

# HDFS TUTORIAL

---

Hands-on Session

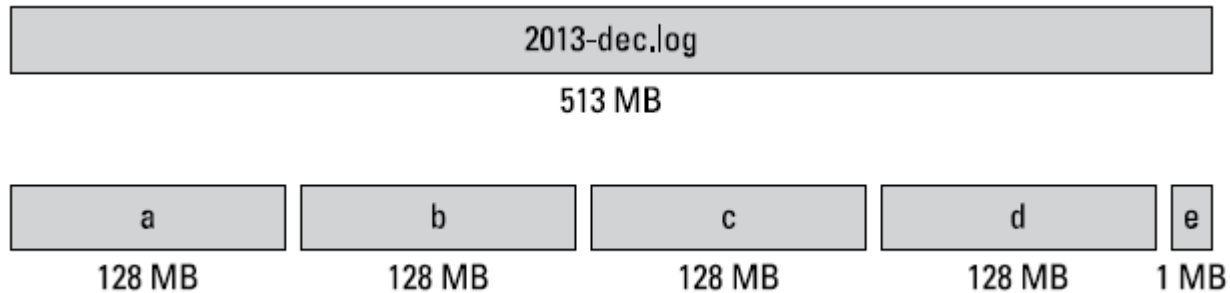
by Suchitra Jayaprakash  
suchitra@cmi.ac.in

# HADOOP COMPONENTS

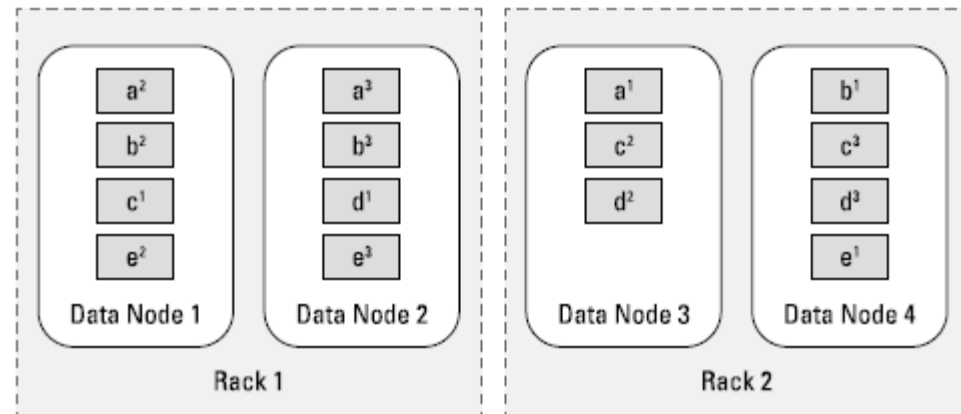
- Hadoop is a framework that allows distributed processing of large data sets across clusters of computers using simple programming models.
- Core Components of Apache Hadoop:
  - HDFS
    - Storage component of Hadoop
  - MapReduce
    - Programming model

# HDFS

- HDFS is distributed file system.
- It stores data by splitting files into blocks.



- Blocks are distributed across multiple nodes.
- Replicates blocks on multiple nodes.
- It provides fault tolerant storage.



(source: Hadoop for Dummies)

# HDFS Architecture

- HDFS follows master/slave architecture.

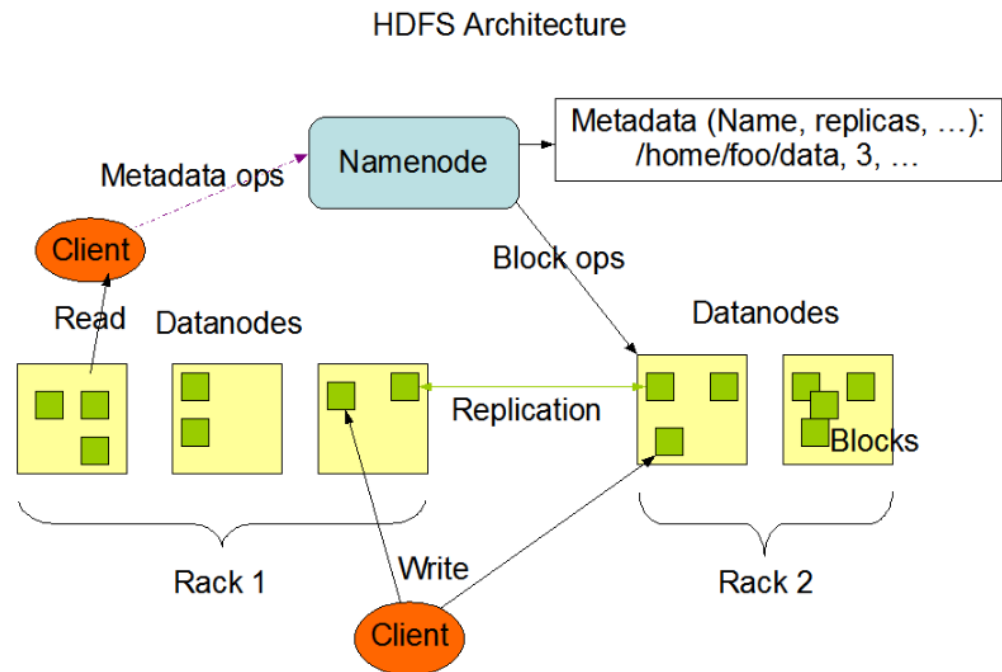
- Master Node / Name Node

manages the file system name space (meta data) and regulates access to files by clients.

- Slave Node / Data Node

Stores data blocks attached to a node. Block size – 64 MB to 128 MB.

- HDFS follows Write once and Read multiple times.



(source: Cloudera website)

# START CLOUDERA

- Start cloudera quick start

```
docker run --hostname=quickstart.cloudera --privileged=true -t -i --  
publish-all=true -p 50070:50070 -p 8088:8088 -p 50075:50075  
cloudera/quickstart /usr/bin/docker-quickstart
```

Port	Purpose
50070	Name node web interface
8088	job tracker :- yarn
50075	Data node

# HDFS Shell Command

- mkdir command

**hadoop fs -mkdir DATA**

**hadoop fs -mkdir DATA2**

It will create a new directory named DATA under the location /user/root

- LS command

**hadoop fs -ls**

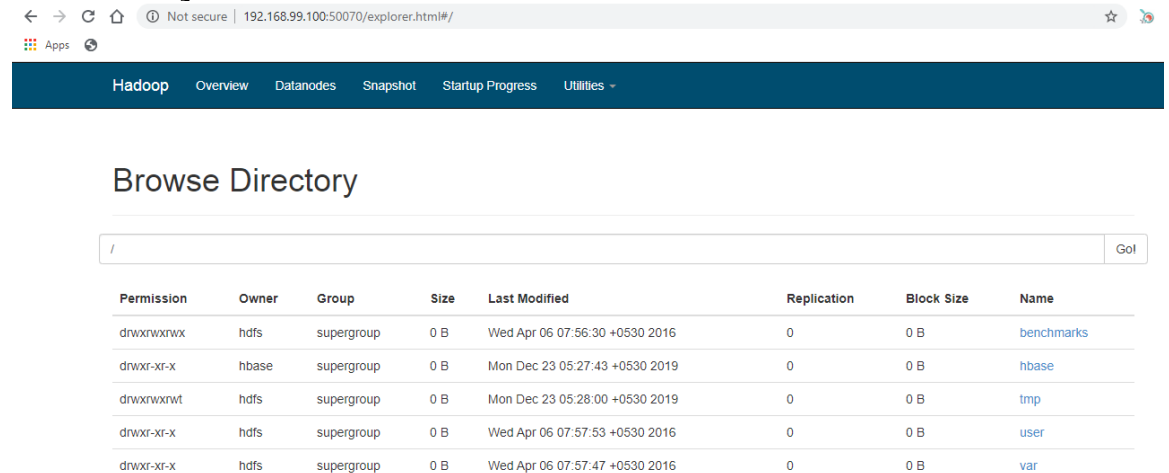
It will list all the available files and subdirectories under default directory.

- Name node Web UI

To see name node web interface , Open <http://192.168.99.100:50070/> for docker tool box or <http://localhost:50070> in browser.

# HDFS Shell Command

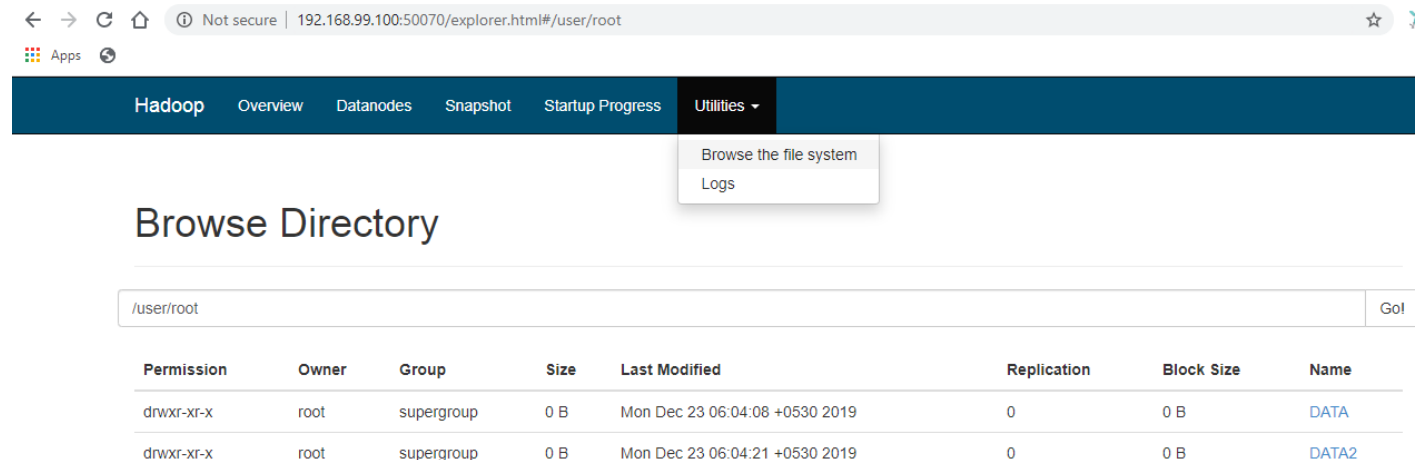
- Click on utilities : browse the file system.



The screenshot shows the Hadoop web interface. The top navigation bar includes 'Hadoop', 'Overview', 'Datanodes', 'Snapshot', 'Startup Progress', and 'Utilities'. The 'Utilities' menu is open, showing 'Browse the file system' and 'Logs'. The main content area is titled 'Browse Directory' and shows a table of files in the root directory.

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwxrwxrwx	hdfs	supergroup	0 B	Wed Apr 06 07:56:30 +0530 2016	0	0 B	<a href="#">benchmarks</a>
drwxr-xr-x	hbase	supergroup	0 B	Mon Dec 23 05:27:43 +0530 2019	0	0 B	<a href="#">hbase</a>
drwxrwxrwt	hdfs	supergroup	0 B	Mon Dec 23 05:28:00 +0530 2019	0	0 B	<a href="#">tmp</a>
drwxr-xr-x	hdfs	supergroup	0 B	Wed Apr 06 07:57:53 +0530 2016	0	0 B	<a href="#">user</a>
drwxr-xr-x	hdfs	supergroup	0 B	Wed Apr 06 07:57:47 +0530 2016	0	0 B	<a href="#">var</a>

- Click on user then root,



The screenshot shows the Hadoop web interface with the 'Utilities' menu open. The main content area is titled 'Browse Directory' and shows a table of files in the /user/root directory.

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwxr-xr-x	root	supergroup	0 B	Mon Dec 23 06:04:08 +0530 2019	0	0 B	<a href="#">DATA</a>
drwxr-xr-x	root	supergroup	0 B	Mon Dec 23 06:04:21 +0530 2019	0	0 B	<a href="#">DATA2</a>

# HDFS Shell Command

- Copy files from HOST

**docker cp c:/sample.txt <containerid>:/tmp/sample.txt**

copy files from host machine to docker container. (run the command in a different docker terminal).

- Put command

**hadoop fs -put /tmp/sample.txt DATA**

It uploads files to hadoop distributed file system.

- Get command

**hadoop fs -get DATA/sample.txt**

copies the files to the local filesystem. Run ls -ltr to check file on docker container.



# HDFS Shell Command

- copyFromLocal command

**hadoop fs -copyFromLocal /tmp/sample1.txt DATA**

copies file from local file system to HDFS location.

- moveFromLocal command

**hadoop fs -moveFromLocal /tmp/sample2.txt DATA**

copies file from local file system to given destination and source file is deleted.

- Print block count

**hadoop fsck /user/root/DATA -files -blocks**

# HDFS Shell Command

- Mv command

**hadoop fs -mv DATA/sample2.txt DATA2/sample2.txt**  
move file from one directory to other directory.

- rm command

**hadoop fs -rm DATA2/Sample3.txt**  
removes the file or empty directory in the given path.

- touchz command

**hadoop fs -touchz DATA2/sample4.txt**  
creates files in given location.

# HDFS Shell Command

- tail command

**hadoop fs -tail DATA/output.txt**

Display trailing Kilobytes of file content.

- Cat command

**hadoop fs -cat DATA/sample1.txt**

Copies file content to *stdout*

- CP command

**hadoop fs -cp DATA2/output.txt DATA3/output.txt**

Copy files from source to destination.

# HDFS Shell Command

- DU command

**hadoop fs -du DATA**

Displays disk usage, in bytes, for all the files in the current folder.

Stat command

**hadoop fs -stat DATA**

Prints information about folder.

chmod command

**hadoop fs -chmod -R 777 DATA**

Changes the file permissions

# Small files problem

- Issues :
  - Processing too many small files is a problem in Hadoop.
  - Overhead for name node.
  - Map task process a block of input at a time. So more number of mapper job.
- Solution : Sequence file format
  - Sequence file is a flat file consisting of binary key/value pairs.
  - Filename is used as key and file content as value.
  - SequenceFile java API provides SequenceFile.Writer, SequenceFile.Reader and SequenceFile.Sorter classes for writing, reading and sorting data.

SequenceFile File Layout

Data	Key	Value	Key	Value	Key	Value	Key	Value
------	-----	-------	-----	-------	-----	-------	-----	-------

# HDFS Shell Command

- How to merge content of two text files?

```
hadoop fs -cat DATA/sample1.txt DATA/sample2.txt
```

```
hadoop fs -cat DATA/* | hadoop fs -put DATA/output.txt
```

# HDFS Config files

- All HDFS related configuration are done by adding or updating properties in xml files:

hdfs-site.xml

core-site.xml

- Type below command to see config file

```
ls -l /etc/hadoop/conf/
```

```
less /etc/hadoop/conf/hdfs-site.xml
```

```
less /etc/hadoop/conf/core-site.xml
```

# Commissioning and Decommissioning Data Node

Step	Add Data Node	Delete Data Node
1	stop hadoop cluster stop-dfs.sh	stop hadoop cluster stop-dfs.sh
2	edit <i>yarn-site.xml</i> in Resource Manager node <pre>&lt;property&gt;   &lt;name&gt;yarn.resourcemanager.nodes.include-path&lt;/name&gt;   &lt;value&gt;/home/hadoop/includes&lt;/value&gt; &lt;/property&gt;</pre>	edit <i>yarn-site.xml</i> in Resource Manager node <pre>&lt;property&gt;   &lt;name&gt;yarn.resourcemanager.node.exclude-path&lt;/name&gt;   &lt;value&gt;/home/hadoop/excludes&lt;/value&gt; &lt;/property&gt;</pre>
3	edit the <i>hdfs-site.xml</i> file in Namenode <pre>&lt;property&gt;   &lt;name&gt;dfs.hosts&lt;/name&gt;   &lt;value&gt;/home/hadoop/includes&lt;/value&gt; &lt;/property&gt;</pre>	edit the <i>hdfs-site.xml</i> file in Namenode <pre>&lt;property&gt;   &lt;name&gt;dfs.hosts.exclude&lt;/name&gt;   &lt;value&gt;/home/hadoop/excludes&lt;/value&gt; &lt;/property&gt;</pre>
4	start cluster run namenode and resource manager start-dfs.sh start-yarn.sh	start cluster run namenode and resource manager start-dfs.sh start-yarn.sh
5	Add the Datanode IP address, in include file on both Resource manager and Namenode machine.	Add the Datanode IP address, in exclude file on both Resource manager and Namenode machine.
6	Reload property of Ressource manage an namenode  yarn radmin -refreshNodes  hdfs dfsadmin –refreshNodes	Reload property of Ressource manage an namenode  yarn radmin -refreshNodes  hdfs dfsadmin –refreshNodes



# Quiz 2

- CMI is known for its pre-poll statistical analysis. If the pre-poll sample data from all around the world for the past 10 years is stored in HDFS as two files of size 120 MB and 5 MB, how many blocks will be created in total?
- Assume block size is 64MB & replication factor is 3
- A) 6
- B) 9
- C) 3
- D) 5

THANK YOU