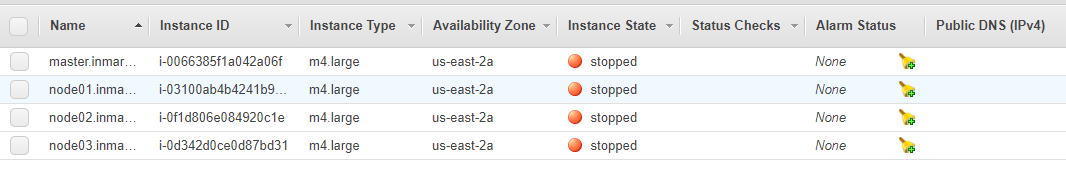
**Task 1 Installation:**

Ambari (2.6.2.2) And HDP (2.6.5) Installation on AWS (EC2 Instance)

1. Create 4 EC2 instances in AWS, on RHEL 6.10 with min 20 GB Size and also, add security group ( and make it as All Traffic)
2. And Create a Keypair and save it in our local
3. Select one instance on which you want to install Ambari server, and disable iptables ( service iptables stop and chkconfig iptables off)
4. And go to /etc/selinux/conf and disable Selinux setting
5. Download Ambari Repo from HW portal and place it in /etc/yum.repos.d and place it in all the servers
6. Go to Ambari server and install ambari-server and ambari-agent (needs to be installed on all the hosts)
7. Once Ambari-server is installed and go to /etc/ambari-server/conf/ambari-server.init and change the hostname from localhost to the ambari server hostname. This needs to be done on all the hosts
8. Do, ambari-server setup. And give all default values
9. Start ambari server and ambari agent ( on all hosts)
10. Go to the web browser and user ambari server hostname and port 8080 and login using admin/admin
11. Click on Launch installation wizard and select the HDP version you would like to install and select public / local repos from the list and type all the hostname in hosts dialogue box.
12. And install the HDP version and select the service you would like to install and review the settings and click deploy
13. Make the Yarn port 8050 and 8088 restrict to use only from your I/P as an application dr.who is hacking the yarn port 8088:  
    <https://community.hortonworks.com/questions/191672/yarn-application-undefined-created-by-drwho.html>

Resolution enable firewall rules/IPtables to open port 8050 and 8088 only for yarn resource node i/p, this can be done by iptables or security group config on AWS

AWS EC2 Instances:



Issues Encountered:  
There is a bug with RHEL 6\* on AWS with the installation of HDP 2.6\* version as the installation will fail with libtirpc-devel package.

Resolution:  
Go the Centos website and search for libtirpc-devel package for RHEL 6.\* version and manually install it on the hosts. And restart the HDP installation from Ambari.

<https://community.hortonworks.com/questions/199453/error-installing-components.html>

<https://community.hortonworks.com/answers/99526/view.html>

After Installation issues:  
  
Hive metastore and HS2 server won’t start after the installation due to mysql-javaconnector.jar issue.

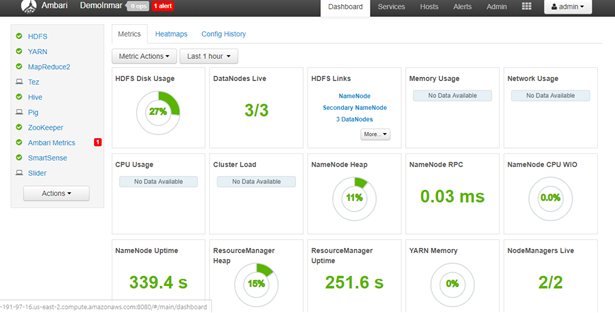
RCA: from logs:  
File "/usr/lib/ambari-agent/lib/resource\_management/core/source.py", line 197, in get\_content

raise Fail("Failed to download file from {0} due to HTTP error: {1}".format(self.url, str(ex)))

resource\_management.core.exceptions.Fail: Failed to download file from http://ip-172-31-15-36.us-east-2.compute.internal:8080/resources/mysql-connector-java.jar due to HTTP error: HTTP Error 404: Not Found

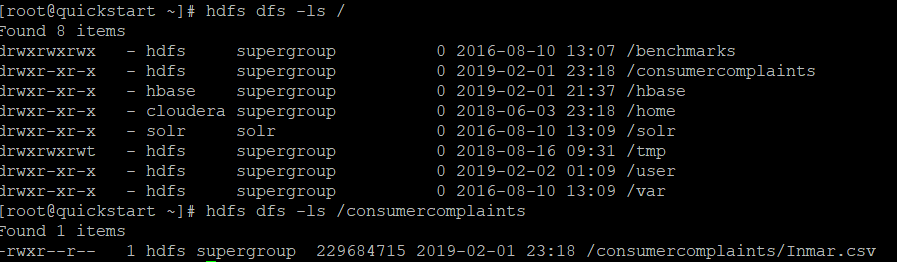
Resolution:  
  
Download my-sql-java-connector.jar and install in it on HS2 node and create a link mentioned the links below and restart Hivemetastore and HS2 from Ambari

<https://community.hortonworks.com/articles/170133/hive-start-failed-because-of-ambari-error-mysql-co.html>

<https://community.pivotal.io/s/article/Hive-Services-Fail-to-Start-giving-Error-HTTP-Error-404-Not-Found>

**Task 2 Hadoop Confidence Test**:  
**HDFS**:

login as hdfs user from terminal  
Create a hdfs folder (consumercomplaints) :  
Command: hdfs dfs –mkdir /consumercomplaints  
Command: hdfs dfs –chown user-name:hdfs /consumercomplaints  
Command: hdfs dfs –chmod 744 /consumercomplaints



**Hive**:

Issues: While creating Partitioned Tables: We need to set 2 parameters :  
  
SET hive.exec.dynamic.partition = true;  
SET hive.exec.dynamic.partition.mode = nonstrict;  
Login to Beeline-hive, using below command: and set this parameters

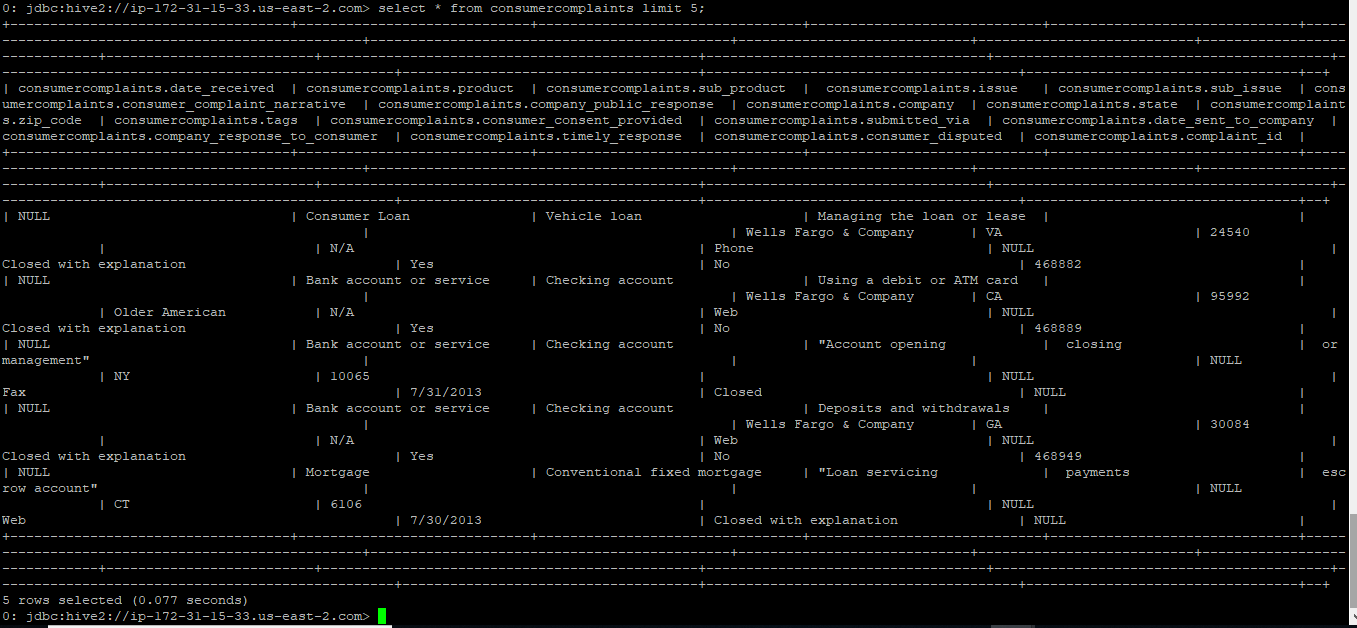
[hive@ip-172-31-15-36 ~]$ beeline -u jdbc:hive2://ip-172-31-15-33.us-east-2.compute.internal:10000

Create an external table called consumercomplaints and other table consumercomplaints\_partitioned by state  
  
Remove the header from csv file before you create an external table  
Command: sed –I 1d Inmar.csv  
External Table Query command:

|  |
| --- |
| CREATE EXTERNAL TABLE IF NOT EXISTS consumercomplaints (  Date\_received DATE,  Product STRING,  Sub\_product STRING,  Issue STRING,  Sub\_issue STRING,  Consumer\_complaint\_narrative STRING,  Company\_public\_response STRING,  Company STRING,  State String,  ZIP\_code INT,  Tags STRING,  Consumer\_consent\_provided STRING,  Submitted\_via STRING,  Date\_sent\_to\_company DATE,  Company\_response\_to\_consumer STRING,  Timely\_response STRING,  Consumer\_disputed STRING,  Complaint\_ID INT)  ROW FORMAT DELIMITED  FIELDS TERMINATED BY ','  STORED AS TEXTFILE  location '/consumercomplaints'; |

Partitioned table Query Command:

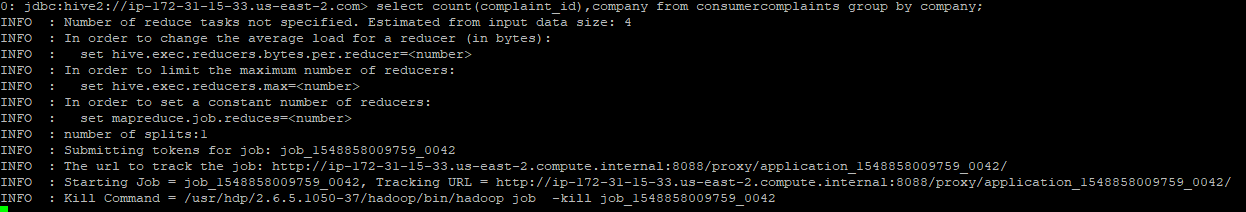
|  |
| --- |
| CREATE EXTERNAL TABLE IF NOT EXISTS consumercomplaints\_partitioned (  Date\_received DATE,  Product STRING,  Sub\_product STRING,  Issue STRING,  Sub\_issue STRING,  Consumer\_complaint\_narrative STRING,  Company\_public\_response STRING,  Company STRING,  ZIP\_code INT,  Tags STRING,  Consumer\_consent\_provided STRING,  Submitted\_via STRING,  Date\_sent\_to\_company DATE,  Company\_response\_to\_consumer STRING,  Timely\_response STRING,  Consumer\_disputed STRING,  Complaint\_ID INT)  Partitioned by (State STRING)  ROW FORMAT DELIMITED  FIELDS TERMINATED BY ','  STORED AS TEXTFILE  location '/consumercomplaints'; |
|  |
|  |



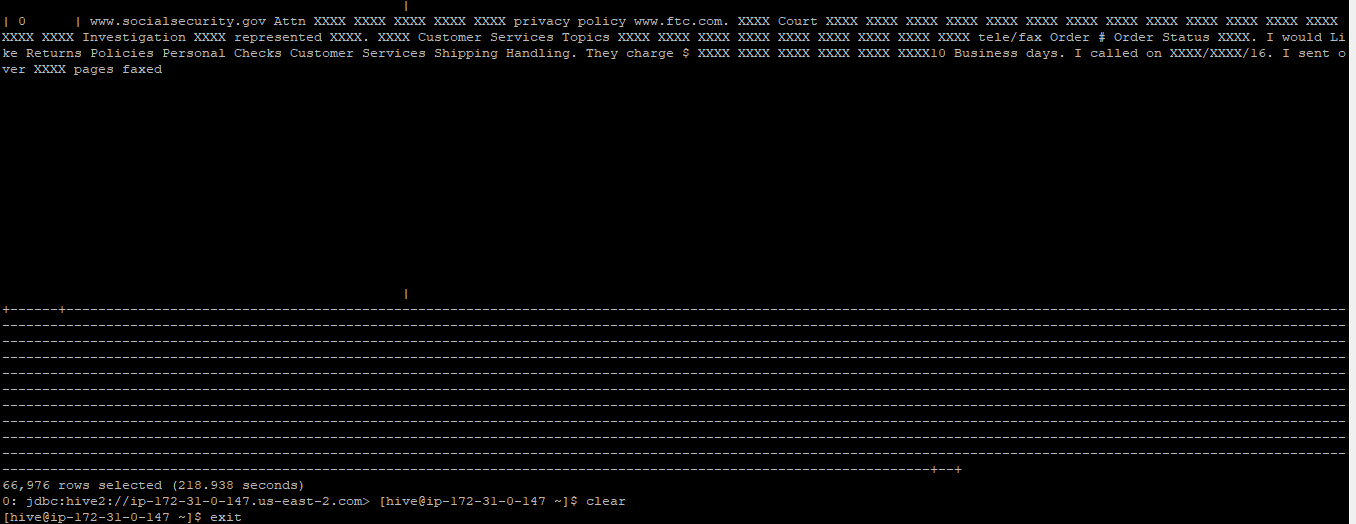
Issue: Hive query taking too long time with   
jdbc:hive2://ip-172-31-15-33.us-east-2.com> select count(complaint\_id),company from consumercomplaints group by company;

INFO : Tez session hasn't been created yet. Opening session

Resolution: changed: set hive.execution.engine=mr;



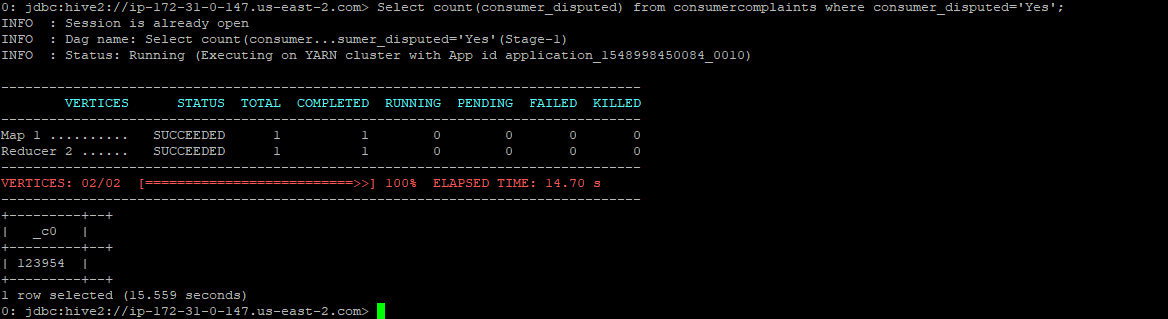
1. Total number of complaints for each company:  
   select count(complaint\_id),company from consumercomplaints group by company

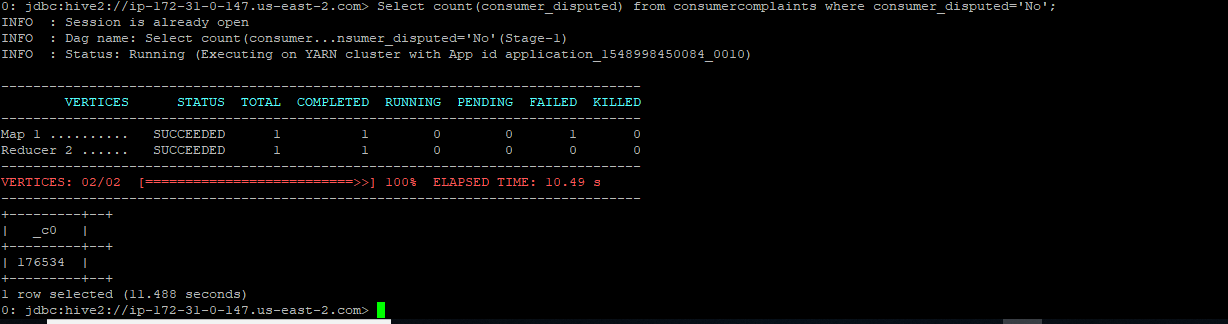


1. Total number of Disputed vs. Undisputed consumer complaints

Select count(consumer\_disputed) from consumercomplaints where consumer\_disputed=Yes

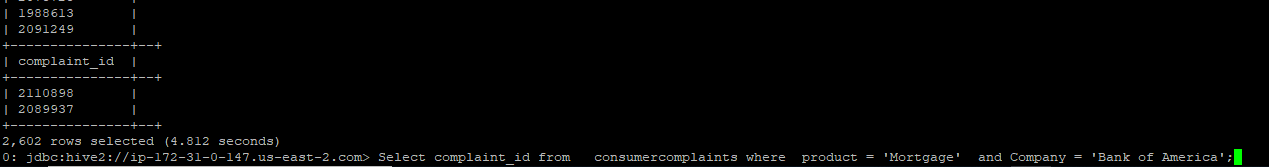
Select count(consumer\_disputed) from consumercomplaints where consumer\_disputed=No





1. All consumer complaints centered around Mortgages that were being foreclosed upon from Bank of America

Select complaint\_id from consumercomplaints where product=”Mortgage” and Company=”BankofAmerica”

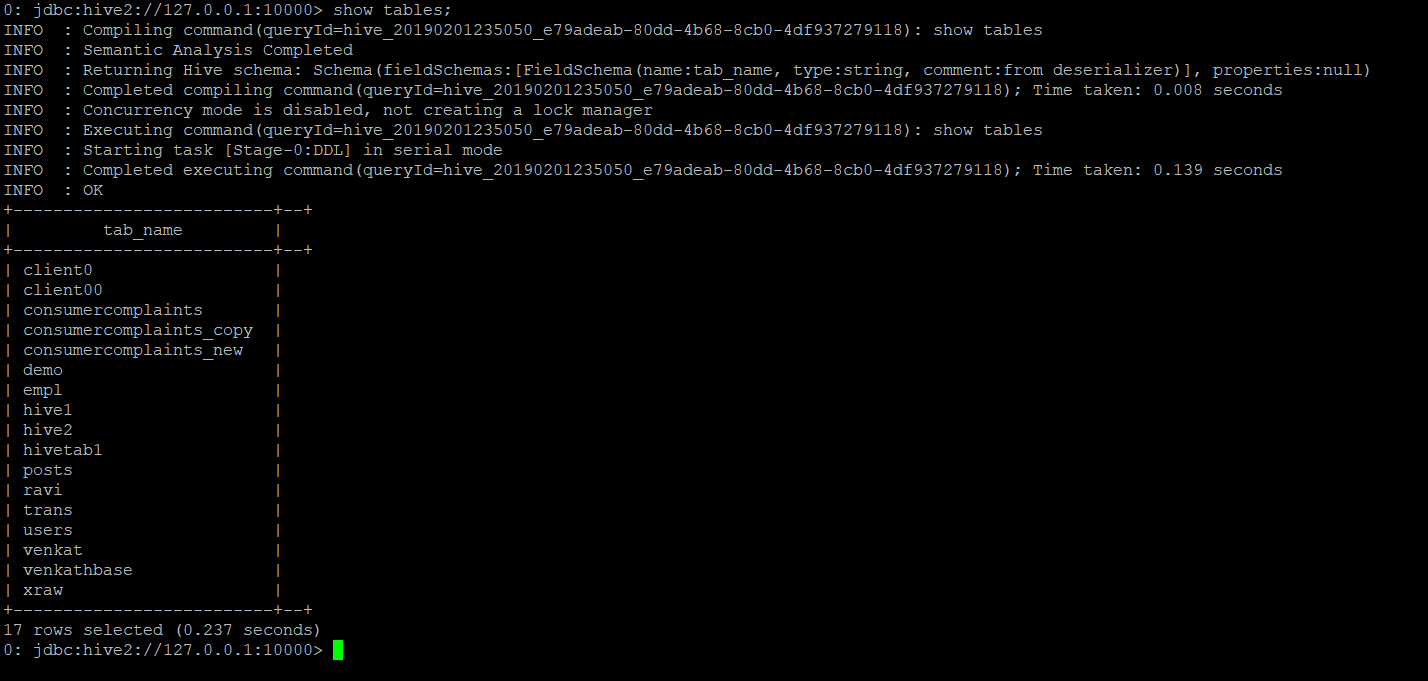


Create a new table called: consumercomplaints\_new. Insert data from consumercomplaints to consumercomplaints\_new:

Create a new table: consumercomplaint\_new:

Query:

|  |
| --- |
| CREATE TABLE IF NOT EXISTS consumercomplaints\_new (  Date\_received DATE,  Product STRING,  Sub\_product STRING,  Issue STRING,  Sub\_issue STRING,  Consumer\_complaint\_narrative STRING,  Company\_public\_response STRING,  Company STRING,  State STRING,  ZIP\_code INT,  Tags STRING,  Consumer\_consent\_provided STRING,  Submitted\_via STRING,  Date\_sent\_to\_company DATE,  Company\_response\_to\_consumer STRING,  Timely\_response STRING,  Consumer\_disputed STRING,  Complaint\_ID INT)  ROW FORMAT DELIMITED  FIELDS TERMINATED BY ','  STORED AS TEXTFILE  location '/consumercomplaints'; |



Create a copy table partitioned by state:

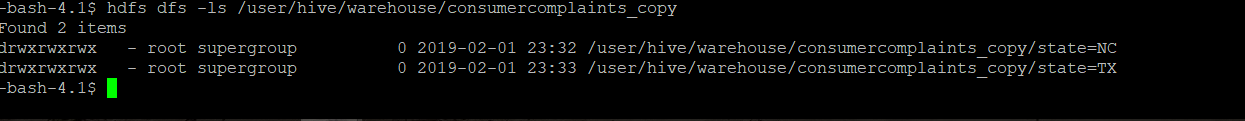
|  |
| --- |
| CREATE TABLE IF NOT EXISTS consumercomplaints\_copy (  Date\_received DATE,  Product STRING,  Sub\_product STRING,  Issue STRING,  Sub\_issue STRING,  Consumer\_complaint\_narrative STRING,  Company\_public\_response STRING,  Company STRING,  ZIP\_code INT,  Tags STRING,  Consumer\_consent\_provided STRING,  Submitted\_via STRING,  Date\_sent\_to\_company DATE,  Company\_response\_to\_consumer STRING,  Timely\_response STRING,  Consumer\_disputed STRING,  Complaint\_ID INT)  Partitioned by (State STRING) |

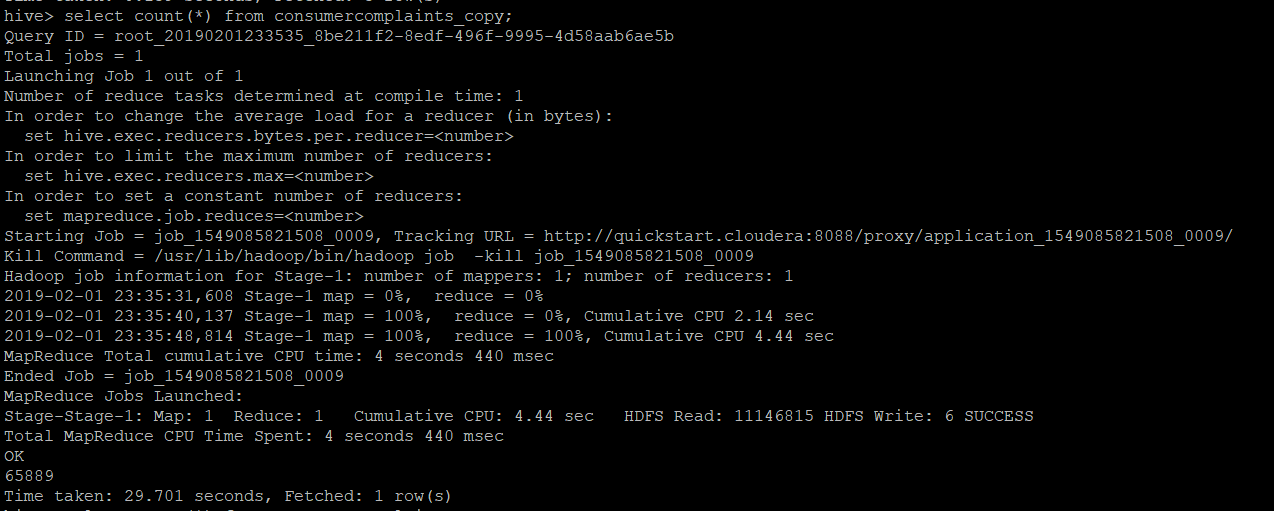
Copy all rows that have state = ‘NC’ or state = ‘TX’ out of the consumercomplaints\_new table into a separate table

Query:

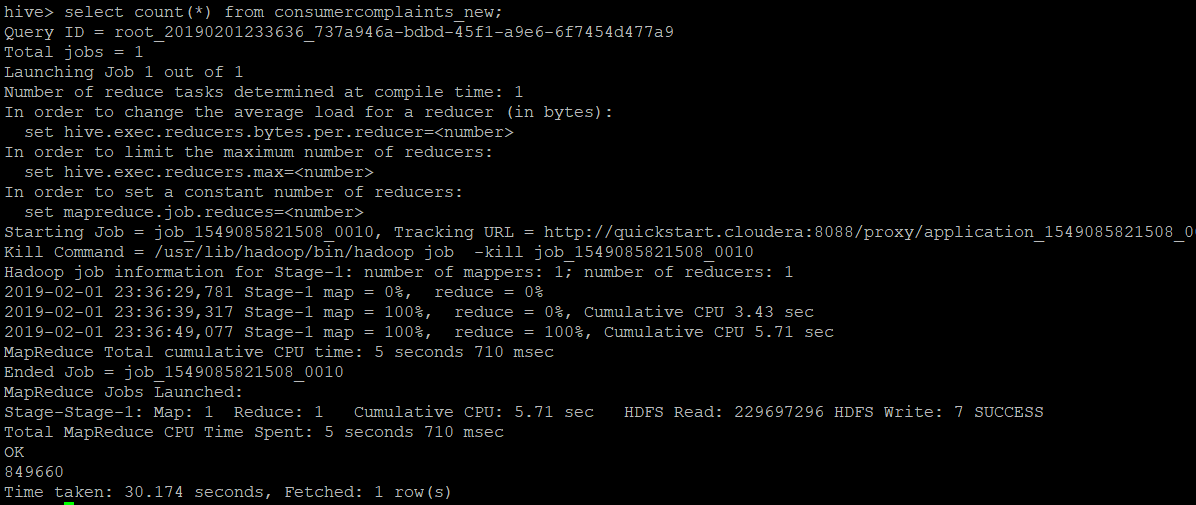
|  |
| --- |
| insert into table consumercomplaints\_copy partition (State='NC')  select Date\_received,Product,Sub\_product,Issue,Sub\_issue,Consumer\_complaint\_narrative,Company\_public\_response,Company,ZIP\_code,Tags,Consumer\_consent\_provided,Submitted\_via,Date\_sent\_to\_company,Company\_response\_to\_consumer ,Timely\_response,Consumer\_disputed,Complaint\_ID from consumercomplaints\_new where State='NC'; |
| insert into table consumercomplaints\_copy partition (State='TX')  select Date\_received,Product,Sub\_product,Issue,Sub\_issue,Consumer\_complaint\_narrative,Company\_public\_response,Company,ZIP\_code,Tags,Consumer\_consent\_provided,Submitted\_via,Date\_sent\_to\_company,Company\_response\_to\_consumer ,Timely\_response,Consumer\_disputed,Complaint\_ID from consumercomplaints\_new where State='TX; |







Delete all rows out of the consumercomplaints\_new table that have state = ‘NC’ or state = ‘TX’  
select count (\*) from consumercomplaints\_new;



Query: delete from consumercomplaints\_new where State='NC';

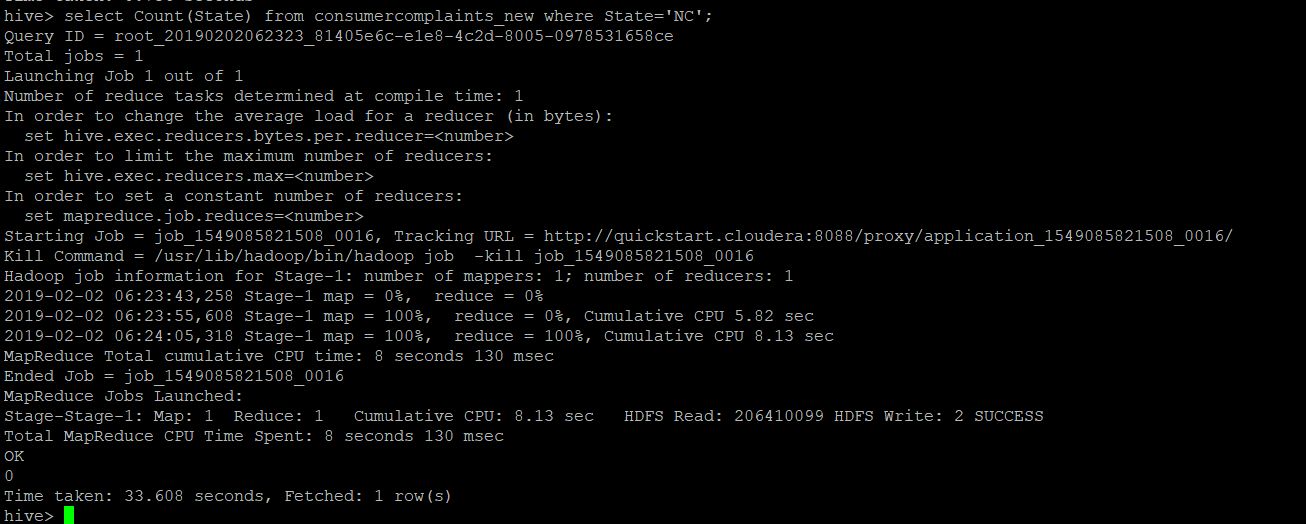
Issue: Error: Error while compiling statement: FAILED: SemanticException [Error 10294]: Attempt to do update or delete using transaction manager that does not support these operations. (state=42000,code=10294)

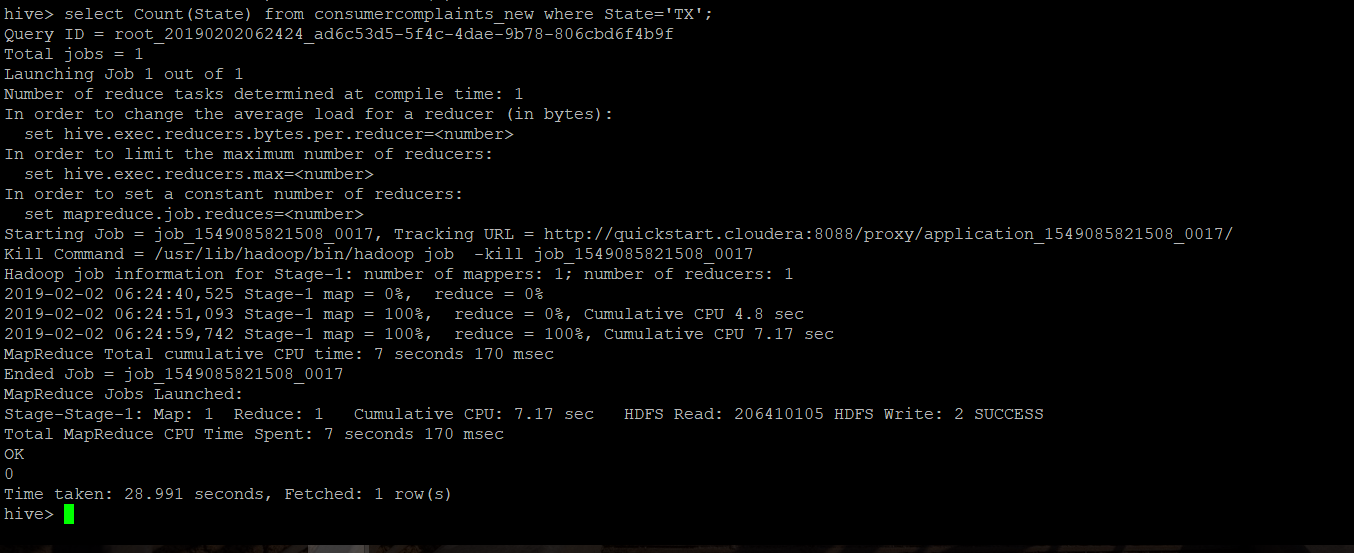
Solution:   
Set this properties from beeline in hive:  
set hive.support.concurrency=true  
set hive.enforce.bucketing=true  
set hive.exec.dynamic.partition.mode=nonstrict  
set hive.txn.manager=org.apache.hadoop.hive.ql.lockmgr.DbTxnManager  
set hive.compactor.initiator.on = true  
set hive.compactor.worker.threads = 1

<https://community.hortonworks.com/questions/21682/delete-and-update-hive.html>

Query:

|  |
| --- |
| Delete from consumercomplaint\_new where State=’NC’ Delete from consumercomplaint\_new where State=’TX’ |

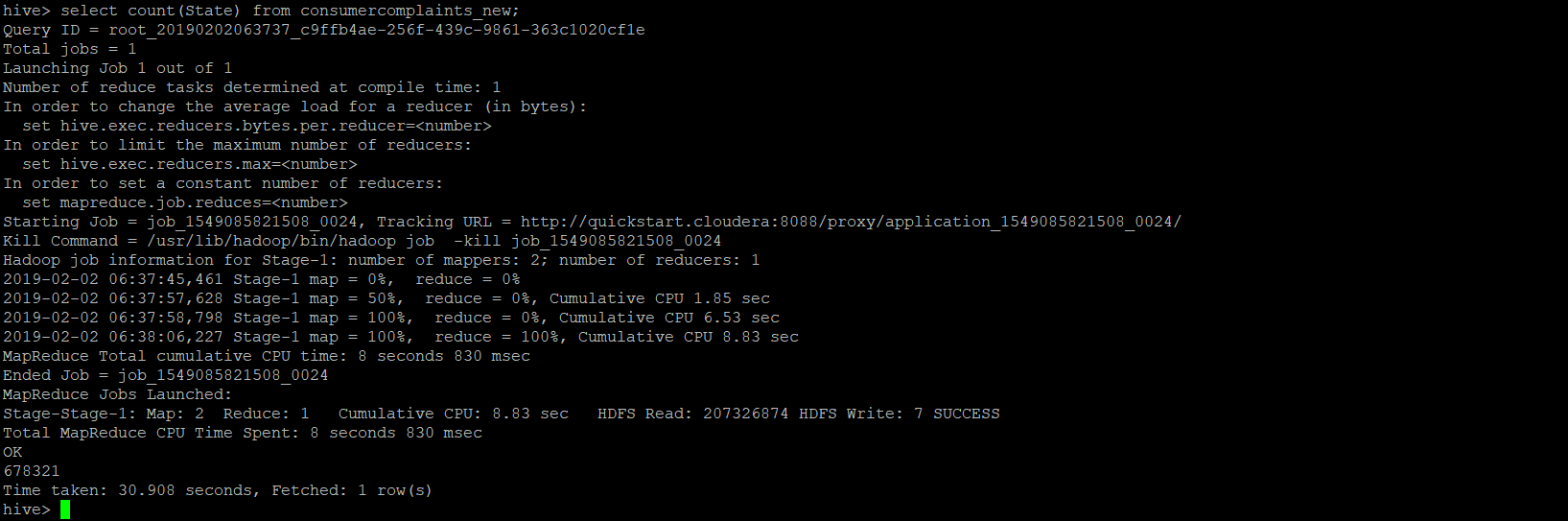




Insert:

|  |
| --- |
| INSERT INTO consumercomplaints\_new SELECT \* FROM consumercomplaints\_copy WHERE State='NC'; |





**General Hadoop Administration:**

**Decommissioning of Datanode:**

Type: Decommissioning from Ambari. Node: ip-172-31-5-153.us-east-2.compute.internal

1) Please first stop Storm components fully, to avoid corrupting files on HDFS

2) Please stop storm supervisor on the Data node that is being decommissioned

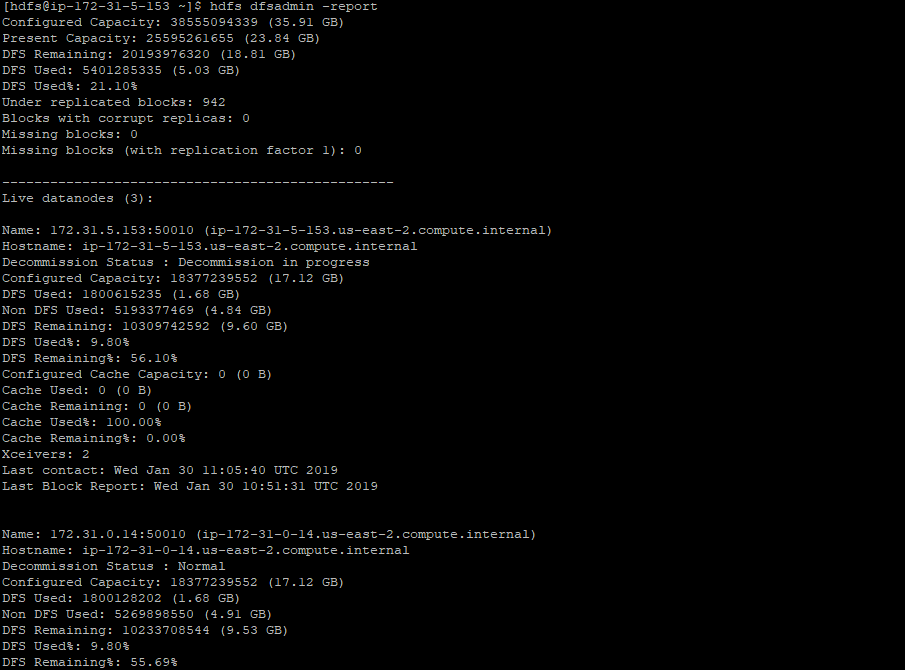
3) Delete Storm Supervisor on the machine being decommissioned

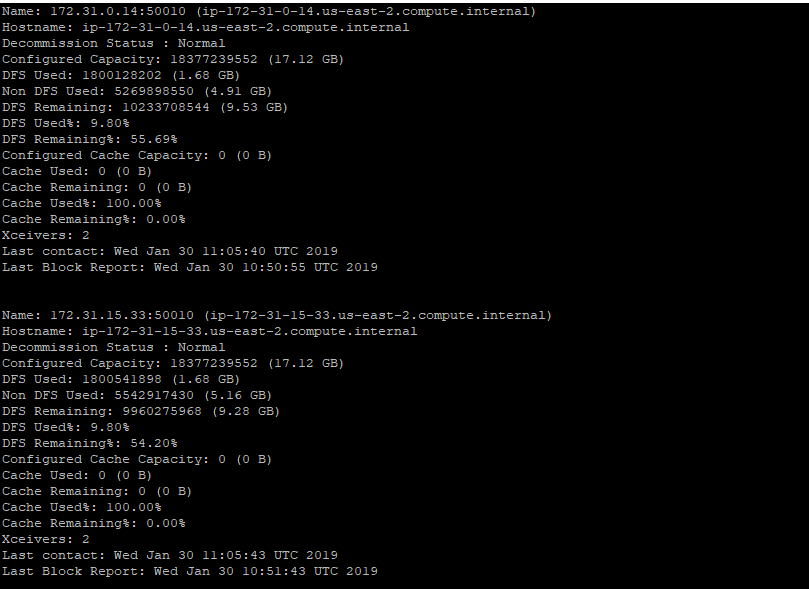
4) If there are any other services that are present on the datanode apart from above mentioned we need to decommission those services as well

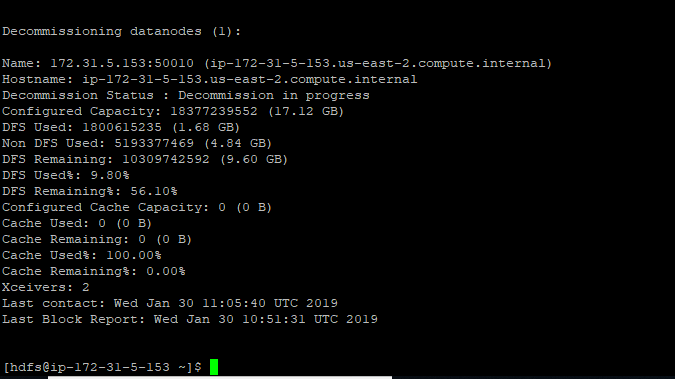
Process:

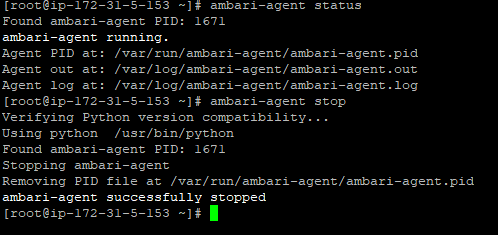
1. Decommission the Node Manager
   1. Ambari --> Hosts --> select Host --> Actions --> Selected Hosts 🡪 NodeManagers --> Decommission
2. Decommission the DataNode
   1. Ambari --> Hosts --> select Host --> Actions --> Selected Hosts 🡪 DataNodes --> Decommission
3. Stop Metrics Monitor
   1. Ambari --> Hosts --> select Host --> Actions --> Selected Hosts 🡪 Metrics Monitor --> Ambari Metrics 🡪 Stop
4. Wait until the data node decommissioning process has been completed.
   1. Run the command `hdfs dfsadmin -report` until all the decommissioned nodes are in the Decommissioned state
5. After above steps completed successfully, we need to stop the ambari-agent on the datanode.
   1. Login to your data node
   2. Stop Ambari Agent - sudo ambari-agent stop

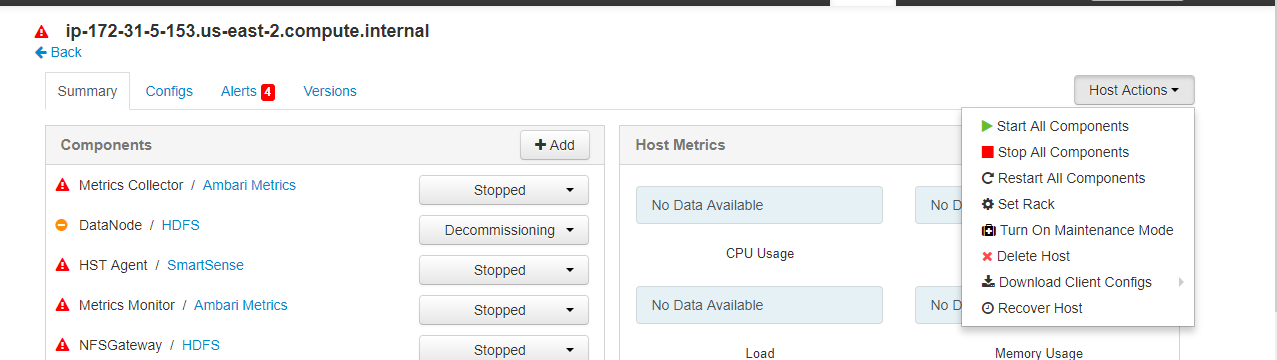
Once all the steps are completed. We can delete the host from Cluster. From the below steps:  
Ambari --> Hosts --> Select Host --> Host Actions --> Delete Host  
  
Since Metrics Collector is installed on this node. Before deleting this host we need to add the same to other host.

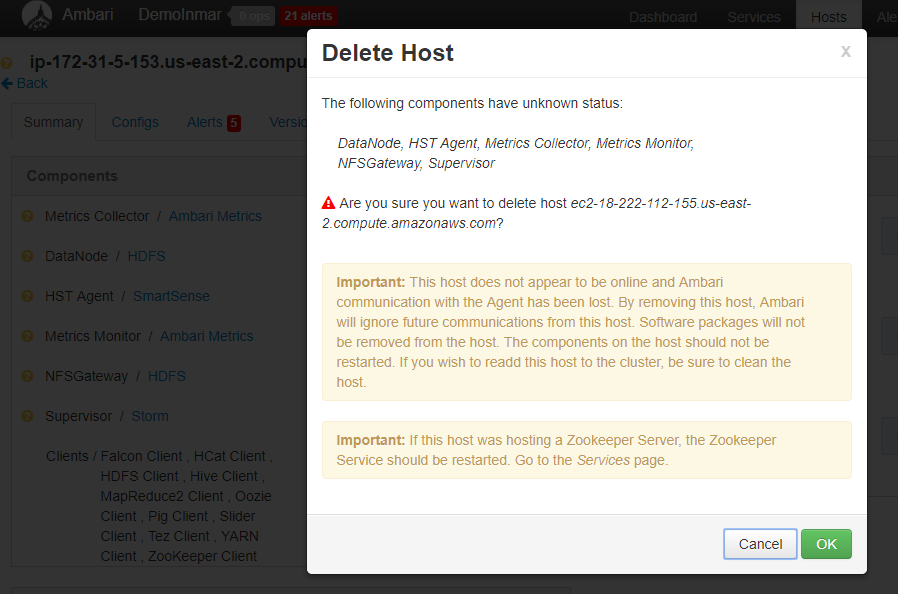


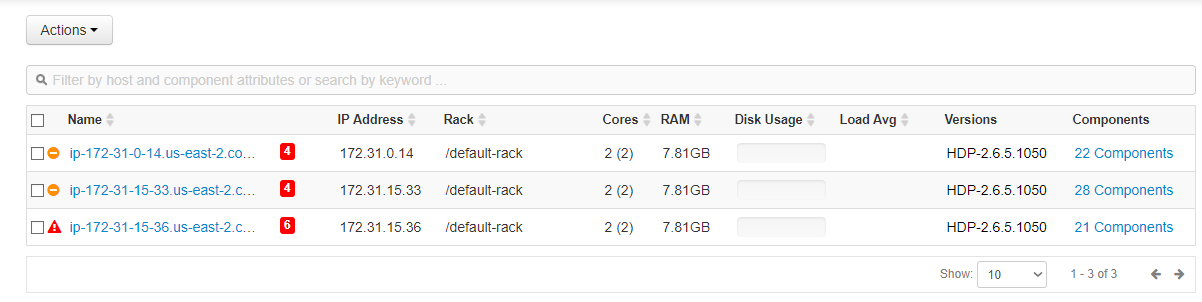






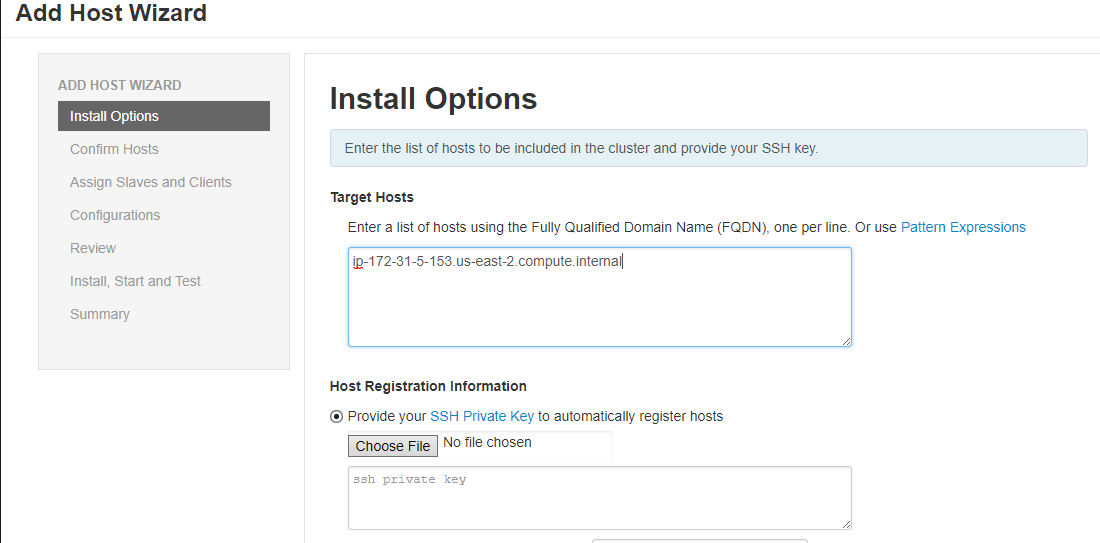


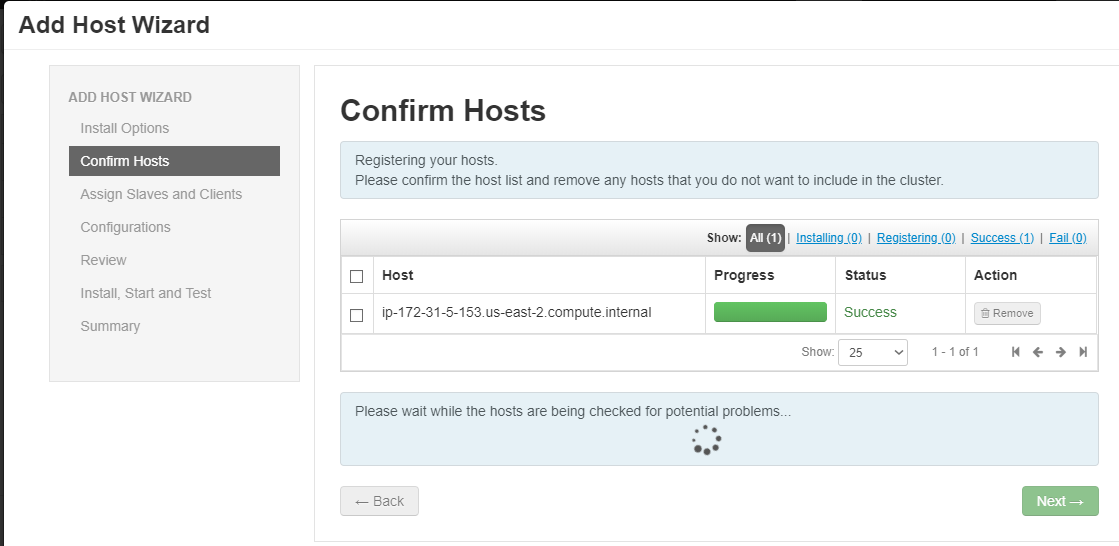
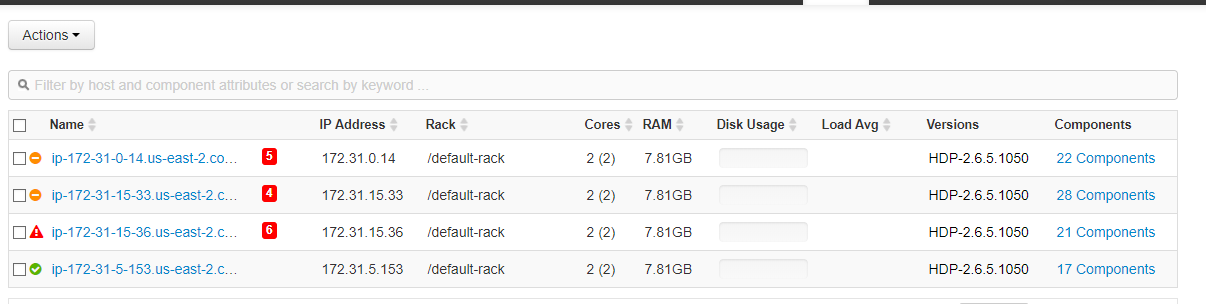


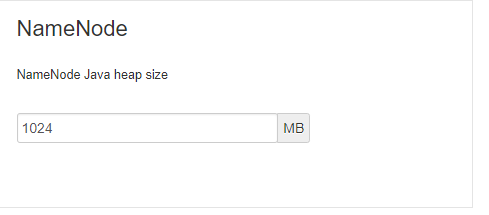


**Commissioning Datanode:**

1. Install Ambari Agent on the Host. If not installed earlier. And start the Ambari Agent.
2. From Ambari goto Hosts and add host. Give manual login

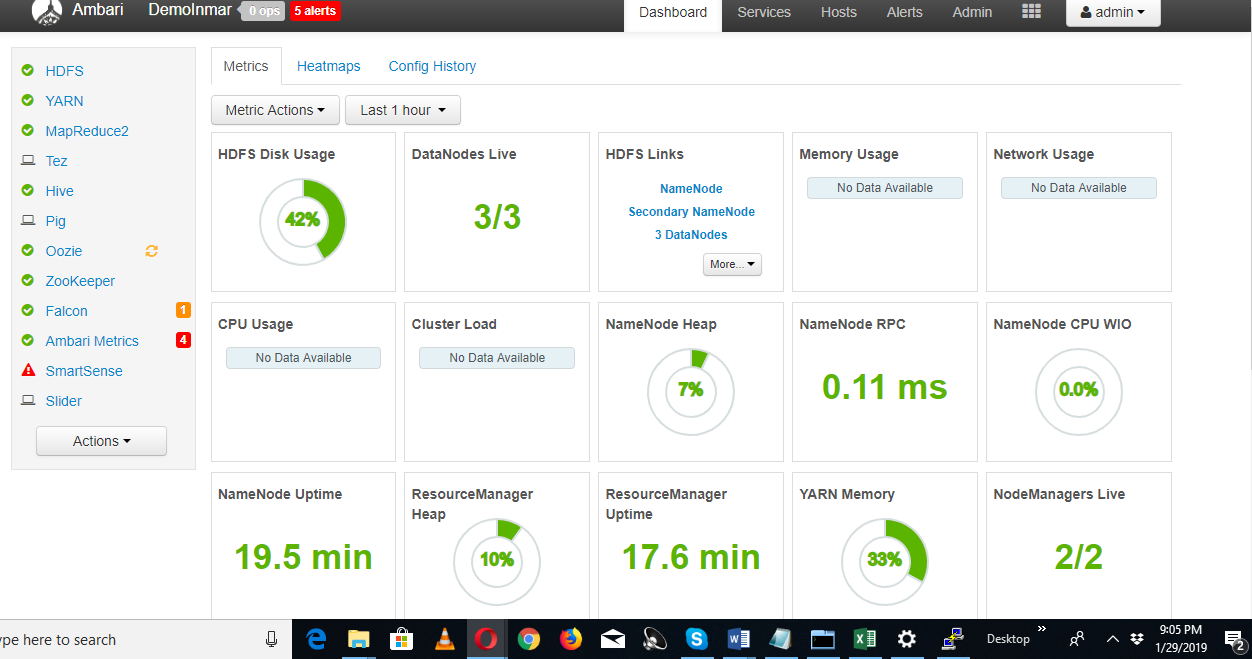


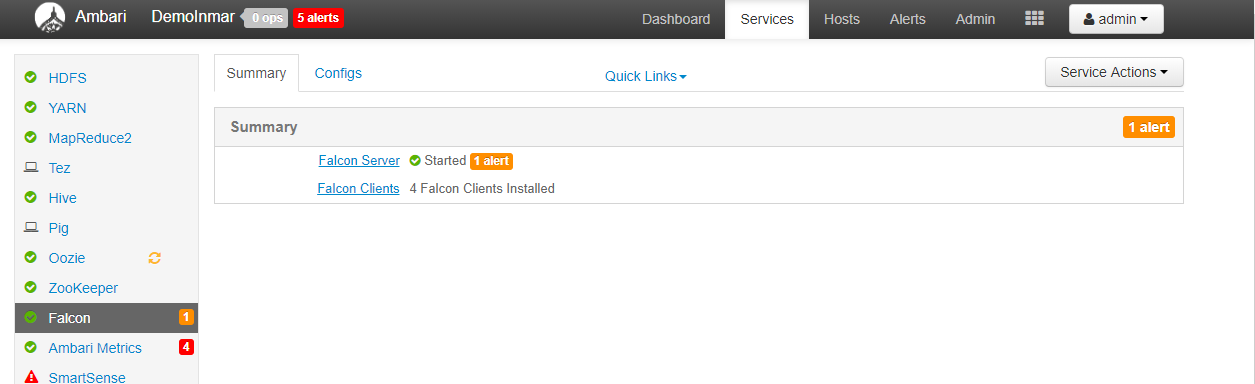
1. 
2. Add the services that need to be installed on the host.
3. Add the host to configuration group.
4. Review the settings and click deploy. Once every thing is good. You will see the host added to Cluster.  
   
5. Run HDFS Rebalnce from Ambari🡪 HDFS🡪 service actions🡪 rebalance HDFS and give the threshold limit. And click OK.
6. Once the rebalance is completed the data will be sync among the Data nodes

**NN Java Heap size change from Ambari:  
1)** Login to Ambari and go to HDFS🡪 settings🡪 click edit and change the name node java heap size and save it. And restart the services that are affected. It has been increased to 2GB from Initial 1GB  


Changing NN Java Heap size from terminal won’t work, even though we changed it. Since the Cluster is being managed by Ambari the values will be overtaken by Ambari Changes.

**Falcon Server Installation:**





Oozie Installation:

RCA from logs:  
DB connection check failed from logs : Access denied to user 'oozie'@'ip-172-31-15-33.us-east-2.compute.internal’ using Password: yes

Resolution login to MYSQL with root and create a user (oozie). And then login back MYSQL with user (oozie) and grant permissions to user and explained in the below article. And restart the oozie.

Issue with MYSQL DB connection issue:  
<https://community.hortonworks.com/questions/203589/oozie-database-access-problem.html>

Sqoop:

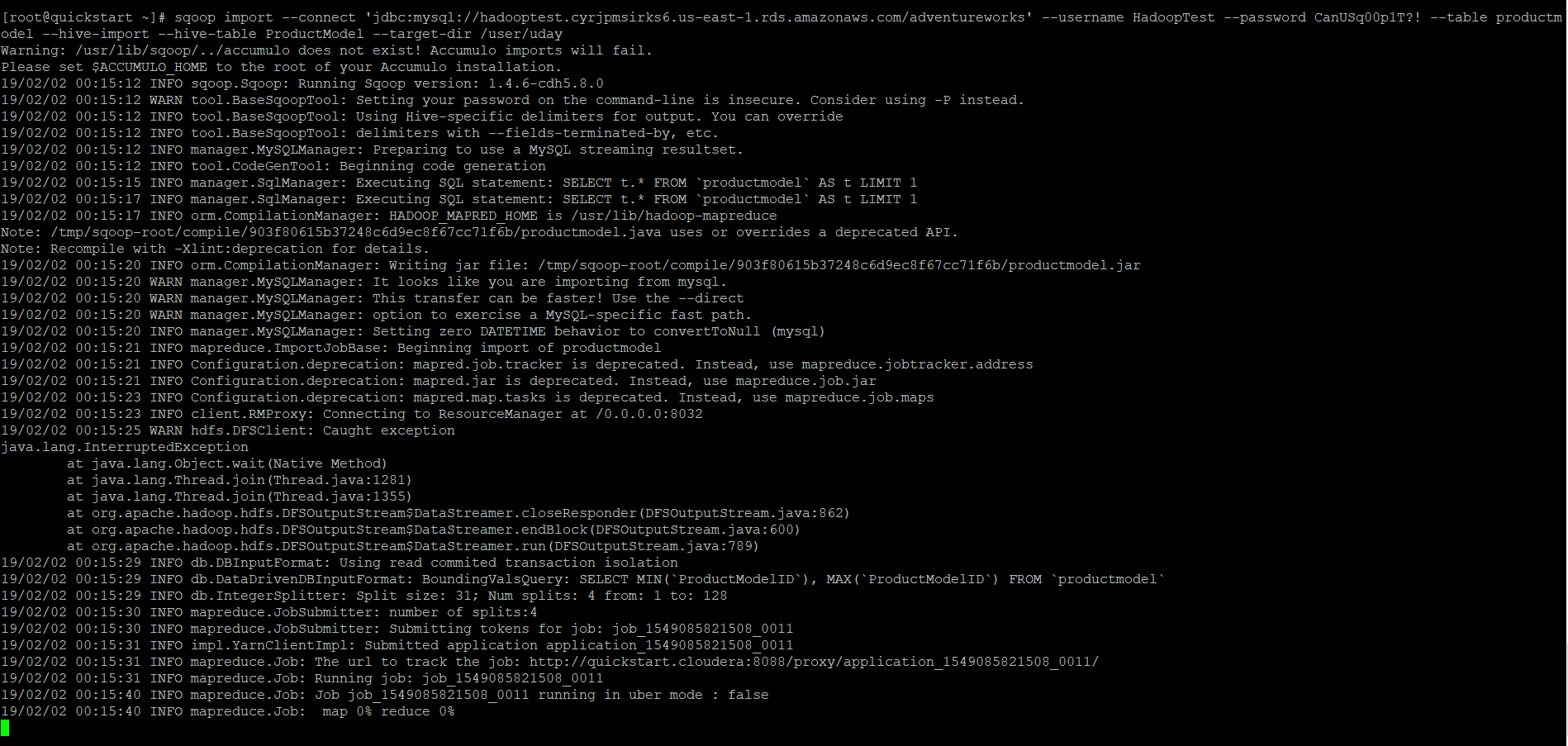
Query to list tables from MYSQL DB:

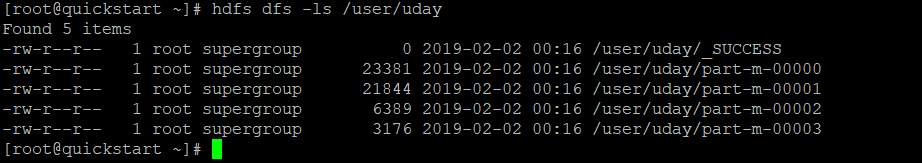
|  |
| --- |
| sqoop list-tables --connect 'jdbc:mysql://hadooptest.cyrjpmsirks6.us-east-1.rds.amazonaws.com/adventureworks' --username HadoopTest --password CanUSq00p1T?! –target-dir /user/ec2-user |

Create Hive DB with name (adventureWorks) to be with same scheme as MYSQL DB.

Query to sqoop a table (ProductModel) from MYSQL DB to HCatalog tables:

|  |
| --- |
| sqoop import --connect 'jdbc:mysql://hadooptest.cyrjpmsirks6.us-east-1.rds.amazonaws.com/adventureworks' --username HadoopTest --password CanUSq00p1T?! --table productmodel --hive-import --hive-table ProductModel --target-dir /user/uday --create-hive-table --hive-table adventureworks.productmodel |
| This Sqoop doen’t work as Hive doesn’t support the SQL type for column rowguid (varbinary) |
|  |
|  |





Workflow.xml:

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <workflow-app xmlns="uri:oozie:workflow:0.2" name="sqoop-wf">  <start to="sqoop-node"/>  <action name="sqoop-node">  <sqoop xmlns="uri:oozie:sqoop-action:0.2">  <job-tracker>localhost:8050</job-tracker>  <name-node>hdfs://localhost</name-node>  <configuration>  <property>  <name>tez.queue.name</name>  <value>default</value>  </property>  <property>  <name>oozie.coord.action.nominal\_time</name>  <value>1527761100000</value>  </property>  <property>  <name>processingQueueName</name>  <value>default</value>  </property>  <property>  <name>mapred.compress.map.output</name>  <value>false</value>  </property>  <property>  <name>mapreduce.framework.name</name>  <value>yarn</value>  </property>  </configuration>  <command>import --connect 'jdbc:mysql://hadooptest.cyrjpmsirks6.us-east-1.rds.amazonaws.com/adventureworks' --username HadoopTest --password CanUSq00p1T?! --table productmodel --hive-import --hive-table ProductModel --target-dir /user/uday</command>  </sqoop>  <ok to="end"/>  <error to="fail"/>  </action>  <kill name="fail">  <message>Sqoop failed, error message[${wf:errorMessage(wf:lastErrorNode())}]</message>  </kill>  <end name="end"/>  </workflow-app> |

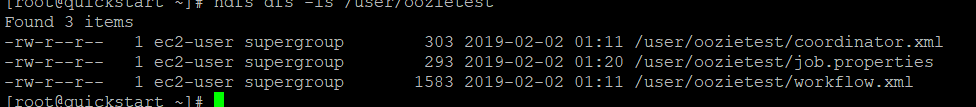
Job.Properties:

|  |
| --- |
| nameNode=localhost  jobTracker=localhost:8050  hdfsRoot=hdfs://${nameNode}/user  #Job Timing  frequency=1440  start=2019-02-01T01:00Z  end=2100-01-01T00:00Z  timeZone=UTC  oozie.coord.application.path=${hdfsRoot}/oozietest  oozie.use.sysem.libpath=true  workflowPath=${oozie.coord.application.path} |

Coordinator.xml:

|  |
| --- |
| <coordinator-app name="oozietest" frequency="${frequency}" start="${start}" end="${end}" timezone="${timeZone}" xmlns="uri:oozie:coordinator:0.1">  <controls> <concurrency>1</concurrency> </controls>  <action>  <workflow>  <app-path>${workflowPath}</app-path>  </workflow>  </action>  </coordinator-app> |

Create a directory under your user name in hdfs to upload your upload items ( workflow/job properties/coordinator.xml)  
command: hdfs dfs –mkdir /user/user name/name of the directory (example: /user/ec2-user/oozietest)  
  
Upload the all the items to Hadoop under directory: /user/your-user-name, using command:  
  
hdfs dfs –put –f /oozietest(local directory)/\* /user/your-user-name

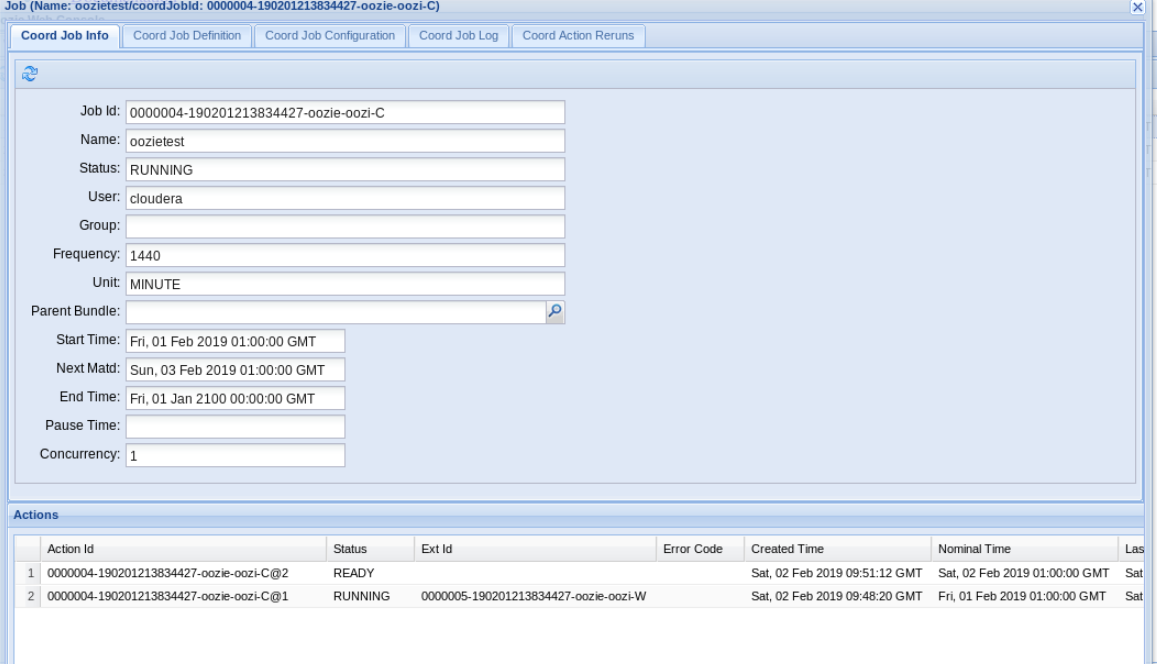


Submit Oozie Job from terminal :

|  |
| --- |
| oozie job –oozie httplocalhost:11000/oozie -config oozietest/job.properties –run |



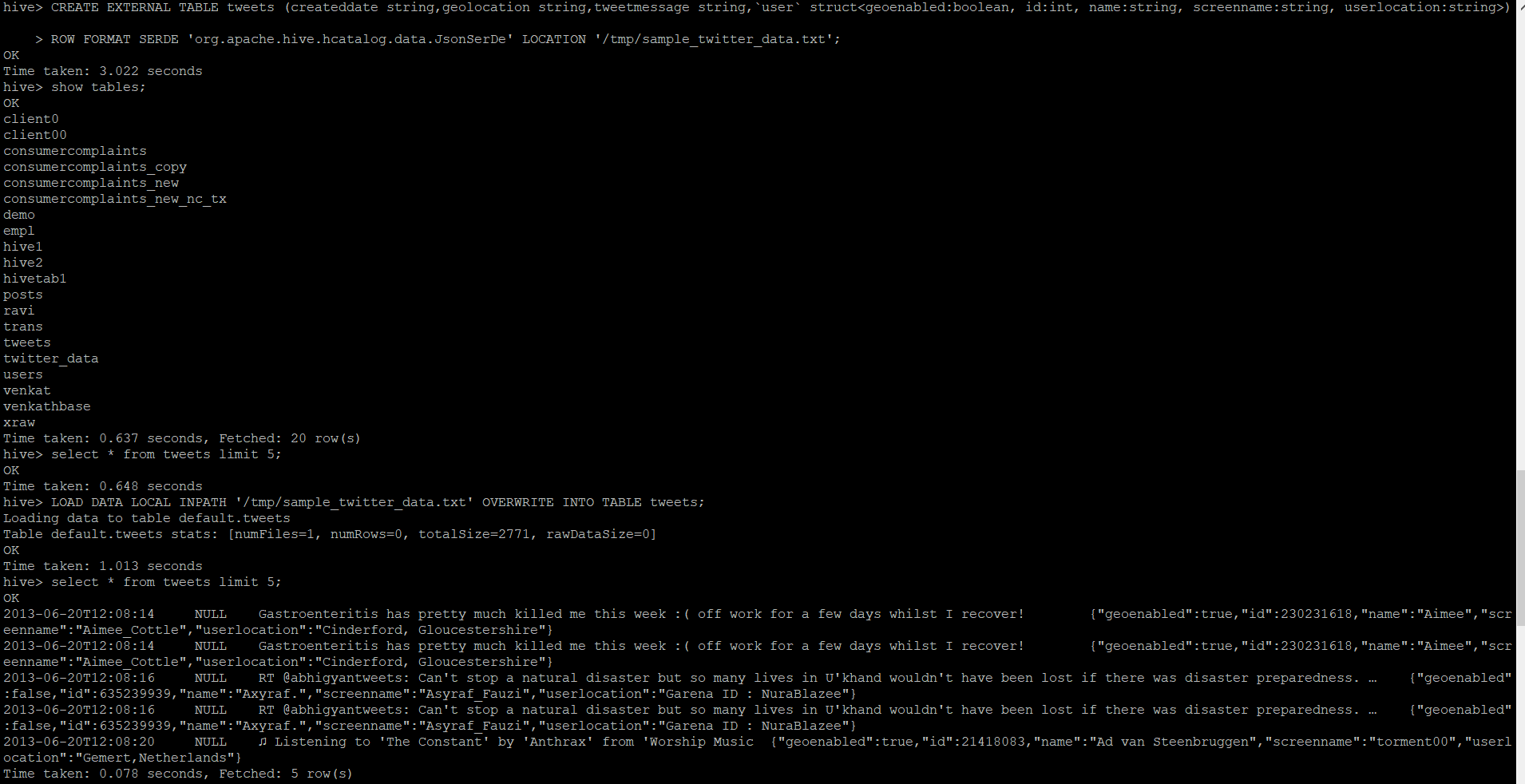
Oozie web UI:



**Task 3:  
Hive**:  
<https://community.hortonworks.com/questions/28684/creating-a-hive-table-with-orgapachehcatalogdatajs.html>

1. Create an external table and load data into it:  
   Query:

|  |
| --- |
| CREATE EXTERNAL TABLE tweets (createddate string,geolocation string,tweetmessage string,`user` struct<geoenabled:boolean, id:int, name:string, screenname:string, userlocation:string>)  ROW FORMAT SERDE 'org.apache.hive.hcatalog.data.JsonSerDe' LOCATION '/tmp/sample\_twitter\_data.txt'; |

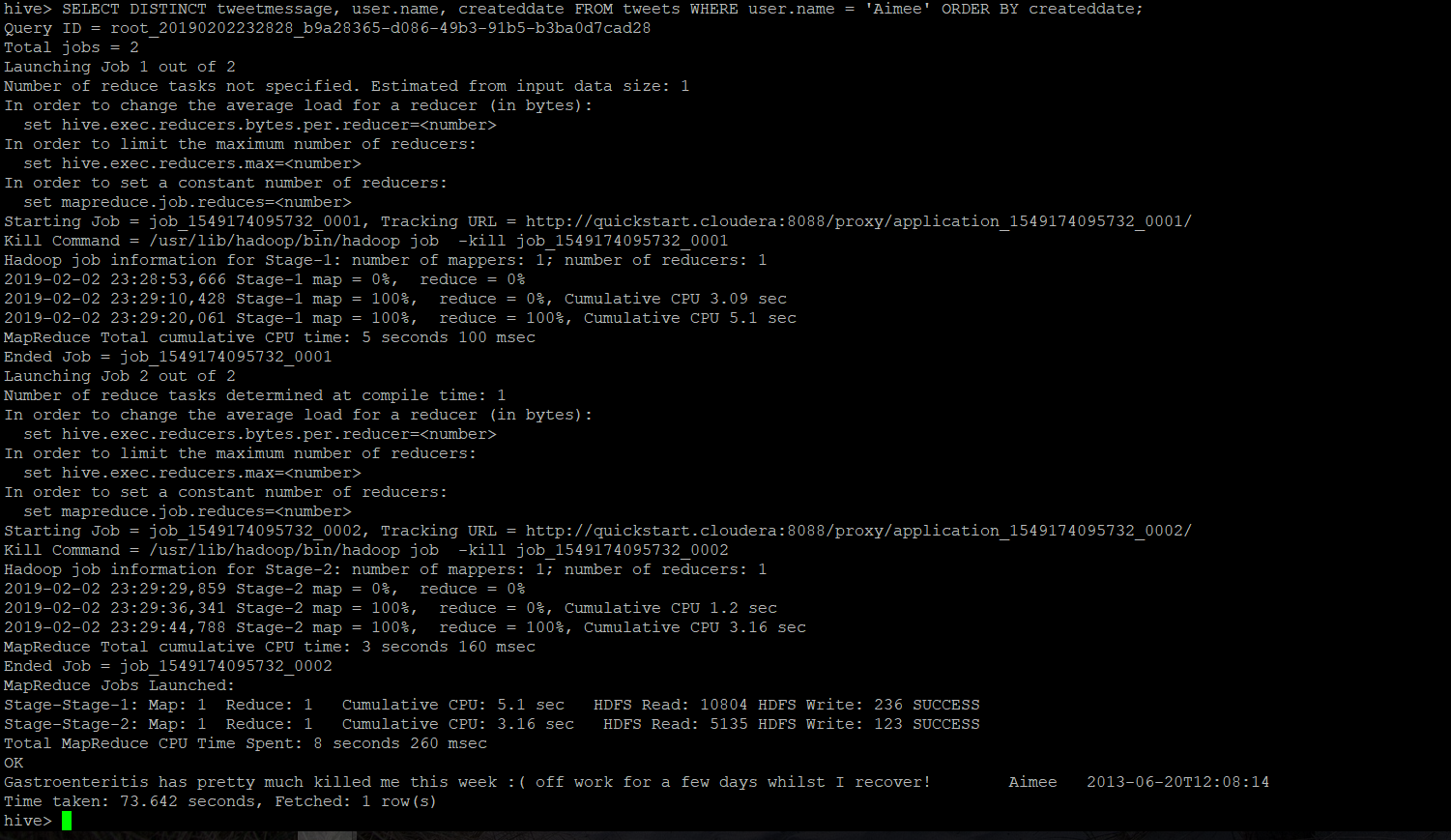


1. Load Data into Tweets tables:  
   Query

|  |
| --- |
| LOAD DATA LOCAL INPATH '/tmp/sample\_twitter\_data.txt' OVERWRITE INTO TABLE tweets; |

1. Execute query in the question:  
   Query:

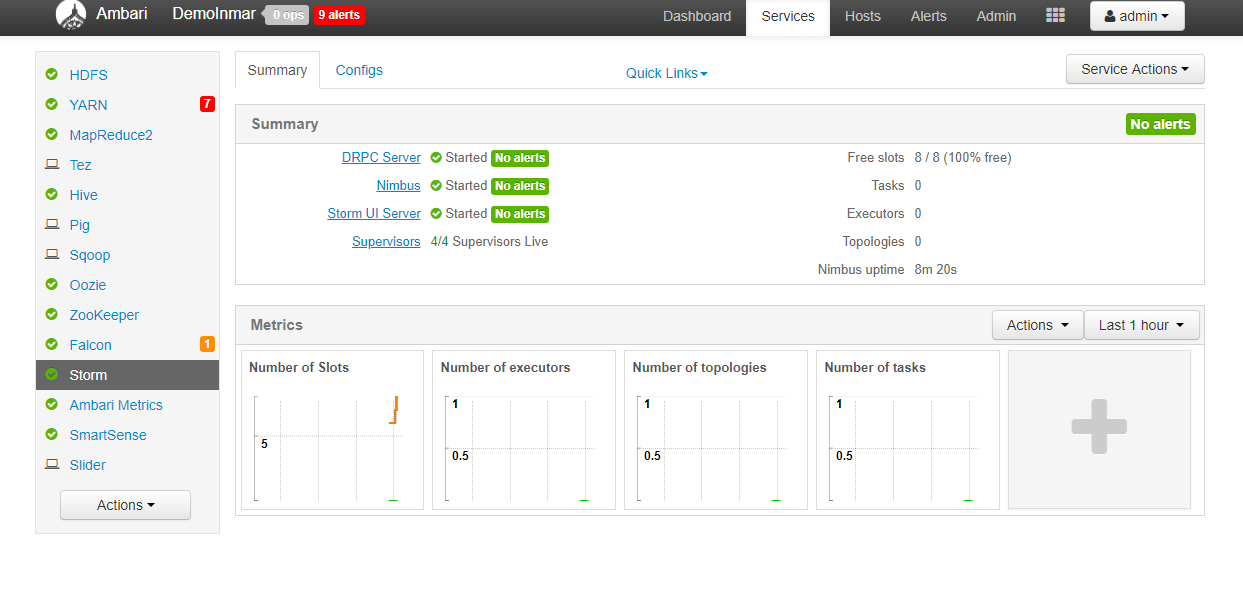
|  |
| --- |
| SELECT DISTINCT tweetmessage, user.name, createddate FROM tweets WHERE user.name = 'Aimee' ORDER BY createddate; |

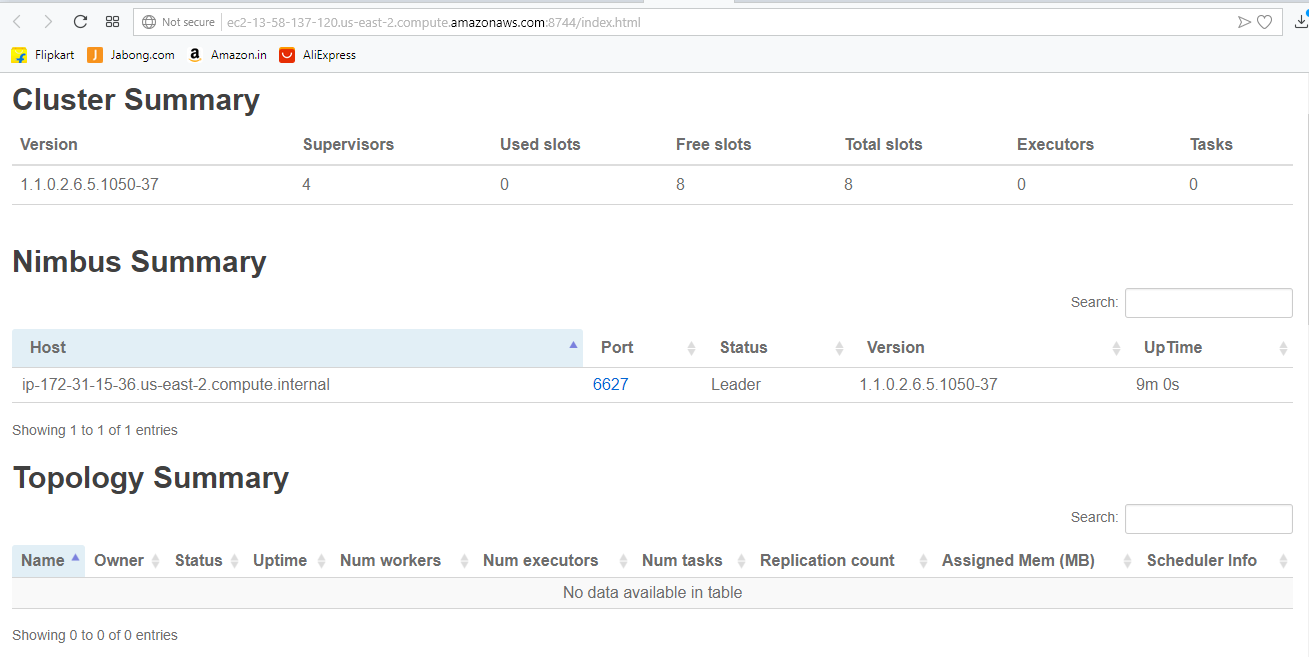


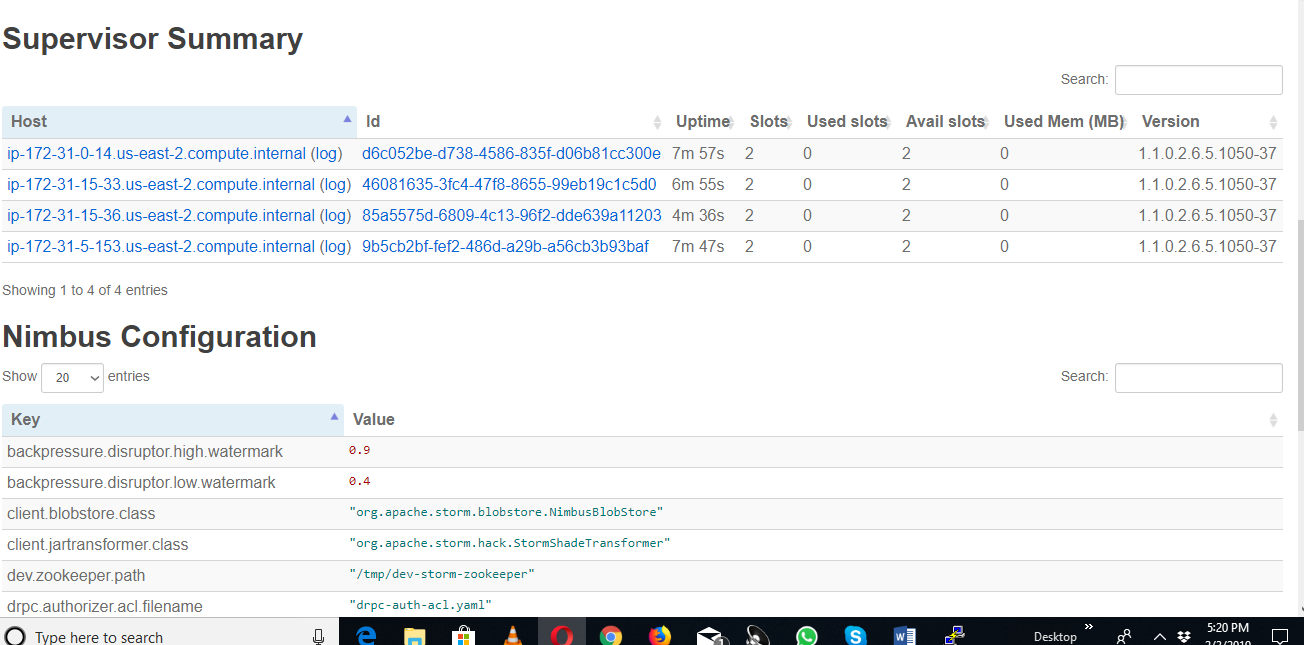
Issue:  
When creating the table, hit with below error:  
Execution Error, return code 1 from org.apache.hadoop.hive.ql.exec.DDLTask. Cannot validate serde: org.apache.hive.hcatalog.data.JsonSerDe site:community.cloudera.com  
Solution:  
Downloaded the hive-hcatalog-core-0.13.0.jar and add it to $HCAT\_HOME class path, and reran the load command.

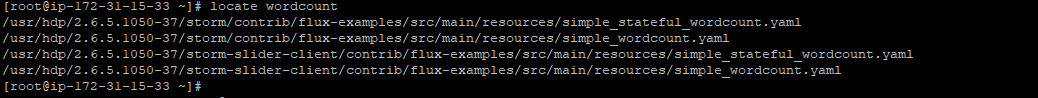
ADD JAR /LOCAL\_PATH\_OF\_JAR;

<https://community.hortonworks.com/content/supportkb/148578/errorcannot-validate-serde-orgapachehivehcatalogda.html>

**Storm:**Installed Storm using Ambari :  


Storm Web UI:  






I don’t have experience in writing a JAR for Storm or creating a topology that run on storm.