Vivekanand Education Society's
Institute of Technology
Approved by AICTE & Affiliated to University of Mumbai

Artificial Intelligence and Data Science Department

BDA/Odd Sem 2023-23/Experiment 1

Name : Shreya Singh	**Class/Roll No. : 55/D16AD**	**Grade :**

Title of Experiment : To study Hadoop Ecosystem and to demonstrate Basic Hadoop Commands.

Objective of Experiment :
Acquire a foundational understanding of the Hadoop ecosystem and its components, focusing on basic Hadoop commands for effective data management and processing.

Outcome of Experiment :
We successfully installed Hadoop Eco-system and executed basic Hadoop commands on it.

Problem Statement :
Learn how to use Hadoop for managing and processing big data by mastering essential commands and concepts

Description / Theory :
Hadoop:
Hadoop is an open-source framework designed for processing and managing large volumes of data across distributed computing clusters. It was initially developed by the Apache Software Foundation and is widely used in the field of big data analytics.

Purpose and Use:

Hadoop is used to process and store massive amounts of data in a scalable and reliable manner. It is particularly well-suited for handling unstructured or semi-structured data, such as text, images, videos, and log files. Hadoop is employed for tasks like batch processing, data warehousing, data transformation, and more. It's commonly used in industries like finance, healthcare, e-commerce, and social media to extract valuable insights from vast datasets.

Features:

Distributed Storage: Hadoop's HDFS (Hadoop Distributed File System) breaks data into blocks and stores copies across multiple nodes in a cluster, ensuring redundancy and fault tolerance.

Distributed Processing: The MapReduce programming model allows data processing tasks to be divided into smaller tasks that are distributed across cluster nodes, enabling parallel processing and efficient resource utilization.

Scalability: Hadoop can easily scale by adding more nodes to the cluster, allowing it to handle growing data volumes without significant changes to the architecture.

Fault Tolerance: Hadoop maintains data redundancy, so even if a node or hardware fails, the system can still function without data loss. It automatically replicates data across nodes.

Flexibility: Hadoop can process structured, semi-structured, and unstructured data. This versatility makes it suitable for various data types.

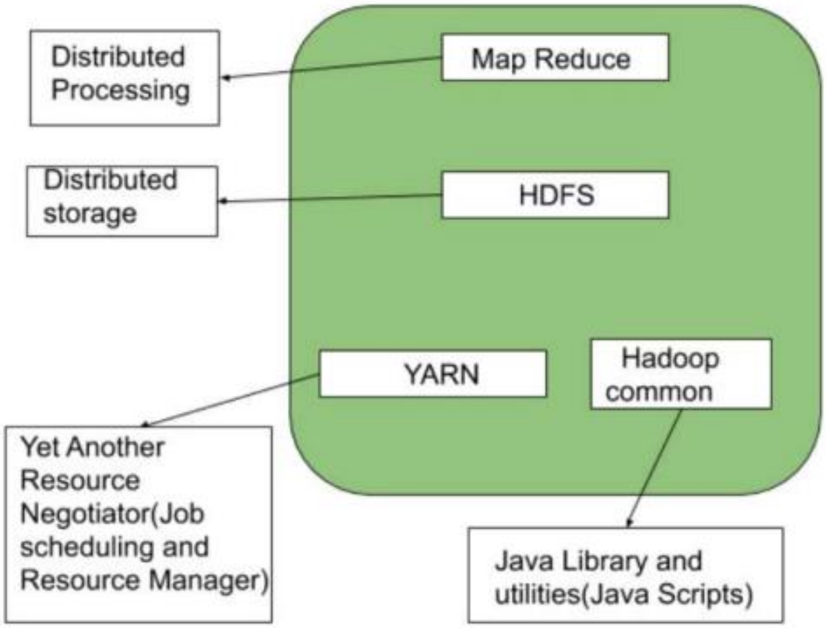
Ecosystem: Hadoop has a rich ecosystem of tools and frameworks, including Hive for querying data using a SQL-like language, Pig for scripting data processing, and Spark for in-memory processing, which enhances its capabilities.

Cost-Effectiveness: Hadoop can run on commodity hardware, making it more costeffective than traditional data processing systems that require specialized hardware.

Open Source: Being open-source, Hadoop is freely available for use, modification, and distribution, which has contributed to its widespread adoption.

Data Locality: Hadoop's data processing paradigm strives to process data on the nodes
where it resides. This reduces network traffic and enhances performance.
Highly Parallel Processing: Hadoop's MapReduce model and its successors enable distributed data processing, facilitating efficient parallelism to speed up computation.

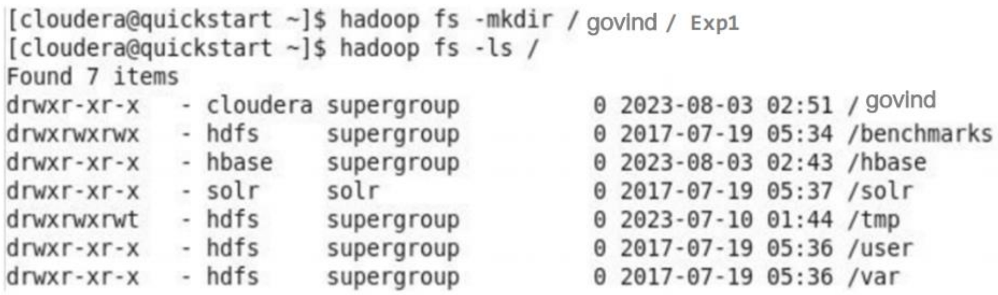
Hadoop Architecture:


Distributed Processing
Map Reduce
Distributed storage
HDFS
YARN
Hadoop common
Yet Another Resource Negotiator(Job scheduling and Resource Manager)
Java Library and utilities(Java Scripts)

Output:

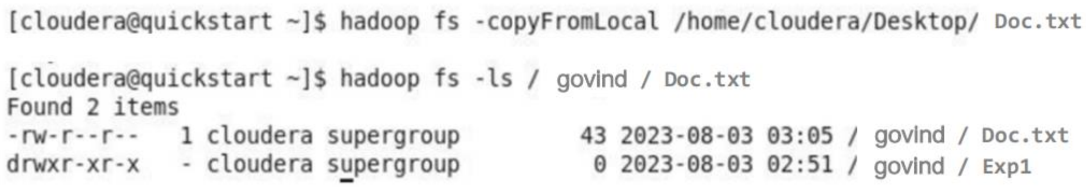
HDFS Commands:

1. **Create Directory and list items of Directory:**



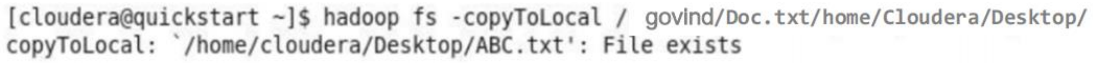
```
[cloudera@quickstart ~]$ hadoop fs -mkdir / govind / Exp1
[cloudera@quickstart ~]$ hadoop fs -ls /
Found 7 items
drwxr-xr-x   - cloudera supergroup          0 2023-08-03 02:51 / govind
drwxrwxrwx   - hdfs     supergroup          0 2017-07-19 05:34 /benchmarks
drwxr-xr-x   - hbase    supergroup          0 2023-08-03 02:43 /hbase
drwxr-xr-x   - solr     solr                0 2017-07-19 05:37 /solr
drwxrwxrwt   - hdfs     supergroup          0 2023-07-10 01:44 /tmp
drwxr-xr-x   - hdfs     supergroup          0 2017-07-19 05:36 /user
drwxr-xr-x   - hdfs     supergroup          0 2017-07-19 05:36 /var
```

2. **Copy File to HDFS:**



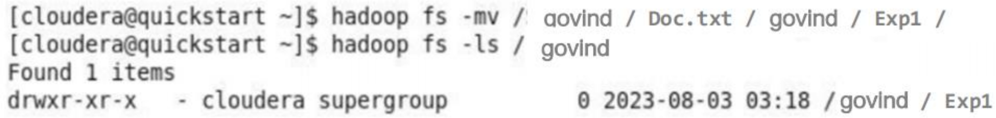
```
[cloudera@quickstart ~]$ hadoop fs -copyFromLocal /home/cloudera/Desktop/ Doc.txt

[cloudera@quickstart ~]$ hadoop fs -ls / govind / Doc.txt
Found 2 items
-rw-r--r--   1 cloudera supergroup         43 2023-08-03 03:05 / govind / Doc.txt
drwxr-xr-x   - cloudera supergroup          0 2023-08-03 02:51 / govind / Exp1
```

3. **Copy HDFS File to Local:**



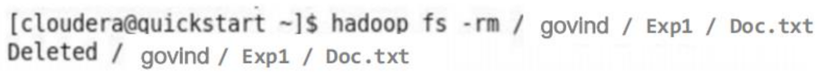
```
[cloudera@quickstart ~]$ hadoop fs -copyToLocal / govind/Doc.txt/home/Cloudera/Desktop/
copyToLocal: `/home/cloudera/Desktop/ABC.txt': File exists
```

4. **Move File from One Location to another in HDFS**



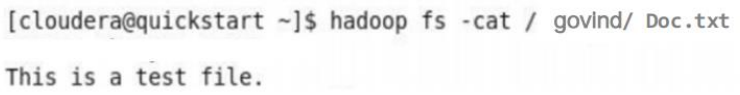
```
[cloudera@quickstart ~]$ hadoop fs -mv / govind / Doc.txt / govind / Exp1 /
[cloudera@quickstart ~]$ hadoop fs -ls / govind
Found 1 items
drwxr-xr-x   - cloudera supergroup          0 2023-08-03 03:18 / govind / Exp1
```

5. **Remove File from Specified Location**



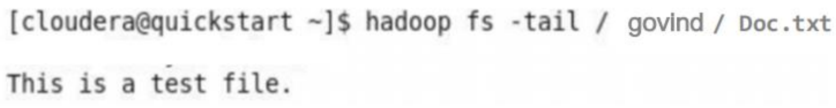
```
[cloudera@quickstart ~]$ hadoop fs -rm / govind / Exp1 / Doc.txt
Deleted / govind / Exp1 / Doc.txt
```

6. **Show Content of File Stored in HDFS**



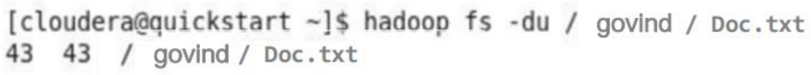
```
[cloudera@quickstart ~]$ hadoop fs -cat / govind/ Doc.txt

This is a test file.
```

7. **Show last 10 lines of HDFS File**



```
[cloudera@quickstart ~]$ hadoop fs -tail / govind / Doc.txt

This is a test file.
```

8. **Count the number of characters in file**



```
[cloudera@quickstart ~]$ hadoop fs -du / govind / Doc.txt
43   43   / govind / Doc.txt
```

Results and Discussions :

We learnt about Hadoop's important features and structure. We practiced basic Hadoop
commands, like making folders, moving files, and checking data. These actions showed us how
Hadoop can handle big data tasks effectively.