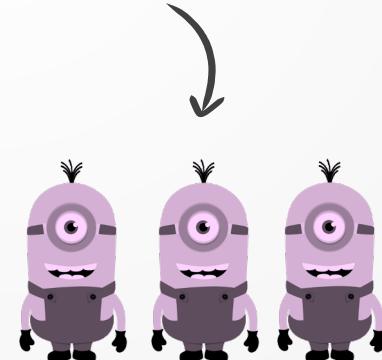
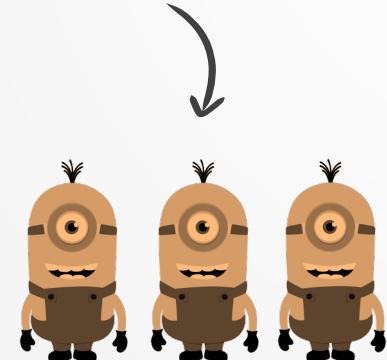
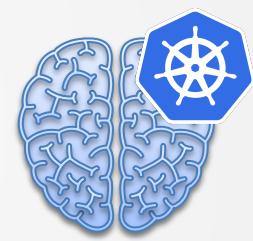
Yarn

Runs with YARN (Cluster) Application Master. Node
Manager & Resource Allocate Containers for
Execution

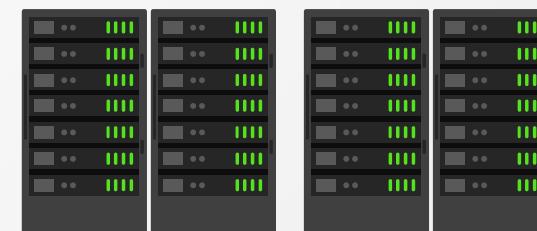
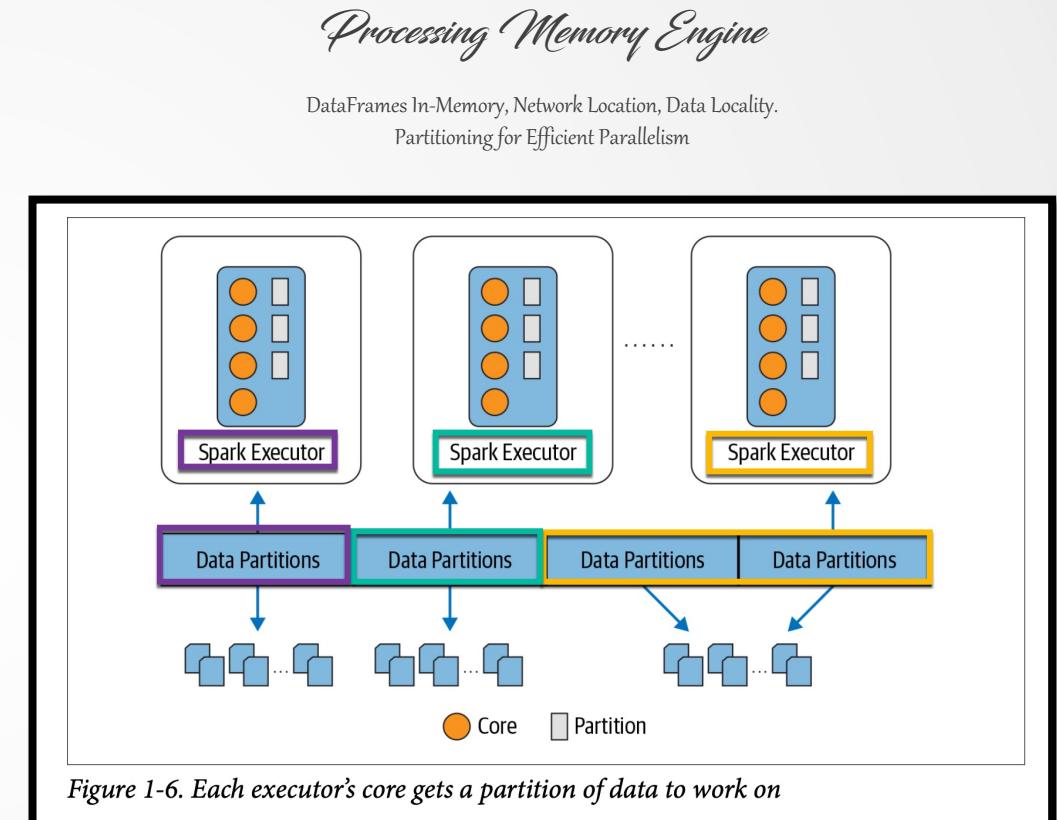
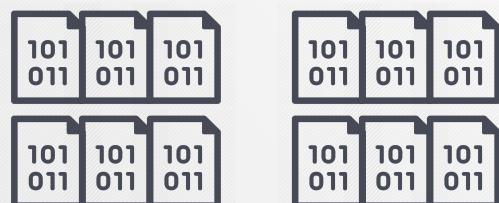
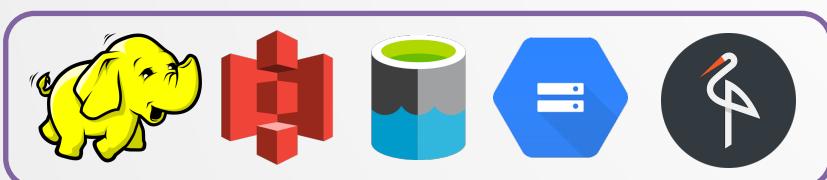
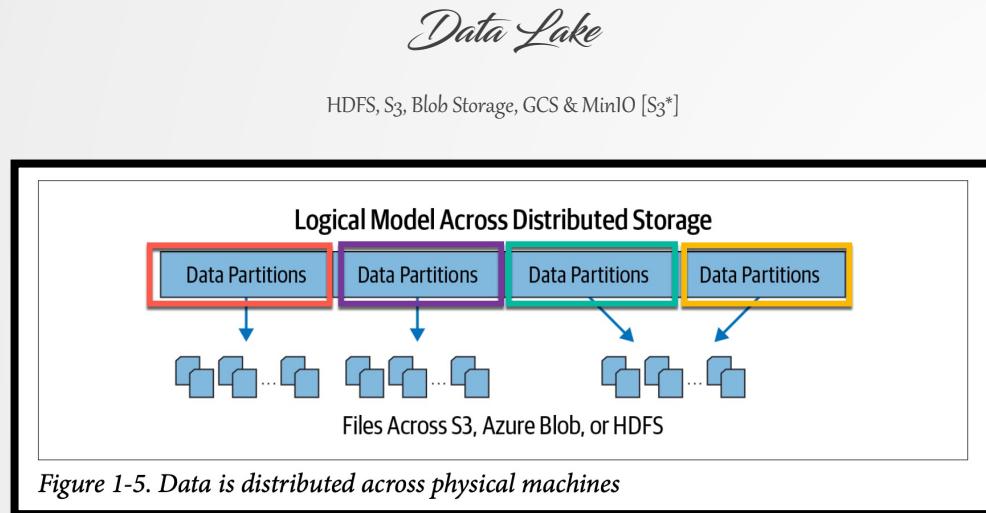


Kubernetes

Runs in a Kubernetes Pod. Each Worker Runs Within
on Pod Context, use Kubernetes Master for Cluster
Management



Apache Spark [Distributed Data & Partitions]



RDD | DataFrame & DataSet [Comparison]



Apache Spark
DataFrame & DataSet API Unified
Apache Spark 2.0

	RDD	DataFrame	DataSet
› Structured & Unstructured	✓	✓	✓
› Java & Scala	✓	✓	✓
› Python & R	✓	✓	✗
› Any Data Source	✓	✗	✓
› Schema Infer	✗	✓	✓
› Optimization Engine	✗	✓	✓
› Fast Aggregation	✗	✓	✓
› In-Memory Serialization	✗	✓	✓

Dataframes vs. Datasets vs. RDDs



DataFrames

- Best Choice in Most Situations ~ 95%
- Provides Query Optimization using Catalyst
- Whole-Stage Code Generation
- Direct Memory Access
- Low GC Overhead
- For Extensibility Not as Developer-Friendly as DataSets



DataSets

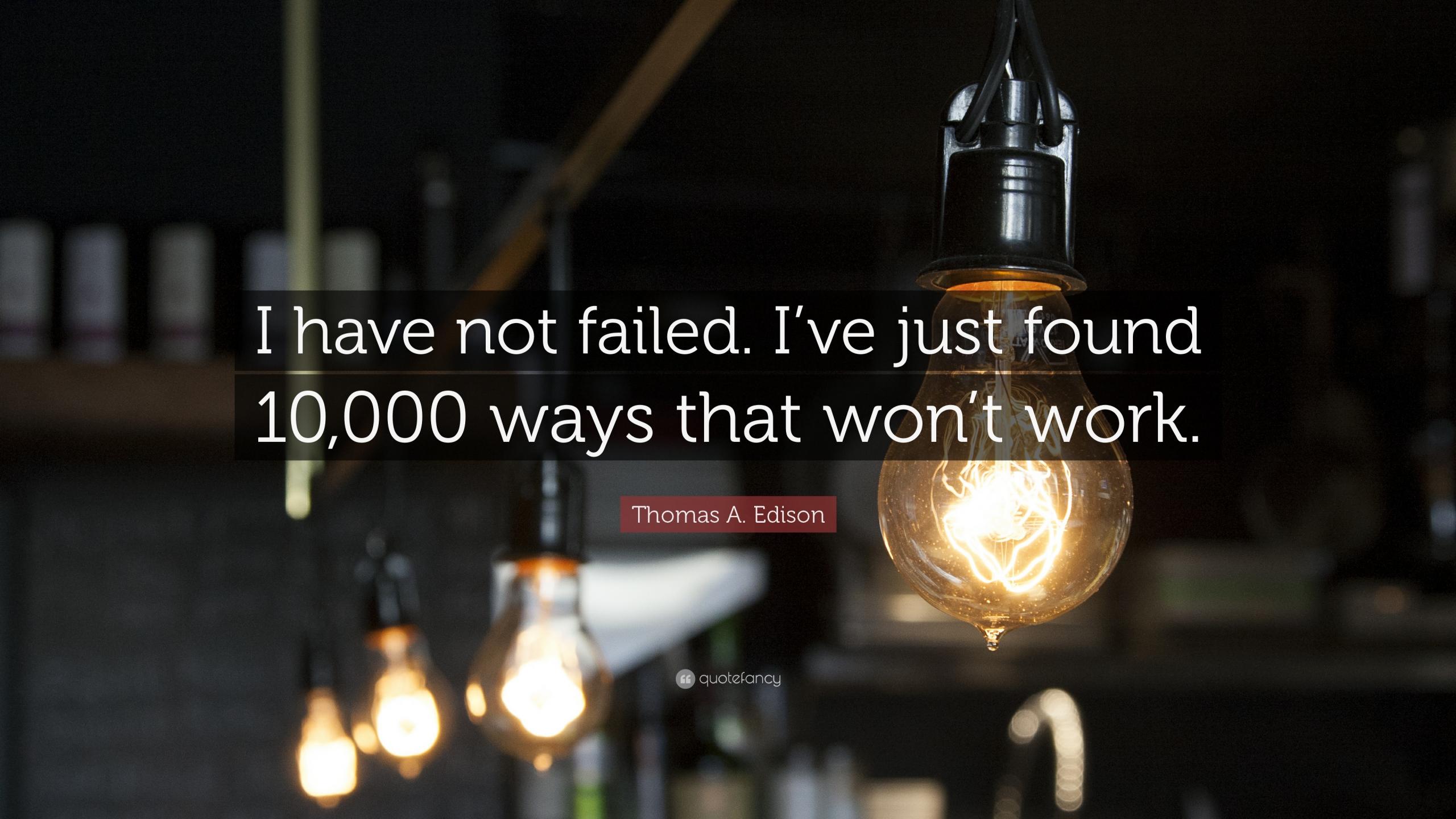
- Good in Complex ETL Pipelines
- Not Good in Aggregations
- Provides Query Optimization using Catalyst
- Developer-Friendly by Providing Domain Object Programming & Compile-Time Checks



Resilient Distributed DataSet [RDD]



- Not Recommended Approach
- Without Query Optimization using Catalyst
- Without Whole-Stage Code Generation
- High GC Overhead
- For Apache Spark 1.X Legacy APIs



I have not failed. I've just found
10,000 ways that won't work.

Thomas A. Edison

PySpark - Concepts & Understanding



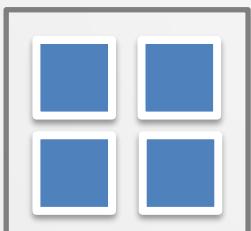
Pandas

Fast, Powerful, Flexible & Easy to Use Open-Source Data Analysis and Manipulation Tool, Built on Top of Python Programming Language. Data Scientists

Additional Information

- Open-Sourced in **2009**
- Fast & Efficient DataFrame Object for Data Manipulation
- Reading & Writing Data ~ In-Memory Data Structures
- Optimized for Performance using Cython & C

Single-Node



Friction



Frictionless



PySpark

Interface for Apache Spark in Python. Allows You to Write Applications using Python APIs. Provides Shell for Interactively Analyzing Your Data in a Distributed Environment

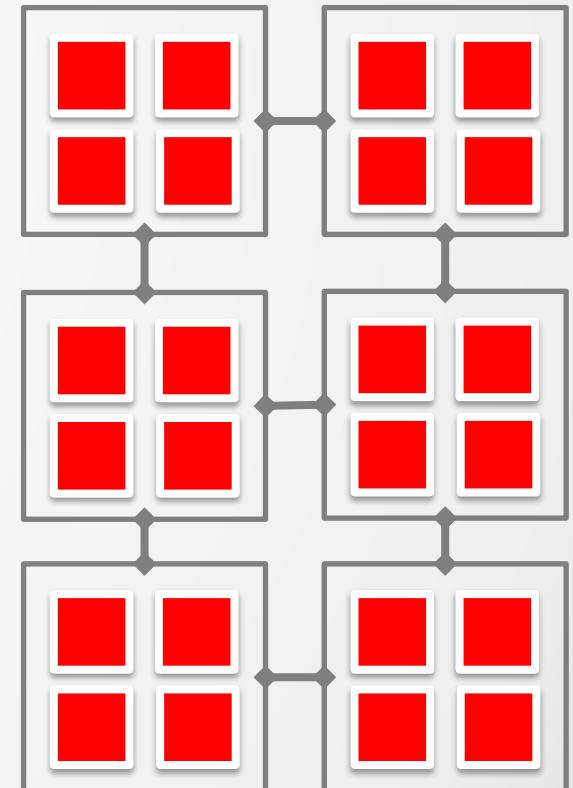
Additional Information

- Open-Sourced in **2016**
- Dependencies ~ Pandas, Numpy, PyArrow
- Scaling Python Application Transparently



Writing Pandas at Scale using Koalas DataFrame API. Without Learning Curve for Data Scientists and Engineers

*Distributed Computing
on Big Data*



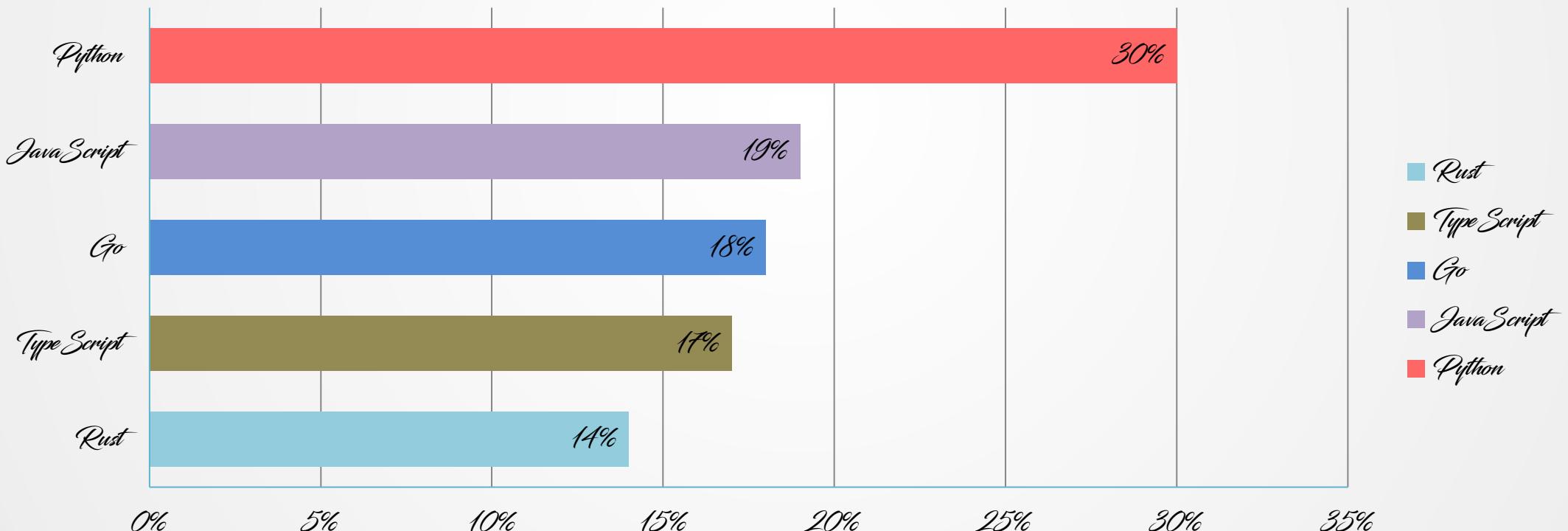
Stack Overflow Developer Survey 2020 ~ Python



Most Loved, Dreaded, and Wanted Languages



if we look at technologies that developers report that they do not use but want to learn, `python` takes the top spot for the *fourth year in a row*. we also see some modest gains in the interest in learning rust.



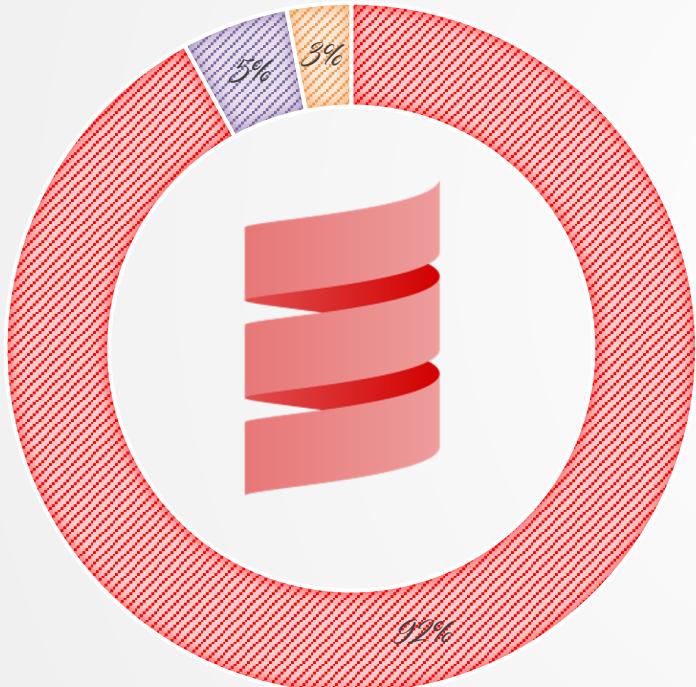
PySpark Utilization Report for Apache Spark Workloads



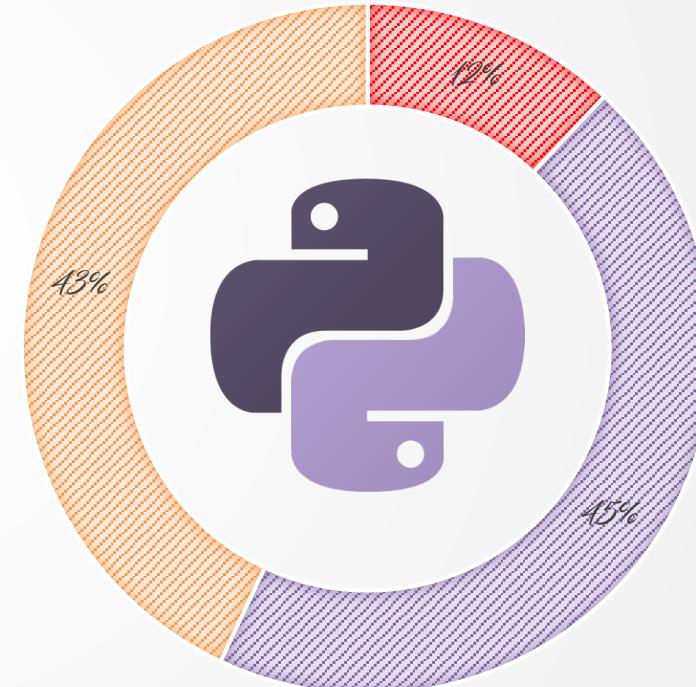
2013

2021

7 Years



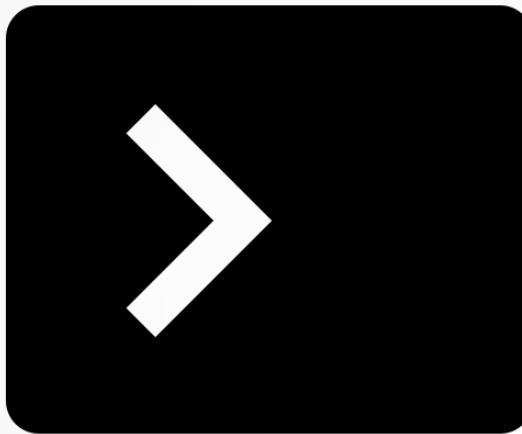
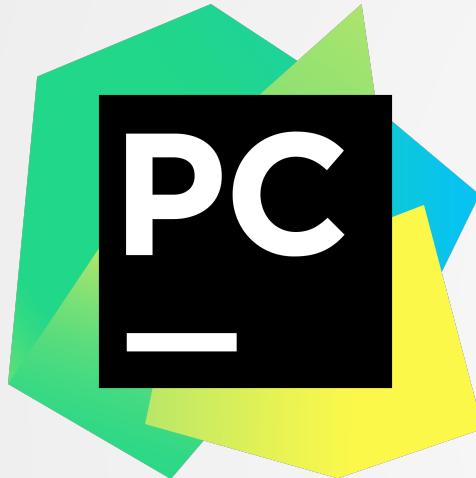
Scala Python SQL



Scala Python SQL

Local Development for Apache Spark

<https://spark.apache.org/docs/latest/api/python>



PyCharm & PySpark

Interface for Apache Spark in Python. Allows You to Write Applications using Python APIs. Provides Shell for Interactively Analyzing Your Data in a Distributed Environment.

Features Supported

- Core, SQL, DataFrame, Streaming, ML

Spark Shell

Simple Way to Learn Apache Spark's APIs.
Powerful Tool to Analyze Data Interactively.

Commands

- pyspark
- sc.version

Spark Submit

Used to Launch Applications on a Cluster. Single Script Used to Submit an Apache Spark Program.

Important Notes

- Class ~ Entry Point for Your Application
- Master ~ URL
- Deploy-Mode ~ Client & Cluster
- Conf ~ Configuration Property
- Jars ~ Code Dependencies
- Arguments ~ Passed to Main Method

```
# Run application locally on 8 cores
./bin/spark-submit \
--class org.apache.spark.examples.SparkPi \
--master local[8] \
/path/to/examples.jar \
100

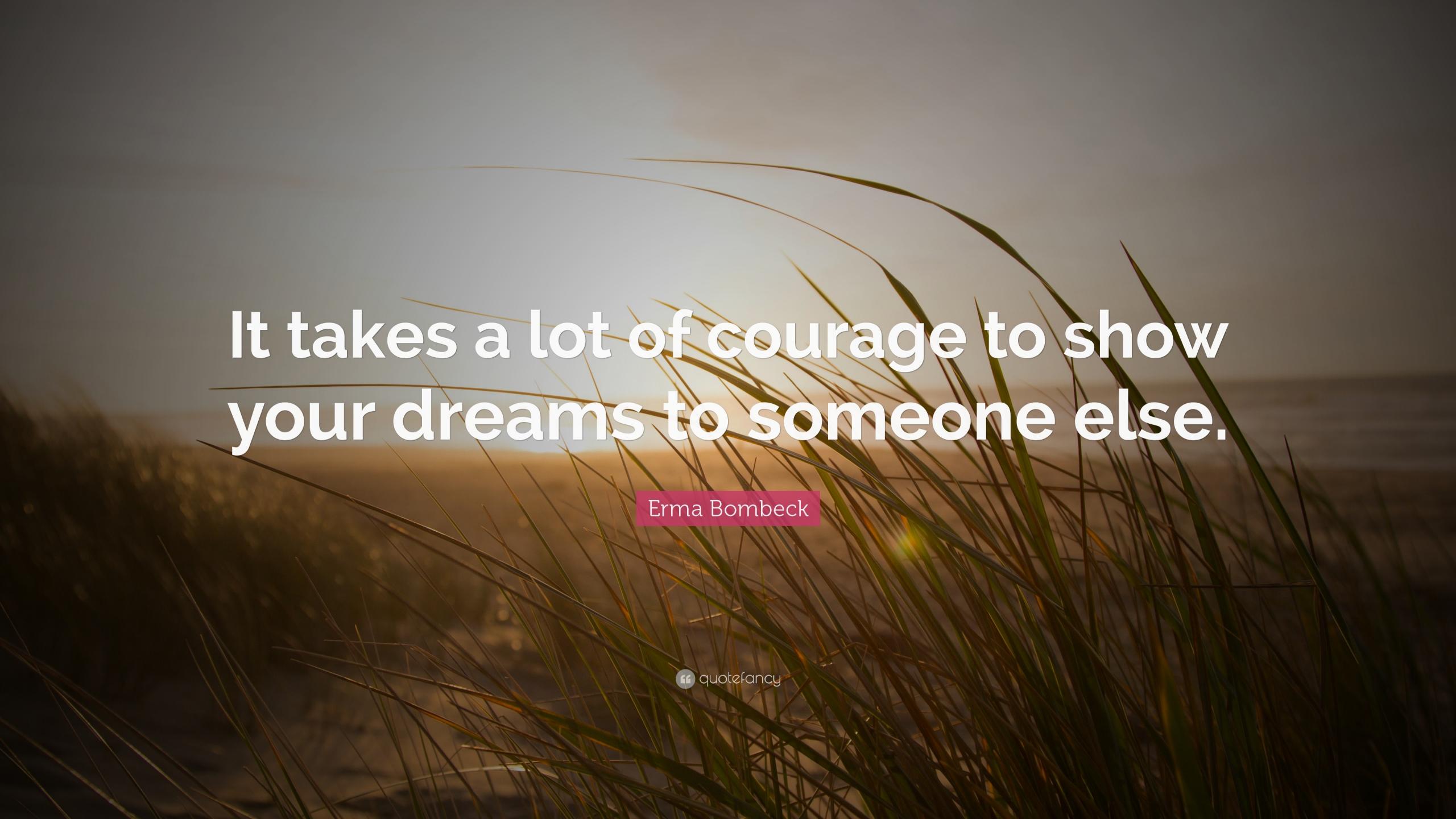
# Run on a Spark standalone cluster in client deploy mode
./bin/spark-submit \
--class org.apache.spark.examples.SparkPi \
--master spark://207.184.161.138:7077 \
--executor-memory 20G \
--total-executor-cores 100 \
/path/to/examples.jar \
1000

# Run on a Spark standalone cluster in cluster deploy mode with supervise
./bin/spark-submit \
--class org.apache.spark.examples.SparkPi \
--master spark://207.184.161.138:7077 \
--deploy-mode cluster \
--supervise \
--executor-memory 20G \
--total-executor-cores 100 \
/path/to/examples.jar \
1000
```



Develop PySpark Program using PyCharm [Local]





**It takes a lot of courage to show
your dreams to someone else.**

Erma Bombeck

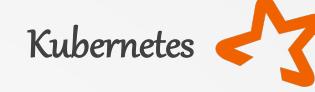
Apache Spark Deployment Options



Azure HDInsight

Parallel Processing Framework with In-Memory Support to Boost Performance of Big Data Analytics

- HDInsight 4.0
- Ubuntu 16.0.4 LTS
- Apache Spark 3.0.0



Kubernetes Operator for Apache Spark.
Running Spark Applications at Scale
with Low and Affordable Cost

- Apache Spark 3.1.1



Spark Pools

Microsoft's Implementation of Apache Spark in Azure Synapse Compatible with Azure Storage

- Apache Spark 3.0.0
- Delta Lake Integration
- .NET for Apache Spark



Amazon EMR

Industry-Leading Cloud Big Data Platform for Processing Vast Amount of Data Using Open-Source Tools

- EMR 6.3.0
- Linux AMI
- Apache Spark 3.1.1



Google DataProc

Fully Managed and Highly Scalable Service for Running Apache Spark & +30 OSS

- BORG = Kubernetes
- Debian, Ubuntu & CentOS
- Apache Spark 3.1.1



Databricks

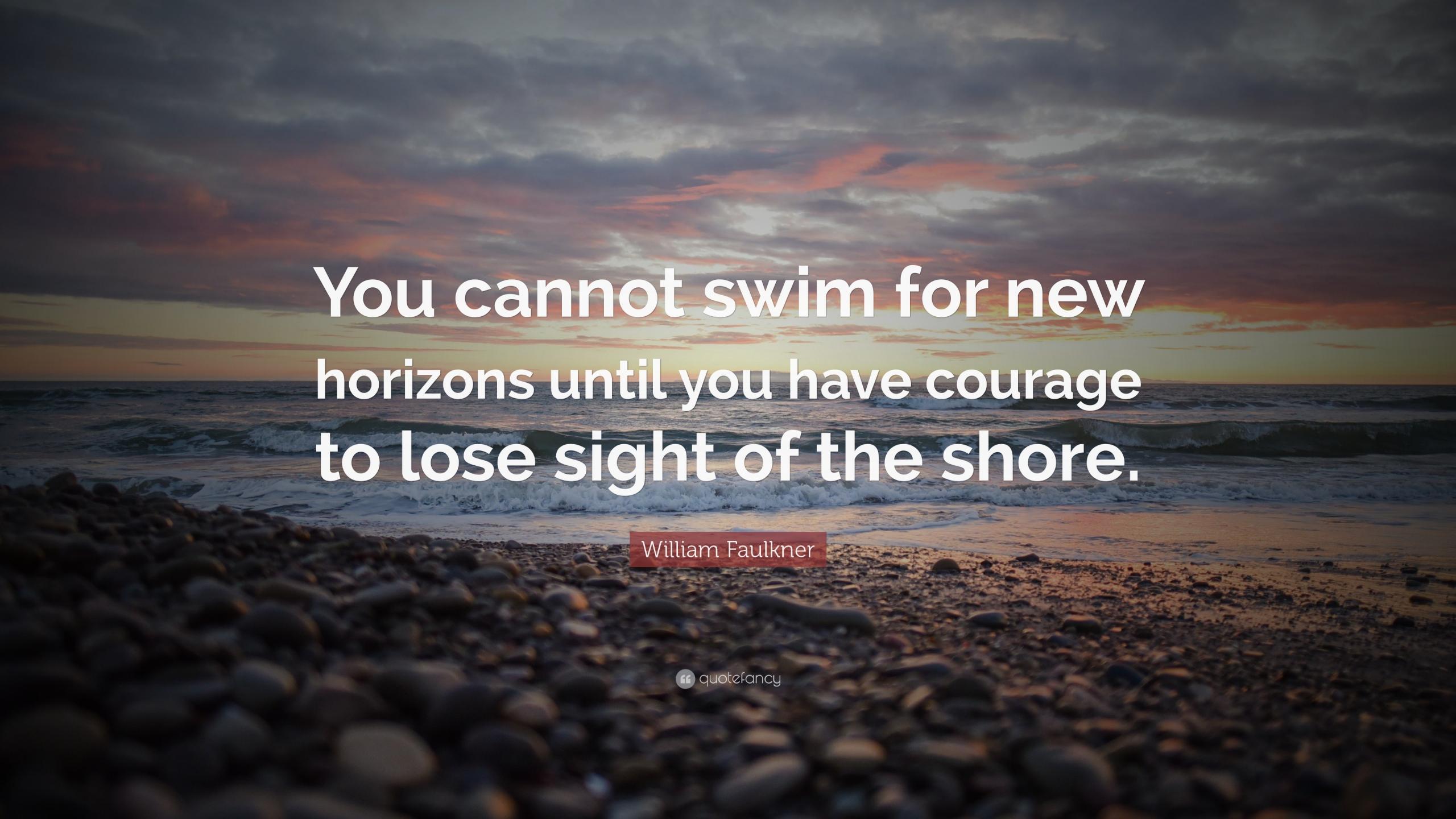


Data + AI Company. Open-Source Apache Spark, Founded in 2013 by Original Creators of Apache Spark

- Apache Spark
- Delta Lake
- MLFlow
- SQL Analytics
- Runtime 8.3
- Apache Spark 3.1.1

Scaling-Out Spark Applications on Cloud Managed Clusters

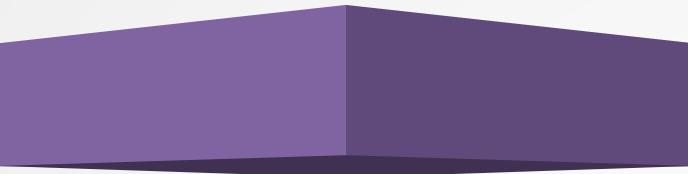


A wide-angle photograph of a sunset over the ocean. The sky is filled with heavy, dark clouds, with patches of bright orange and yellow light breaking through at the horizon. The ocean waves are visible in the foreground, crashing onto a rocky beach. The overall mood is contemplative and inspiring.

You cannot swim for new
horizons until you have courage
to lose sight of the shore.

William Faulkner

Big Data Eras

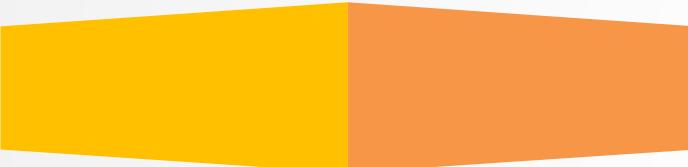


Kubernetes as Cornerstone Solution for Apps & Data ~ [2018]



- Multi-Cloud Strategy First
- Use of Managed Kubernetes Solutions with 99.99%
- Simplify Operations By using Kubernetes as Infrastructure Backbone
- Microservice Driven Approach ~ Containerize Environments
- Cheapest Solution for Application, DataStore & Big Data Solution

Cons ~ Steep Learning Curve, Mindset Change

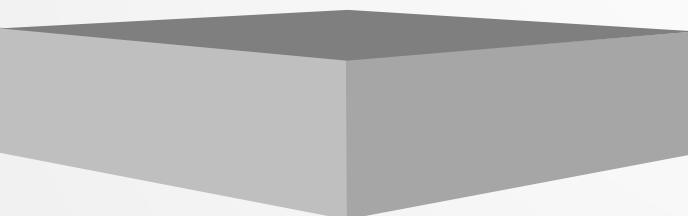


Use of PaaS & SaaS Software's ~ [2014]



- Solution Managed by Cloud Vendor
- Big Data as a Service ~ BDaaS
- Decoupling Computation from Storage
- Pay for Use Only
- Strategy to Reduce Overall Big Data Costs

Cons ~ Promise of Cost Reduction, Myriad of Options



Big Data on Premises using Vendors ~ [2009]



- Cloudera, MapR & HortonWorks
- Based on Open-Source Technologies
- On-Premises Data Centers using Physical Hardware
- Expensive & Dependency Wise
- Large Operations Team ~ Manage Clusters

Cons ~ Expensive, High-Level of Complexity, Burdensome

Virtual Machines [VMs] vs. Containers



Virtual Machines [VM]

- Abstraction of Physical Hardware
- Hypervisor ~ Multiple VMs in a Single Machine
- Machines ~ Full Copy of an Operating System [Binaries & Libraries]
- GB in Size & Slow to Boot

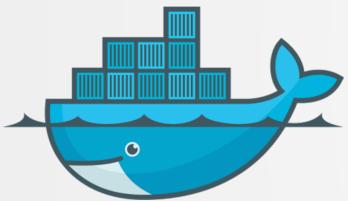
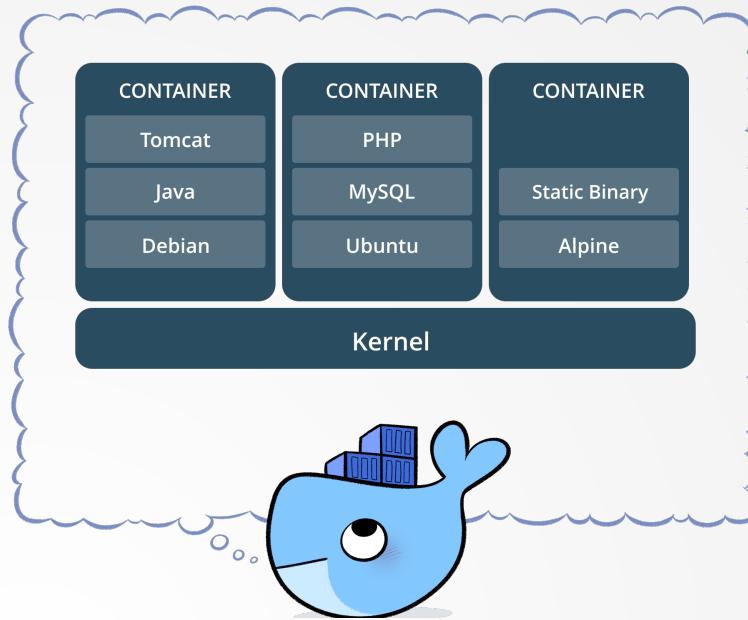


Containers

- Abstraction of an Application [App]
- Packages Code & Dependencies
- Multiple Containers ~ Same Machine & Share OS Kernel
- MBs in Size & Fast Boot



Container [Docker] & Kubernetes



Container Orchestrator

- Portable, Extensible, OSS Platform
- Manage Containerized Workloads & Services



Use-Cases



- Micro-Services
- Stateless & Stateful Applications



Open-Source

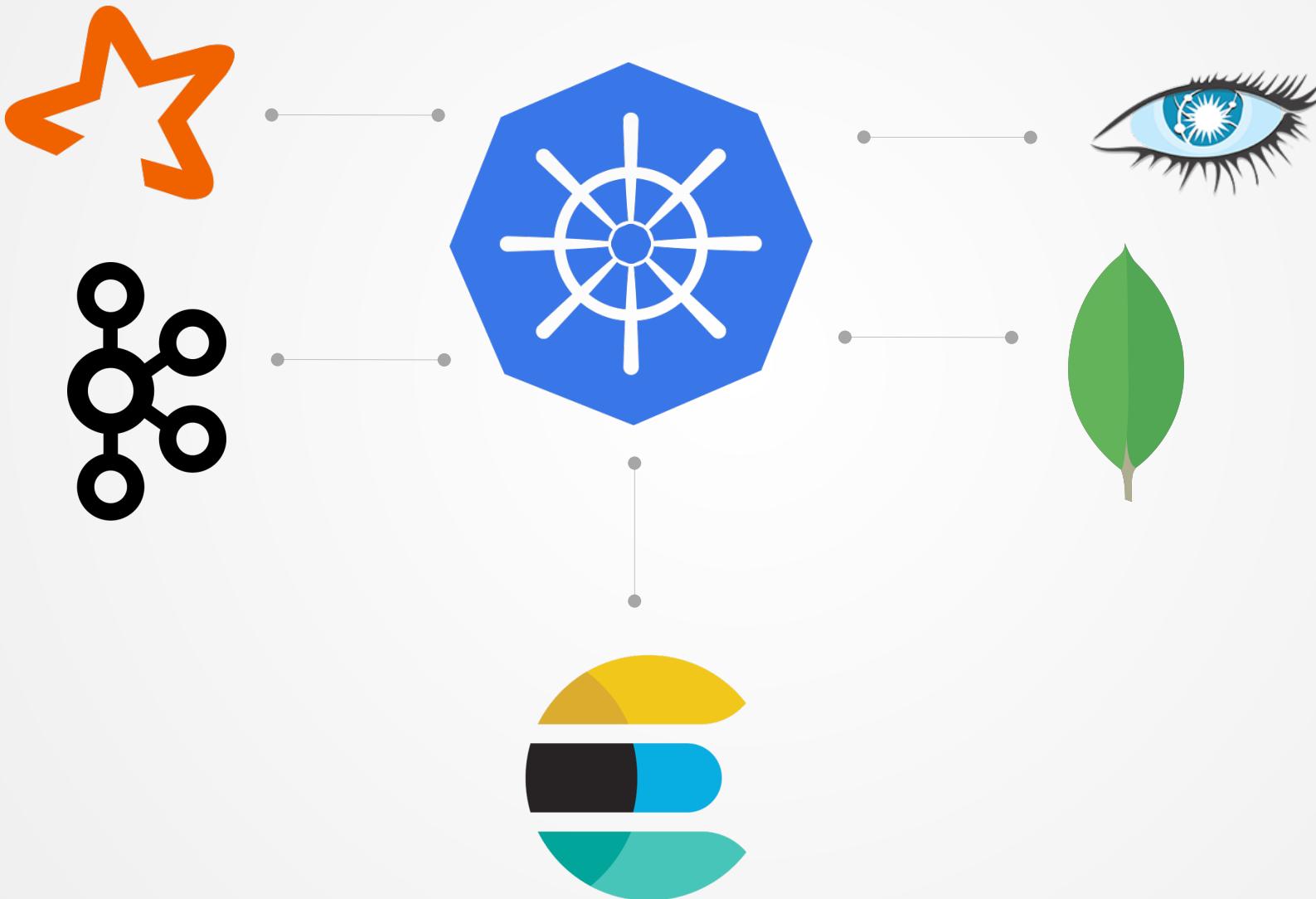
- Open-Sourced in 2014 by Google
- Maintained by Cloud Native Computing Foundation



Motivators

- Scale
- Portability
- Agility
- Cost Savings
- Community

Kubernetes & Big Data



Kubernetes & Cloud Computing



Open-Source Platform [OSS]

- Kubernetes [K8S]

Microsoft Azure

- AKS – Azure Kubernetes Service

Google Cloud Platform [GCP]

- GKE – Google Kubernetes Engine

Amazon Web Services [AWS]

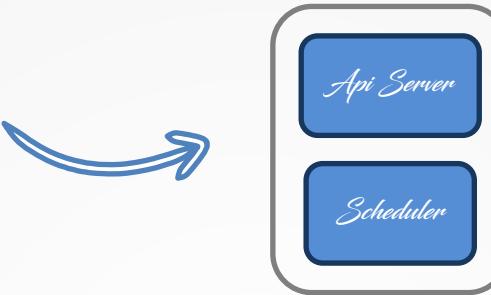
- EKS – Elastic Kubernetes Service

Spark-on-Kubernetes ~ Operator



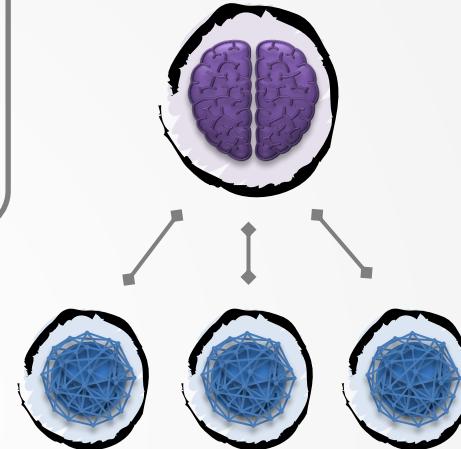
Spark-Submit

Submit an Apache Spark Program
and Launches on Cluster



Spark Driver

Creates Spark Context,
Session, & Executes User Code



Spark Executor

Executes Individual Tasks &
Return Result Set to Driver



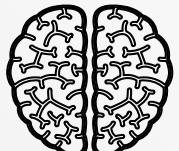
Spark-on-K8s-Operator

Kubernetes Operator ~ Method of Packaging, Deploying and
Managing an Application. Extends Functionality to Create,
Configure, and Manage Instances of Complex Applications

Spark-on-K8s-Operator ~ Aims to Specifying and Running
Apache Spark Applications as Easy and Idiomatic as
Running Kubernetes Workloads Instances

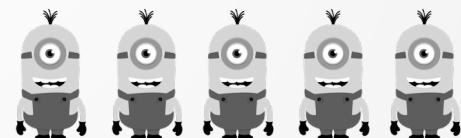


`spark-pi.yaml`



Spark Operator

- Controllers
- Submission Runner
- Pod Monitor
- Mutating Admission



Deployment Mode

Runs in a Kubernetes Pod. Each Worker
Runs Within on Pod Context, use
Kubernetes Master for Cluster Management



Big Data on Kubernetes ~ Spark-on-Kubernetes [Operator]





ONE WAY
SOLUTION