



# Introduction to Data Engineering



## Content

- **What is Data Engineering**
- **Best Practices for ETL/ELT**
- **The Scopes of Data Engineering**
  - **Data Extraction and Loading(Ingestion)**
  - **Data Cleansing**
  - **Data Transformation/Aggregation**
- **Data Engineering Challenges**

---

# What is Data Engineering

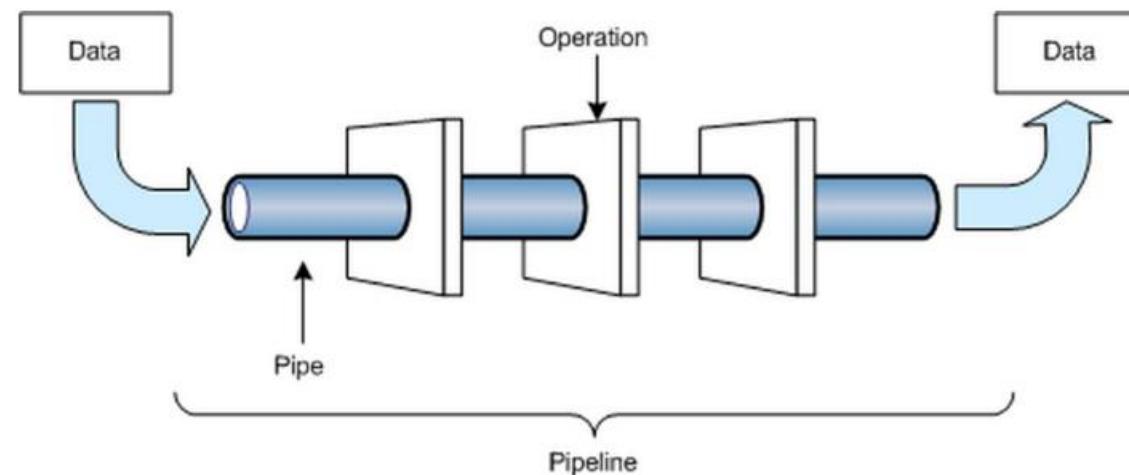
---

## What is Data Engineering?

Process of collecting and transforming

**RAW DATA** into **AGGREGATED DATA**

with formats that data scientists can use



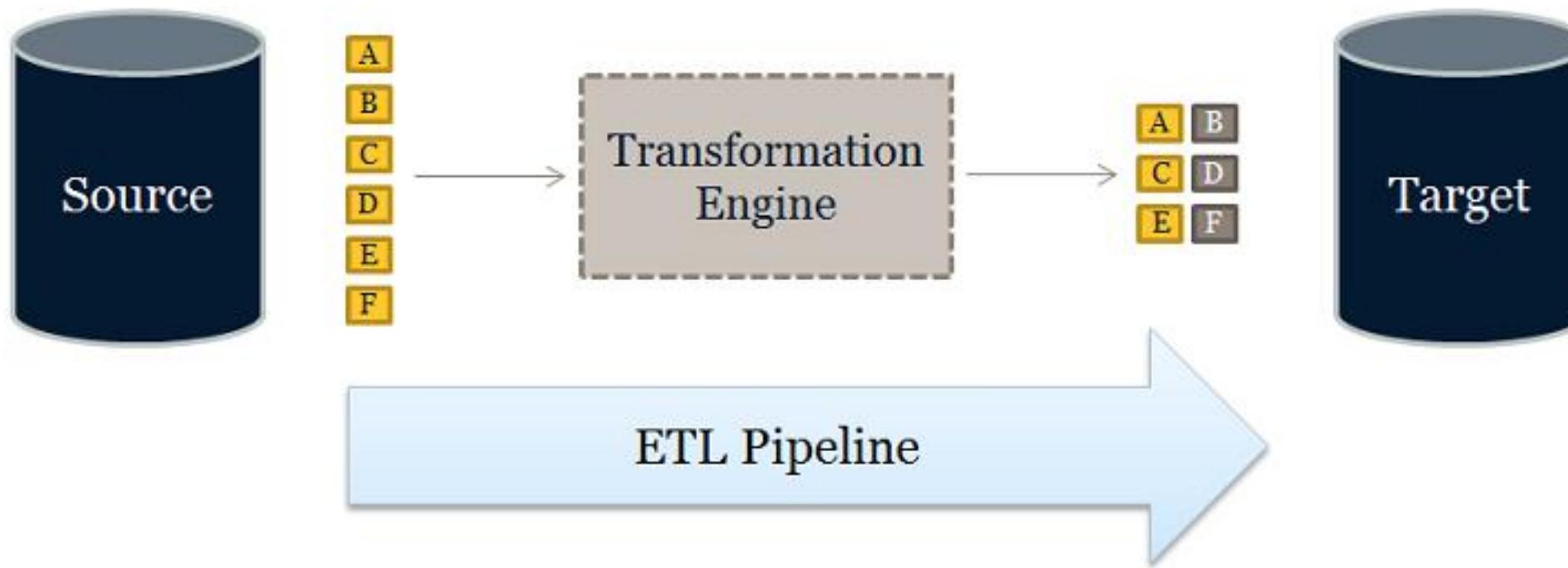
## What is Data Engineering?

**RAW DATA** – Operational, sensor, social media, logs, etc.

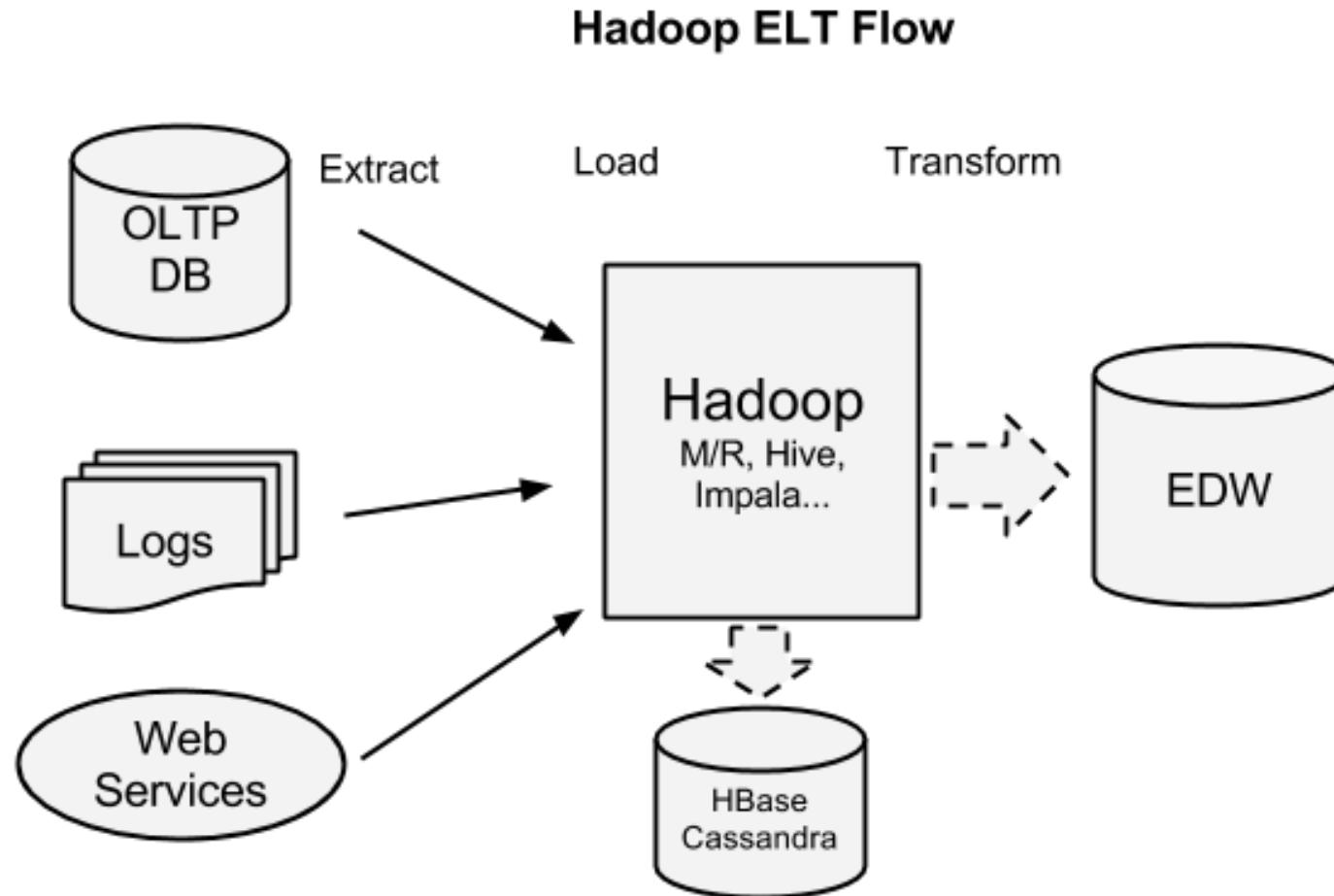
**AGGREGATED DATA** – Data presented

in a summarized format with statistical  
methods

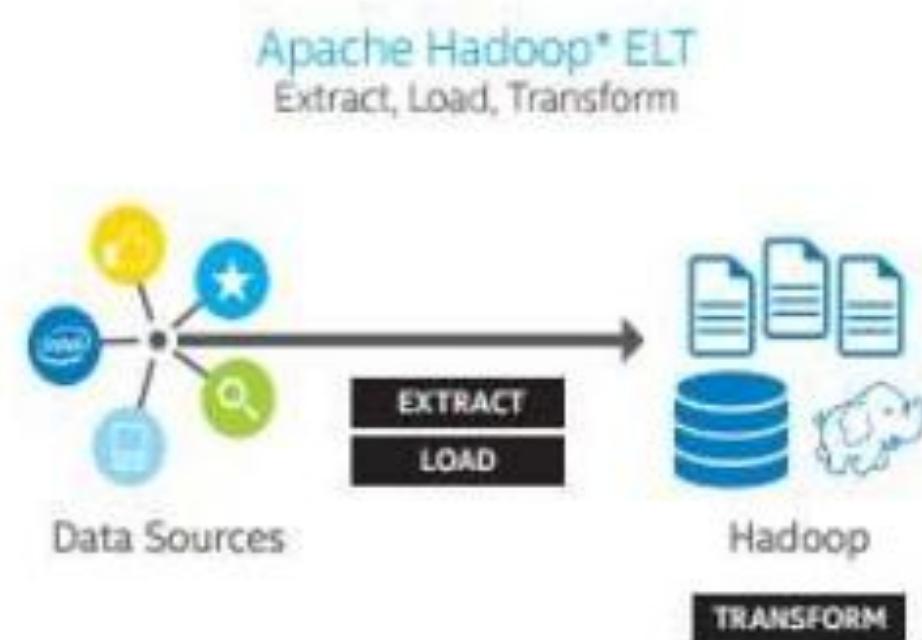
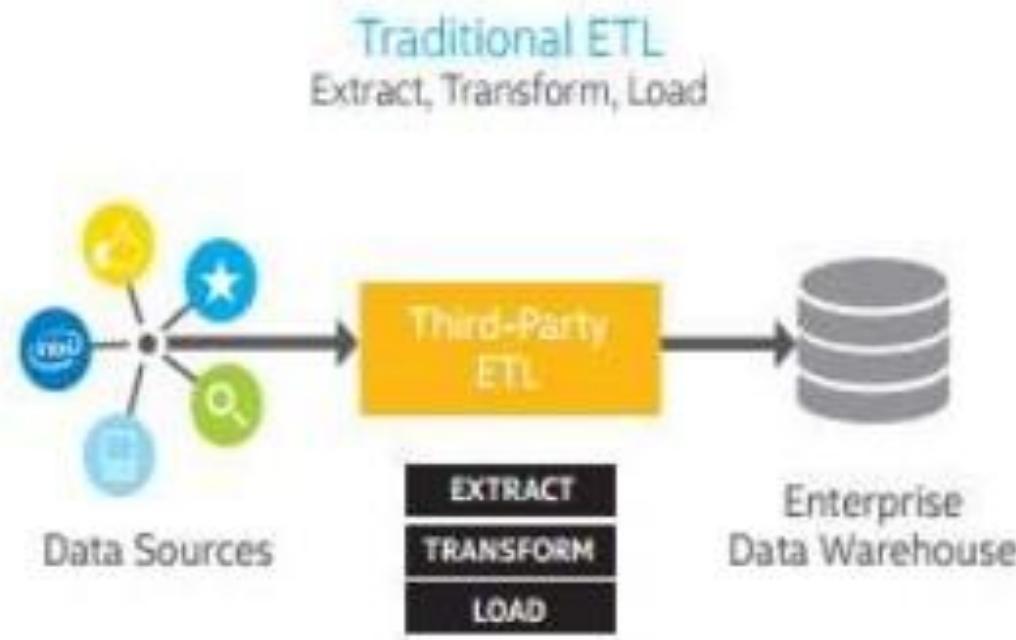
## Data Pipeline - ETL or ELT?



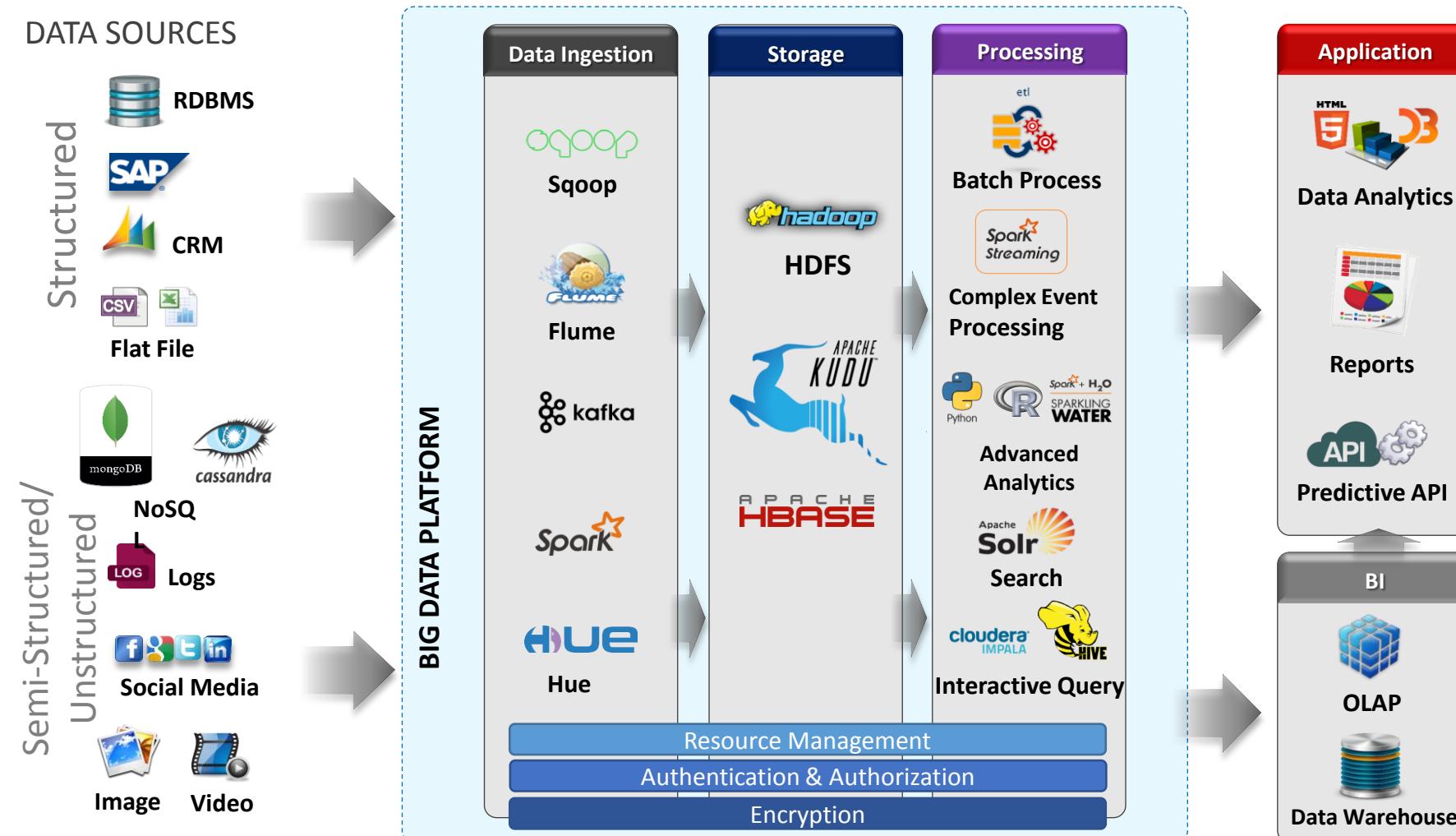
## Data Pipeline - ETL or ELT?



## Data Pipeline - ETL or ELT?



# Big Data Analytics Platform





# Best Practices for ETL/ELT



## Best Practices for ETL/ELT

- In Big Data Platform(Hadoop)
  - ELT is more popular than ETL because Data is huge
  - Transformation is mostly done on Big Data Platform
- Data **will not be checked for integrity during ingestion unlike RDBMS**
- Data **will only be checked when there is a query to process the data**

## Best Practices in ETL/ELT

### **Logging:**

A proper **logging strategy** is key to the success of any ETL architecture.

## Best Practices in ETL/ELT

### Auditing:

A load without errors is **not necessarily a successful load**. A well-designed process will not only check for errors but also **support auditing** of row counts, financial amounts, and other metrics.

## Best Practices in ETL/ELT

### *Data Lineage.*

Understanding where **data originated from, when it was loaded, and how it was transformed** is essential for the integrity of the downstream data and the process that moves it there.

## Best Practices in ETL/ELT

### Modularity.

Creating **reusable code structures** is important in most development realms, and even more so in ELT processes.

ELT modularization helps **avoid writing the same difficult code over and over**, and **reduces the total effort** required to maintain the ELT architecture.

## Best Practices in ETL/ELT

### Atomicity:

How big should each ETL process be?

### Error Handling:

What happens when things go wrong?

## Best Practices in ETL/ELT

### *Managing Bad Data:*

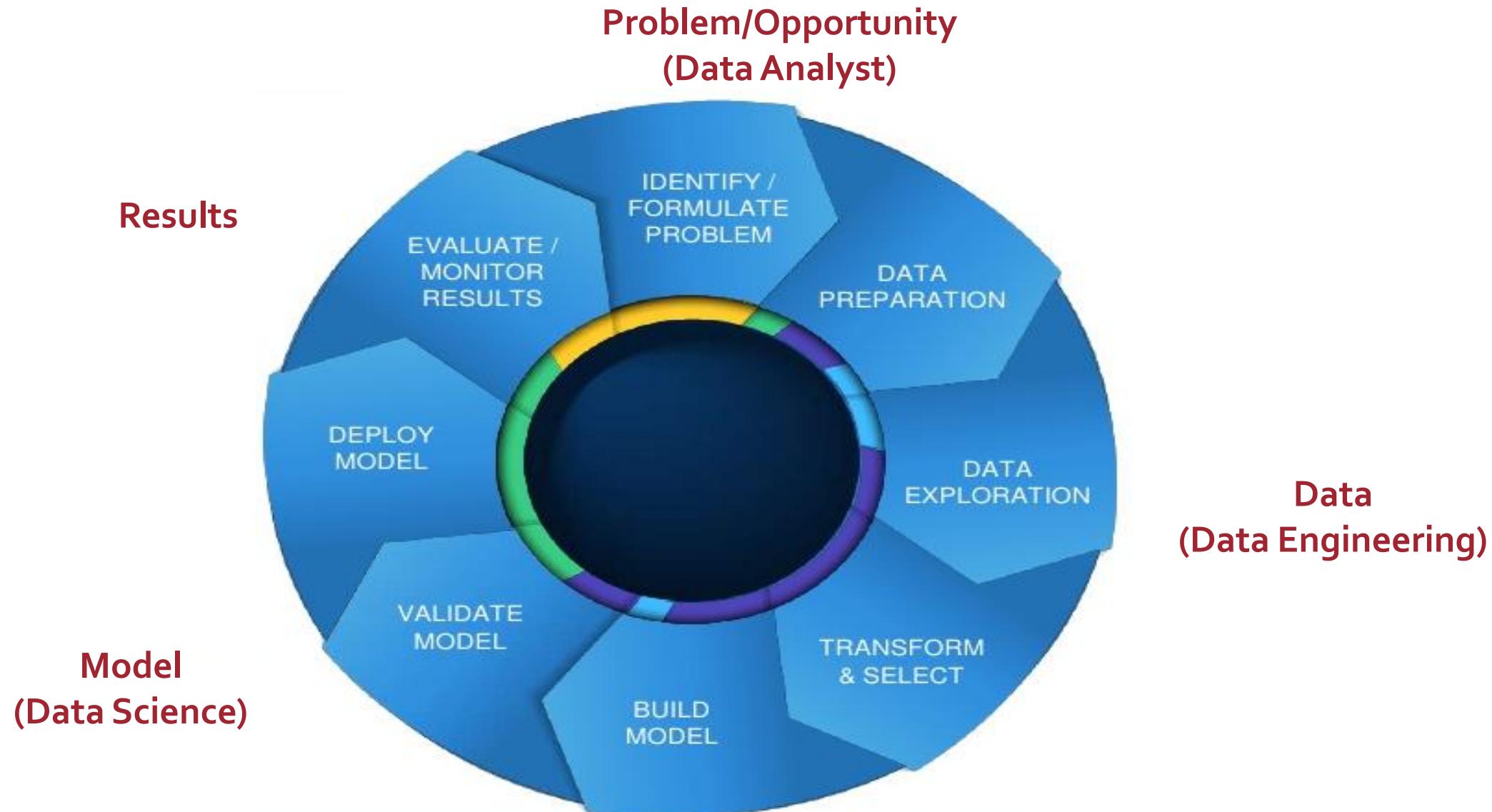
When suspect data is discovered, there needs to be a **system for cleansing or otherwise managing non-conforming rows of data**

---

# The Scopes and Challenges of Data Engineering

---

# Big Data Analytics Lifecycle



# Comparison of Scopes

## Data Scientist

- Building Models
- Validation/Testing
- Algorithms
- Continuous Improvement
- Knowledge of :
  - Statistics
  - Linear Algebra
  - Machine Learning
  - R,Matlab etc.

## Data Engineer

- Data Pipelines
- Manage Platforms
- Productionalize Algorithms
- Agile Development
- Knowledge of :
  - Platforms
  - Algorithms
  - Java, C++ etc.
  - Scripting languages like python

## Data Analyst

- Deep Domain Knowledge
- Report Generation
- Data Exploration
- Hypotheses Testing
- Pattern Discovery
- Correlations
- Serendipitous Discovery

## Before the explosion of Big Data!!

- Life was simple ... well mostly
- The ETL engineers managed data pipelines
- The Data Scientists (they weren't called that, btw, they were mostly Statisticians who programmed in SAS, SPSS or S) did the analysis
- Data Warehouses, Data marts and OLAP cubes were the platforms
- Data Analysts mostly generated reports but they were proficient in SQL, Excel, Pivot Tables etc.
- Data Architects ... well, they architected 😊
- They managed :
  - Data models
  - Star Schemas
  - Data Governance
  - Master Data Management (MDM)
  - Data Security
- For the most part, they had to coax different groups to share data

## Big Data Era – Huge Changes!!

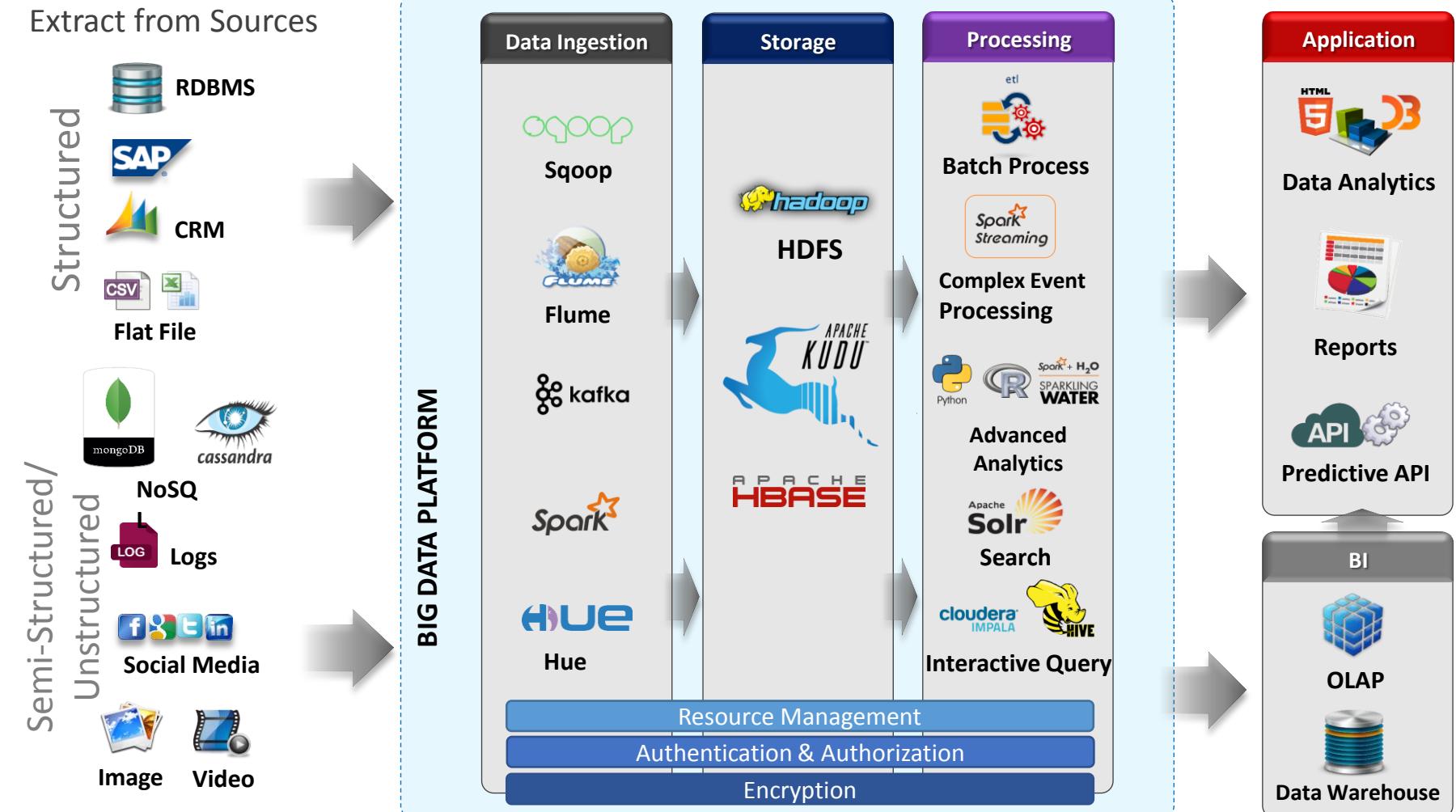
- Life ... got interesting
- Huge data volumes – ETL became a problem
- Traditional Statistical tools couldn't handle the volume
- Data Warehouses, Data marts and OLAP cubes not primary analytical means – “in situ” analysis preferred i.e. no moving data to an analytics platform
- Data Analysts still on point for reports but now they no longer had SQL interfaces (thanks to NoSQL and Map Reduce)
- Data Architects ... well, they still need to architect ☺
- Still need :
  - Data models
  - Data Governance
  - Data Security
- For the most part, they had to coax different groups to share data
- They have to do all of this when the technology is rapidly evolving

## Scope of Data Engineering

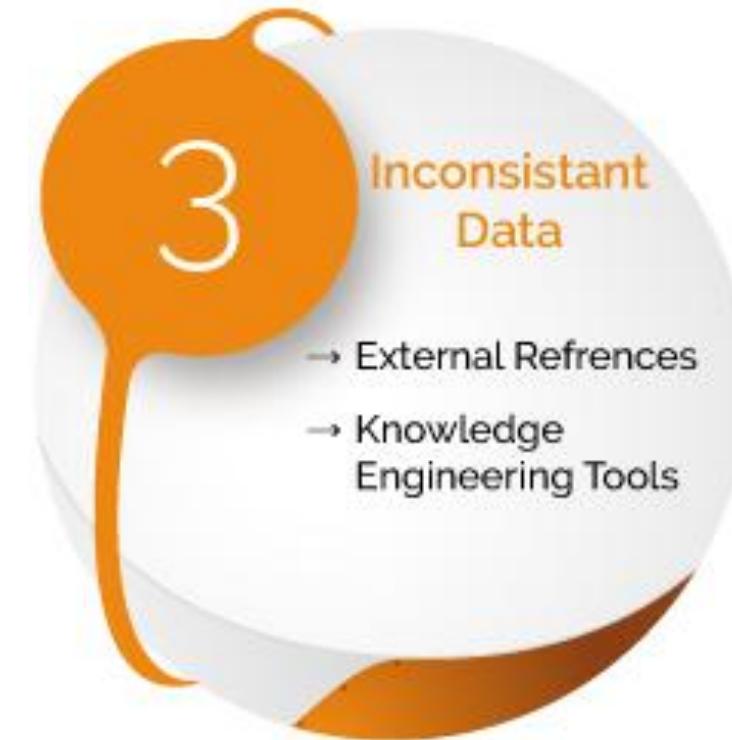
- **Data Preparation**
  - **Data Extraction, Loading, Cleansing**
- **Data Exploration**
  - **Data Visualization**
- **Data Transformation and Feature Selection**
  - **Data Aggregation to find useful features**

## Scope of Data Engineering – Data Preparation

- **Data Extraction**
- **Data Loading**
- **into Big Data platform**
- **Most of the time this is solely ingestion, no checking is done**



## Scope of Data Engineering – Data Cleansing



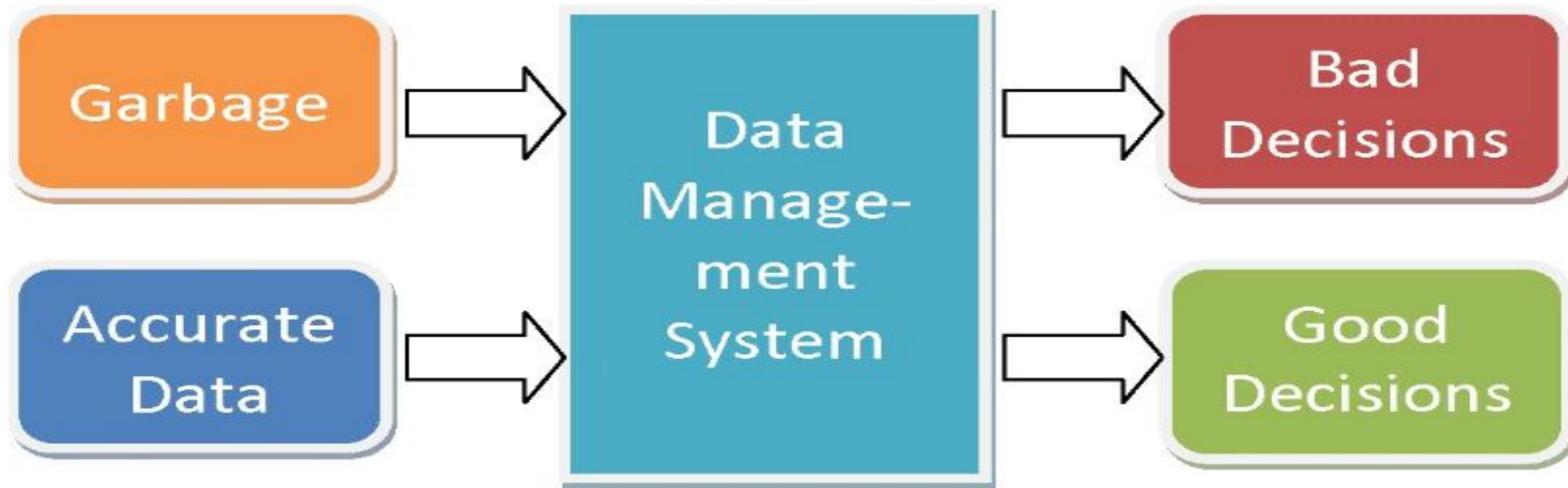
## Missing Data

Day	Aircond sales (unit)	Temperature °C
1	54	36
2	53	35.5
3	49	33
4	45	30
5		28
6	54	36
7	51	
8	53	35.5
9	43	29
10	46	31.1
11		32.5
12	20	33.6
13	49	32.5
14	52	
15	53	35.4
16	43	29.1
17		28.1
18	44	29.5
19	42	
20	39	26.2

Cleansing of data →

Day	Aircond sales (unit)	Temperature °C
1	54	36
2	53	35.5
3	49	33
4	45	30
6	54	36
8	53	35.5
9	43	29
10	46	31.1
12	20	33.6
13	49	32.5
15	53	35.4
16	43	29.1
18	44	29.5
20	39	26.2

## Data Cleansing



Beware of dirty data !!!

## Data Cleansing



## Scope of Data Engineering – Data Exploration

- Perform Descriptive Analysis
  - Explore common relevant variables, attributes and records
  - Explore possible insights from data to create more representative variables
  - Splitting and Joining variables

## Relevant data

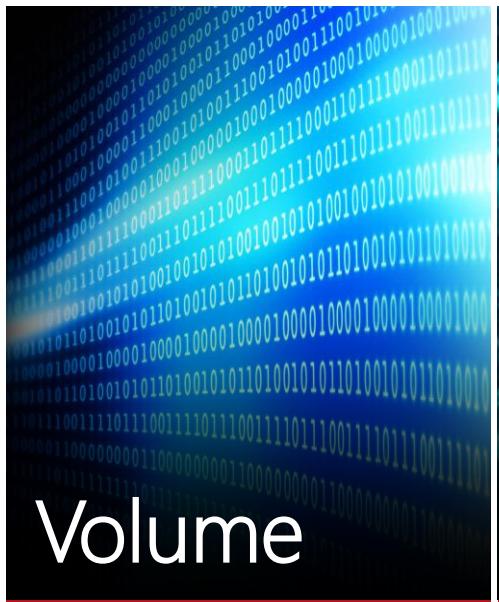
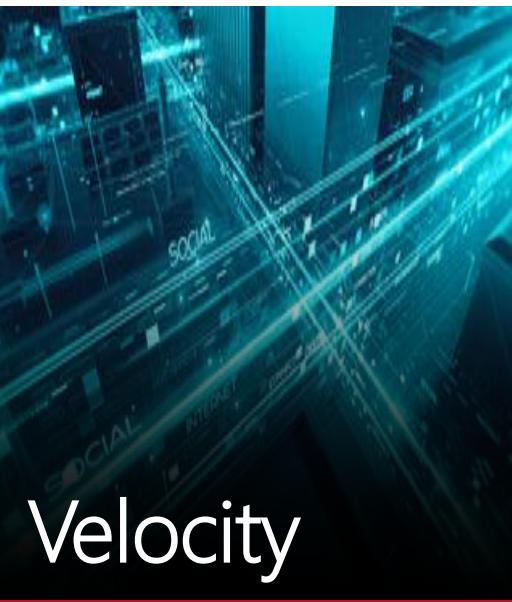
Day	Number of followers for LCW on fb	Temperature °C	Price of petrol per liter (RM)
1	674215	28	2.1
2	675842	27.5	2.1
3	679512	29	2.1
4	685214	28.5	2.1
5	692145	27.3	2.1
6	695412	27.9	2.1
7	701245	28.1	2.1
8	721455	29.4	2.01
9	742154	30.1	2.01
10	754521	31.1	2.01
11	762513	28.5	2.01
12	798546	29.4	2.01
13	814562	28.1	2.01
14	825462	27.6	2.01
15	852164	26.8	2.13
16	895412	27.9	2.13
17	921456	28.1	2.13
18	995123	29.5	2.13
19	1101238	30.1	2.13
20	1181074	31.5	2.13

Day	Aircond sales (unit)	Temperature °C
1	54	36
2	53	35.5
3	49	33
4	45	30
5	42	28
6	54	36
7	51	34
8	53	35.5
9	43	29
10	46	31.1
11	48	32.5
12	20	33.6
13	49	32.5
14	52	34.8
15	53	35.4
16	43	29.1
17	42	28.1
18	44	29.5
19	42	28.2
20	39	26.2

## Scope of Data Engineering – Data Transformation and Feature Selection

- **Data Transformation and feature selection**
  - Transforming data into more useful information by clustering, classifying, categorizing data, etc.
  - Remove redundant or variables that contribute little information or with high co-relationship with others. For example, age and year of birth, city and postcode,
  - Shortlist features that are important and significant for analytics
  - Example of features: min/max/avg/med/stddev/var of daily/monthly/yearly by country/region/state/city/street if weather.....

# Challenges for Data Engineers – from the 5Vs perspective

 Volume	 Velocity	 Variety	 Veracity	 Value
<b>Data At Rest</b>  Terabytes to Exabytes of existing data to process	<b>Data in Motion</b>  Streaming data, requiring milliseconds to seconds to respond	<b>Data in Many Forms</b>  Structured, Unstructured, Text, Multimedia etc.	<b>Data in Doubt</b>  Uncertainty due to data inconsistency & incompleteness, ambiguities, latency, deception, model approximations	<b>Data into Money</b>  Business model can be associated to the data

## What is Data Engineering?

### **Special Characteristics of BIG DATA – Data is not STATIC!**

- Data sizes keep growing
- Data sources keep increasing
- Data types keep changing
- Data values keep rising (insights are unlimited, what is limited is our imagination!)

## In-class Activity

- Work in a group of 4-5
- Discuss the followings:
  - Choose a set of data from your organization or that you are working on
  - Explain how you perform ETL/ELT on them
    - Discuss if ETL is possible or ELT would be a better option? WHY?
  - Tell us about your current platform and practices
  - Describe the challenges based on the 4Vs(Volume, Velocity, Veracity, Variety)



# Hadoop and Ecosystem Tools



---

# Hadoop Architecture and Ecosystem

---

## Learning Outcomes

After completing this session, you will be able to:

- Understand the Hadoop Architecture
- Explain the various Hadoop Ecosystem tools
- Perform simple commands in HDFS
- Ingest data into Hadoop

# Content

- **Apache Hadoop Overview**
- **Hadoop Architecture**
- **Data Storage (HDFS, HBASE)**
- **Data Ingestion and Tools (HDFS, Sqoop, Flume)**
- **Data Processing Engines (Spark, MapReduce)**
- **Data Analysis Tools (Impala, Hive)**
- **Data Exploration Tools (Search, Solr)**
- **Other Ecosystem Tools (Hue, Oozie, Sentry)**

## Hadoop Architecture & Ecosystem

### Apache Hadoop

- Apache Hadoop is a software framework meant for distributed processing and distributed storage of very large data sets on a group of computer hardware that is affordable and easy to obtain.
- Hadoop is open-source



## Hadoop Architecture & Ecosystem

### Advantages of Hadoop

- Scalable – Can be expanded
- Economical – Running on commodity products. Nothing proprietary
- Resilient – Data storing flexibility
- Reliable – Data processing can be redirected in case of failed nodes
- Performance – Clustered and multiprocessing allows better performance

## Hadoop Architecture & Ecosystem

What are Hadoop commonly used for:

- Extract/Load/Transform (ELT) big data
- Data storage
- Data Analysis
  - Sentiment analysis
  - Risk assessment
  - Classification and feature selection
  - Novelty detection
  - Machine Learning and Deep learning
  - Predictive modelling
  - Text mining

# Hadoop Architecture & Ecosystem

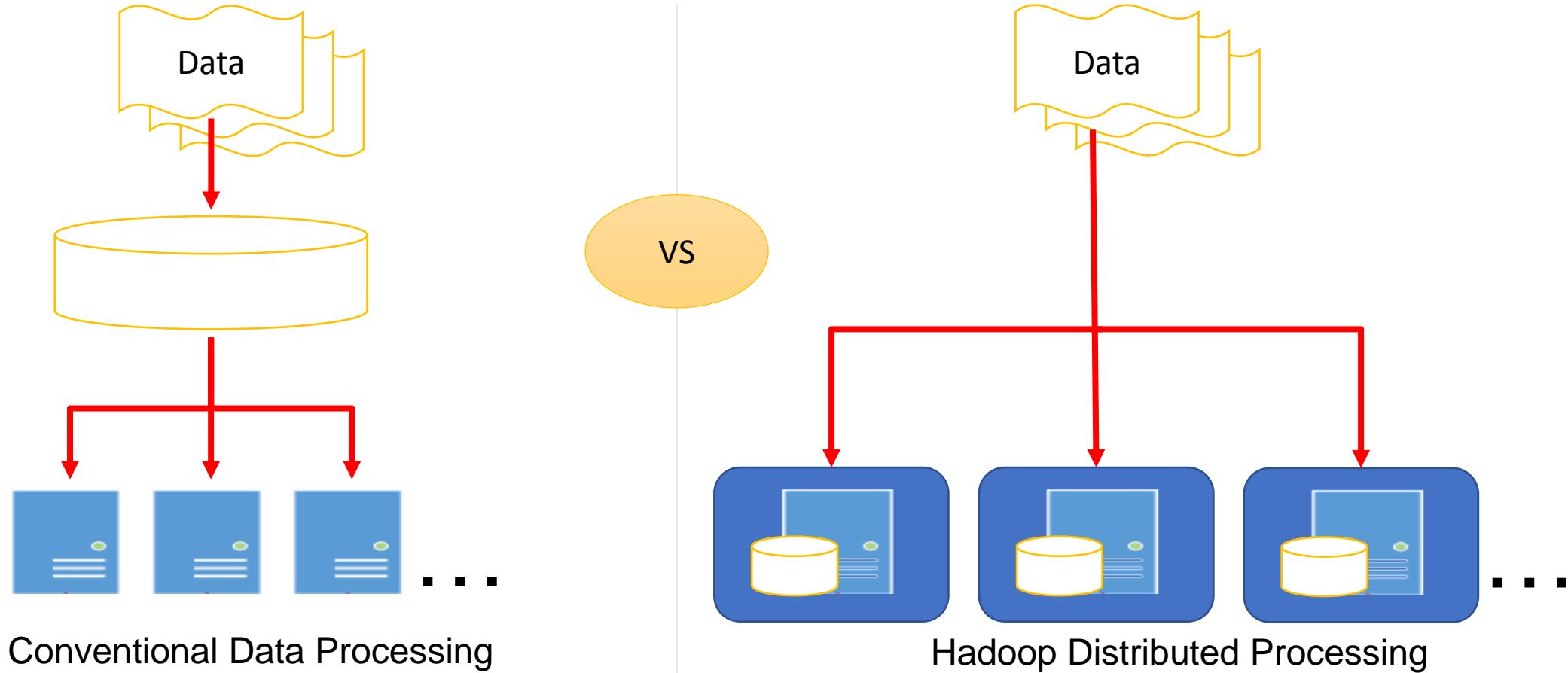
## Hadoop Core

- Hadoop Common
- Hadoop Distributed File System (HDFS)
- MapReduce
- Spark
- YARN

\*Other related projects – Hive, Impala, Flume, Sqoop, Kafka, Pig etc...

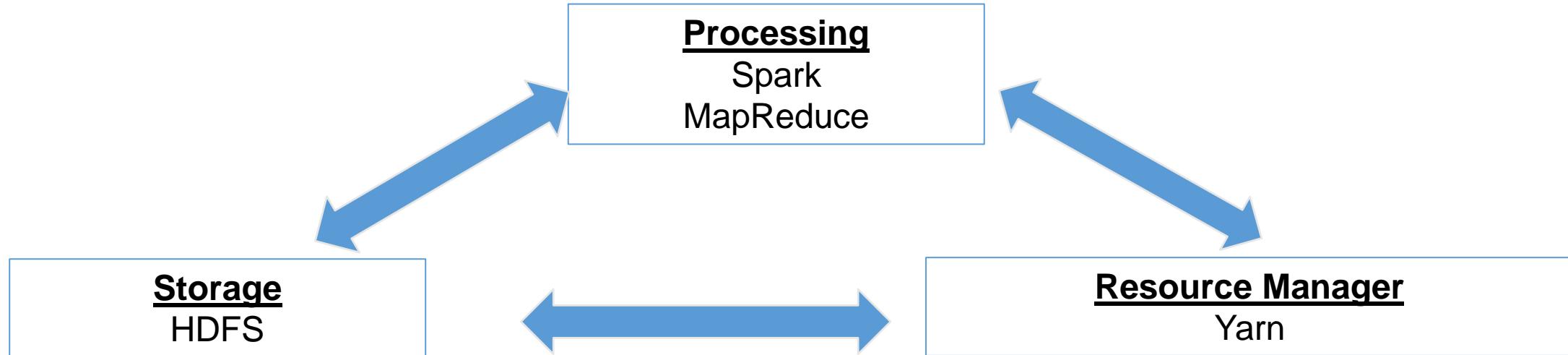
# Hadoop Architecture & Ecosystem

## Conventional vs Hadoop Data Processing



# Hadoop Architecture & Ecosystem

Core Hadoop - 3 components



## Hadoop Architecture & Ecosystem

- Common terms used for Hadoop Cluster

CLUSTER – A group of inter-connected computers that can share processing power and storage

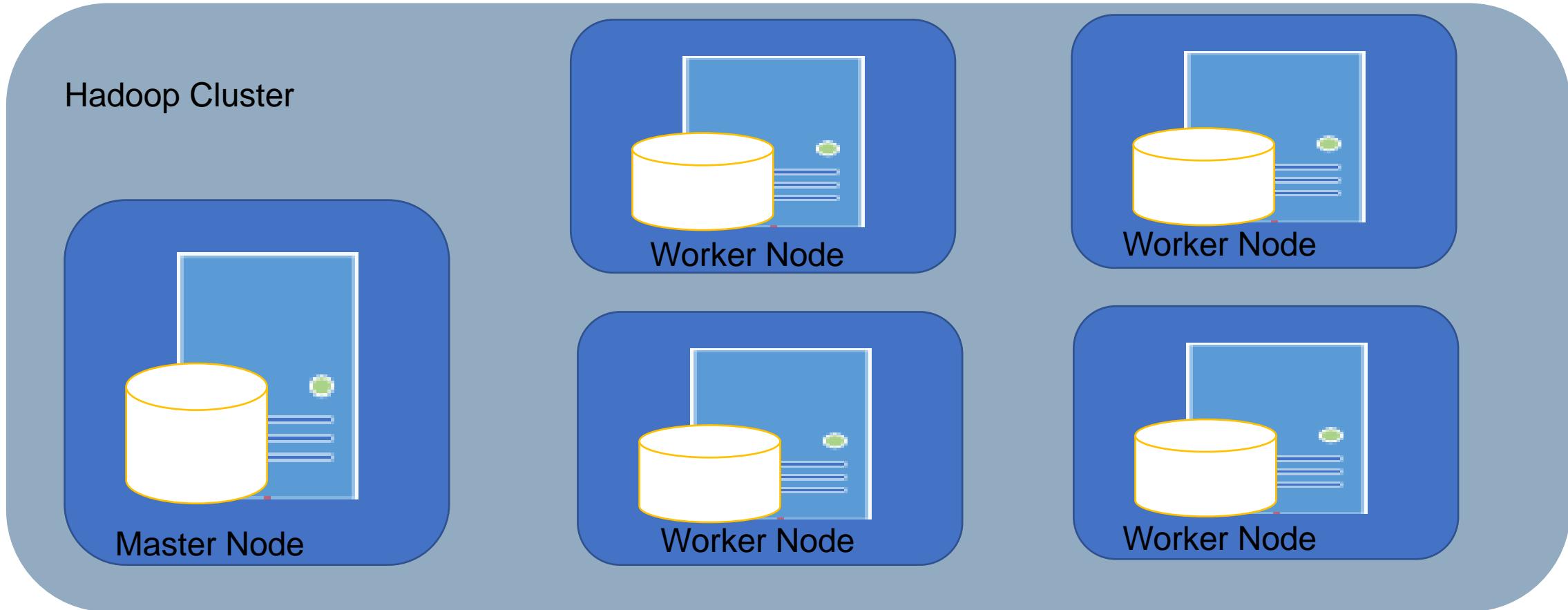
NODE – A single node is an individual entity handling certain responsibilities in a cluster (Master and Worker/Slave)

MASTER NODE – The node that handles management of resources, assignment and killing of tasks, and monitoring of job completion

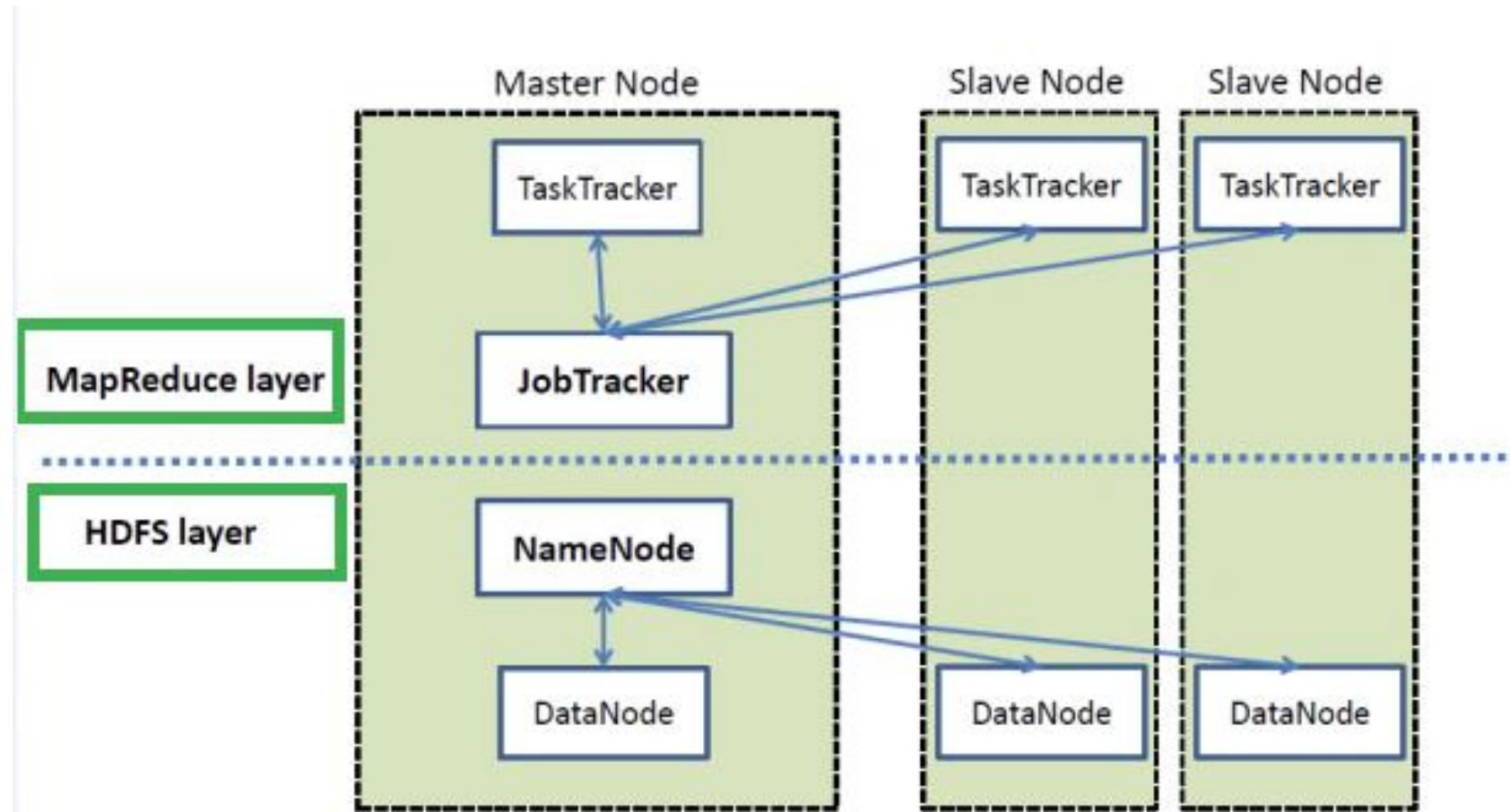
WORKER/SLAVE NODE – Responsible in running the jobs assigned by master node

DAEMON – An instance of task running on a node

# Hadoop Architecture & Ecosystem



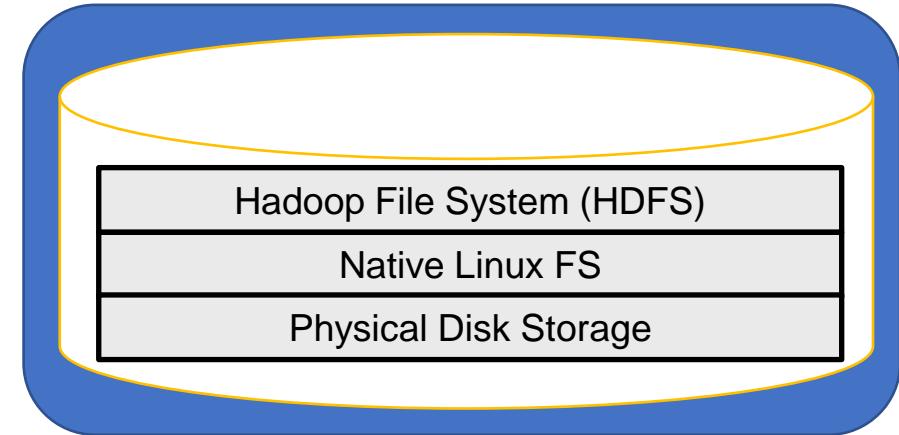
# Hadoop Architecture & Ecosystem



## Hadoop Architecture & Ecosystem

### Basic Concepts of Storage HDFS

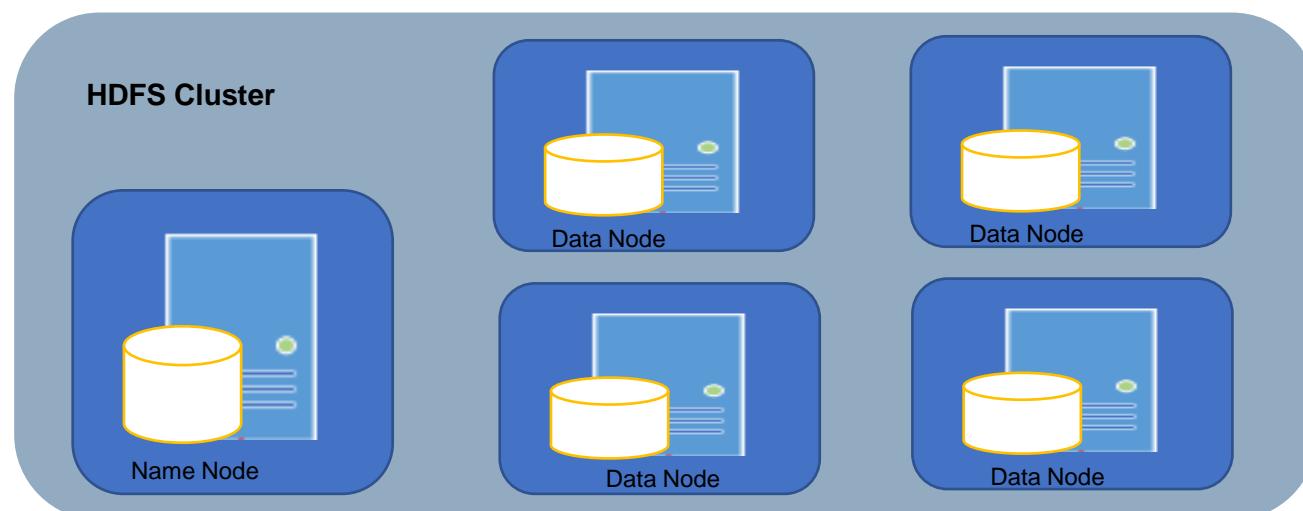
- Written in Java
- Provides customizable redundancy
- Supports large amount of data
- Each file typically more than 100MB
- Optimized for simultaneous large and streaming reads
- “Write Once” concept and not suitable for random writes



## Hadoop Architecture & Ecosystem

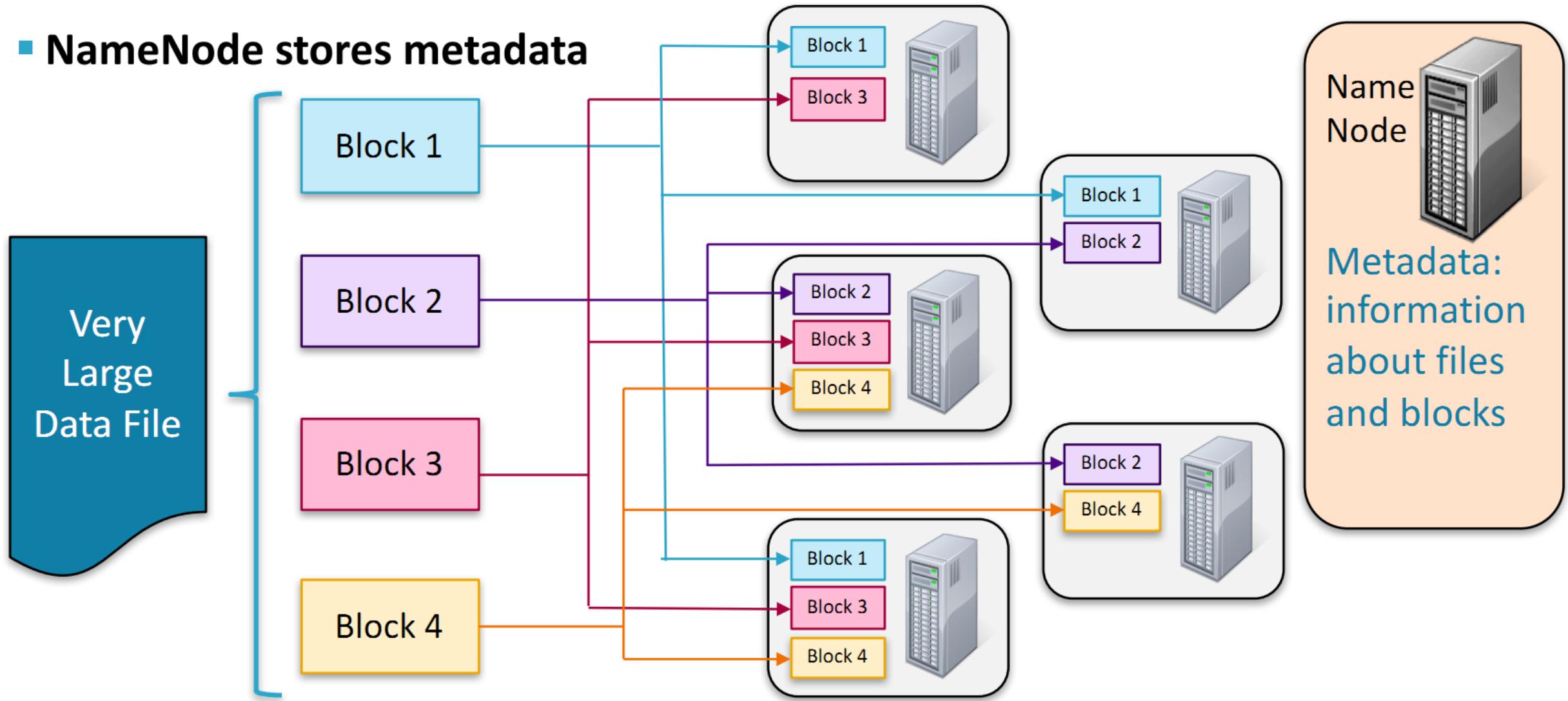
### File storage system (HDFS)

- Data files are split into multiple blocks and blocks are replicated and stored in multiple Data Nodes
- Storing of blocks are indexed by Name Node as metadata
- Default size of 1 block is 128MB



- Each block is replicated on multiple data nodes (default 3x)

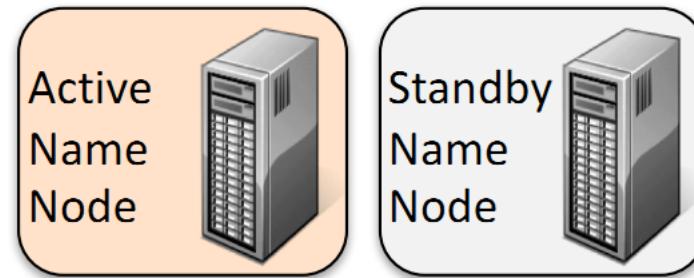
- NameNode stores metadata



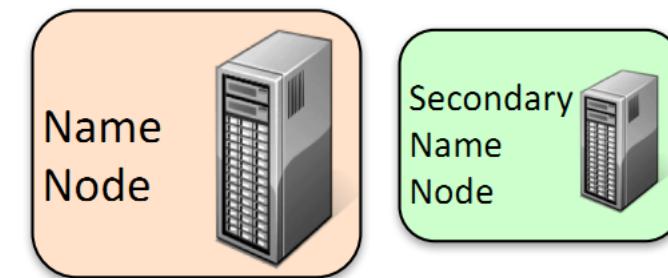
# HDFS Name Node Availability

- **The NameNode daemon must be running at all times**
  - If the NameNode stops, the cluster becomes inaccessible

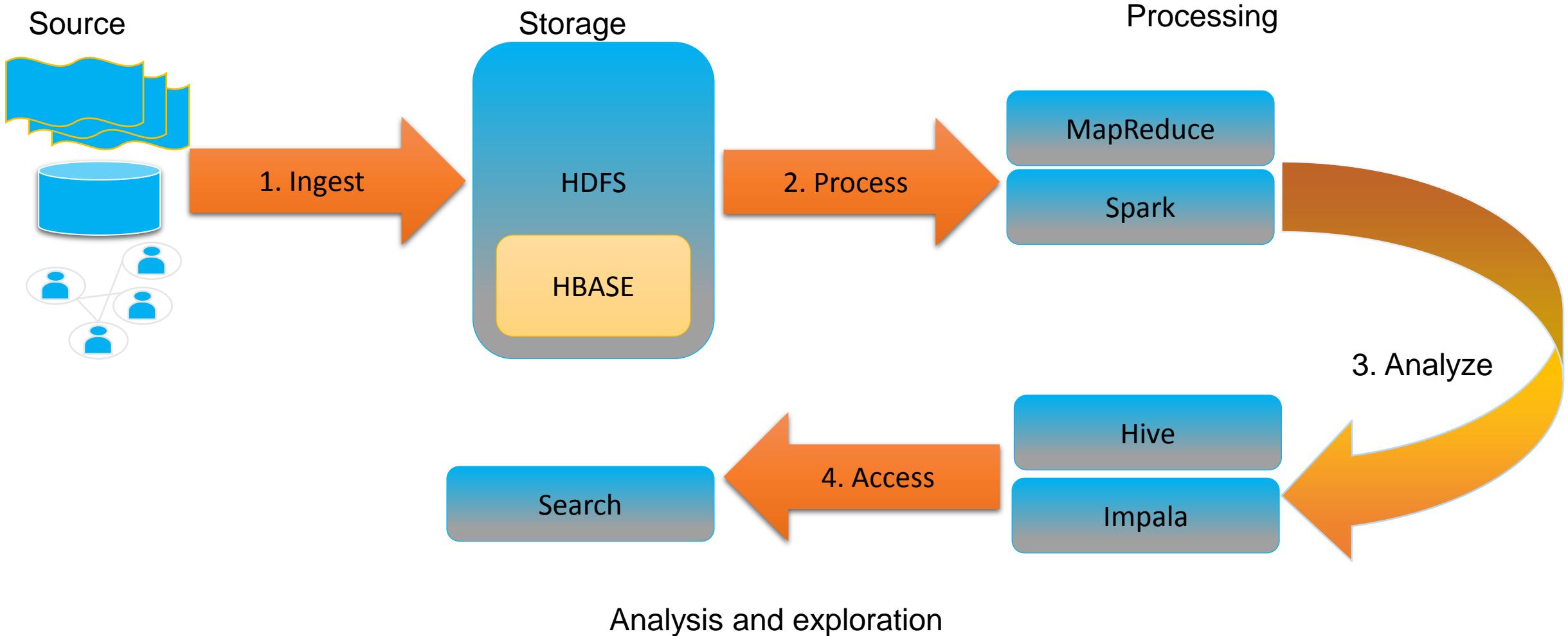
- **HDFS is typically set up for High Availability**
  - Two NameNodes: Active and Standby



- **Small clusters may use “Classic mode”**
  - One NameNode
  - One “helper” node called the Secondary NameNode
    - Bookkeeping, not backup

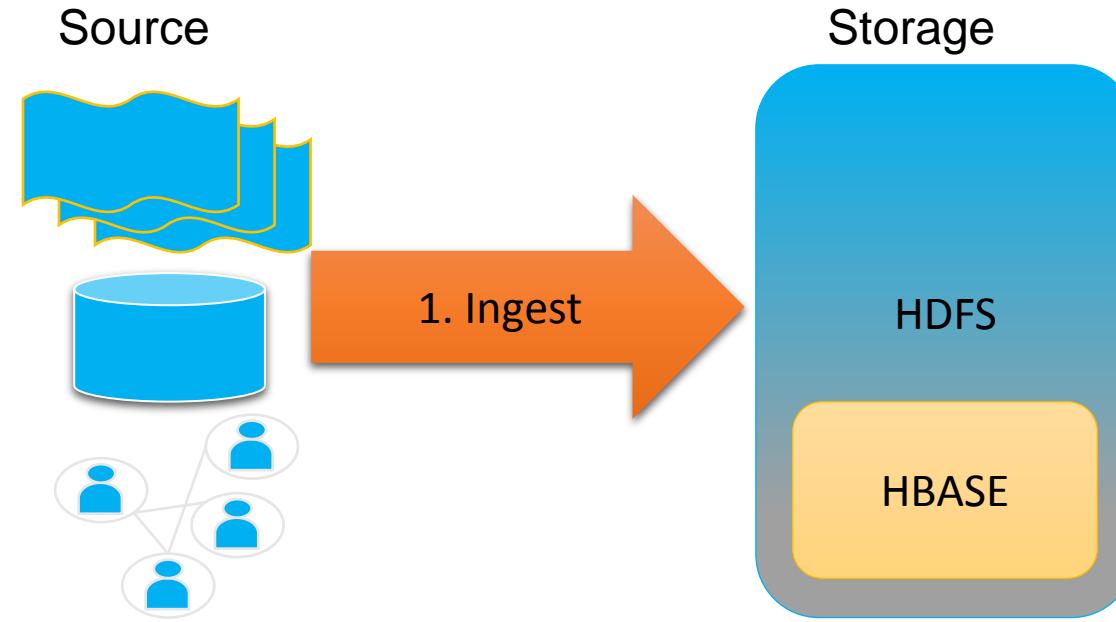


# Hadoop Architecture & Ecosystem



# Hadoop Architecture & Ecosystem

## Data ingestion



Hadoop ingests data from multiple formats

- RDBMS
- Imported Files
- Logs

# Hadoop Architecture & Ecosystem

## Data Ingestion Tools

### HDFS

- Uses direct file transfer with web-based interface HUE or
- command line “\$ hdfs dfs –put ~”

### Apache Sqoop

- Fast importing from and to RDBMS
- Supports various standard databases such as PostgreSQL, MySQL, MongoDB, Oracle, etc.

### Apache Flume

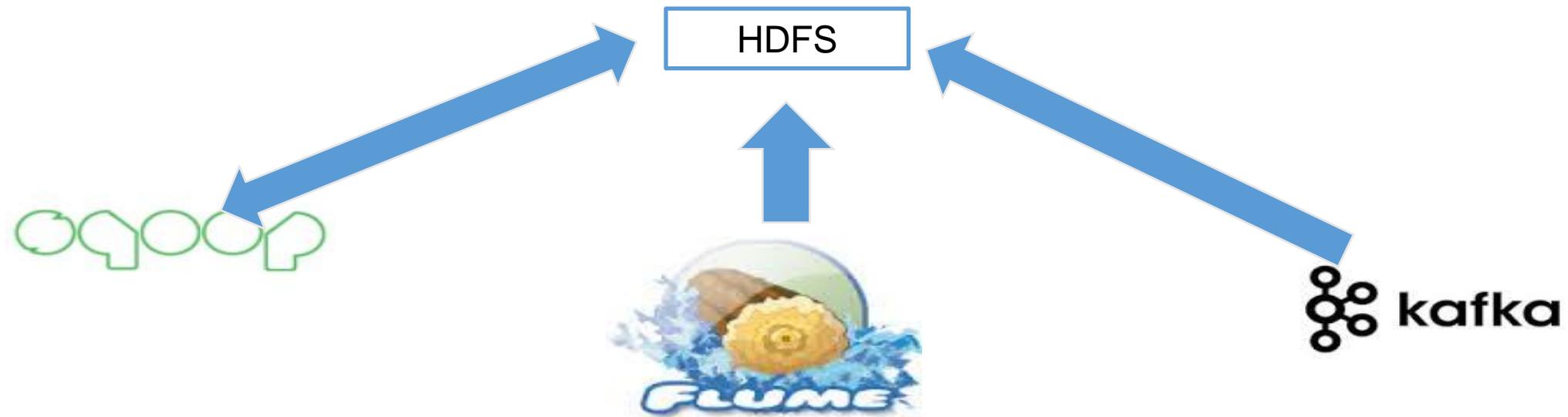
- Supports multi-channel streaming data ingestion
- Suitable for event based data from multiple systems such as log files, sensor data, click streams, etc.

# Hadoop Architecture & Ecosystem

## Data Ingestion Tools

### Kafka

- Integrated with Flume and Spark Streaming
- A distributed streaming platform
- Used for building real-time data pipelines and streaming apps.
- Highly scalable, fault-tolerant, fast



# Hadoop Architecture & Ecosystem

## Data Processing Engine

### Apache Spark – Large scale

- General Purpose
- Runs on distributed environment and processing data in HDFS
- Supports wide range of applications
  - Streaming data processing
  - Business intelligence
  - Machine learning
  - Text processing



# Hadoop Architecture & Ecosystem

## Data Processing Engine

### Hadoop MapReduce – Original Hadoop Framework

- Java-based
- Majority of existing tools still built on MapReduce but are slowly migrating to Spark
- Highly fault tolerant framework, however, much slower compare to Spark



# Hadoop Architecture & Ecosystem

## Data Analysis Tools

### Apache Impala - High performance SQL

- **100% open source**
- **high-performance SQL engine and support SQL commands**
- **runs on Hadoop clusters**
- **data are stored as HDFS files or HBase tables**
- **very low latency**
- **ideal for interactive analysis**



# Hadoop Architecture & Ecosystem

## Data Exploration Tools

Cloudera Search – Data exploration platform

- 100% open source
- Supports full interactive text search for any data on Hadoop with web-based dashboard on Hue
- User-friendly search interface
- Cloudera Search is an integration of Apache Solr with HDFS, MapReduce, HBase, and Flume



# Hadoop Architecture & Ecosystem

## Other Ecosystem Tools

### HUE – Hadoop User Experience

- 100% open source
- Provides web-based interactive interface for
  - Files and Data upload, browsing, and search
  - Hive and Impala scripting interface
  - Etc...
- User-friendly search interface



# Hadoop Architecture & Ecosystem

## Other Ecosystem Tools

### Oozie – Workflow Engine

- Coordinate and monitor the workflow
- Arrange and submit jobs in the right sequence
- Trigger event or time based reports



# Hadoop Architecture & Ecosystem

## Other Ecosystem Tools

### Sentry – Hadoop Security

- **Supports authentication and authorization access control to HDFS, Cloudera Search, Hive and Impala**
- **Provides security solution for entire cluster**



## Hands-On Session – Exercise 1

### Introduction to Hands-on environment setup

- Server Access
- Google drive access
- PostgresSQL create database and tables



# Data Ingestion

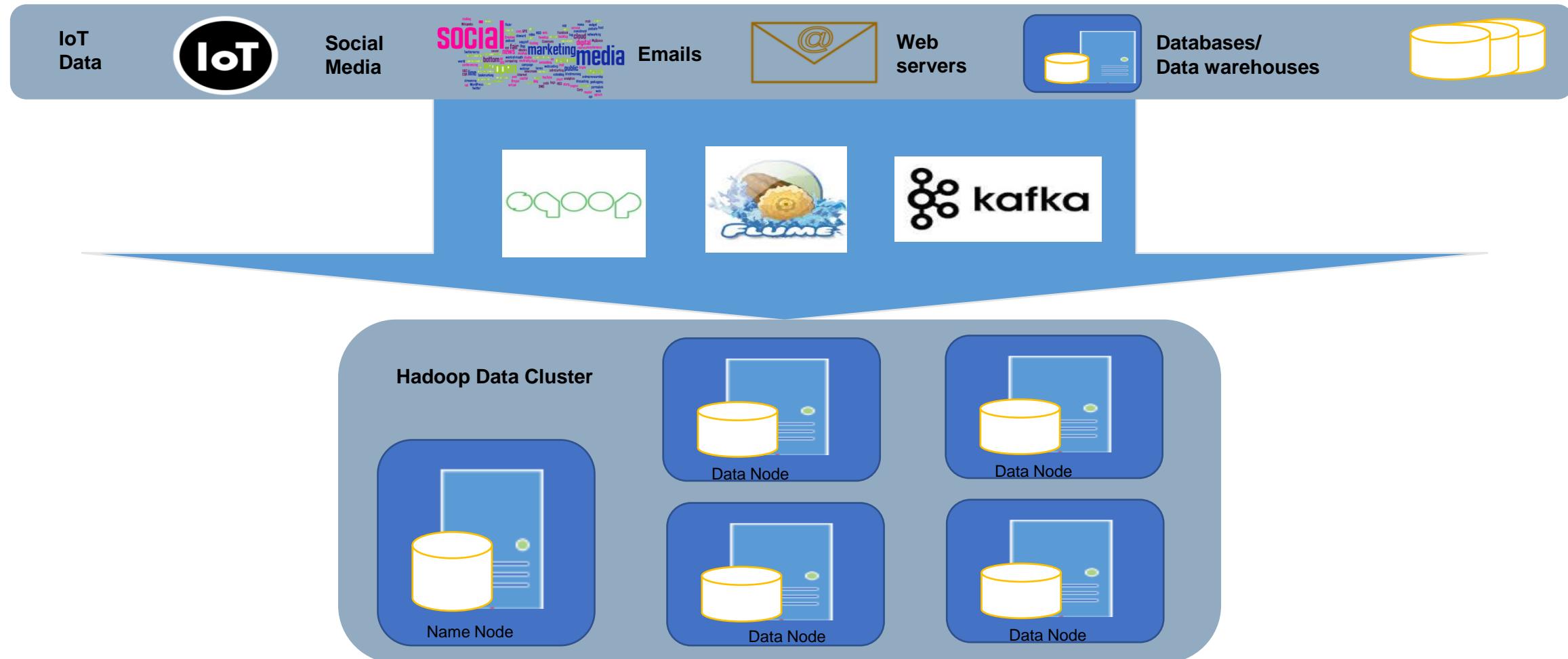
# Data Ingestion

## Data Ingestion Overview

- Hadoop supports data ingestions from various sources:
  - Flat files
  - Web logs
  - RDBMS
  - Streaming data from sensors
  - Social media feeds
  - Click streams logs
  - Syslogs
  - Etc.

# Data Ingestion

# Data Ingestion Overview



## Data Ingestion

### Accessing HDFS

- **Various methods of accessing HDFS**
  - **Command line:**
    - - **FsShell;**
    - **\$hdfs dfs**
  - **In Spark environment**
    - **Hdfs://nnhost:port/file...**
  - **Other programs**
    - **Java API**
      - **Backend support for Hadoop tools such as MapReduce, Impala, Hue, Sqoop, Flume**

## Data Ingestion

### Accessing HDFS – Importing Flat Files with commands

- Examples of HDFS Commands
  - To copy a file from local directory into HDFS (into /user/username/filename.txt)

```
$ hdfs dfs -put filename.txt filename.txt
```

- To perform a directory listing in HDFS home directory

```
$ hdfs dfs -ls
```

- To perform a directory listing in HDFS root directory

```
$ hdfs dfs -ls /
```

# Data Ingestion

## Accessing HDFS

- Examples of HDFS Commands
  - To display the content of a file

```
$ hdfs dfs -cat /user/username/filename.txt
```

- To extract file from HDFS to local directory

```
$ hdfs dfs -get /user/username/file.txt file.txt
```

- To create a directory in the user HDFS directory

```
$ hdfs dfs -mkdir newDirectoryName
```

- To delete the directory and its content

```
$ hdfs dfs -rm -r newDirectoryName
```

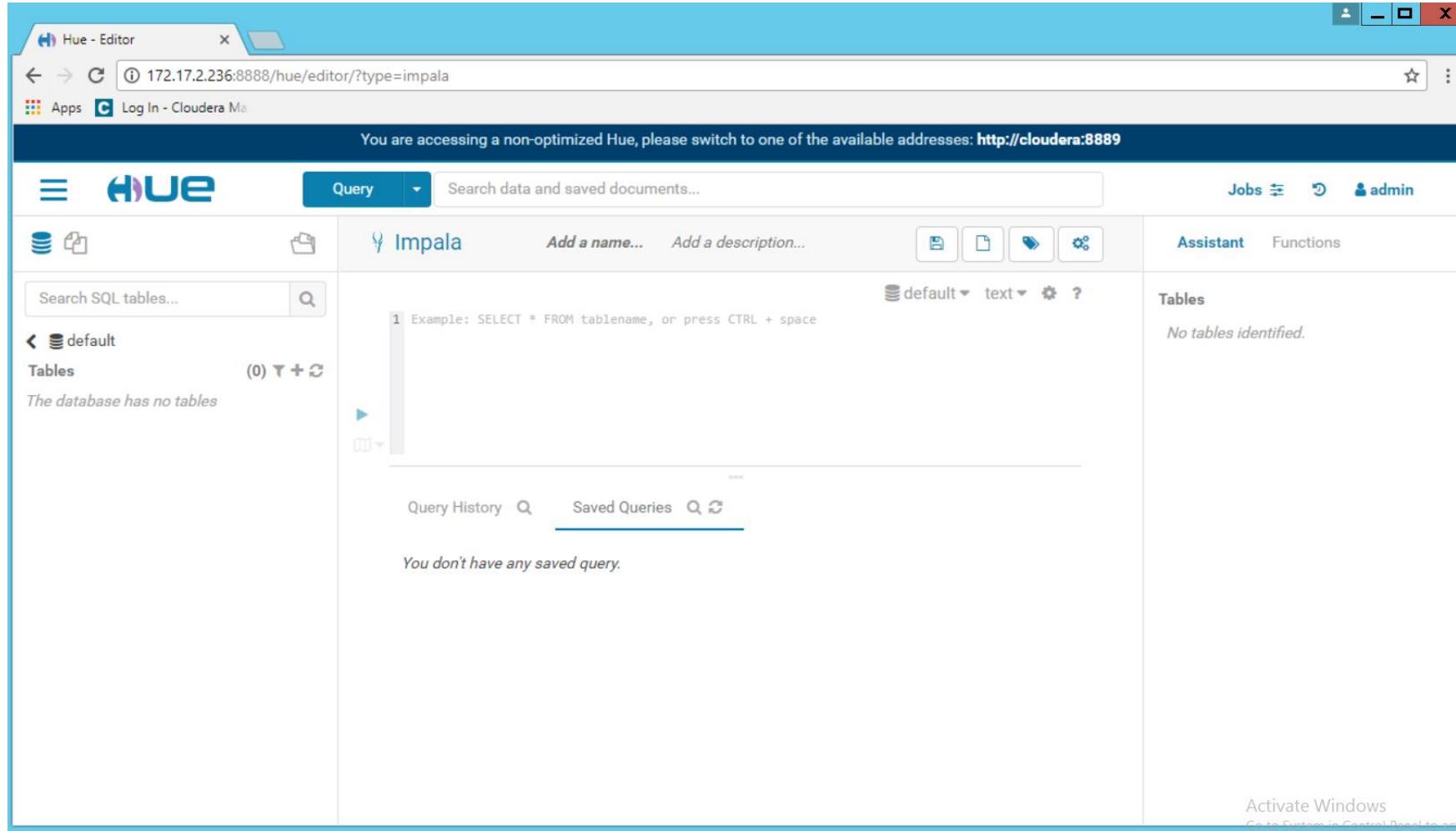
## Data Ingestion

### Accessing HDFS - HUE

- **Using File Browser – Hadoop User Experience(HUE)**
- **HUE can be used to manage HDFS files and directories**
- **Accessing using web interface**
  - **<http://localhost:8888>**

# Data Ingestion

## Accessing HDFS - HUE

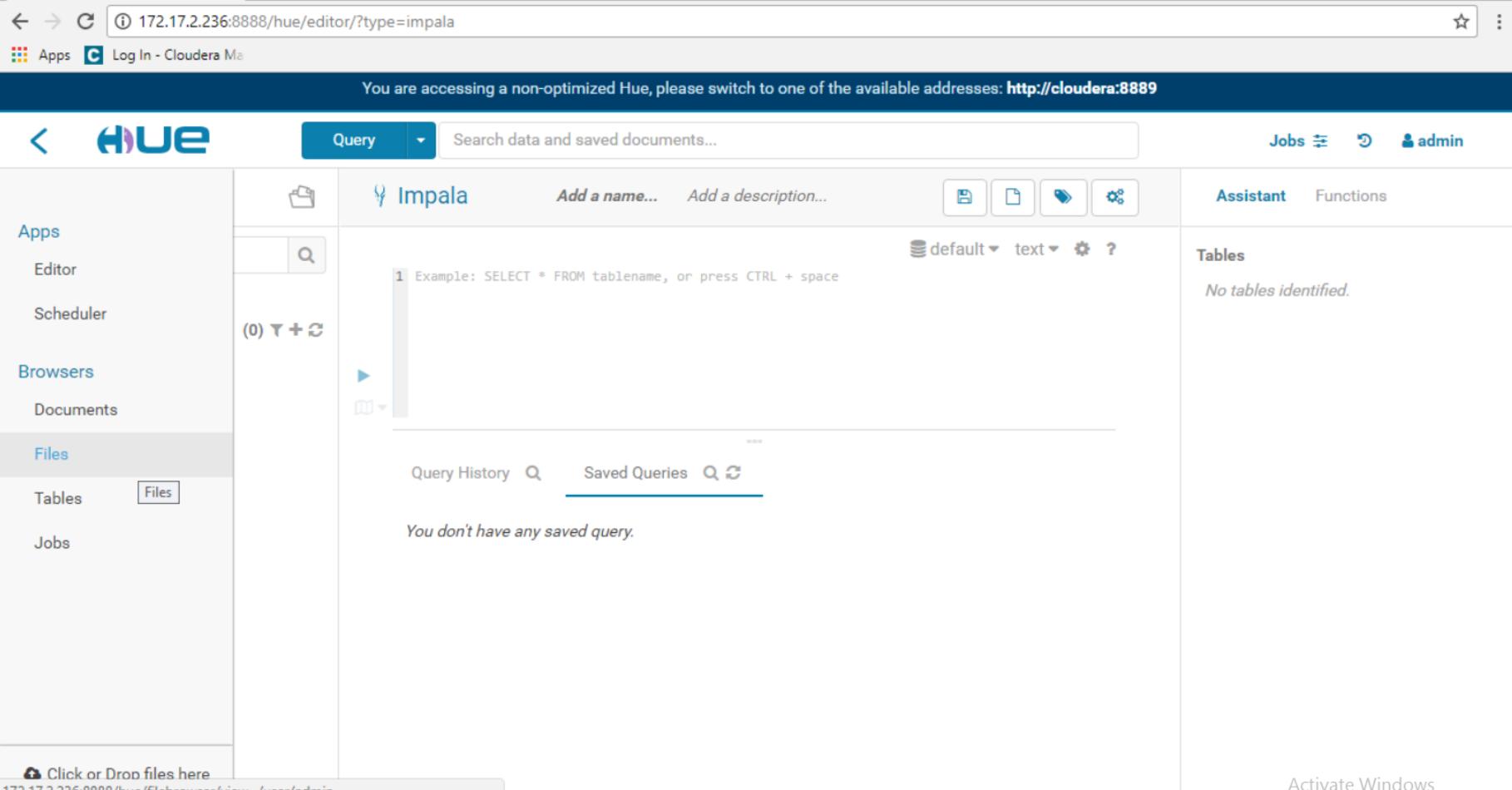


The screenshot shows the Hue Editor interface for interacting with an Impala database. The URL in the browser bar is [172.17.2.236:8888/hue/editor/?type=impala](http://172.17.2.236:8888/hue/editor/?type=impala). A message at the top of the page reads: "You are accessing a non-optimized Hue, please switch to one of the available addresses: <http://cloudera:8889>".

The interface includes a search bar for "Search data and saved documents..." and a user menu showing "admin". On the left, there's a sidebar for "Tables" under the "default" database, which currently has no tables. The main area is titled "Impala" and contains fields for "Add a name..." and "Add a description...". It also includes buttons for "File", "Edit", "Delete", and "Run". Below this is a query input field with placeholder text: "Example: SELECT \* FROM tablename, or press CTRL + space". To the right, there are sections for "Assistant" and "Functions", and a "Tables" section which states "No tables identified". At the bottom, there are links for "Query History" and "Saved Queries". A note at the bottom center says "You don't have any saved query.". A footer at the bottom right provides instructions to "Activate Windows" by going to the System Control Panel.

# Data Ingestion

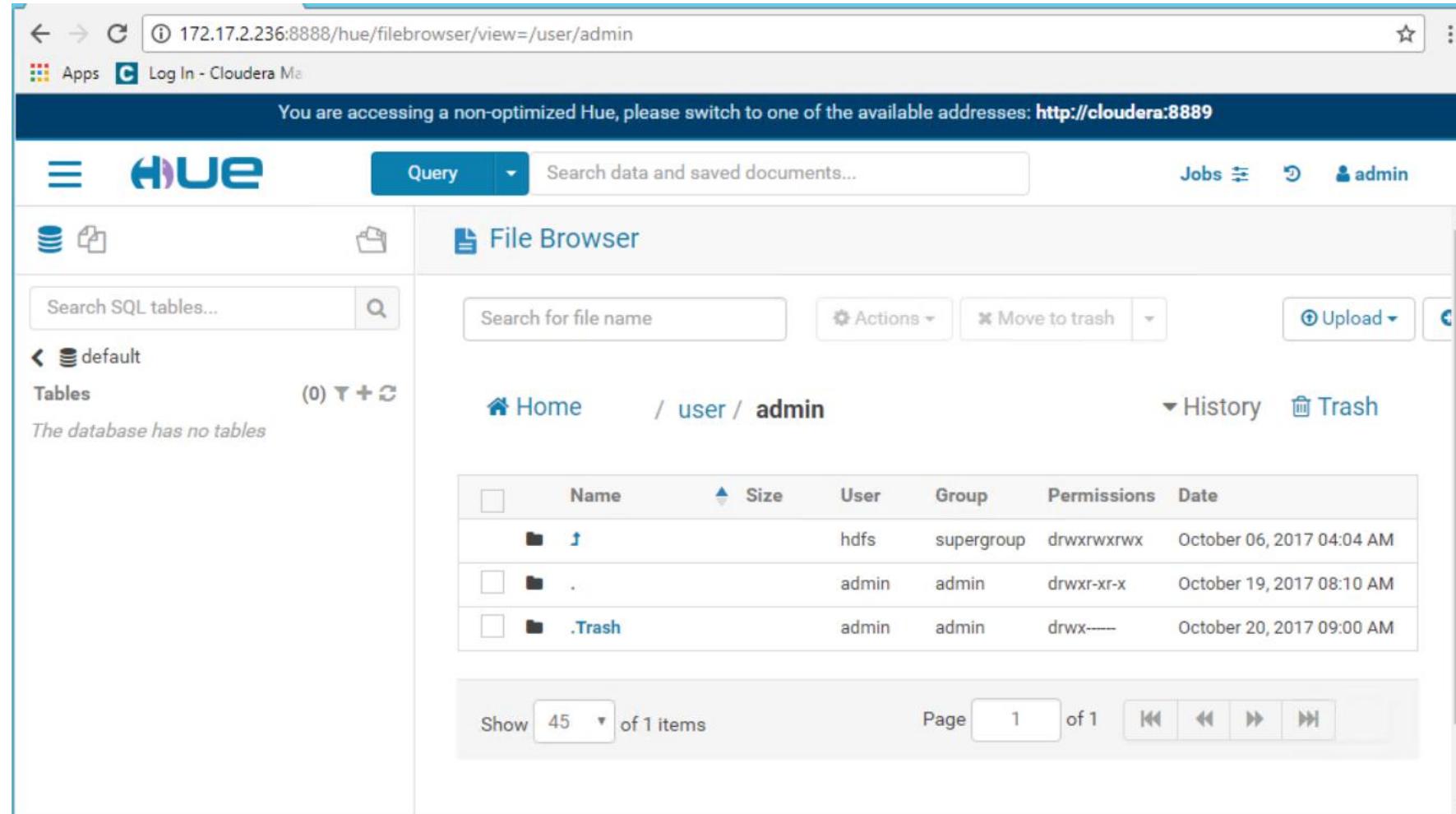
## Accessing HDFS - HUE



The screenshot shows the Apache Hue web interface at the URL [172.17.2.236:8888/hue/editor/?type=impala](http://172.17.2.236:8888/hue/editor/?type=impala). The interface is designed for querying data stored in HDFS using various engines like Impala. The main navigation bar includes links for Apps, Log In - Cloudera Manager, and a search bar. On the left, a sidebar lists the Apps (Editor, Scheduler), Browsers (Documents), and Files (Tables, Jobs). The central area is titled "Impala" and contains fields for "Add a name..." and "Add a description...". It features a query editor with a placeholder "Example: SELECT \* FROM tablename, or press CTRL + space", a dropdown for "default", and a "text" dropdown. Below the editor, tabs for "Query History" and "Saved Queries" are shown, with the latter being active. A message "You don't have any saved query." is displayed. At the bottom, there's a note "Click or Drop files here" and the URL [172.17.2.236:8888/hue/filebrowser/view=/user/admin](http://172.17.2.236:8888/hue/filebrowser/view=/user/admin).

# Data Ingestion

## Import Flat Files – HUE – Drag and Drop to upload files



The screenshot shows the Hue File Browser interface at the URL <http://172.17.2.236:8888/hue/filebrowser/view=/user/admin>. A warning message at the top states: "You are accessing a non-optimized Hue, please switch to one of the available addresses: <http://cloudera:8889>". The interface includes a sidebar for SQL tables and a main area for file browsing. The current path is /user/admin. The file browser table lists the following items:

	Name	Size	User	Group	Permissions	Date
<input type="checkbox"/>	hdfs		hdfs	supergroup	drwxrwxrwx	October 06, 2017 04:04 AM
<input type="checkbox"/>	.		admin	admin	drwxr-xr-x	October 19, 2017 08:10 AM
<input type="checkbox"/>	.Trash		admin	admin	drwx----	October 20, 2017 09:00 AM

Navigation and search controls are also visible.

## What is YARN?

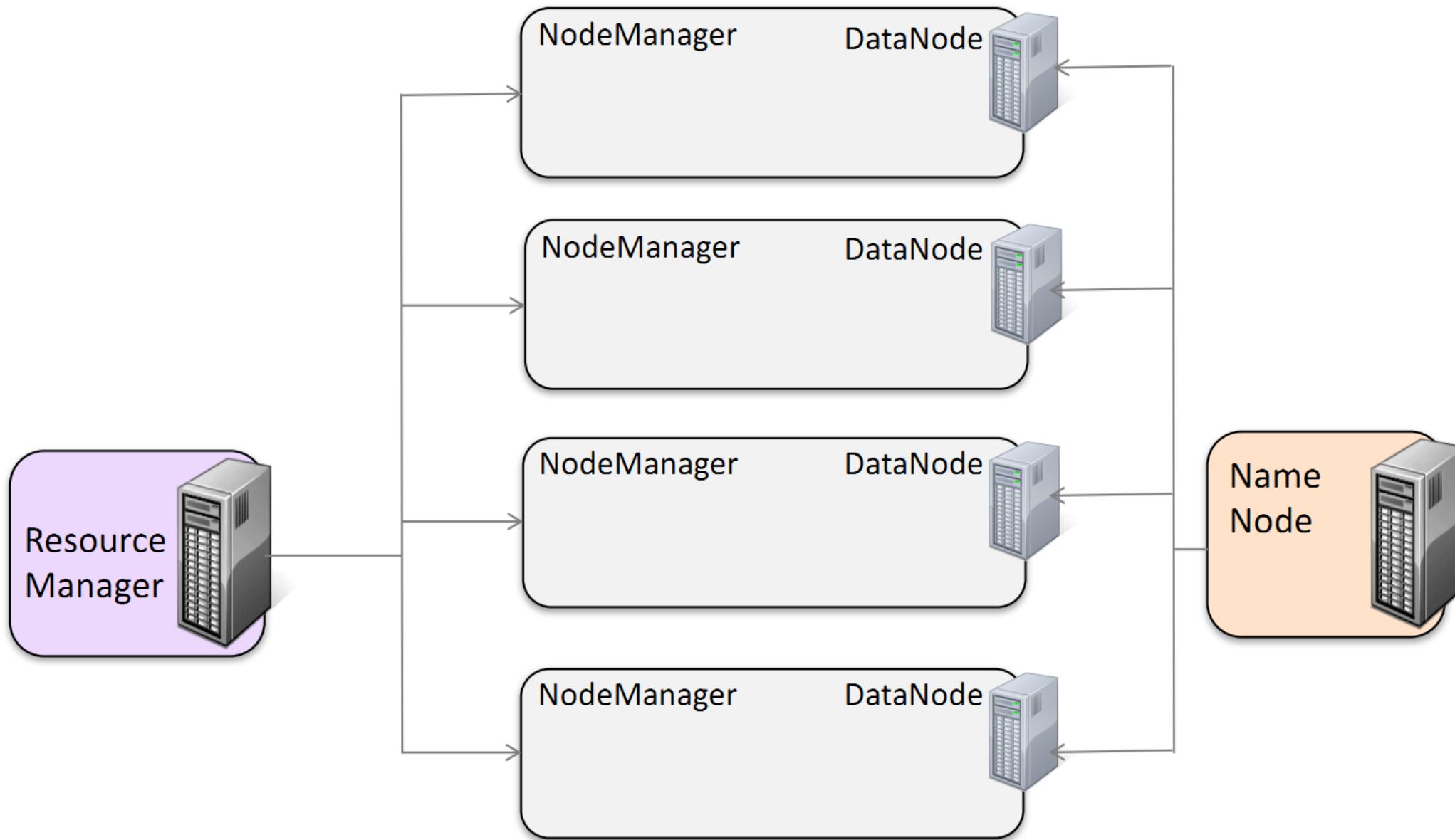
- **YARN = Yet Another Resource Negotiator**
- **Yarn is the Hadoop processing layer that contains**
  - **Resource Manager**
  - **Job Scheduler**

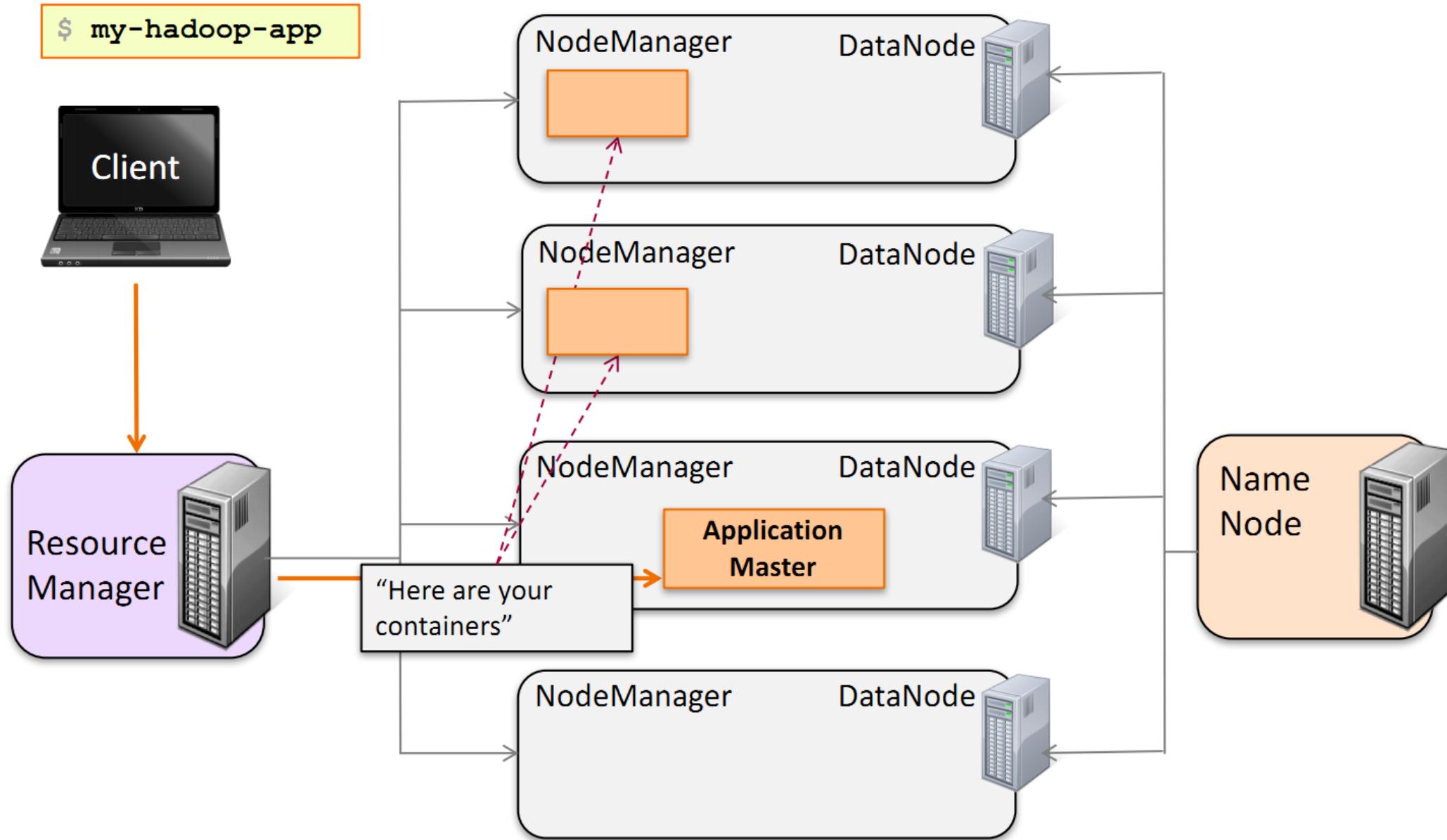
## YARN Daemons

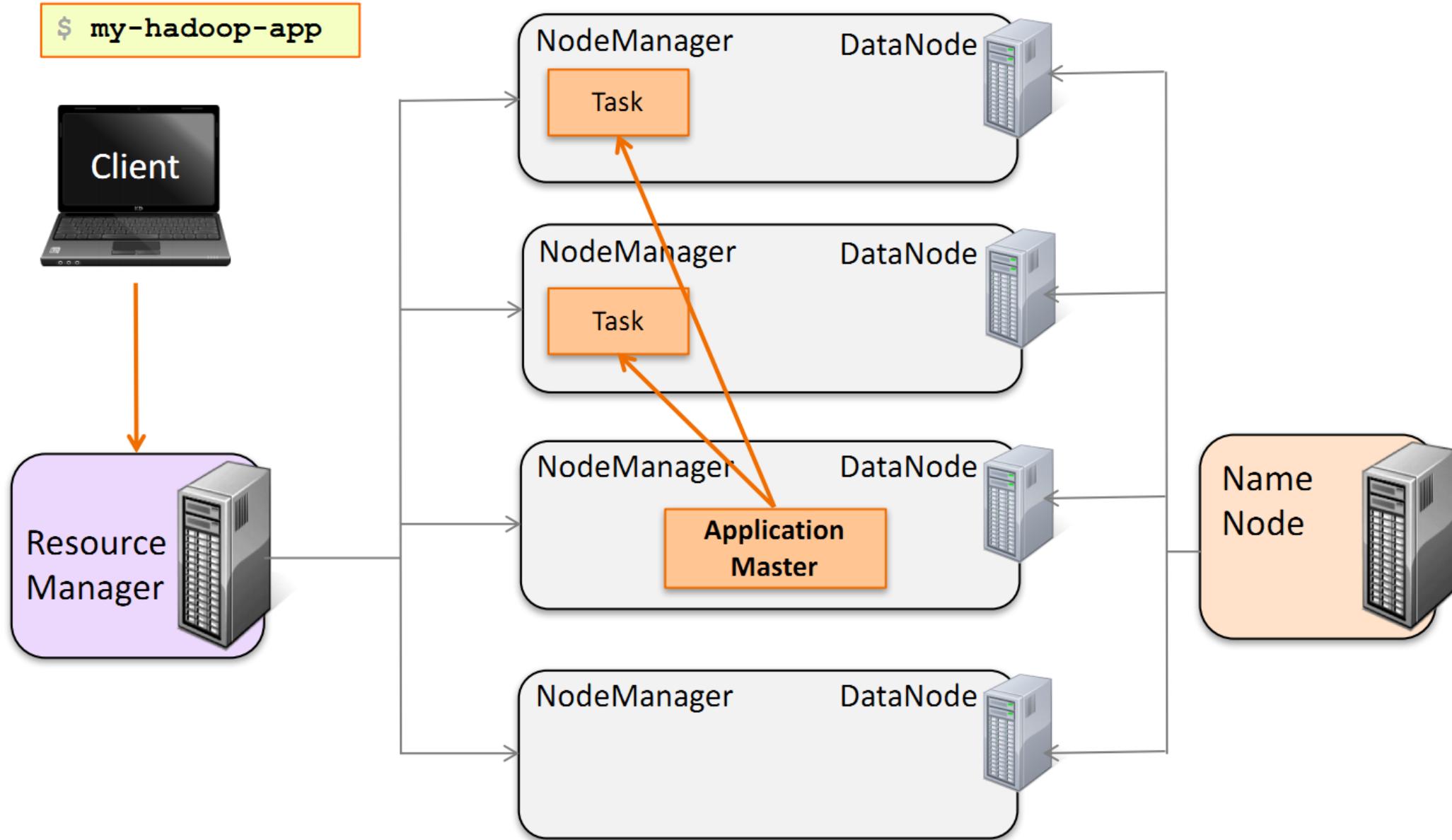
- **Resource Manager (RM)**
  - Runs on master node
  - Global resource scheduler
  - Coordinate between competing applications
  - Has a fair scheduler for different algorithms
- **Node Manager (NM)**
  - Runs on worker nodes
  - Communicates with RM
  - Manages node resources
  - Launches Containers

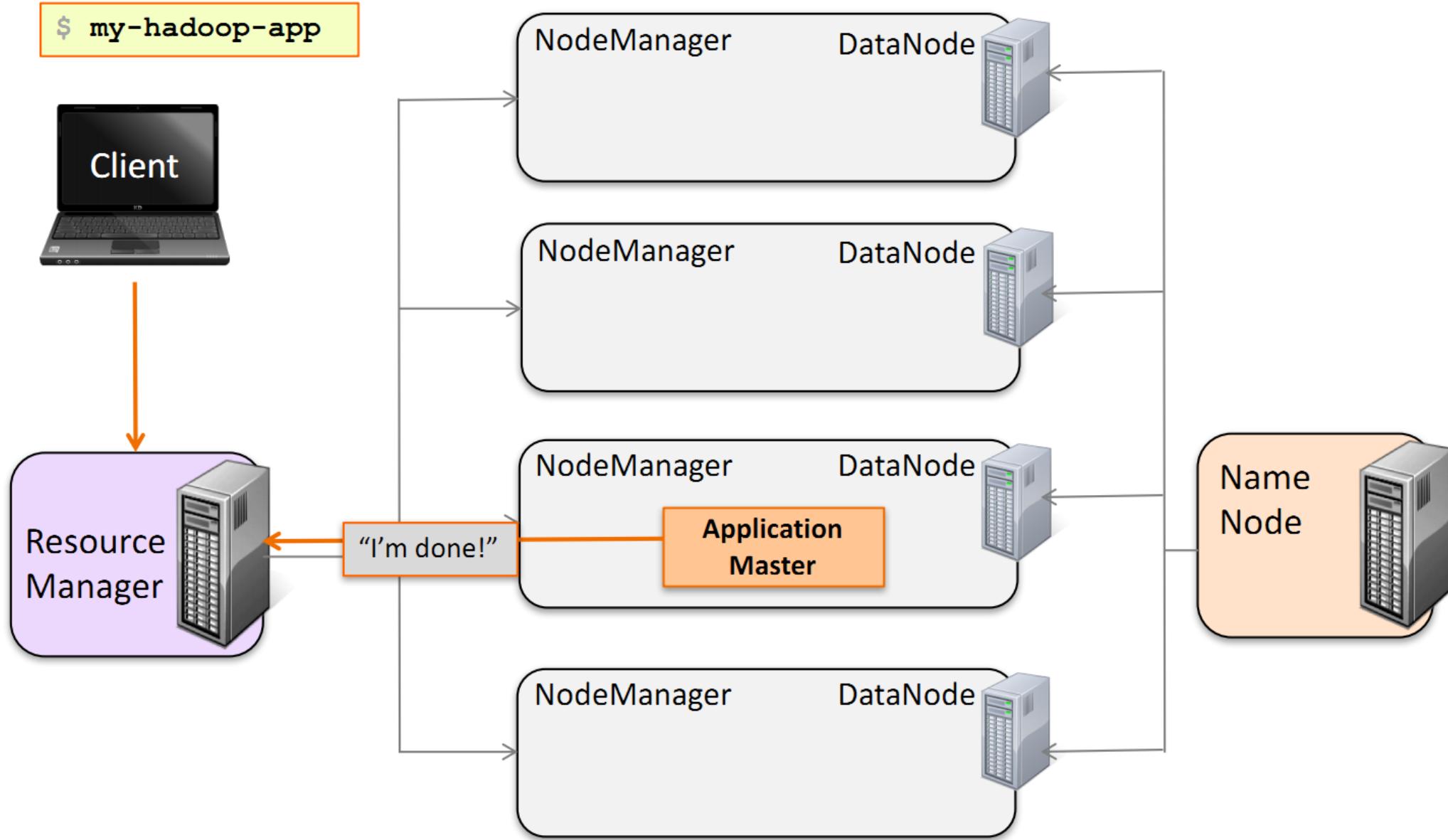
## Running an Application in Yarn

- **Containers**
  - Created by RM upon request
  - Allocate resources (memory/CPU) on a worker node
  - Applications run in one or more containers
- **Application Master (AM)**
  - One per application
  - Framework/application specific
  - Runs in a container
  - Requests more containers to run more application tasks





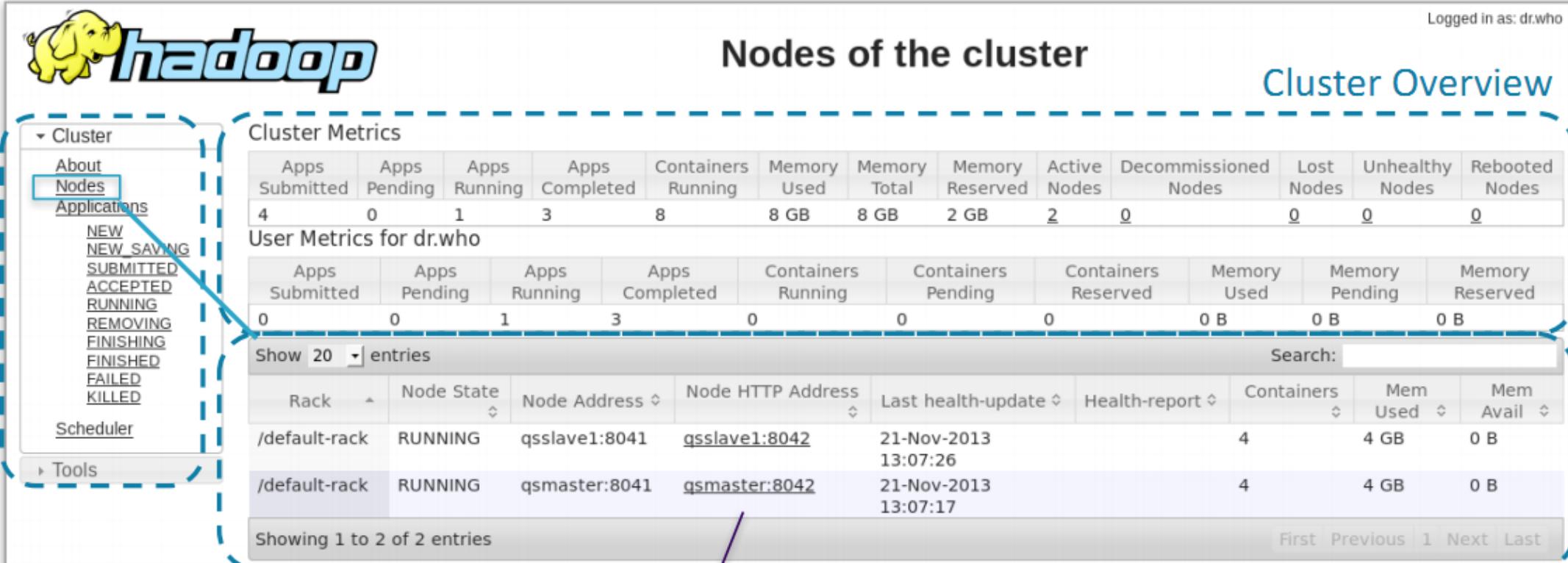




## The YARN Web UI

- Resource Manager UI is the main entry point
  - Runs on the RM host on port 8080 or 8088 by default
- Provides more detailed view than Hue
- Does not provide any control or configuration

# Resource Manager UI: Nodes



The screenshot shows the Hadoop Resource Manager UI with the following details:

- Header:** Nodes of the cluster, Cluster Overview, Logged in as: dr.who
- Left Sidebar (Cluster Metrics):**
  - Cluster Metrics table:

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	Active Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes
4	0	1	3	8	8 GB	8 GB	2 GB	2	0	0	0	0
  - User Metrics for dr.who table:

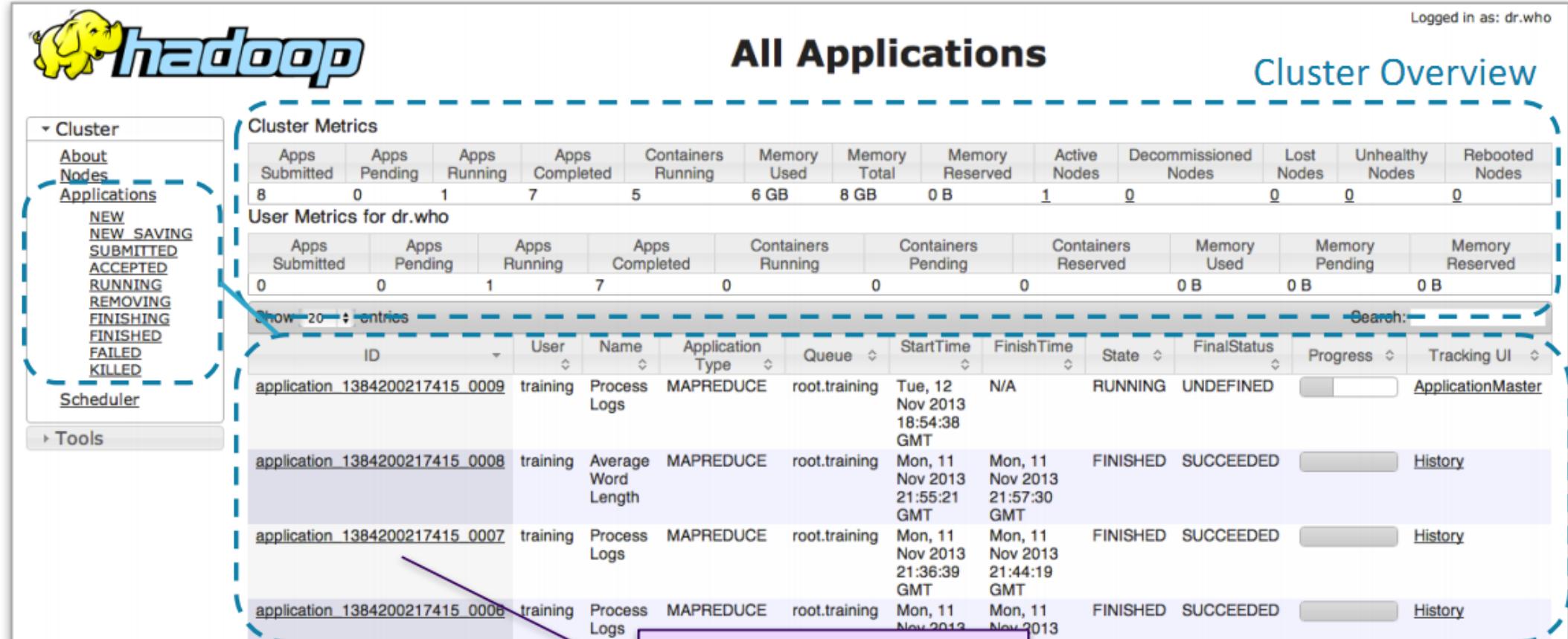
Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Containers Pending	Containers Reserved	Memory Used	Memory Pending	Memory Reserved
0	0	1	3	0	0	0	0 B	0 B	0 B
  - Buttons: Show 20 entries, Search, Rack, Node State, Node Address, Node HTTP Address, Last health-update, Health-report, Containers, Mem Used, Mem Avail.
  - Table Headers: Showing 1 to 2 of 2 entries, First, Previous, 1, Next, Last.
- Right Content Area:** Nodes of the cluster table (dashed border).

Rack	Node State	Node Address	Node HTTP Address	Last health-update	Health-report	Containers	Mem Used	Mem Avail
/default-rack	RUNNING	qsslave1:8041	gsslave1:8042	21-Nov-2013 13:07:26		4	4 GB	0 B
/default-rack	RUNNING	qsmaster:8041	gsmaster:8042	21-Nov-2013 13:07:17		4	4 GB	0 B

link to Node Manager UI

List of each node in cluster

# Resource Manager UI: Applications



The screenshot shows the Hadoop Resource Manager UI for managing applications. It includes a sidebar with cluster status and tools, and a main area for viewing all applications.

**Cluster Metrics:**

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	Active Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes
8	0	1	7	5	6 GB	8 GB	0 B	1	0	0	0	0

**User Metrics for dr.who:**

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Containers Pending	Containers Reserved	Memory Used	Memory Pending	Memory Reserved
0	0	1	7	0	0	0	0 B	0 B	0 B

**Application Table Headers:**

ID	User	Name	Application Type	Queue	StartTime	FinishTime	State	FinalStatus	Progress	Tracking UI
----	------	------	------------------	-------	-----------	------------	-------	-------------	----------	-------------

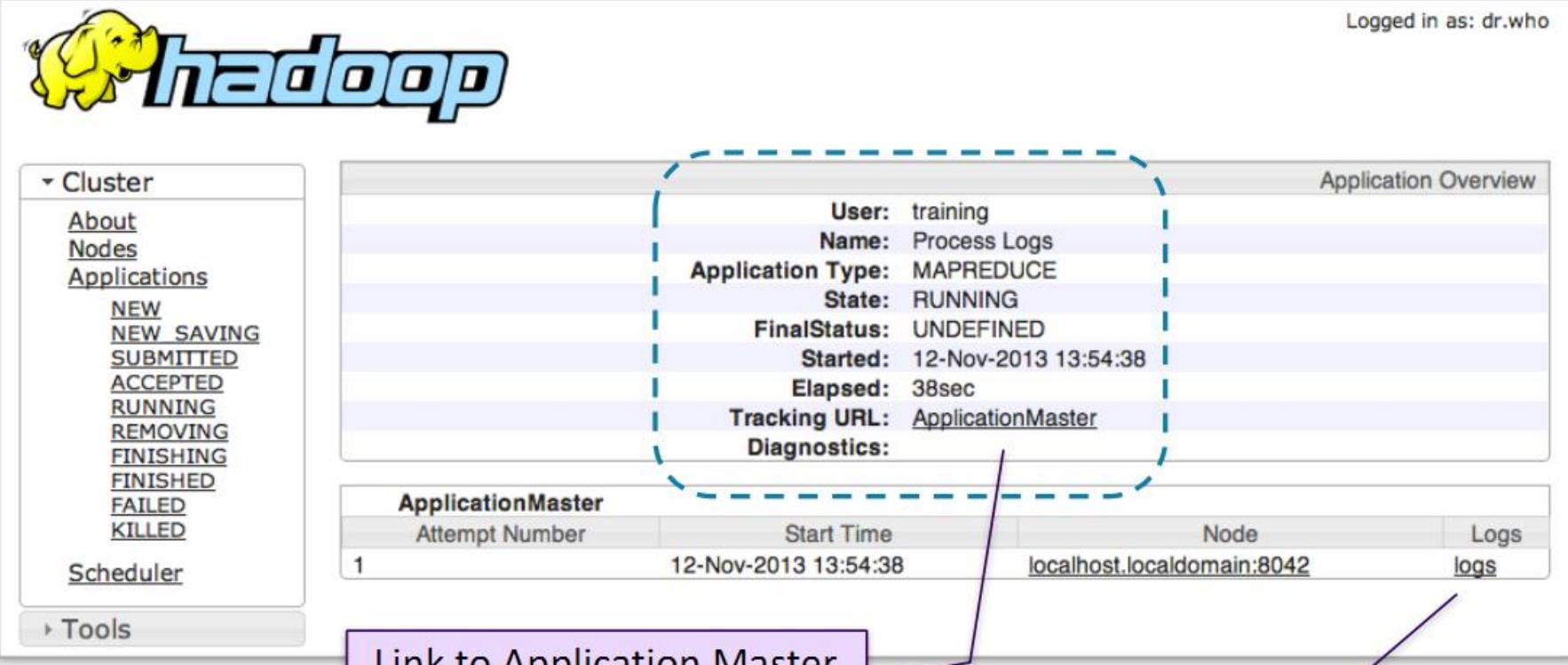
**Applications Listed:**

application_1384200217415_0009	training	Process Logs	MAPREDUCE	root.training	Tue, 12 Nov 2013 18:54:38 GMT	N/A	RUNNING	UNDEFINED	<input type="button"/>	ApplicationMaster
application_1384200217415_0008	training	Average Word Length	MAPREDUCE	root.training	Mon, 11 Nov 2013 21:55:21 GMT	Mon, 11 Nov 2013 21:57:30 GMT	FINISHED	SUCCEEDED	<input type="button"/>	History
application_1384200217415_0007	training	Process Logs	MAPREDUCE	root.training	Mon, 11 Nov 2013 21:36:39 GMT	Mon, 11 Nov 2013 21:44:19 GMT	FINISHED	SUCCEEDED	<input type="button"/>	History
application_1384200217415_0006	training	Process Logs	MAPREDUCE	root.training	Mon, 11 Nov 2013	Mon, 11 Nov 2013	FINISHED	SUCCEEDED	<input type="button"/>	History

Link to Application  
Details... (next slide)

List of running and  
recent applications

# Resource Manager UI: Application Details



The screenshot shows the Hadoop Resource Manager UI with a red header bar. The main content area displays application details for a MapReduce job.

**Application Overview:**

- User: training
- Name: Process Logs
- Application Type: MAPREDUCE
- State: RUNNING
- FinalStatus: UNDEFINED
- Started: 12-Nov-2013 13:54:38
- Elapsed: 38sec
- Tracking URL: [ApplicationMaster](#)
- Diagnostics:

**ApplicationMaster:**

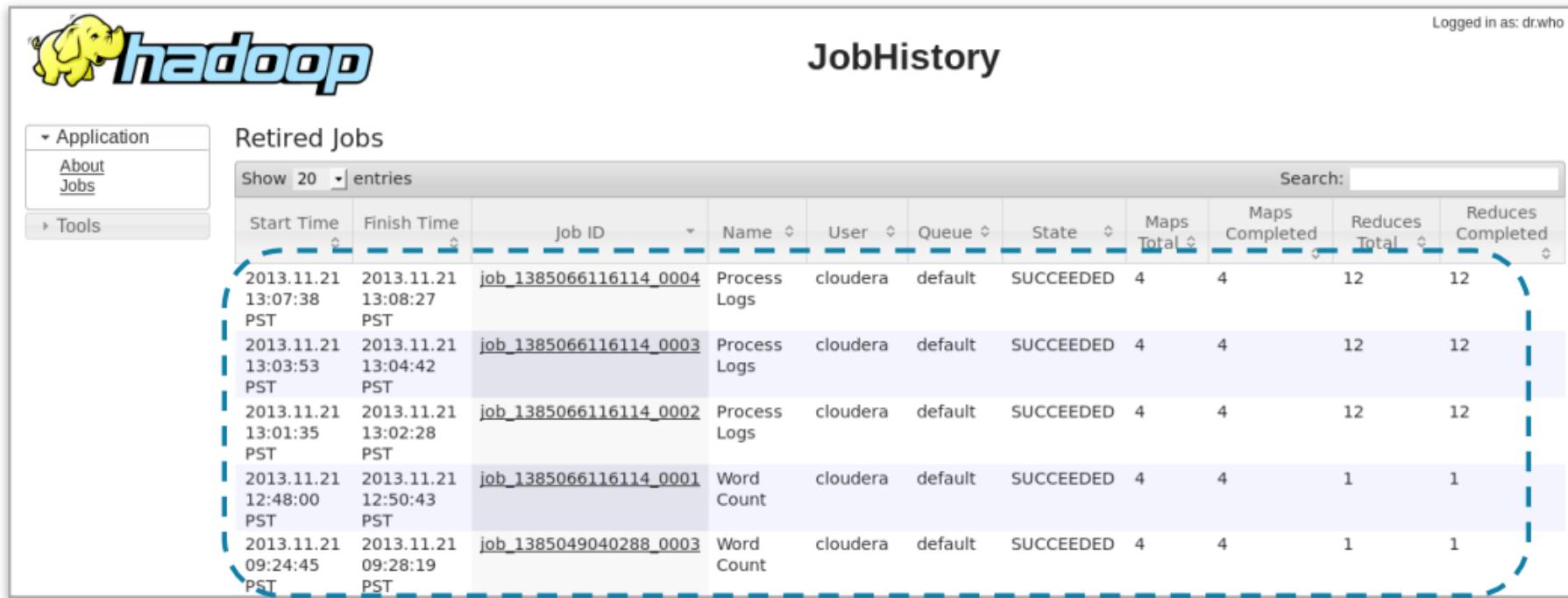
Attempt Number	Start Time	Node	Logs
1	12-Nov-2013 13:54:38	localhost.localdomain:8042	logs

**Annotations:**

- A purple callout box points to the "Tracking URL" field in the Application Overview section, containing the text: "Link to Application Master (UI depends on specific framework)".
- A purple callout box points to the "Logs" column in the ApplicationMaster table, containing the text: "View aggregated log files (optional)".

# Job History Server

- YARN does not keep track of job history
  - Runs on the RM host on port 8080 by default
- Spark and MapReduce each provide their own history server
  - Archive their job details and can be accessed through Job History UI or Hue



The screenshot shows the Hadoop JobHistory UI. At the top left is the Hadoop logo. To its right is the title "JobHistory". In the top right corner, it says "Logged in as: dr:who". On the left, there's a sidebar with "Application" dropdown, "About" link, and a "Tools" button. Below the sidebar is a section titled "Retired Jobs" with a table. The table has columns: Start Time, Finish Time, Job ID, Name, User, Queue, State, Maps Total, Maps Completed, Reduces Total, and Reduces Completed. The table lists five jobs, all of which completed successfully (SUCCEEDED). The first four jobs have "Process Logs" listed under "Name", while the fifth job has "Word Count". All jobs were run by user "cloudera" in the "default" queue. The last job (Word Count) had 4 maps and 1 reduce.

Start Time	Finish Time	Job ID	Name	User	Queue	State	Maps Total	Maps Completed	Reduces Total	Reduces Completed
2013.11.21 13:07:38 PST	2013.11.21 13:08:27 PST	<a href="#">job_1385066116114_0004</a>	Process Logs	cloudera	default	SUCCEEDED	4	4	12	12
2013.11.21 13:03:53 PST	2013.11.21 13:04:42 PST	<a href="#">job_1385066116114_0003</a>	Process Logs	cloudera	default	SUCCEEDED	4	4	12	12
2013.11.21 13:01:35 PST	2013.11.21 13:02:28 PST	<a href="#">job_1385066116114_0002</a>	Process Logs	cloudera	default	SUCCEEDED	4	4	12	12
2013.11.21 12:48:00 PST	2013.11.21 12:50:43 PST	<a href="#">job_1385066116114_0001</a>	Word Count	cloudera	default	SUCCEEDED	4	4	1	1
2013.11.21 09:24:45 PST	2013.11.21 09:28:19 PST	<a href="#">job_1385049040288_0003</a>	Word Count	cloudera	default	SUCCEEDED	4	4	1	1

## YARN Command Line syntax

- Command to configure and view information about the YARN cluster
  - -yarn <command>
  - Most YARN commands are for administrators rather than developers
- Some helpful commands for developers
  - -yarn application
    - Use –list to see running app
    - Use –kill to kill a running app
  - yarn logs –applicationId <app-id>
    - View the logs of the specified app
  - yarn –help
    - View all command options

## Hands-On Session – Exercise 2

### HDFS and Data Ingestion

## Data Ingestion

### Import Flat Files – HUE – Drag and Drop to upload files

- HUE demo
  - Upload a file
  - Rename, Copy, Move, Download, Change Permission
  - Viewing/Edit a file
  - View summary of a file

## Summary of Day 1 - Discussion

**Recap of what we have learned today: -**

- **Data Engineering Fundamentals**
- **Best Practices of ETL/ELT**
- **Hadoop architecture and Ecosystem**
- **Basic Hadoop command line syntax**
- **Basic file ingestions**
- **Yarn resource management**



# Thank You

