Installation of Cloudera CDH, HDFS Basics, Hadoop Ecosystem Tools Overview and Case Study on

Role of Big Data in any one of the following domain:

(Education/Government/Insurance/Hotel/Healthcare/Banking/stock market /Pharma etc)

| Sr. No | Date | Title | Grade | Sign |
|---|---|---|---|---|
| 1 | 27-07-2021 | **Demonstration of Basic HDFS Commands** | | |

Date of submission:: 3/8/2021

**Aim : To study Hadoop Ecosystem and to demonstrate Basic Hadoop Commands.**

**Theory** Hadoop File System was developed using distributed file system design. It is run on commodity hardware. Unlike other distributed systems, HDFS is highly fault tolerant and designed using low-cost hardware.

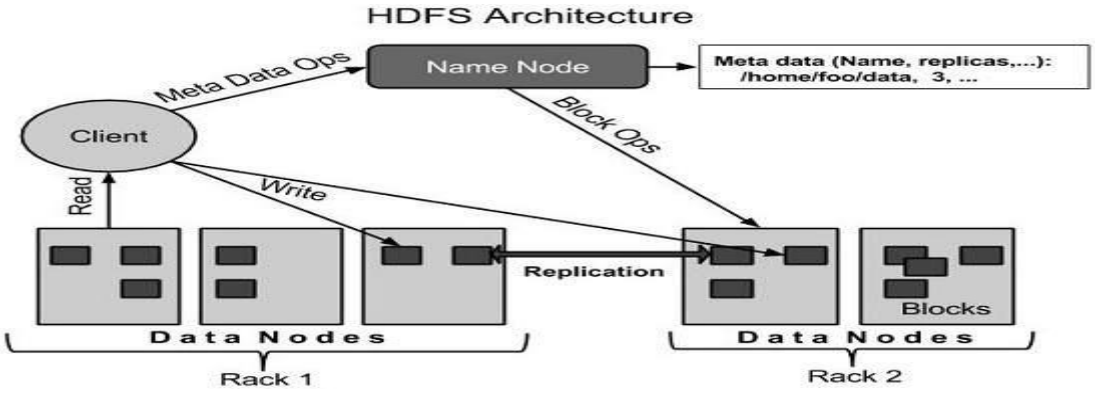
HDFS holds very large amount of data and provides easier access. To store such huge data, the files are stored across multiple machines. These files are stored in redundant fashion to rescue the system from possible data losses in case of failure. HDFS also makes applications available to parallel processing.

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**

Given below is the architecture of a Hadoop File System.




HDFS follows the master-slave architecture and it has the following elements.

The main components of HDFS are as described below:

**1. NameNode** is the master of the system. It maintains the name system i.e. the directories and files and manages the blocks which are present on the DataNodes.

**2. DataNodes** are the slaves which are deployed on each machine and provide the actual storage. They are responsible for serving read and write requests for the clients

**Secondary NameNode** is responsible for performing periodic checkpoints. In the event of NameNode failure, the NameNode can be restarted using the checkpoint

**Block**

Generally the user data is stored in the files of HDFS. 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. In other words, the minimum amount of data that HDFS can read or write is called a Block. The default block size is 64MB, but it can be increased as per the need to change in HDFS configuration.

**Goals of HDFS**

- **Fault detection and recovery** : Since HDFS includes a large number of commodity hardware, failure of components is frequent. Therefore HDFS should have mechanisms for quick and automatic fault detection and recovery.
- **Huge datasets** : HDFS should have hundreds of nodes per cluster to manage the applications having huge datasets.
- **Hardware at data** : A requested task can be done efficiently, when the computation takes place near the data. Especially where huge datasets are involved, it reduces the network traffic and increases the throughput.

**Execution :**

**1.Create a directory in HDFS at given path(s).**

Usage:

hadoop fs -mkdir <paths>
Example:
hadoop fs -mkdir /user/saurzcode/dir1
hadoop fs -mkdir /user/saurzcode/dir1

**2. List the contents of a directory.**

Usage :
hadoop fs -ls <args>
Example:
H adoop fs -ls /user/saurzcode

**3. Upload and download a file in HDFS.**

***Upload*:**

**hadoop fs -put:**

Copy single src file, or multiple src files from local file system to the Hadoop data file system

Usage:
hadoop fs -put <localsrc> ... <HDFS_dest_Path>
Example:
hadoop fs -put /home/saurzcode/Samplefile.txt /user/saurzcode/dir3/

**Download:**

**hadoop fs -get:**

Copies/Downloads files to the local file system

Usage:
hadoop fs -get <hdfs_src> <localdst>
Example:
hadoop fs -get /user/saurzcode/dir3/Samplefile.txt /home/

**4. See contents of a file**

Same as unix cat command:

Usage:
hadoop fs -cat <path[filename]>
Example:
hadoop fs -cat /user/saurzcode/dir1/abc.txt

**5. Copy a file from source to destination**

This command allows multiple sources as well in which case the destination must be a directory.

Usage:

hadoop fs -cp <source> <dest>
Example:
hadoop fs -cp /user/saurzcode/dir1/abc.txt /user/saurzcode/dir2

**6. Copy a file from/To Local file system to HDFS**

**copyFromLocal**

Usage:
hadoop fs -copyFromLocal <localsrc> URI
Example:
hadoop fs -copyFromLocal /home/saurzcode/abc.txt  /user/saurzcode/abc.txt

Similar to put command, except that the source is restricted to a local file reference.

**copyToLocal**

Usage:
hadoop fs -copyToLocal [-ignorecrc] [-crc] URI <localdst>

Similar to get command, except that the destination is restricted to a local file reference.

**7. Move file from source to destination.**

Note:- Moving files across filesystem is not permitted.

Usage :
hadoop fs -mv <src> <dest>
Example:
hadoop fs -mv /user/saurzcode/dir1/abc.txt /user/saurzcode/dir2

**8. Remove a file or directory in HDFS.**

Remove files specified as argument. Deletes directory only when it is empty

Usage :
hadoop fs -rm <arg>
Example:
hadoop fs -rm /user/saurzcode/dir1/abc.txt

**Recursive version of delete.**

Usage :
hadoop fs -rmr <arg>
Example:
hadoop fs -rmr /user/saurzcode/

**9. Display last few lines of a file.**

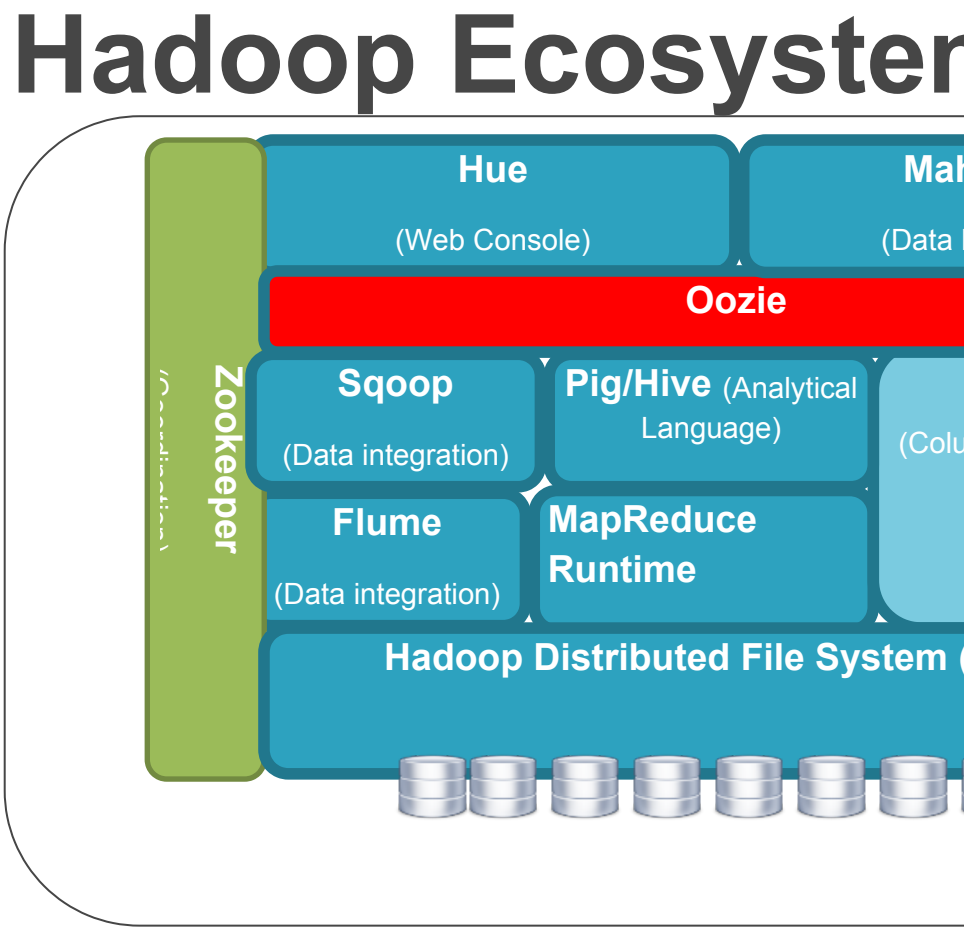
Similar to tail command in Unix.

Usage :
hadoop fs -tail <path[filename]>
Example:
hadoop fs -tail /user/saurzcode/dir1/abc.txt

**10. Display the aggregate length of a file.**

Usage :
hadoop fs -du <path>
Example:
hadoop fs -du /user/saurzcode/dir1/abc.txt

**Theory part 2 :**



Explain all above echo system components

**Theory Part3 :**

Role of Big Data in any one of the following domain:

(Education/Government/Insurance/Hotel/Healthcare/Banking/stock market /Pharma etc)

**Conclusion :**