Convert Hue Backend to PostgreSQL

[Create Hue Database Instance](#h.3oyx9mzdpa8i)

[Validate Existing Sqlite3 Users](#h.tlqtg9576bdq)

[Shutdown Hue](#h.2sxkmzrtrrfl)

[Dump Sqlite3 DB](#h.efxijsh3jnv)

[Check Database Host Dumped](#h.gajf3ii5lz1o)

[Copy Sqlite3 Database to Dump Server if Necessary](#h.xfbzs03h6vgx)

[Edit Dump File](#h.4p68xv8xhm6y)

[Modify Current Hue.ini](#h.5gpnroysq4fo)

[Validate Changes](#h.yxapiqnqra31)

The default DB for Hue is the embedded Sqlite3 DB. This document provides instructions on converting this to use PostgreSQL as the backend database.

Instructions for the process can be found at:

<http://www.cloudera.com/documentation/enterprise/5-5-x/topics/cm_mc_hue_service.html#cmig_topic_15_2_unique_1>

# Create Hue Database Instance

On the PostgreSQL server, create the hue database and grant privileges to a hue user to manage the database.

|  |
| --- |
| [root@bos-rd1-cdh-master1 ~]# **su - postgres**  Last login: Tue Mar 22 16:45:26 EDT 2016 on pts/0  -bash-4.2$ **psql**  psql (9.4.5)  Type "help" for help.  postgres=# **create database hue;**  CREATE DATABASE  postgres=# **\c hue;**  You are now connected to database "hue" as user "postgres".  hue=# **create user hue with password 'XXXXXXXXX';**  CREATE ROLE  hue=# **grant all privileges on database hue to hue;**  GRANT  hue=# **\q**  -bash-4.2$ |

# Validate Existing Sqlite3 Users

We can validate the existing users in each instance as follows:

On the first Hue instance.

|  |
| --- |
| [root@bos-rd1-cdh-edge1 ~]# **sqlite3 /var/lib/hue/desktop.db**  SQLite version 3.7.17 2013-05-20 00:56:22  Enter ".help" for instructions  Enter SQL statements terminated with a ";"  sqlite> **select username from auth\_user;**  admin  dustin  hamel  hue  mmckay  nventandy2  nventchris  nventlindsay  nventmatt  rich  richard.stendardo  romit.singhai  sean  sergey  siva.tetala  sqlite> |

On the second Hue instance.

|  |
| --- |
| [root@bos-rd1-cdh-edge2 ~]# **sqlite3 /var/lib/hue/desktop.db**  SQLite version 3.7.17 2013-05-20 00:56:22  Enter ".help" for instructions  Enter SQL statements terminated with a ";"  sqlite> **select username from auth\_user;**  admin  nventlindsay  nventmatt  sqlite> |

# Shutdown Hue

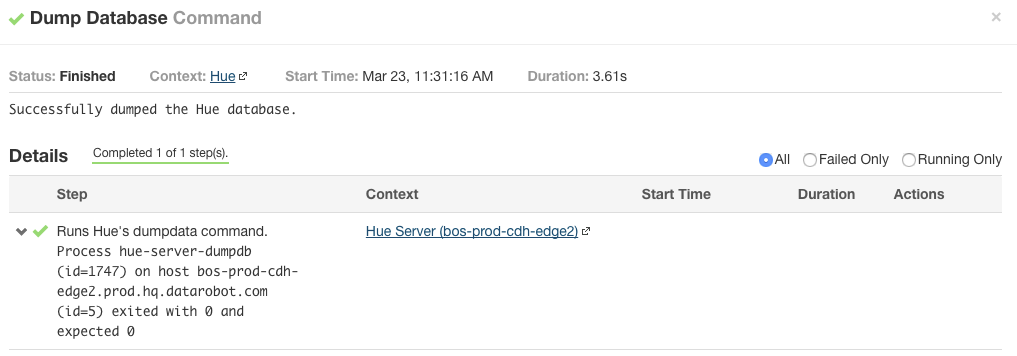
Enable Maintenance Mode for the Hue service in Cloudera Manager and then shutdown all the Hue instances.

# Dump Sqlite3 DB

Next we will dump the Sqlite3 DB instance into a json file and edit the contents.

On the Hue instance screen, selection **Actions->Dump Database**.

## Check Database Host Dumped



When the dump runs, check the Context of the server it is dumping the database from. If it is dumping the wrong database server instance. We need to copy the Sqlite3 database from the active server to the server that the dump is running against.

## Copy Sqlite3 Database to Dump Server if Necessary

Copy the **/var/lib/hue/desktop.db** file from the active server (from the section above where we validated the users in each database) to the server that the dump is running against. Take a copy of the old /var/lib/hue/desktop.db file. Ensure the owner and permissions match as follows.

|  |
| --- |
| [root@bos-prod-cdh-edge1 ~]# **ls -l /var/lib/hue/desktop.db**  -rw-r--r-- 1 hue hue 368640 Mar 21 12:52 /var/lib/hue/desktop.db  [root@bos-prod-cdh-edge1 ~]# |

After copying the file, connect to Sqlite3 and check the users now match the first server as follows.

|  |
| --- |
| [root@bos-prod-cdh-edge2 ~]# **sqlite3 /var/lib/hue/desktop.db**  SQLite version 3.7.17 2013-05-20 00:56:22  Enter ".help" for instructions  Enter SQL statements terminated with a ";"  sqlite> **select username from auth\_user;**  admin  hue  hue\_user  mmckay  nventmatt  richard.stendardo  romit.singhai  sean  siva.tetala  sqlite> |

## Edit Dump File

Open a terminal window for the host and navigate to the dump file in /tmp/hue\_database\_dump.json.

Open (**vi /tmp/hue\_database\_dump.json**) and remove all JSON objects with useradmin.userprofile in the model field. Here are some examples of JSON objects that should be deleted.

|  |
| --- |
| {  "pk": 1,  **"model": "useradmin.userprofile",**  "fields": {  "last\_activity": "2016-01-28T17:15:53.593",  "creation\_method": "HUE",  "first\_login": false,  "user": 1,  "home\_directory": "/user/admin"  }  },  {  "pk": 2,  "model": "useradmin.userprofile",  "fields": {  "last\_activity": "2016-02-16T14:18:19.861",  "creation\_method": "EXTERNAL",  "first\_login": false,  "user": 1100714,  "home\_directory": "/user/nventchris"  }  },  ..... |

# Modify Current Hue.ini

Configure the Hue database:

1. In the Cloudera Manager Admin Console, click the **HUE** service.
2. Click the **Configuration** tab.
3. Select **Scope** > **Hue Server**.
4. Select **Category** > **Advanced**.
5. Set **Hue Server Advanced Configuration Snippet (Safety Valve) for hue\_safety\_valve\_server.ini** with the following:

Make the following changes.

|  |
| --- |
| [desktop]  [[database]]  host=bos-rd1-cdh-master1.rd1.hq.YourCompanyDomain.com  port=5432  engine=postgresql\_psycopg2  user=hue  password=XXXXXXXX  name=hue |

Just for reference, the old setting in hue.ini was the following:

|  |
| --- |
| [[database]]  engine=sqlite3  host=localhost  port=3306  user=hue  name=/var/lib/hue/desktop.db |

Save Changes.

Optionally restore the Hue data to the new database:

1. Select **Actions** > **Synchronize Database**.
2. Determine the foreign key ID.

|  |
| --- |
| -bash-4.2$ **psql -h localhost -U hue -d hue**  psql (9.4.5)  Type "help" for help.  hue=> **\d auth\_permission;**  Table "public.auth\_permission"  Column | Type | Modifiers  -----------------+------------------------+--------------------------------------------------------------  id | integer | not null default nextval('auth\_permission\_id\_seq'::regclass)  name | character varying(50) | not null  content\_type\_id | integer | not null  codename | character varying(100) | not null  Indexes:  "auth\_permission\_pkey" PRIMARY KEY, btree (id)  "auth\_permission\_content\_type\_id\_codename\_key" UNIQUE CONSTRAINT, btree (content\_type\_id, codename)  "auth\_permission\_content\_type\_id" btree (content\_type\_id)  Foreign-key constraints:  "**content\_type\_id\_refs\_id\_d043b34a**" FOREIGN KEY (content\_type\_id) REFERENCES django\_content\_type(id) DEFERRABLE INITIALLY DEFERRED  Referenced by:  TABLE "auth\_group\_permissions" CONSTRAINT "auth\_group\_permissions\_permission\_id\_fkey" FOREIGN KEY (permission\_id) REFERENCES auth\_permission(id) DEFERRABLE INITIALLY DEFERRED  TABLE "auth\_user\_user\_permissions" CONSTRAINT "auth\_user\_user\_permissions\_permission\_id\_fkey" FOREIGN KEY (permission\_id) REFERENCES auth\_permission(id) DEFERRABLE INITIALLY DEFERRED  hue=> |

1. Drop the foreign key that you retrieved in the previous step.

|  |
| --- |
| hue=> **ALTER TABLE auth\_permission DROP CONSTRAINT content\_type\_id\_refs\_id\_d043b34a;**  ALTER TABLE  hue=> |

1. Delete the rows in the django\_content\_type table.

|  |
| --- |
| hue=> **TRUNCATE django\_content\_type CASCADE;**  NOTICE: truncate cascades to table "django\_admin\_log"  NOTICE: truncate cascades to table "desktop\_document"  NOTICE: truncate cascades to table "desktop\_documentpermission"  NOTICE: truncate cascades to table "desktop\_document\_tags"  NOTICE: truncate cascades to table "documentpermission\_users"  NOTICE: truncate cascades to table "documentpermission\_groups"  TRUNCATE TABLE  hue=> |

1. In Hue service instance page, **Actions** > **Load Database**. Confirm you want to load the database by clicking **Load Database**.
2. Add back the foreign key you dropped.

|  |
| --- |
| hue=> **ALTER TABLE auth\_permission ADD CONSTRAINT content\_type\_id\_refs\_id\_d043b34a FOREIGN KEY (content\_type\_id) REFERENCES django\_content\_type(id) DEFERRABLE INITIALLY DEFERRED;**  ALTER TABLE  hue=> |

Start the Hue Service again.

# Validate Changes

The database settings are not written into the default hue.ini file but into the hue\_safety\_valve\_server.ini file in the current running instance as shown below.

|  |
| --- |
| [root@bos-rd1-cdh-edge1 ~]# **ls -d /var/run/cloudera-scm-agent/process/\*-hue-HUE\_SERVER | tail -1**  /var/run/cloudera-scm-agent/process/961-hue-HUE\_SERVER  [root@bos-rd1-cdh-edge1 ~]# **cd /var/run/cloudera-scm-agent/process/961-hue-HUE\_SERVER**  [root@bos-rd1-cdh-edge1 961-hue-HUE\_SERVER]# **ls**  altscript.sh hue.ini navigator.client.properties yarn-conf  cloudera-monitor.properties hue\_safety\_valve.ini redaction-rules.json  creds.localjceks hue\_safety\_valve\_server.ini sentry-conf  hive-conf logs service-metrics.properties  [root@bos-rd1-cdh-edge1 961-hue-HUE\_SERVER]# **cat hue\_safety\_valve\_server.ini**  [desktop]  [[database]]  engine=postgresql\_psycopg2  name=hue  host=bos-rd1-cdh-master1.rd1.hq.YourCompanyDomain.com  port=5432  user=hue  password=XXXXXXX |