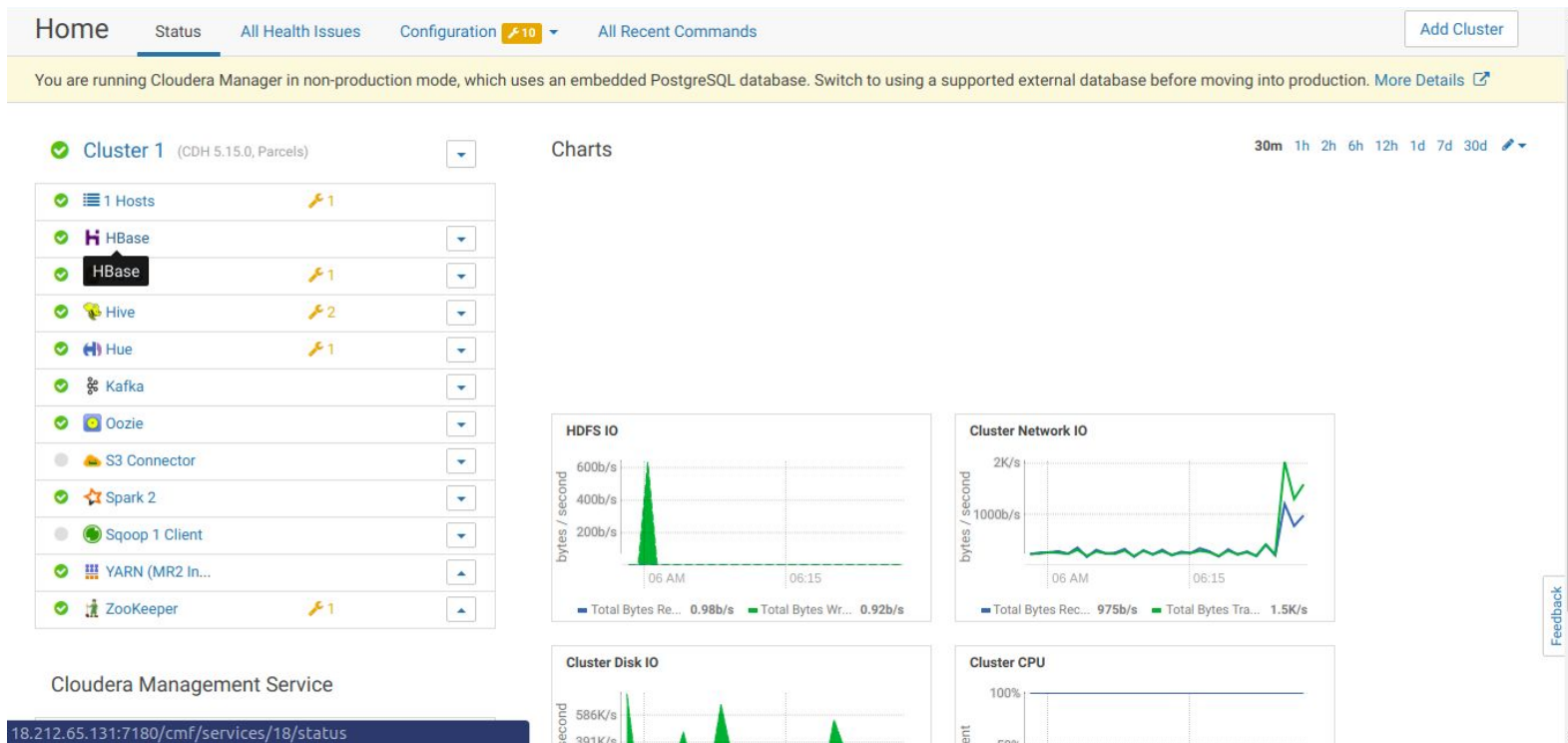


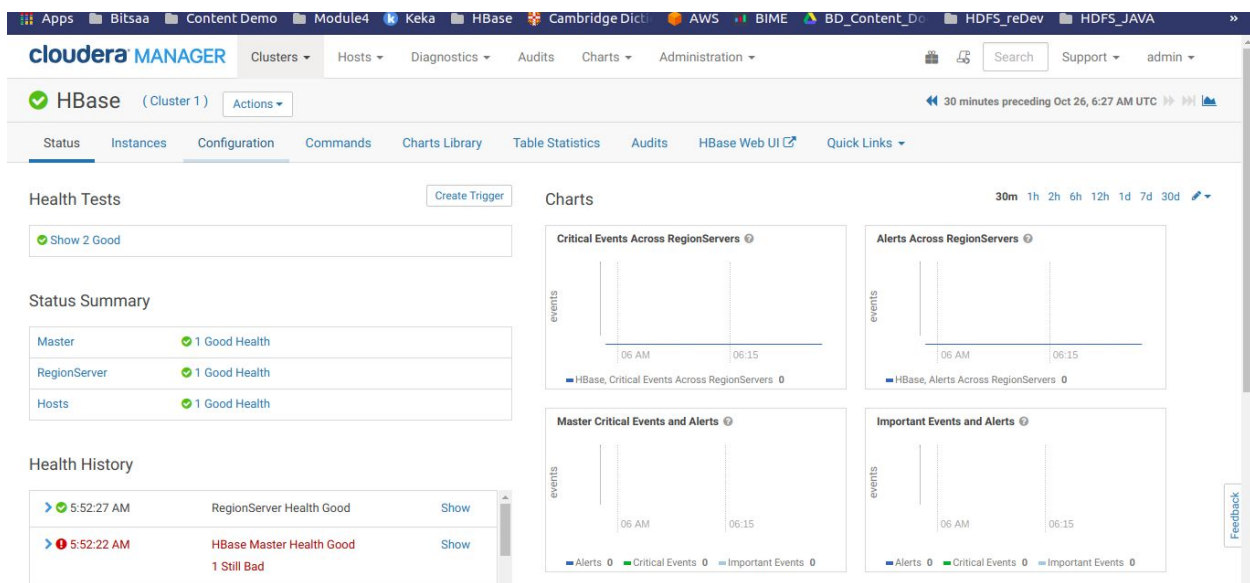
# ACL Configuration in HBase

## 1. Open **Cloudera Manager** and Click on **HBase**



The screenshot shows the Cloudera Manager Home page. The top navigation bar includes 'Home', 'Status', 'All Health Issues', 'Configuration' (highlighted with a pencil icon), and 'All Recent Commands'. A yellow banner at the top states: 'You are running Cloudera Manager in non-production mode, which uses an embedded PostgreSQL database. Switch to using a supported external database before moving into production. More Details'. Below this, the 'Cluster 1' (CDH 5.15.0, Parcels) is selected. On the left, a list of services is shown, with 'HBase' highlighted. On the right, the 'Charts' section displays four graphs: 'HDFS IO' (showing bytes/second), 'Cluster Network IO' (showing bytes/second), 'Cluster Disk IO' (showing bytes/second), and 'Cluster CPU' (showing percentage). The bottom of the page shows the Cloudera Management Service URL: 18.212.65.131:7180/cm/svc/status.

## 2. Click on **Configurations** and from the search bar, Search for following properties-



The screenshot shows the Cloudera Manager HBase configuration page. The top navigation bar includes 'Apps', 'Clusters', 'Hosts', 'Diagnostics', 'Audits', 'Charts', and 'Administration'. The 'HBase' service is selected. The 'Configuration' tab is active. The page displays 'Health Tests' (Show 2 Good), 'Status Summary' (Master: 1 Good Health, RegionServer: 1 Good Health, Hosts: 1 Good Health), and 'Health History' (5:52:27 AM: RegionServer Health Good, 5:52:22 AM: HBase Master Health Good, 1 Still Bad). The 'Charts' section shows four graphs: 'Critical Events Across RegionServers', 'Alerts Across RegionServers', 'Master Critical Events and Alerts', and 'Important Events and Alerts'. The bottom of the page shows the Cloudera Management Service URL: 18.212.65.131:7180/cm/svc/status.



a. Search **hbase.superuser** and put the value **hbase**, click on **Save Changes**

The screenshot shows the HBase Configuration page for Cluster 1. The search bar at the top contains 'hbase superuser'. The left sidebar shows filters for SCOPE and CATEGORY. The main content area displays the configuration for 'hbase.superuser' with the value 'hbase' entered in the 'HBase (Service-Wide)' field. The 'HBase User to Impersonate' field is empty. The bottom of the page shows a '1 Edited Value' summary and a 'Save Changes' button.

Filter	Count
<b>SCOPE</b>	
HBase (Service-Wide)	2
Gateway	0
HBase REST Server	0
HBase Thrift Server	0
Master	0
RegionServer	0
<b>CATEGORY</b>	
Advanced	1
Backup	0
Logs	0
Main	1

**HBase Superusers**  
hbase.superuser

**HBase (Service-Wide)**  
hbase

**HBase User to Impersonate**  
HBase (Service-Wide)

1 Edited Value Reason for change...

**Save Changes**

B. search an **hbbase.coprocessor.region.classes** and put value as **org.apache.hadoop.hbase.security.access.AccessController**, click on **Save Changes**



**cloudera MANAGER** Clusters Hosts Diagnostics Audits Charts Administration

Search Support admin

Oct 26, 6:31 AM UTC

Configuration

Status Instances Configuration Commands Charts Library Table Statistics Audits HBase Web UI Quick Links

hbase.coprocessor.region.classes

Switch to the classic layout Role Groups

Filters

SCOPE

HBase (Service-Wide)	0
Gateway	0
HBase REST Server	0
HBase Thrift Server	0
Master	0
RegionServer	1

CATEGORY

Advanced	1
Backup	0
Logs	0
Main	0

HBase Coprocessor Region Classes

RegionServer Default Group

hbase.coprocessor.region.classes

org.apache.hadoop.hbase.security.access.AccessController

25 Per Page

Save Changes

C. search **hbase.coprocessor.master.classes** and put the value as **org.apache.hadoop.hbase.security.access.AccessController** then **Save Changes**



✓ HBase (Cluster 1)

Actions

Oct 26, 6:32 AM UTC

Status

Instances

Configuration

Commands

Charts Library

Table Statistics

Audits

HBase Web UI

Quick Links

hbase.coprocessor.master.classes

Switch to the classic layout

Role Groups

Filters

SCOPE

Category

Advanced

Backup

Logs

Main

HBase Coprocessor Master Classes

hbase.coprocessor.master.class

Master Default Group

org.apache.hadoop.hbase.security.access.AccessController

25

Per Page

1 Edited Value

Reason for change...

Save Changes

D. Search **Hbase.security.authorization**, tick the checkbox and **Save Changes**

cloudera MANAGER

Clusters

Hosts

Diagnostics

Audits

Charts

Administration

Search

Support

admin

✓ HBase (Cluster 1)

Actions

Oct 26, 6:34 AM UTC

Status

Instances

Configuration

Commands

Charts Library

Table Statistics

Audits

HBase Web UI

Quick Links

hbase.security.authorization

Switch to the classic layout

Role Groups

Filters

SCOPE

Category

Advanced

Backup

Logs

Main

HBase Secure Authorization

hbase.security.authorization

☒ HBase (Service-Wide)

25

Per Page

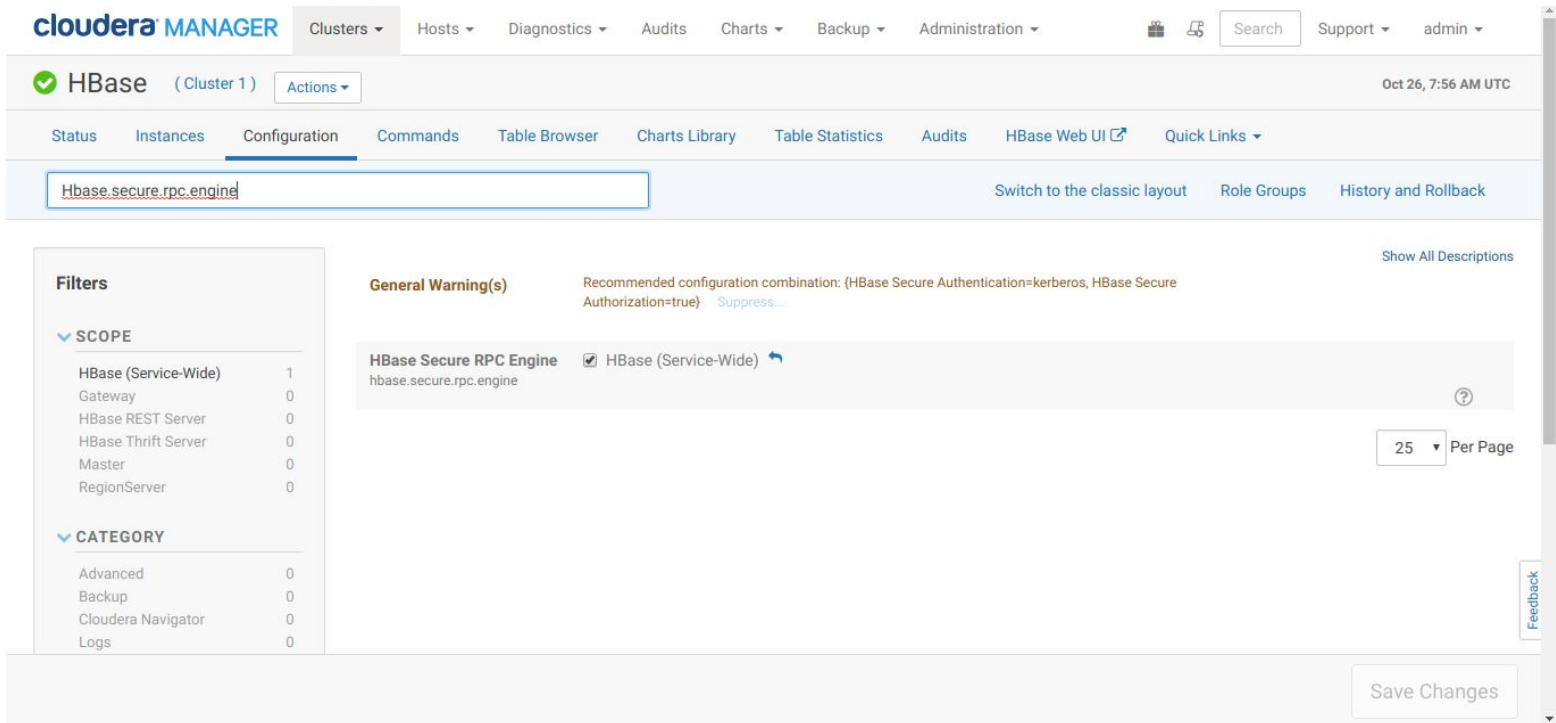
1 Edited Value

Reason for change...

Save Changes

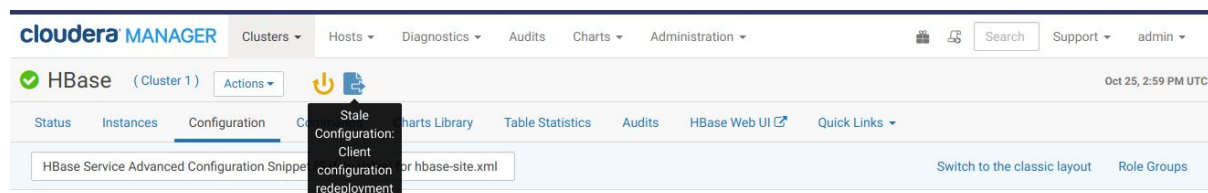
E. search **hbase.secure.rpc.engine** tick the checkbox then **Save Changes**.

**Note: if this property doesn't appear on your system, Ignore it**



The screenshot shows the Cloudera Manager interface for the HBase configuration of Cluster 1. The 'Configuration' tab is selected, and a search filter 'Hbase.secure.rpc.engine' is applied. The configuration table shows the 'HBase Secure RPC Engine' property for 'hbase.secure.rpc.engine' is set to 'HBase (Service-Wide)'. A 'General Warning(s)' section indicates a recommended configuration combination. The 'Filters' sidebar on the left shows the scope and category of the configuration. The 'Save Changes' button is visible at the bottom right.

3. Click in **Stale Configuration > Restart Now > Finish**



This screenshot shows the same Cloudera Manager HBase configuration page, but with the 'Stale Configuration' menu open. The menu options are 'Client configuration redeployment' and 'or hbase-site.xml'. The 'Restart Now' button is also visible in the top right corner of the configuration table.

4. Restart your Cloudera Manager and Cluster services, if Needed.

Now your Security setup is ready Now, let's see how to use the **grant**, **revoke** and **user\_permission** command in HBase. The first we set up, we actually secured the superUser scope. We gave '**hbase**' user the superuser rights, Note that the **superUser** above all scopes.

Open hbase shell from as **ec2-user**, and enter **list** command, as you can see below you can't see any table(if there was any before), that is because now you have the security parameters, hence by default, the **ec2-user** doesn't have any kind of rights.

```
[ec2-user@ip-172-31-91-254 ~]$ hbase shell
Java HotSpot(TM) 64-Bit Server VM warning: Using incremental CMS is deprecated and will likely be removed in a future release
18/10/26 07:45:25 INFO Configuration.deprecation: hadoop.native.lib is deprecated. Instead, use io.native.lib.available
HBase Shell; enter 'help<RETURN>' for list of supported commands.
Type "exit<RETURN>" to leave the HBase Shell
Version 1.2.0-cdh5.15.0, rUnknown, Thu May 24 04:27:47 PDT 2018

hbase(main):001:0>
```

```
hbase(main):001:0> list
TABLE
0 row(s) in 0.9190 seconds
=> []
hbase(main):002:0>
```

Now let's grant some permission to ec2-user for that we need to open hbase shell as the superuser, which is **hbase**. For that exit from this shell.

To open hbase shell as **hbase** superuser, enter the following command

```
sudo -u hbase hbase shell
```

and now if you do a list, you can see all your tables, because the superuser by default has all the permissions.



```
[ec2-user@ip-172-31-91-254 ~]$ sudo -u hbase hbase shell
18/10/26 07:41:03 INFO Configuration.deprecation: hadoop.native.lib is deprecated. Instead, use io.native.lib.available
HBase Shell; enter 'help<RETURN>' for list of supported commands.
Type "exit<RETURN>" to leave the HBase Shell
Version 1.2.0-cdh5.15.0, rUnknown, Thu May 24 04:27:47 PDT 2018

hbase(main):001:0> list
TABLE
SYSTEM.CATALOG
SYSTEM.FUNCTION
SYSTEM.SEQUENCE
SYSTEM.STATS
USER_DETAILS
t1
t11
t2
table1
9 row(s) in 1.6660 seconds

=> ["SYSTEM.CATALOG", "SYSTEM.FUNCTION", "SYSTEM.SEQUENCE", "SYSTEM.STATS", "USER_DETAILS", "t1", "t11", "t2", "table1"]
hbase(main):002:0> █
```

Now let's grant read, write, create, and admin(**RWCA**) permission to the **ec2-user**

```
hbase(main):002:0> grant 'ec2-user','RWCA'
0 row(s) in 1.0740 seconds

hbase(main):003:0> █
```

Now exit and open hbase shell as **ec2-user**

```
[ec2-user@ip-172-31-91-254 ~]$ hbase shell
Java HotSpot(TM) 64-Bit Server VM warning: Using incremental CMS is deprecated and will likely be removed in a future release
18/10/26 07:45:25 INFO Configuration.deprecation: hadoop.native.lib is deprecated. Instead, use io.native.lib.available
HBase Shell; enter 'help<RETURN>' for list of supported commands.
Type "exit<RETURN>" to leave the HBase Shell
Version 1.2.0-cdh5.15.0, rUnknown, Thu May 24 04:27:47 PDT 2018

hbase(main):001:0> █
```

Now **list** all your table,

```
hbase(main):001:0> list
TABLE
SYSTEM.CATALOG
SYSTEM.FUNCTION
SYSTEM.SEQUENCE
SYSTEM.STATS
USER_DETAILS
t1
t11
t2
table1
9 row(s) in 1.6980 seconds

=> ["SYSTEM.CATALOG", "SYSTEM.FUNCTION", "SYSTEM.SEQUENCE", "SYSTEM.STATS", "USER_DETAILS", "t1", "t11", "t2", "table1"]
hbase(main):002:0> █
```

Let's create a table named 'ec2', Put a value in it

```
hbase(main):002:0> create 'ec2','cf1'
0 row(s) in 5.1420 seconds

=> Hbase::Table - ec2
hbase(main):003:0> put 'ec2','row1','cf1:c1','rawat'
0 row(s) in 0.5760 seconds

hbase(main):004:0> █
```

Let's grant the **root** user only the **read** permission for the 'ec2' table

```
hbase(main):005:0> grant 'root','R', 'ec2'
0 row(s) in 1.3430 seconds

hbase(main):006:0> █
```

You can see all the permission granted for a table using user\_permission,

```
hbase(main):002:0> user_permission 'ec2'
User                               Namespace,Table,Family,Qualifier:Permission
ec2-user                           default,ec2,, [Permission: actions=READ,WRITE,EXEC,CREATE,ADMIN]
root                               default,ec2,, [Permission: actions=READ]
2 row(s) in 2.1020 seconds

hbase(main):003:0> █
```

Let's verify it from the root user, From this shell, and open hbase shell from **root** user

```
[ec2-user@ip-172-31-91-254 ~]$ sudo -i
[root@ip-172-31-91-254 ~]# hbase shell
```



Do a **list**, you can and see the 'ec2' table

```
hbase(main):001:0> list
TABLE
ec2
1 row(s) in 0.9750 seconds

=> ["ec2"]
hbase(main):002:0> █
```

Let's read the 'ec2' table

```
hbase(main):001:0> scan 'ec2'
ROW
row1
1 row(s) in 1.1200 seconds
hbase(main):002:0> █
```

COLUMN+CELL
column=cf1:c1, timestamp=1540540149068, value=rawat

Now, let's write a value in it,

```
hbase(main):003:0> put 'ec2','row2','cf1:c2','abhinav'
ERROR: Failed 1 action: org.apache.hadoop.hbase.security.AccessDeniedException: Insufficient permissions (user=root, scope=default:ec2, family=cf1:c2, params=[table=default:ec2,family=cf1:c2],action=WRITE)
```

Because as ec2-user we only granted read permission for this 'ec2' table, the root can only read it and can't write any value.

To revoke the permissions of the root user, go back and again open hbase shell from **ec2-user**. Following command is used to revoke all the permission of the root user from the 'ec2' table

```
revoke 'root','ec2'
```

```
hbase(main):004:0> revoke 'root','ec2'
0 row(s) in 0.4920 seconds

hbase(main):005:0> █
```

As you can see below, the **read** permission of the **root** user has been revoked.

```
[ec2-user@ip-172-31-91-254 ~]$ sudo -i
[root@ip-172-31-91-254 ~]# hbase shell
Java HotSpot(TM) 64-Bit Server VM warning: Using incremental CMS is deprecated and will likely be removed in a future release
18/10/26 08:08:36 INFO Configuration.deprecation: hadoop.native.lib is deprecated. Instead, use io.native.lib.available
HBase Shell; enter 'help<RETURN>' for list of supported commands.
Type "exit<RETURN>" to leave the HBase Shell
Version 1.2.0-cdh5.15.0, rUnknown, Thu May 24 04:27:47 PDT 2018

hbase(main):001:0> list
TABLE
0 row(s) in 0.9190 seconds
=> []
hbase(main):002:0> █
```