# cloudera

# Hue Guide



# **Important Notice**

© 2010-2017 Cloudera, Inc. All rights reserved.

Cloudera, the Cloudera logo, and any other product or service names or slogans contained in this document are trademarks of Cloudera and its suppliers or licensors, and may not be copied, imitated or used, in whole or in part, without the prior written permission of Cloudera or the applicable trademark holder.

Hadoop and the Hadoop elephant logo are trademarks of the Apache Software Foundation. All other trademarks, registered trademarks, product names and company names or logos mentioned in this document are the property of their respective owners. Reference to any products, services, processes or other information, by trade name, trademark, manufacturer, supplier or otherwise does not constitute or imply endorsement, sponsorship or recommendation thereof by us.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Cloudera.

Cloudera may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Cloudera, the furnishing of this document does not give you any license to these patents, trademarks copyrights, or other intellectual property. For information about patents covering Cloudera products, see http://tinv.cloudera.com/patents.

The information in this document is subject to change without notice. Cloudera shall not be liable for any damages resulting from technical errors or omissions which may be present in this document, or from use of this document.

Cloudera, Inc. 1001 Page Mill Road, Bldg 3 Palo Alto, CA 94304 info@cloudera.com US: 1-888-789-1488

Intl: 1-650-362-0488 www.cloudera.com

#### **Release Information**

Version:

Date: May 10, 2017

# **Table of Contents**

Hue Versions	5
Hue Installation	6
Install Hue 1-2-3	6
Easy Install	6
Easy Install Variations	8
Install Hue for Production	11
Custom Database	11
How to Install a Specific Version of Hue	16
How to Install the Latest Version of Hue	16
How to Install a Specific Version of Hue	17
Install Hue on EC2 in AWS	18
Launch EC2 instances in AWS	
Configure Instances and Install Cloudera Manager	18
Install CDH and Hue with Cloudera Manager	20
Hue Custom Databases	22
Connect Hue to an External Database	22
Custom Database Concepts	22
Connect Hue to MySQL or MariaDB	
Install and Configure MySQL or MariaDB Server	
Create Hue Database	
Connect Hue Service to MySQL	
Connect Hue to PostgreSQL	
Install and Configure PostgreSQL Server	
Create Hue Database	
Connect Hue Service to PostgreSQL	31
Connect Hue to Oracle with Client Parcel	33
Install and Configure Oracle Server	
Create Hue Database	35
Create Oracle Client Parcel Repository	36
Connect Hue Service to Oracle	38
Connect Hue to Oracle with Client Package	42
Install and Configure Oracle Server	42
Create Hue Database	43
Install Oracle Client Package	44

Connect Hue Service to Oracle	47
Migrate Hue Database	49
Dump Database	50
Connect New Database	50
Synchronize and Load	51
Hue Custom Database Tutorial	53
Prepare Hosts	53
Install Custom Database	53
Install CM and CDH	54
Populate Database (optional)	
Dump, Synchronize, and Load	
How to Populate the Hue Database	56
Hue Administration	58
Hue Security	59
Hue How-tos	60
How to Add a Hue Load Balancer	60
How to Enable SQL Editor Autocompleter in Hue	60
How to Enable and Use Navigator in Hue	
Enable Navigator in Hue	
Tag Metadata with Navigator in Hue	
Search Metadata with Navigator in Hue	
How to Enable S3 Cloud Storage in Hue	63
Enable S3 in Hue with the S3 Connector Service	63
Enable S3 in Hue with Safety Valves	65
Generate Access Keys in AWS	67
How to Use S3 as Source or Sink in Hue	68
Populate S3 Bucket	68
Create Table with S3 File	68
Export Query Results to S3	69
Troubleshoot Errors	70
How to Run Hue Shell Commands	71
Hue Troubleshooting	73
Potential Misconfiguration Detected	73
Preferred Storage Engine	
MySQL Storage Engine	

# **Hue Versions**

Hue is released upstream, and is also packaged with CDH.

Hue that is packaged with CDH is tightly coupled and cannot be installed or upgraded separately.



Note: Hue package names = <hue version>+<cdh version>+<changes.log>. In CDH 5.11, the package name is hue-3.9.0+cdh5.11.0+5033 because there are 5033 records in the corresponding changes.log.

#### Table 1: Hue Version in CDH

CDH Version	Hue Version
5.11	3.12
5.10	3.11
5.9	3.11
5.8	3.10
5.7	3.9
5.6	3.9
5.5	3.9
5.4	3.7
5.3	3.7
5.2	3.6
5.1	3.6
5.0	3.5

# Links:

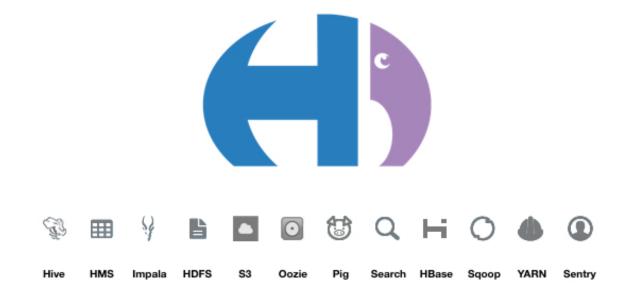
- Hue versions for each CDH 5.x.x release: CDH 5 Packaging and Tarball Information
- Upstream Releases: <a href="http://gethue.com/category/release/">http://gethue.com/category/release/</a>
- GitHub repository: <a href="https://github.com/cloudera/hue">https://github.com/cloudera/hue</a>

# **Hue Installation**

There are two ways to install CDH and Hue, which are packaged together—with Cloudera Manager (managed) and without Cloudera Manager (unmanaged). Each method has multiple variations. The Hue Guide includes two managed installs:

- <u>Easy Install</u> on page 6 (proof-of-concept): Managed deployment with the embedded PostgreSQL database.
- Production Install: Managed deployment with a custom database across the CDH ecosystem.

The Hue Server is a container web application that sits between your CDH installation and the browser. The Hue server hosts a suite of Hue applications and communicates with CDH component servers.



# Install Hue 1-2-3

Minimum Required Role: Full Administrator

This page explains the simplest way to install Cloudera Manager and CDH 5 with Hue.



**Important:** Easy Install uses an embedded database and should not be used in production. For custom database solutions, see <u>Hue Databases</u>.

# Easy Install

The simplest way to install CDH and Hue is with Cloudera Manager, using the embedded database. Easy Install is for **proof-of-concept** installations only.



**Note:** We install the **latest version** of CDH on **CentOS 7**. For more platforms, see <u>Easy Install Variations</u> on page 8.

#### Install Cloudera Manager at the Command Line

- 1. Prepare a cluster of four or more Linux machines with a <u>supported operating system</u>.
- 2. Download a compatible <u>Cloudera Manager package</u> repo (or list) to one host.

- 3. Install Oracle JDK, Cloudera Manager server and daemons, and the embedded PostgreSQL database (from the repo).
- 4. Start the embedded database and Cloudera Manager server.

```
## Download Cloudera Manager to your package manager source directory.
wget https://archive.cloudera.com/cm5/redhat/7/x86_64/cm/cloudera-manager.repo -P
/etc/yum.repos.d/
## Install Cloudera Manager and Dependencies (sourced from the Cloudera Manager repo)
sudo yum install -y oracle-j2sdk1.7
sudo yum install -y cloudera-manager-daemons cloudera-manager-server sudo yum install -y cloudera-manager-server-db-2
## Start the database and server
sudo service cloudera-scm-server-db start
sudo service cloudera-scm-server start
```

```
Starting cloudera-scm-server-db (via systemctl):
Starting cloudera-scm-server (via systemctl):
```

## Install CDH and Hue with Cloudera Manager Installation Wizard

1. Point a browser to the host with the Cloudera Manager server:

```
http:/myhost.example.com:7180
```

- 2. Log on to cloudera MANAGER as admin/admin.
- **3.** Specify hosts with patterns (myhost-[1-n].example.com), or use a legal delimiter.
- 4. Run Cluster Installation to install Cloudera Manager agents:
  - Use parcels (or packages). Keep the other repository defaults.
  - Check both boxes to install JDK 7u67. Do not check if using another supported version.
  - Skip Single User Mode if possible.
  - Inter SSH login credentials. For tips with ec2, see Install Hue on EC2 in AWS on page 18.
  - Solution Cloudera Manager agents are installed. If they fail, click Uninstall failed hosts and Retry.
  - <sup>6</sup> Parcels are downloaded, distributed, and activated across the cluster.
  - Run the Host Inspector to repair issues and click Finish.



- **5.** Run **Cluster Setup** to install Hue and other CDH services:
  - Select services that include Hue, such as **Core with Impala**. Check **Include Cloudera Navigator**.
  - Add 2 roles for the ZooKeeper Server (for a total of 3).
  - Use default **Embedded Database** and store password (or see <u>Hue Custom Databases</u>).
  - Review Changes.
  - <sup>5</sup> First Run commands deploy all selected services.

- 6 Click Finish.
- 6. Go to the Hue Service.

# **Easy Install Variations**

This section expands Install Cloudera Manager at the Command Line on page 6 for other supported platforms.

#### CentOS / RHEL

Available versions: http://archive.cloudera.com/cm5/redhat/

```
## Download cloudera-manager.repo to your package manager source directory
wget https://archive.cloudera.com/cm5/redhat/7/x86_64/cm/cloudera-manager.repo -P
/etc/yum.repos.d/
#wget https://archive.cloudera.com/cm5/redhat/6/x86_64/cm/cloudera-manager.repo -P
/etc/yum.repos.d/
#wget https://archive.cloudera.com/cm5/redhat/5/x86_64/cm/cloudera-manager.repo -P
/etc/yum.repos.d/
## Install Oracle JDK (sourced from the Cloudera repo)
sudo yum install -y oracle-j2sdk1.7
## Install Cloudera Manager daemons and server
sudo yum install -y cloudera-manager-daemons cloudera-manager-server
## Install the embedded PostreSQL database
sudo yum install -y cloudera-manager-server-db-2
## Start the database and server
sudo service cloudera-scm-server-db start
sudo service cloudera-scm-server start
```



**Important:** CentOS/RHEL 5: You must install python26 and <u>psycopg2</u> (Python-PostgreSQL Database Adapter).

```
## On all machines in the cluster
yum install -y python26

## On the machine hosting the Hue server
yum install -y gcc python26-devel postgresq184-devel
wget http://initd.org/psycopg/tarballs/PSYCOPG-2-6/psycopg2-2.6.2.tar.gz
tar zxvf psycopg2-2.6.2.tar.gz
cd psycopg2-2.6.2
/usr/bin/python2.6 ./setup.py install
```

If the Hue Server is already installed, import or link to psycopg2 into the Hue environment:

```
## Navigate to Hue within your specific CDH parcel version
cd /opt/cloudera/parcels/`ls -l /opt/cloudera/parcels | grep CDH | tail
-l | awk '{print $9}'`/lib/hue/build/env/bin
./python2.6
>>>> import psycopg2

cd /opt/cloudera/parcels/`ls -l /opt/cloudera/parcels | grep CDH | tail
-l | awk '{print $9}'`/lib/hue/build/env/lib/python2.6/site-packages/
ln -s /usr/lib64/python2.6/site-packages/psycopg2
```



**Note:** CentOS/RHEL 7.3: To resolve transparent huge page compaction errors, you must run:

# Run on each host in cluster and add to rc.local echo never > /sys/kernel/mm/transparent\_hugepage/defrag echo never > /sys/kernel/mm/transparent\_hugepage/enabled

#### SLES

Available versions: http://archive.cloudera.com/cm5/sles/

#### SLES 12sp2

## Refresh repo for python-psycopg2 on each host in the cluster zypper addrepo

http://download.opensuse.org/repositories/server:/database:/postgresql/SIE\_12\_SP2/server:database:postgresql.repo zypper refresh

## Download cloudera-manager.repo to your package manager source directory wget https://archive.cloudera.com/cm5/sles/12/x86\_64/cm/cloudera-cm.repo -P /etc/zypp/repos.d/

## Install Oracle JDK (sourced from the Cloudera repo) sudo zypper install oracle-j2sdk1.7

## Install Cloudera Manager daemons and server sudo zypper install cloudera-manager-daemons cloudera-manager-server

## Install the embedded PostreSQL database sudo zypper install cloudera-manager-server-db-2

## Start the database and server sudo service cloudera-scm-server-db start sudo service cloudera-scm-server start



Note: SLES 12sp1: Two workarounds are required:

## Edit baseurl in devel languages python.repo to avoid refresh error vi /etc/zypp/repos.d/devel\_languages\_python.repo baseurl=http://download.opensuse.org/repositories/devel:/languages:/python/SLE\_12\_SP1 sudo zypper refresh

## To start the Cloudera Manager server, reload the daemon) cd /var/log/cloudera-scm-server sudo systemctl daemon-reload

# SLES 11sp4

## Refresh repo for python-psycopg2 on each host in the cluster zypper addrepo

http://download.opensuse.org/repositories/server:database:postgresql/SIE 11 SP4/server:database:postgresql.repo zypper refresh

## Download cloudera-manager.repo to your package manager source directory wget https://archive.cloudera.com/cm5/sles/11/x86\_64/cm/cloudera-manager.repo -P /etc/zypp/repos.d/

## Install Oracle JDK (sourced from the Cloudera repo) sudo zypper install oracle-j2sdk1.7

## Install Cloudera Manager daemons and server sudo zypper install cloudera-manager-daemons cloudera-manager-server

```
## Install the embedded PostreSQL database
sudo zypper install cloudera-manager-server-db-2
## Start the database and server
sudo service cloudera-scm-server-db start
sudo service cloudera-scm-server start
```

#### Ubuntu

Available versions: <a href="http://archive.cloudera.com/cm5/ubuntu/">http://archive.cloudera.com/cm5/ubuntu/</a>

```
## Download cloudera.list to your package manager source directory
wget https://archive.cloudera.com/cm5/ubuntu/xenial/amd64/cm/cloudera.list -P
/etc/apt/sources.list.d/
#wget https://archive.cloudera.com/cm5/ubuntu/trusty/amd64/cm/cloudera.list -P
/etc/apt/sources.list.d/
#wget https://archive.cloudera.com/cm5/ubuntu/precise/amd64/cm/cloudera.list -P
/etc/apt/sources.list.d/
## Add a repository key
wget https://archive.cloudera.com/cm5/ubuntu/xenial/amd64/cm/archive.key -O archive.key
#wget https://archive.cloudera.com/cm5/ubuntu/trusty/amd64/cm/archive.key -O archive.key
#wget https://archive.cloudera.com/cm5/ubuntu/precise/amd64/cm/archive.key -O archive.key
sudo apt-key add archive.key
sudo apt-get update
## Install Oracle JDK (sourced from the Cloudera repo)
sudo apt-get install oracle-j2sdk1.7
## Install Cloudera Manager daemons and server
sudo apt-get install cloudera-manager-daemons cloudera-manager-server
## Install the embedded PostreSQL database
sudo apt-get install cloudera-manager-server-db-2
## Start the database and server
sudo service cloudera-scm-server-db start
sudo service cloudera-scm-server start
```

#### Debian

Available versions: http://archive.cloudera.com/cm5/debian/

```
## Download cloudera.list to your package manager source directory
wget https://archive.cloudera.com/cm5/debian/jessie/amd64/cm/cloudera.list -P
/etc/apt/sources.list.d/
#wget https://archive.cloudera.com/cm5/debian/wheezy/amd64/cm/cloudera.list -P
/etc/apt/sources.list.d/
## Add a repository key
wget https://archive.cloudera.com/cm5/debian/jessie/amd64/cm/archive.key -0 archive.key
#wget https://archive.cloudera.com/cm5/debian/wheezy/amd64/cm/archive.key -O archive.key
sudo apt-key add archive.key
sudo apt-get update
## Install Oracle JDK (sourced from the Cloudera repo)
sudo apt-get install oracle-j2sdk1.7
## Install Cloudera Manager daemons and server
sudo apt-get install cloudera-manager-daemons cloudera-manager-server
## Install the embedded PostreSQL database
sudo apt-get install cloudera-manager-server-db-2
## Start the database and server
sudo service cloudera-scm-server-db start
sudo service cloudera-scm-server start
```

# Install Hue for Production

Minimum Required Role: Full Administrator



Warning: This page is 1 step toward a production-ready installation. Watch the Hue Guide for more!



Important: Enterprise customers should review Cloudera Installation and Cloudera Security and work with customer support to ensure a fully secure and production-quality system.

A production-ready installation of CDH and Hue must be robust and secure. At a minimum, it has:

- Custom database for Cloudera Manager and select servers such as Hue, Hive Metastore, Oozie
- High Availability for Hue, Hive, Impala, YARN (Resource Manager), HDFS (NameNode)
- Secure ecosystem with authentication, authorization, data protection, and data governance.

This page explains how to install a custom (external) database for Cloudera Manager server and all applicable CDH components (not only Hue). We also configure Hue with multiple servers and a Load Balancer to increase performance.

#### **Custom Database**

On this page, we configure Cloudera Manager and applicable CDH services with MySQL 5.7 on CentOS 7.3.

#### **Prepare Cluster**

When setting up a cluster of machines, you can refresh the default packages, but it is best not to upgrade. Cloudera packages are built with the defaults.

- 1. Prepare at least four machines with a <u>supported operating system</u> and uniform capacity:
  - Data <u>replication</u> defaults to three, so a good minimum configuration is three DataNodes, plus the NameNode, each on their own hosts.
  - The HDFS balancer distributes data according to the capacity of the host, so uniform DataNode hosts allows for the even distribution of bytes.
- 2. Ensure that the primary host, with Cloudera Manager server, has uniform SSH access on the same port to all hosts.
- 3. Ensure that all hosts have internet access to (a) standard package repositories and the (b) archives for Cloudera Manager and CDH.

## Install Cloudera Manager



Note: In this installation, we let Cloudera Manager install its agents across the cluster.

1. Download cloudera-manager.repo to your package manager source directory. For CentOS 7:

wget https://archive.cloudera.com/cm5/redhat/7/x86\_64/cm/cloudera-manager.repo -P /etc/yum.repos.d/

2. Install Oracle JDK (sourced from the Cloudera repo)

sudo yum install -y oracle-j2sdk1.7

3. Install Cloudera Manager daemons and server (and let Cloudera Manager install the agents)

```
sudo yum install cloudera-manager-daemons cloudera-manager-server
```

**Table 2: Available Cloudera Manager Packages** 

OS	Target Dir	Repo
CentOS / RHEL	/etc/yum.repos.d/	/rednat/ <ver>/x86_64/an/cloudera-manager.repo</ver>
SLES	/etc/zypp/repos.d/	/sles/ <ver>/x86_64/an/cloudera-manager.repo</ver>
Ubuntu	/etc/apt/sources.list.d/	/ubuntu/ <ver>/x86_64/cm/cloudera.list</ver>
Debian	/etc/apt/sources.list.d/	/debian/ <ver>/x86_64/an/cloudera.list</ver>

#### **Install Custom Database**

We install MySQL 5.7 (64 bit) on CentOS 7.3. It can be on one host within the cluster or remotely. For help with other databases and platforms, see <a href="Hue Custom Databases">Hue Custom Databases</a>.

Tip: To view logs: tail /var/log/mysqld.log

1. Download MySQL 5.7 from the MySQL Yum repository, install, and start:

```
wget http://repo.mysql.com/mysql57-community-release-el7-9.noarch.rpm
rpm -ivh mysql57-community-release-el7-9.noarch.rpm
yum install -y mysql-community-server
systemctl start mysqld.service
```

2. Update the temporary password to one that passes the default <u>Validation Plugin</u> options:

```
sudo grep 'temporary password' /var/log/mysqld.log

mysql -uroot -p
Enter password: <Enter temporary password>

mysql> ALTER USER 'root'@'localhost' IDENTIFIED BY 'Create new root password';
mysql> quit
```

**3.** Secure your installation. If you make a mistake, simply rerun:

```
sudo /usr/bin/mysql_secure_installation

Enter password for user root: [Enter new root password]
[...]
Change the password for root ? n
    ... skipping.
Remove anonymous users? Y
[...]
Disallow root login remotely? n
    ... skipping.
Remove test database and access to it? Y
[...]
Reload privilege tables now? Y
All done!
```

4. Enable the server to automatically start on boot:

```
sudo systemctl enable mysqld.service
```

- 5. Configure /etc/my.cnf. The bind/listen address and storage engine are important:
  - Set <u>bind-address</u>=0.0.0.0 (or comment-out, or remove, if the default).
  - Set <u>default-storage-engine</u>=innodb (the <u>default</u> in 5.5 and higher: mysql -V).
  - Ensure sql mode=STRICT ALL TABLES to avoid the Known Issue of columns being truncated during migration.

A typical my.cnf file *might* have these settings:



**Important:** Update my . cnf properties as appropriate for your environment.

```
## Example of my.cnf settings
[mysqld]
bind-address=0.0.0.0
default-storage-engine=innodb
innodb_lock_wait_timeout=240
innodb_file_per_table=1
innodb_buffer_pool_size=200M
innodb_flush_method=O_DIRECT
innodb_file_per_table=1
innodb_flush_log_at_trx_commit=2
innodb_log_buffer_size=4M
## Set value to ~ (CPUs/cores) * 2
innodb_thread_concurrency=8
## For log_file_size errors, see <a href="InnoDB">InnoDB</a> <a href="Error ib logfile">Error ib logfile</a> of different size</a>
innodb_log_file_size=100M
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
user=mysql
max_allowed_packet=64M
lower_case_table_names=1
sql-mode=STRICT_ALL_TABLES
## Disable to prevent various security risks
symbolic-links=0
## Do not skip InnoDB!
#skip-innodb
[mysqld safe]
log-error=/var/log/mysqld.log
pid-file=/var/run/mysqld/mysqld.pid
```

## 6. Restart the server:

```
sudo systemctl restart mysqld.service
```

#### **7.** Ensure InnoDB is the default storage engine:

```
mysql -u root -p
Enter password: <root password>
mysql> show engines;
mysql> quit
```

```
mysql> show engines;
                                                                                                 | Transactions | XA | Savepoints |
                    | Support | Comment
                    | DEFAULT | Supports tran
MRG MYISAM
                     I YES
                              | Collection of identical MyISAM tables
                                                                                                   NO
                                                                                                                 I NO
                                                                                                                        I NO
                                                                                                                       I NO
MEMORY
                    I YES
                              I Hash based, stored in memory, useful for temporary tables
                                                                                                   NO
                                                                                                                I NO
BLACKHOLE
                              I /dev/null storage engine (anything you write to it disappears)
                                                                                                                 I NO
                                                                                                                       I NO
                    I YES
                                                                                                   NO
MVISAM
                              | MyISAM storage engine
                                                                                                                 I NO
                                                                                                                       I NO
                     I YES
                                                                                                   NO
                              I CSV storage engine
                                                                                                   NO
                                                                                                                 I NO
                                                                                                                       I NO
CSV
                    I YES
ARCHIVE
                    I YES
                               I Archive storage engine
                                                                                                   NO
                                                                                                                 NO
                                                                                                                        I NO
PERFORMANCE_SCHEMA
                    I YES
                                                                                                                  NO
                                Performance Schema
                                                                                                   NO
                                                                                                                         NO
FEDERATED
                    I NO
                               I Federated MySQL storage engine
                                                                                                 I NULL
rows in set (0.00 sec)
```

**8.** Create databases and grant permissions (with scripts):

```
# Paste into database script
create database scm default character set utf8 default collate utf8_general_ci;
create database amon default character set utf8 default collate utf8_general_ci;
create database rman default character set utf8 default collate utf8_general_ci;
create database hive default character set utf8 default collate utf8_general_ci;
create database hue default character set utf8 default collate utf8_general_ci;
create database nav default character set utf8 default collate utf8_general_ci;
create database navms default character set utf8 default collate utf8_general_ci;
create database nozie default character set utf8 default collate utf8_general_ci;
create database sentry default character set utf8 default collate utf8_general_ci;
```

Tip: Consider generating random passwords: <a href="http://passwordsgenerator.net/">http://passwordsgenerator.net/</a>.

```
vi mysql_grant_perms.ddl
# Assign passwords and paste into grant script
grant all on scm.* to 'scm'@'%'
                                        identified by 'scmP@ssw0rd';
grant all on amon.*
                       to 'amon'@'%'
                                        identified by 'amonP@ssw0rd';
                      to 'rman'@'%'
                                        identified by 'rmanP@ssw0rd';
grant all on rman.*
grant all on hive.*
                       to 'hive'@'%'
                                        identified by 'hiveP@ssw0rd';
                                        identified by 'hueP@ssw0rd';
                       to 'hue'@'%'
grant all on hue.*
                                        identified by 'navP@ssw0rd';
grant all on nav.*
                       to 'nav'@'%'
grant all on navms.* to 'navms'@'%' grant all on oozie.* to 'oozie'@'%'
                                        identified by 'navmsP@ssw0rd';
                                        identified by 'oozieP@ssw0rd';
grant all on sentry.* to 'sentry'@'%' identified by 'sentryP@ssw0rd';
# Run scripts
mysql -u root -p < mysql_create_dbs.ddl</pre>
Enter password: <root password>
mysql -u root -p < mysql_grant_perms.ddl</pre>
Enter password: <root password>
```

**9.** Verify the connection to the scm database:

```
mysql -u scm -p
Enter password: scmP@ssw0rd
mysql> quit
```

#### **Prepare Database**

The Cloudera Manager server only starts with the JDBC driver on every applicable host.

1. Install the JDBC driver on every host in the cluster (or on every host with a server role that uses a custom database).

```
# Downlaod and unpack MySQL/J connector
wget https://dev.mysql.com/get/Downloads/Connector-J/mysql-connector-java-5.1.41.tar.gz
tar zxvf mysql-connector-java-5.1.41.tar.gz
```

```
# Create directory /usr/share/java if it does not exist
sudo mkdir -p /usr/share/java/
# Copy to /usr/share/java
sudo cp mysql-connector-java-5.1.41/mysql-connector-java-5.1.41-bin.jar
/usr/share/java/mysql-connector-java.jar
```

2. Prepare scm database (only on the host with Cloudera Manager server):

/usr/share/cmf/schema/scm\_prepare\_database.sh mysql scm scm Enter SCM password: scmP@ssw0rd



Note: If you installed MySQL remotely, use the options to set hostname. See Syntax for scm prepare database.sh.

/usr/share/cmf/schema/scm\_prepare\_database.sh mysql [options] dbname username password

3. Start the Cloudera Manager server!

sudo service cloudera-scm-server start

Starting cloudera-scm-server (via systemctl):

#### Install CDH and Hue

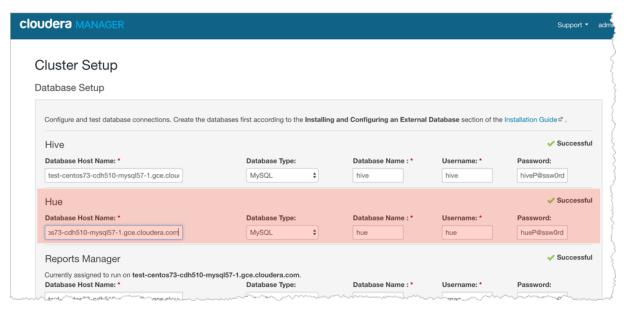


Note: Ideally, Hue should always use a custom database and be configured with 3+ servers and 1 or 2 load balancers.

1. Point a browser to the host with Cloudera Manager server:

http://myhost.example.com:7180/

- 2. Log on to cloudera MANAGER as admin/admin.
- 3. Install CDH and Hue with Cloudera Manager Installation Wizard on page 7.
  - Stop at the Database Setup page and set all database properties for name, username, and password. Hostname and database type should be set.



- **4.** When Hue is running, add more Hue servers and a Load Balancer:
  - a. Select Actions > Add Role Instances.
  - b. Add 2 additional Hue servers (for a total of 3) and click OK.
  - c. Add 1 Load Balancer and click OK > Continue.
  - d. Check the boxes for the new servers and load balancer.
  - e. Select Actions for Selected > Start > Start.
  - f. Click Save Changes and Restart Hue.
- 5. Click Hue Web UI > Load Balanced Hue Web UI.
- **6.** Log on to Hue and ensure the port is 8889.

# How to Install a Specific Version of Hue

For a specific version of Hue, you need a specific version of CDH. You cannot install Hue independently from CDH. See **Hue Versions** on page 5

## How to Install the Latest Version of Hue

Easy Install on page 6 explains how to deploy the latest version of CDH and Hue. The key parts are:

1. Download the Cloudera Manager package for your OS on the primary host in your cluster:

```
$ cd /etc/<package_manager_source_dir>/
$ wget
https://archive.cloudera.com/cm5/<os_type>/<os_ver>/x86_64/cm/<repo_name_can_differ>
```

**Table 3: Available Cloudera Manager Packages** 

OS	Target Dir	Repo
CentOS / RHEL	/etc/yum.repos.d/	/rechet/ser>/x86_64/an/cloudera-manager.repo
SLES	/etc/zypp/repos.d/	/sles/ <e>/x86_64/an/cloudera-manager.repo</e>
Ubuntu	/etc/apt/sources.list.d/	/ubuntu/ <ver>/x86_64/an/cloudera.list</ver>
Debian	/etc/apt/sources.list.d/	/debian/ <ver>/x86_64/cm/cloudera.list</ver>

2. Run the Cloudera Manager wizard in a browser and use the default repository paths.

# How to Install a Specific Version of Hue

For a specific version, download the same Cloudera Manager package and then edit some paths:

**1.** Download the Cloudera Manager 5 package for your OS, for example:

```
wget https://archive.cloudera.com/cm5/redhat/6/x86_64/cm/cloudera-manager.repo -P
/etc/yum.repos.d/
```

**2.** Edit baseurl in the repo file (the filename can differ):

```
vi /etc/yum.repos.d/cloudera-manager.repo
#vi /etc/yum.repos.d/cloudera-cm.repo
```

**a.** For the latest maintenance version of a *specific minor version* (for example, the latest 5.7 version):

```
baseurl = https://archive.cloudera.com/cm5/redhat/6/x86_64/cm/5.7/
```

**b.** For a specific maintenance version (for example, 5.7.2):

```
baseurl = https://archive.cloudera.com/cm5/redhat/6/x86_64/cm/5.7.2/
```

- Run cloudera MANAGER wizard and stop at Step 1 of Cluster Installation.
- 4. Click More Options for CDH parcels.
- **5.** Edit CDH 5 **Remote Parcel Repository URL** from {latest\_supported} to match your version and save:
  - **a.** For the latest maintenance release of a *specific minor* version:

```
http://archive.cloudera.com/cdh5/parcels/5.7/
```

**b.** Or, for a *specific maintenance* release:

```
http://archive.cloudera.com/cdh5/parcels/5.7.2/
```

6. Select the newly enabled radio button for your version under Select the version of CDH:

```
Select the version of CDH
                          CDH-5.7.2-1.cdh5.7.2.p0.18
                          ODH-4.7.1-1.cdh4.7.1.p0.47
                          Versions of CDH that are too new for this version of Cloudera Manager (5.8.4) will not be shown.
```

- 7. Scroll down and select the **Custom Repository** button.
- 8. Enter a custom URL for the matching Cloudera Manager Agent repo:
  - **a.** For the latest maintenance release of a *specific minor* version:

```
https://archive.cloudera.com/cm5/redhat/6/x86_64/cm/5.7/
```

**b.** Or, for a *specific maintenance* release:

```
https://archive.cloudera.com/cm5/redhat/6/x86_64/cm/5.7.2/
```

**9.** Enter a custom URL for the location of the **GPG signing key**:

```
https://archive.cloudera.com/cm5/redhat/6/x86_64/cm/RPM-GPG-KEY-cloudera
```

The rest of the installation should be the same as **Easy Install** on page 6.

# Install Hue on EC2 in AWS

This page explains one way to install CDH and Hue on an EC2 cluster in AWS.

#### Launch EC2 instances in AWS

This is *one way* to create a cluster of <u>ec2</u> instances for installing CDH. Ideally, you have **four instances** with at least 2 cores and 8 GB of RAM.



**Note:** Data <u>replication</u> defaults to three, so a good minimum configuration is 3 DataNodes + 1 NameNode, each on its own host.

- 1. Log on to Amazon Web Services and go to the EC2 Dashboard.
- 2. Click Launch Instance.
- **3.** Select a Linux distribution (here we use RedHat 7.3).
- 4. Select m3.large (at a minimum).
- 5. Click Next: Configure Instance Details.
- **6.** Increase the **Number of Instances** to 4 (at a minimum).
- 7. Click Next: Add Storage and increase size to 100 GB.
- 8. Click Next: Add Tags and name your instances.
- 9. Click Next: Configure Security Group and click Add Rule:
  - Select Custom TCP Rule = port 7180
  - Select Custom source = 0.0.0.0/0.
- 10 Repeat for other Ports Used by Cloudera Manager and Cloudera Navigator to build a list:
  - 7180: Cloudera Manager http web console
  - 7183: Cloudera Manager https web console
  - 7182: Cloudera Manager listens to agent heartbeats
  - 7432: Embedded PostgreSQL database
  - 9000: Cloudera Manager server and agent communication
  - 9001: Cloudera Manager server and agent communication
- 11. Click Review and Launch.
- 12 Select Create a new key pair, name it, and click Download Key Pair (or use an existing one).
- 13 Click Launch Instances, and when ready, View Instances.

### Configure Instances and Install Cloudera Manager

These steps are for RedHat 7.3 (user\_name= ec2-user). No matter the distribution, always:

- Disable SE Linux.
- Disable transparent huge page compaction.
- Set swappiness to 10.

Instructions are below. Also see Connecting to Your Linux Instance Using SSH.

#### Run on all instances in the cluster

Update settings and reboot instances for them to take effect.



**Note:** Consider creating an image of one instance when done with this configuration.

1. Log on to each EC2 instance from a terminal:

```
chmod 400 <private_key>.pem
ssh -i <private_key>.pem user_name@<public_dns_name>
sudo su -
```

2. Update yum and install wget:

```
yum -y update
yum -y install wget
```

- **3.** Set swappiness to 10 by editing /etc/sysctl.conf:
  - **a.** Run for this shell:

```
sysctl -w vm.swappiness=10
```

**b.** Append property to /etc/sysctl.conf:

```
vi /etc/sysctl.conf
vm.swappiness=10
```

To check the status: sysctl -n vm.swappiness.

- **4.** Disable transparent huge page compaction by editing /etc/rc.local:
  - a. Run on each instance:

```
echo never > /sys/kernel/mm/transparent_hugepage/defrag
echo never > /sys/kernel/mm/transparent_hugepage/enabled
```

**b.** Append commands to /etc/rc.local and change permissions:

```
vi /etc/rc.local
echo never > /sys/kernel/mm/transparent_hugepage/defrag
echo never > /sys/kernel/mm/transparent_hugepage/enabled
chmod 755 /etc/rc.d/rc.local
source /etc/rc.local
```

**5.** Disable SE Linux by editing /etc/selinux/config and rebooting the instance:

```
vi /etc/selinux/config
SELINUX=disabled
reboot
```

To check the status: sestatus

#### Run on one instance only

Install Cloudera Manager and its dependencies. Create a small script or run the commands individually:

```
#!/bin/bash
## Download the Cloudera Manager repository for the latest release (on your OS/ver):
wget https://archive.cloudera.com/cm5/redhat/7/x86_64/cm/cloudera-manager.repo -P
/etc/yum.repos.d/

## Install Cloudera Manager and dependencies:
yum install -y oracle-j2sdk1.7
yum install -y oracle-j2sdk1.7
yum install -y cloudera-manager-daemons cloudera-manager-server
yum install -y cloudera-manager-server-db-2

## Start servers:
service cloudera-scm-server-db start
service cloudera-scm-server start

chmod 744 install_cm.sh
./install_cm.sh
```

# Install CDH and Hue with Cloudera Manager

Follow the wizard defaults for a simple installation. Less intuitive areas are explained below.

- 1. Point a browser to: http://<public dns>.<region>.compute.amazonaws.com:7180.
- **2.** After a minute or two, log on as admin / admin.
- **3.** Accept the user agreement and continue until you reach the wizard.

#### **Welcome Steps**

- 1. End User License Terms and Conditions.
- 2. Which edition do you want to deploy? >> Select Cloudera EnterpriseData Hub Edition Trial
- Thank you for choosing Cloudera Manager and CDH.
- Specify hosts for your CDH cluster installation. >> Input comma-separated hostnames.

# Cluster Installation (7 steps)

- **1.** Select Repository. >> For the latest release in parcels, keep the defaults.
- 2. JDK Installation Options. >> Check both boxes.
- **3.** Enable Single User Mode. >> Ignore single user mode if possible.
- 4. Provide SSH login credentials. >> Set user = ec2-user & upload <private\_key>.pem.
- **5.** Installation in progress.
- **6.** Installing Selected Parcels.
- 7. Inspect hosts for correctness and click Finish. >> Repair issues as necessary.

# Cluster Setup (6 steps)

- 1. Choose the CDH 5 services that you want to install on your cluster. >> Select Core services with Impala.
- 2. Customize Role Assignments >> Add 2 Zookeeper roles (for a total ensemble of 3). See <u>Designing a ZooKeeper Deployment</u>.
- 3. Database Setup >> Use Embedded Database (Postgres). Copy Hue password for safekeeping.



Note: Copy the Hue database hostname and password for safekeeping.

4. Review Changes.

5. First Run Command.



Note: HDFS may fail with the message, SafeModeException: Cannot create directory /tmp/. If so, turn off safemode on the failing host then click **Retry**:

sudo -u hdfs hdfs dfsadmin -safemode leave

6. Click Finish.

# **Hue Custom Databases**

Hue needs its own database for such things as user account information, job submissions, and Hive queries.

Hue is packaged with a lightweight **embedded database** (PostgreSQL) for proof-of-concept deployments with one Hue server. Hue also supports connections to a custom **external database**, local or remote.



**Important:** Cloudera recommends an external database in production environments.

# Connect Hue to an External Database

- Connect Hue to MySQL or MariaDB on page 23
- Connect Hue to PostgreSQL on page 28
- Connect Hue to Oracle with Client Parcel
- Connect Hue to Oracle with Client Package

# **Custom Database Concepts**

- There are two ways to connect Hue to an external database:
  - During a new CDH installation with the Cloudera Manager Installation Wizard at **Database Setup**. The external (or custom) database must be installed, configured, and running.
  - After CDH is installed with Cloudera Manager on the Hue > Configuration tab. You can migrate and connect, or simply connect to the new database without saving the data in the old database.
- Migrate to a new database *only if* you want to save data in your current database. Otherwise, simply connect to your new database and restart Hue.
  - 1. [migrate] Stop the Hue service.
  - 2. [migrate] Dump database (and delete "useradmin.userprofile" objects from . json file).
  - 3. Connect to new database.
  - 4. [migrate] Synchronize database (and drop foreign key to clean tables).
  - 5. [migrate] Load database (and add foreign key).
  - **6. Re/Start** Hue service.
- **Install Oracle Instant Client libraries** (Basic and SDK with headers) to use an Oracle database with Hue. You can use the <u>zip files</u> from Oracle *or* the <u>parcel</u> from Cloudera.
- An external database can be remote—it does not need to be on the same host as the Hue server. Ensure the database server is properly configured (particularly the bind or listen address).
- Managed CDH deployments must use Cloudera Manager to configure hue.ini:

```
[desktop]
...
[[database]]
host=Database server host
port=Database server port
engine=Database server type (mysql, postgresql, oracle)
name=Hue database name (or SID)
user=Hue datbase username
password=Hue database password
```

# Connect Hue to MySQL or MariaDB

If you have an external database installed, review MySQL/MariaDB Troubleshooting on page 23 before creating a database for Hue.

# Install and Configure MySQL or MariaDB Server

MariaDB is a fork of the MySQL relational database. Refer to the MariaDB documentation or MySQL documentation for more help on how to install a MariaDB or MySQL database.

# MySQL/MariaDB Troubleshooting

Pay close attention to these areas and revisit when troubleshooting:

- · Remote connections:
  - The bind or address should be set to 0.0.0.0 so it can listen to multiple hosts.
  - Grant wildcard (%) permissions to the Hue database user so it can connect from any host.
  - Install a JDBC connector if necessary, for example, if your CDH version does not include it.
- Security: Delete anonymous users because they are able to log on without a password.
- **Storage engine**: Use <a href="InnoDB">InnoDB</a> (the default engine in version 5.5.5 and higher: <a href="mysql">mysql</a> -V).
- Data validation: Use sql mode=STRICT\_ALL\_TABLES to prevent columns being truncated during migration.

#### Install MySQL or MariaDB Server

1. Install MariaDB or MySQL. The table lists the max version of each supported distribution for this CDH release, and corresponding default database versions.

**Table 4: Install Commands for Supported OS Versions** 

os	OS Ver	DB Ver	Command	
	7.3		No package mysql-server available.	
		5.5	sudo yum install mariadb-server	
	6.8	5.1	sudo yum install mysql-server	
			No package mariadb-server available.	
CentOS / RHEL	5.10	5.6	# CentOS 5 needs MySOL Connector/J for remote connections wget http://dww.lasfamy.td/Mg.lowlasfcometrja513targ tar zxvf mysql-connector-java-5.1.39.tar.gz  No package mariadb-server available.	
			'mysql' not found in package names.	
CLEC	12.2	10.0	sudo zypper install mariadb	
SLES	11.4	5.5	sudo zypper install mysql	
			'mariadb' not found in package names.	

# **Hue Custom Databases**

os	OS Ver	DB Ver	Command		
	16.04	5.7	sudo apt-get install mysql-server #set root psswd when prompted		
	10.04	10.0	sudo apt-get install <b>mariadb</b> -server #set root psswd when prompted		
Ubuntu	14.04	5.5	sudo apt-get install mysql-server #set root psswd when prompted		
	14.04	5.5	sudo apt-get install <b>mariadb</b> -server #set root psswd when prompted		
	12.04	5.5	sudo apt-get install mysql-server #set root psswd when prompted		
			Unable to locate package mariadb-server		
	0.4	5.5	sudo apt-get install mysql-server #set root psswd when prompted		
Debian	8.4	10.0	sudo apt-get install <b>mariadb</b> -server #set root psswd when prompted		
	7.8	5.5	sudo apt-get install mysql-server #set root psswd when prompted		
			Package 'mariadb-server' has no installation candidate		

**2.** Start the database server as necessary (some are automatically started):

**Table 5: Start Commands** 

os	OS Ver	Command
ContOS / PHEI	7.3	sudo systemctl start mariadb
CentOS / RHEL  5.10, 6.8 sudo servi		sudo service mysql <b>d</b> start
SLES	11.4, 12.1, 12.2	sudo rcmysql start
Ubuntu	12.04, 14.04, 16.04	sudo service mysql start
Debian	7.8, 8.4	sudo service mysql start

**3.** Secure your installation. If you make a mistake, simply rerun:

```
sudo /usr/bin/mysql_secure_installation
Enter current password for root (enter for none): [If unset, press Enter.]
OK, successfully used password, moving on...
Set root password? [Y/n] Y [Enter n if password is set.]
New password:
Re-enter new password:
Remove anonymous users? [Y/n] Y
Disallow root login remotely? [Y/n] N
[...]
Remove test database and access to it [Y/n] Y
[...]
Reload privilege tables now? [Y/n] Y
```

# Configure MySQL or MariaDB Server

- 1. Configure my.cnf (only as necessary).
  - Ensure bind-address=0.0.0.0 (or is commented out if the default).
  - Ensure <u>default-storage-engine</u>=innodb (which is the <u>default</u> in 5.5 and higher: mysql -V).
  - Ensure sql mode=STRICT ALL TABLES to avoid the Known Issue of columns being truncated during migration.

```
[mysqld]
bind-address=0.0.0.0
default-storage-engine=innodb
sql_mode=STRICT_ALL_TABLES
```

- CentOS/RHEL/SLES: /etc/my.cnf
- Ubuntu/Debian: /etc/mysql/my.cnf
- 2. Restart the database server.



Note: See the Table 5: Start Commands on page 25 table above and replace with "restart".

**3.** Enable the server to automatically start on boot:

**Table 6: Enable Automatic Start** 

os	OS Ver	Command			
CentOS / RHEL	7.3	sudo systemctl enable mariadb			
Centos/ KILL	5.10, 6.8	sudo <u>chkconfig</u> mysqld on			
SLES	11.4, 12.1, 12.2	sudo chkconfig mysql on sudo rcmysql status			
<u>Ubuntu</u>	12.04, 14.04, 16.04	# preconfigured to start at boot sudo service mysql status			
Debian	7.8, 8.4	# preconfigured to start at boot sudo service mysql status			

#### Create Hue Database

1. Log on to MySQL with your root password:

```
mysql -u root -p
Enter password: <root password>
```

2. Create a database for Hue (we call it "hue" but any name works) with UTF8 collation and grant user privileges:

```
create database hue default character set utf8 default collate utf8_general_ci;
grant all on hue.* to 'hue'@'%' identified by 'huepassword';
select * from information_schema.schemata;
quit
```

**3.** Verify the connection to the Hue database:

```
mysql -u hue -p
Enter password: <your hue password>
quit
```



**Note:** Ensure Hue uses UTF8 collation and character set. Some commands:

```
# To create (use utf8_general_ci or utf8mb4_general_ci):
CREATE DATABASE hue COLLATE = 'utf8_general_ci';
# To view default_character_set_name and default_collation_name
SELECT * FROM INFORMATION_SCHEMA.SCHEMATA;
# To alter if not created with UTF8 collation
ALTER DATABASE hue COLLATE = 'utf8_general_ci';
```

See Setting Character Sets and Collations.

# Connect Hue Service to MySQL

Tip: To save the data in your current database (embedded or external), you must migrate (dump, synch, load) before connecting to the new database. Otherwise, skip those steps.

1. Stop Hue Service

- **a.** In Cloudera Manager, navigate to **Cluster > Hue**.
- **b.** Select **Actions** > **Stop**.



**Note:** Refresh the page if the Hue service does not look stopped: •.

- 2. [migration only] Dump Current Database
  - a. Select Actions > Dump Database.
  - b. Click Dump Database. The file is written to /tmp/hue\_database\_dump.json on the host of the Hue server.
  - **c.** Log on to the *host of the Hue server* in a command-line terminal.
  - d. Edit /tmp/hue\_database\_dump.json by removing all objects with useradmin.userprofile in the model field. For example:

```
# Count number of objects
grep -c useradmin.userprofile /tmp/hue_database_dump.json
vi /tmp/hue_database_dump.json
  "pk": 1,
  "model": "useradmin.userprofile",
  "fields": {
    "last_activity": "2016-10-03T10:06:13",
    "creation_method": "HUE",
    "first_login": false,
    "user": 1,
    "home_directory": "/user/admin"
  "pk": 2,
  "model": "useradmin.userprofile",
  "fields": {
    "last_activity": "2016-10-03T10:27:10",
    "creation_method": "HUE",
    "first_login": false,
    "user": 2,
    "home_directory": "/user/alice"
},
```

#### 3. Connect to New Database

- a. Go to Hue > Configuration.
- **b.** Filter by category, **Database**.
- c. Set the following database parameters :

```
DB Hostname = <fqdn of host with postgres server>:3306
DB Type
            = <PostgreSQL>
DB Name
            = hue
Username
            = hue
Password
             = <hue database password set when granting hue permissions>
```

- d. Click Save Changes.
- 4. [migration only] Synchronize New Database
  - a. Select Actions > Synchronize Database
  - b. Click Synchronize Database.
- 5. [migration only] Load Data from Old Database

a. Log on to the host of the MySQL server in a command-line terminal.

```
mysql -u root -p
Enter password: <root password>
```

**b.** Drop the foreign key constraint (replace the ID value).

```
SHOW CREATE table hue.auth_permission;
ALTER TABLE hue.auth_permission DROP FOREIGN KEY content_type_id_refs_id_id value;
```

**c.** Clean the table, django\_content\_type.

```
DELETE FROM hue.django_content_type;
```

```
auth_permission | CREATE TABLE `auth_permission` (
 id int(11) NOT NULL AUTO_INCREMENT,
 'name` varchar(50) NOT NULL,
 content_type_id` int(11) NOT NULL,
 'codename' varchar(100) NOT NULL,
PRIMARY KEY ('id'),
UNIQUE KEY `content_type_id` (`content_type_id`, `codename`),
KEY `auth_permission_37ef4eb4` (`content_type_id`),
CONSTRAINT `content_type_id_refs_id_d043b34a` FOREIGN KEY (`content_type_id`) REFERENCES `django_content_type` ('id')
ENGINE=InnoDB AUTO_INCREMENT=217 DEFAULT CHARSET=latin1 |
```

- d. In Cloudera Manager, load the JSON file: select Actions > Load Database and click Load Database.
- e. Add the foreign key back:

```
ALTER TABLE hue.auth_permission ADD FOREIGN KEY (content_type_id) REFERENCES
django_content_type (id);
```

#### 6. Start Hue service

- **a.** Navigate to **Cluster > Hue**, if not already there.
- **b.** Select **Actions** > **Start**.
- c. Click Start.
- d. Click Hue Web UI to log on to Hue with a custom MySQL database.

# Connect Hue to PostgreSQL

If you have an external database installed, review Postgres Troubleshooting on page 28 before creating a database for Hue.

## Install and Configure PostgreSQL Server

Refer to the PostgreSQL documentation for more help on how to install a PostgreSQL database.

#### **Postgres Troubleshooting**

Pay close attention to these areas and revisit when troubleshooting:

- Python: Some Linux distributions need <a href="https://python-psycopg2">python-psycopg2</a> (for PostgreSQL). See the <a href="mailto:community thread">community thread</a>.
- Security: Delete anonymous users because they are able to log on without a password.
- Remote connections: The listen address should be set to 0.0.0.0 so it can listen to multiple hosts.
- Authentication: Configure pg hba.conf as follows (and change database/user as appropriate):

```
# TYPE
        DATABASE
                    USER
                                CIDR-ADDRESS
                                                       METHOD
local
        all
                    all
                                                       trust
                                                                 # Remote access
                                127.0.0.1/32
host
        all
                    all
                                                       password # IPv4
```

host host	all <b>hue_d</b>	all <b>hue_u</b>	::1/128 0.0.0.0/0	password # IPv6 <b>md5</b>	
--------------	---------------------	---------------------	----------------------	----------------------------	--

• Schemas: For private schemas, configure Django with the schema owner to DROP objects.

# Install PostgreSQL Server

1. Install and initialize the PostgreSQL server. The table lists the max version of each supported distribution for this CDH release, and corresponding default database versions.

**Table 7: Install Commands** 

os	OS Ver	DB Ver	Command
	7.3	9.2	sudo yum install postgresql-server sudo postgresql-setup initdb
CentOS / RHEL	6.8	8.4	sudo yum install postgresql-server sudo service postgresql initdb
	5.10	8.1	<pre>sudo yum install postgresql-server sudo /etc/init.d/postgresql start</pre>
	12.1, 12.2	9.4	zypper install postgresql postgresql-server systemctl start postgresql
SLES	11.4	8.4	# Refresh repo for python-psycopg2 zypper addrepo http://dw/datassacy/spicies/conditions/
	16.04	9.5	sudo apt-get install postgresql
Ubuntu	14.04	9.3	sudo apt-get install postgresql
	12.04	9.1	sudo apt-get install postgresql
Debian	8.4	9.4	sudo apt-get install postgresql
Debian	7.8	9.1	sudo apt-get install postgresql

**Tip:** If you need to start over, you can reinitialize:

```
rm -rf /var/lib/pgsql/*
<reinitialize per your os>
```

# Configure PostgreSQL Server

**1.** Configure pg hba.conf to set authentication methods:

# TYPE local	DATABASE all	USER all	CIDR-ADDRESS	METHOD <b>trust</b>	# Remote	access
host	all all	all all	127.0.0.1/32 ::1/128	password password	# IPv4 # IPv6	
host	hue_d	hue_u	0.0.0.0/0	md5	# IFVO	

• CentOS/RHEL/SLES:/var/lib/pgsql/data/pg\_hba.conf:

```
vi /var/lib/pgsql/data/pg_hba.conf
```

• Ubuntu/Debian: /etc/postgresql/<pgres version>/main/pg\_hba.conf:

```
\label{lem:conf}  \mbox{vi /etc/postgresql } \mbox{ | tail -1 | awk '{print $9}'`/main/pg_hba.conf} \\
```

**2.** Configure postgresql.conf to listen to all available addresses:

```
listen_addresses = '0.0.0.0'
```

 $\bullet \quad \textbf{CentOS/RHEL/SLES:} \\ / var/lib/pgsql/data/postgresql.conf$ 

```
vi /var/lib/pgsql/data/postgresql.conf
```

• Ubuntu/Debian: /etc/postgresql/<version>/main/postgresql.conf:

```
vi /etc/postgresql/`ls -l /etc/postgresql | tail -l | awk '{print $9}'`/main/postgresql.conf
```

3. Start (or restart) the database and enable automatic start on boot if necessary.

**Table 8: Restart Commands** 

os	OS Ver	Command
CentOS / RHEL	7.3	sudo systemctl restart postgresql sudo systemctl enable postgresql
	5.10, 6.8	sudo service postgresql restart sudo chkconfig postgresql on sudo chkconfig postgresqllist
SLES	12.1, 12.2	systemctl restart postgresql
	11.4	rcpostgresql restart
Ubuntu	12.04, 14.04, 16.04	sudo /etc/init.d/postgresql restart
Debian	7.8, 8.4	sudo /etc/init.d/postgresql restart

#### Create Hue Database



Important: If you use a private schema, you must configure Django to use the schema owner (which can be a user or group) to DROP objects, because <u>DROP is not a grantable permission in postgreSQL</u>.

1. Create the hue\_d database and grant privileges to the hue\_u user:

```
sudo -u postgres psql
postgres=# create database hue_d with lc_collate='en_US.UTF-8';
CREATE DATABASE
postgres=# create user hue_u with password 'huepassword';
CREATE ROLE
postgres=# grant all privileges on database hue_d to hue_u;
GRANT
```



**Note:** You can name the Hue database and user anything you like.

**2.** Verify the connection to the hue\_d database.

```
psql -h localhost -U hue_u -d hue_d
Password for user hue_u:
hue=> \q
```



Note: If you cannot connect, try typing the command manually. The hyphens may become corrupted when copied.

# Connect Hue Service to PostgreSQL

Tip: To save the data in your current database (embedded or external), you must migrate (dump, synch, load) before connecting to the new database. Otherwise, skip those steps.

1. Stop Hue Service

- **a.** In Cloudera Manager, navigate to **Cluster > Hue**.
- **b.** Select **Actions** > **Stop**.



**Note:** If necessary, refresh the page to ensure the Hue service is stopped: **①**.

- 2. [migration only] Dump Current Database
  - a. Select Actions > Dump Database.
  - b. Click Dump Database. The file is written to /tmp/hue\_database\_dump.json on the host of the Hue server.
  - **c.** Log on to the *host of the Hue server* in a command-line terminal.
  - d. Edit /tmp/hue\_database\_dump.json by removing all objects with useradmin.userprofile in the model field. For example:

```
# Count number of objects
grep -c useradmin.userprofile /tmp/hue_database_dump.json
vi /tmp/hue_database_dump.json
  "pk": 1,
  "model": "useradmin.userprofile",
  "fields": {
    "last_activity": "2016-10-03T10:06:13",
    "creation_method": "HUE",
    "first_login": false,
    "user": 1,
    "home_directory": "/user/admin"
  "pk": 2,
  "model": "useradmin.userprofile",
  "fields": {
    "last_activity": "2016-10-03T10:27:10",
    "creation_method": "HUE",
    "first_login": false,
    "user": 2,
    "home_directory": "/user/alice"
},
```

- 3. Connect to New Database
  - a. Go to Hue > Configuration.
  - **b.** Filter by category, **Database**.
  - c. Set the following database parameters :

```
DB Hostname = <fqdn of host with postgres server>:5432
DB Type
            = <PostgreSQL>
DB Name
            = hue_d
Username
            = hue_u
Password
             = <hue database password set when granting hue permissions>
```

- d. Click Save Changes.
- 4. [migration only] Synchronize New Database
  - a. Select Actions > Synchronize Database
  - b. Click Synchronize Database.
- 5. [migration only] Load Data from Old Database

a. Log on to the host of the PostgreSQL server in a command-line terminal.

```
psql -h localhost -U hue u -d hue d
Password for user hue_u: <hue user password>
```

**b.** Drop the foreign key constraint (replace the ID value).

```
hue=# \d auth_permission;
hue=# ALTER TABLE auth_permission DROP CONSTRAINT content_type_id_refs_id_id value;
```

**c.** Clean the table, django\_content\_type.

hue=# TRUNCATE django\_content\_type CASCADE;

```
ue=> \d auth_permission;
                                       Table "public.auth_permission"
     Column
                                                                         Modifiers
                                             I not null default nextval('auth_permission_id_seq'::regclass)
                 l integer
                  I character varying(50)
                                            I not null
content_type_id | integer
                  | character varying(100) | not null
Indexes:
    "outh_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)
    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_permissi
    TABLE "auth_user_user_permissions" CONSTRAINT "auth_user_user_permissions_permission_id_fkey" FOREIGN KEY (permission_id) REFERENCES auth
```

d. In Cloudera Manager, load the JSON file: select Actions > Load Database and click Load Database. **Tip:** If you are blocked by a duplicate key value such as this:

```
django.db.utils.IntegrityError: Problem installing fixture '/tmp/hue_database_dump.json':
Could not load desktop.DocumentTag(pk=1): duplicate key value violates unique constraint
 "desktop_documenttag_owner_id_1d5f76680ee9998b_uniq"
DETAIL: Key (owner_id, tag)=(1100713, default) already exists.
```

Delete that value and try loading again, for example:

```
DELETE FROM desktop_documenttag WHERE owner_id = '1100713' and tag = 'default';
```

**e.** Add the foreign key back (still logged on to the Hue database):

```
ALTER TABLE auth_permission ADD FOREIGN KEY (content_type_id) REFERENCES
django_content_type (id);
```

#### 6. Start Hue service

- **a.** Navigate to **Cluster > Hue**, if not already there.
- **b.** Select **Actions** > **Start**.
- c. Click Start.
- d. Click Hue Web UI to log on to Hue with a custom PostgreSQL database.

# Connect Hue to Oracle with Client Parcel

To connect to an Oracle database, Hue needs Oracle client libraries (Basic and SDK). These are available from Oracle as packages (zip files) or from Cloudera as a parcel (for CDH parcel deployments).

This page covers connecting with the Oracle client parcel.



**Important:** Currently, Cloudera only provides a parcel for the Oracle 11 client (which works with the Oracle 12 server). For the Oracle 12 client package (which can be used for either CDH parcel or package deployments), see <u>Connect Hue to Oracle with Client Package</u> on page 42.

# Install and Configure Oracle Server

Refer to the Oracle documentation for help on how to install an Oracle database.

Tip: Daniel Westermann has a helpful blog post: a simple script to automate the oracle 12c setup.

#### Set Environment Variables

1. Set all necessary Oracle environment variables. For example:

```
## Example Environment Variables
VERSION=12.1.0.2
ORACLE_HOSTNAME=<your hostname>
ORACLE_BASE=/ora01/app/oracle/product/base
ORACLE_HOME=${ORACLE_BASE}/${VERSION}
ORACLE_SID=orcl
ORACWNER_BIN=/home/oracle/bin
LD_LIBRARY_PATH=${ORACLE_HOME}/lib:${LD_LIBRARY_PATH}
```

**2.** Ensure that your shell .profile resembles:

```
## Example from /home/oracle/.bash_profile
TMP=/tmp
ORACLE_HOSTNAME=<your hostname>
ORACLE_BASE=/ora01/app/oracle/product/base
ORACLE_HOME=/ora01/app/oracle/product/base/12.1.0.2
ORACLE_SID=orcl
ORACWNER_BIN=/home/oracle/bin
LD_LIBRARY_PATH=${ORACLE_HOME}/lib:${LD_LIBRARY_PATH}
PATH=${ORACLE_HOME}/bin:${ORACWNER_BIN}:${PATH}
CLASSPATH=${ORACLE_HOME}/jlib:${ORACLE_HOME}/rdbms/jlib;
export ORACLE_HOSTNAME ORACLE_BASE ORACLE_HOME ORACLE_SID LD_LIBRARY_PATH PATH CLASSPATH
TMP
```

#### Configure Character Set

1. Log on as the oracle user:

```
su - oracle
```

**2.** Start the listener control (as user oracle):

```
$ORACLE_HOME/bin/lsnrctl start
```

**3.** Log on to SQL\*Plus:

```
sqlplus / as sysdba
```

**4.** Ensure character set is AL32UTF8 and national character set is UTF8:

```
SELECT * FROM v$nls_parameters where parameter like '%CHARACTERSET';
```

To update, **quit the shell** and run these commands in a SQL\*Plus script:

```
vi alter_charset.ddl
## Save in alter_charset.ddl (script takes 2-3 minutes)
CONNECT / as sysdba
```

```
SHUTDOWN immediate
STARTUP mount
ALTER SYSTEM ENABLE RESTRICTED SESSION;
ALTER SYSTEM SET JOB_QUEUE_PROCESSES=0 SCOPE = MEMORY;
ALTER SYSTEM SET AQ TM_PROCESSES=0 SCOPE = MEMORY;
ALTER DATABASE OPEN;
ALTER DATABASE CHARACTER SET AL32UTF8;
ALTER DATABASE NATIONAL CHARACTER SET INTERNAL USE UTF8;
SHUTDOWN immediate
STARTUP
sqlplus /nolog < alter_charset.ddl
```

#### Create Hue Database

1. Create the hue schema, set quotas, and grant select permissions (do not grant all): Tip: Oracle 12 users must ALTER session set to avoid creating a common user with prefix, c##.

```
vi create_hue_database.ddl
## Save in create_hue_database.ddl
## Change huepassword to something more secure
CONNECT / as sysdba
ALTER session set "_ORACLE_SCRIPT"=true;
DROP user hue cascade;
CREATE user hue identified by huepassword;
ALTER user hue quota 1000m on users;
ALTER user hue quota 100m on system;
GRANT create sequence to hue;
GRANT create session to hue;
GRANT create table to hue;
GRANT create view to hue;
GRANT create procedure to hue;
GRANT create trigger to hue;
GRANT execute on sys.dbms_crypto to hue;
GRANT execute on sys.dbms_lob to hue;
sqlplus /nolog < create_hue_database.ddl</pre>
```

**2.** Verify that you can connect to hue:

```
sqlplus hue/<your hue password>
```

3. Clean all hue user tables. Create a script to spool delete statements into a new file, delete\_from\_tables.ddl:

```
vi spool_statements.ddl
## Save in spool_statements.ddl (which generates delete_from_tables.ddl)
spool delete_from_tables.ddl
set pagesize 100;
SELECT 'DELETE FROM ' | table_name | ';' FROM user_tables;
commit;
spool off
quit
## Create delete_from_tables.ddl
sqlplus hue/<your hue password> < spool_statements.ddl
## Run delete_from_tables.ddl
sqlplus hue/<your hue password> < delete from tables.ddl
```

```
[oracle@oracle12c-centos68 ~]$ sqlplus hue/huepassword < spool_statements.ddl
SQL*Plus: Release 12.1.0.2.0 Production on Fri Mar 10 10:58:59 2017
Copyright (c) 1982, 2014, Oracle. All rights reserved.
Last Successful login time: Fri Mar 10 2017 10:54:46 -08:00
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing options
SQL> SQL> SQL>
 'DELETEFROM'||TABLE_NAME||';'
DELETE FROM AUTH_PERMISSION;
DELETE FROM AUTH_GROUP_PERMISSIONS;
DELETE FROM AUTH_GROUP;
DELETE FROM AUTH_USER_GROUPS;
DELETE FROM AUTH_USER_USER_PERMISSIONS;
DELETE FROM AUTH_USER;
DELETE FROM DJANGO_OPENID_AUTH_NONCE;
DELETE FROM DJANGO_OPENID_AUTH_ASSOCIATION;
[oracle@oracle12c-centos68 ~]$ sqlplus hue/huepassword < delete_from_tables.ddl
SQL*Plus: Release 12.1.0.2.0 Production on Fri Mar 10 10:59:07 2017
Copyright (c) 1982, 2014, Oracle. All rights reserved.
Last Successful login time: Fri Mar 10 2017 10:58:59 -08:00
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production With the Partitioning, OLAP, Advanced Analytics and Real Application Testing options
SQL> SP2-0734: unknown command beginning "SQL> set p..." - rest of line ignored. SQL> SP2-0734: unknown command beginning "SQL> SELEC..." - rest of line ignored. SQL> SQL> SQL> SP2-0734: unknown command beginning "'DELETEFRO..." - rest of line ignored.
 SQL> SQL>
228 rows deleted.
SQL>
0 rows deleted.
1 row deleted.
```

# Create Oracle Client Parcel Repository

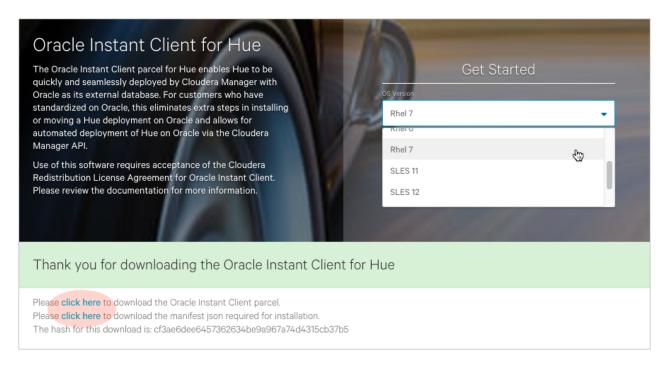
Cloudera provides the Oracle Instant Client for Hue (11.2 only) as a parcel for CDH parcel deployments.



Important: The Oracle 11 client works with the Oracle 12 server, but if you prefer the Oracle 12 client, see Connect Hue to Oracle with Client Package on page 42.



**Important:** There is currently no parcel support for Ubuntu 16 (xenial).



### Download and Stage Oracle Instant Client Parcel

- 1. Point a browser to https://www.cloudera.com/downloads/oracle instant client hue.html.
- 2. Select your OS and click Get It Now!
- 3. Check the box to accept Cloudera's Standard Licence Agreement and click Submit.
- 4. Download the parcel: ORACLE\_INSTANT\_CLIENT-11.2-1.oracleinstantclient1.0.0.p0.130-<your linux distro>.parcel.
- **5.** Download the manifest for the mirrored repository.
- **6.** Upload the parcel and manifest to the host with Cloudera Manager server, for example:

```
scp ORACLE_INSTANT_CLIENT-11.2-1* manifest.json root@<Cloudera Manager server hostname>:.
```

### Install Asynchronous I/O Library

- 1. Log on to the host of Cloudera Manager server.
- 2. Install the Asynchronous I/O library, libaio/libaio1:

```
## CentOS/RHEL (yum), SLES (zypper), Ubuntu/Debian (apt-get)
sudo yum install -y libaio
#sudo zypper install -y libaio
#sudo apt-get install -y libaio1
```

### Create Mirrored Parcel Repository

When manually adding parcels it is best to use mirrored repository as it preserves the metadata that enforces relation constraints.

1. Create a temporary repository , for example:

```
mkdir -pm 755 /var/www/html/cdh511
mv ~/ORACLE_INSTANT_CLIENT-11.2-1* ~/manifest.json /var/www/html/cdh511
```

2. Start a web server with any available port, for example:

```
cd /var/www/html/cdh511/
python -m SimpleHTTPServer 8900
```

**3.** Test the repository in a browser:

http://<server hostname>:8900/



### [Optional]

In fact, the Oracle parcel does not have any constraints, but using a repository allows you to more easily connect to an Oracle database during a new CDH installation if necessary. It is also a best practice and not more work.

However, if you have an existing CDH installation, you *can* simply copy the parcel (in this case) and add a corresponding SHA-1 file to /opt/cloudera/parcel-repo.

You must have CDH installed because the directory, parcel-repo, is created during step 6 of a CDH parcel installation.

```
sha1sum ORACLE_INSTANT_CLIENT-11.2-1.oracleinstantclient1.0.0.p0.130-<your linux
distro>.parcel | awk '{ print $1 }' >
ORACLE_INSTANT_CLIENT-11.2-1.oracleinstantclient1.0.0.p0.130-<your linux
distro>.parcel.sha1
mv ORACLE_INSTANT_CLIENT* /opt/cloudera/parcel-repo/
```

### Connect Hue Service to Oracle

You can connect Hue to your Oracle database while installing CDH (and Hue) or with an existing installation. With existing CDH installations, you can connect and restart Hue, without saving the data in your current database, or you can migrate the old data into Oracle.

### **New CDH Installation**

See <u>Install Hue 1-2-3</u> on page 6 to install Cloudera Manager (and its Installation Wizard), which you will use here to install CDH and the Oracle client.

### Install CDH and Oracle Parcel

- 1. Run the Cloudera Manager Installation Wizard to install CDH (and Hue): http://cm server hostname>:7180
- 2. Stop eat Select Repository to add the Oracle client parcel repository (Cluster Installation, step 1):
  - a. Choose Method Use Parcels and click More Options.
  - b. +

and add the URL for your Oracle Remote Parcel Repository:



c. Click Save Changes.

d. Select the newly added radio button by ORACLE\_INSTANT\_CLIENT and click Continue.



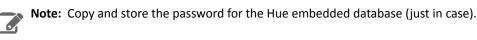
The Oracle parcel is downloaded, distributed, and activated at Cluster Installation, step <sup>6</sup> (Installing Selected Parcels).

Installing Selected Parcels					
he selected parcels are being downloaded and insta	alled on all the hosts in the cluste	er.			
> CDH 5.10.0-1.cdh5.10.0.p0.41	Downloaded: 100%	Distributed: 4/4 (93.4 MiB/s)	Unpacked: 4/4	Activated: 4/4	
➤ORACLE_INSTANT_CLIENT 11.2-1.oracleinst	Downloaded: 100%	Distributed: 4/4 (54.2 MiB/s)	Unpacked: 4/4	Activated: 4/4	

### Connect Hue to Oracle

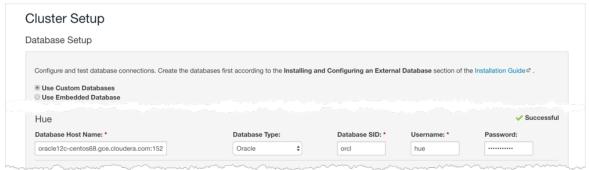
Continuing with Cloudera Manager Installation Wizard ...

- 1. Stop at Database Setup to set connection properties (Cluster Setup, step 3).
  - a. Select Use Custom Database.
  - **b.** Under **Hue**, set the connection properties to the Oracle database.



```
Database Hostname (and port): <fqdn of host with Oracle server>:1521
Database Type (or engine): Oracle
Database SID (or name): orcl
Database Username: hue
Database Password: <hue database password>
```

c. Click Test Connection and click Continue when successful.



2. Continue with the installation and click Finish to complete.

- **3.** Add support for a multi-threaded environment:
  - a. Go to Clusters > Hue > Configuration.
  - b. Filter by Category, Hue-service and Scope, Advanced.
  - c. Add support for a multi-threaded environment by setting Hue Service Advanced Configuration Snippet (Safety Valve) for hue\_safety\_valve.ini:

```
[desktop]
[[database]]
options={"threaded":true}
```

- d. Click Save Changes.
- 4. Restart the Hue service: select Actions > Restart and click Restart.
- 5. Log on to Hue by clicking Hue Web UI.

# Existing CDH Installation

**Activate Oracle Client Parcel** 

- 1. Log on to Cloudera Manager.
- 2. Go to the Parcels page by clicking Hosts > Parcels (or clicking the parcels icon 4).
- 3. Click the Configuration > Check for New Parcels.
- 4. Find ORACLE\_INSTANT\_CLIENT and click **Download**, **Distribute**, and **Activate**.



#### Connect Hue to Oracle

If you are not migrating the current (or old) database, simply connect to your new Oracle database and restart Hue (steps  $\frac{3}{2}$  on page 22 and  $\frac{6}{2}$  on page 22).

- 1. [migration only] Stop Hue Service
  - a. In Cloudera Manager, navigate to Cluster > Hue.
  - **b.** Select **Actions** > **Stop**.



**Note:** If necessary, refresh the page to ensure the Hue service is stopped: •.

- 2. [migration only] Dump Current Database
  - a. Select Actions > Dump Database.
  - $\textbf{b. Click Dump Database}. The file is written to \verb|/tmp/hue_database_dump.json| on the host of the Hue server.$
  - c. Log on to the host of the Hue server in a command-line terminal.
  - **d.** Edit /tmp/hue\_database\_dump.json by removing all objects with useradmin.userprofile in the model field. For example:

```
# Count number of objects
grep -c useradmin.userprofile /tmp/hue_database_dump.json

vi /tmp/hue_database_dump.json

{
   "pk": 1,
   "model": "useradmin.userprofile",
   "fields": {
        "last_activity": "2016-10-03T10:06:13",
        "creation_method": "HUE",
```

```
"first_login": false,
    "user": 1,
    "home_directory": "/user/admin"
  "pk": 2,
  "model": "useradmin.userprofile",
  "fields": {
    "last_activity": "2016-10-03T10:27:10",
    "creation_method": "HUE",
    "first_login": false,
"user": 2,
    "home_directory": "/user/alice"
},
```

### 3. Connect to New Database

- **a.** Configure Database connections:
  - Go to Hue > Configuration and filter by category, Database.
  - Set database properties and click Save Changes:

```
Hue Database Type (or engine): Oracle
Hue Database Hostname: <fqdn of host with Oracle server>
Hue Database Port: 1521
Hue Database Username: hue
Hue Database Password: <hue database password>
Hue Database Name (or SID): orcl
```

- **b.** Add support for a multi-threaded environment:
  - Filter by Category, Hue-service and Scope, Advanced.
  - Set Hue Service Advanced Configuration Snippet (Safety Valve) for hue\_safety\_valve.ini and click Save Changes:

```
[desktop]
[[database]]
options={"threaded":true}
```

- 4. [migration only] Synchronize New Database
  - a. Select Actions > Synchronize Database
  - b. Click Synchronize Database.
- 5. [migration only] Load Data from Old Database



Important: All user tables in the Hue database must be empty. You cleaned them at step 3 on page 43 of Create Hue Database on page 43. Ensure they are still clean.

sqlplus hue/<your hue password> < delete\_from\_tables.ddl

### 6. Re/Start Hue service

- a. Navigate to Cluster > Hue.
- **b.** Select **Actions** > **Start**, and click **Start**.
- c. Click Hue Web UI to log on to Hue with a custom Oracle database.

# Connect Hue to Oracle with Client Package

To connect to an Oracle database, Hue needs Oracle client libraries (Basic and SDK). These are available from Oracle as packages (zip files) or from Cloudera as a parcel (for CDH parcel deployments).

This page covers connecting with Oracle client packages.

# Install and Configure Oracle Server

Refer to the Oracle documentation for help on how to install an Oracle database.

Tip: Daniel Westermann has a helpful blog post: a simple script to automate the oracle 12c setup.

### Set Environment Variables

1. Set all necessary Oracle environment variables. For example:

```
## Example Environment Variables
VERSION=12.1.0.2
ORACLE_HOSTNAME=<your hostname>
ORACLE_BASE=/ora01/app/oracle/product/base
ORACLE_HOME=${ORACLE_BASE}/${VERSION}
ORACLE_SID=orcl
ORACWNER_BIN=/home/oracle/bin
LD_LIBRARY_PATH=${ORACLE_HOME}/lib:${LD_LIBRARY_PATH}
```

**2.** Ensure that your shell .profile resembles:

```
## Example from /home/oracle/.bash_profile
TMP=/tmp
ORACLE_HOSTNAME=<your hostname>
ORACLE_BASE=/ora01/app/oracle/product/base
ORACLE_HOME=/ora01/app/oracle/product/base/12.1.0.2
ORACLE_SID=orcl
ORAOWNER_BIN=/home/oracle/bin
LD_LIBRARY_PATH=${ORACLE_HOME}/lib:${LD_LIBRARY_PATH}
PATH=${ORACLE_HOME}/bin:${ORAOWNER_BIN}:${PATH}
CLASSPATH=${ORACLE_HOME}/jlib:${ORACLE_HOME}/rdbms/jlib;
export ORACLE_HOSTNAME ORACLE_BASE ORACLE_HOME ORACLE_SID LD_LIBRARY_PATH PATH CLASSPATH
TMP
```

### **Configure Character Set**

1. Log on as the oracle user:

```
su - oracle
```

2. Start the listener control (as user oracle):

```
$ORACLE_HOME/bin/lsnrctl start
```

**3.** Log on to SQL\*Plus:

```
sqlplus / as sysdba
```

**4.** Ensure character set is AL32UTF8 and national character set is UTF8:

```
SELECT * FROM v$nls_parameters where parameter like '%CHARACTERSET';
```

To update, **quit the shell** and run these commands in a SQL\*Plus script:

```
vi alter_charset.ddl
## Save in alter_charset.ddl (script takes 2-3 minutes)
CONNECT / as sysdba
SHUTDOWN immediate
STARTUP mount
ALTER SYSTEM ENABLE RESTRICTED SESSION;
ALTER SYSTEM SET JOB_QUEUE_PROCESSES=0 SCOPE = MEMORY;
ALTER SYSTEM SET AQ_TM_PROCESSES=0 SCOPE = MEMORY;
ALTER DATABASE OPEN;
ALTER DATABASE CHARACTER SET AL32UTF8;
ALTER DATABASE NATIONAL CHARACTER SET INTERNAL_USE UTF8;
SHUTDOWN immediate
STARTUP
sqlplus /nolog < alter_charset.ddl</pre>
```

### Create Hue Database

1. Create the hue schema, set quotas, and grant select permissions (do not grant all): Tip: Oracle 12 users must ALTER session set to avoid creating a common user with prefix, c##.

```
vi create_hue_database.ddl
## Save in create_hue_database.ddl
## Change huepassword to something more secure
CONNECT / as sysdba
ALTER session set "_ORACLE_SCRIPT"=true;
DROP user hue cascade;
CREATE user hue identified by huepassword;
ALTER user hue quota 1000m on users;
ALTER user hue quota 100m on system;
GRANT create sequence to hue;
GRANT create session to hue;
GRANT create table to hue;
GRANT create view to hue;
GRANT create procedure to hue;
GRANT create trigger to hue;
GRANT execute on sys.dbms_crypto to hue;
GRANT execute on sys.dbms_lob to hue;
sqlplus /nolog < create_hue_database.ddl</pre>
```

**2.** Verify that you can connect to hue:

```
sqlplus hue/<your hue password>
```

**3.** Clean all hue user tables. Create a script to spool delete statements into a new file, delete\_from\_tables.ddl:

```
vi spool_statements.ddl
## Save in spool_statements.ddl (which generates delete_from_tables.ddl)
{\tt spool \ delete\_from\_tables.ddl}
set pagesize 100;
SELECT 'DELETE FROM ' | | table_name | | ';' FROM user_tables;
commit;
```

```
spool off
quit

## Create delete_from_tables.ddl
sqlplus hue/<your hue password> < spool_statements.ddl

## Run delete_from_tables.ddl
sqlplus hue/<your hue password> < delete_from_tables.ddl</pre>
```

[oracle@oracle12c-centos68 ~]\$ sqlplus hue/huepassword < spool\_statements.ddl

```
SQL*Plus: Release 12.1.0.2.0 Production on Fri Mar 10 10:58:59 2017
Copyright (c) 1982, 2014, Oracle. All rights reserved.
Last Successful login time: Fri Mar 10 2017 10:54:46 -08:00
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing options
SQL> SQL> SQL>
'DELETEFROM'||TABLE_NAME||';'
DELETE FROM AUTH_PERMISSION;
DELETE FROM AUTH_GROUP_PERMISSIONS;
DELETE FROM AUTH_GROUP;
DELETE FROM AUTH_USER_GROUPS;
DELETE FROM AUTH_USER_USER_PERMISSIONS;
DELETE FROM AUTH_USER;
DELETE FROM DJANGO_OPENID_AUTH_NONCE;
DELETE FROM DJANGO_OPENID_AUTH_ASSOCIATION;
[oracle@oracle12c-centos68 ~]$ sqlplus hue/huepassword < delete_from_tables.ddl
SQL*Plus: Release 12.1.0.2.0 Production on Fri Mar 10 10:59:07 2017
Copyright (c) 1982, 2014, Oracle. All rights reserved.
Last Successful login time: Fri Mar 10 2017 10:58:59 -08:00
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production With the Partitioning, OLAP, Advanced Analytics and Real Application Testing options
SQL> SP2-0734: unknown command beginning "SQL> set p..." - rest of line ignored. SQL> SP2-0734: unknown command beginning "SQL> SELEC..." - rest of line ignored. SQL> SQL> SP2-0734: unknown command beginning "'DELETEFRO..." - rest of line ignored.
SQL> SQL>
228 rows deleted.
SOL>
0 rows deleted.
SOL>
1 row deleted.
```

# Install Oracle Client Package

Cloudera Manager requires the Oracle instant client libraries to be in /usr/share/oracle/instantclient/lib/. The following commands arrange the files as such.



**Important:** You must add client libraries to each machine that hosts a Hue server.

### Install Asynchronous I/O Library

- 1. Log on to the host of Cloudera Manager server.
- 2. Install the Asynchronous I/O library, libaio/libaio1:

```
## CentOS/RHEL (yum), SLES (zypper), Ubuntu/Debian (apt-get)
sudo yum install -y libaio
```

```
#sudo zypper install -y libaio
#sudo apt-get install -y libaio1
```

#### Install Oracle Client

- 1. Download zip files for Instant Client Package, Basic and SDK (with headers).
- 2. For this step, switch to the host with the downloaded files and upload zip to the Cloudera Manager server host:

scp instantclient-\*.zip root@<CM server hostname>:.

#### Version 12.1.0.2.0

Instant Client Package - Basic: All files required to run OCI, OCCI, and JDBC-OCI applications

- instantclient-basic-linux.x64-12.1.0.2.0.zip (63,352,239 bytes) (cksum 109893216)
- oracle-instantclient12.1-basic-12.1.0.2.0-1.x86\_64.rpm (62,587,782 bytes) (cksum -2840691603)

Instant Client Package - SDK: Additional header files and an example makefile for developing Oracle applications with Instant Client

- instantclient-sdk-linux.x64-12.1.0.2.0.zip (667,174 bytes) (cksum 1047596065)
- oracle-instantclient12.1-devel-12.1.0.2.0-1.x86 64.rpm (634,803 bytes) (cksum -2599726994)
- **3.** Arrange the client libraries to mirror the tree structure in the image. Here is *one way* to do this:

```
# Create nested directories: /usr/share/oracle/instantclient/lib/
mkdir -pm 755 /usr/share/oracle/instantclient/lib
# Unzip. The files expand into /usr/share/oracle/instantclient/instantclient_<ver>/
unzip '*.zip' -d /usr/share/oracle/instantclient/
# Move lib files from instantclient_<ver> to /usr/share/oracle/instantclient/lib/
mv /usr/share/oracle/instantclient/`ls -l /usr/share/oracle/instantclient/ | grep instantclient_ | awk '{print $9}'`/lib* /usr/share/oracle/instantclient/lib/
# Move rest of the files to /usr/share/oracle/instantclient/
mv /usr/share/oracle/instantclient/`ls -l /usr/share/oracle/instantclient/ | grep
instantclient_ | awk '{print $9}'`/* /usr/share/oracle/instantclient/
# Create symbolic links. Remember to edit version numbers as necessary
cd /usr/share/oracle/instantclient/lib
ln -s libclntsh.so.12.1 libclntsh.so
ln -s libocci.so.12.1 libocci.so
```

```
[root@test2-ec2-rhel73-cdh5100-1 instantclient]# tree /usr/share/oracle/instantclient/lib -C

    libclntshcore.so.12.1

  - libclntsh.so -> libclntsh.so.12.1
   libclntsh.so.12.1
  - libipc1.so
  - libmql1.so
   libnnz12.so
  - libocci.so -> libocci.so.12.1
    libocci.so.12.1
   · liboramysql12.so
```

**4.** Set \$ORACLE\_HOME and \$LD\_LIBRARY\_PATH:

```
export ORACLE_HOME=/usr/share/oracle/instantclient
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ORACLE_HOME
```



**Note:** If using the Oracle 11 instant client you are ready to Connect Hue to Oracle. Else if using the Oracle 12 instant client, upgrade the Python module, cx\_Oracle.

### Apply Temporary Workaround for Oracle 12 Client

Update the cx\_Oracle package in your native Python environment and copy it to Hue's Python environment.

1. Install gcc and Python development tools:

```
## CentOS/RHEL (yum), SLES (zypper), Ubuntu/Debian (apt-get)
yum install -y python-setuptools python-devel gcc
#zypper install -y python-setuptools python-devel gcc
#apt-get install -y python-setuptools python-dev
```

2. Install pip:

```
easy_install pip
```

**3.** Install cx\_Oracle. Ensure that ORACLE\_HOME and \$LB\_LIBRARY\_PATH are properly set so that pip knows which version to install.

```
echo $ORACLE_HOME $LD_LIBRARY_PATH
pip install cx_Oracle
```

Tip: You can also wget the proper cx\_Oracle file yourself: <a href="https://pypi.python.org/pypi/cx\_Oracle/">https://pypi.python.org/pypi/cx\_Oracle/</a>.

- **4.** Get the version of the new cx\_Oracle package:
  - CentOS/RHEL and SLES:

```
ls /usr/lib64/python2.7/site-packages/cx_Oracle*
```

• Ubuntu/Debian:

```
ls /usr/local/lib/python2.7/dist-packages/cx_Oracle*
```

- 5. If this is a New CDH Installation on page 47, stop here to run the first 5 or 6 steps of the Cloudera Manager Installation Wizard (packages= 5/parcels= 6). Do not go past Cluster Installation.
- **6.** Navigate to Hue's python environment, \$HUE\_HOME/build/env/lib/<python version>/site-packages.
  - CDH Parcel installation:

```
cd /opt/cloudera/parcels/`ls -l /opt/cloudera/parcels | grep CDH | tail -l | awk '{print $9}'`/lib/hue/build/env/lib/python2.7/site-packages
```

• CDH package installation:

cd /usr/lib/hue/build/env/lib/python2.7/site-packages



**Important:** The parcel path is created during step 5 or 6 of **Cluster Installation**, so you must have completed this to continue.

7. Move the existing cx\_Oracle file:

```
mv cx_Oracle-5.2.1-py2.7-linux-x86_64.egg cxfoo
```

**8.** Copy the new cx\_Oracle module to Hue's python environment. The version can change:

CentOS/RHEL and SLES:

```
cp -a /usr/lib64/python2.7/site-packages/cx_Oracle-5.3-py2.7.egg-info .

    Ubuntu/Debian
```

cp -a /usr/local/lib/python2.7/dist-packages/cx\_Oracle-5.3.egg-info .

### Connect Hue Service to Oracle

You can connect Hue to your Oracle database while installing CDH (and Hue) or with an existing installation. With existing CDH installations, you can connect and restart Hue, without saving the data in your current database, or you can migrate the old data into Oracle.

### **New CDH Installation**

See Install Hue 1-2-3 on page 6 to install Cloudera Manager (and its Installation Wizard), which you will use here to install CDH and the Oracle client.

- 1. Run the Cloudera Manager Installation Wizard to install CDH (and Hue): http://cm server hostname>:7180
- 2. Stop at the end of Cluster Installation to copy the latest cx\_Oracle package into Hue's Python environment.

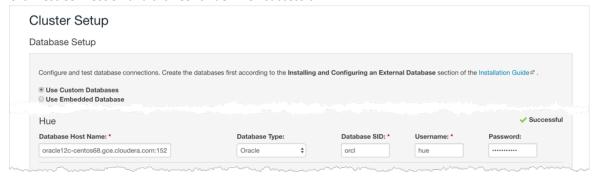


- Stop at Database Setup to set connection properties (Cluster Setup, step 3).
  - Select Use Custom Database.
  - **b.** Under **Hue**, set the connection properties to the Oracle database.

Note: Copy and store the password for the Hue embedded database (just in case).

```
Database Hostname (and port): <fqdn of host with Oracle server>:1521
Database Type (or engine): Oracle
Database SID (or name): orcl
Database Username: hue
Database Password: <hue database password>
```

c. Click **Test Connection** and click **Continue** when successful.



- 4. Continue with the installation and click Finish to complete.
- **5.** Add support for a multi-threaded environment:

- a. Go to Clusters > Hue > Configuration.
- b. Filter by Category, Hue-service and Scope, Advanced.
- c. Add support for a multi-threaded environment by setting **Hue Service Advanced Configuration Snippet** (Safety Valve) for hue\_safety\_valve.ini:

```
[desktop]
[[database]]
options={"threaded":true}
```

- d. Click Save Changes.
- 6. Restart the Hue service: select Actions > Restart and click Restart.
- 7. Log on to Hue by clicking Hue Web UI.

### **Existing CDH Installation**

If you are not migrating the current (or old) database, simply connect to your new Oracle database and restart Hue (steps 3 on page 22 and 6 on page 22).

- 1. [migration only] Stop Hue Service
  - a. In Cloudera Manager, navigate to Cluster > Hue.
  - **b.** Select **Actions** > **Stop**.



**Note:** If necessary, refresh the page to ensure the Hue service is stopped: ①.

- 2. [migration only] Dump Current Database
  - a. Select Actions > Dump Database.
  - **b.** Click **Dump Database**. The file is written to /tmp/hue\_database\_dump.json on the host of the Hue server.
  - **c.** Log on to the *host of the Hue server* in a command-line terminal.
  - **d.** Edit /tmp/hue\_database\_dump.json by removing all objects with useradmin.userprofile in the model field. For example:

```
# Count number of objects
grep -c useradmin.userprofile /tmp/hue_database_dump.json
vi /tmp/hue_database_dump.json
{
  "pk": 1,
  "model": "useradmin.userprofile",
  "fields": {
    "last_activity": "2016-10-03T10:06:13",
    "creation_method": "HUE",
    "first_login": false,
"user": 1,
    "home_directory": "/user/admin"
  "pk": 2,
"model": "useradmin.userprofile",
  "fields": {
    "last_activity": "2016-10-03T10:27:10",
    "creation_method": "HUE",
    "first_login": false,
    "user": 2,
    "home_directory": "/user/alice"
},
```

#### 3. Connect to New Database

a. Configure Database connections: Go to Hue > Configuration, filter by Database, set properties, and click Save

```
Hue Database Type (or engine): Oracle
Hue Database Hostname: <fqdn of host with Oracle server>
Hue Database Port: 1521
Hue Database Username: hue
Hue Database Password: <hue database password>
Hue Database Name (or SID): orcl
```

b. Add support for a multi-threaded environment: Filter by Hue-service, set Hue Service Advanced Configuration Snippet (Safety Valve) for hue\_safety\_valve.ini, and click Save Changes:

```
[desktop]
[[database]]
options={"threaded":true}
```

- 4. [migration only] Synchronize New Database
  - a. Select Actions > Synchronize Database
  - b. Click Synchronize Database.
- 5. [migration only] Load Data from Old Database



Important: All user tables in the Hue database must be empty. You cleaned them at step 3 on page 43 of Create Hue Database on page 43. Ensure they are still clean.

```
sqlplus hue/<your hue password> < delete_from_tables.ddl
```

## 6. Re/Start Hue service

- a. Navigate to Cluster > Hue.
- **b.** Select **Actions** > **Start**, and click **Start**.
- c. Click Hue Web UI to log on to Hue with a custom Oracle database.

# Migrate Hue Database



Note: Hue Custom Databases includes database-specific pages on how to migrate from an old to a new database. This page summarizes across supported database types.

When you change Hue databases, you can migrate the existing data to your new database. If the data is dispensable, there is no need to migrate.

The Hue database stores things like user accounts, Hive queries, and Oozie workflows, and you may have accounts, queries, and workflows worth saving. See How to Populate the Hue Database on page 56.

Migrating your existing database currently requires some work-arounds (in parentheses):

- Stop the Hue service.
- Dump database (and delete "useradmin.userprofile" objects from .json file).
- · Connect to new database.
- Synchronize database (and drop foreign key to clean tables).
- Load database (and add foreign key).
- Start Hue service.

# **Dump Database**

- 1. In the **Hue Web UI**, click the home icon 6 to see what documents you are migrating.
- 2. In Cloudera Manager, stop the Hue service: go to Hue and select Actions > Stop.



**Note:** Refresh the page to ensure that the Hue service is stopped: •.

- 3. Select Actions > Dump Database and click Dump Database. The file is written to /tmp/hue\_database\_dump.json on the host of the Hue server.
- 4. Log on to the host of the *Hue server* in a command-line terminal. You can find the hostname on the Dump Database window and at **Hue** > **Hosts**.
- 5. Edit /tmp/hue\_database\_dump.json by removing all objects with useradmin.userprofile in the model field. For example:

```
# Count number of objects
grep -c useradmin.userprofile /tmp/hue_database_dump.json
vi /tmp/hue_database_dump.json
  "pk": 1,
  "model": "useradmin.userprofile",
  "fields": {
    "last_activity": "2016-10-03T10:06:13",
    "creation_method": "HUE",
    "first_login": false,
    "user": 1,
    "home_directory": "/user/admin"
},
```

### Connect New Database

In Cloudera Manager, connect Hue to the new database. See <u>Hue Custom Databases</u> for help on installing and configuring a custom database.

- 1. Go to Hue > Configuration.
- 2. Filter by category, **Database**.
- **3.** Set the appropriate database parameters :

```
Hue Database Type: MySQL or PostgreSQL or Oracle
Hue Database Hostname: <fqdn of host with database server>
Hue Database Port: 3306 or 5432 or 1521
Hue Database Username: <hue database username>
Hue Database Password: <hue database password>
Hue Database Name: <hue database name or SID>
```

- 4. Click Save Changes.
- **5. Oracle users only** should add support for a multithreaded environment:
  - a. Filter by Category, Hue-service and Scope, Advanced.
  - b. Add support for a multithreaded environment by setting Hue Service Advanced Configuration Snippet (Safety Valve) for hue\_safety\_valve.ini:

```
[desktop]
[[database]]
options={"threaded":True}
```

c. Click Save Changes.

# Synchronize and Load

- 1. Synchronize: select Actions > Synchronize Database and click Synchronize Database.
- 2. Log on to the host of the *database* server in a command-line terminal and clean tables:
  - MySQL and PostgreSQL on page 52 users remove a foreign key from auth.permission and clean django\_content\_type.
  - Oracle on page 52 users delete content from all tables.
- 3. Load: select Actions > Load Database and click Load Database.
- 4. Return to the host of the database server:
  - MySQL and PostgreSQL on page 52 users add the foreign key to auth\_permission.
- **5. Start**: select **Actions** > **Start** and click **Start**.



**Note:** Refresh the page to ensure that the Hue service is running: •.

6. In the **Hue Web UI**, click the home icon 6 to ensure that all documents were migrated.

### MariaDB / MySQL

- 1. Synchronize Database in Cloudera Manager.
- 2. Log on to MySQL:

```
mysql -u root -p
Enter password: <root password>
```

**3.** Drop the foreign key constraint from auth\_permission:

```
SHOW CREATE table hue.auth_permission;
ALTER TABLE hue.auth_permission DROP FOREIGN KEY
content_type_id_refs_id_<id value>;
```

```
mysql> ALTER TABLE hue.auth_permission DROP FOREIGN KEY content_type_id_refs_id_d043b34a;
Query OK, 228 rows affected (0.01 sec)
Records: 228 Duplicates: 0 Warnings: 0
```

**4.** Delete the contents of django\_content\_type:

DELETE FROM hue.django\_content\_type;

```
mysql> DELETE FROM hue.django_content_type;
Query OK, 76 rows affected (0.00 sec)
```

- 5. Load Database in Cloudera Manager.
- **6.** Add the foreign key, content\_type\_id, to auth\_permission:

ALTER TABLE hue.auth\_permission ADD FOREIGN KEY (content\_type\_id) REFERENCES django\_content\_type (id);

```
mysql> ALTER TABLE hue.auth_permission ADD FOREIGN KEY (content_type_id) REFERENCES django_content_type (id);
Query OK, 228 rows affected (0.01 sec)
Records: 228 Duplicates: 0 Warnings: 0
```

7. Start Hue in Cloudera Manager.

### **PostgreSQL**

- 1. Synchronize Database in Cloudera Manager.
- 2. Log on to PostgreSQL:

```
psql -h localhost -U hue -d hue
Password for user hue:
```

**3.** Drop the foreign key constraint from auth\_permission:

```
\d auth_permission;
ALTER TABLE auth_permission DROP CONSTRAINT content_type_id_refs_id_<id value>;
```

**4.** Delete the contents of django\_content\_type:

```
TRUNCATE django_content_type CASCADE;
```

- 5. Load Database in Cloudera Manager.
- **6.** Add the foreign key, content\_type\_id, to auth\_permission:

```
ALTER TABLE auth_permission ADD FOREIGN KEY (content_type_id) REFERENCES django_content_type(id) DEFERRABLE INITIALLY DEFERRED;
```

7. Start Hue in Cloudera Manager.

#### Oracle

Oracle users should delete all content from the Oracle tables after synchronizing and before loading:

- 1. Synchronize Database in Cloudera Manager.
- 2. Log on to Oracle:

```
su - oracle
sqlplus / as sysdba
```

**3.** Grant a quota to the tablespace where tables are created (the default is SYSTEM). For example:

```
ALTER USER hue quota 100m on system;
```

4. Log on as the hue:

```
sqlplus hue/<hue password>
```

5. Create a spool script that creates a delete script to clean the content of all tables.

```
vi spool_statements.ddl

## Save in spool_statements.ddl (which generates delete_from_tables.ddl)
spool delete_from_tables.ddl
set pagesize 100;
SELECT 'DELETE FROM ' || table_name || ';' FROM user_tables;
commit;
spool off
quit
```

6. Run both scripts:

```
## Create delete_from_tables.ddl
sqlplus hue/<your hue password> < spool_statements.ddl</pre>
```

```
## Run delete_from_tables.ddl
sqlplus hue/<your hue password> < delete_from_tables.ddl
```

- 7. Load Database in Cloudera Manager.
- 8. Start Hue in Cloudera Manager.

### **Hue Custom Database Tutorial**

This page explains how to configure Hue with a custom database from end to end by migrating your existing database and synching to a new custom database. Learn how to switch databases for:

- A new installation of CDH, with the Cloudera Manager Installation Wizard
- An existing installation of CDH, with the Cloudera Manager Admin Console.



Note: On this page we use CentOS 6 with MySQL. For instructions on other platforms and databases, see Hue Databases.

### **Prepare Hosts**

Create, or prepare, five machines, each with CentOS 6 and at least 8 GB of RAM:

- 1. Create a cluster of four machines. Name them cdh-cluster-[1-4].<your domain>.com.
- 2. Create one machine for the database. Name it cdh-db. <your domain > . com.

Separating the database from the CDH cluster is a best practice, but if necessary, you can install it on one of the hosts in the cluster (for example, cdh-cluster-1).

### **Install Custom Database**

Install MySQL on the single machine you designated for this purpose (cdh-db.<your domain>.com).

1. Install MySQL server on cdh-db.domain.com:

```
sudo yum install -y mysql-server
```

**2.** Start the server:

```
sudo service mysqld start
```

**3.** Secure your installation:

```
sudo /usr/bin/mysql_secure_installation
Enter current password for root (enter for none): [Press Enter if the password is unset]
OK, successfully used password, moving on...
[\ldots]
Set root password? [Y/n] Y
New password:
Re-enter new password:
Remove anonymous users? [Y/n] Y
Disallow root login remotely? [Y/n] N
[\ldots]
Remove test database and access to it [Y/n] Y
Reload privilege tables now? [Y/n] Y
```

**4.** Configure /etc/my.cnf:

```
[mysqld]
...
bind-address=0.0.0.0
default-storage-engine=innodb
sql_mode=STRICT_ALL_TABLES
```

5. Restart the server

```
sudo service mysqld restart
```

**6.** Log on with your new root password:

```
mysql -u root -p<root password>
```

7. Create the hue database with UTF8 collation and configure the hue user (with your own password):

```
create database hue collate = 'utf8_general_ci';
grant all on hue.* to 'hue'@'%' identified by 'huepassword';
quit
```

### Install CM and CDH

In this section, we test connecting to a custom database with the installation wizard; then we undo the connection so we can connect with the admin console in <a href="Dump, Synchronize">Dump, Synchronize</a>, and <a href="Load">Load</a> on page 55.

Run Easy Install on page 6 on one machine in the cluster (and stop at the Database Setup page). As a reminder:

1. Install Cloudera Manager on on cdh-cluster-1.<your domain>.com:

```
## Download Cloudera Manager to your package manager source directory.
wget https://archive.cloudera.com/cm5/redhat/6/x86_64/cm/cloudera-manager.repo -P
/etc/yum.repos.d/

## Install Cloudera Manager and Dependencies (sourced from the Cloudera Manager repo)
sudo yum install -y oracle-j2sdk1.7
sudo yum install -y cloudera-manager-daemons cloudera-manager-server
sudo yum install -y cloudera-manager-server-db-2

## Start the database and server
sudo service cloudera-scm-server-db start
sudo service cloudera-scm-server start
```

- 2. Log on to Cloudera Manager at http://cdh-cluster-1.<your domain>.com:7180/.
- **3.** Follow the <u>installation wizard</u> to the **Database Setup** page.
- **4.** Connect to your custom database, then reconnect to the embedded database before moving on:
  - a. Select Custom Database, set parameters, and Test Connection and stop 

     I

```
DB Hostname = cdh-db.<your domain>.com:3306

DB Type = MySQL

DB Name = hue

Username = hue

Password = <hue password set when granting hue permissions>
```

- b. After successfully connecting to MySQL, reset the hostname and database type to PostgreSQL.
- c. Select Embedded Database and again, click Test Connection.



**Note:** Before moving on, click **Show Passwords** and copy the Hue password for safe-keeping.

- **d.** After successfully connecting to the embedded database, click **Continue**. We reconnect to MySQL in <u>Dump</u>, <u>Synchronize</u>, <u>and Load</u> on page 55, after migrating the embedded database.
- 5. Finish the wizard, go to the Hue service and click Hue Web UI to open Hue.
- 6. Assign a username/password and log on.

# Populate Database (optional)

<u>Populate the Hue database</u> with user account information, a Hive query, and an Oozie workflow (to ensure that the database migration works).

### Dump, Synchronize, and Load

To connect to other supported databases, see <u>Hue Custom Databases</u>.

1. Stop the Hue service: go to **Hue** and select **Actions** > **Stop**.



**Note:** Refresh the page if the Hue service does not look stopped: •.

- 2. Dump the existing database:
  - a. Select Actions > Dump Database.
  - b. Click Dump Database. The file is written to /tmp/hue\_database\_dump. json on the host of the Hue server.
  - **c.** Log on to the *host of the Hue server* in a command-line terminal.
  - **d.** Edit /tmp/hue\_database\_dump.json by removing all objects with useradmin.userprofile in the model field. For example:

```
# Count number of objects
grep -c useradmin.userprofile /tmp/hue_database_dump.json
vi /tmp/hue_database_dump.json
  "pk": 1,
  "model": "useradmin.userprofile",
  "fields": {
    "last_activity": "2016-10-03T10:06:13",
    "creation_method": "HUE",
    "first_login": false,
"user": 1,
    "home_directory": "/user/admin"
  }
  "pk": 2,
  "model": "useradmin.userprofile",
  "fields": {
    "last_activity": "2016-10-03T10:27:10",
    "creation_method": "HUE",
    "first_login": false,
    "user": 2,
    "home_directory": "/user/alice"
},
```

- **3.** Connect Hue to the new MySQL database:
  - a. Go to Hue > Configuration.
  - b. Filter by category, Database.

c. Set the following database parameters:

```
DB Hostname = <fqdn of host with postgres server>:3306

DB Type = <PostgreSQL>

DB Name = hue

Username = hue

Password = <hue database password set when granting hue permissions>
```

- d. Click Save Changes.
- 4. Synchronize the new database: select Actions > Synchronize Database and click Synchronize Database.
- **5.** Load the database after removing the foreign key constraint:
  - **a.** Log on to the *host of the MySQL server* in a command-line terminal.
  - **b.** Delete the foreign key constraint and clean the table, django\_content\_type:

```
mysql -u root -p

SHOW CREATE table hue.auth_permission;
ALTER TABLE hue.auth_permission DROP FOREIGN KEY content_type_id_refs_id_<input id>;

DELETE FROM hue.django_content_type;
```

- c. In Cloudera Manager, load the JSON file: select Actions > Load Database and click Load Database.
- d. Add the foreign key back:

```
ALTER TABLE hue.auth_permission ADD FOREIGN KEY (content_type_id) REFERENCES django_content_type (id);
```

**6.** Start the Hue service: select **Actions** > **Start** and click **Start**. If you went through <u>Use Hue</u>, ensure your data was migrated properly.

# How to Populate the Hue Database

Not every action in the Hue UI touches the Hue database (embedded or custom). This page explains how to populate the database with user account information, Hive queries, and Oozie workflows. This is useful when testing the <u>migration of a database</u>.

- 1. Add New User (Alice)
  - a. Log on to Hue as the administrator.
  - b. Open the Administration drop down and select Manage Users
  - c. Click Add user and follow the three steps.
    - Add a username (for example, "Alice") and password and click **Next**.
    - Ensure Alice belongs to the default group and click Next.
    - Give Alice Superuser status (for Hue, not HDFS) and click Add user.
  - d. Log out as the administrator and log on as Alice.

### 2. Save Hive Query (customers.sql)

- a. Go to About Hue > Quick Start by clicking the eve logo.
- b. Click the Examples tab ("Step 2").
- c. Click download Hive to install sample databases.
- d. Go to the Metastore Manager (or Data Browser > Metastore Tables).
- e. Click the default database and customers (sample) table.
- f. Click Browse Data to automatically generate a select \* query in the Hive editor.
- g. Run the query with your cursor in the editor and CTRL + Enter, or by clicking the Run icon
- **h.** Save the query as customers.sql by clicking the **Save** icon ...
- i. View the guery on the Saved Queries tab in the Hive editor.

### 3. Save Oozie Workflow (Customers Workflow)

- a. Go to Oozie by selecting, Workflows > Editors > Workflows.
- b. Click the Create button.
- c. Rename "My Workflow" as "Customers Workflow" and click the Save icon
- **d.** Drag the action icon for **Saved Hive Query** of to the field, "Drop your action here."
- e. Select a saved query (customers.sql) from the drop down and click Add.
- **f.** Save the workflow by clicking the **Save**  $\square$  icon.
- g. Submit the workflow by clicking the icon and clicking Submit. You should see the workflow status change to SUCCEEDED
- **h.** View the saved workflow (and all documents) by clicking the home icon  $\stackrel{\frown}{\alpha}$ .

# **Hue Administration**

This section consolidates administration and configuration documents related to Hue that live across the Cloudera document set.

- Supported Browsers for Hue
- Administering Hue
- Adding a Hue Service and Role Instance
- Enabling Hue Applications Using Cloudera Manager
- Managing Hue Analytics Data Collection
- Configuring CDH Components for Hue
- Hue Configuration
- Using Hue with Cloudera Search

# **Hue Security**

This section consolidates security documents related to Hue that live across the Cloudera document set.

- <u>Hue Authentication</u>
- Configuring Kerberos Authentication for Hue
- Integrating Hue with LDAP
- Configuring Hue for SAML
- Configuring TLS/SSL for Hue
- Hue High Availability
- Configuring Other CDH Components to Use HDFS HA

# **Hue How-tos**

Watch this space for more Hue How-tos!

### How to Add a Hue Load Balancer

- 1. Log on to cloudera MANAGER and click Hue.
- 2. Select Actions > Add Role Instances.
- 3. Add 1 Load Balancer:
  - a. Click Select hosts in the field under Load Balancer ...
  - b. Select a host and click OK.
- **4.** [Optional] Add 2 additional Hue servers (for a total of 3) to boost performance:
  - a. Click Select hosts in the field under Hue Server ...
  - b. Select a host and click OK > Continue.
- **5.** Check the boxes for the new servers and load balancer.
- 6. Select Actions for Selected > Start > Start.



Note: Hue servers can share hosts with Load Balancers. But Hue servers must be on distinct hosts from other Hue servers, and Load Balancers must be on distinct hosts from other Load Balancers.

- 7. Click Save Changes and Restart Hue.
- 8. Click Hue Web UI > Load Balanced Hue Web UI.
- **9.** Log on to Hue and ensure the port is 8889.

Tip: The Load Balancer instance can always be accessed on the Hue Instances tab.

# How to Enable SQL Editor Autocompleter in Hue

Autocompleter provides finely tuned SQL suggestions for Hive and Impala dialects. See Brand new Autocompleter for Hive and Impala.

Autocompleter is enabled by default. To manually enable or disable, use the Enable Autocompleter flag.

- 1. Log on to Hue and go to either the Hive or Impala editor.
- 2. Place your cursor in the editor window.
- 3. Open the Autocompleter settings panel with the shortcut, command-, (Mac) or Ctrl-, (Windows). Do not miss the comma.

Tip: Type ? (anywhere but the active editor) to open a menu of Editor keyboard shortcuts.

- **4.** To **Enable Autocompleter**, check the box. To disable, uncheck the box.
- 5. To Enable Live Autocompletion, check the box. To disable, uncheck the box.
  - **Tip:** To use Autocompleter with Live Autocompletion *off*, use **Ctrl + Space key**.
- 6. Place your cursor in the editor window to close the panel. Autocompleter is now turned on or off based on your flag setting.



# How to Enable and Use Navigator in Hue

This page assumes that Cloudera Navigator was installed, either during CDH installation, or by adding roles for the Metadata and Audit servers to the Cloudera Management Service.

Figure 1: Include Cloudera Navigator when Installing CDH

This wizard will also install the Cloudera Management Service. These are a set of components that enable monitoring, reporting, events, and alerts; these components require databases to store information, which will be configured on the next page. ✓ Include Cloudera Navigator Please ensure that you have the appropriate license for Cloudera Navigator or contact Cloudera for assistance . Some actions (for example, those performed by specific, internal users) are discarded by default. You can review and update these settings in the Audit Event Filter for each audited service.

Figure 2: Add Roles for Navigator Metadata Server and Navigator Audit Server to CMS



### **Enable Navigator in Hue**

To use Navigator in Hue, you must give the Hue server access to the Navigator administrator account.

For example, to use Hue and Navigator independently of each other, LDAP users must set external LDAP credentials for both Hue and Navigator. To use Navigator in Hue, each Hue server must have access to the Navigator URL with Navigator administrator credentials. That is what we are enabling here.

- 1. Log on to cloudera MANAGER and click Hue.
- 2. Click the Configuration tab.
- 3. Filter by scope=Service-wide and category=Cloudera Navigator.
- **4.** Check two properties:
  - Enable Navigator Metadata Server Integration
  - Enable Audit Collection
- 5. Select one authorization method for Navigator Metadata Server Auth:
  - Cloudera Manager (generates and encrypts credentials, and recycles them on reboot)
  - LDAP (and set Navigator Metadata Server LDAP User and LDAP Password)
  - SAML (and set Navigator Metadata Server SAML User and SAML Password)



**Important:** This method must match that used by Navigator itself. If Navigator uses LDAP to authorize users, you must select LDAP here as well, because each Hue server *is* a Navigator user.

- 6. Click Save Changes and Restart Hue.
- 7. Log on to Hue by selecting Hue Web UI.
- **8.** Log on to Cloudera Navigator from the **Clusters** drop-down.

# Tag Metadata with Navigator in Hue

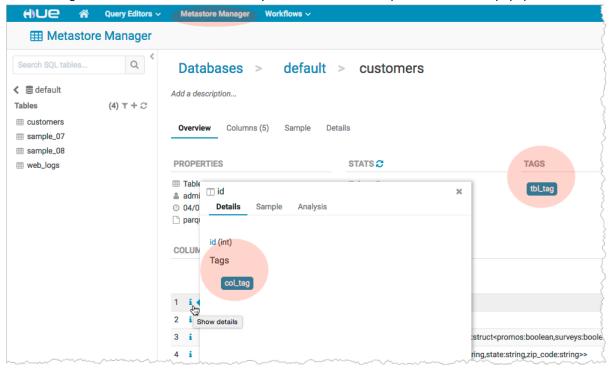
### **Prepare Hue Tables**

Install sample tables and refresh Impala metadata.

- 1. Log on to Hue by selecting **Hue Web UI** in Cloudera Manager.
- 2. Go to About Hue > Step 2: Examples and install sample tables for Hive and Impala.
- 3. Go to Query Editors > Impala and click the refresh icon €.
- **4.** Select **Perform incremental metadata update** to display sample tables.
- 5. Go to Metastore Tables Manager and click the refresh icon 2.

### Tag Database, Table, and Field

- **1.** In **Metastore Tables Manager**, click the default (or some other) database.
- 2. Add database tag: Hover over TAGS, click the edit icon 3, enter a tag of your choice, and save.
- 3. Add table tag: Click a table name (such as "customers"), hover over TAGS, and repeat.
- 4. Add field tag: Click the "show details" icon i by a column name and repeat in the context popup.



# Search Metadata with Navigator in Hue

The Navigator search field can be accessed in the Metastore Tables Manager and both Hive and Impala editors.

- 1. Go to Query Editors > Impala.
- 2. Search on "type:field tags:" in the Navigator search field.





Note: You can search for table tags with "tags:". For other types, input "type:database tags: "or "type: field tags: ".

3. Create a view of customers named David:

CREATE VIEW IF NOT EXISTS davids AS SELECT \* FROM customers WHERE customers.name LIKE 'David%';

4. Search on "davids". You should see, No recent match found, until Navigator can process the new view.



**Warning:** New tables and views can take ~1 hour to register in Navigator and be searchable.

- **5.** Log on to **cloudera** NAVIGATOR at http://<cloudera manager hostname>:7187.
- **6.** On the Search tab, select type=**View**. When you see "davids," return to Hue and retry your search.
- 7. To see your tags in Cloudera Navigator, click Add New Value under tags.

# How to Enable S3 Cloud Storage in Hue

Cloudera S3 Connector bin Cloudera Manager securely connects your CDH cluster to Amazon S3.



### Note:

- C5.11 adds S3 Guard for list consistency and support for IAM roles in Cloudera Manager.
- C5.10 connects Hue, Impala, and Navigator securely with the Cloudera S3 Connector Service.
- C5.9 adds support for Amazon S3 with plain-text credentials using Cloudera Manager safety valves.

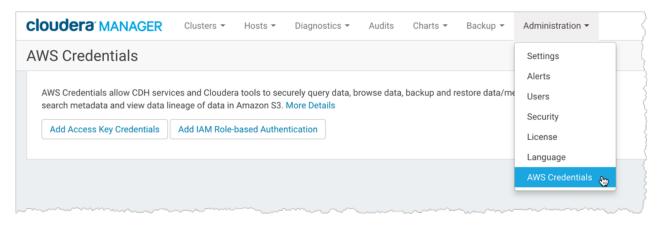
### Enable S3 in Hue with the S3 Connector Service

For a secure and fine-grained connection to Amazon S3 (for Hue, Impala, and Navigator), Cloudera recommends its S3 Connector service in Secure Mode with encrypted access keys and Kerberos and Sentry installed.



Important: Hive is not yet supported in Secure Mode. To connect Hive to S3, use "Unsecure" Mode.

Method	Security	Required	Services
Secure Mode	High	Kerberos, Sentry	Hue, Impala, Navigator
Unsecure Mode	Medium		Hue, Impala, Navigator, <b>Hive</b>



- 1. Log on to Cloudera Manager.
- 2. Select Administration > AWS Credentials.
- 3. Click Add Access Key Credentials or Add IAM Role-based Authentication.



Important: IAM Role-based Authentication is not fine-grained authentication. Also, to use it with Hue, configure the region in hue\_safety\_valve.ini—see step <a href="step-11">step 11</a>.

- **4.** Add any **Name** and enter your S3 credentials:
  - a. To connect your AWS root user, add the Access Key ID and Secret Access Key for your root account.
  - b. To connect an IAM user, add the Access Key ID and Secret Access Key for a read-only IAM account.
- 5. If you have an Amazon DynamoDB database, check Enable S3Guard for consistent read operations.

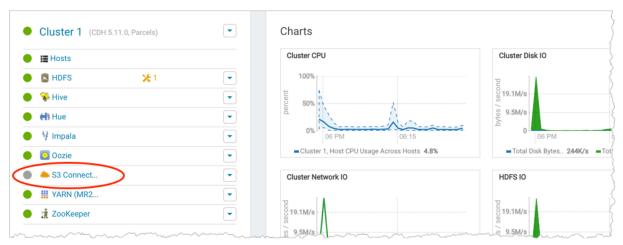


Warning: Components writing data to S3 are constrained by the inherent Amazon S3 limitation known as "eventual consistency." This can lead to data loss when a Spark or Hive job writes output directly to S3. Cloudera recommends that you use S3 Guard or write to HDFS and distop to S3.

- 6. Click Enable for <cluster name> to give Hue access to S3 and S3-backed tables. Impala must have permissions defined in Sentry.
- 7. If using access keys, select Secure or Unsecure mode. Select Unsecure to use Hive.
- Click **Continue** (at Step ) if your cluster passes validation. You are automatically taken to step 5.
- Click **Continue** (at Step <sup>5</sup>) to restart Hive, Impala, Oozie, and Hue.
- 10 When finished, click Home to see the S3 Connector.



**Note:** A gray status icon means the S3 Connector service was successfully added.



**11** If using IAM roles, set the region to us-east-1 (N. Virginia) in hue\_safety\_valve.ini. If not, ignore this step.



Note: Configuring hue\_safety\_valve.ini is a temporary Hue workaround for CDH 5.10.

- a. Select Configuration > Advanced Configuration Snippets.
- **b.** Filter by **Scope** > **Hue**.
- c. Set Hue Service Advanced Configuration Snippet (Safety Valve) for hue\_safety\_valve.ini with the following:

```
[[aws_accounts]]
[[[default]]]
region=us-east-1
```

- d. Click Save Changes.
- e. Restart Hue: select Cluster > Hue and Actions > Restart.



**Note:** The S3 Connector service is not added when you use IAM roles.

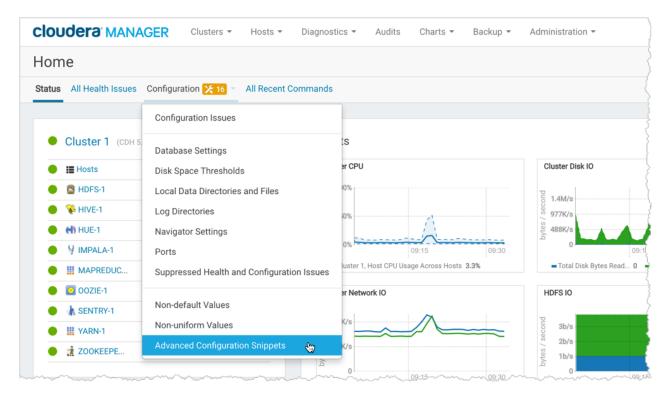
Related topics: How to Configure AWS Credentials and Configuring the Amazon S3 Connector.

### Enable S3 in Hue with Safety Valves

This section assumes an AWS account with access keys, but not necessarily a Kerberized cluster.

You can connect to S3 using three safety valves (also known as Advanced Configuration Snippets):

- Hue Service Advanced Configuration Snippet (Safety Valve) for hue\_safety\_valve.ini
- Cluster-wide Advanced Configuration Snippet (Safety Valve) for core-site.xml
- Hive Service Advanced Configuration Snippet (Safety Valve) for core-site.xml.



- **1.** Log on to Cloudera Manager and select **Clusters** > **your cluster**.
- 2. Select Configuration > Advanced Configuration Snippets.
- **3.** Filter by **Scope** > **Hue**.
- 4. Set your S3 credentials in Hue Service Advanced Configuration Snippet (Safety Valve) for hue\_safety\_valve.ini:



Note: Store your credentials in a script that outputs to stdout. A security\_token is optional.

```
[aws]
[[aws_accounts]]
[[[default]]]
access_key_id_script=</path/to/access_key_script>
secret_access_key_script=</path/to/secret_key_script>
#security_token=<your AWS security token>
allow_environment_credentials=false
region=<your region, such as us-east-1>
```

For a proof-of-concept installation, you can add the IDs directly.

```
access_key_id=<your_access_key_id>
secret_access_key=<your_secret_access_key>
```

- 5. Clear the scope filters and search on "core-site.xml".
- 6. To enable the S3 Browser, set your <u>S3 credentials</u> in Cluster-wide Advanced Configuration Snippet (Safety Valve) for core-site.xml:

```
<property>
<name>fs.s3a.access.key</name>
<value>AWS access key ID</value>
</property>

cproperty>
<name>fs.s3a.secret.key</name>
<value>AWS secret key</value>
</property>
</property>
</property>
```

- 7. To enable Hive with S3, set your S3 credentials in Hive Service Advanced Configuration Snippet (Safety Valve) for core-site.xml.
- 8. Click Save Changes.
- 9. Restart Hue: select Cluster > Hue and Actions > Restart.
- 10 Restart Hive: select Cluster > Hive and Actions > Restart.

Related topics: Amazon Web Services (AWS) Security.

# Generate Access Keys in AWS

To integrate Hue with S3, you must have an Amazon Web Services (AWS) account 략, with access keys for *either* your root user or a read-only IAM user.

#### **Root Account**

- 1. Create an AWS account and sign in to the AWS Console.
- 2. Create access keys for this AWS <u>root account</u>:
  - a. Expand the drop-down menu under your account name and select My Security Credentials.
  - b. Click Continue to Security Credentials.
  - c. Expand Access Keys (Access Key ID and Secret Access Key).
  - d. Click Create New Access Key.
  - e. Click Show Access Key or Download Key File. These are your AWS root credentials.

#### **IAM Account**

1. Create two IAM groups (AWS admin and S3 Read-only):



Important: AWS requires that your first IAM group and associated user has administrator access.

- a. Go to the IAM service.
- b. Click Groups and Create New Group.
- c. Enter a name and click Next Step.
- d. Filter on "admin" and select the AdministratorAccess policy.
- e. Click Next Step and Create Group.
- f. Create a second group with AmazonS3ReadOnlyAccess.
- 2. Create two IAM users and assign one to the admin policy and one to the S3 read policy.
  - a. Click Users and Add User.
  - **b.** Enter a name, and at a minimum, select **Programmatic access**.
  - c. Click Next: Permissions.
  - **d.** Select the group with administrator permissions.
  - e. Click Next: Review and Create User.
  - f. Create a second user and assign the group with S3 read-only access.
- **3.** Create access keys for your *read-only* IAM user:
  - a. Click the name of your read-only IAM user.
  - b. Click the Security Credentials tab.
  - c. Click Create Access Key.
  - d. Click Show Access Key or Download Key File. These are your IAM user credentials.

# How to Use S3 as Source or Sink in Hue

On this page, we demonstrate how to write to, and read from, an S3 bucket in Hue.

# Populate S3 Bucket

In this section, we use open data from the U.S. Geological Survey.

- 1. Download 30 days of earthquake data (all\_month.csv) from the USGS (~2 MB).
- 2. Log on to the Hue Web UI from Cloudera Manager.
- 3. Select File Browser > S3 Browser &.
- 4. Click New > Bucket, name it "quakes\_<any unique id>" and click Create.
  Tip: Unique bucket names are important per S3 bucket naming conventions.
- **5.** Navigate into the bucket by clicking the bucket name.
- **6.** Click **New > Directory**, name it "input" and click **Create**.
- 7. Navigate into the directory by clicking the directory name.
- 8. Click Upload and select, or drag, all\_month.csv. The path is s3a://quakes/input/all\_month.csv.



**Important:** Do not add anything else to the "input" directory—no extra files, no directories.

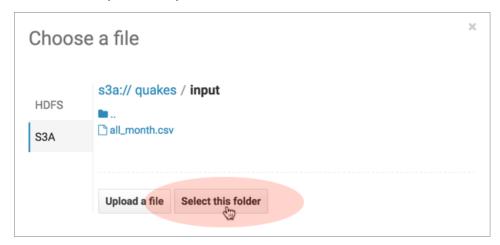
### Create Table with S3 File

- 1. Go to the Metastore Manager by clicking Data Browsers > Metastore Tables.
- 2. Create a new table from a file by clicking the

a

icon.

- 3. Enter a Table Name such as "earthquakes".
- **4.** Browse for the **Input Directory**, s3a://quakes/input/, and click **Select this folder**.



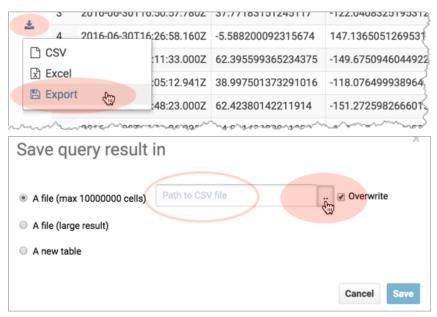
- 5. Select Create External Table from the Load Data menu and click Next.
- 6. Delimit by Comma(,) and click Next.
- 7. Click Create Table.
- **8.** Click **Browse Data 1** to automatically generate a SELECT query in the **Hive** editor:

```
SELECT * FROM `default`.`earthquakes` LIMIT 10000;
```

# **Export Query Results to S3**

### 1. Run and Export Results in Hive

- **a.** Run the query by clicking the **Execute** button.
- **b.** Click the **Get Results b** button.
- c. Select Export to open the Save query result dialog.



### 2. Save Results as Custom File

- a. Select In store (max 10000000 cells) and open the Path to CSV file dialog.
- **b.** Navigate into the bucket, **s3a://quakes**.
- c. Create folder named, "output."
- d. Navigate into the output directory and click Select this folder.
- e. Append a file name to the path, such as quakes.cvs.
- **f.** Click **Save**. The results are saved as s3a://quakes/ouput/quakes.csv.



### 3. Save Results as MapReduce files

- a. Select In store (large result) and open the Path to empty directory dialog.
- **b.** Navigate into the bucket, **s3a://quakes**.
- c. If you have not done so, create a folder named, "output."
- **d.** Navigate into the **output** directory and click **Select this folder**.
- e. Click Save. A MapReduce job is run and results are stored in s3a://quakes/output/.



### 4. Save Results as Table

**a.** Run a query for "moment" earthquakes and export:

```
SELECT time,
         latitude,
         longitude,
```

```
FROM `default`.`earthquakes`
WHERE magtype IN ('mw','mwb','mwc','mwr','mww');
```

- b. Select A new table and input <database>. <new table name>.
- c. Click Save.
- **d.** Click **Browse Data to** view the new table.
  - A new table default.quakes\_moment

### Troubleshoot Frrors

This section addresses some error messages you may encounter when attempting to use Hue with S3.

Tip: Restart the Hue service to view buckets, directories, and files added to your upstream S3 account.

### Failed to access path

Failed to access path: "s3a://quakes". Check that you have access to read this bucket and that the region is correct.

Possible solution: Check your bucket region:

- 1. Log on to your AWS account and navigate to the S3 service.
- 2. Select your bucket, for example "quakes", and click Properties.
- **3.** Find your region. If it says <u>US Standard</u>, then region=us-east-1.
- 4. Update your configuration in Hue Service Advanced Configuration Snippet (Safety Valve) for hue\_safety\_valve.ini.
- 5. Save your changes and restart Hue.

### The table could not be created

The table could not be created. Error while compiling statement: FAILED: SemanticException com.cloudera.com.amazonaws.AmazonClientException: Unable to load AWS credentials from any provider in the chain.

Possible solution: Set your S3 credentials in Hive core-site.xml:

- 1. In Cloudera Manager, go to Hive > Configuration.
- 2. Filter by Category > Advanced.
- 3. Set your credentials in Hive Service Advanced Configuration Snippet (Safety Valve) for core-site.xml.
  - a. Click the + button and input Name and Value for fs.s3a.AccessKeyId.
  - **b.** Click the **button** and input Name and Value for fs.s3a.SecretAccessKey.
- 4. Save your changes and restart Hive.

### · The target path is a directory

Possible solution: Remove any directories or files that may have been added to s3a://quakes/input/ (so that all\_month.csv is alone).

### • Bad status for request TFetchResultsReq ... Not a file

```
Bad status for request TFetchResultsReq(...):
TFetchResultsResp(status=TStatus(errorCode=0, errorMessage='java.io.IOException:
java.io.IOException: Not a file: s3a://not a file: s3a://quakes/input/output' ...
```

Possible solution: Remove any directories or files that may have been added to s3a://quakes/input/ (so that all\_month.csv is alone). Here, Hive cannot successfully query the earthquakes table (based on all\_month.csv) due to the directory, s3a://quakes/input/output.

**Tip:** Run tail -f against the Hive server log in: /var/log/hive/.

### How to Run Hue Shell Commands

You may need to administer Hue programmatically, for example, to reset the superuser password. This page addresses managed deployments of CDH 5.5 and higher.

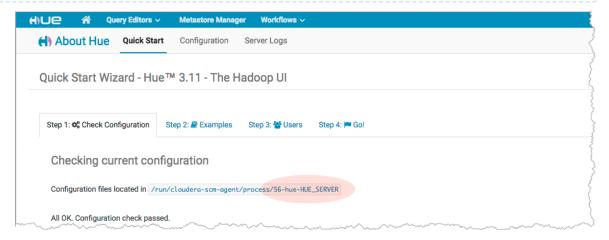
### 1. Gather the following information:

- Hue server database password (embedded or external).
- Path to /build/env/bin/hue:

```
# Parcels (e.g., /opt/cloudera/parcels/CDH-5.9.0-1.cdh5.9.0.p0.23/lib/hue)
realpath /opt/cloudera/parcels/`ls -l /opt/cloudera/parcels | grep CDH | tail -l | awk '{print $9}'`/lib/hue
# Packages
/usr/lib/hue
```

Path to the current Hue process directory (with Hue configuration files):

#Example path: /var/run/cloudera-scm-agent/process/56-hue-HUE\_SERVER/ realpath /var/run/cloudera-scm-agent/process/`ls -alrt /var/run/cloudera-scm-agent/process grep HUE | tail -1 | awk '{print \$9}'



**2. Set HUE\_CONF\_DIR** to the latest Hue process directory:

```
export HUE_CONF_DIR="/var/run/cloudera-scm-agent/process/`ls -alrt
/var/run/cloudera-scm-agent/process | grep HUE | tail -1 | awk '{print $9}'`"
echo $HUE_CONF_DIR
```

### 3. Run shell subcommands

When true, HUE\_IGNORE\_PASSWORD\_SCRIPT\_ERRORS runs the Hue shell even if hue.ini contains passwords generated by Cloudera Manager (such as bind\_password and ssl\_password).



Note: Do not export HUE\_IGNORE\_PASSWORD\_SCRIPT\_ERRORS or HUE\_DATABASE\_PASSWORD to ensure that they are not stored and only apply to this command.

### Parcels

· List available subcommands

HUE\_IGNORE\_PASSWORD\_SCRIPT\_ERRORS=1 HUE\_DATABASE\_PASSWORD=<db password> opt/cloudera/parcels/`ls -l /opt/cloudera/parcels | grep CDH | tail -1 | awk '{print/ \$9}'`/lib/hue/build/env/bin/hue

• Run the interactive Hue Python shell (Ctrl+D to quit)

HUE\_IGNORE\_PASSWORD\_SCRIPT\_ERRORS=1 HUE\_DATABASE\_PASSWORD=<db\_password>
/opt/cloudera/parcels/`ls -1 /opt/cloudera/parcels | grep CDH | tail -1 | awk '{print \$9}'`/lib/hue/build/env/bin/hue shell

· Change a user password

HUE\_IGNORE\_PASSWORD\_SCRIPT\_ERRORS=1 HUE\_DATABASE\_PASSWORD=<db\_password>
/opt/cloudera/parcels/`ls -1 /opt/cloudera/parcels | grep CDH | tail -1 | awk '{print \$9}'`/lib/hue/build/env/bin/hue changepassword <username>

### **Packages**

· List available subcommands

HUE\_IGNORE\_PASSWORD\_SCRIPT\_ERRORS=1 HUE\_DATABASE\_PASSWORD=<db\_password>
/usr/lib/hue/build/env/bin/hue

• Run the interactive Hue Python shell (Ctrl+D to quit)

HUE\_IGNORE\_PASSWORD\_SCRIPT\_ERRORS=1 HUE\_DATABASE\_PASSWORD=<db\_password>
/usr/lib/hue/build/env/bin/hue shell

· Change a user password

HUE\_IGNORE\_PASSWORD\_SCRIPT\_ERRORS=1 HUE\_DATABASE\_PASSWORD=<db\_password>
/usr/lib/hue/build/env/bin/hue changepassword <username>

### For unmanaged and lower CDH versions, see:

- Execute some builtin or shell commands
- Storing passwords in file script
- How to change or reset a forgotten password?

# **Hue Troubleshooting**

This section addresses possible obstacles when installing, configuring, and using Hue. Watch this space for more topics!

# Potential Misconfiguration Detected

This page covers various configuration errors. The goal is for all configuration checks to pass.

Checking current configuration

Configuration files located in /var/run/cloudera-scm-agent/process/108-hue-HUE\_SERVER

All OK. Configuration check passed.

### **Preferred Storage Engine**

PREFERRED\_STORAGE\_ENGINE: We recommend MySQL InnoDB engine over MyISAM which does not support transactions.

Checking current configuration

Configuration files located in /var/run/cloudera-scm-agent/process/233-hue-HUE\_SERVER

Potential misconfiguration detected. Fix and restart Hue.

PREFERRED\_STORAGE\_ENGINE

We recommend MySQL InnoDB engine over MyISAM which does not support transactions.



Warning: Talk to your DBA before changing the storage engine for the Hue database tables.

# Alter Hue database tables from MyISAM to InnoDB

- 1. Stop the Hue service in Cloudera Manager: go to Cluster > Hue and select Actions > Stop.
- 2. Log on to the host of your MySQL server.
- **3.** Look for any MyISAM tables in your Hue server database:

```
mysql -u root -p<root password>
SELECT table_schema, table_name, engine
FROM information_schema.tables
WHERE engine = 'MyISAM' AND table_schema = '<hue database name>';
quit
```

# **Hue Troubleshooting**

**4.** Set the engine to InnoDB for all Hue database tables:

```
# Create script, /tmp/set_engine_innodb.ddl
mysql -u root -p<root password> -e \
"SELECT CONCAT('ALTER TABLE ',table_schema,'.',table_name,' engine=InnoDB;') \
FROM information_schema.tables \
WHERE engine = 'MyISAM' AND table_schema = '<hue database name>';" \
| grep "ALTER TABLE <hue database name>" > /tmp/set_engine_innodb.ddl

# Run script
mysql -u root -p<root password> < /tmp/set_engine_innodb.ddl</pre>
```

- 5. Verify that no MyISAM tables exist by rerunning the SELECT statement in step 3 on page 73.
- 6. Start the Hue service.

### MySQL Storage Engine

MYSQL\_STORAGE\_ENGINE: All tables in the database must be of the same storage engine type (preferably InnoDB).

Follow the instructions in the section, Preferred Storage Engine on page 73, to ensure all Hue tables use InnoDB.