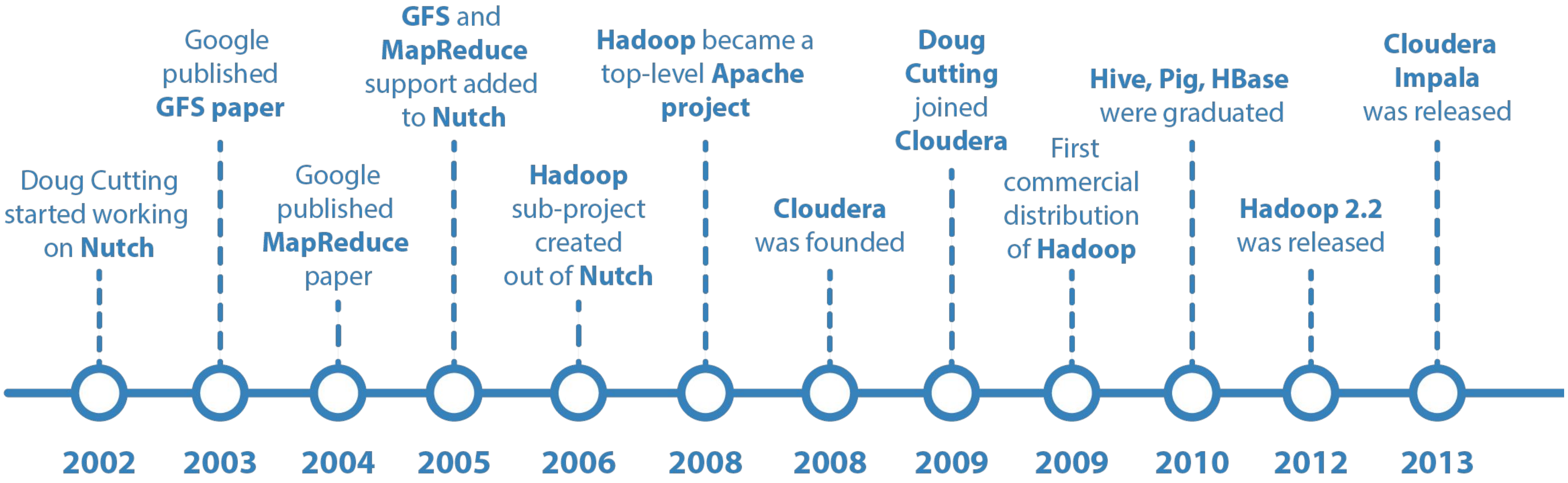
Apache Hadoop - HDFS

Apache Hadoop Framework

The Apache Hadoop software library is a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models

Apache Hadoop: Brief history



Doug Cutting started working on **Nutch**

2002

Google published **GFS paper**

2003

Google published **MapReduce** paper

2004

GFS and **MapReduce** support added to **Nutch**

2005

Hadoop sub-project created out of **Nutch**

2006

Hadoop became a top-level **Apache project**

2008

Cloudera was founded

2008

Doug Cutting joined **Cloudera**

2009

First commercial distribution of **Hadoop**

2009

Hive, Pig, HBase were graduated

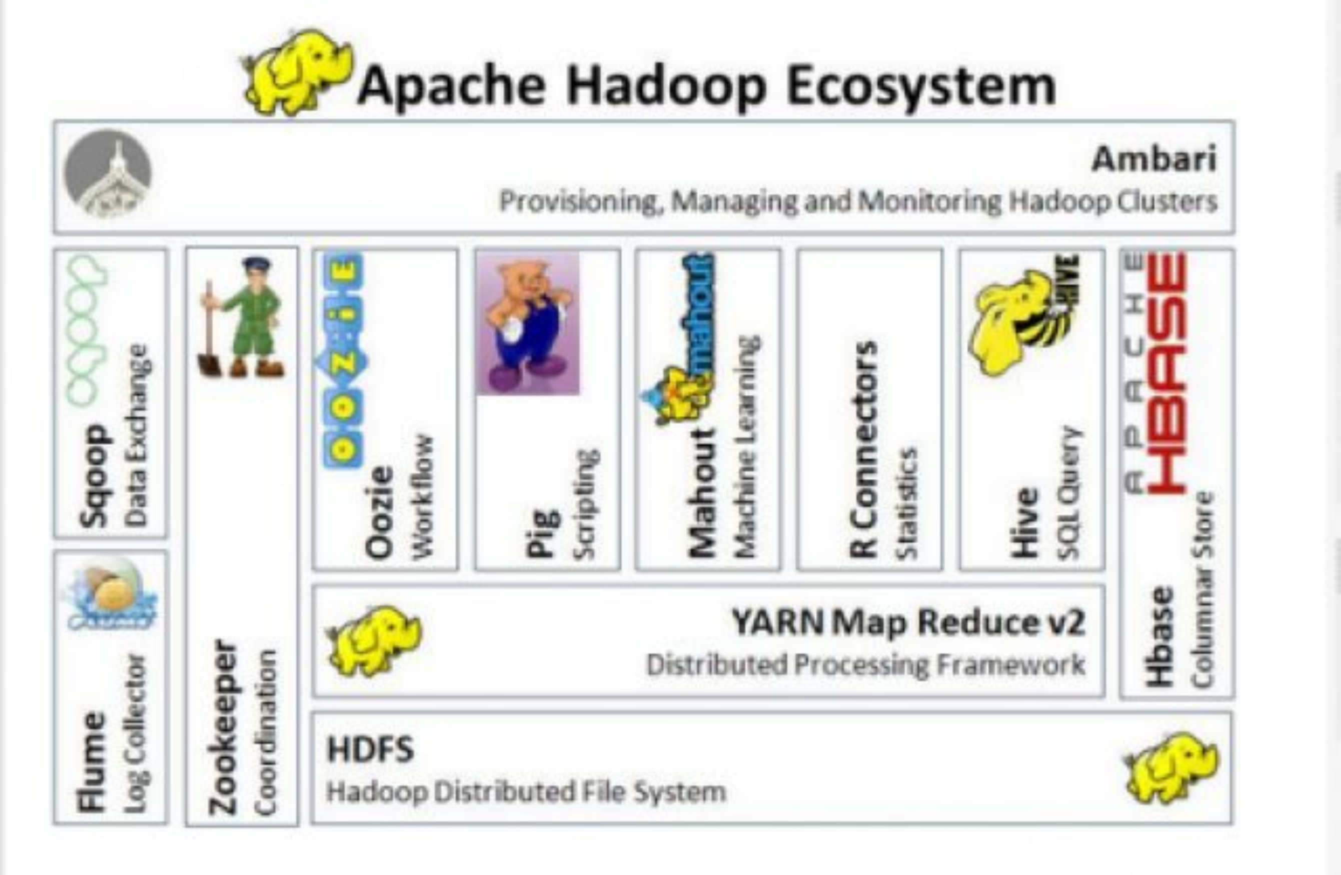
2010

Hadoop 2.2 was released

2012

Cloudera Impala was released

2013

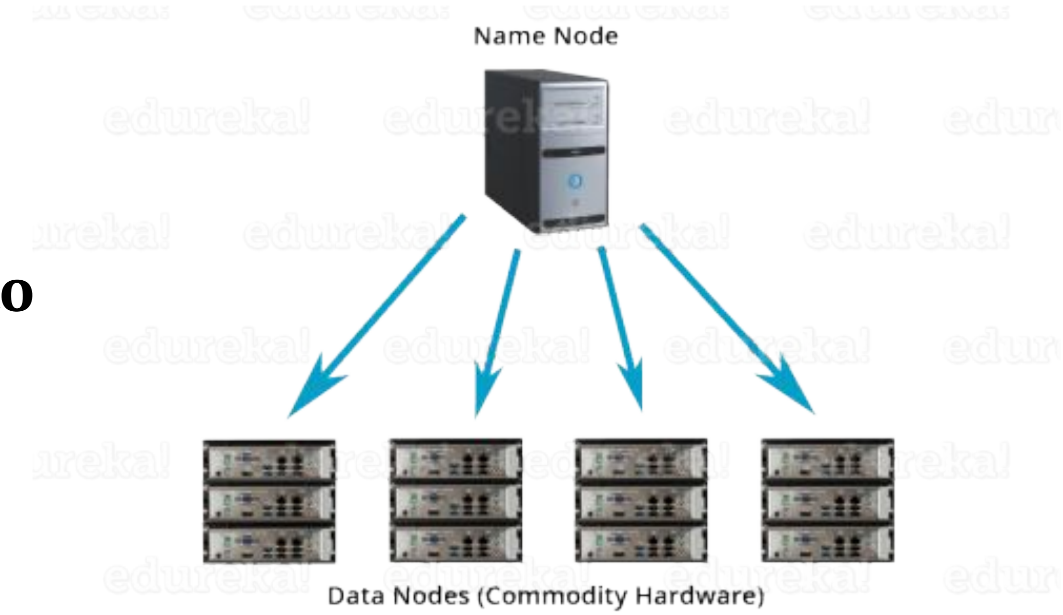

Apache Hadoop Ecosystem
Ambari
Provisioning, Managing and Monitoring Hadoop Clusters
Sqoop
Data Exchange
Flume
Log Collector
Zookeeper
Coordination
Oozie
Workflow
Pig
Scripting
Mahout
Machine Learning
R Connectors
Statistics
Hive
SQL Query
APACHE HBASE
Hbase
Columnar Store
YARN Map Reduce v2
Distributed Processing Framework
HDFS
Hadoop Distributed File System

HDFS (Hadoop Distributed File Systems)

- Hadoop Distributed File Systems
- HDFS **holds very large amount of data** and provides **easier access**. To store such huge data, **the files are stored across multiple machines**. These files are stored in **redundant fashion** to rescue the system from possible data losses in case of failure. HDFS also makes applications available to parallel processing.

HDFS (Hadoop Distributed File Systems)

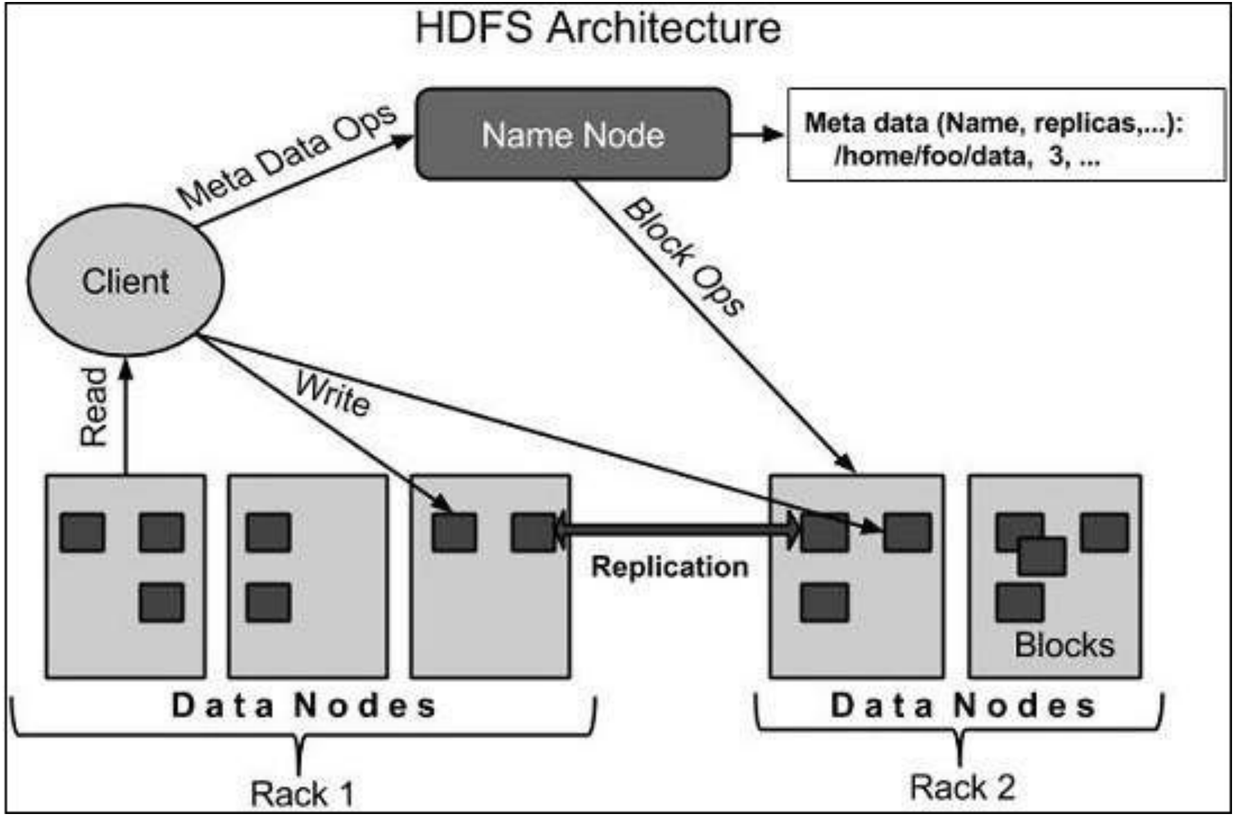
- Master/Slave (NameNode & DataNodes).
- Each file is divided into blocks of a pre-determined size.



Features of HDFS

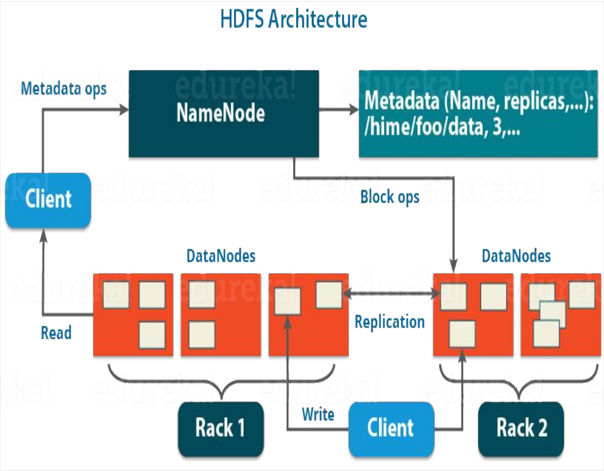
- It is suitable for the **distributed storage and processing**.
- Hadoop provides **a command interface** to interact with HDFS.
- The built-in servers of namenode and datanode help users to **easily check the status of cluster**.
- **Streaming access** to file system data.
- HDFS provides file **permissions** and **authentication**.

HDFS Architecture



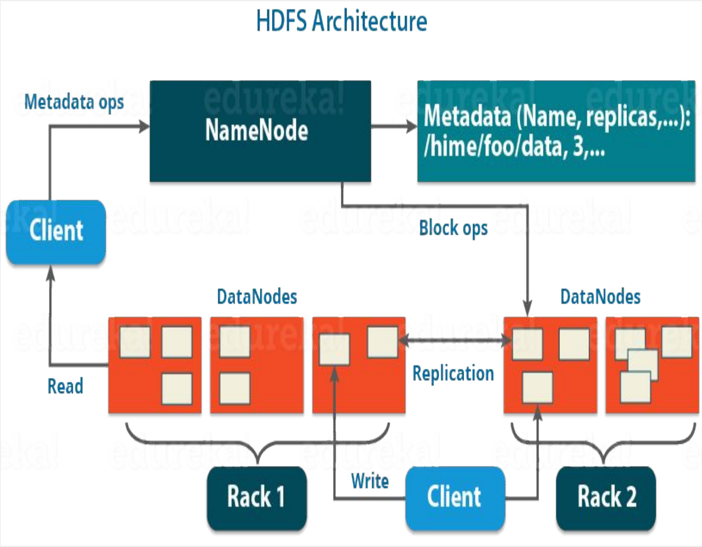
HDFS Architecture - Namenode

- **Maintains and manages the blocks** present on the DataNodes (slave nodes). It r**ecords the metadata of all the files stored in the cluster, e.g. The location of blocks stored, the size of the files, permissions, hierarchy, etc**. There are two files associated with the metadata: **FsImage**, **EditLogs**.
- The NameNode is also responsible to take care of the **replication factor** of all the blocks.
- It regularly **receives a Heartbeat and a block report from all the DataNodes** in the cluster to ensure that the DataNodes are live.



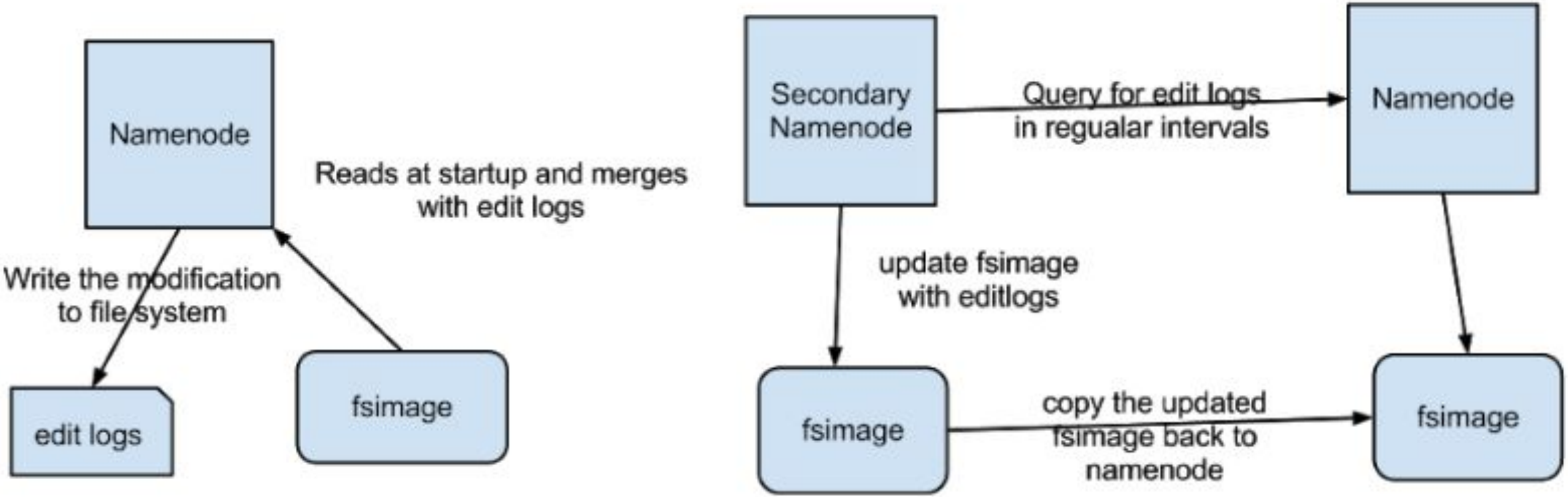
HDFS Architecture - Datanode

- DataNodes are the slave nodes in the HDFS Architecture that store the data in the local file.
- Function of **DataNodes**:
  - They **send heartbeats to the NameNode periodically to report the overall health of HDFS**.
  - The actual data is stored on them.
  - They perform the low-level read and write requests from the file system’s clients.



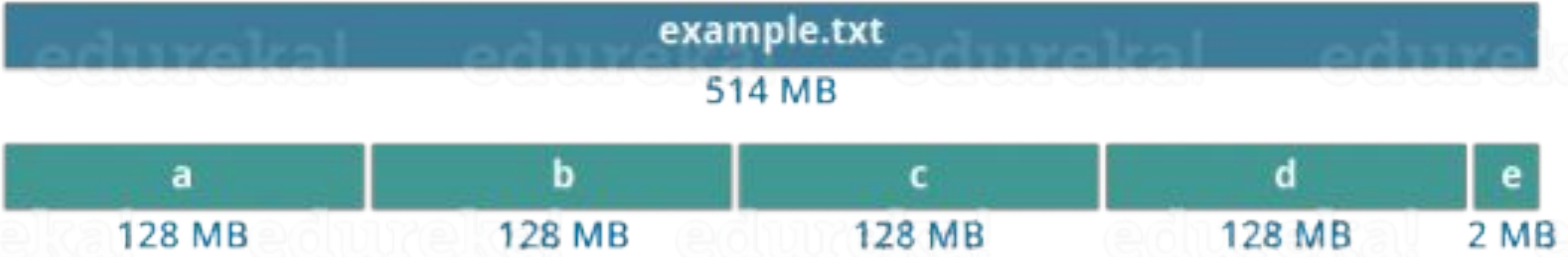
HDFS Architecture - Secondary Namenode

- Stores a copy of FsImage and EditLog files.


Namenode
Reads at startup and merges with edit logs
Write the modification to file system
edit logs
fsimage
Secondary Namenode
Query for edit logs in regualar intervals
Namenode
update fsimage with editlogs
fsimage
copy the updated fsimage back to namenode
fsimage

HDFS Architecture - Blocks

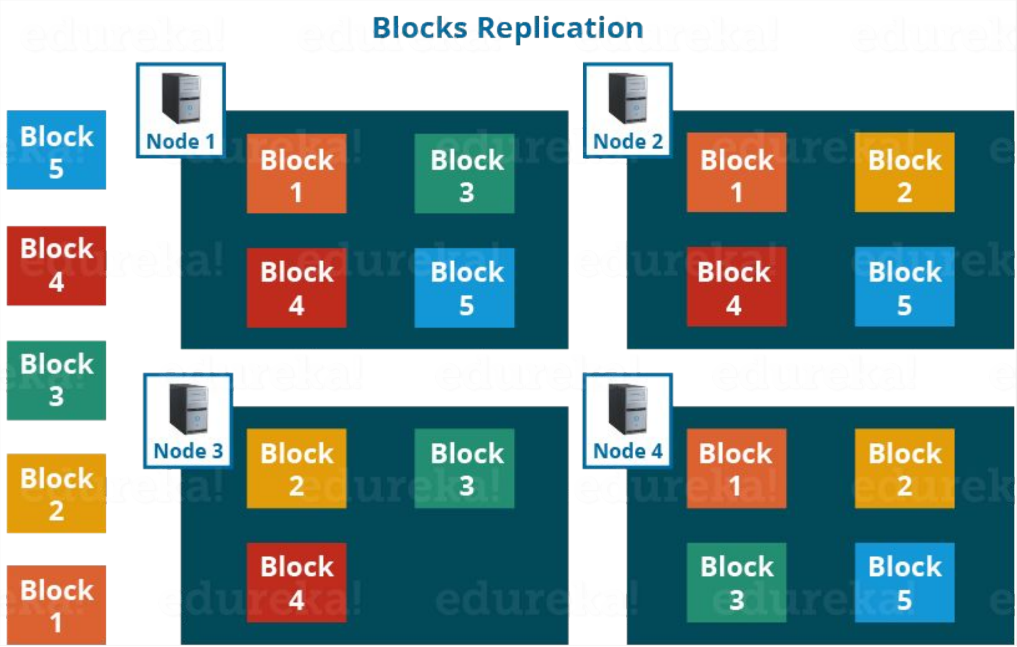
- The file in a file system will be divided into one or more segments and/or stored in individual data nodes. These file segments are called as blocks. The default size of each block is 128 MB in Apache Hadoop 2.x (64 MB in Apache Hadoop 1.x) which you can configure as per your requirement.



HDFS Architecture - Replication

- The blocks are also replicated to provide fault tolerance.
- The default replication factor is 3 which is again configurable.

→ In case of the DataNode failure, the NameNode chooses new DataNodes for new replicas, balance disk usage and manages the communication traffic to the DataNodes.



HDFS Commands

hdfs version

hdfs dfs -ls <path> : liệt kê file và folder tại path chỉ định

hdfs dfs -mkdir [-p] <path> : tạo folder. Nếu có -p, tạo luôn folder cha nếu folder cha chưa tồn tại

hdfs dfs -ls [-R] <path>: liệt kê file và folder tại path. Nếu có -R thì liệt kê cả thư mục con bên trong

HDFS Commands

hdfs dfs -put <localSrc> <dest> : put file or folder từ local lên hdfs

hdfs dfs -get <srcHdfs> <localDest> : lấy file or folder từ hdfs về local

hdsf dfs -mv <src> <dest>: di chuyển file hoặc folder trên hdfs

hdfs dfs -cp <src> <dest>: copy file or folder trên hdfs

HDFS Commands

- chown
- du
- df
- cat
- chmod

HDFS Commands (for admin)

- hdfs dfsadmin -report: nhận report về tình trạng của hdfs như: đã dùng hết bao nhiêu dung lượng, còn trống vao nhiêu. Bao nhiêu datanode còn sống,...
- hdfs namenode -format: Format lại toàn bộ hdfs. dữ liệu sẽ bị mất

How to install Hadoop & Spark on MACOS

Source:
https://www.quickprogrammingtips.com/big-data/how-to-install-hadoop-on-mac-os-x-el-capitan.html

- Install Hadoop, Spark
- Tổng quan Hadoop ,HDFS, MapReduce,
- Apache Spark
- Case Study

This tutorial uses **pseudo-distributed mode for running hadoop** which allows us to use a single machine to run different components of the system in different Java processes. We will also configure YARN as the resource manager for running jobs on hadoop.

- Source: https://www.quickprogrammingtips.com/big-data/how-to-install-hadoop-on-mac-os-x-el-capitan.html

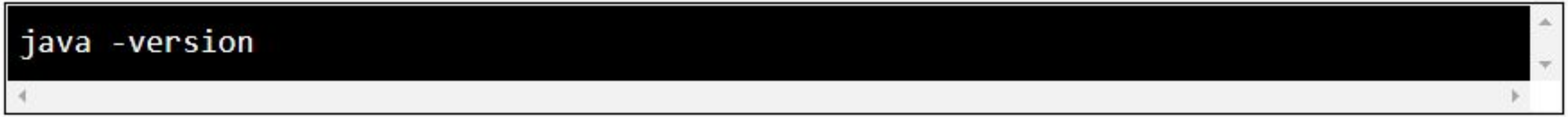
Versions

- Java 7 or higher. Java 8 is recommended.
- Hadoop 2.7.3 or higher.

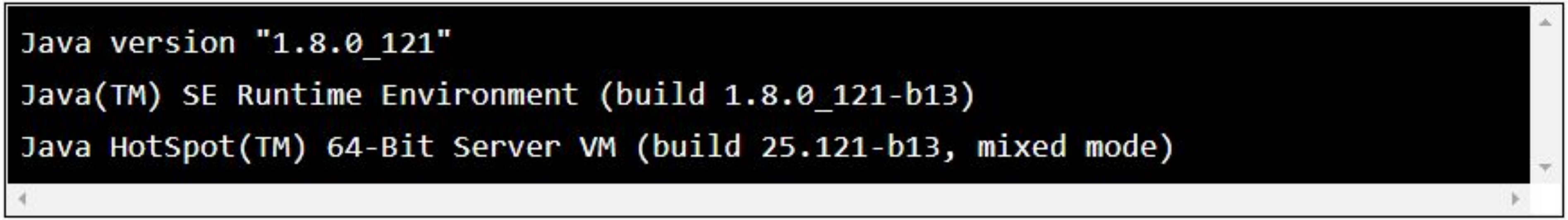
- Source: https://www.quickprogrammingtips.com/big-data/how-to-install-hadoop-on-mac-os-x-el-capitan.html

Step 1: Install Java

Verify the Java version installed on the system.



```
java -version
```



```
Java version "1.8.0_121"
Java(TM) SE Runtime Environment (build 1.8.0_121-b13)
Java HotSpot(TM) 64-Bit Server VM (build 25.121-b13, mixed mode)
```

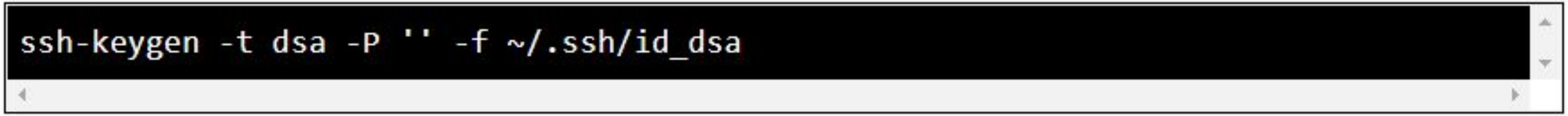
- Source: https://www.quickprogrammingtips.com/big-data/how-to-install-hadoop-on-mac-os-x-el-capitan.html

Step 2: Configure SSH

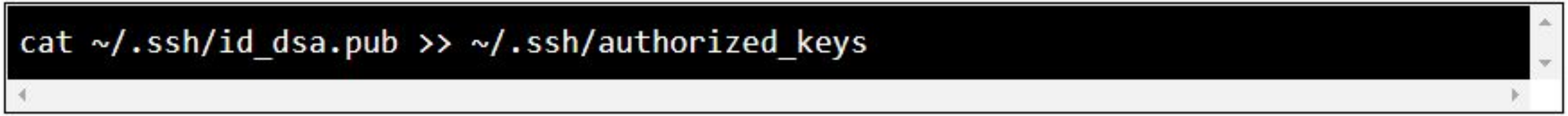
When hadoop is installed in distributed mode, it uses a password less SSH for master to slave communication. To enable SSH daemon in mac, go to **System Preferences => Sharing**. Then click on **Remote Login** to enable SSH. Execute the following commands on the terminal to enable password less login to SSH,

- Source: https://www.quickprogrammingtips.com/big-data/how-to-install-hadoop-on-mac-os-x-el-capitan.html

Step 2: Configure SSH



```
ssh-keygen -t dsa -P '' -f ~/.ssh/id_dsa
```



```
cat ~/.ssh/id_dsa.pub >> ~/.ssh/authorized_keys
```

- Source: https://www.quickprogrammingtips.com/big-data/how-to-install-hadoop-on-mac-os-x-el-capitan.html

Step 3: Install Hadoop

Download hadoop 2.7.3 binary zip file from this link (200MB). Extract the contents of the zip to a folder of your choice.

(http://hadoop.apache.org/releases.html)

- Source: https://www.quickprogrammingtips.com/big-data/how-to-install-hadoop-on-mac-os-x-el-capitan.html

Step 4: Configure Hadoop

1. Configure the location of our Java installation in etc/hadoop/hadoop-env.sh

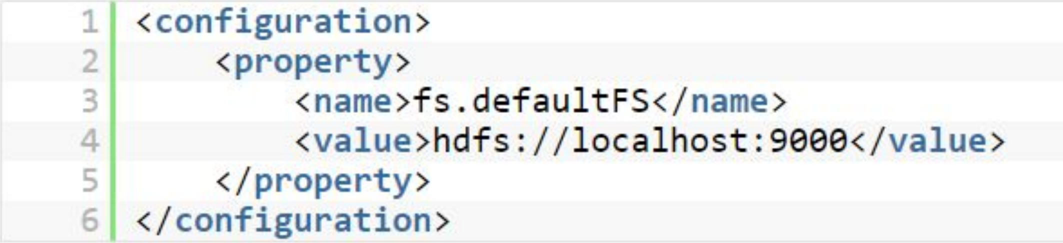
```
export
JAVA_HOME=/Library/Java/JavaVirtualMachines/jdk1.8.0_121.jdk
/Contents/Home
```

- Source: https://www.quickprogrammingtips.com/big-data/how-to-install-hadoop-on-mac-os-x-el-capitan.html

Step 4: Configure Hadoop

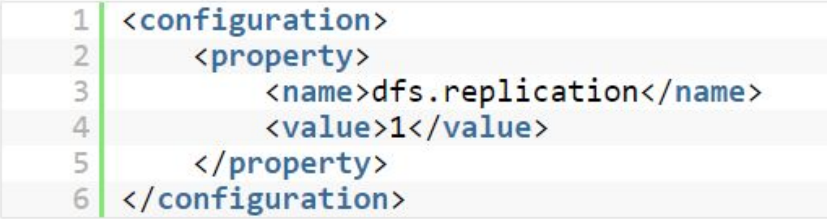
2. Modify various hadoop configuration files to properly setup hadoop and yarn. These files are located in etc/hadoop.

etc/hadoop/core-site.xml



```
<configuration>
    <property>
        <name>fs.defaultFS</name>
        <value>hdfs://localhost:9000</value>
    </property>
</configuration>
```

etc/hadoop/hdfs-site.xml

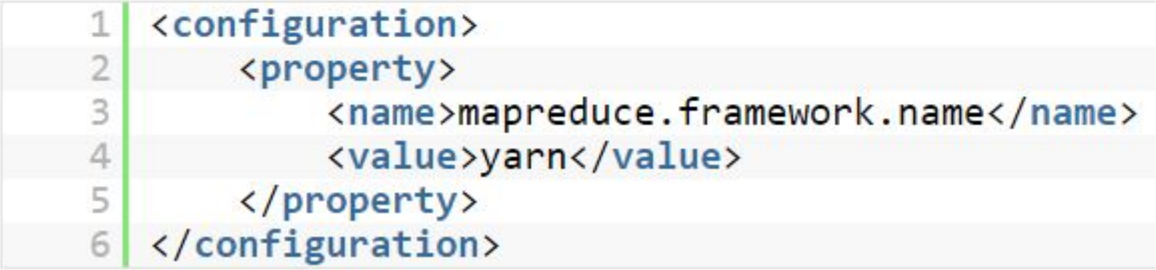


```
<configuration>
    <property>
        <name>dfs.replication</name>
        <value>1</value>
    </property>
</configuration>
```

- Source: https://www.quickprogrammingtips.com/big-data/how-to-install-hadoop-on-mac-os-x-el-capitan.html

Step 4: Configure Hadoop

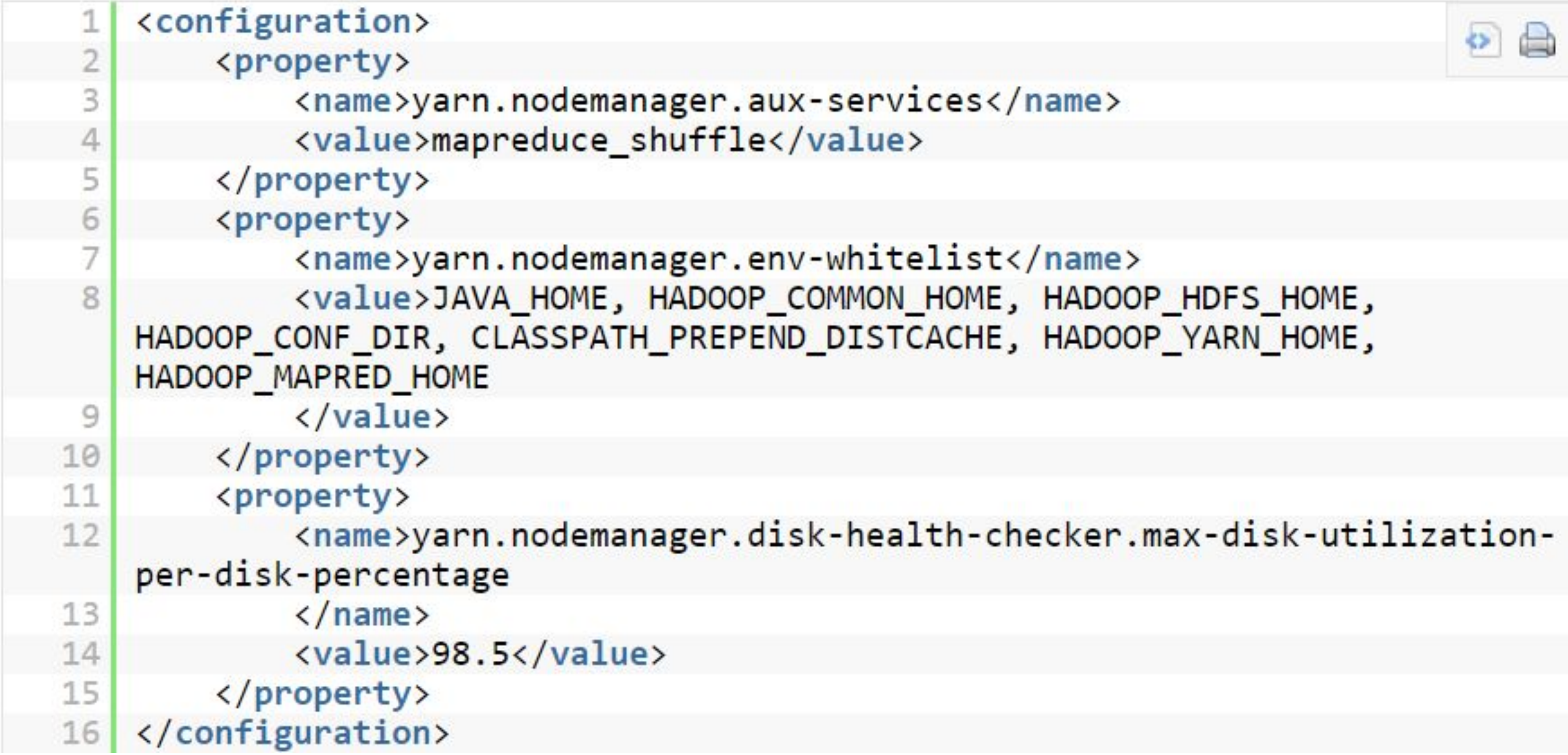
etc/hadoop/mapred-site.xml



```
<configuration>
    <property>
        <name>mapreduce.framework.name</name>
        <value>yarn</value>
    </property>
</configuration>
```

- Source: https://www.quickprogrammingtips.com/big-data/how-to-install-hadoop-on-mac-os-x-el-capitan.html

etc/hadoop/yarn-site.xml

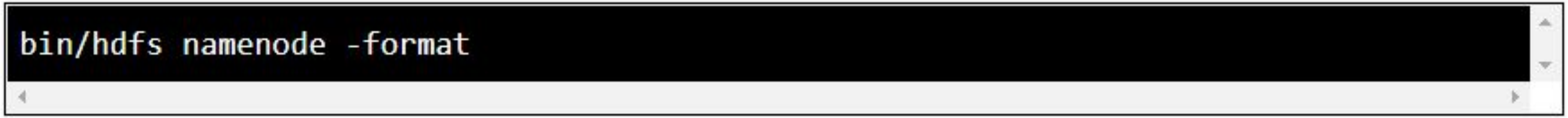


```
<configuration>
    <property>
        <name>yarn.nodemanager.aux-services</name>
        <value>mapreduce_shuffle</value>
    </property>
    <property>
        <name>yarn.nodemanager.env-whitelist</name>
        <value>JAVA_HOME, HADOOP_COMMON_HOME, HADOOP_HDFS_HOME,
HADOOP_CONF_DIR, CLASSPATH_PREPEND_DISTCACHE, HADOOP_YARN_HOME,
HADOOP_MAPRED_HOME
        </value>
    </property>
    <property>
        <name>yarn.nodemanager.disk-health-checker.max-disk-utilization-
per-disk-percentage
        </name>
        <value>98.5</value>
    </property>
</configuration>
```

If disk utilization goes above the configured threshold, yarn will report the node instance as unhealthy nodes with error **"local-dirs are bad"**.

Step 5: Initialize Hadoop Cluster

- From a terminal window switch to the hadoop home folder
- Run the following command to initialize the metadata for the hadoop cluster. This formats the hdfs file system and configures it on the local system. By default, files are created in /tmp/hadoop-<username> folder.

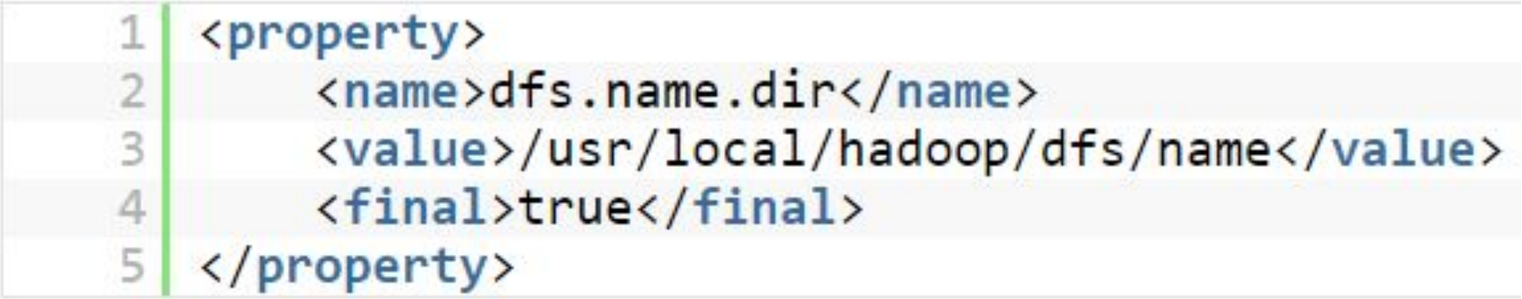


```
bin/hdfs namenode -format
```

- Source: https://www.quickprogrammingtips.com/big-data/how-to-install-hadoop-on-mac-os-x-el-capitan.html

Step 5: Initialize Hadoop Cluster

It is possible to modify the default location of name node configuration by adding the following property in the **hdfs-site.xml** file. Similarly the hdfs data block storage location can be changed using **dfs.data.dir** property.

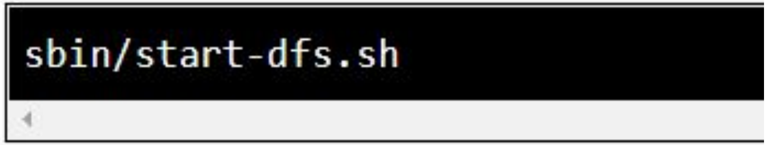


```
<property>
     <name>dfs.name.dir</name>
     <value>/usr/local/hadoop/dfs/name</value>
     <final>true</final>
</property>
```

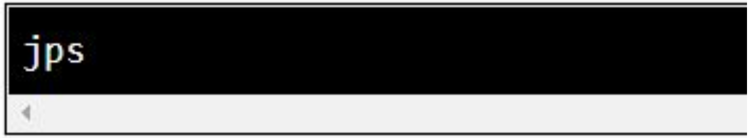
- Source: https://www.quickprogrammingtips.com/big-data/how-to-install-hadoop-on-mac-os-x-el-capitan.html

Step 6: Start Hadoop Cluster

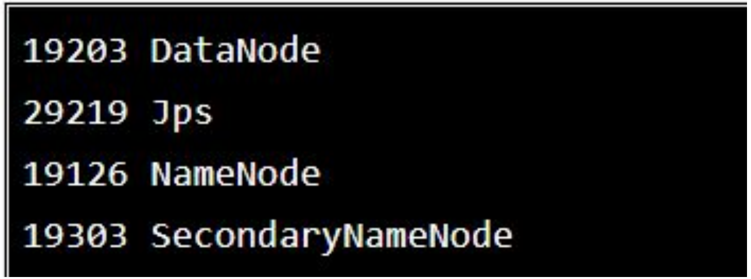
- Run the following command from terminal (after switching to hadoop home folder) to start the hadoop cluster. This starts name node and data node on the local system.
- To verify that the namenode and datanode daemons are running, execute the following command on the terminal. This displays running Java processes on the system.



```
sbin/start-dfs.sh
```



```
jps
```

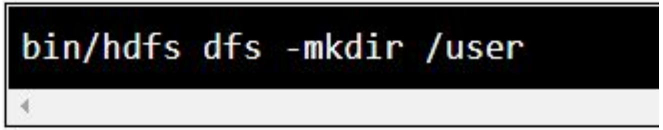


```
19203 DataNode
29219 Jps
19126 NameNode
19303 SecondaryNameNode
```

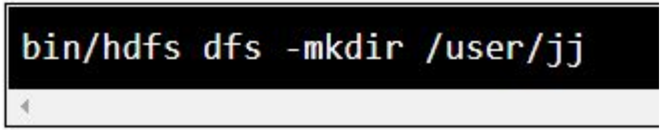
- Source: https://www.quickprogrammingtips.com/big-data/how-to-install-hadoop-on-mac-os-x-el-capitan.html

Step 7: Configure HDFS Home Directory

The home directory is of the form – /user/<username>. My user id on the mac system is jj. Replace it with your user name. Run the following commands on the terminal,



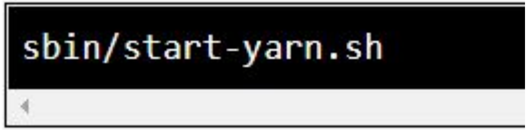
```
bin/hdfs dfs -mkdir /user
```



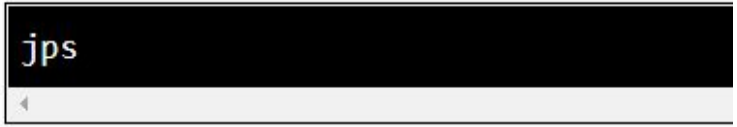
```
bin/hdfs dfs -mkdir /user/jj
```

Step 8: Run YARN Manager

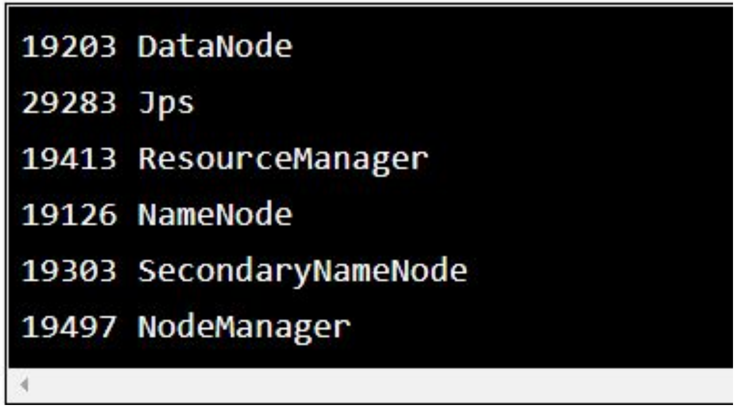
- Start YARN resource manager and node manager instances by running the following command on the terminal,
- Run jps command again to verify all the running processes,



```
sbin/start-yarn.sh
```



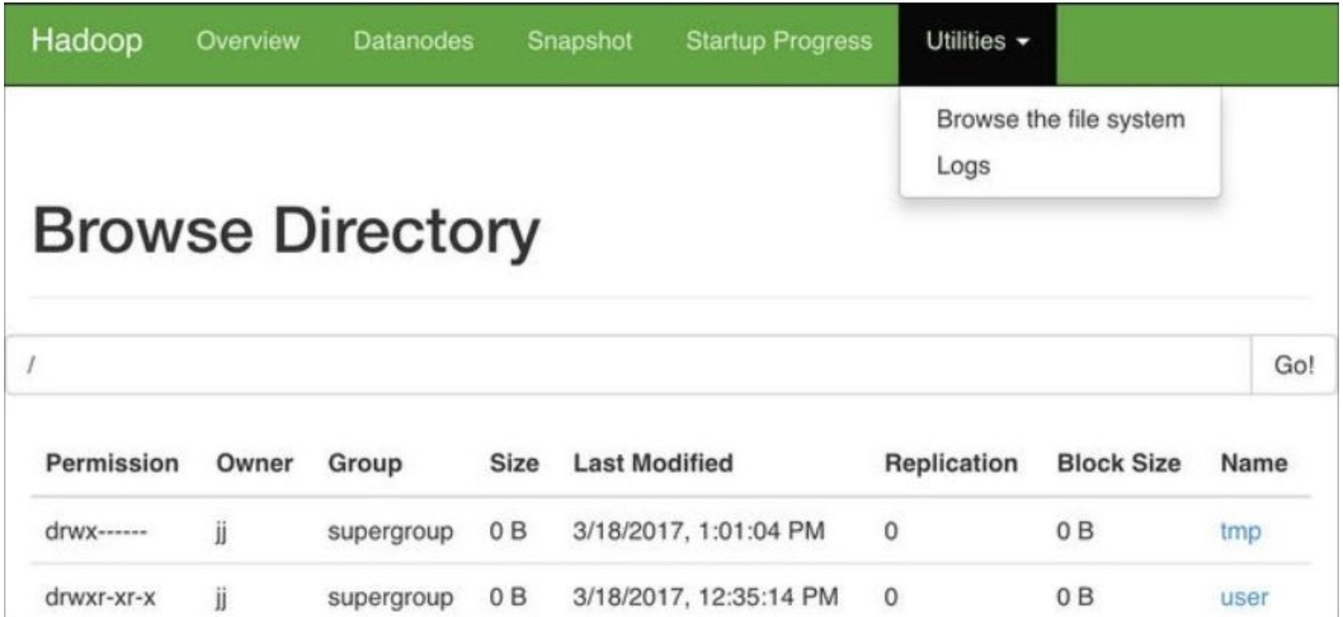
```
jps
```



```
19203 DataNode
29283 Jps
19413 ResourceManager
19126 NameNode
19303 SecondaryNameNode
19497 NodeManager
```

Step 9: Verify Hadoop Installation

Access the URL http://localhost:50070/dfshealth.html to view hadoop name node configuration. You can also navigate the hdfs file system using the menu **Utilities => Browse the file system**.

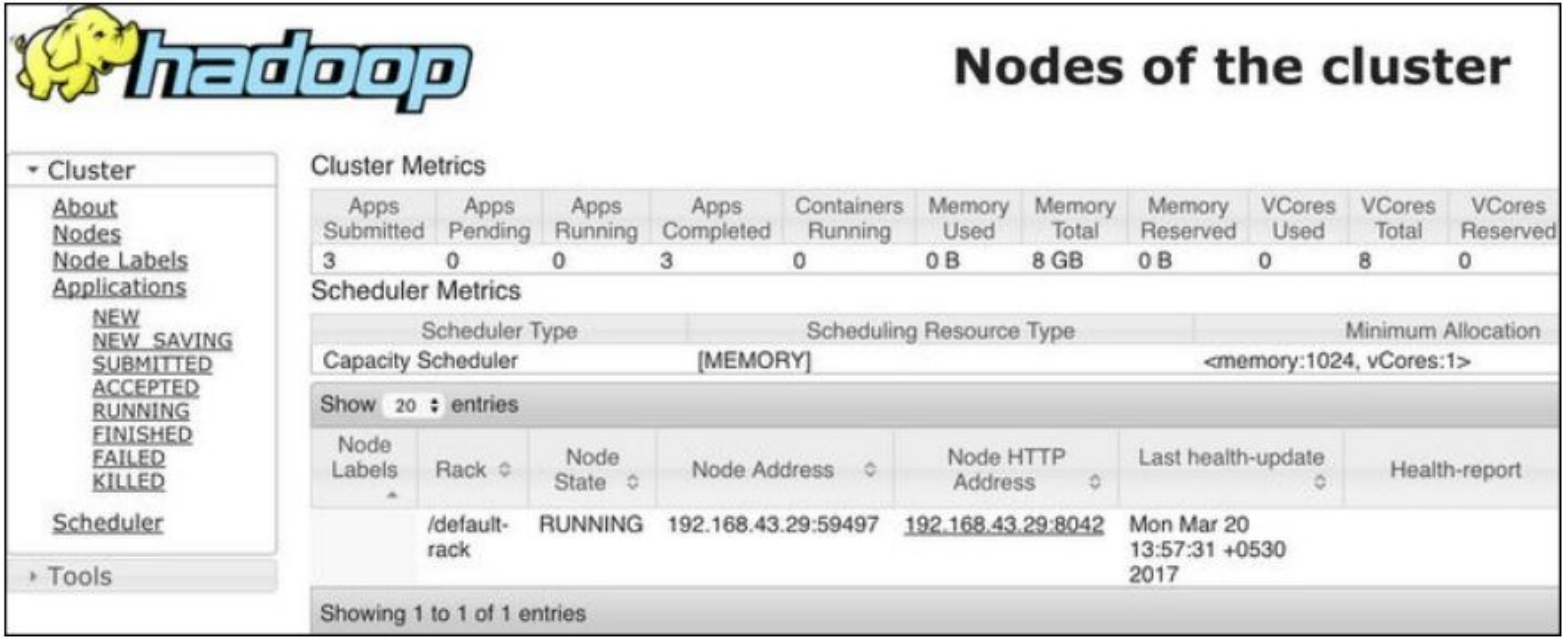


Browse Directory

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwx------	jj	supergroup	0 B	3/18/2017, 1:01:04 PM	0	0 B	tmp
drwxr-xr-x	jj	supergroup	0 B	3/18/2017, 12:35:14 PM	0	0 B	user

Step 9: Verify Hadoop Installation

Access the URL http://localhost:8088/cluster to view the hadoop cluster details through YARN resource manager.



Nodes of the cluster

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total	VCores Reserved
3	0	0	3	0	0 B	8 GB	0 B	0	8	0

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation
Capacity Scheduler	[MEMORY]	<memory:1024, vCores:1>

Show 20 entries

Node Labels	Rack	Node State	Node Address	Node HTTP Address	Last health-update	Health-report
	/default-rack	RUNNING	192.168.43.29:59497	192.168.43.29:8042	Mon Mar 20 13:57:31 +0530 2017	

Showing 1 to 1 of 1 entries

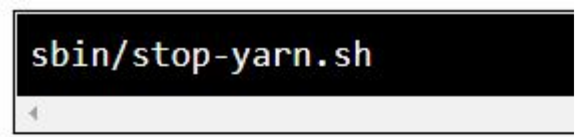
Step 10: Run Sample MapReduce Job

Run Sample MapReduce Job

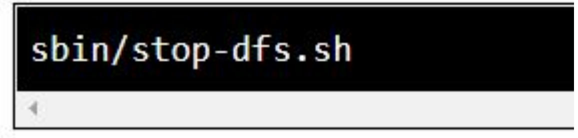
Step 10: Run Sample MapReduce Job

Step 11: Stop Hadoop/YARN cluster

Run the following commands to stop hadoop/YARN daemons. This stops name node, data node, node manager and resource manager.



```
sbin/stop-yarn.sh
```



```
sbin/stop-dfs.sh
```

References

1. Paul Zikopoulos, Chris Eaton. 2011. Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data (1st ed.). McGraw-Hill Osborne Media.
2. https://www.tutorialspoint.com/hadoop/hadoop_hdfs_overview.htm
3. https://www.quickprogrammingtips.com/big-data/how-to-install-hadoop-on-mac-os-x-el-capitan.html
4. http://blog.prabeeshk.com/blog/2016/12/07/install-apache-spark-2-on-ubuntu-16-dot-04-and-mac-os/
5. http://data-flair.training/blogs/top-hadoop-hdfs-commands-tutorial/