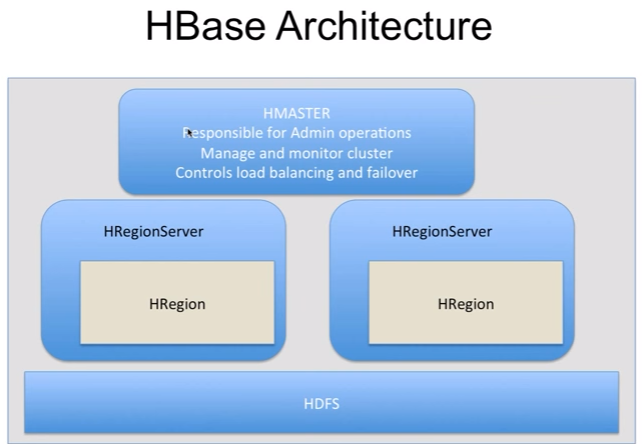
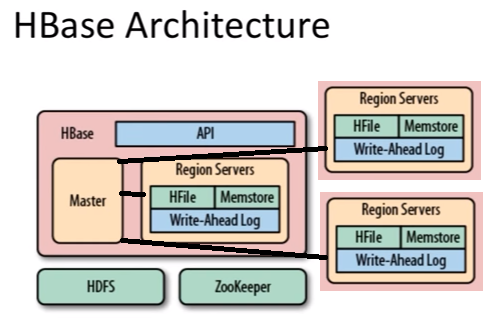
**Quick Ref: HBase**

****

****

|  |  |  |
| --- | --- | --- |
| **S.No** | **Topic** | **Desc** |
|  | **Web Ref** |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  | **General Info** |  |
| 1 | **Difference between Hive, Impala and HBase** | **Hive:**   1. Used for both reading & writing data into DB. But mainly used for reading data from DB. 2. Use HDFS for storage and MapReduce for processing 3. Hive will use MapReduce 4. Hive is used for batch processing   **Impala:**   1. Used for both reading & writing data into DB. But mainly used for reading data from DB. 2. Impala will NOT use MapReduce 3. Impala is for kind of adhoc query   **HBase:**   1. Used for both reading & writing data into DB. But mainly used for writing data into DB 2. HBase will NOT use MapReduce for processing. Instead it uses other components 3. Hbase is not a relational DB. It uses (key,value) concept. So it has different syntax. 4. Create table with column family name, no need to give column name while create a table. ex: add skills in linked in (new column), endorsed by multiple people (is data stored in the column) |
| 2 | About HBase | HBase is a NoSQL data store which uses HDFS as storage and can be used for applications that are operational in nature. HBase will not use MapReduce for processing. HBase is not a right candidate for complex transaction. |
| 3 | Parameter file | Hbase-site.xml |
| 4 | How uses HBase | Facebook messenger: Chat tool which also stores data. No complex transaction involved. But operationally nature.  LinkedIn: |
| 5 | HBase Architecture | 1. Three main components: 2. HMaster (At least 3 Master nodes): For admin operation 3. HRegionServer (All slave nodes) 4. ZooKeeper: Coordination service 5. Three Sub Components in HRegionServer: 6. Memstore 7. HFile 8. WAL (Write Ahead Log) 9. Finally the data will be stored in HDFS |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  | **Command** |  |
| 1 | Login | # To login  >hbase shell |
| 2 | Help | # To get help  >help |
| 4 | Command Group: General  *<The command help will provide the complete details>* | # Commands available under ‘General’ group  >status;  >table\_help;  >version;  >whoami; |
| 5 | **Command Group: DDL** | # ‘some’ Commands available under ‘DDL’ group  >alter  >create  >describe  >disable –or- disable all  >drop –or – drop all  >enable –or – enable all  >list |
| 5.1 | Create table  *<under DDL group>* | Htable1 is table name and cf01 is column family name  >create ‘Htable1’,’cf01’ |
| 5.2 | To list all tables  *<under DDL group>* | # To list all tables  >list |
| 6 | **Command Group: DML** | # ‘some’ commands available under ‘DML’ group  >count  >delete –or—delete all  >get  >scan  >put  >truncate |
| 6.1 | To get count of records from a table  *<under DML group>* | # Count command returns number of rows.  >count ‘table\_name’ |
| 6.2 | To get records from table – using scan  *<under DML group>* | # Scan will return all the records/keys in the table. Note: If table has 10 columns then it will return 10 rows for each record. Because it is column based  >scan ‘Htable1’ |
| 6.3 | Write data into table name  *<under DML group>* | >put ‘Htable1’,’row\_key1’,’cf01:col1’,’col\_value1’ |
| 6.4 | To get records from table – using get command  *<under DML group>* | # Get will be used to get a count for a single key. – Need more R&D  >get ‘Htable1’, ‘cf01:col1’ – to be verified |
| 7 | Command Group: Namespace | # commands available under ‘Namespace’ group  >Create\_namespace  >alter\_namespace;  >describe\_namespace;  >drop\_namespace;  >list\_namespace;  >list\_namespace\_tables; |
| 8 | Command Group: tools | # ‘some’ commands available under ‘tools’ group  >balancer  >splitter |
| 9 | Command Group: snapshot | # ‘some’ commands available under ‘snapshot’ group  >clone\_snapshot  >delete\_snapshot  >list\_snapshot  >snapshot –or—snapshot\_all |
| 10 | Command group : configuration | # Commands available under ‘configuration’ group  >update\_all\_config  >update\_config |
| 11 | Command group: security | # Commands available under ‘security’ group  >grant  >revoke  >user\_permission |
| 12 | Distcp | DistCp (distributed copy) is a tool used for large inter/intra-cluster copying. It uses Map/Reduce to effect its distribution, error handling and recovery, and reporting. It expands a list of files and directories into input to map tasks, each of which will copy a partition of the files specified in the source list. Its Map/Reduce pedigree has endowed it with some quirks in both its semantics and execution.  Reference: Hadoop DistCp Guide |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Which three distcp features can you utilize on a Hadoop cluster?

1. Use distcp to copy HBase table files
2. Use distcp to copy data between directories inside the same cluster.
3. Use distcp to run an internal MapReduce job to copy files