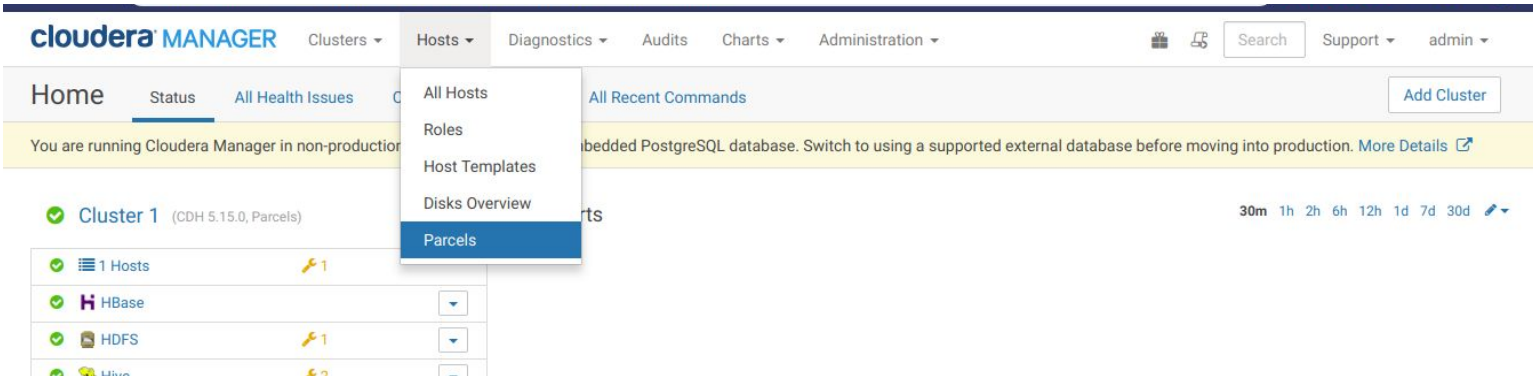
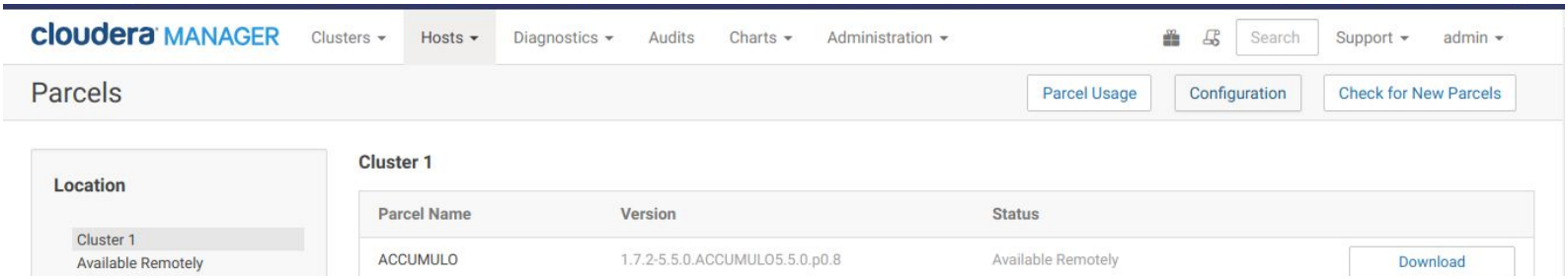
Apache Phoenix Setup Guide

1. Turn on your ec2 instance and log in to the Cloudera manager dashboard. And from the top menu bar, click on **Hosts > Parcels**



2. Click on the top right corner, Click on **configuration.**

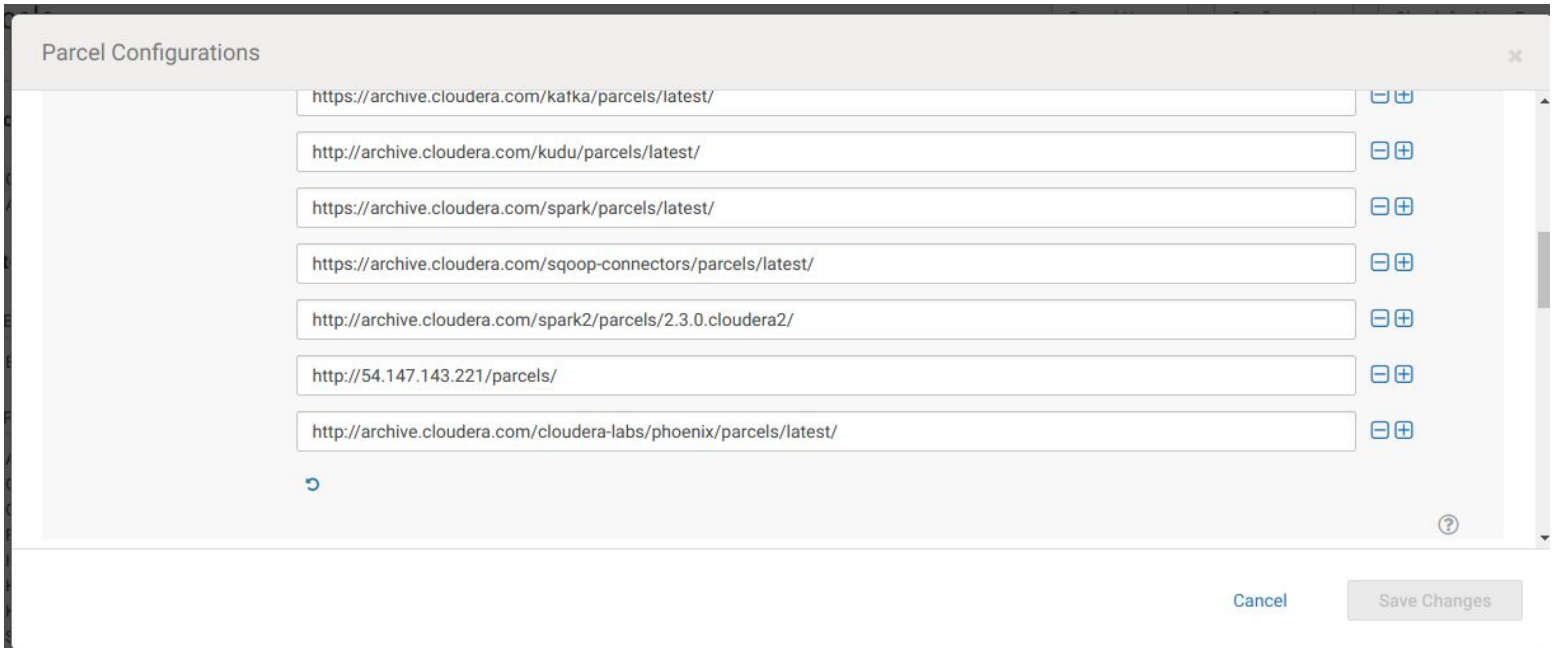


3. Then click on the '+' sign next to an existing Remote Parcel Repository URL,

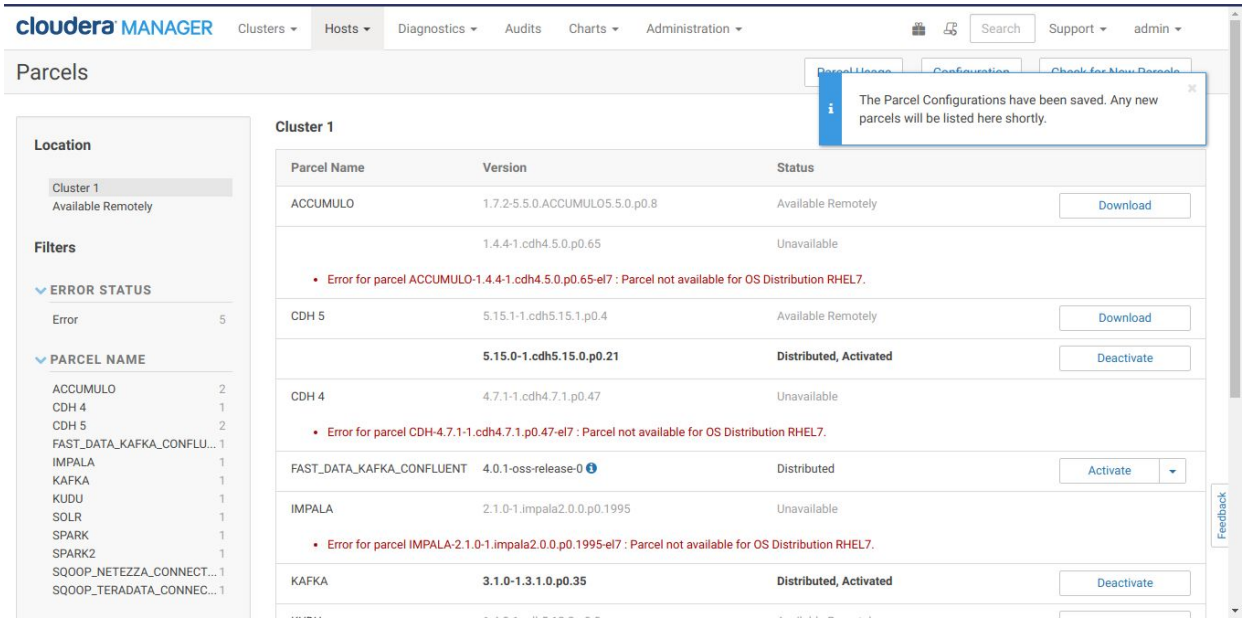
4. A new empty text box will appear, now paste the following URL into it.

```
http://archive.cloudera.com/cloudera-labs/phoenix/parcels/latest/
```

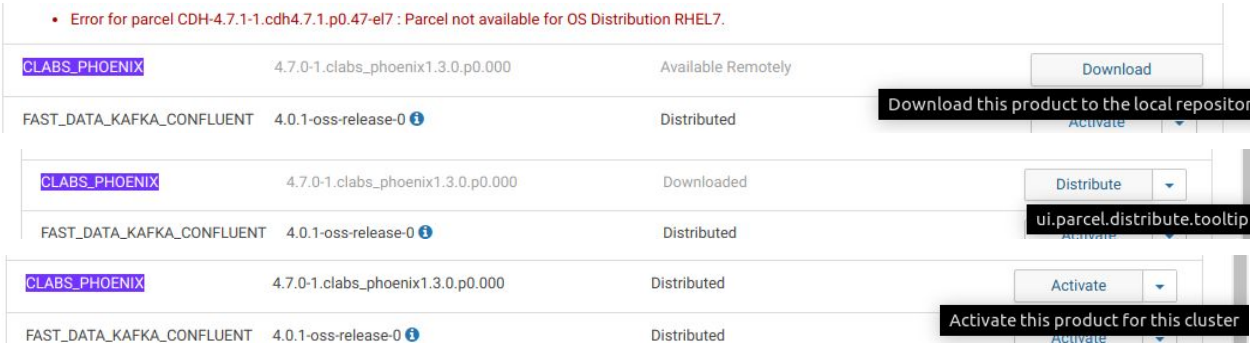
After that click on **Save Changes.**



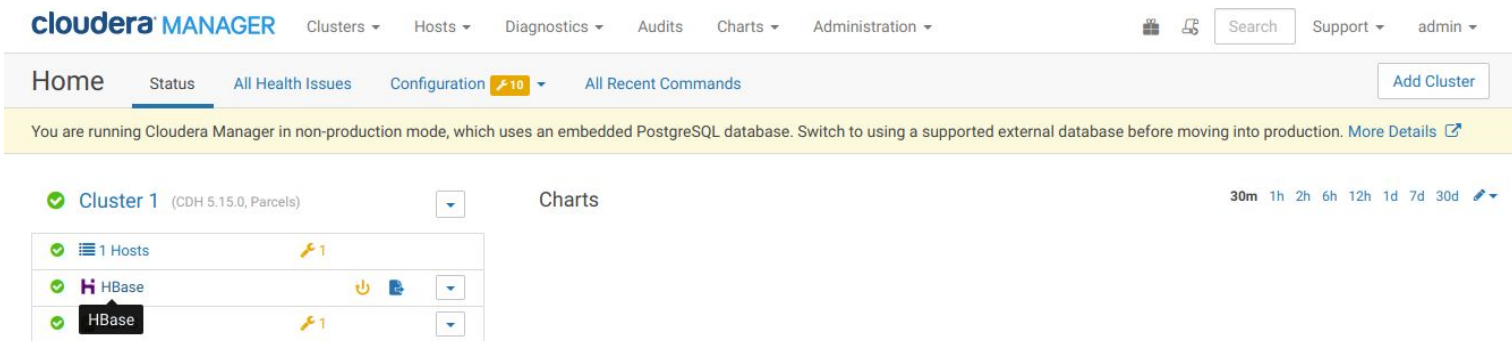
5. Wait for some time until you see a parcel named CLAB PHOENIX, as shown below



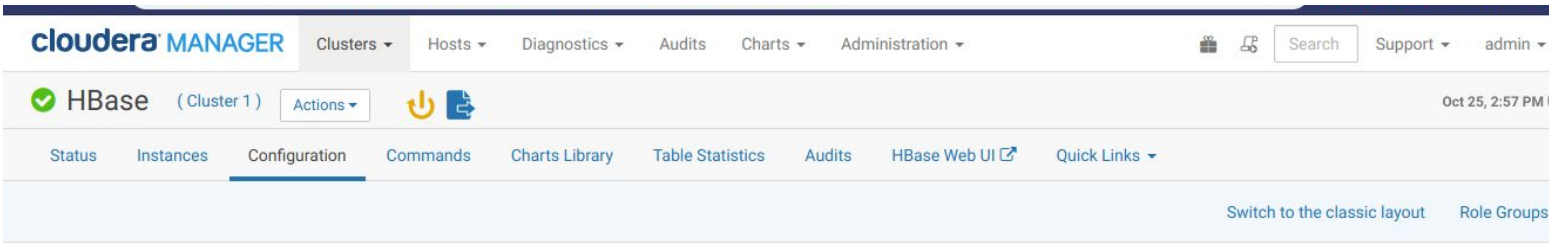
6. Once the parcel is there, click on **Download** then **Distribute** and at last **Activate**, as they appear on the right side of it.



7. Now, to enable, secondary indexing, add the following property to the hbase-site.xml advanced configuration snippet. For that Go to the first go to **Cloudera Manage**r dashboard and click on **HBase** from the services



8. Now from HBase services, click Configuration, from the top menu bar.

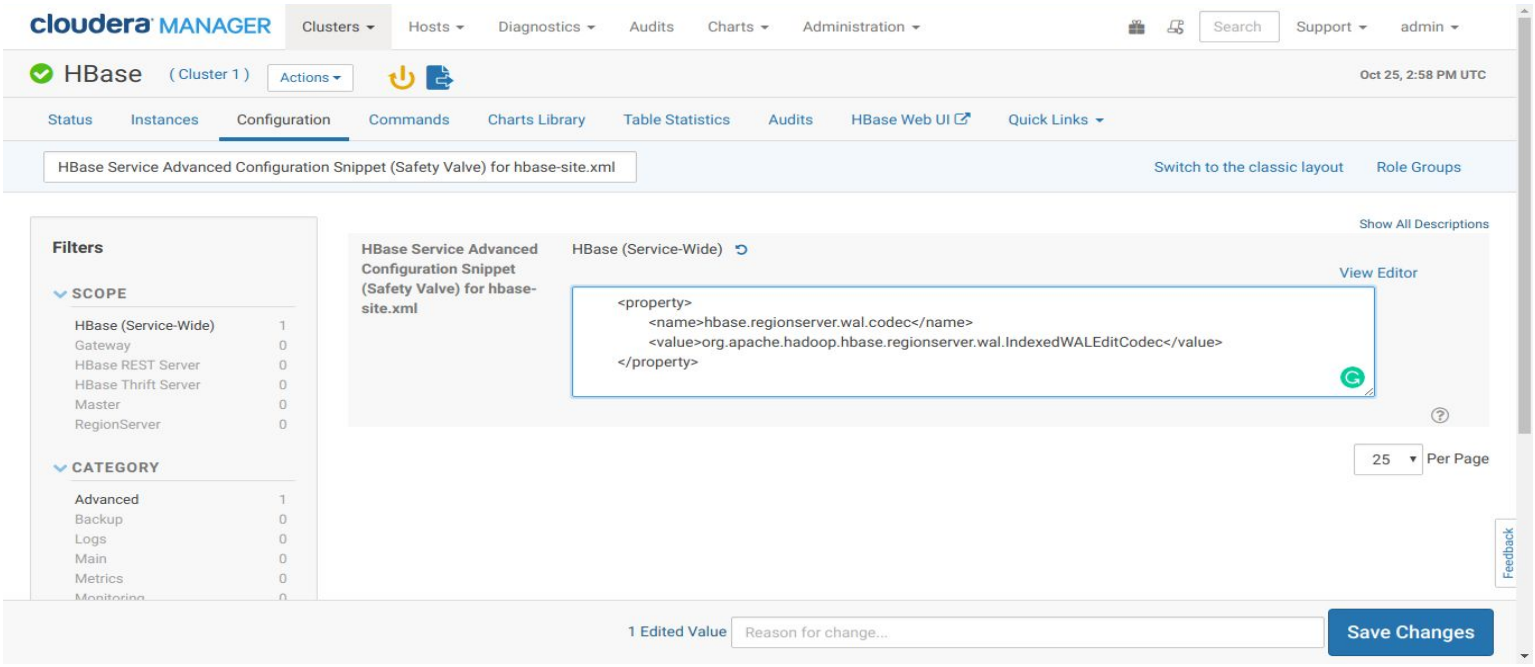


9. Now in the search bar, paste the following file name.

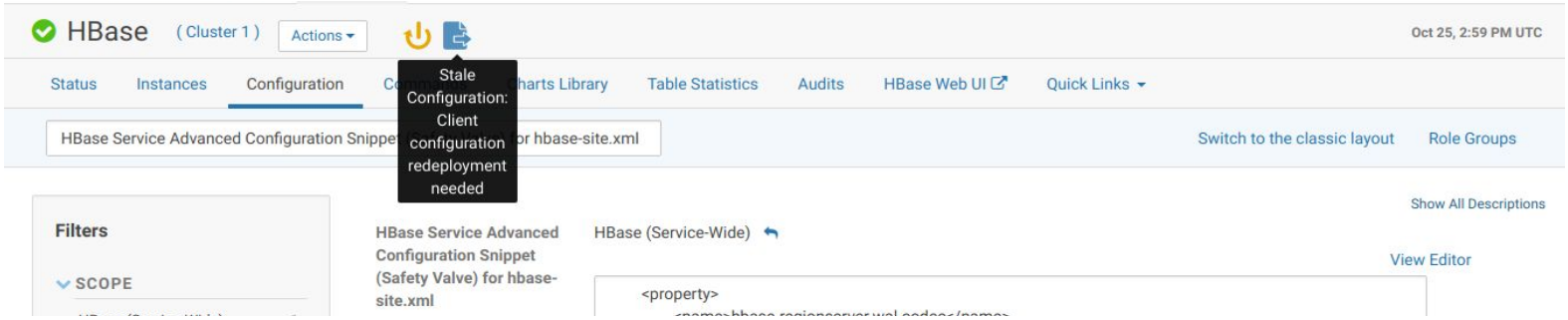
```
HBase Service Advanced Configuration Snippet (Safety Valve) for hbase-site.xml
```

10. Once it is search result are appeared click on **View as xml** and paste the following property to it. As shown below.

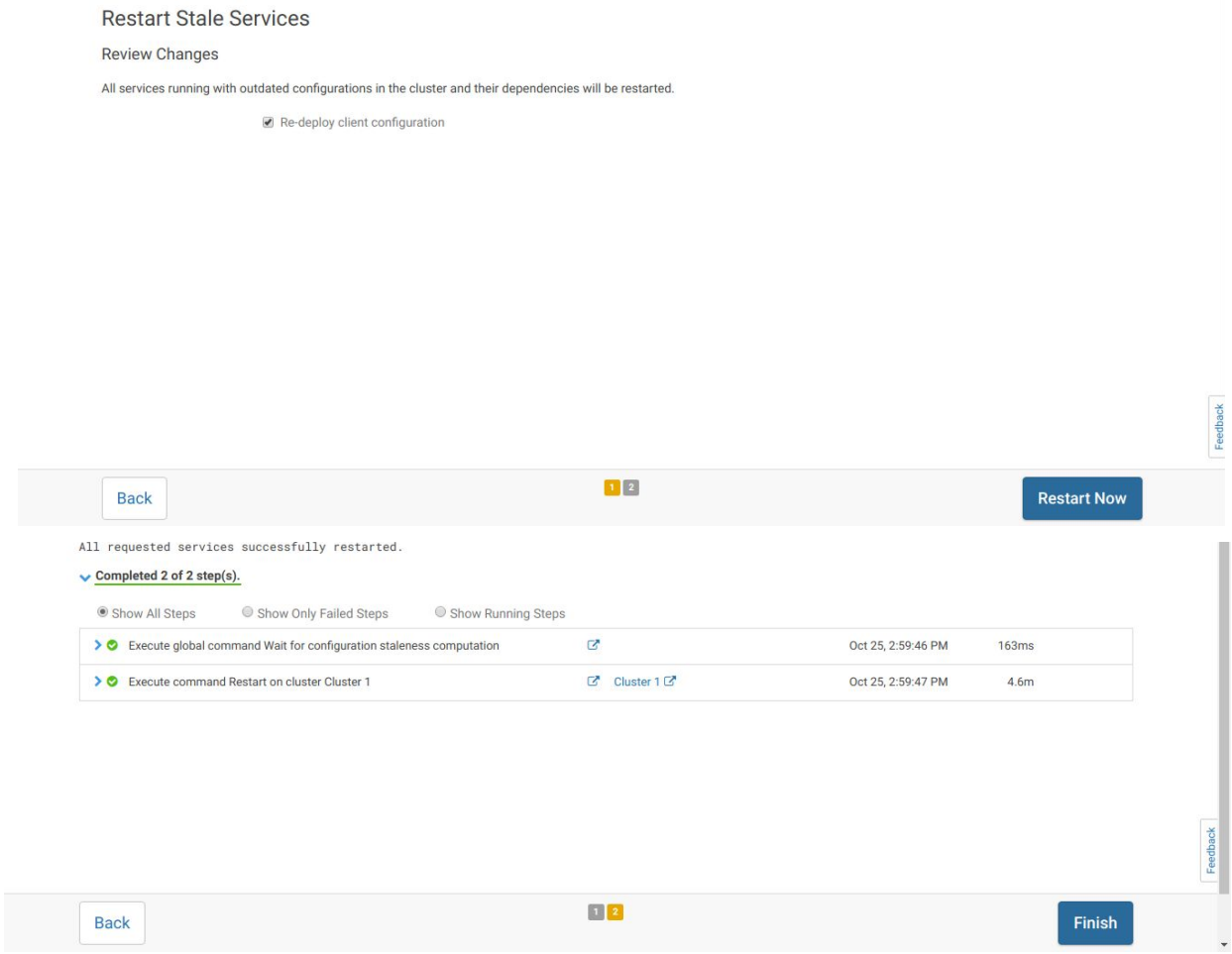
```
<property>
      <name>hbase.regionserver.wal.codec</name>
     <value>org.apache.hadoop.hbase.regionserver.wal.IndexedWALEditCodec</value>
</property>
```



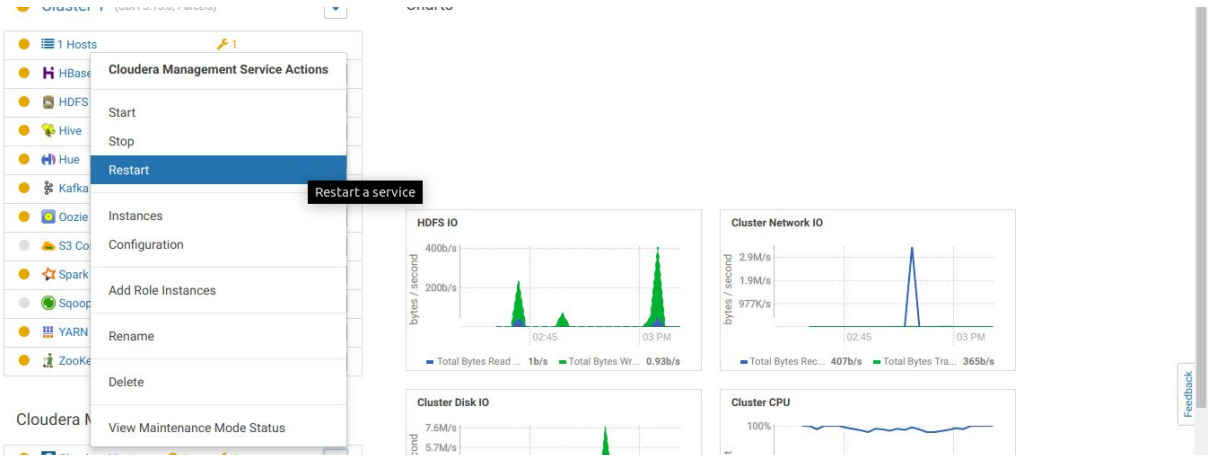
11. Now click on **Save Changes**.
12. Now Click on **Stale Configuration**.

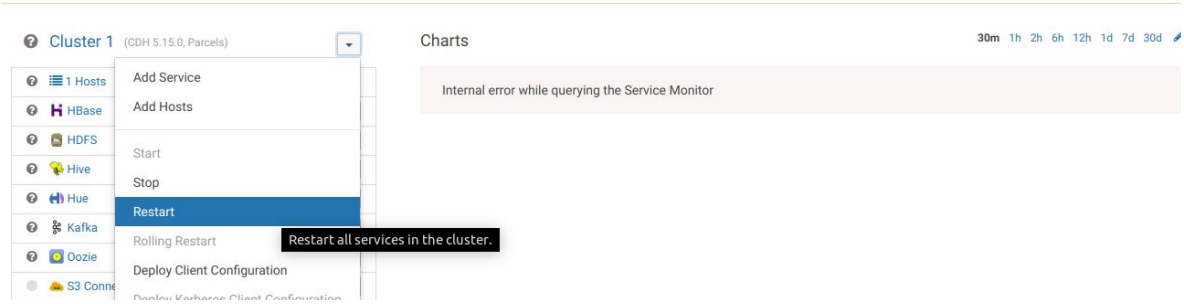


13. Click on **Restart Servies** > **Restart Now** and then **Finish.**



13. Now Restart first **Cloudera Manager** and then **Cluster Services.**





14. Suppress the warnings if any. Nad Restart again Unit all services shows a green tick.

15. Our Phoenix set up is complete, now go to ec2 instance/ Putty termonal and run the following command to open Phoenix shell.

```
phoenix-sqlline.py localhost:2181
```

```
[ec2-user@ip-172-31-91-254 ~]$ phoenix-sqlline.py localhost:2181
Setting property: [incremental, false]
Setting property: [isolation, TRANSACTION_READ_COMMITTED]
issuing: !connect jdbc:phoenix:localhost:2181 none none org.apache.phoenix.jdbc.PhoenixDriver
Connecting to jdbc:phoenix:localhost:2181
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/cloudera/parcels/CLABS_PHOENIX-4.7.0-1.clabs_phoenix1.3.0.p0.000/lib/phoenix/phoenix-4.7.0-clabs-phoenix1.3.0-c
lient.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/opt/cloudera/parcels/CDH-5.15.0-1.cdh5.15.0.p0.21/jars/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.c
lass]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
18/10/25 16:03:16 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
18/10/25 16:03:17 WARN impl.MetricsConfig: Cannot locate configuration: tried hadoop-metrics2-phoenix.properties,hadoop-metrics2.properties
Connected to: Phoenix (version 4.7)
Driver: PhoenixEmbeddedDriver (version 4.7)
Autocommit status: true
Transaction isolation: TRANSACTION_READ_COMMITTED
Building list of tables and columns for tab-completion (set fastconnect to true to skip)...
86/86 (100%) Done
Done
sqlline version 1.1.8
0: jdbc:phoenix:localhost:2181>
```