# Introduction to Big Data

# ECAP456

Dr. Rajni Bhalla

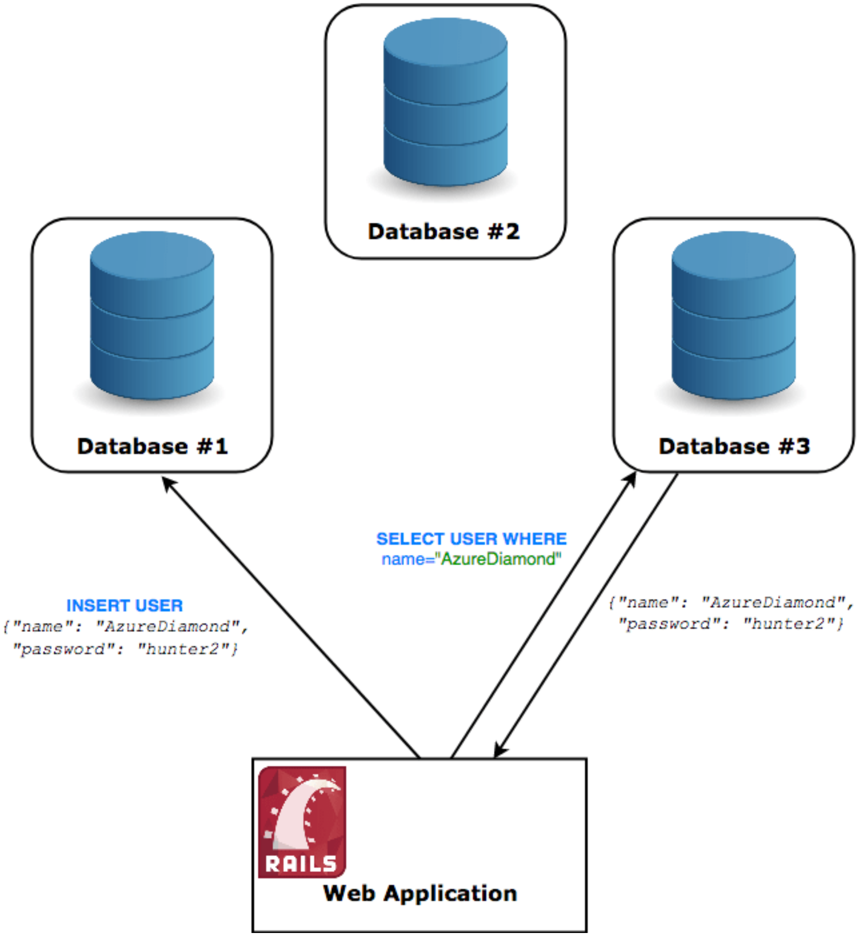
Associate Professor

# Learning Outcomes

After this lecture, you will be able to

- learn what is Hadoop,
- understand the Hadoop Core components,
- learn Hadoop Daemons,
- learn How Hdfs Works.

# Introduction



Database #2

Database #1

Database #3

SELECT USER WHERE
name="AzureDiamond"

INSERT USER
{"name": "AzureDiamond",
"password": "hunter2"}

{"name": "AzureDiamond",
"password": "hunter2"}

RAILS
Web Application

Distributed Manner

# Introduction

- Hadoop provides the world's most reliable storage layer

Storage Layer

# Introduction

- Hadoop provides the world's most reliable storage layer

Hadoop
MapReduce

Application Layer

# Introduction

- Hadoop provides the world's most reliable storage layer

Hadoop
YARN

Resource Management Layer

# What is Hadoop Distribute File System(HDFS)

Maintain huge volumes of data

Break down the data into smaller chunks

Distributed file systems.

Hadoop Distributed File System (HDFS) is the storage component

# What is Hadoop Distribute File System(HDFS)

Maintain huge volumes of data

Break down the data into smaller chunks

Distributed file systems.

Hadoop Distributed File System (HDFS) is the storage component

# What is Hadoop Distribute File System(HDFS)

Maintain huge volumes of data

Break down the data into smaller chunks

Distributed file systems.

Hadoop Distributed File System (HDFS) is the storage component

# What is Hadoop Distribute File System(HDFS)

Maintain huge volumes of data

Break down the data into smaller chunks

Distributed file systems.

Hadoop Distributed File System (HDFS) is the storage component

# What is Hadoop Distribute File System(HDFS)

It has a few properties that define its existence:-

- Huge volumes
- Data access
- Cost-effective

# What is Hadoop Distribute File System(HDFS)

It has a few properties that define its existence:-

- Huge volumes
- Data access
- Cost-effective

# What is Hadoop Distribute File System(HDFS)

It has a few properties that define its existence:-

Huge volumes

Data access

Cost-effective

# Hadoop Components and Domains

Hadoop consists of three layers (core components) and they are:-

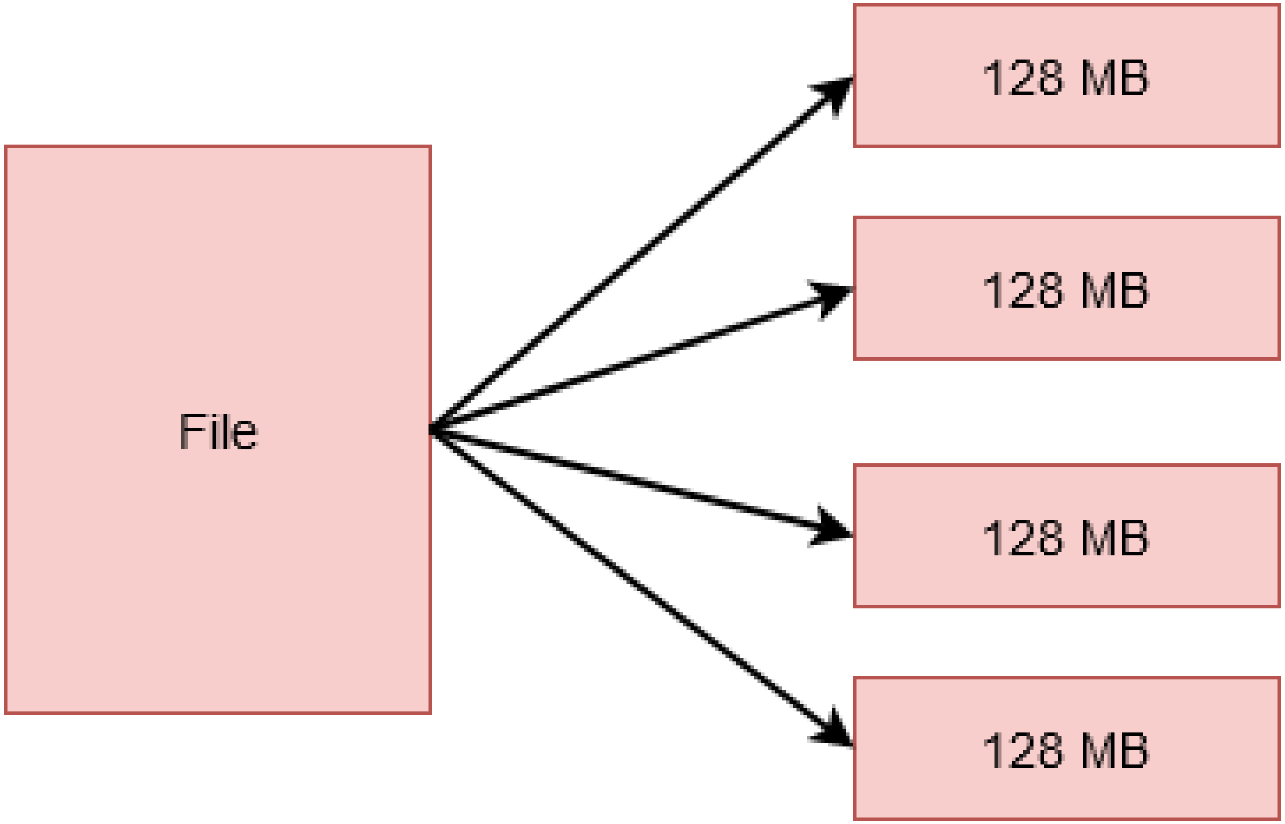
- HDFS – Hadoop Distributed File System
- MapReduce
- Yarn – Yet Another Resource Manager

# Hadoop Components

- The storage of Hadoop.
- Stores the data in a distributed manner.
- The file gets divided into a number of blocks.

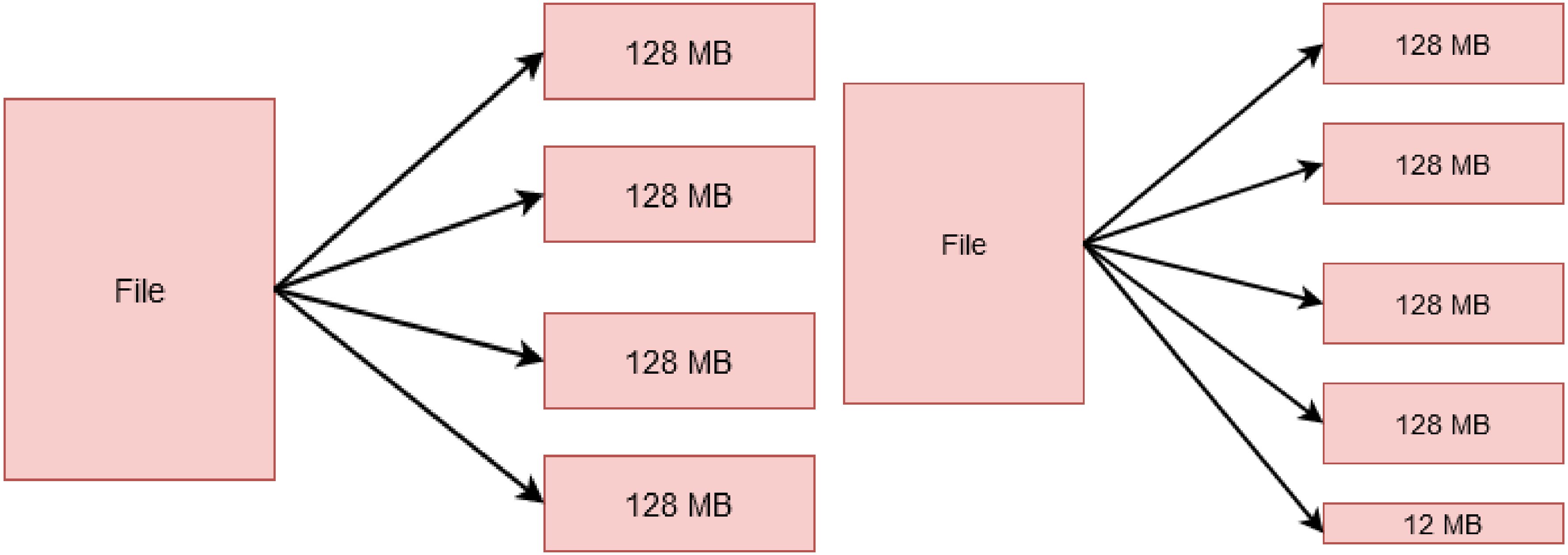
# Hadoop Components

- Example



# Hadoop Components

- Example



# Hadoop Components

- Why such a huge amount in a single block?
- Why not multiple blocks of 10KB each?

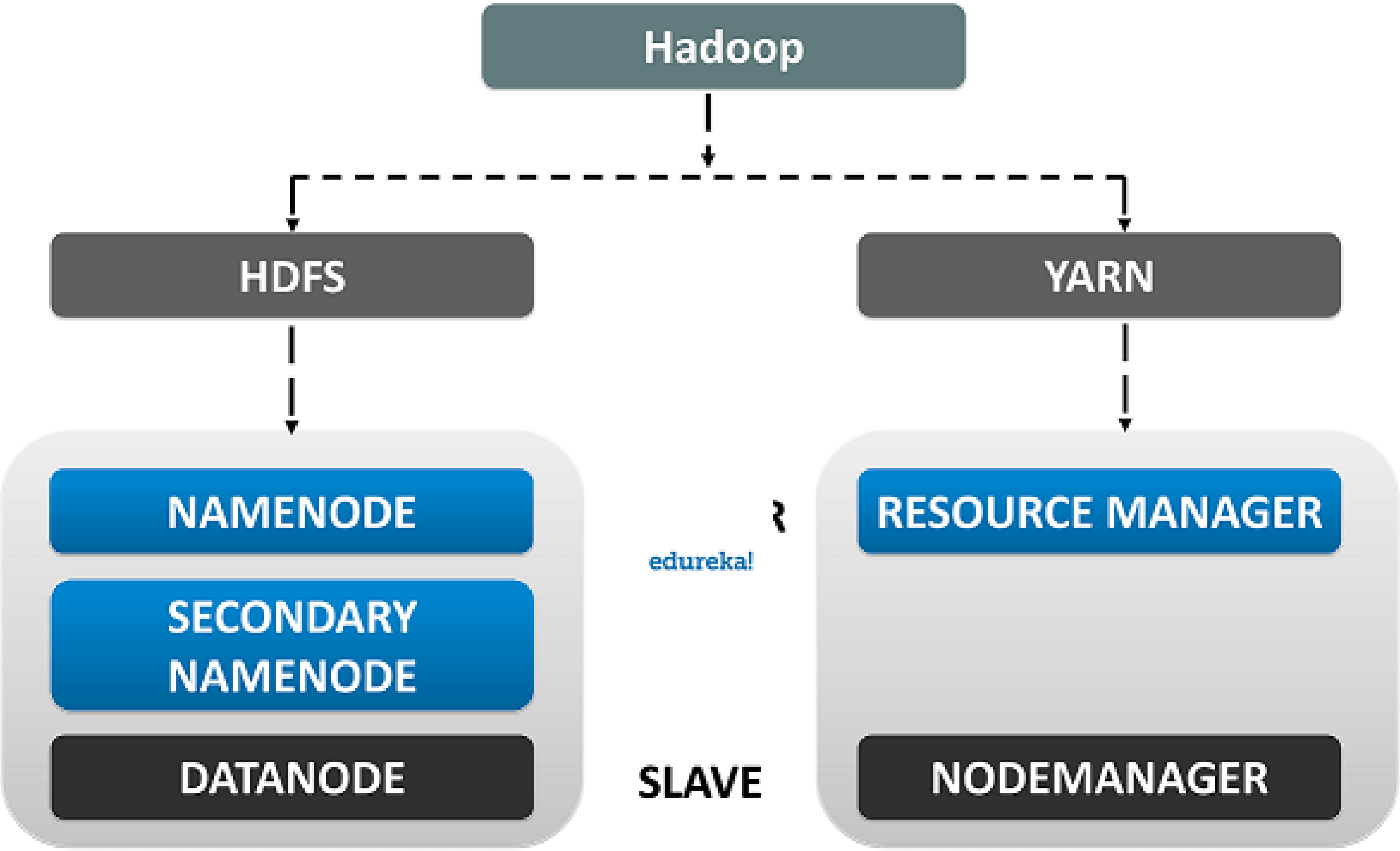
# Hadoop Components

There are several perks to storing data in blocks rather than saving the complete file.

- The file itself would be too large to store on any single disk alone.
- Proper spread of the workload
- Prevent the choke of a single machine

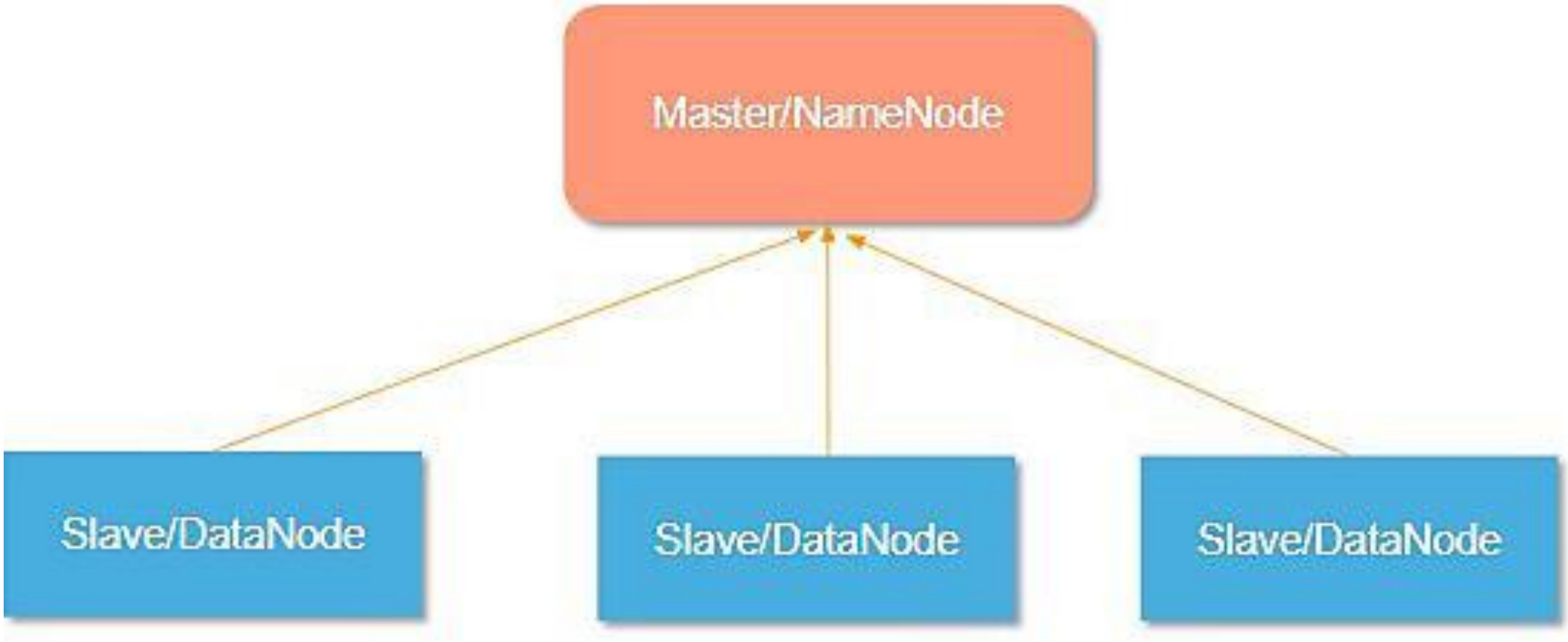
# Hadoop Components

The HDFS comprises the following components



# Hadoop Components

## Namenode in HDFS



# Hadoop Components

Namenode is the master node that runs on a separate node in the cluster.

Manages the filesystem namespace

Stores information

aware of the locations of all the blocks

# Hadoop Components

Namenode is the master node that runs on a separate node in the cluster.

- Manages the filesystem namespace
- Stores information
- aware of the locations of all the blocks

# Hadoop Components

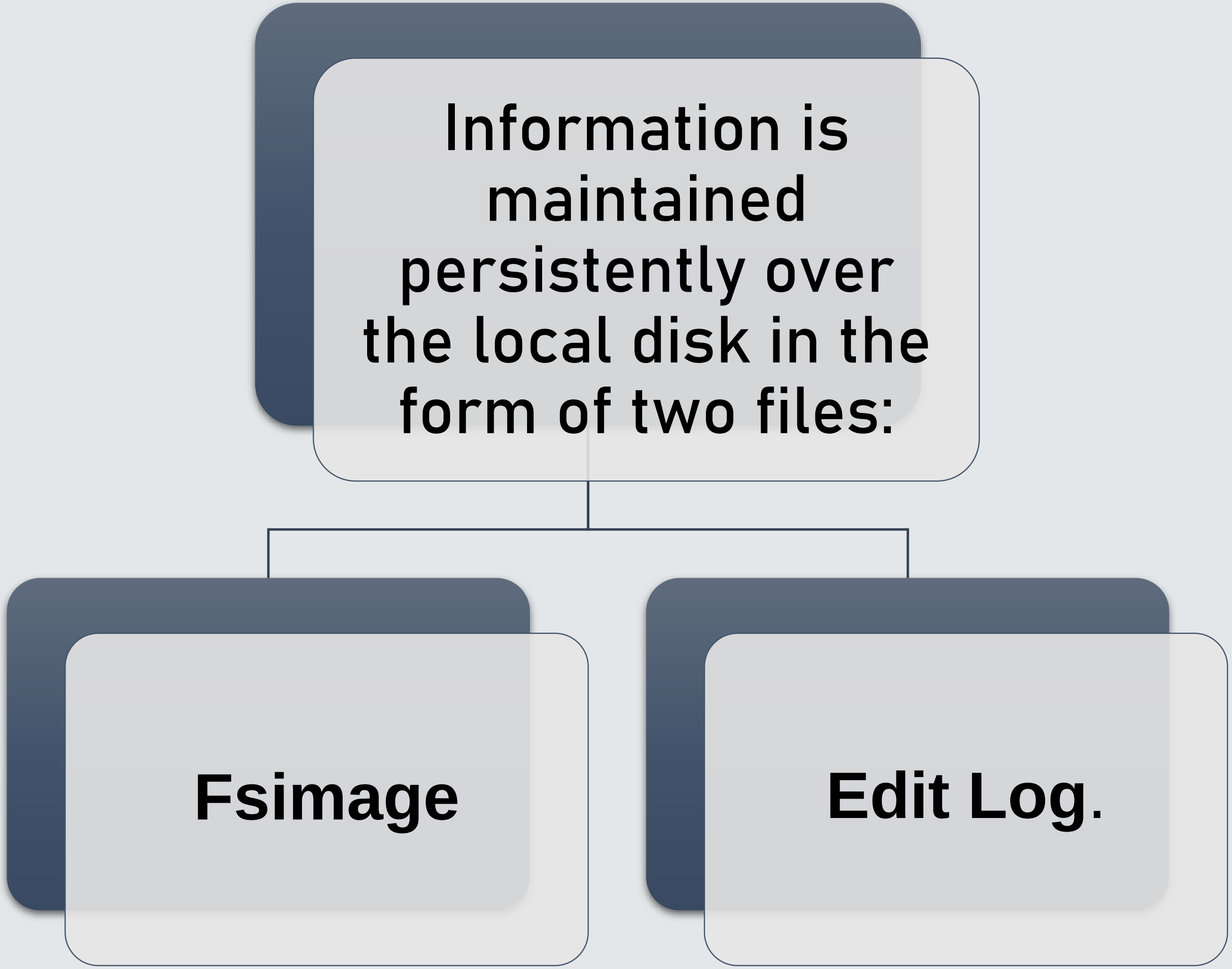
Namenode is the master node that runs on a separate node in the cluster.

Manages the filesystem namespace

Stores information

aware of the locations of all the blocks

# Hadoop Components



Information is maintained persistently over the local disk in the form of two files:

Fsimage

Edit Log.

# Hadoop Components

## Data node in HDFS

- Worker nodes
- Inexpensive commodity hardware
- Responsible for storing, retrieving, replicating, deletion, etc.
- Send heartbeats to the Namenode
- Sends a list of blocks

# Hadoop Components

## Data node in HDFS

- Worker nodes
- Inexpensive commodity hardware
- Responsible for storing, retrieving, replicating, deletion, etc.
- Send heartbeats to the Namenode
- Sends a list of blocks

# Hadoop Components

## Data node in HDFS

- Worker nodes
- Inexpensive commodity hardware
- Responsible for storing, retrieving, replicating, deletion, etc.
- Send heartbeats to the Namenode
- Sends a list of blocks

# Hadoop Components

## Data node in HDFS

- Worker nodes
- Inexpensive commodity hardware
- Responsible for storing, retrieving, replicating, deletion, etc.
- Send heartbeats to the Namenode
- Sends a list of blocks

# Hadoop Components

## Data node in HDFS

- Worker nodes
- Inexpensive commodity hardware
- Responsible for storing, retrieving, replicating, deletion, etc.
- Send heartbeats to the Namenode
- Sends a list of blocks

# Hadoop Components

## Secondary Namenode in HDFS

- Copy the Fsimage from disk to memory
- Copy the latest copy of Edit Log to Fsimage
- If we restart the node after a long time, then the Edit log could have grown in size.
- Lot of time to apply the transactions from the Edit log.

# Hadoop Components

## Secondary Namenode in HDFS

- Copy the Fsimage from disk to memory
- Copy the latest copy of Edit Log to Fsimage
- If we restart the node after a long time, then the Edit log could have grown in size.
- Lot of time to apply the transactions from the Edit log.

# Hadoop Components

## Secondary Namenode in HDFS

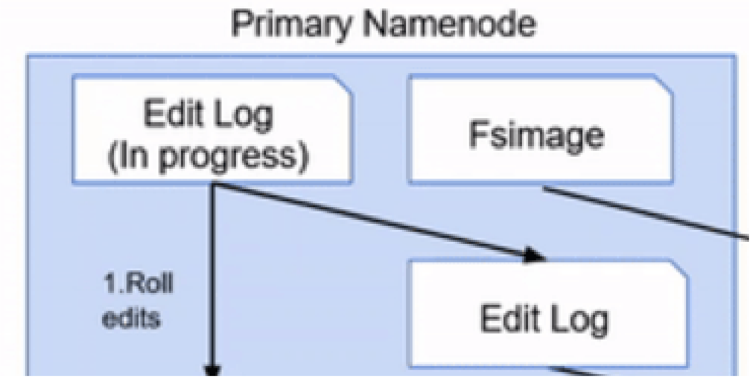
- Copy the Fsimage from disk to memory
- Copy the latest copy of Edit Log to Fsimage
- If we restart the node after a long time, then the Edit log could have grown in size.
- Lot of time to apply the transactions from the Edit log.

# Hadoop Components

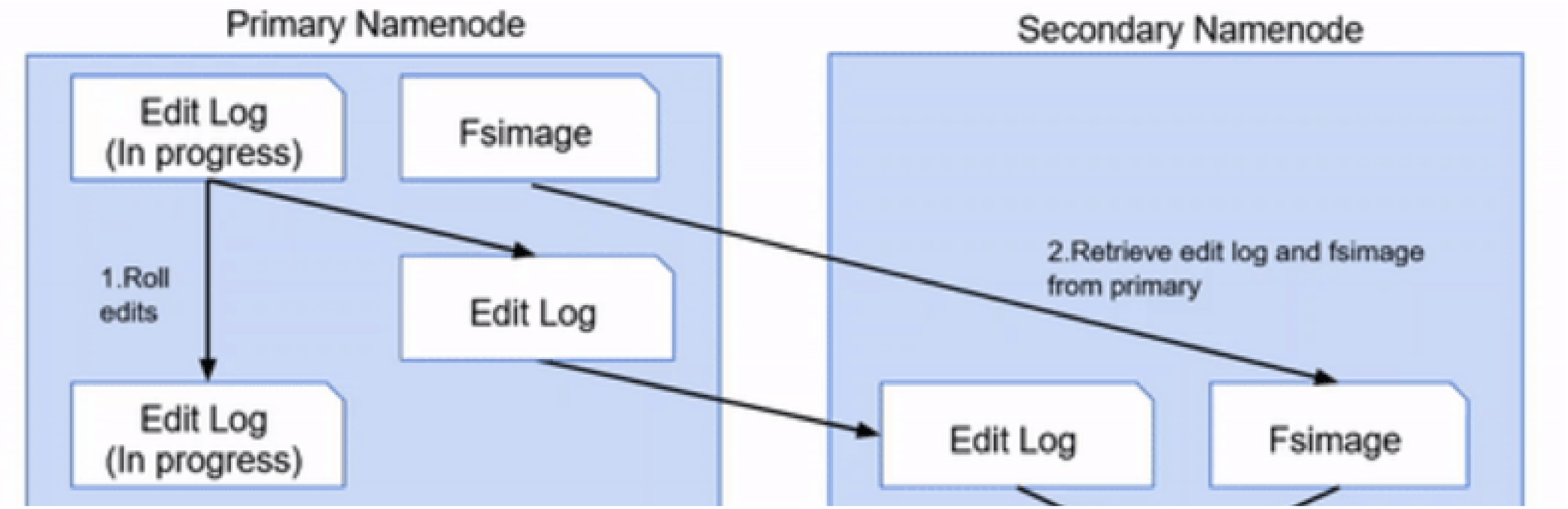
## Secondary Namenode in HDFS

- Copy the Fsimage from disk to memory
- Copy the latest copy of Edit Log to Fsimage
- If we restart the node after a long time, then the Edit log could have grown in size.
- Lot of time to apply the transactions from the Edit log.

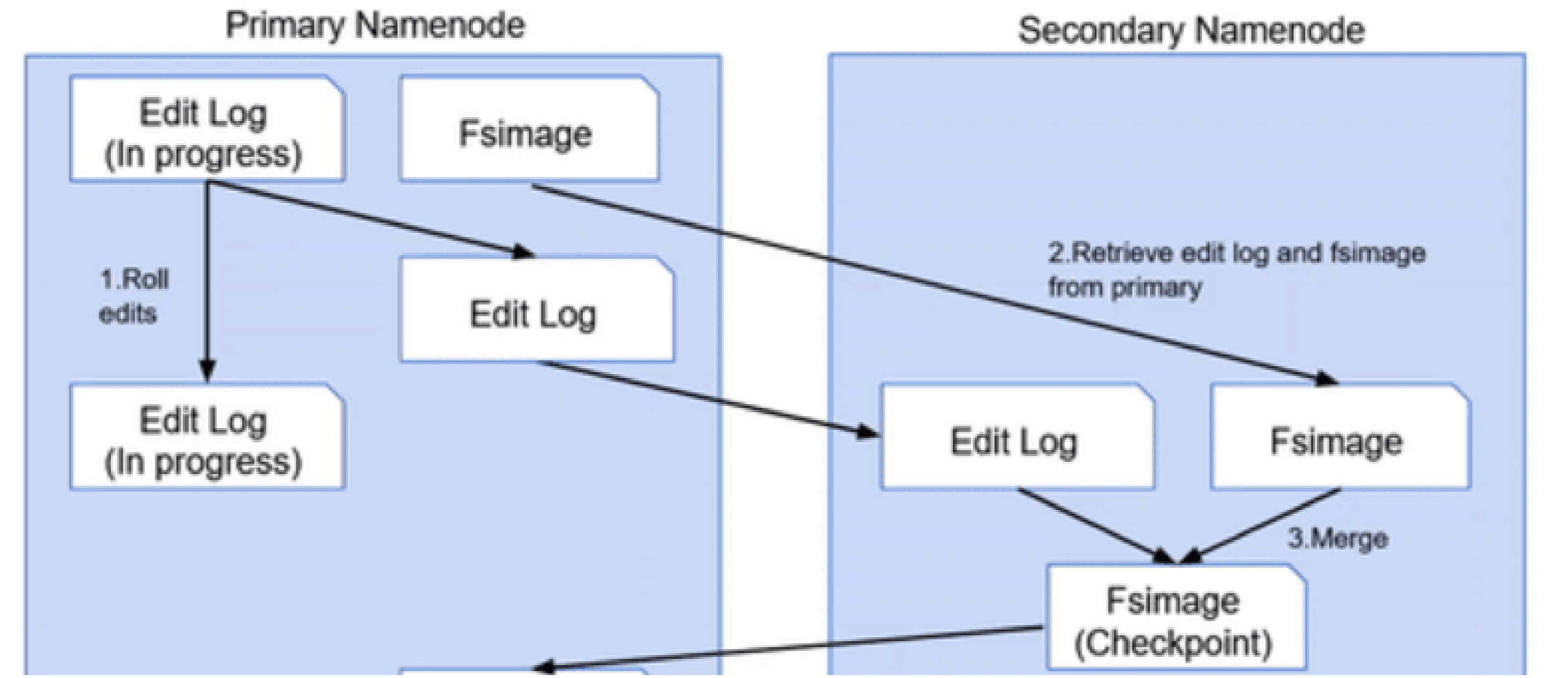
# Hadoop Components



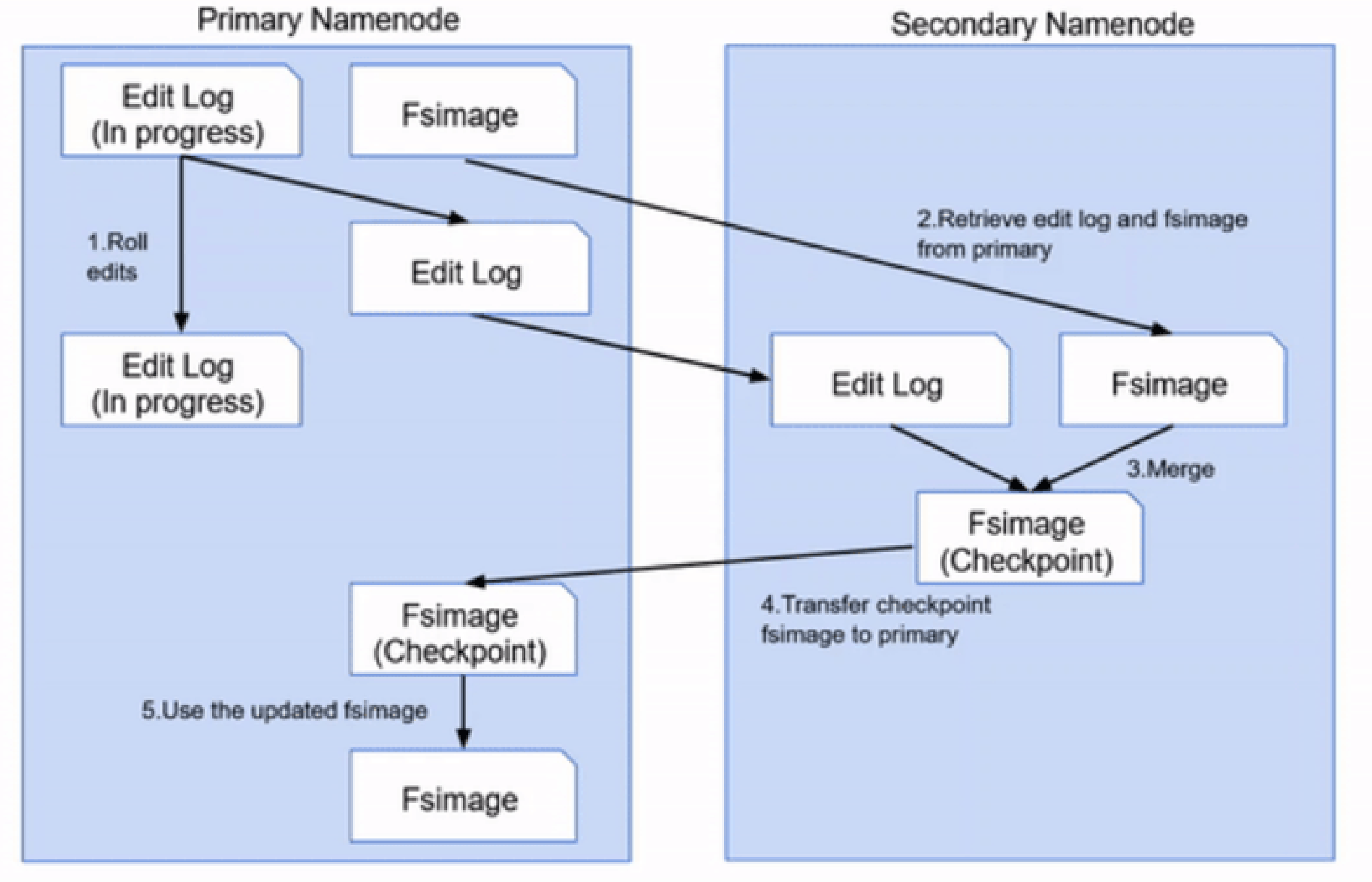
# Hadoop Components



# Hadoop Components



# Hadoop Components



# Hadoop Components

## Secondary Namenode in HDFS

- checkpointing procedure is computationally very expensive
- Secondary namenode runs on a separate node on the cluster.
- Secondary Namenode does not act as a Namenode.
- Keeping a copy of the latest Fsimage.

# Hadoop Components

## Secondary Namenode in HDFS

- checkpointing procedure is computationally very expensive
- Secondary namenode runs on a separate node on the cluster.
- Secondary Namenode does not act as a Namenode.
- Keeping a copy of the latest Fsimage.

# Hadoop Components

## Secondary Namenode in HDFS

- checkpointing procedure is computationally very expensive
- Secondary namenode runs on a separate node on the cluster.
- Secondary Namenode does not act as a Namenode.
- Keeping a copy of the latest Fsimage.

# Hadoop Components

## Secondary Namenode in HDFS

- checkpointing procedure is computationally very expensive
- Secondary namenode runs on a separate node on the cluster.
- Secondary Namenode does not act as a Namenode.
- Keeping a copy of the latest Fsimage.

# Hadoop Components

## Mapreduce

- This is the processing engine of Hadoop.
- MapReduce works on the principle of distributed processing.
- It divides the task submitted by the user into a number of independent subtasks.

# Hadoop Components

## Mapreduce

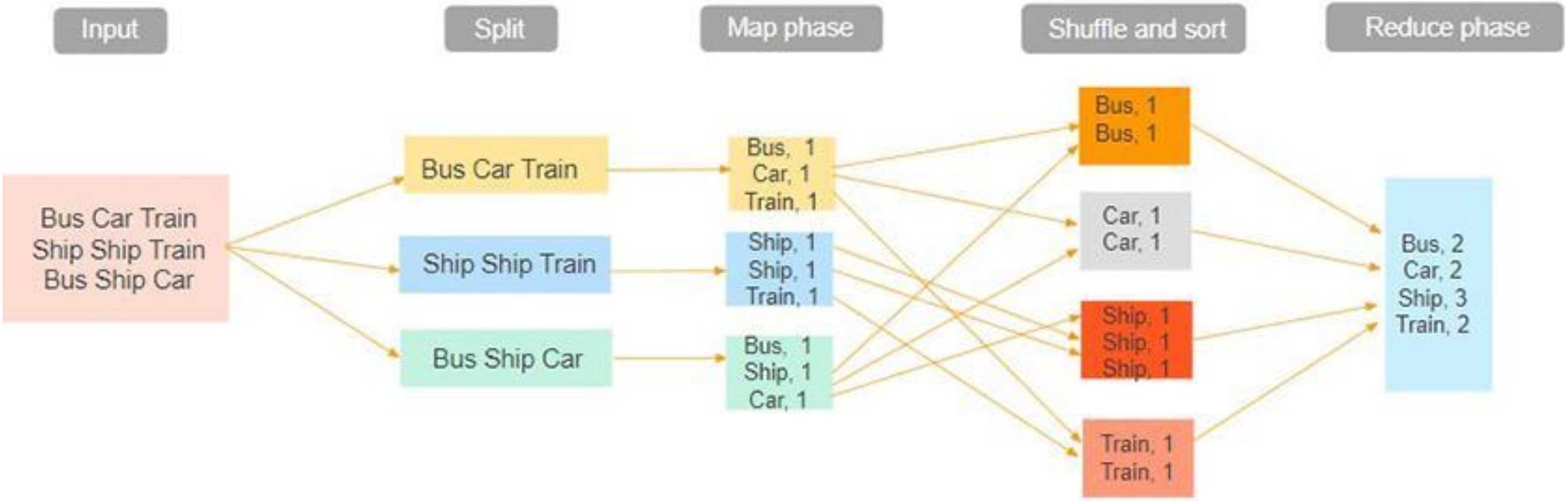
- This is the processing engine of Hadoop.
- MapReduce works on the principle of distributed processing.
- It divides the task submitted by the user into a number of independent subtasks.

# Hadoop Components

## Mapreduce

- This is the processing engine of Hadoop.
- MapReduce works on the principle of distributed processing.
- It divides the task submitted by the user into a number of independent subtasks.

# Hadoop Components

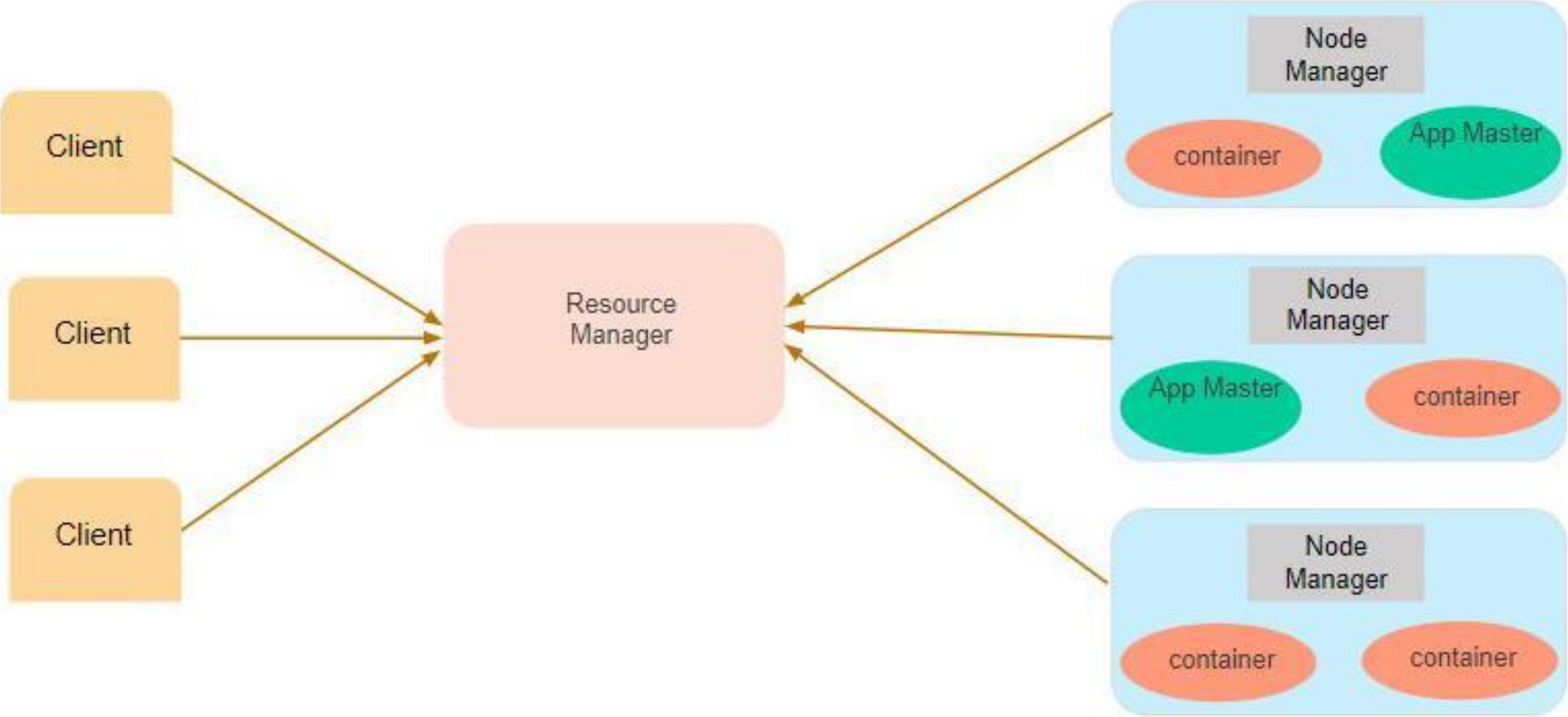


# Hadoop YARN

Hadoop YARN stands for Yet Another Resource Negotiator. It is the resource management unit of Hadoop and is available as a component of Hadoop version 2.

- Hadoop YARN acts like an OS to Hadoop. It is a file system that is built on top of HDFS.
- It is responsible for managing cluster resources to make sure you don't overload one machine.
- It performs job scheduling to make sure that the jobs are scheduled in the right place

# Hadoop YARN


Client
Client
Client
Resource Manager
Node Manager
container
App Master
Node Manager
App Master
container
Node Manager
container
container

# Hadoop Daemons

The Hadoop Daemons are:-

a) Namenode

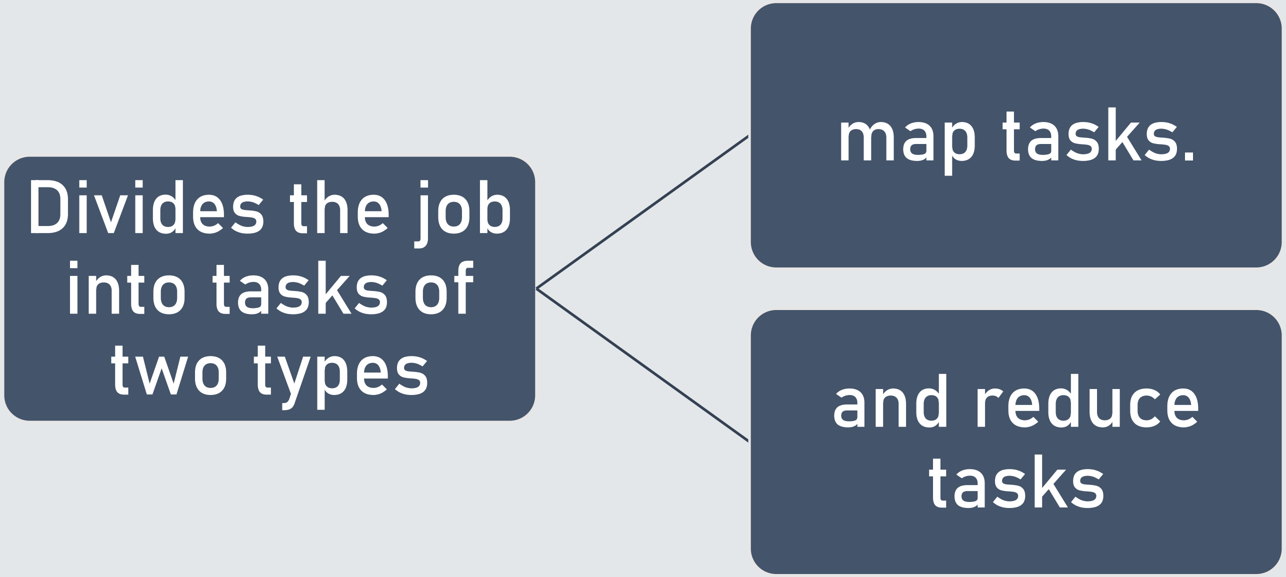
b) Datanode

c) Resource Manager

d) Node Manager

# How HDFS  works?

The Hadoop MapReduce works as follows:



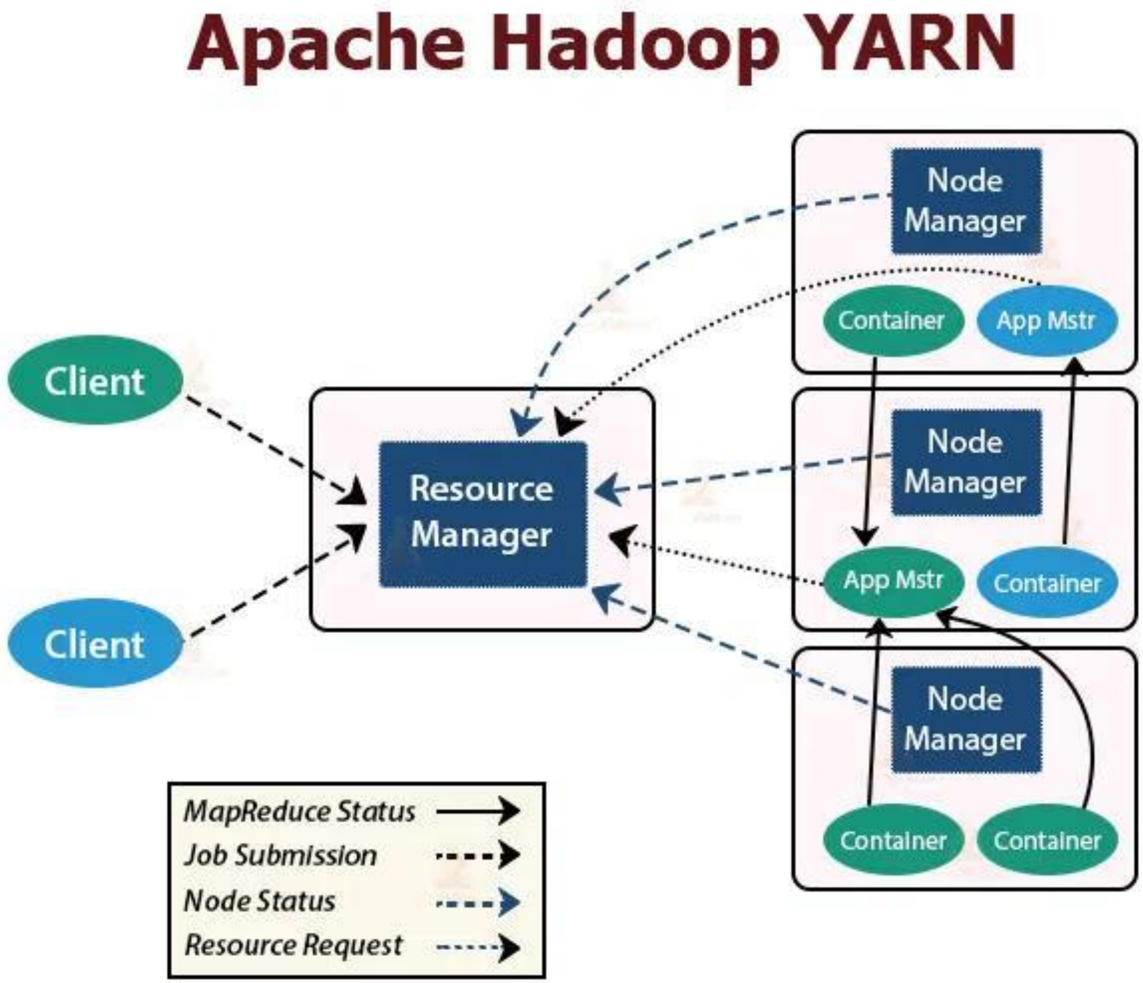
# How HDFS works?

- YARN scheduled these tasks
- MapReduce job is divided into fixed-size pieces
- map tasks run on the DataNodes where the input data resides.
- output of the map task is intermediate output
- intermediate outputs of the map tasks are shuffled

# How HDFS works?

- the sorted intermediate output of mapper is passed to the node where the reducer task is running.
- reduce function summarizes the output
- For multiple reduce functions, the user specifies the number of reducers

# How HDFS  works?



# How HDFS works?

- There are two YARN daemons running in the Hadoop cluster for serving YARN core services. They are:
- Resource Manager
- Node Manager
- Application Master

# Summarize how Hadoop works internally

That’s all for now…