

## **HADOOP ASSIGNMENT II**

**Name - Alvyn Abranches**

**Roll No - 1**

## PIG

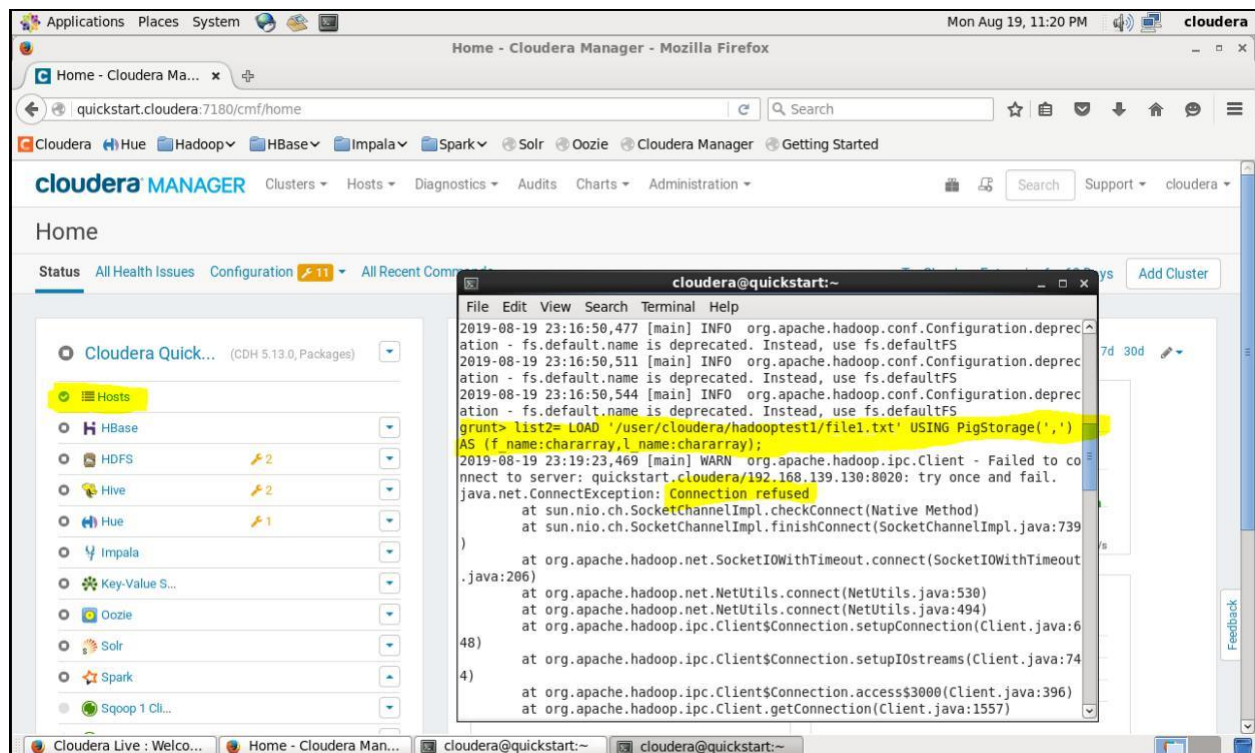
Pig is high-level programming language useful for analyzing large dataset

Pig has a nested relational model. In Pig programs are written in

language pig latin Uses Extract Transform,Load(ETL)

Using pig Latin we can perform MapReduce task easily without writing complex codes in java. Pig Latin is SQL-like query language

➔ When all services are stop



➔ When HDFS is start

The screenshot displays the Cloudera Manager web interface in a Mozilla Firefox browser. The browser's address bar shows the URL `quickstart.cloudera:7180/cm/`. The Cloudera Manager interface includes a top navigation bar with links for Applications, Places, System, and a user profile. Below this is a secondary navigation bar with links for Clusters, Hosts, Diagnostics, Audits, Charts, and Administration. The main content area is titled "Home" and features a "Status" section with a "Configuration" tab. In this tab, a list of services is shown, including HBase, HDFS, Hive, Hue, Impala, Key-Value S..., Oozie, Solr, Spark, and Sqoop 1 Cli... The HDFS service is highlighted with a yellow box and shows a status of "2".

Overlaid on the right side of the Cloudera Manager interface is a terminal window titled `cloudera@quickstart:~`. The terminal displays a series of log messages from the `org.apache.pig.backend.hadoop.executionengine.HExecutionEngine` class, indicating the connection to the Hadoop file system at `hdfs://quickstart.cloudera:8020`. The logs show multiple deprecation warnings for `fs.default.name`, suggesting it should be replaced with `fs.defaultFS`. At the bottom of the terminal, a Pig script is being executed:

```
grunt> list= LOAD '/user/cloudera/hadooptest1/file1.txt' USING PigStorage(',') AS (f_name:chararray,l_name:chararray);
grunt>
```

⇒ Therefore , Pig depends on HDFS.

## **HIVE**

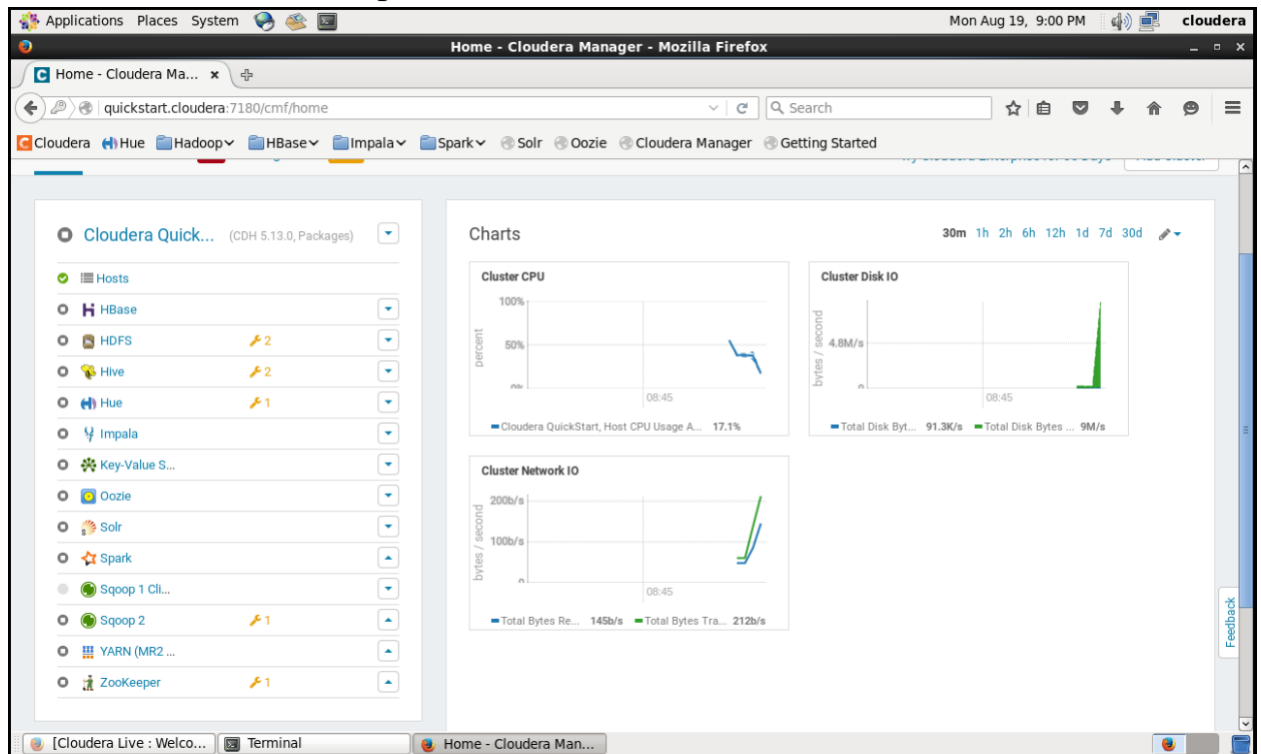
Hive is a data warehouse software built on top of Hadoop for providing query and analysis.

Hive provides SQL-like queries (HiveQL) which are implicitly converted into MapReduce or Spark jobs.

Built-in user-defined functions to manipulate dates, strings, and others.

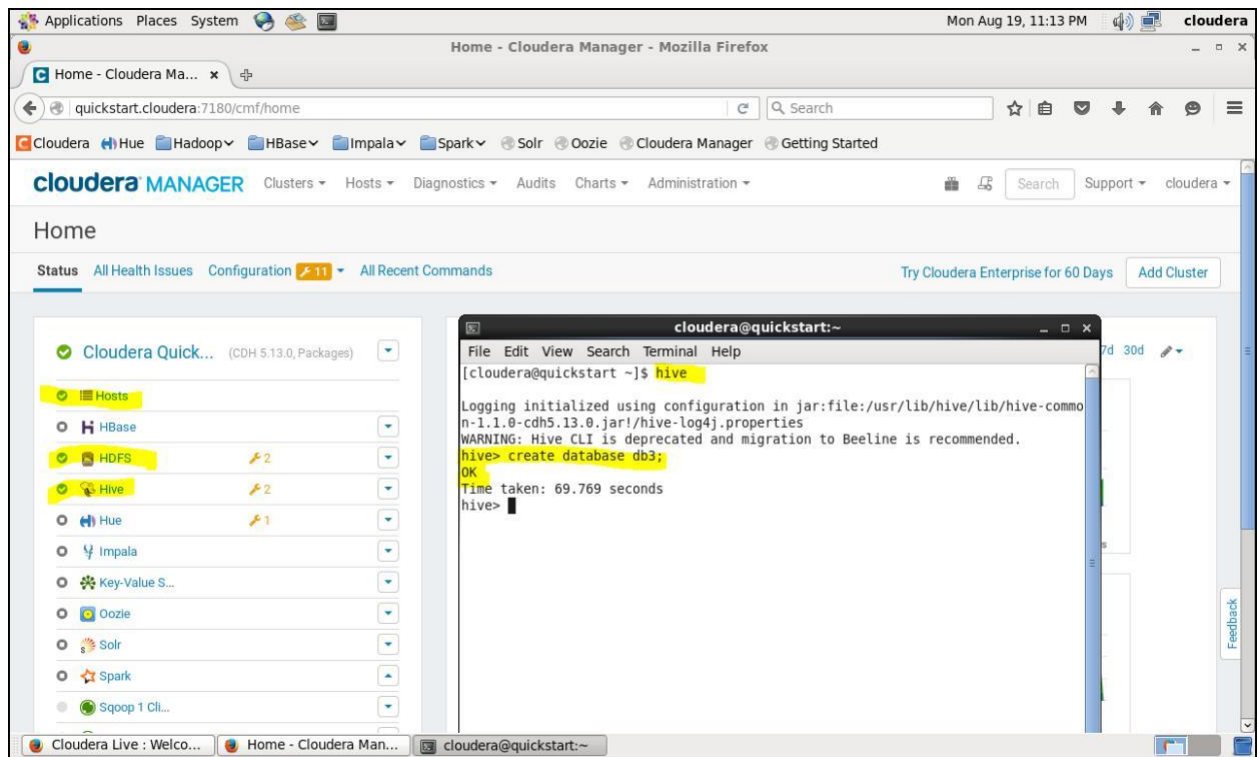
Metadata storage in a relational database management system, significantly reducing the time to perform semantic checks during query execution.

➔ When all services are stop



```
cloudera@quickstart:~$ hive
Logging initialized using configuration in jar:file:/usr/lib/hive/lib/hive-common-1.1.0-cdh5.13.0.jar!/hive-log4j.properties
Exception in thread "main" java.lang.RuntimeException: java.net.ConnectException: Call From quickstart.cloudera/192.168.139.130 to quickstart.cloudera:8020 failed on connection exception: java.net.ConnectException: Connection refused; For more details see: http://wiki.apache.org/hadoop/ConnectionRefused
    at org.apache.hadoop.hive.ql.session.SessionState.start(SessionState.java:571)
    at org.apache.hadoop.hive.cli.CliDriver.run(CliDriver.java:695)
    at org.apache.hadoop.hive.cli.CliDriver.main(CliDriver.java:634)
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
    at java.lang.reflect.Method.invoke(Method.java:606)
    at org.apache.hadoop.util.RunJar.run(RunJar.java:221)
    at org.apache.hadoop.util.RunJar.main(RunJar.java:136)
Caused by: java.net.ConnectException: Call From quickstart.cloudera/192.168.139.130 to quickstart.cloudera:8020 failed on connection exception: java.net.ConnectException: Connection refused; For more details see: http://wiki.apache.org/hadoop/ConnectionRefused
    at sun.reflect.NativeConstructorAccessorImpl.newInstance0(Native Method)
    at sun.reflect.NativeConstructorAccessorImpl.newInstance(NativeConstructorAccessorImpl.java:57)
    at sun.reflect.DelegatingConstructorAccessorImpl.newInstance(DelegatingConstructorAccessorImpl.java:45)
    at java.lang.reflect.Constructor.newInstance(Constructor.java:526)
    at org.apache.hadoop.net.NetUtils.wrapWithMessage(NetUtils.java:791)
    at org.apache.hadoop.ipc.Client.call(Client.java:1508)
    at org.apache.hadoop.ipc.Client.call(Client.java:1441)
    at org.apache.hadoop.ipc.ProtobufRpcEngine$Invoker.invoke(ProtobufRpcEngine.java:230)
    at com.sun.proxy.$Proxy17.getFileInfo(Unknown Source)
    at org.apache.hadoop.hdfs.protocolPB.ClientNamenodeProtocolTranslatorPB.getFileInfo(ClientNamenodeProtocolTranslatorPB.java:786)
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
    at java.lang.reflect.Method.invoke(Method.java:606)
    at org.apache.hadoop.io.retry.RetryInvocationHandler.invokeMethod(RetryInvocationHandler.java:260)
    at org.apache.hadoop.io.retry.RetryInvocationHandler.invoke(RetryInvocationHandler.java:104)
    at com.sun.proxy.$Proxy18.getFileInfo(Unknown Source)
    at org.apache.hadoop.hdfs.DFSClient.getFileInfo(DFSClient.java:2131)
    at org.apache.hadoop.hdfs.DistributedFileSystem$20.doCall(DistributedFileSystem.java:1265)
    at org.apache.hadoop.hdfs.DistributedFileSystem$20.doCall(DistributedFileSystem.java:1261)
    at org.apache.hadoop.fs.FileSystemLinkResolver.resolve(FileSystemLinkResolver.java:81)
    at org.apache.hadoop.hdfs.DistributedFileSystem.getFileStatus(DistributedFileSystem.java:1261)
```

➔ When HDFS is started



⇒ Therefore, Hive is dependent on HDFS.

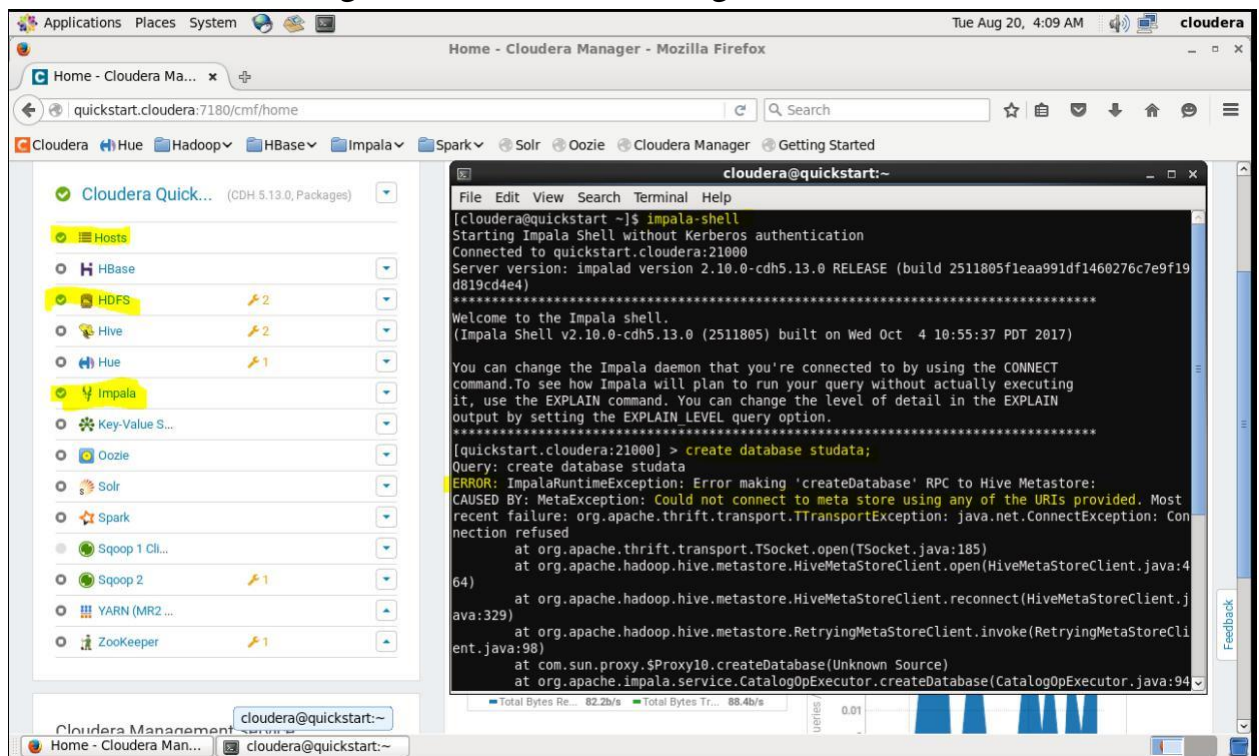
## IMPALA

Impala is MPP (Massive Parallel Processing) SQL query engine for processing huge volume of data stored in Hadoop cluster.

With Impala, user can communicate with HDFS or HBase using SQL queries in a faster way than SQL engine like HIVE.

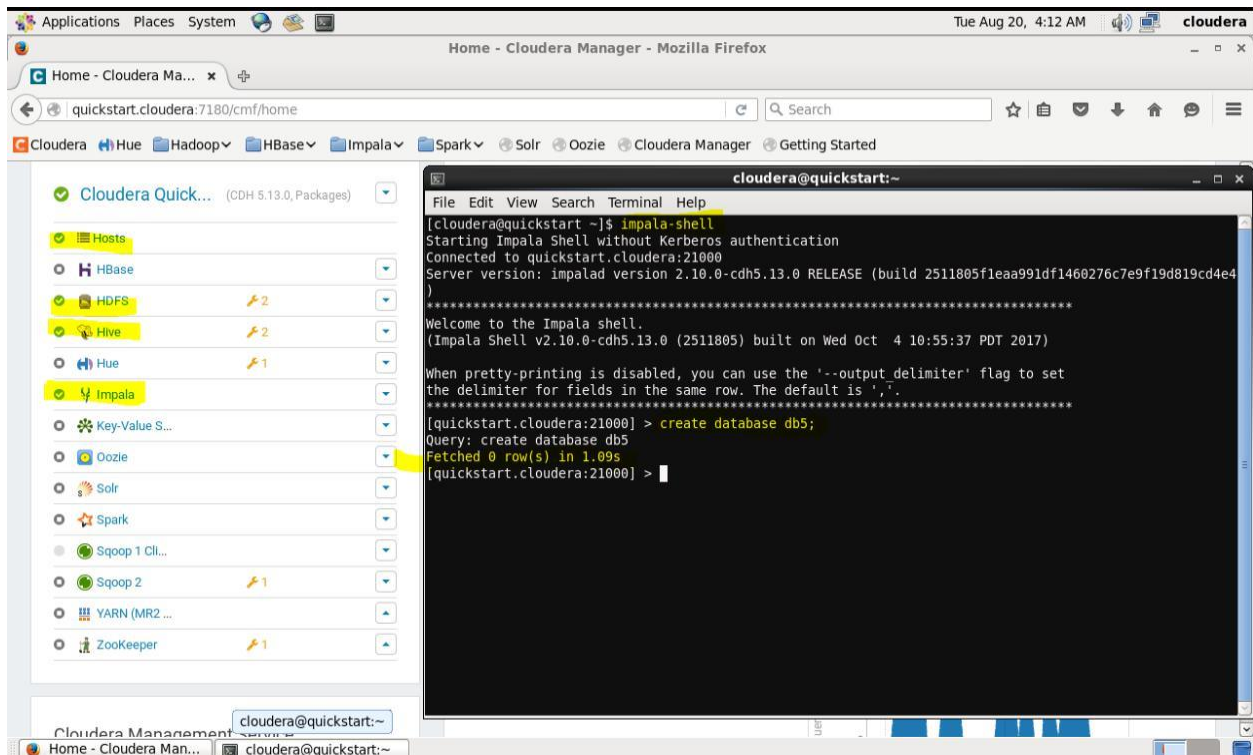
Impala is pioneering the use of the Parquet file format, a columnar storage layout that is optimized for large scale queries typical in data warehouse scenario.

➔ When HDFS is running but HIVE is not running





➔ When HDFS and HIVE both are running



⇒ Therefore ,Impala depends on Hive and HDFS.



## HBASE

HBase is a database built on top of the HDFS.

HBase provides faster lookups for larger tables.

HBase is used to have random, real-time read/write access to Big data.

HBase is a non-relational database modeled after Google's Bigtable.

➔ When HBase, HDFS are running but ZooKeeper is not running

The screenshot shows the Cloudera Manager web interface in a Mozilla Firefox browser. The left sidebar displays a list of services: Cloudera Quickstart, Hosts, HBase, HDFS, Hive, Hue, Impala, Key-Value Store, Oozie, Solr, Spark, Sqoop 1 Client, Sqoop 2, YARN (MR2), and ZooKeeper. A red arrow points to the ZooKeeper service, which is currently stopped. The main panel shows a terminal window titled 'cloudera@quickstart:~' with the following output:

```
at org.jruby.Ruby.runScript(Ruby.java:697)
at org.jruby.Ruby.runScript(Ruby.java:698)
at org.jruby.Ruby.runNormally(Ruby.java:597)
at org.jruby.Ruby.runFromMain(Ruby.java:446)
at org.jruby.Main.doRunFromMain(Main.java:369)
at org.jruby.Main.internalRun(Main.java:258)
at org.jruby.Main.run(Main.java:224)
at org.jruby.Main.run(Main.java:208)
at org.jruby.Main.main(Main.java:188)
19/08/20 05:22:48 ERROR client.ConnectionManager$HConnectionImplementation: Can't get c
connection to ZooKeeper: KeeperErrorCode = ConnectionLoss for /hbase

ERROR: KeeperErrorCode = ConnectionLoss for /hbase

List all tables in hbase. Optional regular expression parameter could
be used to filter the output. Examples:

hbase> list
hbase> list 'abc.*'
hbase> list 'ns:abc.*'
hbase> list 'ns:.*'

hbase(main):002:0>
```

Below the terminal window, there is a graph showing 'bytes / second' over time, with a peak around 05:15. To the right of the graph is a section titled 'Completed Impala Queries'.

➔ When HBASE and ZOOKEEPER are running but HDFS is not running

The screenshot shows the Cloudera Manager interface. On the left, the 'Cloudera Quickstart' page lists various services. HBase and ZooKeeper are highlighted in yellow and show a green status icon. HDFS is also highlighted in yellow but shows a red status icon. The terminal window on the right shows the HBase shell. The user has entered the command 'list' and received an error: 'ERROR: Can't get master address from ZooKeeper; znode data == null'. The terminal output shows the HBase shell prompt and the error message.

➔ When HBase , HDFS and ZooKeeper are running

The screenshot shows the Cloudera Manager interface. On the left, the 'Cloudera Quickstart' page lists various services. HBase, HDFS, and ZooKeeper are all highlighted in yellow and show a green status icon. The terminal window on the right shows the HBase shell. The user has entered the command 'create' to create a table named 'emp' with columns 'personal data' and 'professional data'. The terminal output shows the HBase shell prompt, the command, and the successful creation of the table. The user then enters the command 'list' and receives the output: 'emp' and '1 row(s) in 0.0170 seconds'.

⇒ Therefore ,HBase depends on HDFS and Zookeeper.