COMP 6231

DISTRIBUTED SYSTEM DESIGN

Project: Apache HBase

TEAM 12

JAYNIL SAVANI-40156070

MANTHAN MORADIYA-40156072

VICKY PATEL-40185238

YASH VAGHANI-40155884

SUBMITTED TO: DR. ESSAM MANSOUR

Introduction

- In this project, we are implementing distributed system using Apache HBase. Basically, HBase is an open source, non-relational distributed database model which is written in java and uses column-oriented dataset.
- HBase runs on the top of Hadoop Distributed File System (HDFS). Moreover, it uses the Map-Reduce framework to retrieve and store data. Apache HBase totally depends on Google's BigTable.

Objective

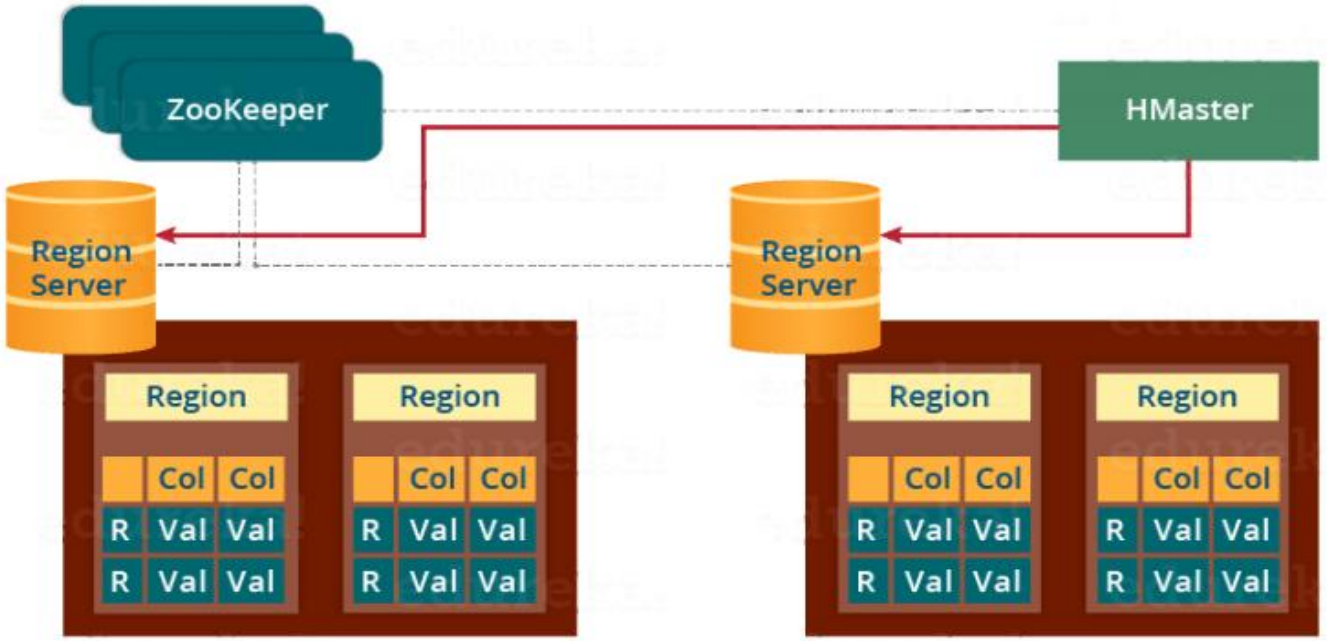
- The objective of this Apache HBase is to store and process large amounts of data, specifically to handle the data which consists thousands of rows and columns to get resultant data in the form of <key, value> pair because it uses the HDFS where MapReduce functioning the data in this format.

PREREQUISITES

Below are the prerequisites which needed to implement this project.

- Google Cloud Platform (GCP)
  - To create VM Instances to create 1 NameNode and 3 DataNodes.
  - OS Type: CENTOS 7
  - Hardware Specification:
    - NameNode
      - 4 core CPU, 8 GB RAM, 50 GB Balanced storage
    - DataNode
      - 2 core CPU, 4 GB RAM, 50 GB Balanced storage
- HADOOP (ver. 3.2.2)
- JAVA (ver. 1.8.0_131)

Apache HBase Architecture



Apache HBase Architecture

Apache HBase Architecture

- HMaster:

  HMaster provides administrative services in cluster. It is responsible for load balancing, region assignment and other data related operations.

- Region Server:

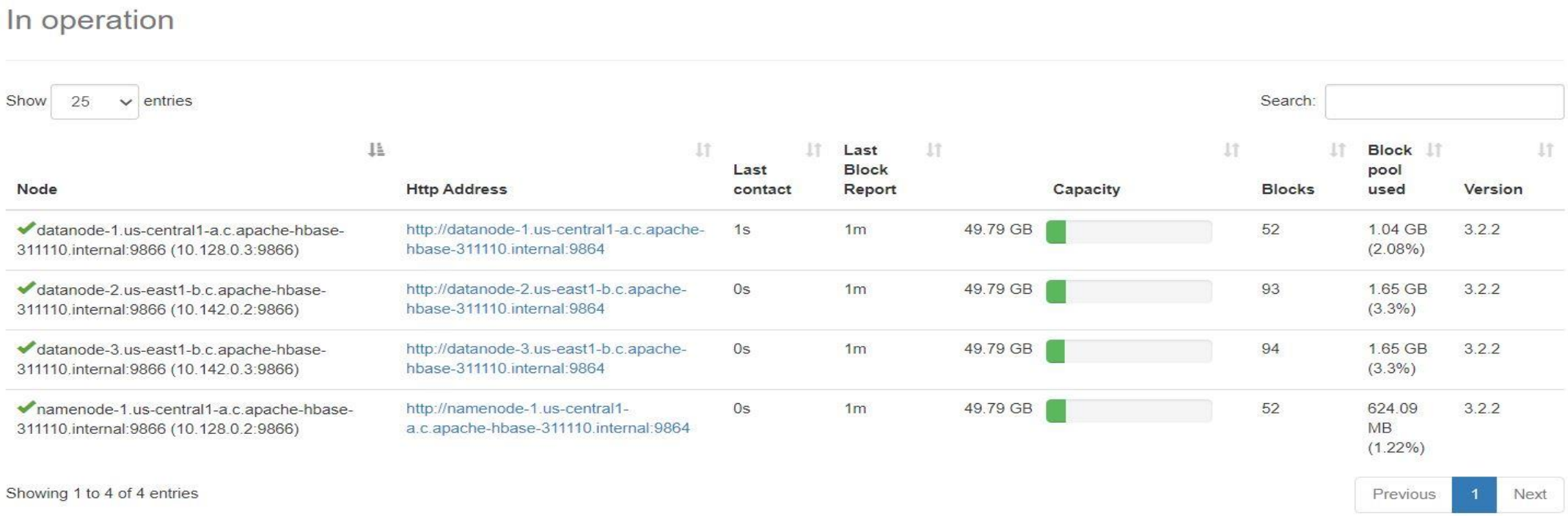
  Region server runs on the Datanodes and handle data related operations such as read and write requests for all the regions.

- ZooKeeper

  Zookeeper is a distributed application that required for coordination between master and region servers. It provides services like maintaining configuration information, naming, distributed synchronization, and group services.

HADOOP IMPLEMENTATION

- As Apache HBase runs on HDFS, so we need to configure HADOOP first. In this project, we have created 1 Namenode and 4 Datanode and we have set replication factor of 3.



In operation

Show 25 entries

Search:

Node	Http Address	Last contact	Last Block Report	Capacity	Blocks	Block pool used	Version
datanode-1.us-central1-a.c.apache-hbase-311110.internal:9866 (10.128.0.3:9866)	http://datanode-1.us-central1-a.c.apache-hbase-311110.internal:9864	1s	1m	49.79 GB	52	1.04 GB (2.08%)	3.2.2
datanode-2.us-east1-b.c.apache-hbase-311110.internal:9866 (10.142.0.2:9866)	http://datanode-2.us-east1-b.c.apache-hbase-311110.internal:9864	0s	1m	49.79 GB	93	1.65 GB (3.3%)	3.2.2
datanode-3.us-east1-b.c.apache-hbase-311110.internal:9866 (10.142.0.3:9866)	http://datanode-3.us-east1-b.c.apache-hbase-311110.internal:9864	0s	1m	49.79 GB	94	1.65 GB (3.3%)	3.2.2
namenode-1.us-central1-a.c.apache-hbase-311110.internal:9866 (10.128.0.2:9866)	http://namenode-1.us-central1-a.c.apache-hbase-311110.internal:9864	0s	1m	49.79 GB	52	624.09 MB (1.22%)	3.2.2

Showing 1 to 4 of 4 entries

Previous 1 Next

DataNodes in the System

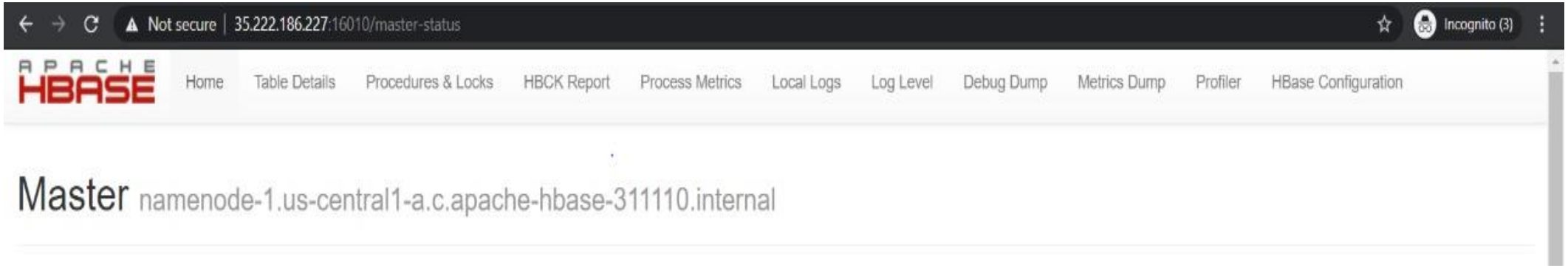
HADOOP IMPLEMENTATION

```
<configuration>
        <property>
                <name>dfs.replication</name>
                <value>3</value>
                <description>Default block replication.
                The actual number of replications can be specified when the file is created.
                The default is used if replication is not specified in create time.
                </description>
        </property>

        <property>
                <name>dfs.namenode.name.dir</name>
                <value>file:/usr/local/hadoop_store/hdfs/namenode</value>
        </property>

        <property>
                <name>dfs.datanode.data.dir</name>
                <value>file:/usr/local/hadoop_store/hdfs/datanode</value>
        </property>

        <property>
                <name>dfs.namenode.checkpoint.dir</name>
                <value>file:/usr/local/hadoop_store/hdfs/secondarynamenode</value>
        </property>

        <property>
                <name>dfs.namenode.checkpoint.period</name>
                <value>3600</value>
        </property>

</configuration>
                                                                              48,0-1
```

Hdfs-site.xml

HBase IMPLEMENTATION

- We used Apache HBase version 2.2.2 in this project which consists 1 HMaster, 3 ZooKeeper, and 3 Region Server.



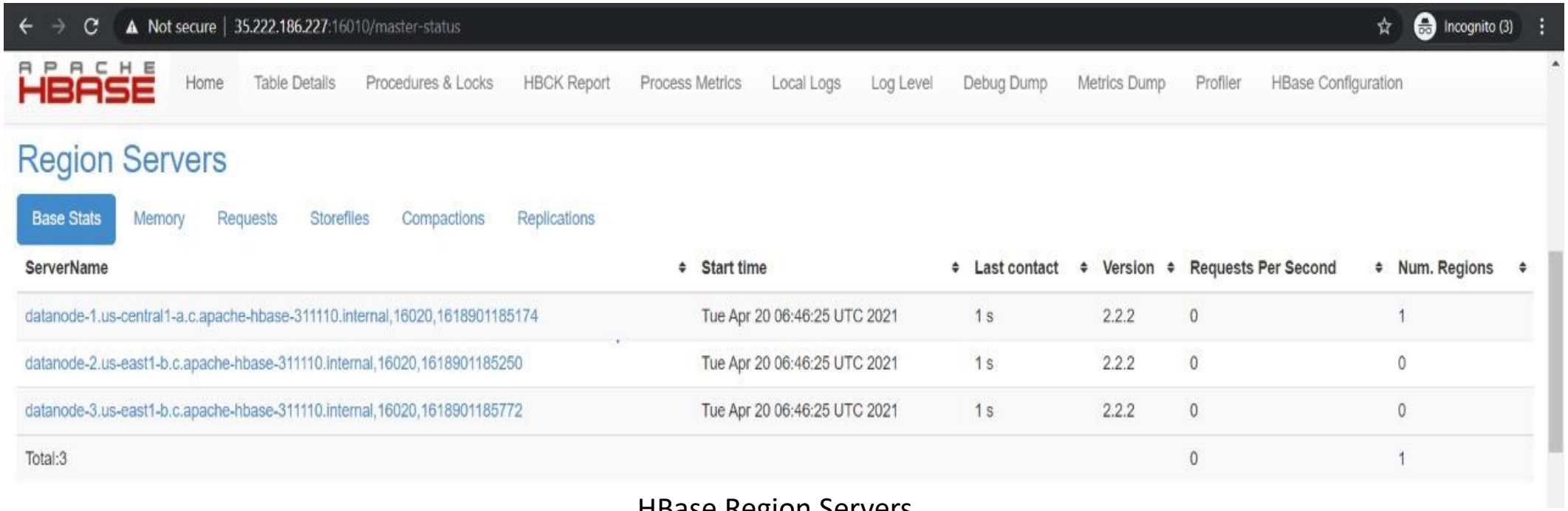
HBase HMaster

HBase IMPLEMENTATION

ZooKeeper Client Version	3.4.10, revision=-1	ZooKeeper client version and revision
ZooKeeper Client Compiled	03/23/2017 10:13 GMT	When ZooKeeper client version was compiled
ZooKeeper Quorum	10.128.0.2:2181 10.128.0.3:2181 10.142.0.2:2181	Addresses of all registered ZK servers. For more, see zk dump.
ZooKeeper Base Path	/hbase	Root node of this cluster in ZK.

HBase ZooKeeper

HBase IMPLEMENTATION



HBase Region Servers

HBase IMPLEMENTATION

- So far, we have configured HADOOP and Apache HBase.

```
[dsd@namenode-1 ~]$ jps
3203 HMaster
1589 DataNode
2040 ResourceManager
1466 NameNode
3066 HQuorumPeer
3515 Jps
1804 SecondaryNameNode
2173 NodeManager
[dsd@namenode-1 ~]$ hostname -f && date
namenode-1.us-central1-a.c.apache-hbase-311110.internal
Tue Apr 20 07:29:04 UTC 2021
[dsd@namenode-1 ~]$
```

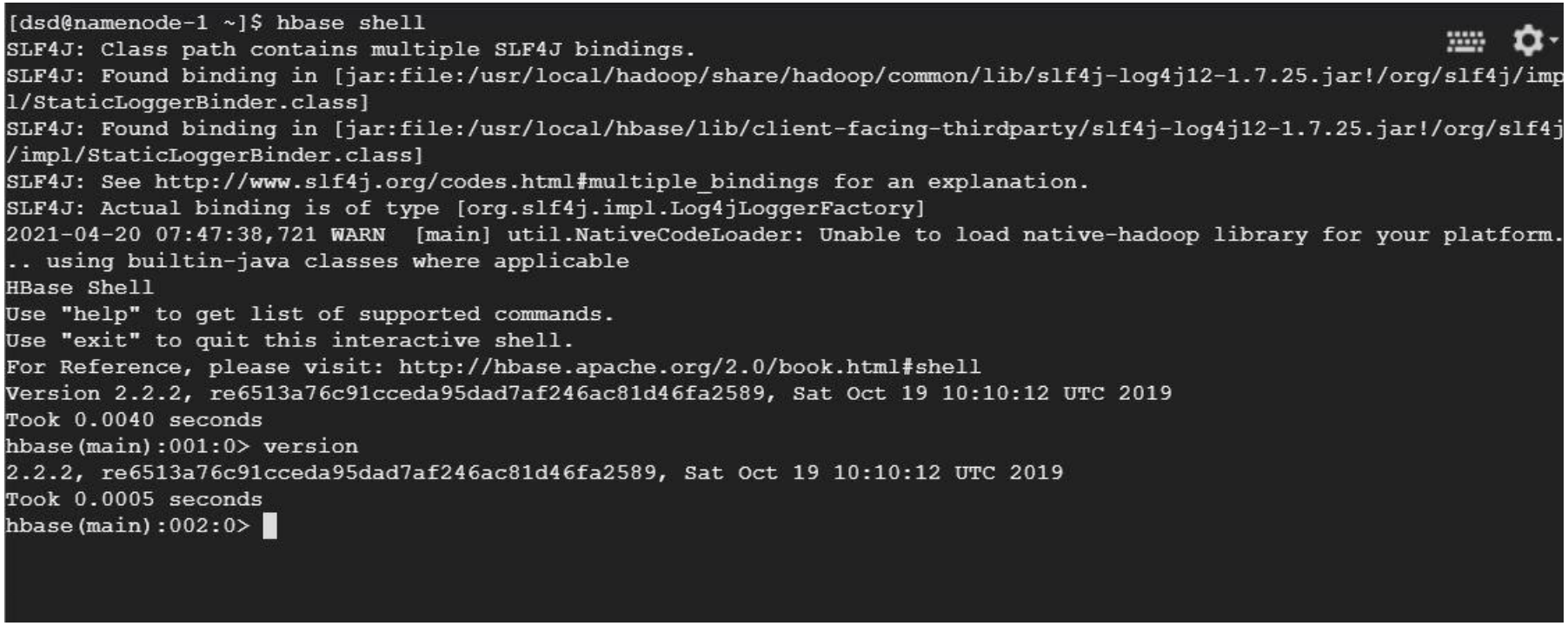
Services running on HMaster

```
[dsd@namenode-1 ~]$ ssh datanode-1
Last login: Mon Apr 19 08:21:27 2021
[dsd@datanode-1 ~]$ jps
1378 NodeManager
1738 HRegionServer
1260 DataNode
1580 HQuorumPeer
2687 Jps
[dsd@datanode-1 ~]$ hostname -f && date
datanode-1.us-central1-a.c.apache-hbase-311110.internal
Tue Apr 20 07:30:01 UTC 2021
[dsd@datanode-1 ~]$
```

Services running on one of the Region Server

HBase Shell

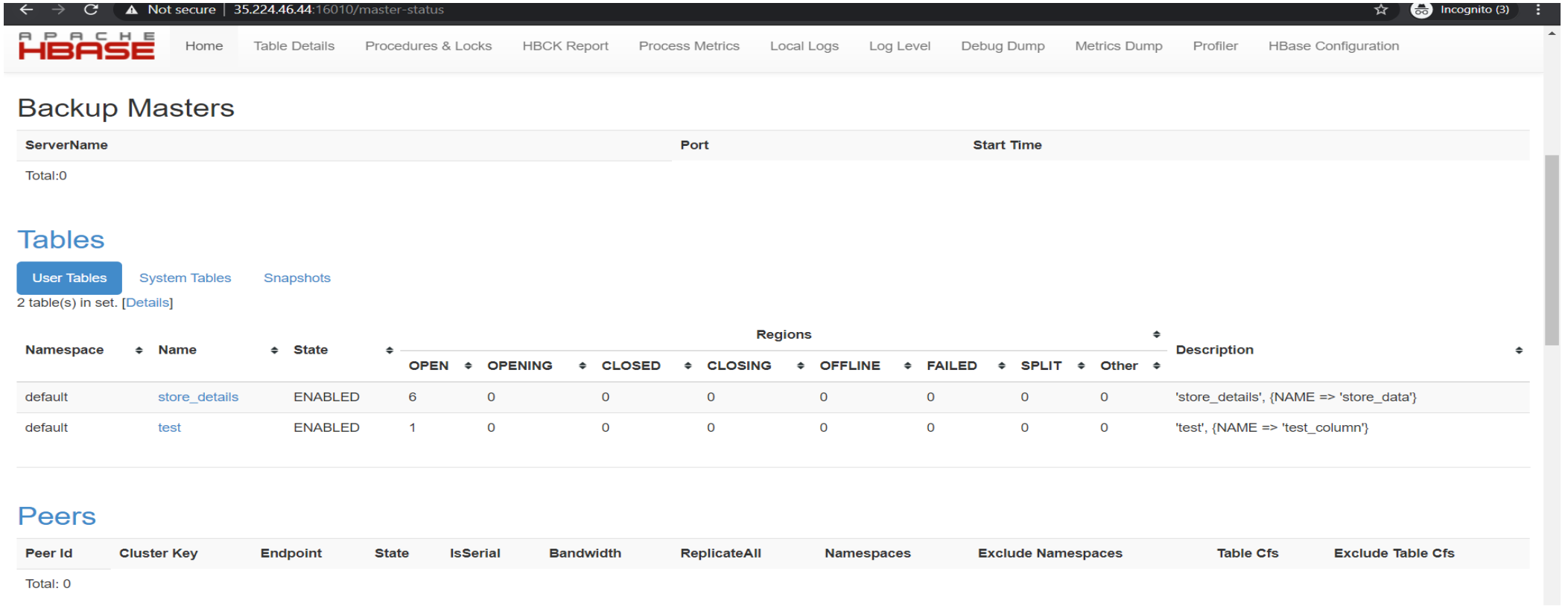
- HBase has a shell through which client can access or update the data which is stored in HDFS.



HBase Shell

HBase User Interface

- One can access HBase user interface to see all the configuration, data and to manage node



User Interface

REFERNCES

- https://www.edureka.co/blog/hbase-architecture/

THANK YOU