

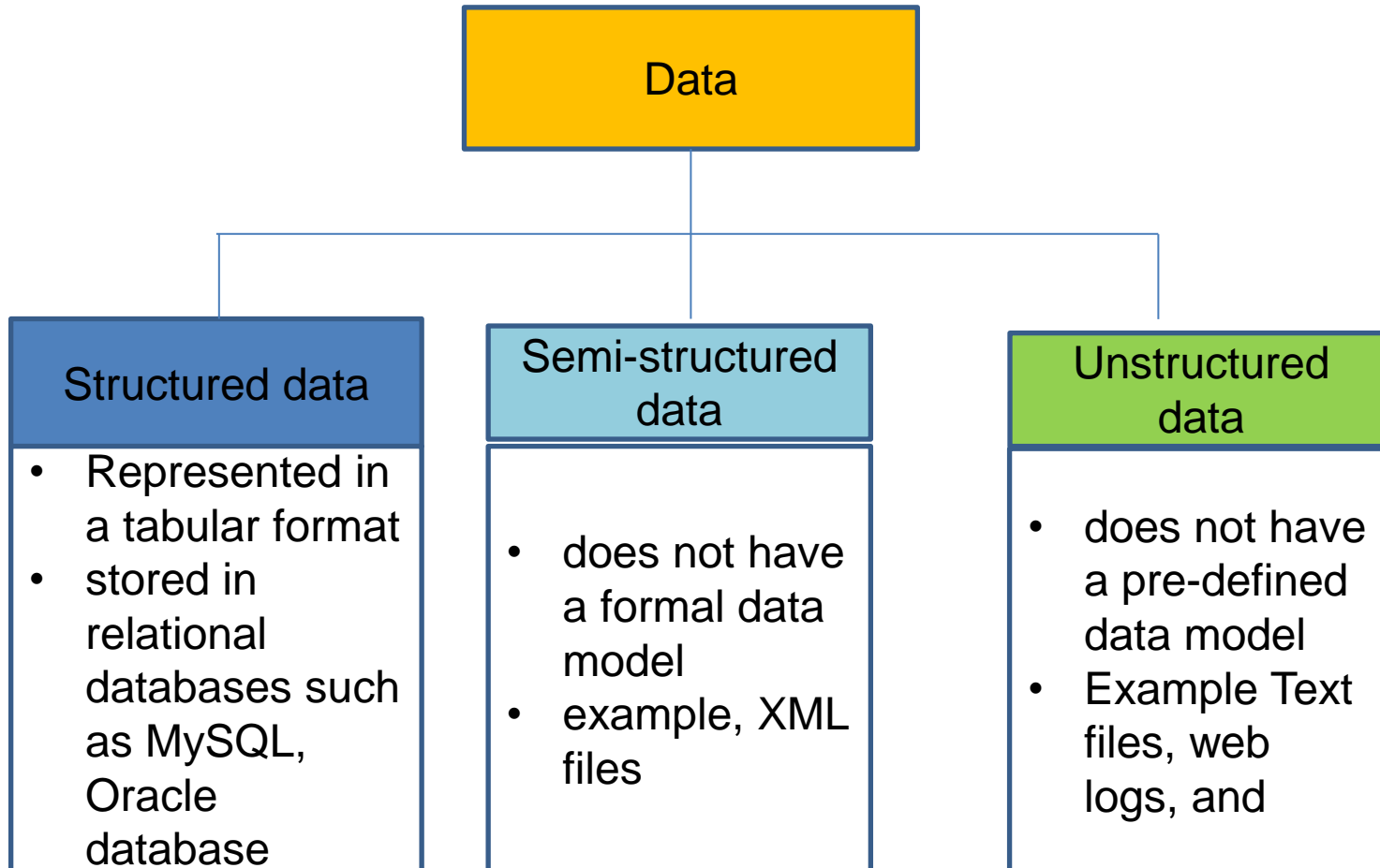
Introduction to Big Data and the Hadoop

Big Data and Hadoop Ecosystem

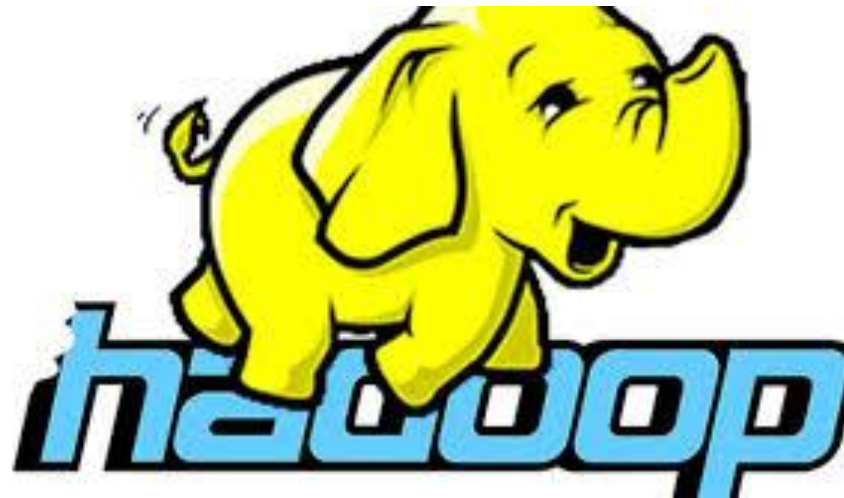
Evolution of Big Data

- Traditionally, computation has been Processor bound
- – Relatively small amounts of data
- – Lots of complex processing
- The early solution: bigger computers
- – Faster processor, more memory
- – But even this couldn't keep up

Types of Data

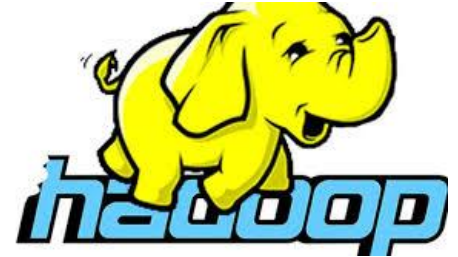


We need a new approach



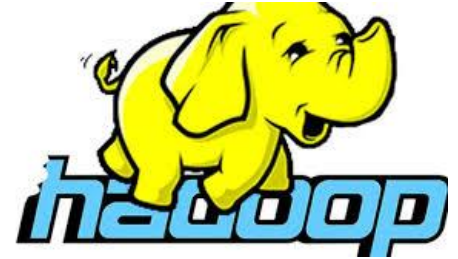
Introducing Hadoop

- A radical new approach to distributed computing
 - Distribute data when the data is stored
 - Run computation where the data is
- Originally based on work done at Google
- Open-source project overseen by the Apache Software Foundation
- Doug Cutting discovered Hadoop and named it after his son's yellow-colored toy elephant.



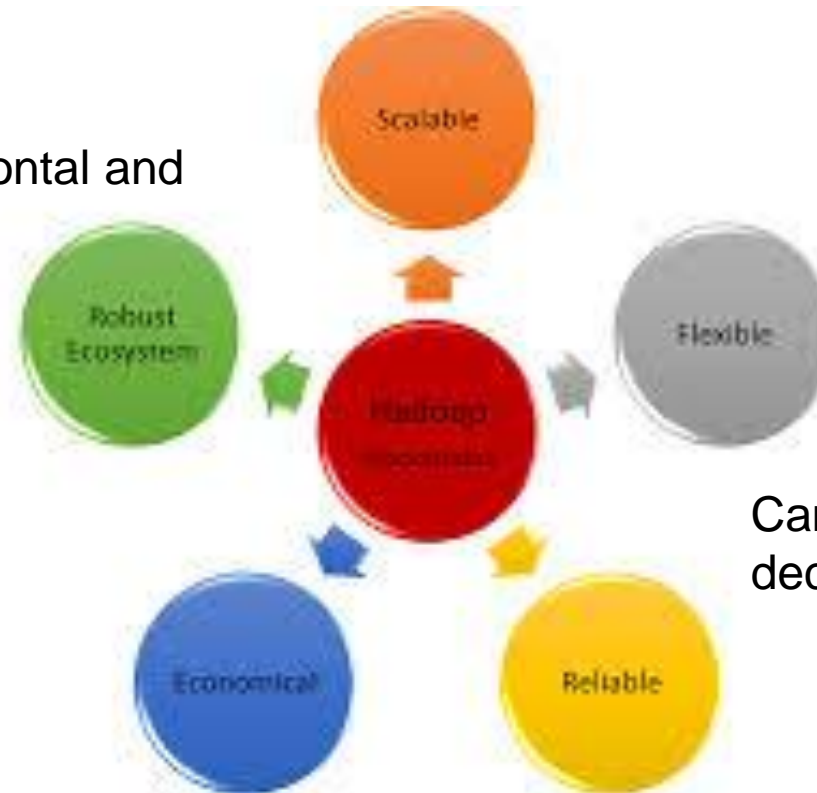
Introducing Hadoop

- Framework for solving data-intensive processes.
- Designed to scale massively
- Processes all the contents of a file (instead of attempting to read portions of a file)
- Hadoop is very fast for very large jobs
- Hadoop is not fast for small jobs
- It does not provide caching or indexing (tools like HBase can provide these features if needed)
- Designed for hardware and software failures



Hadoop Characteristics

Can follow both horizontal and vertical scaling



Can store a lot of data and decide to use it later

Ordinary computers can be used for data processing

Stores copies of the data on different machines and is resistant to hardware failure

Quiz

Do you see any common trend in the following use cases of Hadoop?

- Perception about the organization
- Risk Modelling in Banking domain

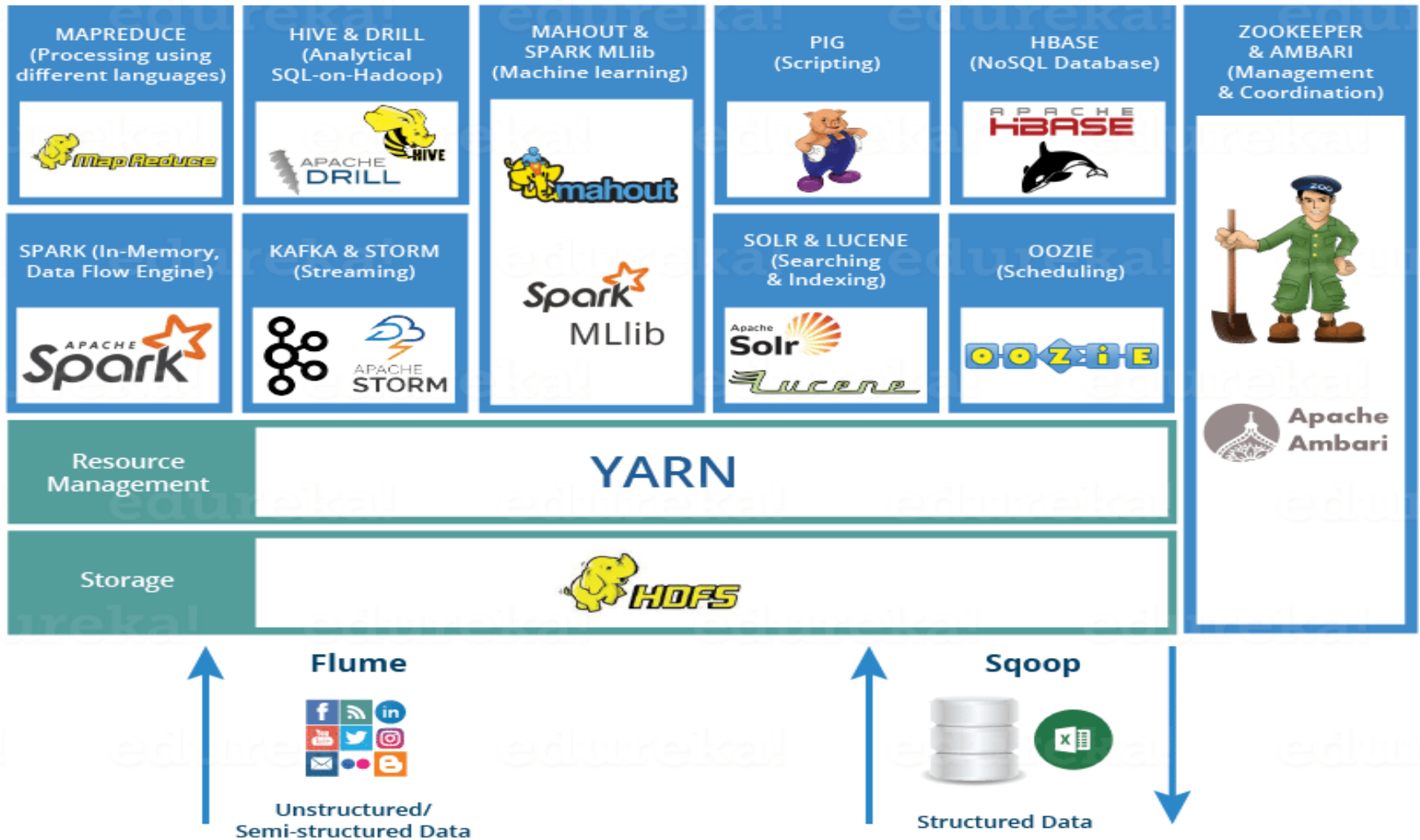
Hadoop-able Problems

- Text mining
- Index building
- Graph creation and analysis
- Pattern recognition
- Collaborative filtering
- Prediction models
- Sentiment analysis
- Risk assessment

Hadoop Components



Hadoop Ecosystem



Hadoop Ecosystem...

A storage layer for Hadoop

Hadoop provides a command line interface to interact with HDFS

Provides file permissions and authentication



Streaming access to file system data

Suitable for the distributed storage and processing

Hadoop Ecosystem...

Stores data in HDFS

A NoSQL database or non-relational database

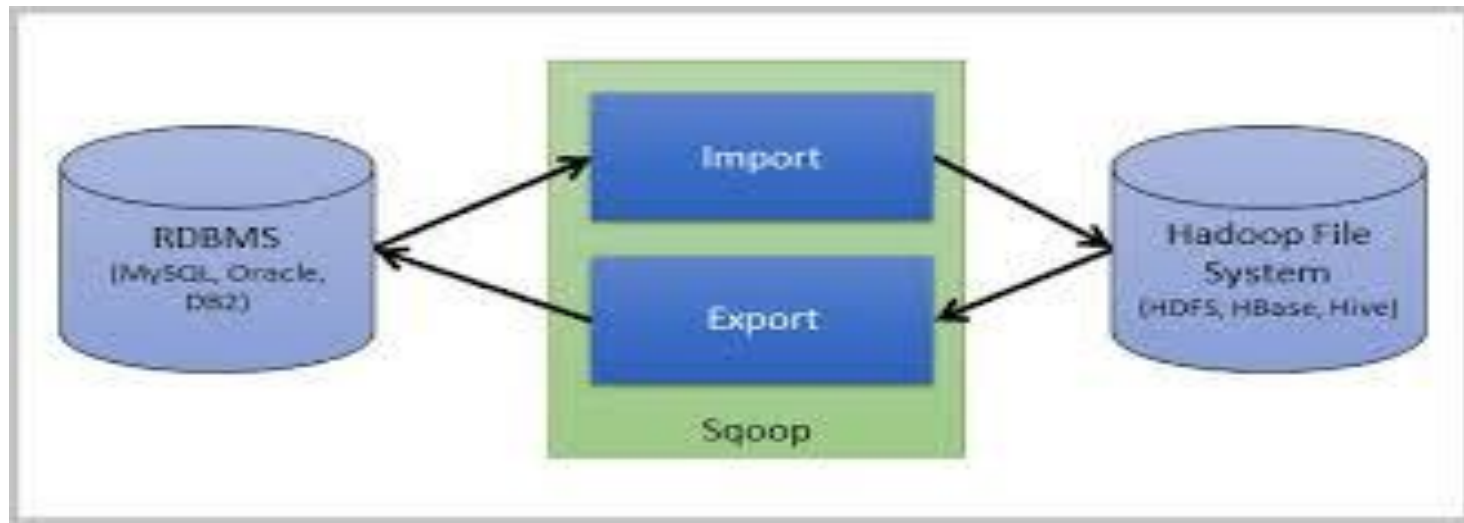
Provides support to high volume of data and high throughput



The table can have thousands of columns

Mainly used when you need random, real-time, read/write access to your Big Data

Hadoop Ecosystem...



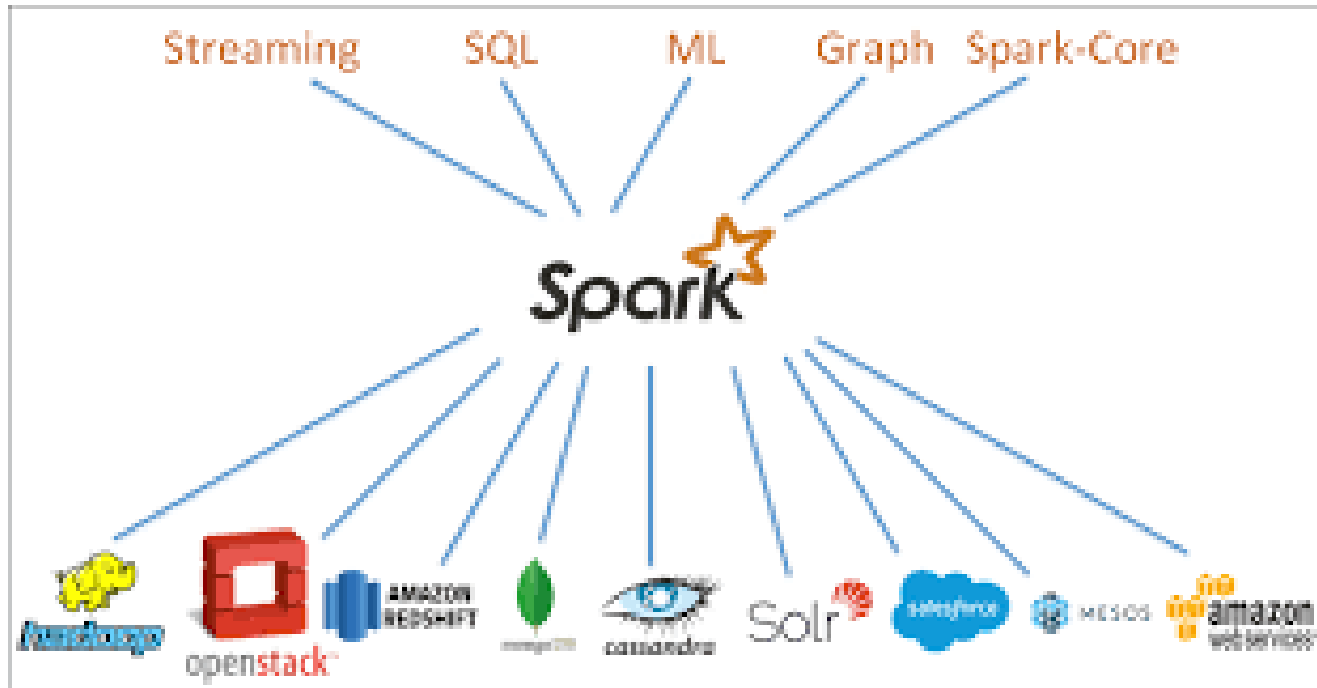
Sqoop is a tool designed to transfer data between Hadoop and relational database servers

Hadoop Ecosystem...



- A distributed service for ingesting streaming data
- Ideally suited for event data from multiple systems

Hadoop Ecosystem...

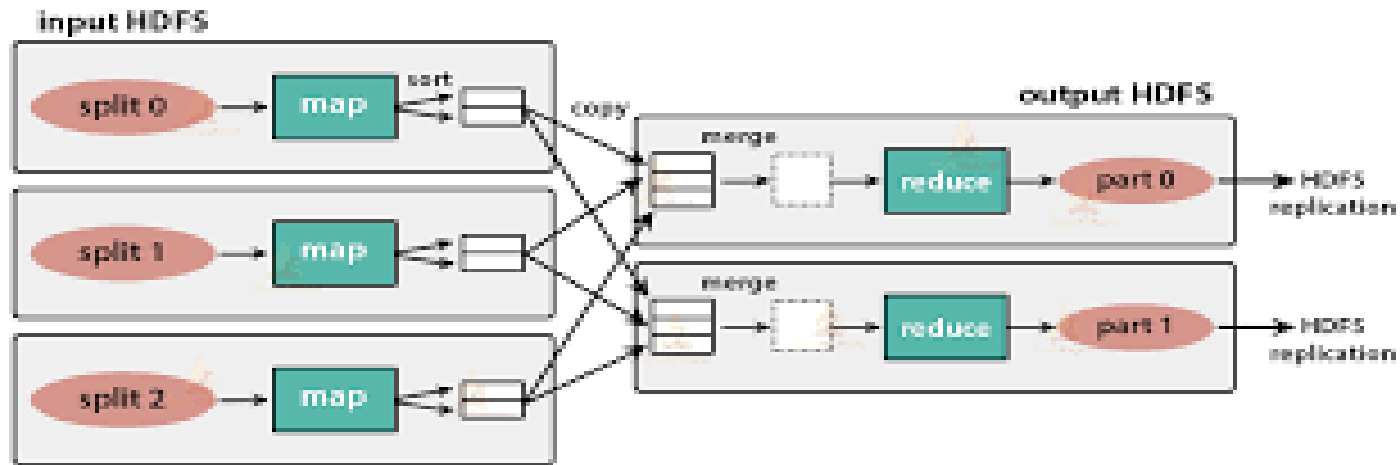


An open-source cluster computing framework

Provides 100 times faster performance than MapReduce

Hadoop Ecosystem...

Apache Hadoop MapReduce



Commonly used

The original Hadoop processing engine which is primarily Java based

An extensive and mature fault tolerance framework

Hadoop Ecosystem...

An open-source dataflow system



Converts pig script to Map-Reduce code

Apache Pig

An alternative to writing Map-Reduce code

Hadoop Ecosystem...

High performance SQL engine which runs on Hadoop cluster



Supports a dialect of SQL (Impala SQL)

Very low latency –measured in milliseconds

Hadoop Ecosystem...

Executes queries using MapReduce



Supports a dialect of SQL (HQL)

Similar to Impala

Hadoop Ecosystem...

Oozie is a workflow or coordination system used to manage the Hadoop jobs

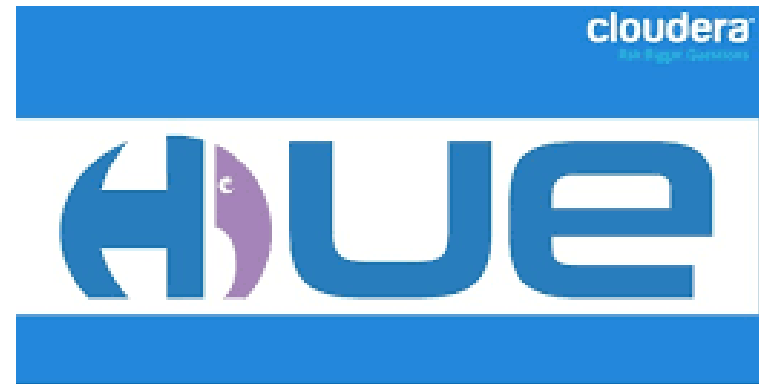


Hadoop Ecosystem...

Hue is an acronym for Hadoop User Experience

Hue is an open source Web interface for analysing data with Hadoop

It provides SQL editors for Hive, Impala, MySQL, Oracle, PostgreSQL, Spark SQL, and SolrSQL



The Hadoop UI

Romain Rigaux - YIHUG Aug 2013

Hadoop Security

Kerberos Authentication

All services can be principals

HBase, Hive & HDFS Authorization

ACLs for each

Wire encryption

HDFS, Shuffle, JDBC

Basic Auditing

Logging of events

Knox: Hadoop REST Gateway

Perimeter security for Hadoop cluster

Centralized Security Administration

Centralized Audit Reporting

Delegated Policy Administration

XA Secure

RDBMS vs Hadoop

Relational

Hadoop

Required on write	schema	Required on read
Reads are fast	speed	Writes are fast
Standards and structured	governance	Loosely structured
Limited, no data processing	processing	Processing coupled with data
Structured	data types	Multi and unstructured
Interactive OLAP Analytics Complex ACID Transactions Operational Data Store	best fit use	Data Discovery Processing unstructured data Massive Storage/Processing

Apache Hadoop - Use cases

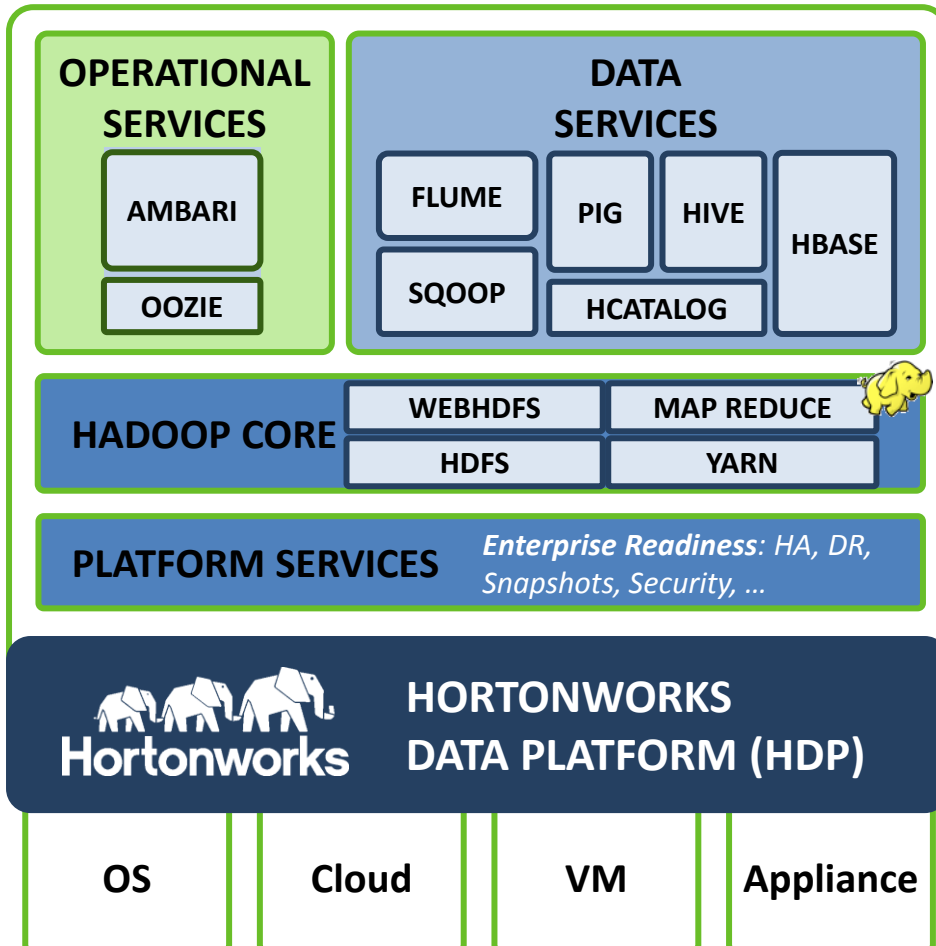


Cloudera Distribution Apache Hadoop



- Founded by leading experts on Hadoop from Facebook, Yahoo, Google and Oracle.
- Provides support, consulting, training and certification for Hadoop users.
- Staff includes committers to virtually all Hadoop projects.
- Many authors of industry standard books on Apache Hadoop projects
 - Tom White, Lars George, Kathleen Ting etc
- Customers includes many ke users of Hadoop
 - Altstate, AOL Advertising, Box, CBS Interactive, eBay, Experian, Groupon, National Cancer Institute, Orbitz, Social Security Administration, Trend Micro, Truila, US Army....

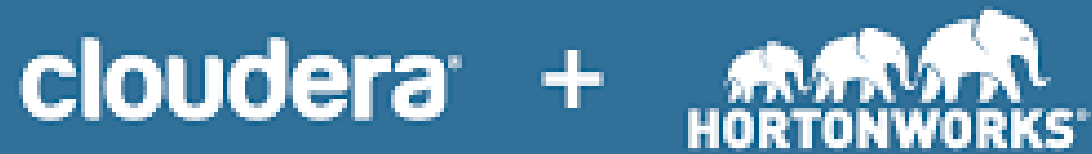
Hortonwork



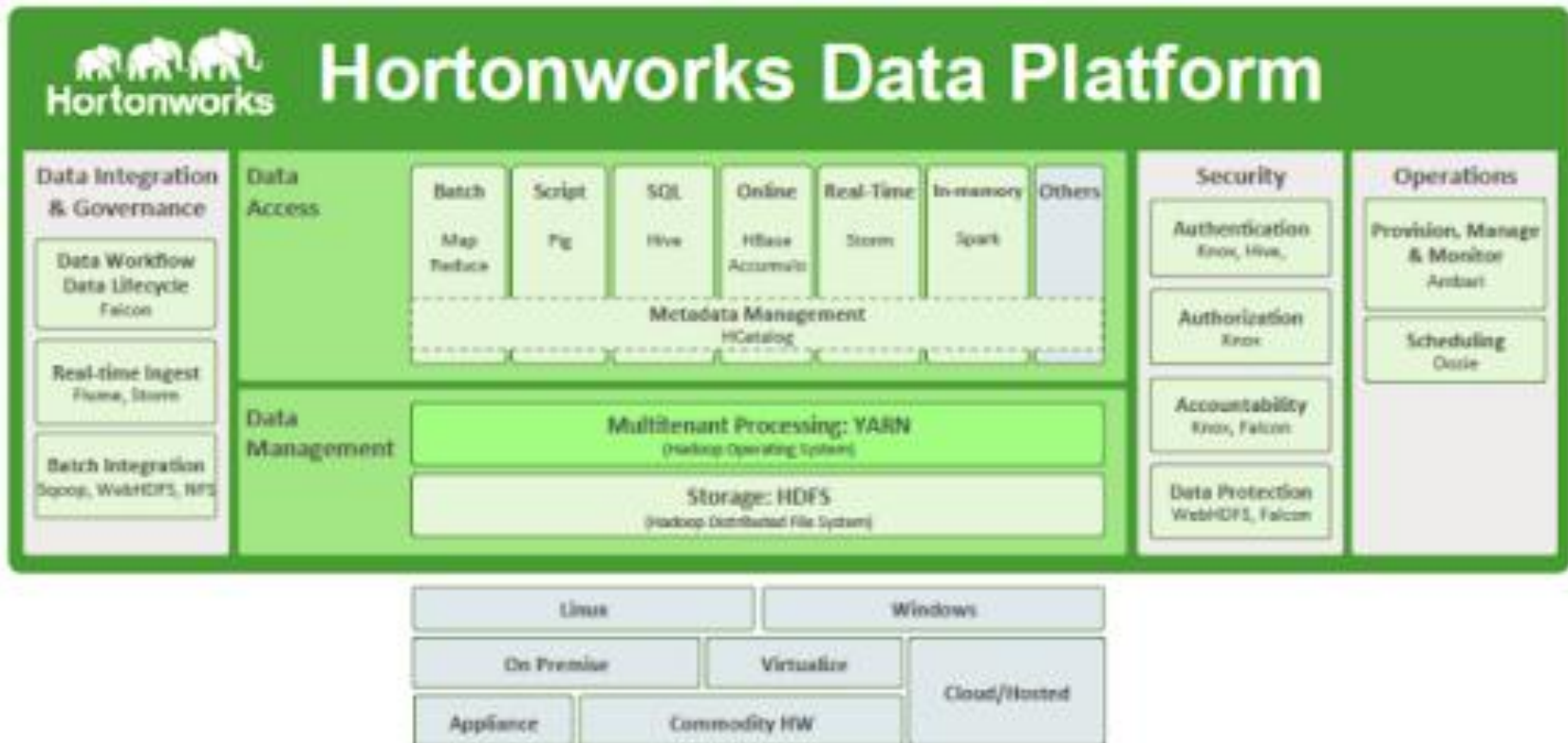
Enterprise Hadoop

- The **ONLY** 100% open source and complete distribution
- Enterprise grade, proven and tested at scale
- Ecosystem endorsed to ensure interoperability

Hortonwork



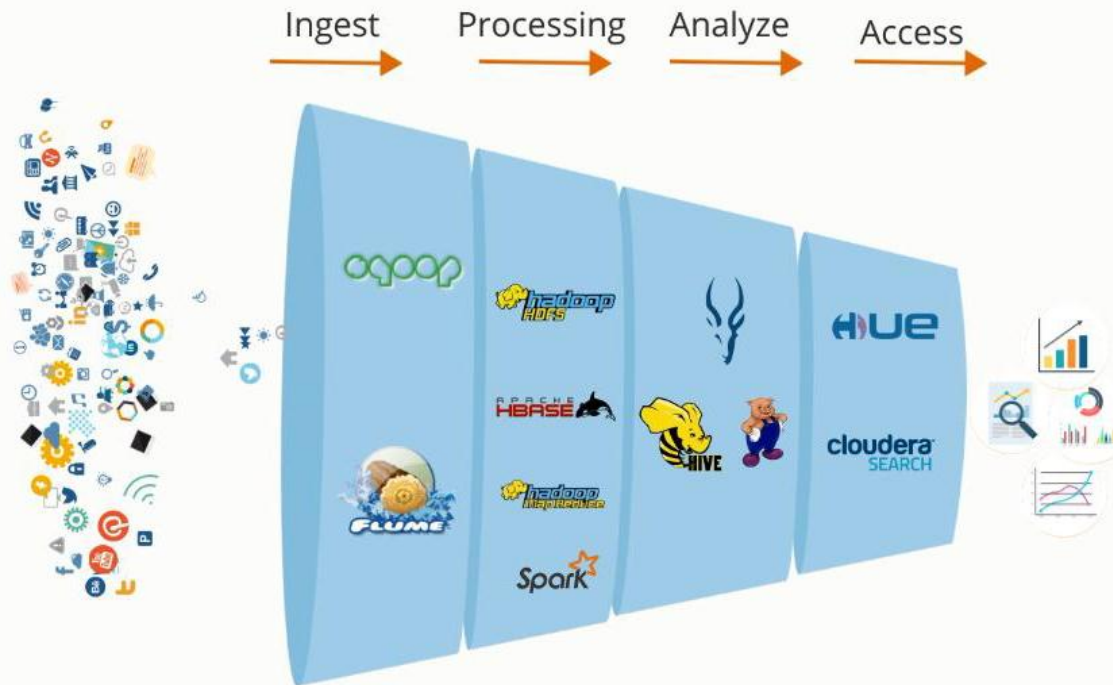
Hortonwork



Big Data Processing

Four stages of Big Data

Components of the Hadoop ecosystem work together to process Big Data

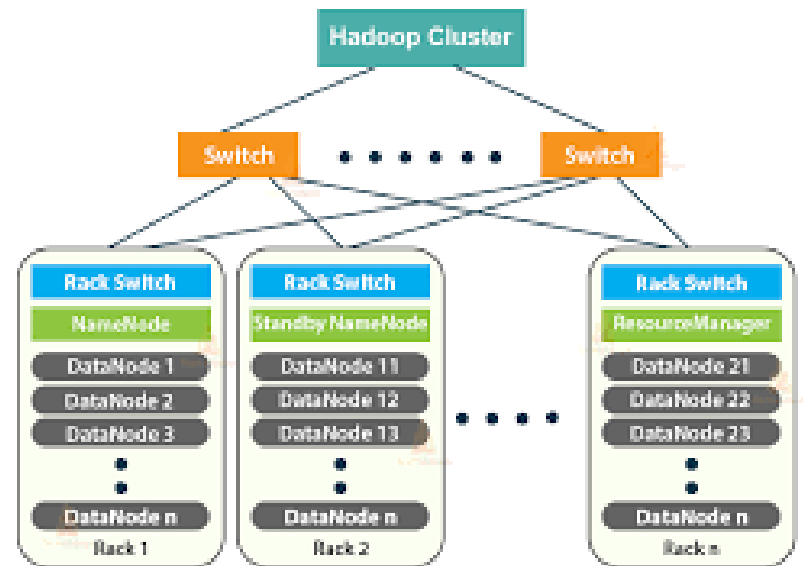




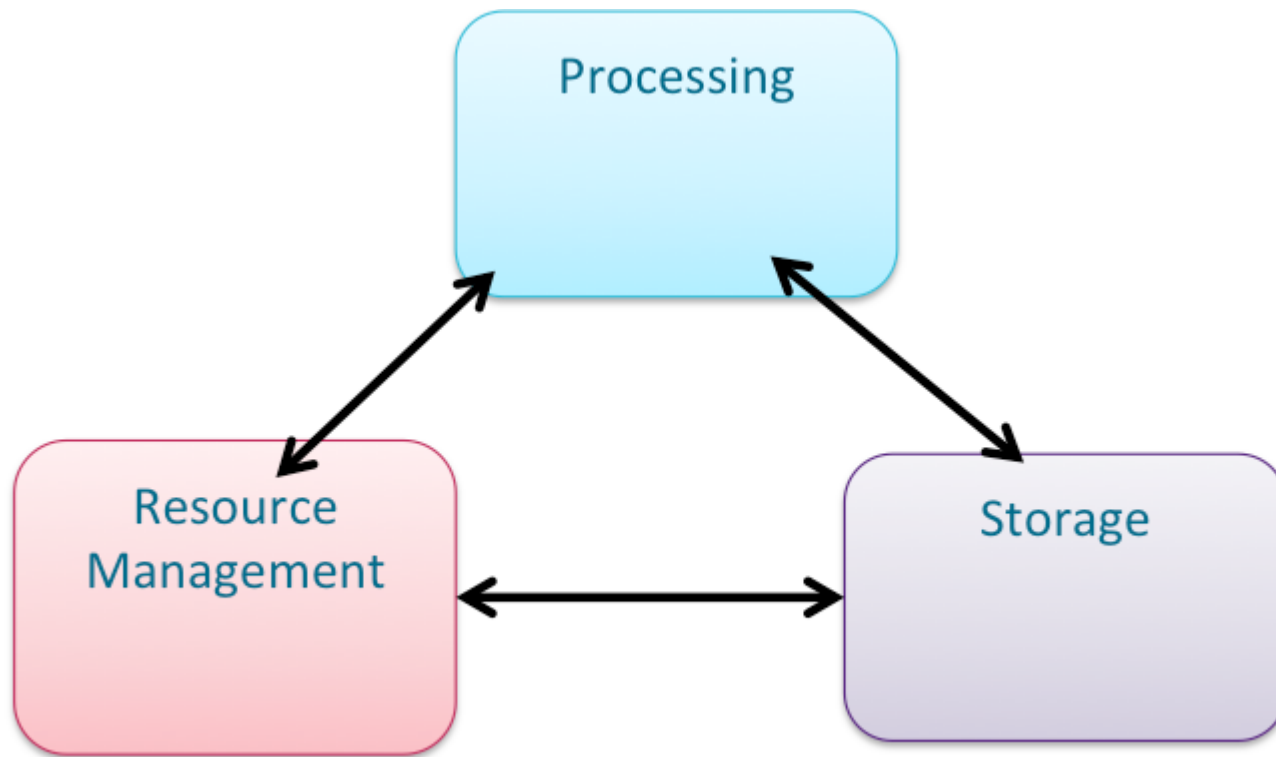
HADOOP DISTRIBUTED FILE SYSTEM

Hadoop Terminology

- A cluster is a group of computers working together
- A node is an individual computer in the cluster
- Daemons are long---running processes that typically start up when the machine starts and listens for requests from client processes.

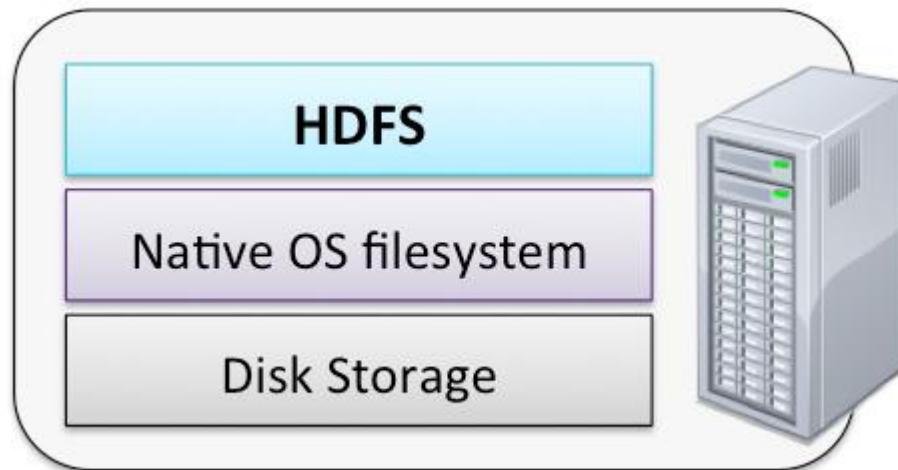


Cluster Components



What is HDFS

- HDFS is a distributed file system .
- It provides great access to data across Hadoop clusters.
- Based on GFS
- Sits on top of native file system
- Provides redundant storage for massive amount of data.
- Using readily available , industry standard computers.



What is HDFS

HDFS operates best with a 'modest' number of large files

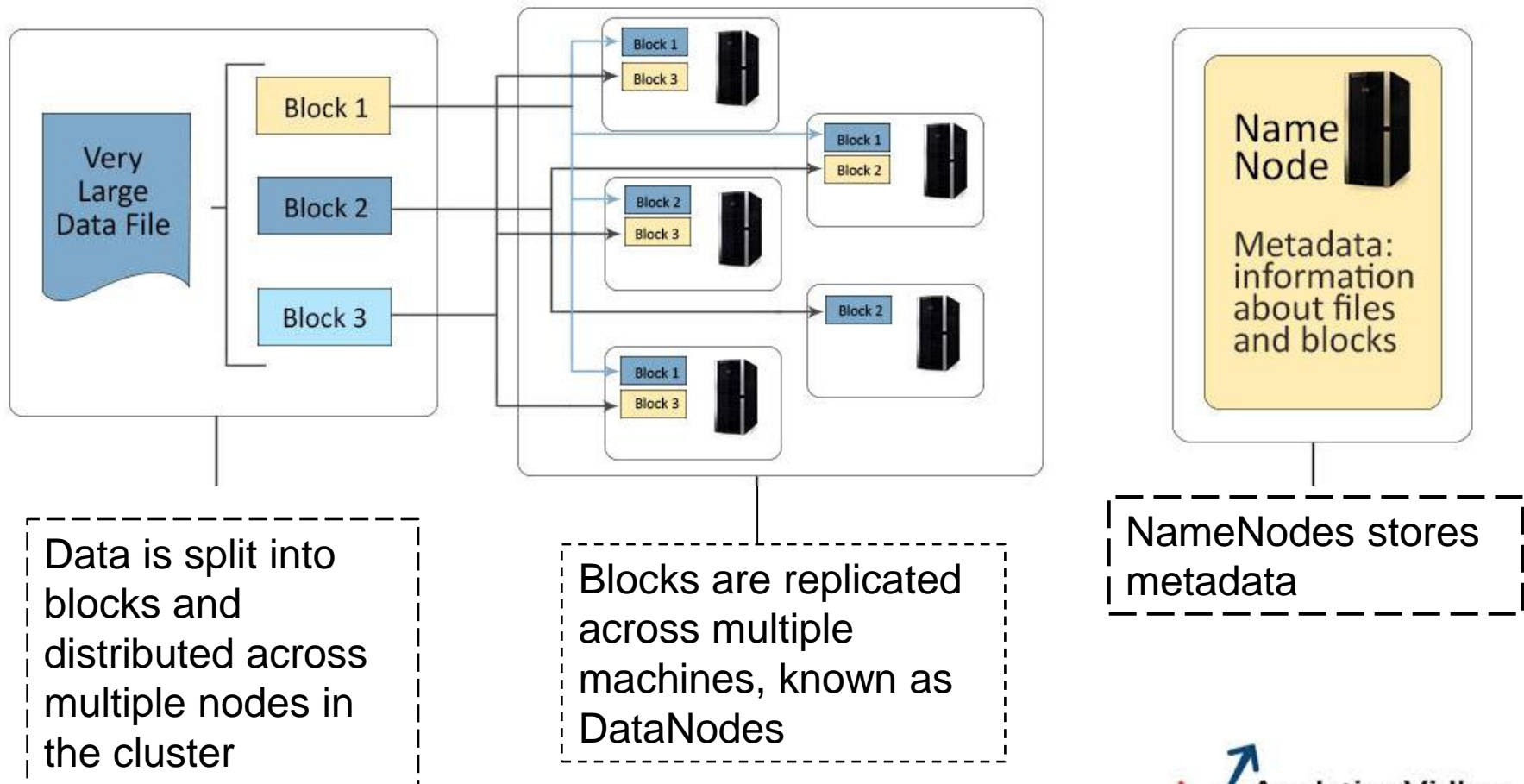
- Millions, rather than billions, of files
- Each file size typically 100MB or more

Files in HDFS are 'write once'

- No random writes to files are allowed

HDFS is optimized for large, streaming reads of files

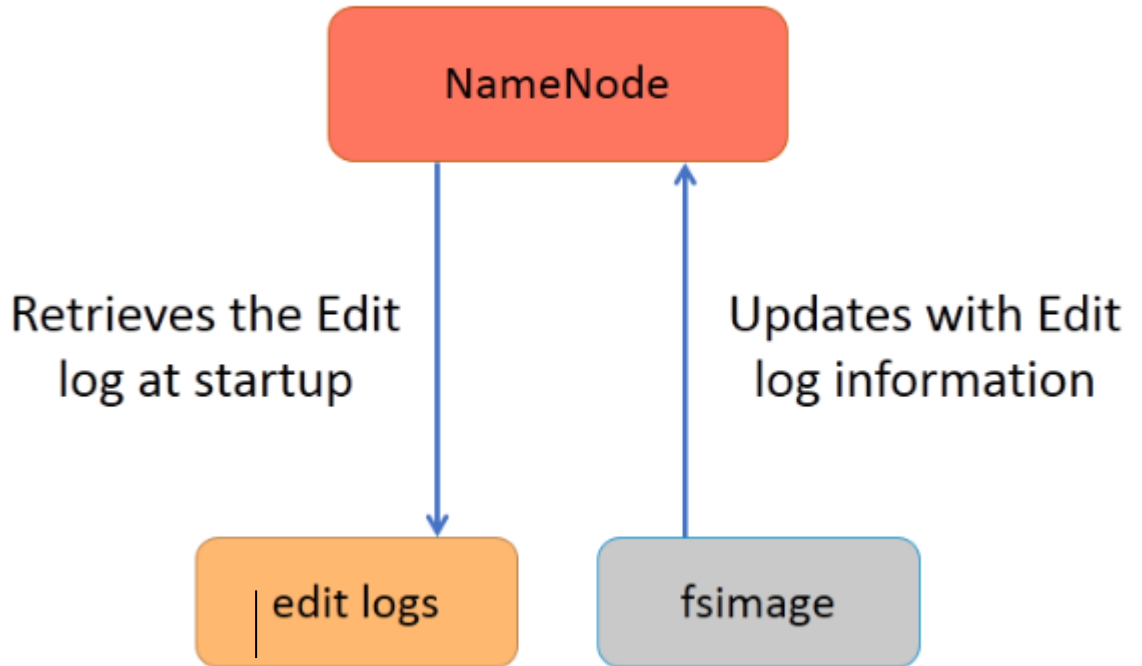
How files are stored in HDFS



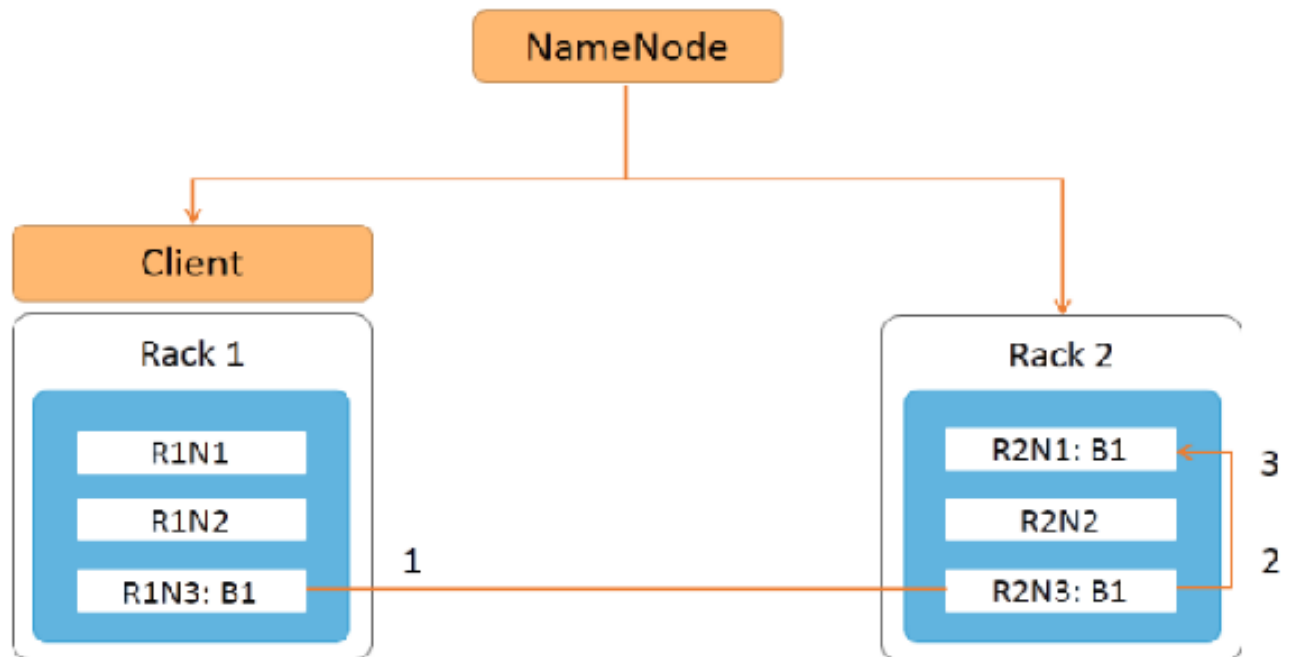
How files are stored in HDFS

The NameNode maintains two persistent files

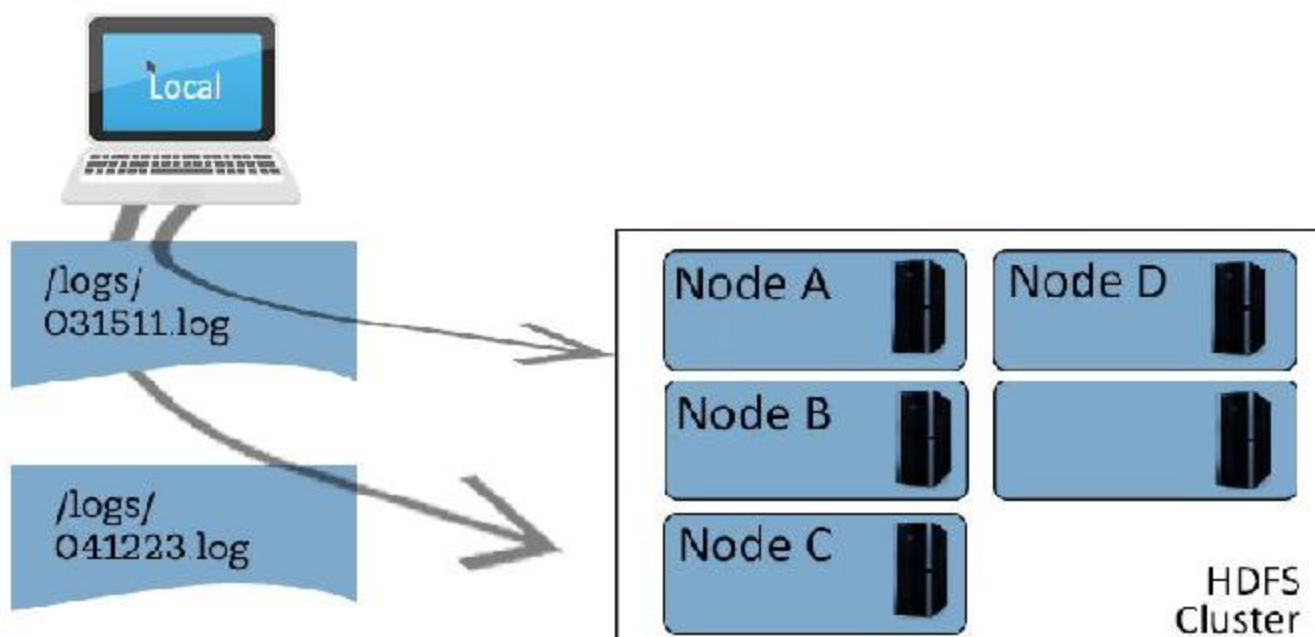
- a transaction log called an **Edit Log**
- a namespace image called an **FsImage**.



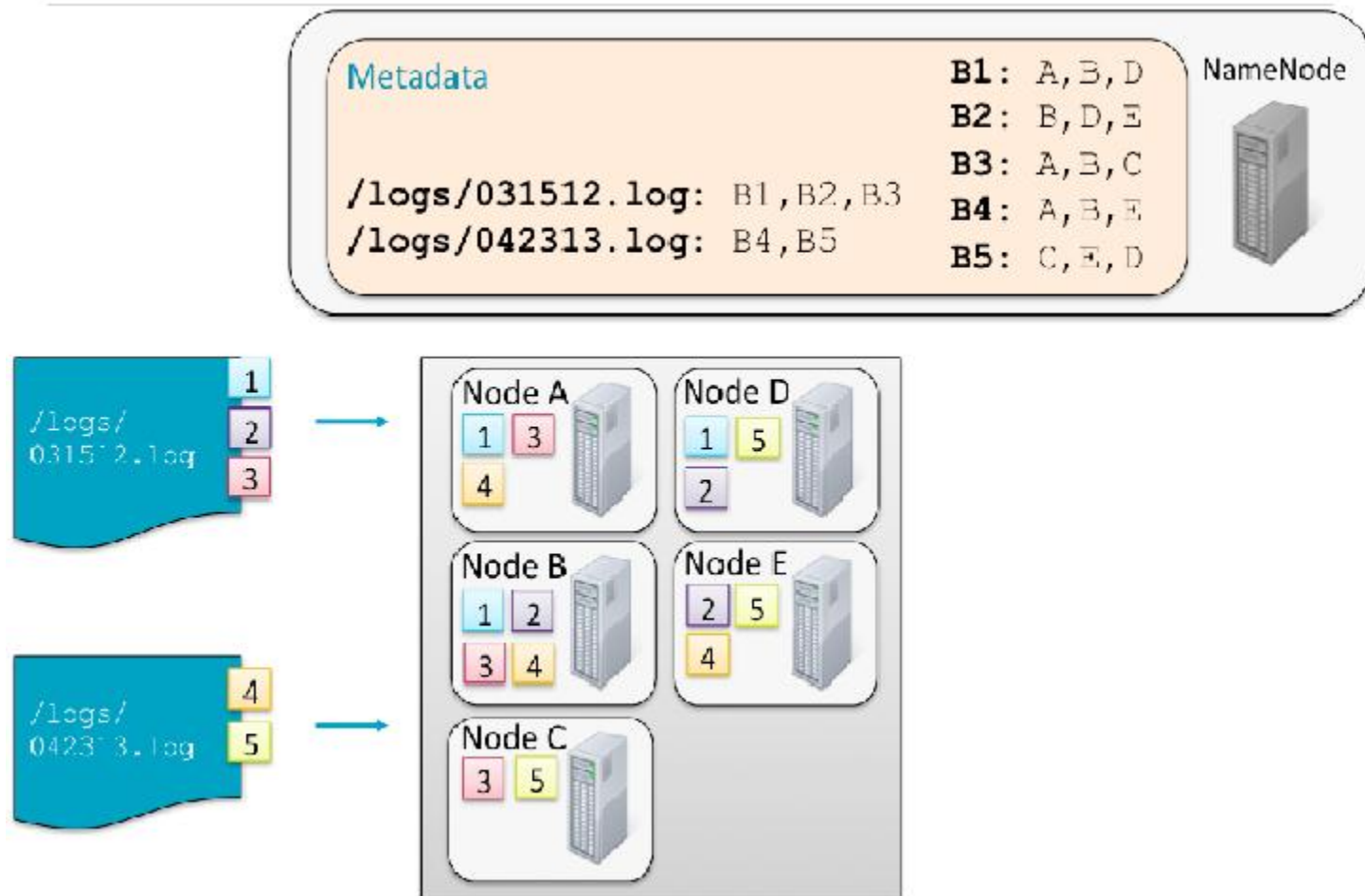
Data Replication Topology



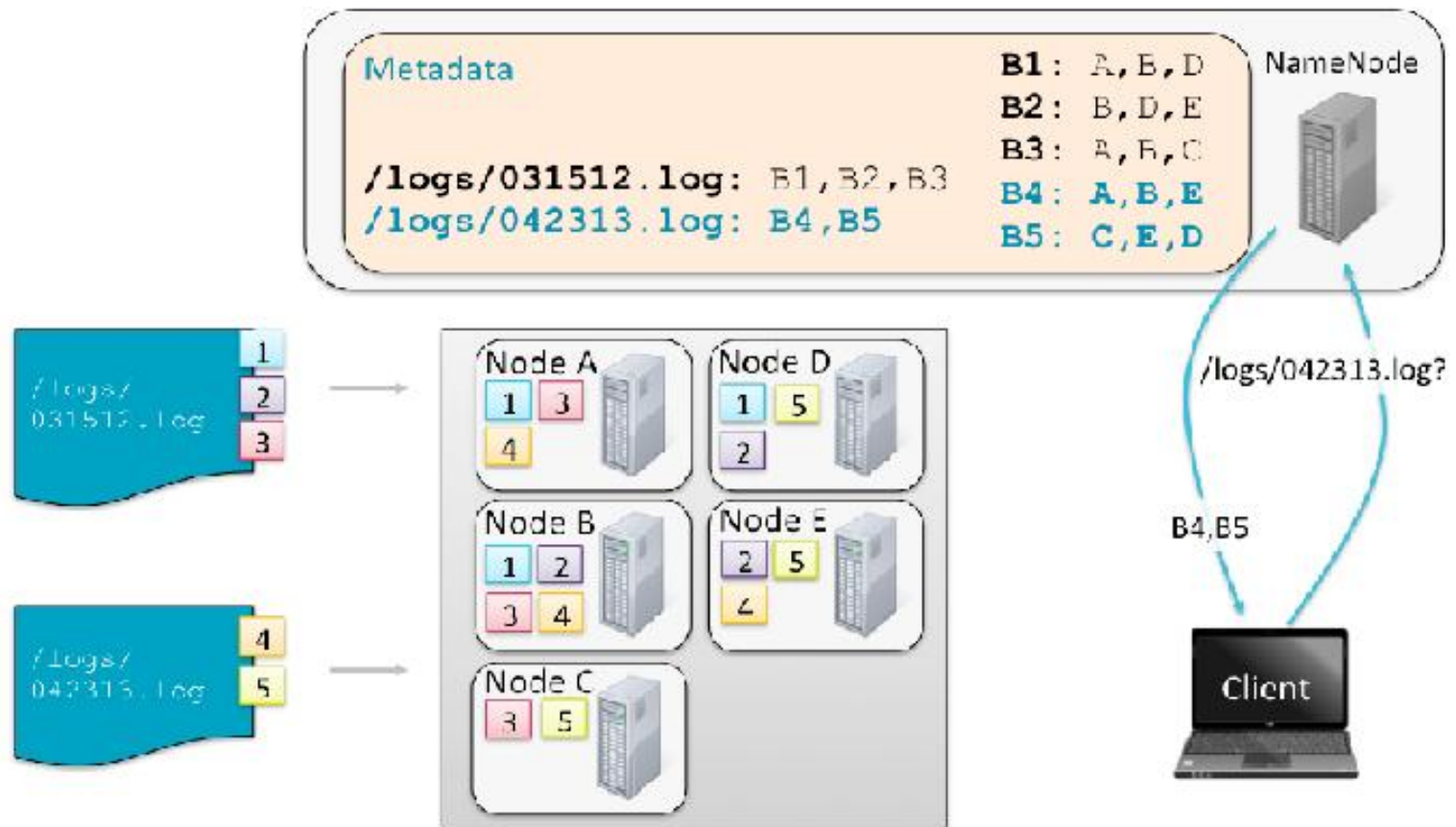
Storing and retrieving files

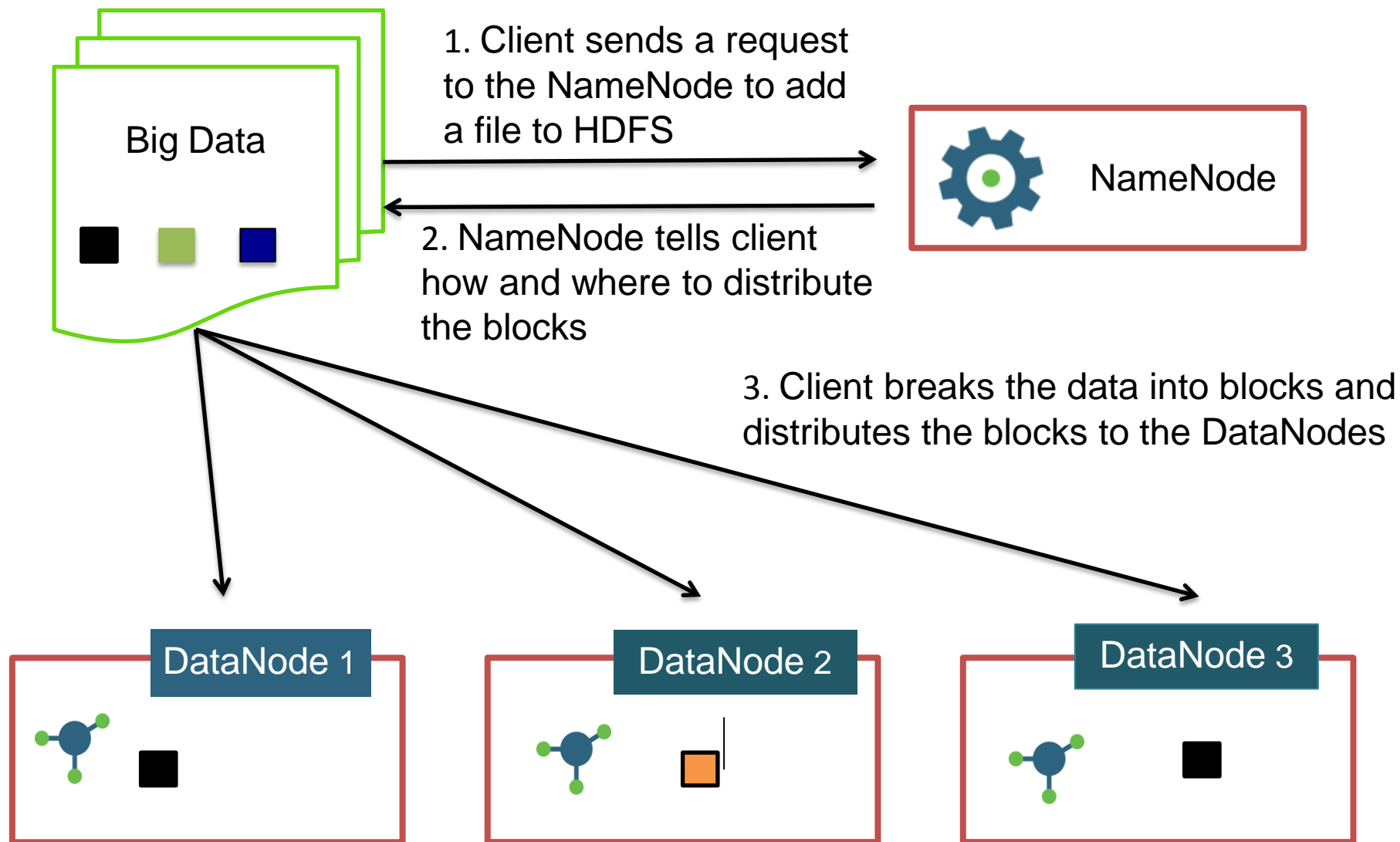


Storing and retrieving files



Storing and retrieving files





Accessing HDFS



HDFS Commands

```
hdfs fs -command [args]
```

Here are a few (of the almost 30) HDFS commands:

- `-cat`: display file content (uncompressed)
- `-text`: just like cat but works on compressed files
- `-chgrp`, `-chmod`, `-chown`: changes file permissions
- `-put`, `-get`, `-copyFromLocal`, `-copyToLocal`: copies files from the local file system to the HDFS and vice-versa. Two versions.
- `-ls`, `-ls -R`: list files/directories
- `-mv`, `-moveFromLocal`, `-moveToLocal`: moves files
- `-stat`: statistical info for any given file (block size, number of blocks, file type, etc.)

HUE Browser

Search for file name

Actions Move to trash

Home / user / training History Trash

Name	Size	User	Group	Permissions	Date
↑		hdfs	supergroup	drwxrwxrwx	March 03, 2015 11:44 AM
.		training	supergroup	drwxrwxrwx	February 25, 2015 01:48 PM

Show 45 of 0 items Page 1 of 1

HDFS NameNode Availability

- Files and directories belong to a Namespace
- A NameNode manages a single Namespace Volume
- **The NameNode daemon must be running at all times**

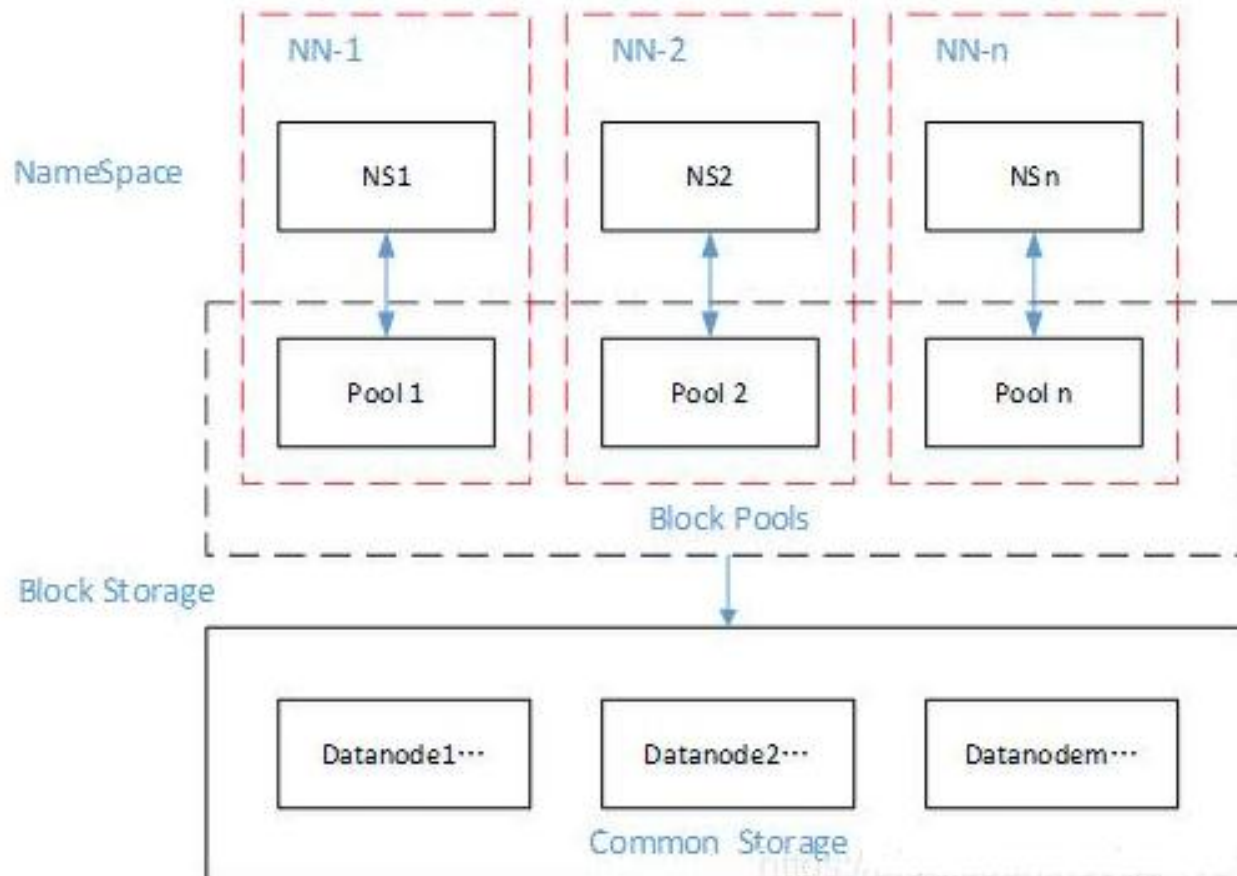
Two NameNodes:

- Active
- Standby Node

➤ **High Availability mode**

- One NameNode
- One “helper” node called Secondary NameNode for Bookkeeping, not backup.

HDFS Federation

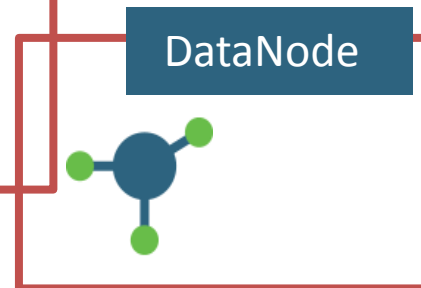
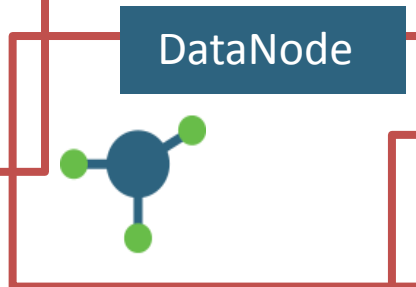
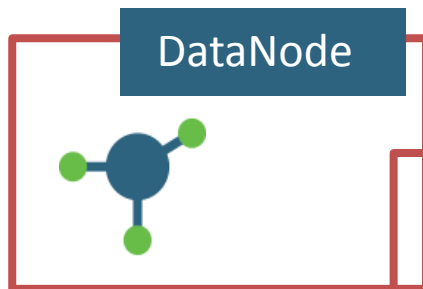




1. Each DataNode registers with all the NameNodes in the cluster

2. DataNodes send periodic heartbeats and block reports to all NameNodes

3. Any NameNode can send a command to any DataNode



NameNode 1

Active

All Namespace modifications are logged durably to a majority of the JournalNode daemons.

NameNode 2

Standby

The Standby node is constantly reading the changes and applying them to its Namespace

All client operations are handled by the Active node



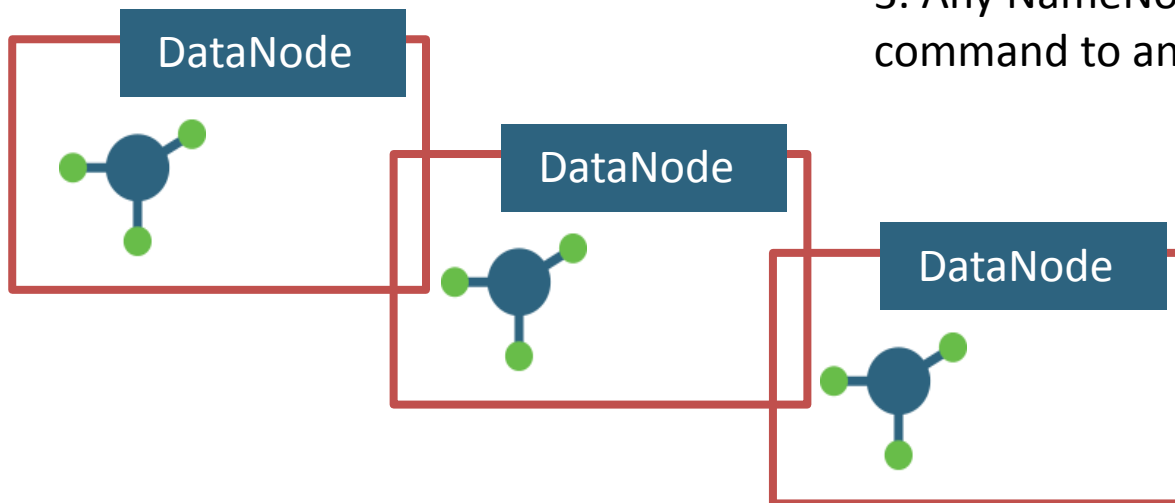
A set of JournalNode daemons



1. Each DataNode registers with all the NameNodes in the cluster

2. DataNodes send periodic heartbeats and block reports to all NameNodes

3. Any NameNode can send a command to any DataNode



HDFS Rebalance

A cluster is in a balanced status when,

- % of space used in each data node is within limits of Average % of space used on data nodes \pm Threshold size .
- Percentage space used on a data node should not be less than Average % of space used on data nodes – Threshold size.
- Percentage space used on a data node should not be greater than Average % of space used on data nodes + Threshold size.

Here Threshold size is configurable value which is 20 % of used spaced by default.

HDFS Rebalance

- Rebalancer is a administration tool in HDFS, to balance the distribution of blocks uniformly across all the data nodes in the cluster.
- Rebalancing will be done on demand only. It will not get triggered automatically.

```
hadoop1@ubuntu-1:~$ hdfs balancer
Time Stamp          Iteration#  Bytes Already Moved  Bytes Left To Move  Bytes Being Moved
The cluster is balanced. Exiting...
Balancing took 1.148 seconds
hadoop1@ubuntu-1:~$
```

HDFS Rebalance

If a Rebalancer is triggered, NameNode will scan entire data node list and when

- Under-utilized data node is found, it moves blocks from over-utilized data nodes or not-under-utilized data nodes to this current data node
- If Over-utilized data node is found, it moves blocks from this data node to other under-utilized or not-over-utilized data nodes.
- Spreads different replicas of a block across the racks so that cluster can survive loss of an entire rack.
- One of the replicas is placed on the same rack as the node writing to the file so that cross-rack network I/O is reduced.
- Spread HDFS data uniformly across the DataNodes in the cluster.



Yet Another Resource Negotiator

What is YARN

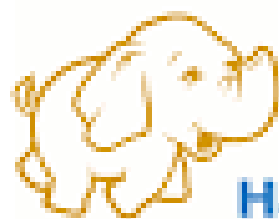
- **YARN** is a resource manager.
- It was created by separating the processing engine and the management function of MapReduce.
- It monitors and manages workloads, maintains a multi-tenant environment, manages the high availability features of Hadoop, and implements security controls
- YARN allow multiple processing engines

Batch Processing

Stream Processing

Advanced Analytics

Before YARN



Hadoop v1.0

MapReduce

Data Processing
& Resource Management

HDFS

Distributed File Storage



Hadoop v2.0

MapReduce

**Other Data
Processing
Frameworks**

YARN

Resource Management


HDFS

Distributed File Storage

YARN Daemons

- **Resource Manager (RM)**

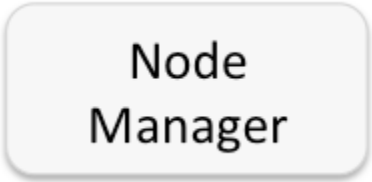
- Runs as Master node
- Global Resource Manager
- Arbitrates system resources between competing applications
- has a pluggable scheduler to support different algorithms

A light purple rounded rectangle with a thin black border and a subtle drop shadow. It contains the text "Resource Manager" in a black, sans-serif font, centered within the box.

Resource
Manager

- **Node Manager (NM)**

- Runs on slave nodes
- communicates with RM
- Node health
- Reporting node and container status to the ResourceManager

A light gray rounded rectangle with a thin black border and a subtle drop shadow. It contains the text "Node Manager" in a black, sans-serif font, centered within the box.

Node
Manager

Run an application on YARN

- **Application Master (AM)**

- One per application
- Framework/app specific
- Runs in a container
- Requests more containers to run application tasks
- manages the application lifecycle, dynamic adjustments to resource consumption, execution flow, faults, and provides status and metrics

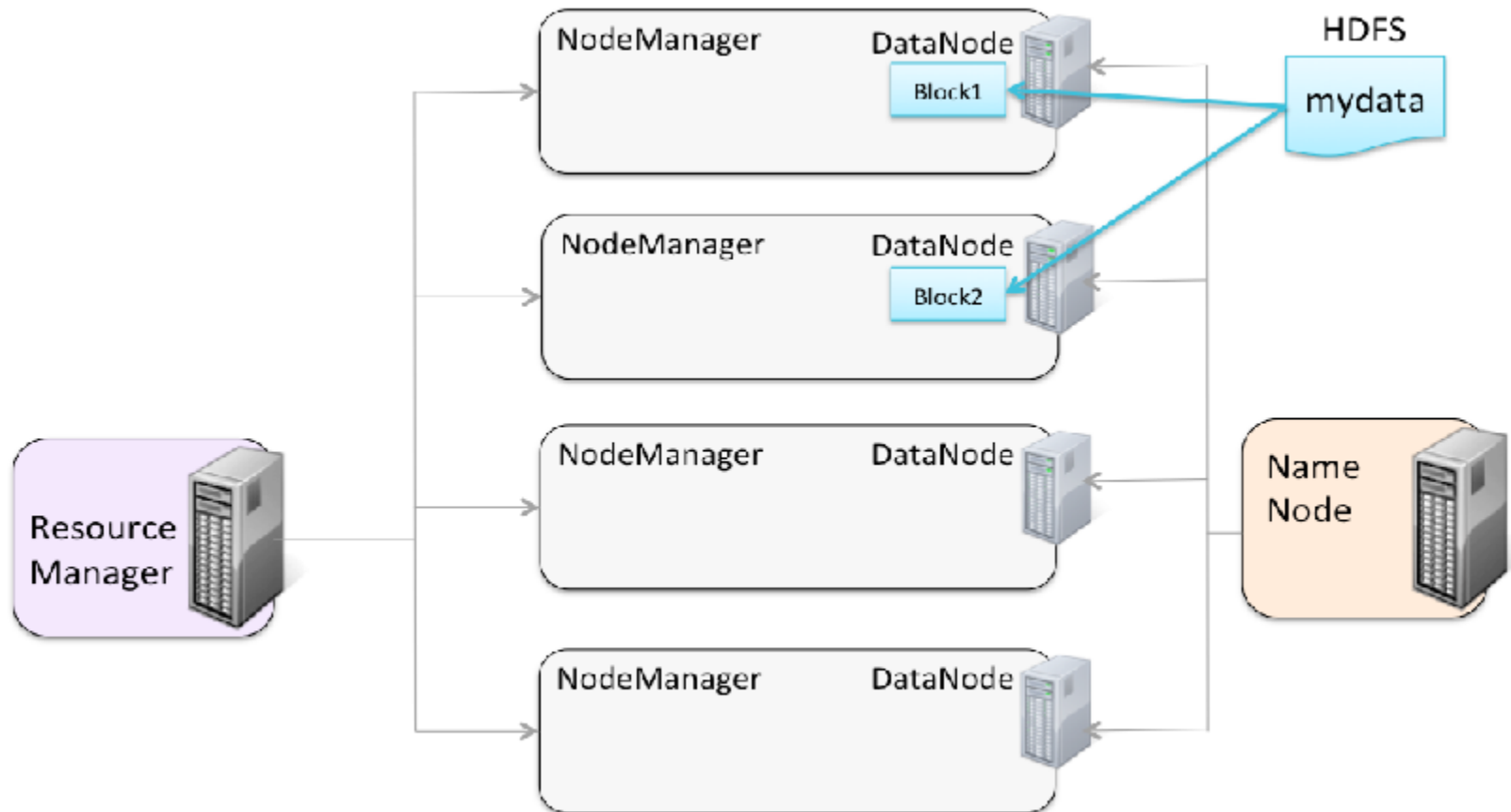


- **Containers**

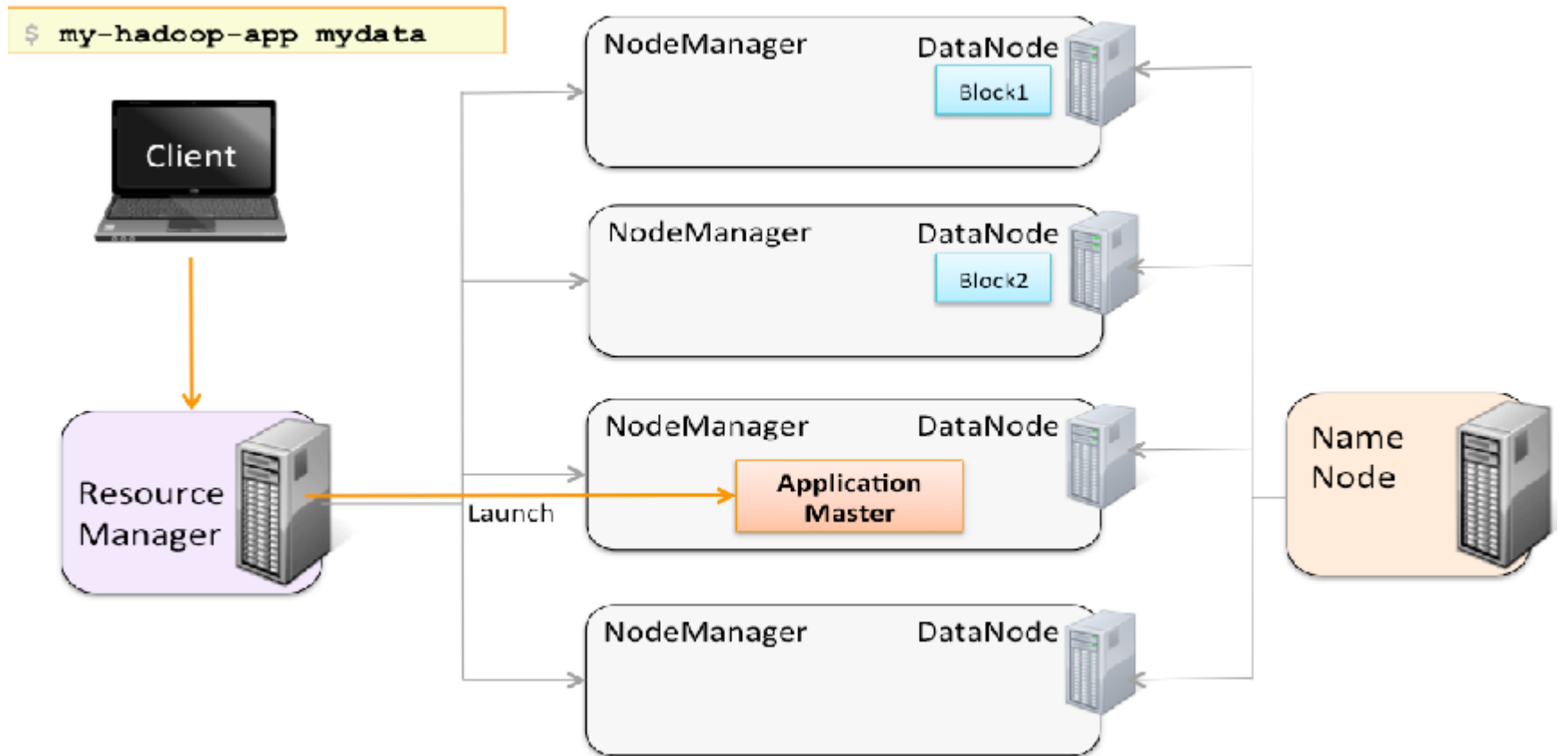
- created by the RM on request
- Allocate a certain amount of resources on the slave nodes
- Applications run in one or more containers



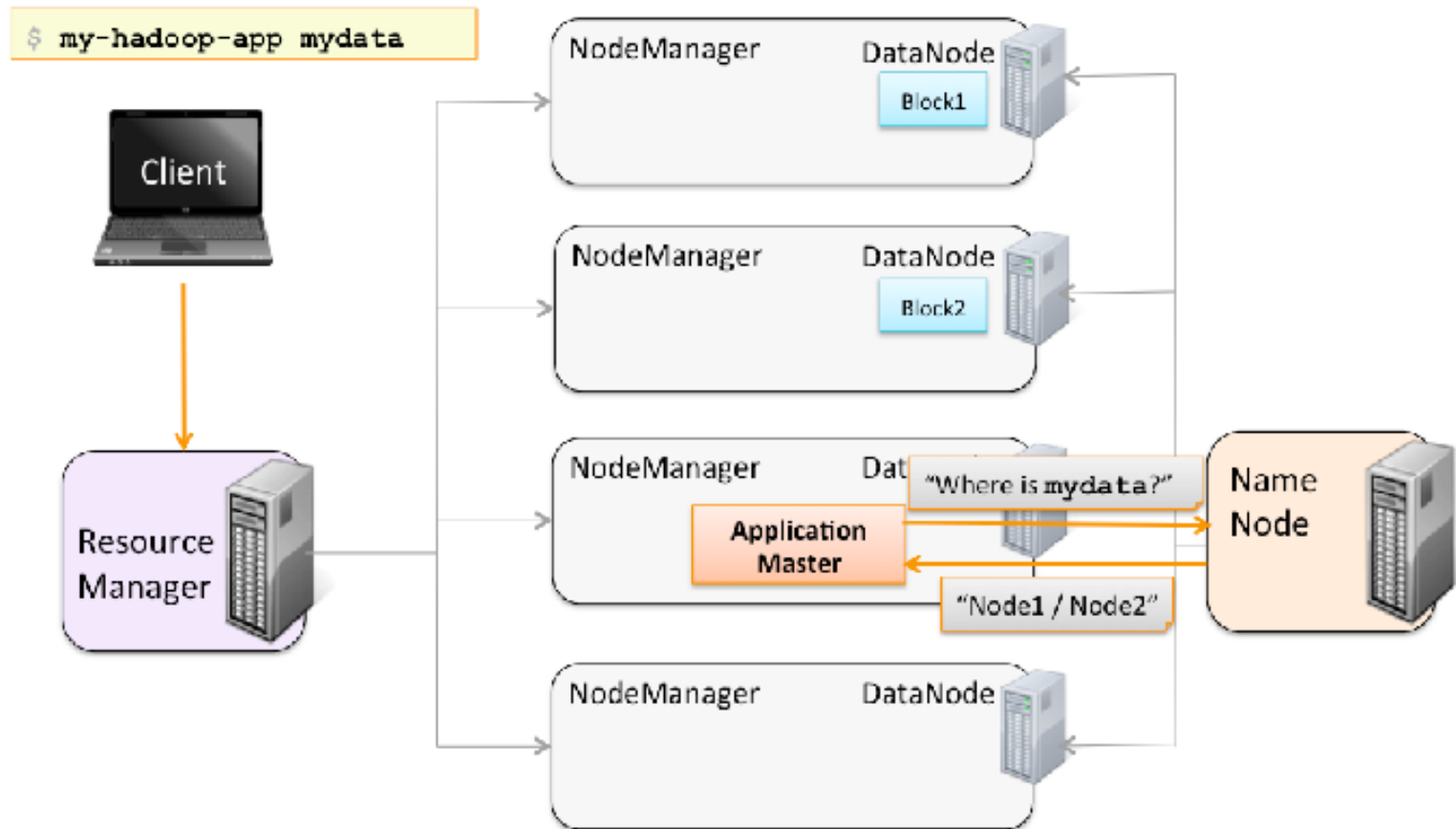
Run an application on YARN



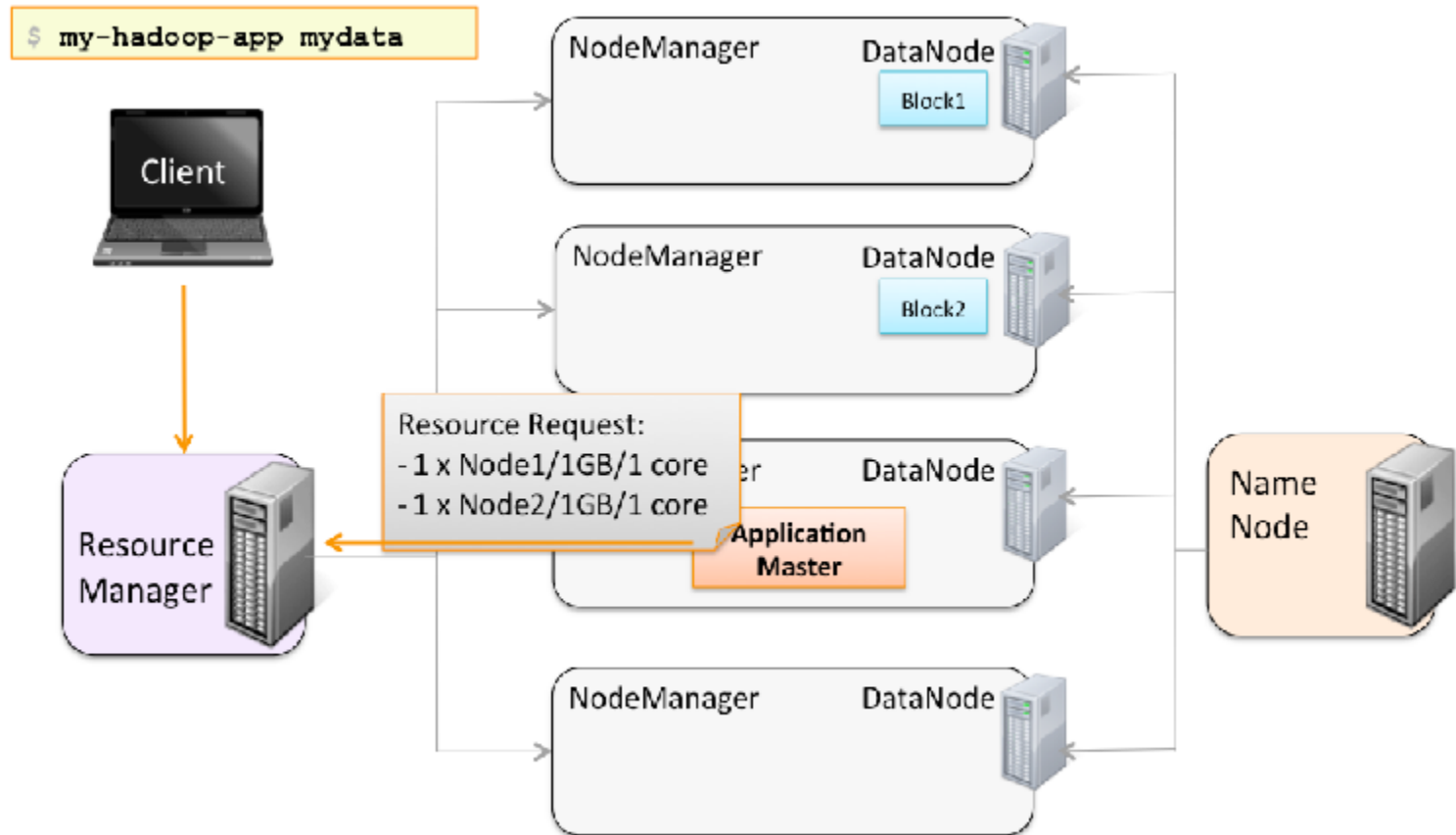
Run an application on YARN



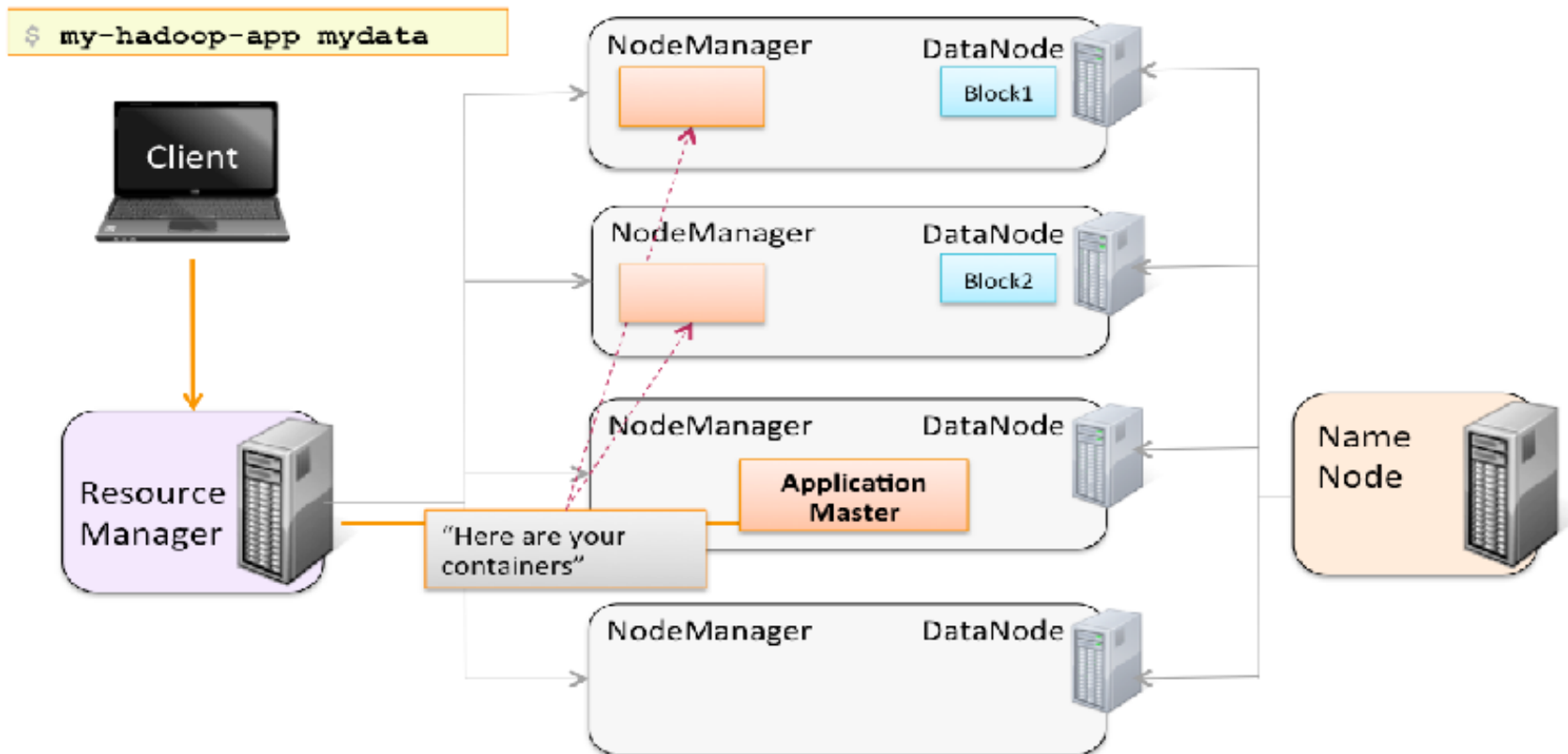
Run an application on YARN



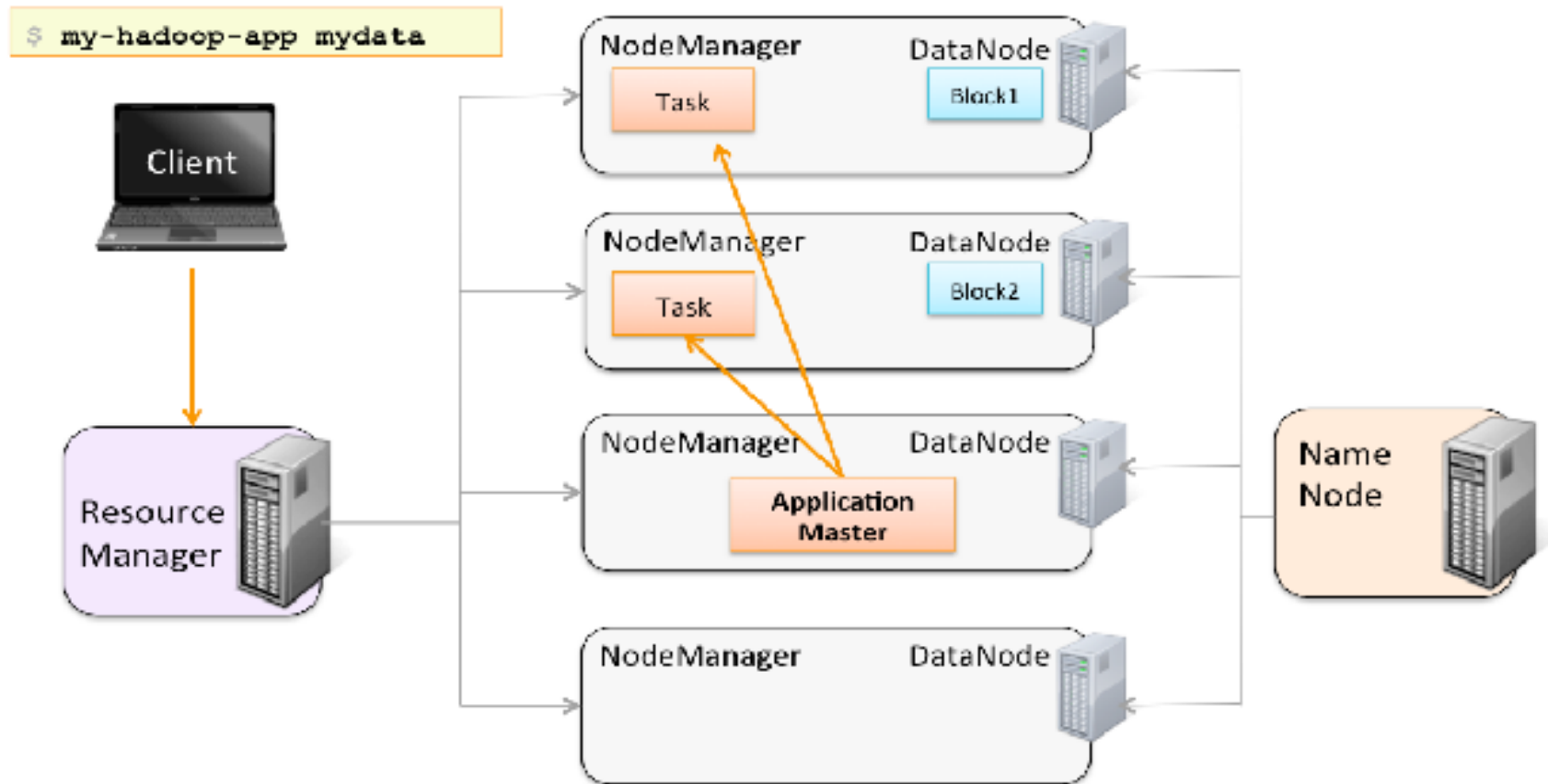
Run an application on YARN



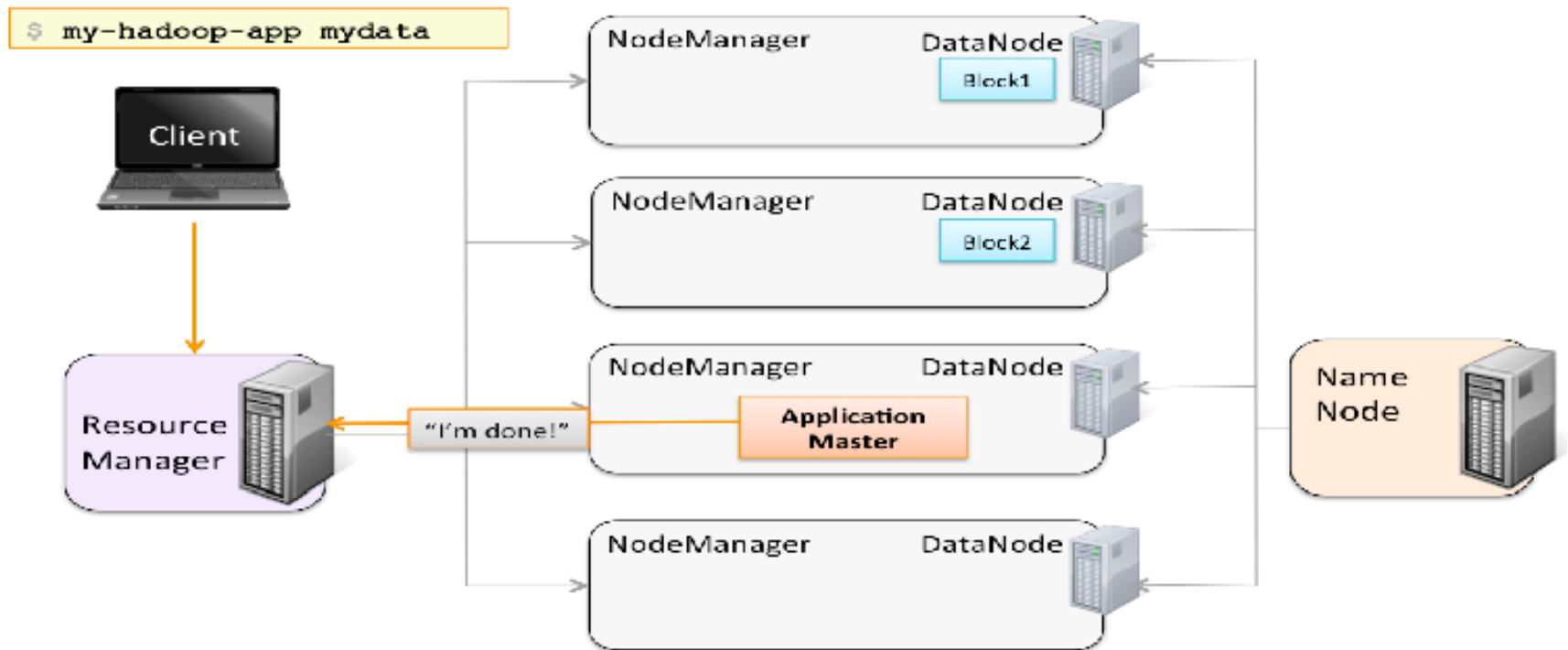
Run an application on YARN



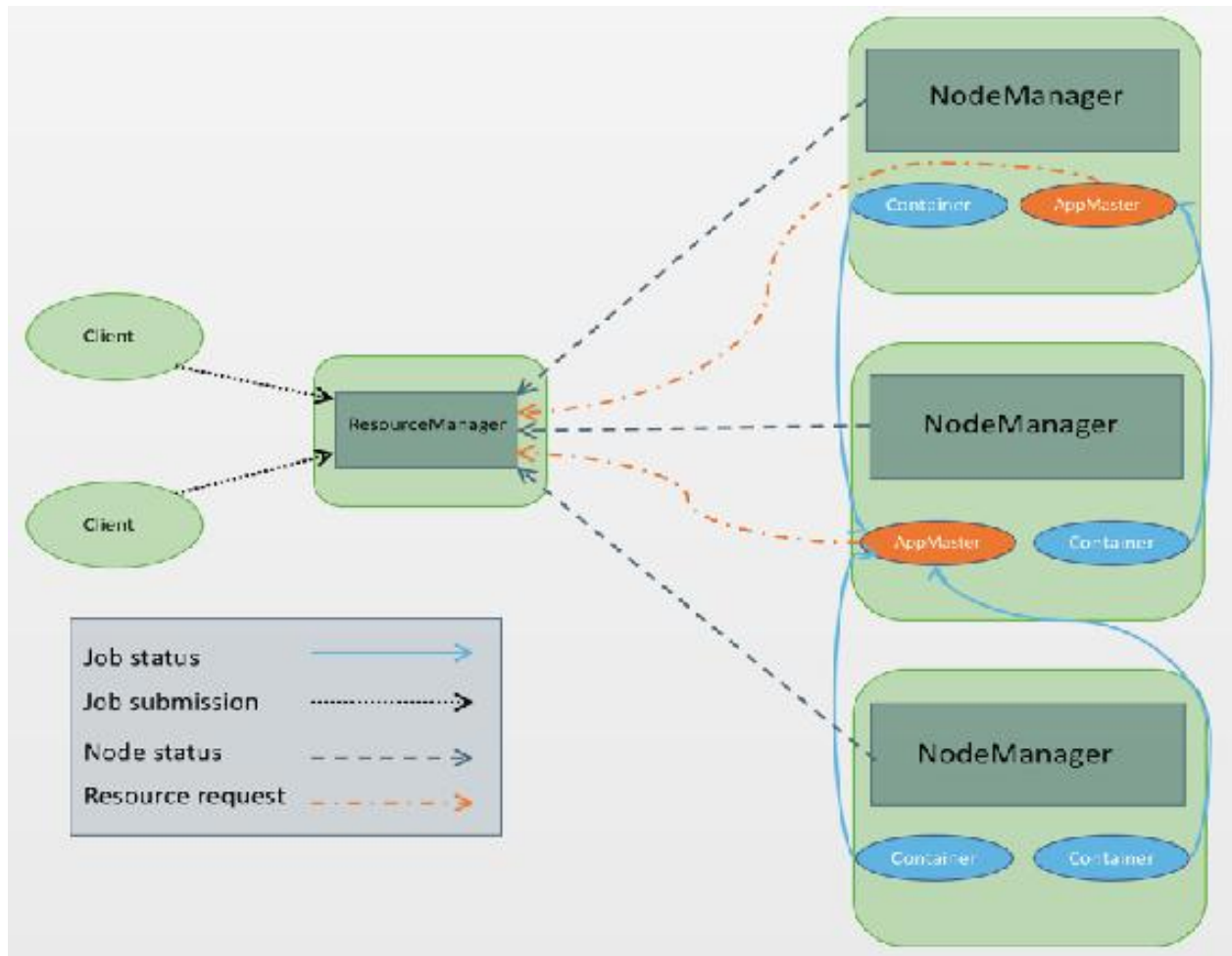
Run an application on YARN

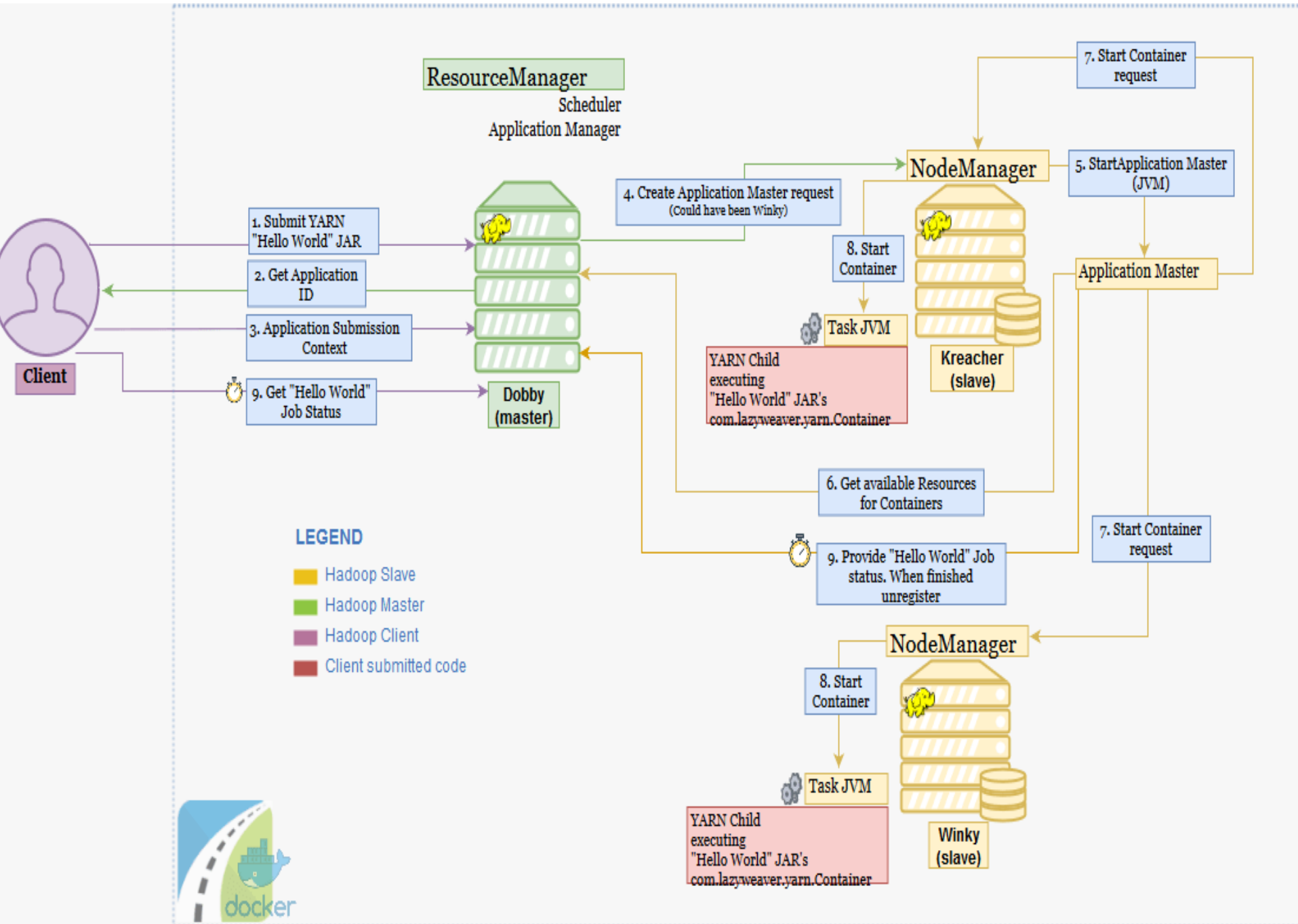


Run an application on YARN



Cluster viewYARN





HUE Browser

Username: training Text: Search for text

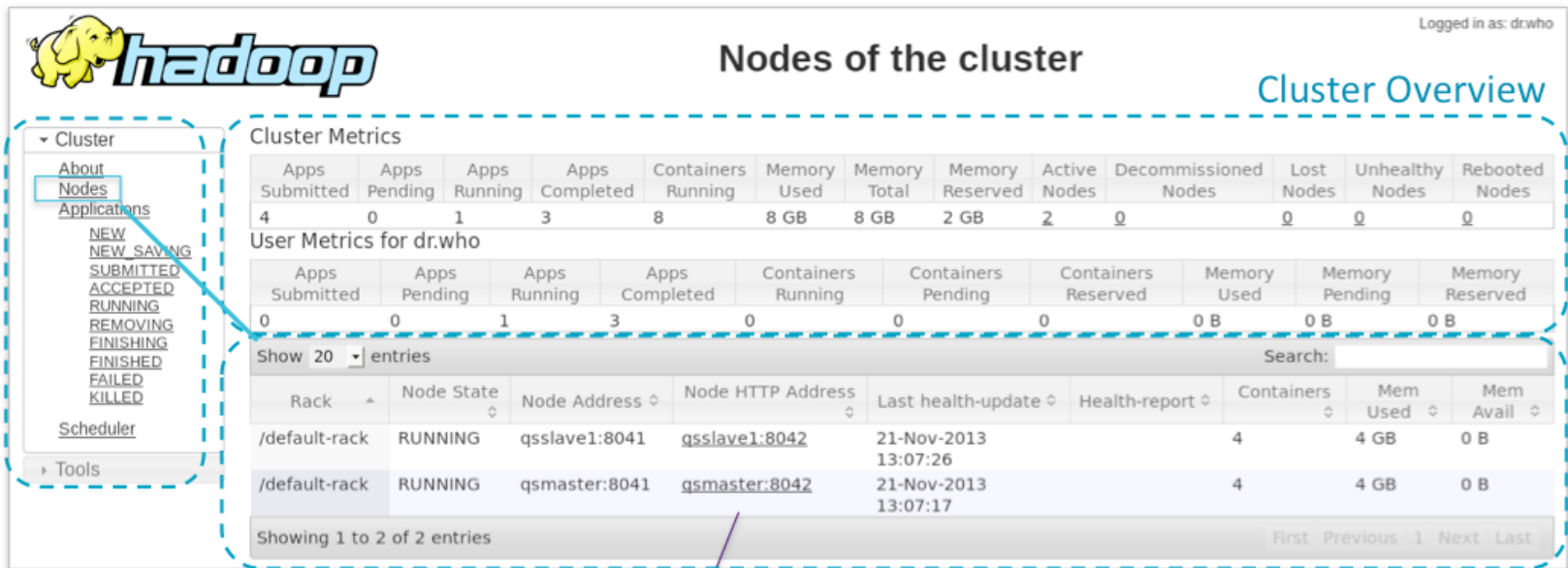
Succeeded Running Failed Killed

Logs	ID	Name	Status	User	Maps	Reduces	Queue	Priority	Duration	Submitted	
	1424901249645_0002	webpage.jar	RUNNING	training	5%	5%	root.training	N/A	18s	03/06/15 11:00:17	Kill
	1424901249645_0001	accounts.jar	SUCCEEDED	training	100%	100%	root.training	N/A	1m:21s	03/06/15 10:57:48	

Showing 1 to 2 of 2 entries

← Previous 1 Next →

YARN Manger Web UI



The screenshot displays the Hadoop YARN Manager Web UI. The top left features the Hadoop logo and a navigation menu with options like Cluster, About, Nodes, Applications, Scheduler, and Tools. The main header area includes the text "Nodes of the cluster" and "Cluster Overview". The "Nodes" menu item is highlighted with a blue box and a blue arrow pointing to the "Nodes" link in the navigation menu. Below the navigation menu, there are two tables: "Cluster Metrics" and "User Metrics for dr.who". The "Cluster Metrics" table shows various metrics such as Apps Submitted, Apps Pending, Apps Running, Apps Completed, Containers Running, Memory Used, Memory Total, Memory Reserved, Active Nodes, Decommissioned Nodes, Lost Nodes, Unhealthy Nodes, and Rebooted Nodes. The "User Metrics for dr.who" table shows similar metrics for the user dr.who. Below these tables is a list of nodes, with columns for Rack, Node State, Node Address, Node HTTP Address, Last health-update, Health-report, Containers, Mem Used, and Mem Avail. The list shows two nodes: /default-rack RUNNING qsslave1:8041 and /default-rack RUNNING qsmaster:8041. A purple box with the text "link to Node Manager UI" is positioned below the list, with a purple arrow pointing to the "qsmaster:8041" link in the "Node HTTP Address" column. The bottom right of the list shows pagination controls: "Showing 1 to 2 of 2 entries" and "First Previous 1 Next Last".

hadoop

Nodes of the cluster

Cluster Overview

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
4	0	1	3	8	8 GB	8 GB	2 GB	2	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	3	0	0	0	0 B	0 B	0 B

Show 20 entries

Search:

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

Showing 1 to 2 of 2 entries


First Previous 1 Next Last

link to Node
Manager UI

List of each node
in cluster

Job History Server

YARN does not keep track of job history



The screenshot shows the Hadoop JobHistory web interface. At the top left is the Hadoop logo. The title 'JobHistory' is centered at the top. On the right, it says 'Logged in as: drawho'. On the left side, there is a sidebar with 'Application' (containing 'About' and 'Jobs' links) and 'Tools'. The main area is titled 'Retired Jobs'. Below this, there is a 'Show 20 entries' dropdown and a 'Search:' input field. A table lists the retired jobs. The table has columns: Start Time, Finish Time, Job ID, Name, User, Queue, State, Maps Total, Maps Completed, Reduces Total, and Reduces Completed. A dashed blue box highlights the first six rows of the table.

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	job_1385066116114_0004	Process Logs	cloudera	default	SUCCEEDED	4	4	12	12
2013.11.21 13:03:53 PST	2013.11.21 13:04:42 PST	job_1385066116114_0003	Process Logs	cloudera	default	SUCCEEDED	4	4	12	12
2013.11.21 13:01:35 PST	2013.11.21 13:02:28 PST	job_1385066116114_0002	Process Logs	cloudera	default	SUCCEEDED	4	4	12	12
2013.11.21 12:48:00 PST	2013.11.21 12:50:43 PST	job_1385066116114_0001	Word Count	cloudera	default	SUCCEEDED	4	4	1	1
2013.11.21 09:24:45 PST	2013.11.21 09:28:19 PST	job_1385049040288_0003	Word Count	cloudera	default	SUCCEEDED	4	4	1	1