

INDUSTRIAL TRAINING
REPORT ON
HADOOP ADMINISTRATION AND ANALYSIS

A report submitted in partial fulfilment of the requirement for the award of the degree of

BACHELOR OF TECHNOLOGY
In
COMPUTER SCIENCE AND ENGINEERING
with specialization in
CLOUD COMPUTING AND VIRTUALIZATION



SUBMITTED BY
Vibhuti Maheshwari
170110048
Sap ID-1000009038

SUBMITTED TO
Mr Shanvendra Rai

Department of Computer Science & Engineering
DIT UNIVERSITY, DEHRADUN

(State Private University through State Legislature Act No. 10 of 2013 of Uttarakhand and approved by UGC)

Mussoorie Diversion Road, Dehradun, Uttarakhand - 248009, India.

2019

DECLARATION

I hereby certify that the work, which is being presented in the project report, entitled **Hadoop Administration and Analytics**, in partial fulfilment of the requirement for the award of the Degree of **Bachelor of Technology** and submitted to **DIT University, Dehradun** is an authentic record of my/our own work carried out during the period from **21 May 2019** to **30 June 2019** under the supervision of **Brillica Services**.

Date:

Signature of the Candidate

Signature of Internal Faculty Supervisor

ACKNOWLEDGEMENT

This summer training is of an immense academic record and value for the student of any professional course and for a B.Tech student who have to be in the industry with theoretical knowledge, this practical experience gives an extra confidence in his/her performance.

I would like to thank Brillica Services Pvt. Ltd. that gave me the honour to complete my summer training in the field of Big Data Analytics and Python. I would like to thank all my comrades & associates of Brillica Services Pvt. Ltd. who really helped me in understanding all the topics and activities of our daily sessions.

My heartiest thanks to my mentor Ms Garima Singh, who encouraged me to cope with the problems that I faced during the course of this project and to my friends in my batch for their help and cooperation with me during this project.

Lastly, I would like to thank all those who helped me in any way in my project.

(Vibhuti Maheshwari)

ABSTRACT

Big data Analytics refers to the techniques that can be used for converting raw data into meaningful information which helps in business analysis and forms a decision support system for the executives in the organization.

Big data is the voluminous and complex collection of data that cannot be processed using traditional tools.

In this project I analysed the Aadhar card data set against different queries for example total number of enrolments rejected by state by gender, names of various districts in each state and names of different enrolment agencies in each state.

The main objective of Aadhar card is to provide 12 digit unique numbers to each Indian resident so that common masses can be benefited by the various governmental policies such as direct benefit transfer, Aadhar enabled biometric attendance and other uses by central government agencies.

CERTIFICATE FROM COMPANY

Date of Issue : 01 July 2019
Certificate No.: BSPL/03/887

Certificate of

Participation

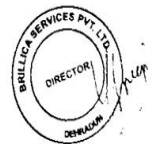


BRILICA
SERVICES

This Certificate accredits that

Vibhuti Maheshwari

*has successfully completed 1.5 Months Training
on Big Data Analytics Using Python & Hadoop*



Technology
Provider



AEP
Authorized Education
Partner



APPRAISAL FORM



PERIODIC APPRAISAL PERFORMANCE

To be filled by external Advisor

Name of Student: Vibhuti Maheshwari Registration No: BSPL/03/887 annexure -IX

Project: - Hadoop Administration and Analytics

Name of Organization & Address: Brillica Services PVT LTD, 1st Floor Lal pul near, Patel Nagar chowk, Dehradun

External Project: - Hadoop Administration and Analytics

Supervisor (with Phone No)- +918882140688

Period of evaluation From 21st May 2019 to 30 June 2019

S.no	Criteria	Marks Awarded (out of 10)
1.	Punctuality	10
2.	Regularity of work	10
3.	Progress in work since last appraisal	10
4.	Improvement in learning	9
5.	Grasp of Applications (s)	9
6.	Consultation and discussion	10
7	Self-motivation/Dedication/Initiative	10
8	Technical competency	10
9	Discipline & Sincerity	10
10	Problem Solving	10
	Grand Total	98

ANNEXURE- IX

General Remarks / Observations with regard to deficiencies / problems / suggestions for improvement

Enthusiastic & Hardworking

Brillica Services Pvt. Ltd.
New Delhi

Signature of External Project Supervisor/Guide (With Seal, Date & Designation)

12th July, 2019

INDEX

<u>CHAPTER</u>	<u>PAGE No.</u>
Chapter 1 – Organization Overview	
1.1 Introduction	1
1.2 Services Provided by the Company	2
Chapter 2 – Introduction to the Project	3
Chapter 3 – Pre-Requisites	
3.1 A Cluster	4
3.2 Non HA Cluster	5
3.3 My Non HA Cluster	7
3.4 HDFS	8
3.5 Apache Hive	9
Chapter 4 – Modules of the Project & Implementation	
4.1 Setting up a Non HA Cluster	10
4.2 Installing Hive	13
4.2.1 Apache Zookeeper	14
4.2.2 Apache Hive	15
4.2.3 Hive Metastore	16
4.2.4 Hive Server2	17
4.3 Uploading Data	
4.3.1 Uploading Aadhar Dataset into Hadoop	18
4.3.2 Performing Analysis	
Chapter 5 – Screenshots	
5.1 Final Cluster	19
5.2 Analysis(Results)	20
Chapter 6 – Conclusion	22
Chapter 7 – Future Scope	23
Bibliography	24
Abbreviations	25

LIST OF FIGURES

	<u>Figure Name</u>	<u>Page No.</u>
1.	Microsoft Authorised Partner	1
2.	IBM Authorised Partner	1
3.	Intel Technology Provider	1
4.	AWS Partner Network	1
5.	MAPR Training Partner	1
6.	A six node Non HA Cluster.	6
7.	My Non HA Cluster	7
8.	HDFS Architecture	8
9.	Apache Hive Logo	9
10.	Apache Zookeeper Architecture	14
11.	Hive Metastore	16

CHAPTER-1

ORGANIZATION OVERVIEW

1.1 INTRODUCTION

Brillica Services Pvt. Ltd. is the Best Technology Provider in India. They provide specialized courses in DATA SCIENCE, INTERNET OF THINGS, PYTHON, MACHINE LEARNING with PYTHON, JAVA, CISCO, ARTIFICIAL INTELLIGENCE, ANDROID APP DEVELOPMENT, MICROSOFT and many other technologies with Live Projects.

Here students don't just complete the course with a single project, they complete their course with a deep practical knowledge and multiple projects.

Brillica Services has successfully executed projects in African continent and is still working with them towards providing emerging technology solutions for various corporates. Brillica Services focuses on the following key features:

- Corporate Training
- Industrial Training
- Online Training
- Language Zone
- Business and Resource Consulting
- Research and Development
- Skills for Schools and Colleges

Brillica Services is proudly associated with MICROSOFT, INTEL and IBM.

As they are the only associated Business and Education partner with Microsoft, Intel & IBM in Uttarakhand, their every student get certification of MICROSOFT & IBM.



Brillica Services is the place where trainer and the students share a comfortable workplace which makes their efforts successful. They give every student and employee a fair time to share their views and thoughts so that they could make them doubt free.

1.2 SERVICES PROVIDED BY THE COMPANY

1. Data Science Training
 - a. Data Science Master
 - b. Machine Learning with Python
 - c. Artificial Intelligence Training
2. Data Analytics Training
 - a. Data Analytics Master
 - b. Big Data Hadoop Training
 - c. Tableau with R Training
 - d. R Programming
 - e. SPSS Data Statistical Training
 - f. Microsoft PowerBI Training
3. Cloud Computing
 - a. AWS Associate Training
 - b. AWS Solution Architect
4. Cisco
 - a. CCNA R&S Training
 - b. CCNP R&S Training
5. Project Management
 - a. CAPM
 - b. CCBA
 - c. PMP
 - d. CBAP

CHAPTER-2

INTRODUCTION TO THE PROJECT

The world's largest democracy, India, is the second largest nation in terms of population, with 1.3 billion population. Among these, 99% of adult population enrolled for Aadhar, the unique identity provided by the Government of India for diverse purposes. The government maintains the Aadhar related data in digital format [.https://data.uidai.gov.in/uiddatacatalog/dataCatalogHome.do](https://data.uidai.gov.in/uiddatacatalog/dataCatalogHome.do) website provides the access to Aadhar card related data set. The public could access some of the sources of these data and they can analyse to extract useful information and generate reports.

The data set covers more than 99% adult population of our nation. So the amount of data generated by Aadhar is very huge. Similarly, all the data collected for this unique identity is not in structured data. It also consists of unstructured and semi-structured data. Also, the enrolment is still in the process. The processing speed of this data generation is high. These characteristics of Aadhar Data make it fall under the concept of Big Data.

The term "Big Data" can be defined as data that becomes so large that it cannot be processed using conventional methods. Data that would take too much time and cost too much money to load into a relational database for analysis. In other words, data which is beyond storage capacity beyond processing power is called Big Data.

Apache Hadoop is a framework for running applications on a large cluster built of commodity hardware. The purpose of Hadoop is storing and processing large amount of data or big data. So this project uses Hadoop framework for processing Aadhar data.

In this project, I have first set up a cluster to work on Hadoop Ecosystem. Aadhar dataset was downloaded in csv format (presently, we are not allowed to download Aadhar data from any government site but previously we were given access to some of the Aadhar data which can still be downloaded from various other sites) and loaded into HDFS (Hadoop Distributed File Systems) which is designed to handle large amount of data. Further, Hive Queries were run on Apache Hive to extract information from the dataset loaded into HDFS.

CHAPTER-3

PRE-REQUISITES

3.1 A CLUSTER

Hadoop is actually the name of the software that runs on a cluster – namely, the HDFS, and the cluster resource manager, YARN, which are collectively composed of six types of background services running on a group of machines.

A set of machines that is running HDFS and YARN is known as a cluster, and the individual machines are called nodes. A cluster can have a single node, or many thousands of nodes, but all clusters scale horizontally, meaning as you add more nodes, the cluster increases in both capacity and performance in a linear fashion.

YARN and HDFS are implemented by several daemon processes – that is, software that runs in the background and does not require user input. Hadoop processes are services, meaning they run all the time on a cluster node and accept input and deliver output through the network.

Each node in the cluster is identified by the type of process or processes that it runs:

- **MASTER NODES**

These nodes run coordinating services for Hadoop workers and are usually the entry points for user access to the cluster. Without masters, coordination would fall apart, and distributed storage or computation would not be possible

- **WORKER NODES**

These nodes are the majority of the computers in the cluster. Worker nodes run services that accept tasks from master nodes—either to store or retrieve data or to run a particular application. A distributed computation is run by parallelizing the analysis across worker nodes

Clusters are of three types:-

- a. Pseudo Cluster – A single machine has all the daemons running in it. It is not present in real life situations and is just for learning purposes.
- b. Non High Availability Cluster - A multinode cluster which has a single point of failure. It is used for learning purposes as well as in real life situations.
- c. High Availability Cluster - A multinode cluster with no single point of failure. It is used in real life scenarios.

3.2 NON HA CLUSTER

A Non High Availability Cluster is a multinode cluster. It consists of one NameNode and a Secondary NameNode. It is a combination of both HDFS and YARN.

The nodes that work under HDFS and their roles are:-

- **NameNode (Master)**
Stores the directory tree of the file system, file metadata, and the locations of each file in the cluster. Clients wanting to access HDFS must first locate the appropriate storage nodes by requesting information from the NameNode.
- **Secondary NameNode (Master)**
Performs housekeeping tasks and checkpointing on behalf of the NameNode. Despite its name, it is not a backup NameNode.
- **DataNode (Worker)**
Stores and manages HDFS blocks on the local disk. Reports health and status of individual data stores back to the NameNode.

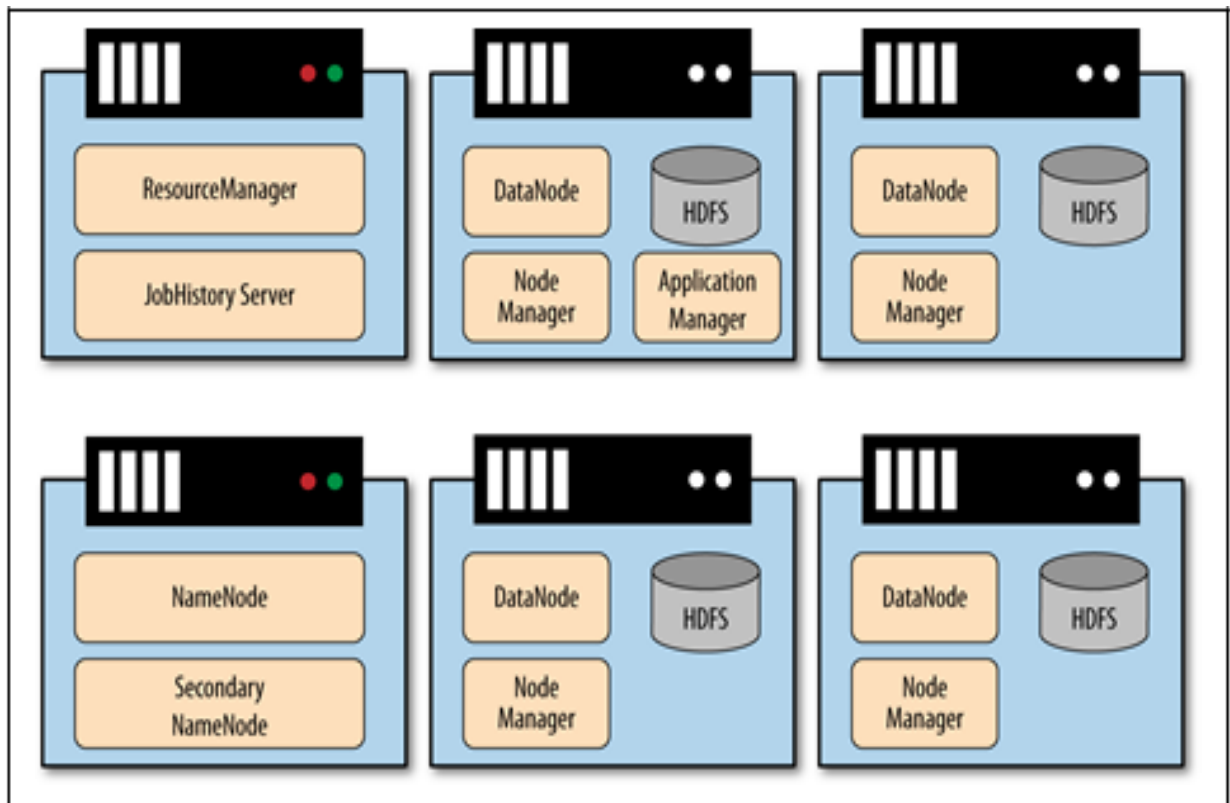
At a high level, when data is accessed from HDFS, a client application must first make a request to the NameNode to locate the data on disk. The NameNode will reply with a list of DataNodes that store the data, and the client must then directly request each block of data from the DataNode. Note that the NameNode does not store data, nor does it pass data from DataNode to client, instead acting like a traffic cop, pointing clients to the correct DataNodes.

The nodes that work under YARN are:-

- **ResourceManager (Master)**
Allocates and monitors available cluster resources (e.g., physical assets like memory and processor cores) to applications as well as handling scheduling of jobs on the cluster.
- **ApplicationMaster (Master)**
Coordinates a particular application being run on the cluster as scheduled by the ResourceManager.
- **NodeManager (Worker)**
Runs and manages processing tasks on an individual node as well as reports the health and status of tasks as they're running.

Similar to how HDFS works, clients that wish to execute a job must first request resources from the ResourceManager, which assigns an application-specific ApplicationMaster for the duration of the job. The ApplicationMaster tracks the execution of the job, while the ResourceManager tracks the status of the nodes, and each individual NodeManager creates containers and executes tasks within them. There may be other processes running on the

Hadoop cluster as well—for example, JobHistory servers or ZooKeeper coordinators, but the above given services are the primary software running in a Non High Availability cluster.



A six node Non High Availability Cluster.

3.3 MY NON HA CLUSTER

For creating a real life-like scenario, I installed used a virtualization software VMWare.

Four machines (nodes) namely

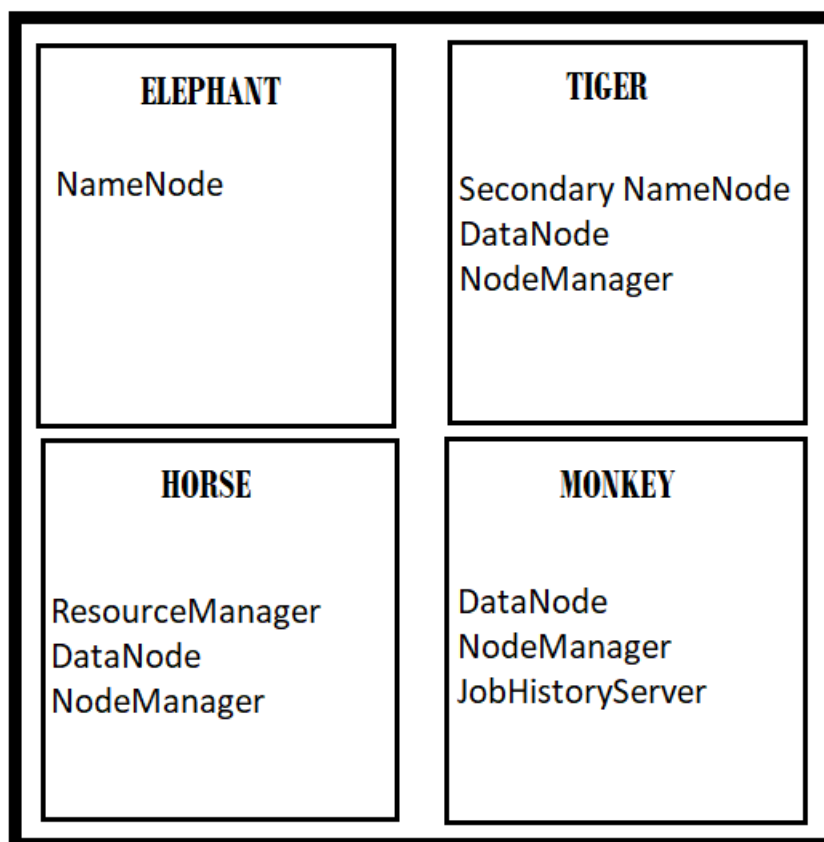
- ELEPHANT
- TIGER
- HORSE
- MONKEY

were provided to me by my trainer and mentor for learning and project building purpose.

They had CentOS, a distribution of Linux, as their operating system.

I set up my NonHA Cluster as below:-

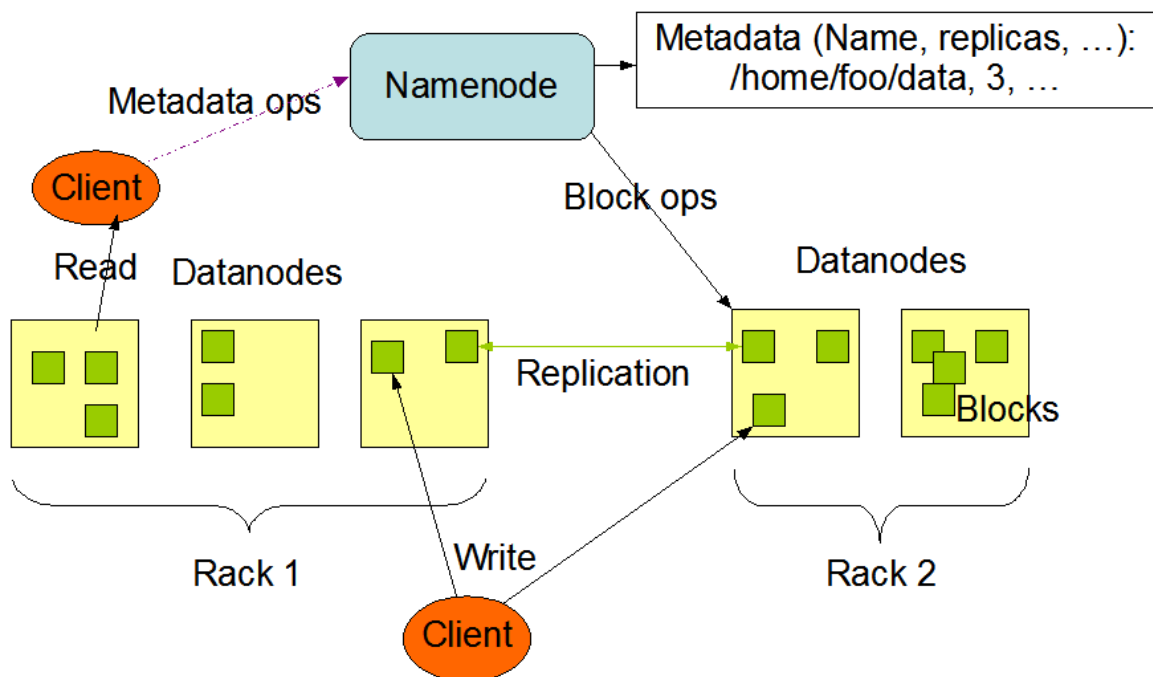
- NameNode on ELEPHANT
- Secondary NameNode on TIGER
- ResourceManager on HORSE
- MapReduce JobHistoryServer on MONKEY
- DataNode on all the nodes except ELEPHANT
- NodeManager on all the nodes except ELEPHANT



3.4 HDFS

The Hadoop Distributed File System (HDFS) is a distributed file system designed to run on commodity hardware. It has many similarities with existing distributed file systems. However, the differences from other distributed file systems are significant. HDFS is highly fault-tolerant and is designed to be deployed on low-cost hardware. HDFS provides high throughput access to application data and is suitable for applications that have large data sets. HDFS relaxes a few POSIX requirements to enable streaming access to file system data. HDFS was originally built as infrastructure for the Apache Nutch web search engine project. HDFS is part of the Apache Hadoop Core project.

HDFS Architecture



HDFS is designed to reliably store very large files across machines in a large cluster. It stores each file as a sequence of blocks. The blocks of a file are replicated for fault tolerance. The block size and replication factor are configurable per file. The NameNode makes all decisions regarding replication of blocks. It periodically receives a Heartbeat and a Blockreport from each of the DataNodes in the cluster. Receipt of a Heartbeat implies that the DataNode is functioning properly. A Blockreport contains a list of all blocks on a DataNode.

HDFS allows user data to be organized in the form of files and directories. It provides a command line interface called FS shell that lets a user interact with the data in HDFS. The syntax of this command set is similar to other shells (e.g. bash, csh) that users are already familiar with. FS shell is targeted for applications that need a scripting language to interact with the stored data.

3.5 Apache Hive

Hadoop's most popular SQL-based querying engine is Apache Hive while NoSQL database for Hadoop is HBase.

Apache Hive is a “data warehousing” framework built on top of Hadoop. Hive provides data analysts with a familiar SQL-based interface to Hadoop, which allows them to attach structured schemas to data in HDFS and access and analyse that data using SQL queries. Hive has made it possible for developers who are fluent in SQL to leverage the scalability and resilience of Hadoop without requiring them to learn Java or the native MapReduce API.

Hive provides its own dialect of SQL called the Hive Query Language, or HQL. HQL supports many commonly used SQL statements including DDL's and DML's.

Hive provides the high-scalability and high-throughput that you would expect from any Hadoop-based application, and as a result, is very well suited to batch-level workloads for online analytical processing (OLAP) of very large datasets at the terabyte and petabyte scale.

Apache Hive



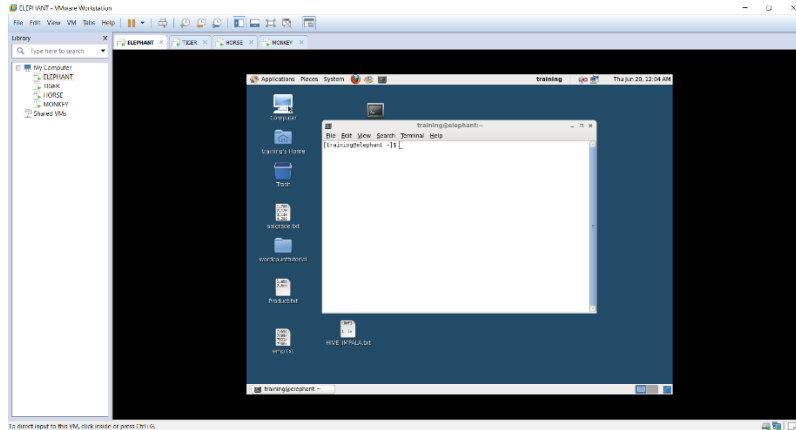
CHAPTER-4

4.1 SETTING UP A NON HA CLUSTER (ADMINISTRATION)

A computer cluster is a set of loosely or tightly connected computers that work together so that, in many respects, they can be viewed as a single system. Non High Availability or Non HA Clusters consists of a single NameNode, a master server that manages the file system namespace and regulates access to files by clients. In addition, there are a number of DataNodes, usually one per node in the cluster, which manage storage attached to the nodes that they run on.

I am here forth providing a stepwise informative description of how to install a Non High Availability cluster on a four node cluster.

1. Power ON all the four machines (ELEPHAT, TIGER, HORSE, MONKEY).



2. Rest cluster to pseudo cluster.
3. Stop all services on ELEPHANT one after another:
 - NameNode
 - Secondary NameNode
 - DataNode
 - ResourceManager
 - NodeManager
 - JobHistoryServer
4. Remove all log files created on ELEPHANT.
5. Remove Secondary NameNode, JobHistoryServer and ResourceManager on ELEPHANT.
6. Install Secondary NameNode on TIGER machine.
7. Install DataNode on TIGER, HORSE and MONKEY machines each.
8. Install ResourceManager on HORSE machine.
9. Install NodeManager on TIGER, HORSE and MONKEY machines.
10. Install MapReduce on TIGER, HORSE and MONKEY machines.
11. Install JobHistoryServer on MONKEY machine.

12. Modifying Hadoop configuration files.

(Just for reference)

We will make changes in four configuration files and hadoop-env.sh file.

- core-site.xml,
- hdfs-site.xml,
- yarn-site.xml,
- mapred-site.xml

We will make changes on ELEPHANT and copy them in all other nodes (TIGER, HORSE and MONKEY).

13. Changes in core-site.xml file.

```
-->
<?xml-stylesheet type="text/xsl" href="configuration.xsl">

<configuration>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://elephant:8020</value>
  </property>
</configuration>
~
```

14. Changes in hdfs-site.xml file.

```
-->
<?xml-stylesheet type="text/xsl" href="configuration.xsl">

<configuration>
  <property>
    <name>dfs.name.dir</name>
    <value>/var/lib/hadoop-hdfs/cache/hdfs/dfs/name</value>
  </property>
  <property>
    <name>dfs.namenode.name.dir</name>
    <value>file:///disk1/dfs/nn,file:///disk2/dfs/nn</value>
  </property>
  <property>
    <name>dfs.datanode.data.dir</name>
    <value>file:///disk1/dfs/dn,file:///disk2/dfs/dn</value>
  </property>
</configuration>
~
```

15. Changes in yarn-site.xml file.

```
  <property>
    <name>yarn.nodemanager.log-dirs</name>
    <value>/var/log/hadoop-yarn/containers</value>
  </property>

  <property>
    <name>yarn.nodemanager.remote-app-log-dir</name>

<value>/var/log/hadoop-yarn/apps</value>
  </property>

  <property>
    <name>yarn.log-aggregation-enable</name>
    <value>true</value>
  </property>
</configuration>
-- INSERT --
```

15. Changes in mapred-site.xml file.

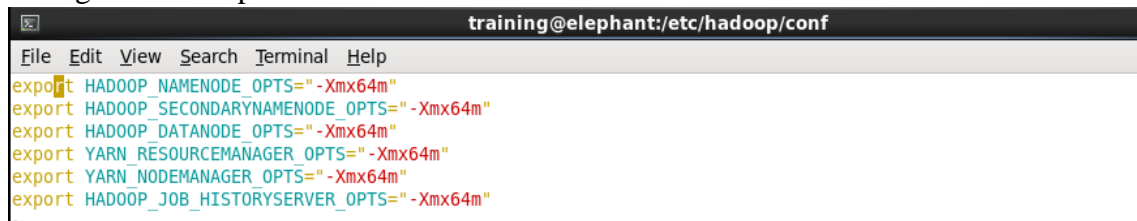
```
-->
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

<configuration>
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>
  <property>
    <name>mapreduce.jobhistory.address</name>
    <value>monkey:10020</value>
  </property>
  <property>
    <name>mapreduce.jobhistory.webapp.address</name>
    <value>monkey:19888</value>
  </property>
  <property>
    <name>yarn.app.mapreduce.am.staging-dir</name>
    <value>/user</value>
  </property>
</configuration>
-- INSERT --
```

38,17

All

16. Changes in hadoop-env.sh file.



```
File Edit View Search Terminal Help
export HADOOP_NAMENODE_OPTS="-Xmx64m"
export HADOOP_SECONDARYNAMENODE_OPTS="-Xmx64m"
export HADOOP_DATANODE_OPTS="-Xmx64m"
export YARN_RESOURCEMANAGER_OPTS="-Xmx64m"
export YARN_NODEMANAGER_OPTS="-Xmx64m"
export HADOOP_JOB_HISTORYSERVER_OPTS="-Xmx64m"
```

17. From terminal on ELEPHANT , copying all configuration files on other nodes as well

18. Giving path to nodes as written in hdfs-site.xml :

19. From terminal on Elephant:

20. Format HDFS (From terminal on Elephant machine)

21. From Elephant machine, open browser and type:

<http://elephant:50070>

22. On TIGER machine Start the Secondary NameNode.

23. Start DATANODE on all 4 machines

24. Create directories for YARN , MapReduce HDFS(On any host in your cluster (Here we will be doing on HORSE))

25. Start ResourceManager daemon on HORSE.

26. From Horse machine open browser and type:

<http://horse:8088>

27. Start NodeManagers on all the hosts (machines)

28. Start JobHistoryServer on MONKEY machine.

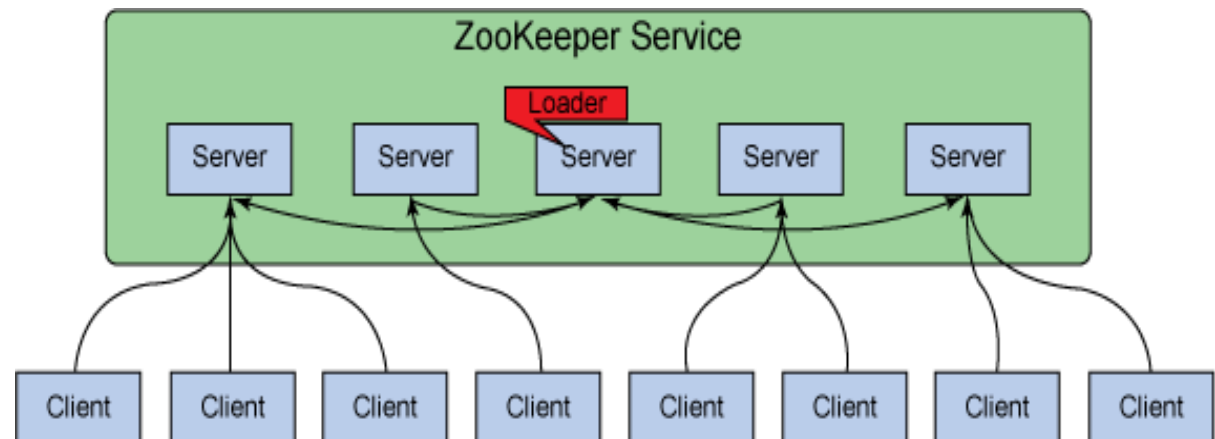
4.2 INSTALLING HIVE

Apache Hive is a [data warehouse](#) software project built on top of [Apache Hadoop](#) for providing data query and analysis. Hive gives a [SQL-like interface](#) to query data stored in various databases and file systems that integrate with Hadoop. While initially developed by Facebook, Apache Hive is used and developed by other companies such as Netflix and the Financial Industry Regulatory Authority (FINRA). Amazon maintains a software fork of Apache Hive included in Amazon Elastic MapReduce on Amazon Web Services.

For installation of Apache Hive, we need to install the following services:-

- Apache Zookeeper - Apache ZooKeeper is an effort to develop and maintain an open-source server which enables highly reliable distributed coordination. ZooKeeper is a centralized service for maintaining configuration information, naming, providing distributed synchronization, and providing group services. All of these kinds of services are used in some form or another by distributed applications.
- Apache Hive
- Hive Metastore - Metastore is the central repository of Apache Hive metadata. It stores metadata for Hive tables (like their schema and location) and partitions in a relational database. It provides client access to this information by using metastore service API.
- Hive Server 2 - HiveServer2 (HS2) is a service that enables clients to execute queries against Hive. HiveServer2 is the successor to HiveServer1 which has been deprecated. HS2 supports multi-client concurrency and authentication. It is designed to provide better support for open API clients like JDBC and ODBC.

4.2.1. APACHE ZOOKEEPER

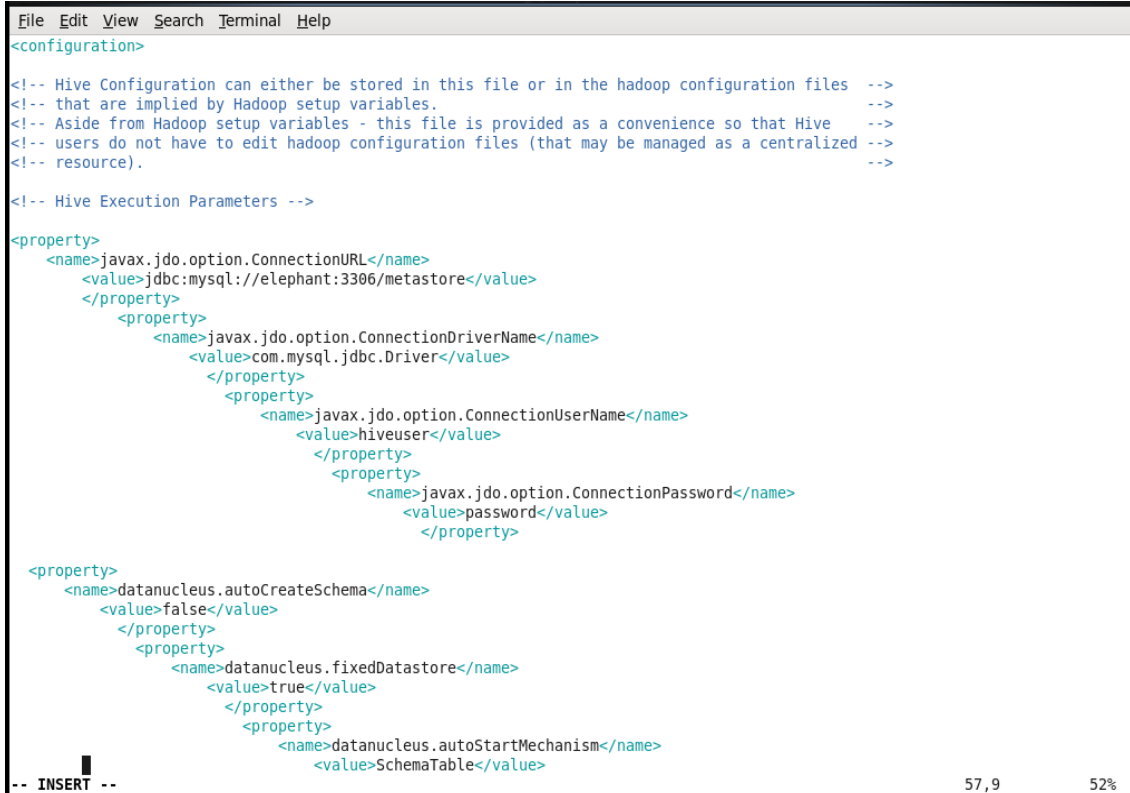


1. Install Zookeeper on ELEPHANT, TIGER AND HORSE
2. Initialize the three ZooKeeper servers with a unique ID.
3. Using sudo, edit the ZooKeeper configuration file at the path /etc/zookeeper/conf/zoo.cfg on elephant.
4. Append the following three lines to the end of the file:
5. Using sudo, create a configuration file for Java options at the path /etc/zookeeper/conf/java.env on elephant.
6. The file should have a single line, as follows
export JVMFLAGS="-Xmx64m"
7. Copy configuration files from ELEPHANT machine.
8. Start the Zookeeper servers.
9. Check for Zookeeper processor.

```
[training@horse ~]$ sudo jps
2202 NodeManager
3541 QuorumPeerMain
2134 DataNode
2326 ResourceManager
3590 Jps
[training@horse ~]$
```

4.2.2. APACHE HIVE

1. Install Hive
2. Making changes to hive-site.xml :



```
<configuration>

<!-- Hive Configuration can either be stored in this file or in the hadoop configuration files -->
<!-- that are implied by Hadoop setup variables. -->
<!-- Aside from Hadoop setup variables - this file is provided as a convenience so that Hive -->
<!-- users do not have to edit hadoop configuration files (that may be managed as a centralized -->
<!-- resource). -->

<!-- Hive Execution Parameters -->

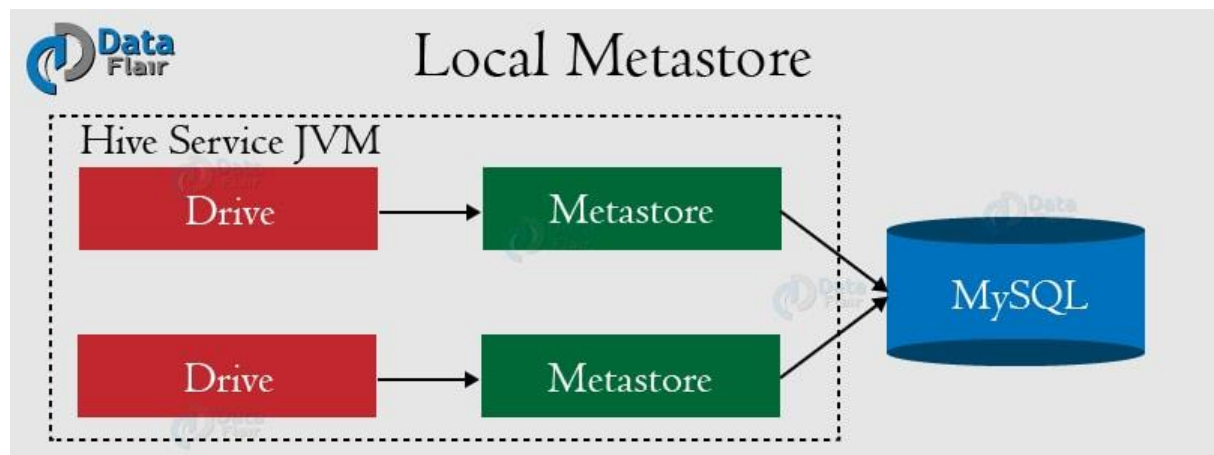
<property>
  <name>javax.jdo.option.ConnectionURL</name>
  <value>jdbc:mysql://elephant:3306/metastore</value>
</property>
  <property>
    <name>javax.jdo.option.ConnectionDriverName</name>
    <value>com.mysql.jdbc.Driver</value>
  </property>
    <property>
      <name>javax.jdo.option.ConnectionUserName</name>
      <value>hiveuser</value>
    </property>
      <property>
        <name>javax.jdo.option.ConnectionPassword</name>
        <value>password</value>
      </property>

  <property>
    <name>datanucleus.autoCreateSchema</name>
    <value>>false</value>
  </property>
    <property>
      <name>datanucleus.fixedDatastore</name>
      <value>true</value>
    </property>
      <property>
        <name>datanucleus.autoStartMechanism</name>
        <value>SchemaTable</value>
      </property>

-- INSERT --
```

3. Create warehouse directory and give it access permissions.
4. Create a symbolic link between /usr/share/java/mysql-connector-java.jar and /usr/lib/hive/lib
5. Enter the SQL command line from terminal.
6. Create a user, set its password and grant(select, insert, update, delete, lock tables, execute, create, alter) permissions and revoke all privileges.
7. Initialise the above schema into Hive.
8. Revoke permissions (alter & create) from Hive database.

4.2.3. HIVE METASTORE



1. Install metastore.
2. Start Hive Metastore.
3. Check for Hive Metastore daemon.

```
File Edit View Search Terminal Help
[training@elephant ~]$ sudo jps
3406 Jps
2420 NodeManager
2322 NameNode
2533 RunJar
2159 QuorumPeerMain
2228 DataNode
2690 RunJar
[training@elephant ~]$
```


4.2.4. HIVE SERVER2

1. Install HiveServer2
2. Edit hiveserver2 file.
3. Add below line and save and exit.

```
# Licensed to the Apache Software Foundation (ASF) under one or more
# contributor license agreements.  See the NOTICE file distributed with
# this work for additional information regarding copyright ownership.
# The ASF licenses this file to You under the Apache License, Version 2.0
# (the "License"); you may not use this file except in compliance with
# the License.  You may obtain a copy of the License at
#
#     http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.

# The port for Hive server2 daemon to listen to.
# Unfortunately, there is no way to specify the interfaces
# to which the daemon binds.
#
#PORT=
export HADOOP_MAPRED_HOME=/usr/lib/hadoop-mapreduce
```

4. Start HiveServer2.
5. Enter beeline using the following command
beeline -u jdbc:hive2://elephant:10000 -n training

6. Check for existing databases.

```
0: jdbc:hive2://elephant:10000> show databases;
+-----+
| database_name |
+-----+
| default       |
+-----+
1 row selected (1.981 seconds)
0: jdbc:hive2://elephant:10000>
```

7. Check for existing tables.

4.3 UPLOADING DATA

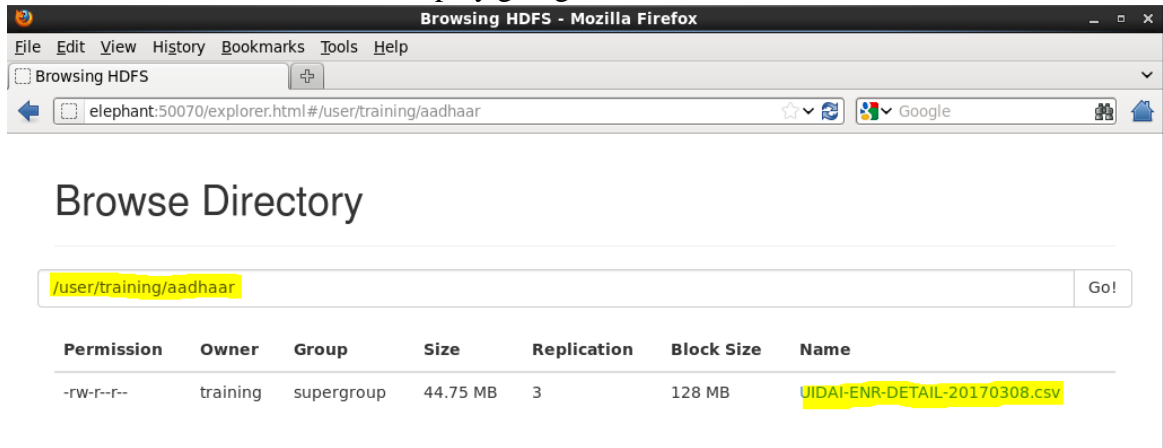
Data analysis is a process of inspecting, cleansing, transforming and modelling data with the goal of discovering useful information, informing conclusions and supporting decision-making.

For performing analysis on a dataset we need to do two things-

- Upload dataset into Hadoop (HDFS).
- Perform analysis on it using Hive,Pig or Impala.

4.3.1 UPLOADING AADHAR DATASET INTO HADOOP

1. Download Aadhar database into local system (preferably in csv format).
2. Make a new directory on Hadoop and put this file into that directory.
3. Check existence of file on Hadoop by going to the browser.



4.3.2. PERFORMING ANALYSIS

1. Enter beeline shell.
2. Create an external table using the csv file uploaded on Hadoop.
3. Check the description of your table.
4. Enter HiveQL queries according to your needs.

CHAPTER-5 SCREENSHOTS

5.1 FINAL CLUSTER

- Elephant

```
[training@elephant conf]$ sudo jps
10125 Jps
9871 DataNode
9732 NameNode
10000 NodeManager
2063 -- process information unavailable
[training@elephant conf]$ █
```

- Tiger

```
[training@tiger ~]$ sudo jps
10149 NodeManager
10004 DataNode
2044 -- process information unavailable
9882 SecondaryNameNode
10286 Jps
[training@tiger ~]$ █
```

- Horse

```
File Edit View Search Terminal Help
[training@horse ~]$ sudo jps
2338 ResourceManager
3223 Jps
2209 NodeManager
2142 DataNode
[training@horse ~]$ █
```

- Monkey

```
[training@monkey ~]$ sudo jps
10024 DataNode
10320 JobHistoryServer
10419 Jps
10171 NodeManager
[training@monkey ~]$ █
```

5.2 ANALYSIS (RESULTS)

1. Find the names of district in each state.

```
0: jdbc:hive2://elephant:10000/default> select distinct state,district from aadhar;
```

state	district
Andaman and Nicobar Islands	North And Middle Andaman
Andaman and Nicobar Islands	South Andaman
Andhra Pradesh	Ananthapuramu
Andhra Pradesh	Chittoor
Andhra Pradesh	Cuddapah
Andhra Pradesh	East Godavari
Andhra Pradesh	Guntur
Andhra Pradesh	Krishna
Andhra Pradesh	Kurnool
Andhra Pradesh	Nellore
Andhra Pradesh	Prakasam
Andhra Pradesh	Srikakulam
Andhra Pradesh	Visakhapatnam
Andhra Pradesh	Vizianagaram
Andhra Pradesh	West Godavari
Arunachal Pradesh	Anjaw
Arunachal Pradesh	Changlang
Arunachal Pradesh	East Kameng
Arunachal Pradesh	East Siang
Arunachal Pradesh	Kurung Kumey
Arunachal Pradesh	Lohit
Arunachal Pradesh	Longding
Arunachal Pradesh	Lower Subansiri
Arunachal Pradesh	Namsai
Arunachal Pradesh	Papum Pare
Arunachal Pradesh	Siang
Arunachal Pradesh	Tawang
Arunachal Pradesh	Tirap
Arunachal Pradesh	Upper Siang
Arunachal Pradesh	Upper Subansiri
Arunachal Pradesh	West Kameng
Arunachal Pradesh	West Siang
Assam	Baksa
Assam	Barpeta
Assam	Bongaigaon

Uttar Pradesh	Siddharthnagar
Uttar Pradesh	Sitapur
Uttar Pradesh	Sonbhadra
Uttar Pradesh	Sultanpur
Uttar Pradesh	Unnao
Uttar Pradesh	Varanasi
Uttarakhand	Almora
Uttarakhand	Bageshwar
Uttarakhand	Chamoli
Uttarakhand	Champawat
Uttarakhand	Dehradun
Uttarakhand	Haridwar
Uttarakhand	Nainital
Uttarakhand	Pauri Garhwal
Uttarakhand	Pithoragarh
Uttarakhand	Rudraprayag
Uttarakhand	Tehri Garhwal
Uttarakhand	Udham Singh Nagar
Uttarakhand	Uttarkashi
West Bengal	Bankura
West Bengal	Bardhaman
West Bengal	Birbhum
West Bengal	Cooch Behar
West Bengal	Dakshin Dinajpur
West Bengal	Darjeeling
West Bengal	Hooghly
West Bengal	Howrah
West Bengal	Jalpaiguri
West Bengal	Kolkata
West Bengal	Malda
West Bengal	Murshidabad
West Bengal	Nadia
West Bengal	North 24 Parganas
West Bengal	Paschim Medinipur
West Bengal	Purba Medinipur
West Bengal	Puruliya
West Bengal	South 24 Parganas
West Bengal	Uttar Dinajpur

2. How many number of enrolments of each gender were rejected?

```
0: jdbc:hive2://elephant:10000/default> select gender,count(enrolmentrejected) from aadhar group by gender;
```

```
+-----+-----+
| gender | _c1 |
+-----+-----+
| F      | 148013 |
| M      | 292798 |
| T      | 7      |
+-----+-----+
3 rows selected (58.785 seconds)
```

3. List the names of enrolment agencies in each state.

```
training@elephant:~
File Edit View Search Terminal Help
0: jdbc:hive2://elephant:10000/default> select distinct state,enrolmentagency from aadhar;
```

state	enrolmentagency
Andaman and Nicobar Islands	AKSH OPTIFIBRE LIMITED
Andaman and Nicobar Islands	Computer LAB
Andaman and Nicobar Islands	PROTEX COMPUTER PVT LTD
Andaman and Nicobar Islands	VEETECHNOLOGIES PVT. LTD
Andaman and Nicobar Islands	Wipro Ltd
Andhra Pradesh	77 Infosystems Pvt Ltd
Andhra Pradesh	AISECT Limited
Andhra Pradesh	AKSH OPTIFIBRE LIMITED
Andhra Pradesh	APOnline Limited
Andhra Pradesh	AVVAS INFOTECH PVT LTD
Andhra Pradesh	Abha Systems And Consultancy
Andhra Pradesh	Akshaya
Andhra Pradesh	Alankit Finsec Ltd
Andhra Pradesh	BNR UDYOG LIMITED
Andhra Pradesh	CHIPS
Andhra Pradesh	CMS Computers Ltd
Andhra Pradesh	CSC SPV
Andhra Pradesh	CSC e-Governance Services India Limited
Andhra Pradesh	Centre for e-Governance GOK
Andhra Pradesh	City Hawks Manpower Services & Consultancy
Andhra Pradesh	Computer LAB
Andhra Pradesh	Directorate of ESD
Andhra Pradesh	Directorate of Public Health and Family Welfare Govt of Andhra Pradesh
Andhra Pradesh	Directorate of Women & Child Department Govt Of Goa
Andhra Pradesh	Electronics Corporation of Tamil Nadu Limited
Andhra Pradesh	Frontech Systems Pvt Ltd
Andhra Pradesh	Karvy Data Management Services
Andhra Pradesh	Lankipalli Integrated Services Private Limited
Andhra Pradesh	Layman Education Society
Andhra Pradesh	Munish Kumar Bansal Contractor
Andhra Pradesh	NPS Technologies Pvt. Ltd
Andhra Pradesh	NetLink software Pvt Ltd
Andhra Pradesh	Nielsen India Private Limited
Andhra Pradesh	Ojus Healthcare Private Limited
Andhra Pradesh	RELIGARE SECURITIES LTD
Delhi	Virinchi Technologies Ltd
Delhi	Wipro Ltd
Delhi	Zephyr System Pvt.Ltd.
Goa	AKSH OPTIFIBRE LIMITED
Goa	Alankit Limited
Goa	Amar Constructions
Goa	CSC SPV
Goa	Centre for e-Governance GOK
Goa	Directorate of Women & Child Department Govt Of Goa
Goa	IPS e Services Pvt Ltd
Goa	M/s Gold Square Builders & Promoters Pvt. Ltd.
Goa	M/s. Goa Electronics Ltd
Goa	NPS Technologies Pvt. Ltd
Goa	Om Softwares
Goa	Omnitech Infosolutions Ltd
Goa	RELIGARE SECURITIES LTD
Goa	Rajcomp Info Services Ltd
Goa	Steel City Securities Limited
Goa	Vakrangee Softwares Limited
Gujarat	A I Soc for Electronics and Comp Tech
Gujarat	A-Onerealtors Pvt Ltd
Gujarat	AKSH OPTIFIBRE LIMITED
Gujarat	AVVAS INFOTECH PVT LTD
Gujarat	Abha Systems And Consultancy
Gujarat	Administration of DNH
Gujarat	Alankit Limited
Gujarat	BASIX
Gujarat	BHAVANAGAR MC
Gujarat	CALANCE SOFTWARE PRIVATE LTD
Gujarat	CMS Computers Ltd
Gujarat	CSC SPV
Gujarat	CSC e-Governance Services India Limited
Gujarat	Compro Systems & Services
Gujarat	Computer LAB
Gujarat	Conatus Infocom Pvt. Ltd
Gujarat	DEVASHISH SECURITIES PVT. LTD.
Gujarat	Dist E-seva Society Anand
Gujarat	Dist. E-seva Society Morbi
Gujarat	District E-Seva Society Gandhinagar

CHAPTER-6

CONCLUSION

1. Big Data refers to extremely large data sets that may be analysed computationally to reveal patterns, trends, and associations, especially relating to human behaviour and interactions.
2. 5 V's of Big Data:-
 - Volume – The size of data
 - Velocity – The speed at which data is generated
 - Variety – The different types of data
 - Veracity – The trustworthiness of data in terms of accuracy
 - Value – Just having Big Data is of no use unless we can turn it into a useful value
3. 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.
4. The base Apache Hadoop framework is composed of the following modules:
 - Hadoop Common – contains libraries and utilities needed by other Hadoop modules.
 - Hadoop Distributed File System (HDFS) – a distributed file-system that stores data on commodity machines, providing very high aggregate bandwidth across the cluster.
 - Hadoop YARN – a platform responsible for managing computing resources in clusters and using them for scheduling users' applications.
 - Hadoop MapReduce – an implementation of the MapReduce programming model for large-scale data processing.
5. Hadoop splits files into large blocks and distributes them across nodes in a cluster. It then transfers packaged code into nodes to process the data in parallel. This approach takes advantage of data locality where nodes manipulate the data they have access to. This allows the dataset to be processed faster and more efficiently than it would be in a more conventional supercomputer architecture that relies on a parallel file system where computation and data are distributed via high-speed networking.
6. Apache Pig, Apache Hive and Apache Impala are used for performing analysis on unstructured, semi-structured and structured data on Hadoop framework.

CHAPTER-7

FUTURE SCOPE

Hadoop is among the major big data technologies and has a vast scope in the future. Being cost-effective, scalable and reliable, most of the world's biggest organizations are employing Hadoop technology to deal with their massive data for research and production.

It includes storing data on a cluster without any machine or hardware failure, adding a new hardware to the nodes etc.

The scope of Hadoop in the future can be traced out by the fact that the availability of tons of data through social networking and other means has been increased and goes on increasing as the world approaches digitalization. This generation of massive data brings into use the Hadoop technology which is highly adopted as compared to other big data technologies. However, there are some other technologies competing with Hadoop as it has not yet gained stability in the big data market. It is still in the adoption phase and will take some time to get stable and lead the big data market.

As the size of data increases, the demand for Hadoop technology will rise. There will be need of more Hadoop developers to deal with the big data challenges.

IT professionals having Hadoop skills will be benefited with increased salary packages and an accelerated career growth.

BIBLIOGRAPHY

- https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html
- <https://raw.githubusercontent.com/stdatalabs/aadhaar-dataset-analysis/master/data/UIDAI-ENR-DETAIL-20170308.csv>
- https://www.google.com/search?ei=zaBvXbmGJcyGvQTv3IOoDw&q=ln+-s+syntax+unix&oq=sudo+ln+command&gs_l=psy-ab.1.0.0i7118.0.0..7547...0.2..0.0.0.....0.....gws-wiz.xoX7lRB7QZU
- Benjamin Bengfort & Jenny Kim, “Data Analytics with Hadoop”, O’Reilly Media, Inc., June 2016
- Vignesh Prajapati, “Big Data Analytics with R and Hadoop”, Packt Publishing Ltd, November 2013

ABBREVIATIONS

- csv – comma separated values
- HA – High Availability
- HDFS – Hadoop Distributed File System
- YARN – Yet Another Resource Negotiator
- API – Application Programming Interface
- SQL – Structured Query Language
- DDL - Data Definition Language
- DML – Data Manipulation Language
- HQL- Hive Query Language