Cloudera Runtime 7.1.9

Administering Hue

Date published: 2020-08-10
Date modified: 2023-09-07

https://docs.cloudera.com/

Legal Notice

© Cloudera Inc. 2024. All rights reserved.

The documentation is and contains Cloudera proprietary information protected by copyright and other intellectual property rights. No license under copyright or any other intellectual property right is granted herein.

Unless otherwise noted, scripts and sample code are licensed under the Apache License, Version 2.0.

Copyright information for Cloudera software may be found within the documentation accompanying each component in a particular release.

Cloudera software includes software from various open source or other third party projects, and may be released under the Apache Software License 2.0 (“ASLv2”), the Affero General Public License version 3 (AGPLv3), or other license terms. Other software included may be released under the terms of alternative open source licenses. Please review the license and notice files accompanying the software for additional licensing information.

Please visit the Cloudera software product page for more information on Cloudera software. For more information on Cloudera support services, please visit either the Support or Sales page. Feel free to contact us directly to discuss your specific needs.

Cloudera reserves the right to change any products at any time, and without notice. Cloudera assumes no responsibility nor liability arising from the use of products, except as expressly agreed to in writing by Cloudera.

Cloudera, Cloudera Altus, HUE, Impala, Cloudera Impala, and other Cloudera marks are registered or unregistered trademarks in the United States and other countries. All other trademarks are the property of their respective owners.

Disclaimer: EXCEPT AS EXPRESSLY PROVIDED IN A WRITTEN AGREEMENT WITH CLOUDERA, CLOUDERA DOES NOT MAKE NOR GIVE ANY REPRESENTATION, WARRANTY, NOR COVENANT OF ANY KIND, WHETHER EXPRESS OR IMPLIED, IN CONNECTION WITH CLOUDERA TECHNOLOGY OR RELATED SUPPORT PROVIDED IN CONNECTION THEREWITH. CLOUDERA DOES NOT WARRANT THAT CLOUDERA PRODUCTS NOR SOFTWARE WILL OPERATE UNINTERRUPTED NOR THAT IT WILL BE FREE FROM DEFECTS NOR ERRORS, THAT IT WILL PROTECT YOUR DATA FROM LOSS, CORRUPTION NOR UNAVAILABILITY, NOR THAT IT WILL MEET ALL OF CUSTOMER’S BUSINESS REQUIREMENTS. WITHOUT LIMITING THE FOREGOING, AND TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, CLOUDERA EXPRESSLY DISCLAIMS ANY AND ALL IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, QUALITY, NON-INFRINGEMENT, TITLE, AND FITNESS FOR A PARTICULAR PURPOSE AND ANY REPRESENTATION, WARRANTY, OR COVENANT BASED ON COURSE OF DEALING OR USAGE IN TRADE.

Contents

Reference architecture

Hue server can support approximately 25 concurrent users, depending on what tasks the users are performing. Most scaling issues occur as a result of users performing resource-intensive operations and not from the number of users. For example, large downloads of query results can impact resource availability for the other users who are using the same Hue instance during the download operation. During that time, the users can experience slow performance. Another common cause of noticeable performance changes are slow RPC calls between Hue and another service. When this happens, queries may appear to suddenly "stop responding" after they are submitted.

As a guide, 2 Hue servers can support up to:

- 100 unique users per week
- 50 users per hour at peak times executing up to 100 queries

A typical setup is 2 Hue servers.

General guidelines

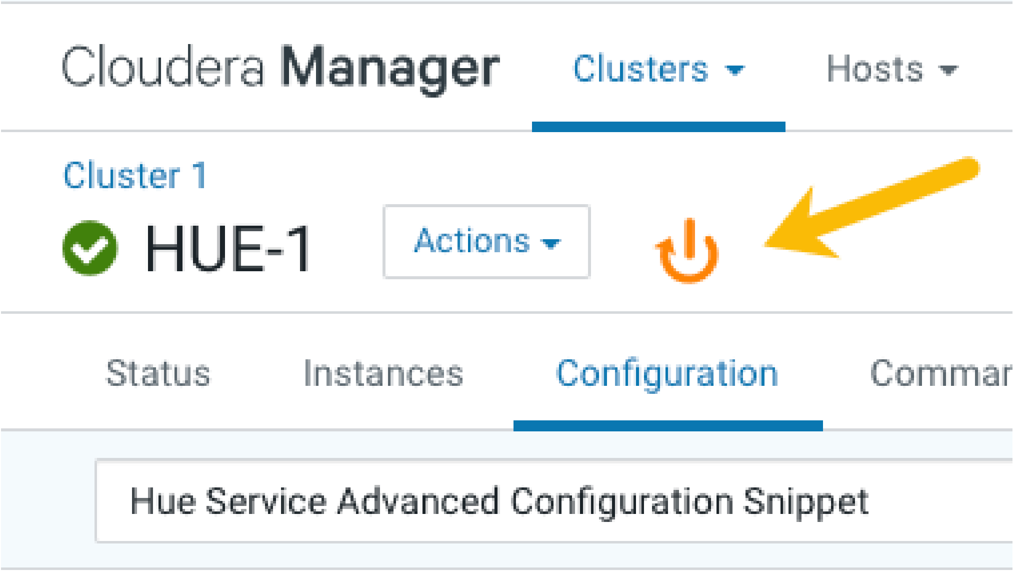
- Deploy a load balancer in front of Hue.
- Use a production-quality database.
- Ensure that other services, such as Impala, Hive, and Oozie, are healthy and not impacted by too few resources. If these services are hanging, it adversely affects Hue performance.
- Consider moving workloads that are subject to SLAs (service-level agreements) or considered "noisy neighbors" to their own compute cluster. Noisy neighbors are workloads that use the majority of available resources and cause performance issues.
- Limit the number of rows that are returned for queries.

  One way to limit the number of rows returned is to specify a value for the download_row_limit configuration property for the Hue Beeswax application. This property can be set in the Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini property in Cloudera Manager:

  1. In Cloudera Manager, click HueConfiguration , and enter Hue Service Advanced Configuration Snippet in the search text box.
  2. In the text box for the Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini, add the following configuration information:

     ```
     [beeswax]
     download_row_limit=number_of_rows
     ```

3. Click Save Changes and click the restart icon at the top of the page to restart the Hue service:



Hue configuration files

Hue roles are configured with the following three configuration files: hue.ini, hue_safety_valve.ini, and hue_safety_valve_server.ini.

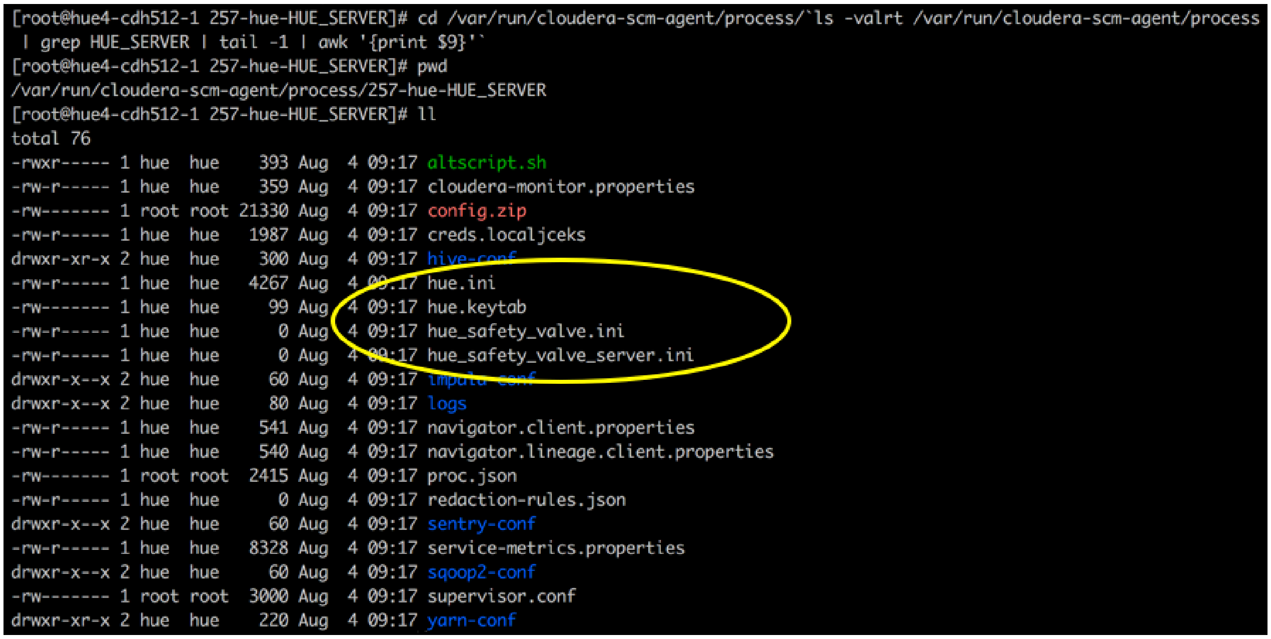
The hue.ini file is the first file that is auto-generated when you add the Hue service to your CDP cluster using Cloudera Manager. The hue_safety_valve.ini file is used to override bugs or configure properties that are missing from the Cloudera Manager UI. The hue_safety_valve_server.ini file is used to configure only the Hue role, and not the full Hue service (which includes the Hue Load Balancer). The hue_safety_valve_server.ini file is not used in practice.

Cloudera recommends that you do not edit the .ini files from the command line because they are stored in dynamic directories named by process ID and populated from the Cloudera Manager database. To add configurations that you cannot add directly from the Cloudera Manager UI, such as Authentication Backend for SAML, use the Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini field under the Hue service configurations in Cloudera Manager.

Run the following command to view the .ini files per process ID:

```
ls -ltr /var/run/cloudera-scm-agent/process/`ls -valrt /var/run/cloudera-scm
-agent/process | grep HUE_SERVER | tail -1 | awk '{print $9}'`
```

Figure 1: Terminal showing Hue configuration files



```
[root@hue4-cdh512-1 257-hue-HUE_SERVER]# cd /var/run/cloudera-scm-agent/process/`ls -valrt /var/run/cloudera-scm-agent/process
 | grep HUE_SERVER | tail -1 | awk '{print $9}'`
[root@hue4-cdh512-1 257-hue-HUE_SERVER]# pwd
/var/run/cloudera-scm-agent/process/257-hue-HUE_SERVER
[root@hue4-cdh512-1 257-hue-HUE_SERVER]# ll
total 76
-rwxr----- 1 hue  hue    393 Aug  4 09:17 altscript.sh
-rw-r----- 1 hue  hue    359 Aug  4 09:17 cloudera-monitor.properties
-rw------- 1 root root 21330 Aug  4 09:17 config.zip
-rw-r----- 1 hue  hue   1987 Aug  4 09:17 creds.localjceks
drwxr-xr-x 2 hue  hue    300 Aug  4 09:17 hive-conf
-rw-r----- 1 hue  hue   4267 Aug  4 09:17 hue.ini
-rw------- 1 hue  hue     99 Aug  4 09:17 hue.keytab
-rw-r----- 1 hue  hue      0 Aug  4 09:17 hue_safety_valve.ini
-rw-r----- 1 hue  hue      0 Aug  4 09:17 hue_safety_valve_server.ini
drwxr-x--x 2 hue  hue     60 Aug  4 09:17 impala-conf
drwxr-x--x 2 hue  hue     80 Aug  4 09:17 logs
-rw-r----- 1 hue  hue    541 Aug  4 09:17 navigator.client.properties
-rw-r----- 1 hue  hue    540 Aug  4 09:17 navigator.lineage.client.properties
-rw------- 1 root root  2415 Aug  4 09:17 proc.json
-rw-r----- 1 hue  hue      0 Aug  4 09:17 redaction-rules.json
drwxr-x--x 2 hue  hue     60 Aug  4 09:17 sentry-conf
-rw-r----- 1 hue  hue   8328 Aug  4 09:17 service-metrics.properties
drwxr-x--x 2 hue  hue     60 Aug  4 09:17 sqoop2-conf
-rw------- 1 root root  3000 Aug  4 09:17 supervisor.conf
drwxr-xr-x 2 hue  hue    220 Aug  4 09:17 yarn-conf
```

The process directory for any given role is mirrored in Cloudera Manager. Go to Hue Instances , select a role such as Hue Server, and then click the Processes tab.

Related Information

hue.ini

Hue configurations in CDP Runtime

Cloudera Manager exposes properties that allow you to insert custom configuration text into XML configuration, property, and text files, or into an environment. The configuration snippet mechanism is intended for use in cases where there is a configuration setting that is not exposed as a configuration property in the Cloudera Manager web interface. Configuration snippets generally override normal configuration. You can configure custom properties for Hue by specifying them in the Advanced Configuration Snippet (Safety valve) field in Cloudera Manager.

The following table lists the safety valve parameters supported by Hue in CDP Runtime:

Parameter	Description

`[[desktop]]`
`app_blacklist`
	Used to add or remove applications, such as the File Browser, Impala, Hive, Oozie, and so on from the blocked list of applications.

`[desktop]`
`   [[session]]`
`    ttl=[***NUMBER-OF-SECONDS***]`
	Used to configure the duration of a user session. The ttl property determines the length of time that the cookie with the user's session ID lives before expiring. After the ttl setting is reached, the user's session expires whether it is active or not.
	The default setting for ttl is 1,209,600 seconds, which equals two weeks.

`[jobbrowser]`
`[[query_store]]`
`server_url=[***QUERY-PROCESSOR-URL***]`
	Used to display the **Queries** tab for Hive and Impala on the **Job Browser** page. This configuration is enabled by default and is not exposed in the Hue safety valve.
	However, to hide the **Queries** tab, you can override the configuration by adding the following lines in Cloudera Manager Clusters Hue Configuration Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini :
	`[jobbrowser]`
	`[[query_store]]`
	`is_enabled=false`
	To enable the **Queries** tab, set is_enabled to true.

`[aws]`
`[[aws_accounts]]`
`[[[default]]]`
`access_key_id=[***AWS-ACCESS-KEY***]`
`secret_access_key=[***SECRET-ACCESS-KEY***]`
`region=[***AWS-REGION***]`
	Used to enable the S3 File Browser for Hue without IDBroker.

`[aws]`
`has_iam_detection=true`
`[[aws_accounts]]`
`[[[default]]]`
`region=[***AWS-REGION***]`
	Used to enable the S3 File Browser for Hue with IDBroker.

`[desktop]`
`[azure]`
`  [[azure_accounts]]`
`    [[[default]]]`
`      client_id=[***AZURE-ACCOUNT-CLIENT-ID***]`
`      client_secret=[***AZURE-ACCOUNT-CLIENT-SECRET***]`
`      tenant_id=[***AZURE-ACCOUNT-TENANT-ID***]`
 	
`    [[abfs_clusters]]`
`      [[[default]]]`
`  fs_defaultfs=abfs://[***CONTAINER-NAME***]@[***AZURE-STORAGE-ACCOUNT-NAME***]>.dfs.core.windows.net`
`              webhdfs_url=https://[***AZURE-STORAGE-ACCOUNT-NAME***].dfs.core.windows.net/`
	Used to enable the ABFS File Browser for Hue without IDBroker.

`[desktop]`
`[azure]`
	Used to enable the ABFS File Browser for Hue with IDBroker.

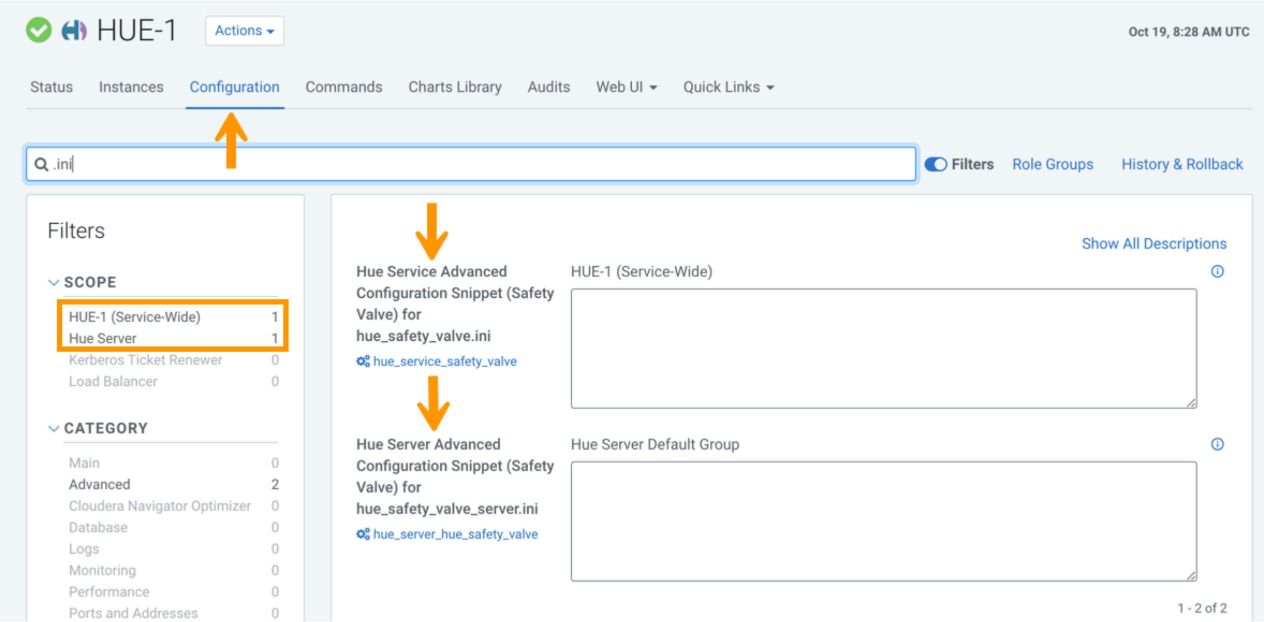
Contact Cloudera Support if you are required to use a configuration snippet that is not explicitly documented.

Introduction to Hue Advanced Configuration Snippet (Safety valves)

To customize and configure Hue properties, such as authentication or enabling S3 or ABFS browsers, and so on that you cannot directly configure from the Cloudera Manager UI, you can use the Advanced Configuration Snippet field under the Hue cluster configurations in Cloudera Manager.

Advanced safety valves

The field for hue_safety_valve.ini is service-wide and affects all Hue roles listed on the Instances tab. The field for hue_safety_valve_server.ini only affects Hue Server role instances.



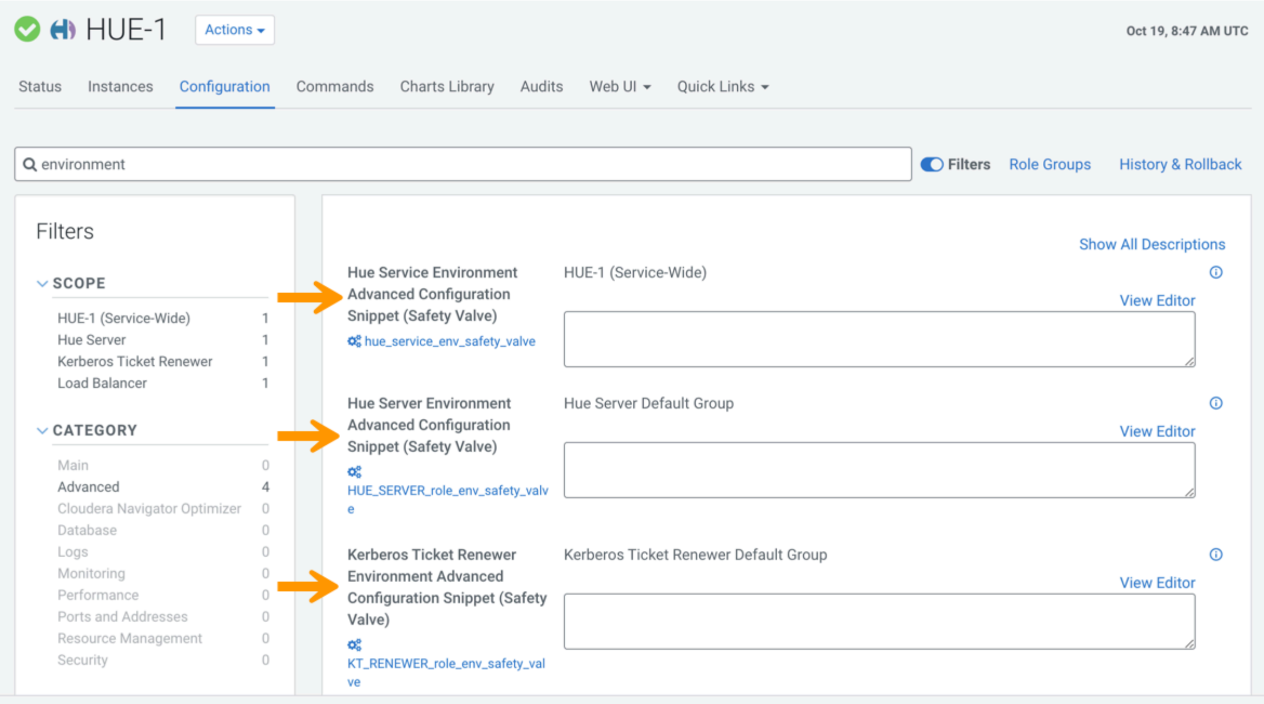
.ini

Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini
hue_service_safety_valve
HUE-1 (Service-Wide)

Hue Server Advanced Configuration Snippet (Safety Valve) for hue_safety_valve_server.ini
hue_server_hue_safety_valve
Hue Server Default Group

Environment safety valves

Environment safety-valves let you configure environment variables across the service or for specific role instances. These safety-valves lack headers.



environment

Filters

SCOPE

HUE-1 (Service-Wide)	1
Hue Server	1
Kerberos Ticket Renewer	1
Load Balancer	1

CATEGORY

Main	0
Advanced	4
Cloudera Navigator Optimizer	0
Database	0
Logs	0
Monitoring	0
Performance	0
Ports and Addresses	0
Resource Management	0
Security	0

Hue Service Environment Advanced Configuration Snippet (Safety Valve)
hue_service_env_safety_valve
HUE-1 (Service-Wide)

Hue Server Environment Advanced Configuration Snippet (Safety Valve)
HUE_SERVER_role_env_safety_valve
Hue Server Default Group

Kerberos Ticket Renewer Environment Advanced Configuration Snippet (Safety Valve)
KT_RENEWER_role_env_safety_valve
Kerberos Ticket Renewer Default Group

Hue logs

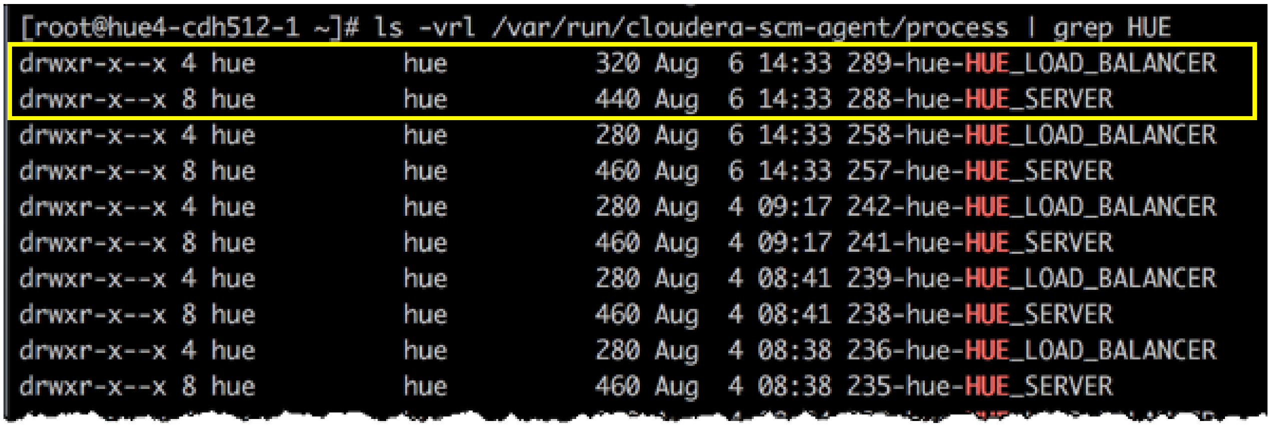
Cloudera Manager generates standard stream logs when each Hue role instance starts and stops. The Hue service, which is built on the Django framework, generates log4j logs when the service is running.

Standard stream logs

Cloudera Manager logs the start and stop of each supervised Hue process in standard stream logs (stdout.log, stderr.log).

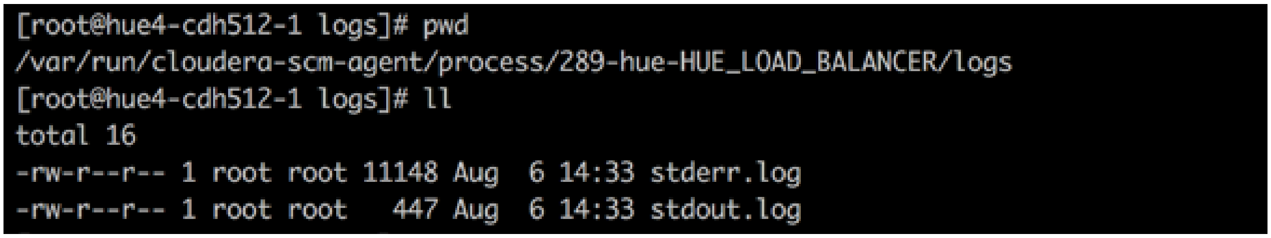
When the Hue service restarts, Cloudera Manager generates a new directory for each supervised process of a Hue role instance:

```
ls -vrl /var/run/cloudera-scm-agent/process | grep HUE
```



```
[root@hue4-cdh512-1 ~]# ls -vrl /var/run/cloudera-scm-agent/process | grep HUE
drwxr-x--x 4 hue          hue          320 Aug  6 14:33 289-hue-HUE_LOAD_BALANCER
drwxr-x--x 8 hue          hue          440 Aug  6 14:33 288-hue-HUE_SERVER
drwxr-x--x 4 hue          hue          280 Aug  6 14:33 258-hue-HUE_LOAD_BALANCER
drwxr-x--x 8 hue          hue          460 Aug  6 14:33 257-hue-HUE_SERVER
drwxr-x--x 4 hue          hue          280 Aug  4 09:17 242-hue-HUE_LOAD_BALANCER
drwxr-x--x 8 hue          hue          460 Aug  4 09:17 241-hue-HUE_SERVER
drwxr-x--x 4 hue          hue          280 Aug  4 08:41 239-hue-HUE_LOAD_BALANCER
drwxr-x--x 8 hue          hue          460 Aug  4 08:41 238-hue-HUE_SERVER
drwxr-x--x 4 hue          hue          280 Aug  4 08:38 236-hue-HUE_LOAD_BALANCER
drwxr-x--x 8 hue          hue          460 Aug  4 08:38 235-hue-HUE_SERVER
```

It writes to a nested logs directory for each running instance:



```
[root@hue4-cdh512-1 logs]# pwd
/var/run/cloudera-scm-agent/process/289-hue-HUE_LOAD_BALANCER/logs
[root@hue4-cdh512-1 logs]# ll
total 16
-rw-r--r-- 1 root root 11148 Aug  6 14:33 stderr.log
-rw-r--r-- 1 root root   447 Aug  6 14:33 stdout.log
```

Configuration errors are written here because they prevent Hue servers and load balancers from starting properly.

Tip: Testing the LDAP configuration from Cloudera Manager (Clusters Hue service Test LDAP Configuration) also writes to standard stream logs which you can search using the following command: ls -vrl /var/run/cloudera-scm-agent/process | grep ldaptest

Hue service Django logs

When the Hue service is running, Hue generates logs in /var/log/hue using log4j. Load balancer logs are in /var/run/httpd. You can view these logs in Hue at http://hueserver:port/logs.

Table 1: Hue service logs

Log Name	Description
access.log	Filtered list of successful attempts to access Hue Web UI.
audit.log	Captures user operations, such as: • USER_LOGIN • USER_LOGOUT • EXPORT • DOWNLOAD • DELETE_USER • DELETE_GROUP • EDIT_USER • CREATE_USER • EDIT_GROUP • CREATE_GROUP • EDIT_PERMISSION • ADD_LDAP_USERS • ADD_LDAP_GROUPS • SYNC_LDAP_USERS_GROUPS
audit/hue_server_audit_wal.log	Audit log visible in Apache Atlas.
error.log	Filtered list of all nontrivial errors
kt_renewer.log	Kerberos ticket renews
metrics-hue_server/metrics.log	Populates charts in Cloudera Manager
migrate.log	Database and table migrations + First Run of Hue server
rungunicornserver.log	Hue (Gunicorn) web server information.
hue_install.log	Contains the log produced during installation

Enabling DEBUG

DEBUG is available for the Hue Django logs in /var/log/hue. By default, the Hue service writes INFO level messages and keeps a small buffer of log messages at all levels in memory.

About this task

In the debug mode, Django displays a detailed traceback when an exception occurs. Debugging information may contain sensitive data.

Note: Django logs every SQL query it executes in the debug mode. This could rapidly increase memory consumption.

You can enable DEBUG messages for all the logs in var/log/hue from Cloudera Manager or from the Hue web UI.

Using Cloudera Manager

1. Go to Cloudera Manager Clusters Hue Configuration .
2. Select the Enable Django Debug Mode option.
3. Click Save Changes.
4. Restart the Hue service.

Using the Hue web interface

1. Log into the Hue web interface as a superuser.
2. Click Administer Server from the left-assist pannel.

   About Hue page is displayed.
3. Click on the Server Logs tab.
4. Select the Force DEBUG level option.

   Debug is enabled on-the-fly.

Enabling httpd log rotation for Hue

Several log files, such as access logs and error logs, are produced by the Apache HTTP server. Over time, these files can grow in size and can cause disk issues. You can prevent this problem by using Apache's rotatelogs utility and defining when to rotate logs – either based on a time interval or maximum size of the log.

About this task

You can configure the Apache server to rotate the logs when a certain filesize is reached (for example, 5 Bytes or 10 KBytes) or at a set time interval (for example, daily or weekly), or both. If you want to specify both the time and size, the size must be specified after the time. Log rotation occurs whenever time or size limits are reached.

The log files are generated as per the configurations defined in the httpd.conf file.

Before you begin

The rotatelogs utility must be present on your Operating System, and you must know its location so that you can use it. Run the following command from the Hue load balancer host system to get the path of the rotatelogs utility:

```
[root@example-domain-hue-HUE_LOAD_BALANCER]# which rotatelogs
/usr/sbin/rotatelogs
```

Procedure

1. Log in to Cloudera Manager as an Administrator.
2. Go to Clusters Hue service Configuration .

3. Configure the log rotation policy based on your criteria in the Load Balancer Advanced Configuration Snippet (Safety Valve) for httpd.conf field.

   To rotate logs whenever the specific file size is reached, add the following lines:

   ```
   CustomLog "|/usr/sbin/rotatelogs /var/log/hue-httpd/access_log.%Y-%m-%d-
   %H_%M_%S [***FILE-SIZE***](B|K|M|G)" common
   ErrorLog "|/usr/sbin/rotatelogs /var/log/hue-httpd/error_log.%Y-%m-%d-%H_
   %M_%S [***FILE-SIZE***](B|K|M|G)"
   ```

   For example:

   ```
   CustomLog "|/usr/sbin/rotatelogs /var/log/hue-httpd/access_log.%Y-%m-%d-
   %H_%M_%S 5B" common
   ErrorLog "|/usr/sbin/rotatelogs /var/log/hue-httpd/error_log.%Y-%m-%d-%H_
   %M_%S 5B"
   ```

   This configuration rotates the log file whenever it reaches a size of 5 bytes, and the suffix to the log file name will be created with the format access_log. YYYY-mm-dd-HH_MM_SS and error_log.YYYY-mm-dd-HH_MM_SS.

   To rotate logs daily, add the following lines:

   ```
   CustomLog "|/usr/sbin/rotatelogs -l /var/log/hue-httpd/access_log.%Y-%m-
   %d-%H_%M_%S 86400" common
   ErrorLog "|/usr/sbin/rotatelogs -l /var/log/hue-httpd/error_log.%Y-%m-%d-
   %H_%M_%S  86400"
   ```

   This creates the files /var/log/access_log.yyyy.mm.dd and /var/log/error_log.yyyy.mm.dd where yyyy is the year, mm is the month, and dd is the day of the month. The log operation switches to a new file every day at midnight, local time.

   For more information, see *rotatelogs - Piped logging program to rotate Apache logs*.
4. Click Save Changes.
5. Restart the affected services.

Related Information

rotatelogs - Piped logging program to rotate Apache logs

Hue supported browsers

Hue works with the two most recent LTS (long term support) or ESR (extended support release) browsers. Cookies and JavaScript must be enabled.

The lists the minimum tested versions of the most common browsers:

- Chrome: (Version history)
- Firefox: (Version history)
- Safari (Mac only): Version history
- Microsoft Edge: (Version history)

Hue can display in other browsers and in older versions of the common browsers, but you might not have access to all features.

Enabling cache-control HTTP headers when using Hue

You can enable Hue to use HTTP headers such as Cache-Control, Pragma, and Expires to ensure that your broswer always uses the fresh version of the resource. You must set the value of the custom_cache_control property to true in Hue's Advanced Configuration Snippet.

About this task

Note: Enabling the use of these cache-control HTTP headers can affect performance because your broswer tries to fetch the latest resources instead of using cached data. To disable cache-control HTTP headers, set the value of the custom_cache_control property to false.

Procedure

1. Log in to Cloudera Manager as an Administrator.
2. Go to Clusters Hue Configuration and add the following line in the Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini field:

   ```
   [desktop]
   custom_cache_control=true
   ```

3. Click Save Changes.
4. Restart the Hue service.

Adding a Hue service with Cloudera Manager

You can use the Add a Service wizard in Cloudera Manager to add and configure a new Hue service instance.

Procedure

1. Sign in to Cloudera Manager as an Administrator.
2. Go to Clusters Actions and click Add Service.

   The **Add Service to Cluster** wizard is displayed.
3. Select Hue as the Service Type and click Continue.

   The **Select Dependencies** page is displayed.
4. On the **Select Dependencies** page, select the optional dependencies that are required for your cluster and click Continue.

   The required dependency is selected by default.

   Note:

   In CDH 6 and earlier, the Hive service included the Hive Metastore and HiveServer2. In Cloudera Runtime 7.0 and later, this service only includes the Hive Metastore. HiveServer2 and other components of the Hive execution engines are part of the HIVE_ON_TEZ service.

   If you need to run Hive queries, then select an optional dependency containing the HIVE_ON_TEZ service.

   If you need to run only Impala queries, then the HIVE_ON_TEZ service is not required.

5. On the **Assign Roles** page, you can perform one of the following actions:

   - To accept the default role assignments, click Continue in the lower right corner of the page. To view the default role assignments by host, click View By Host.
   - To customize role assignments, click the field below Hue Server and Load Balancer roles. This launches a dialog box where you can select hosts where you want to add the role. Click OK after making your selection.

     The wizard evaluates host hardware configurations to determine the best hosts for each role. All worker roles are automatically assigned to the same set of hosts as the HDFS DataNode. You can reassign if necessary. Specify hostnames by IP address, rack name, or by range:

Range Definition	Matching Hosts
10.1.1.[1-4]	10.1.1.1, 10.1.1.2, 10.1.1.3, 10.1.1.4
host[1-3].company.com	host1.company.com, host2.company.com, host3.company.com
host[07-10].company.com	host07.company.com, host08.company.com, host09.company.com, host10.company.com

6. On the **Setup Database** page:
   a) Select a database vendor from the Type field and specify the Database Hostname, Database Name, Username, and Password.
   b) Click Test Connection, and when the success message appears, click Continue and Cloudera Manager starts the Hue service.
7. Review your configurations on the **Review Changes** page and click Continue.
8. On the **Command Details** page, expand the running commands to view the details of any step, including log files and command output. You can filter the view by selecting Show All Steps, Show Only Failed Steps, or Show Only Running Steps.

   After the First Run command completes, click Continue to go to the **Summary** page.
9. After the service is started, click Continue and then click Finish.

   If your cluster uses Kerberos, Cloudera Manager automatically adds a Hue Kerberos Ticket Renewer role to each host where you assigned the Hue Server role instance. See "Enabling Kerberos Authentication" for more information.

Adding a Hue role instance with Cloudera Manager

Roles are functions that comprise a service. Role instances must be assigned to one or more hosts. You can assign roles to hosts by using Cloudera Manager.

Procedure

1. In the Cloudera Manager Home page, select the Hue service.
2. On the Hue service page, click the Instances tab.
3. On the Instances tab page, click Add Role Instances on the right side of the page.
4. On the Assign Roles page:
   a) Click Select hosts under the role names to select hosts where you want to add the role. This launches a dialog box where you can select hosts where you want to add the role.

      The wizard evaluates host hardware configurations to determine the best hosts for each role. All worker roles are automatically assigned to the same set of hosts as the HDFS DataNode. You can reassign if necessary. Specify hostnames by IP address, rack name, or by range:

Range Definition	Matching Hosts
10.1.1.[1-4]	10.1.1.1, 10.1.1.2, 10.1.1.3, 10.1.1.4
host[1-3].company.com	host1.company.com, host2.company.com, host3.company.com

Range Definition	Matching Hosts
host[07-10].company.com	host07.company.com, host08.company.com, host09.company.com, host10.company.com

b) Click OK to save your hostname selection for a role.

c) When you are finished assigning roles to hosts, click Continue in the lower right corner of the page.

5. Cloudera Manager automatically adds the roles and returns you to the Hue Instances tab.

Note: If your cluster uses Kerberos, Cloudera Manager automatically adds a Hue Kerberos Ticket Renewer role to each host where you assigned the Hue Server role instance. Cloudera Manager throws a validation error if the new Hue Server role does not have a co-located Kerberos Ticket Renewer role. See "Enabling Kerberos for Authentication" for more information.

Options to restart the Hue service

Restarting the Hue service is necessary after upgrading CDP or making configuration changes to the Hue service. You can restart Hue in regular or rolling restart mode. Rolling restart reduces the service downtime to approximately 80 to 90 seconds.

You can restart the Hue service independently or as a part of the cluster restart. When you restart the Hue service normally, Hue can become unavailable for an average of 30 minutes from the restart time (this could vary depending on the number of services in your cluster). When you restart the cluster containg the Hue service, Hue is stopped first among other services in the cluster and restarted at the end. To reduce the service downtime, Cloudera recommends that you use the “Rolling Restart” option.

Note: Cluster-level rolling restart is supported only in High Availability (HA) clusters. However, you can restart the Hue service and its roles independently in the rolling restart mode in both HA and non-HA clusters.

Restarting the Hue service independently in rolling restart mode

To restart Hue service in the rolling restart mode, use one of the following steps:

- Go to Cloudera Manager Clusters and click ⋮ corresponding to the Hue service and then click Rolling Restart, or
- Go to Cloudera Manager Clusters Hue and click Actions Rolling Restart .

Cloudera Manager shows the following options to restart the Hue service in the rolling restart mode:

Category	Restart options	Description
Restart filters	Restart roles with stale configuration only	Select this option to restart Hue if you have updated the Hue configuration.
	Restart roles with the old software version only	Select this option to restart Hue if you have upgraded the cluster.
Role types to restart	Kerberos Ticket Renewer	Selected by default. Restarts the Kerberos Ticket Renewer role.
	Load Balancer	Selected by default. Restarts the Load Balancer role after the Kerberos Ticket Renewer.
	Hue Server	Selected by default. Restarts the Hue Server role after the load balancer.

When you restart Hue, Hue’s non-worker roles, such as the load balancer, Kerberos ticket renewer, and Hue server restart one after the other.

Restarting a CDP cluster containing the Hue service in the rolling restart mode

To restart an HA cluster containing the Hue service in the rolling restart mode, go to Cloudera Manager Clusters and click ⋮ corresponding to your cluster name and select Rolling Restart. You see the **Rolling Restart** modal with the list of services that support rolling restart.

Note: All Hue roles (namely the load balancer, Kerberos ticket renewer, and Hue server) are of the non-worker type. You must select one of the following options under the Roles to include section for ensuring that the Hue service is restarted in the rolling restart mode, depending on your requierments:

- Non-Workers Only, or
- All Roles

Related Information

Rolling Restart

Customizing the Hue web interface

To customize the Hue web interface, add configuration properties in Cloudera Manager. You can customize the banner, the page logo, the cache timeout setting, and you can enable or disable anonymous usage data collection.

Adding a custom banner

You can add a custom banner to the Hue web interface by adding your custom HTML to the Top Banner Custom HTML property in Cloudera Manager.

Procedure

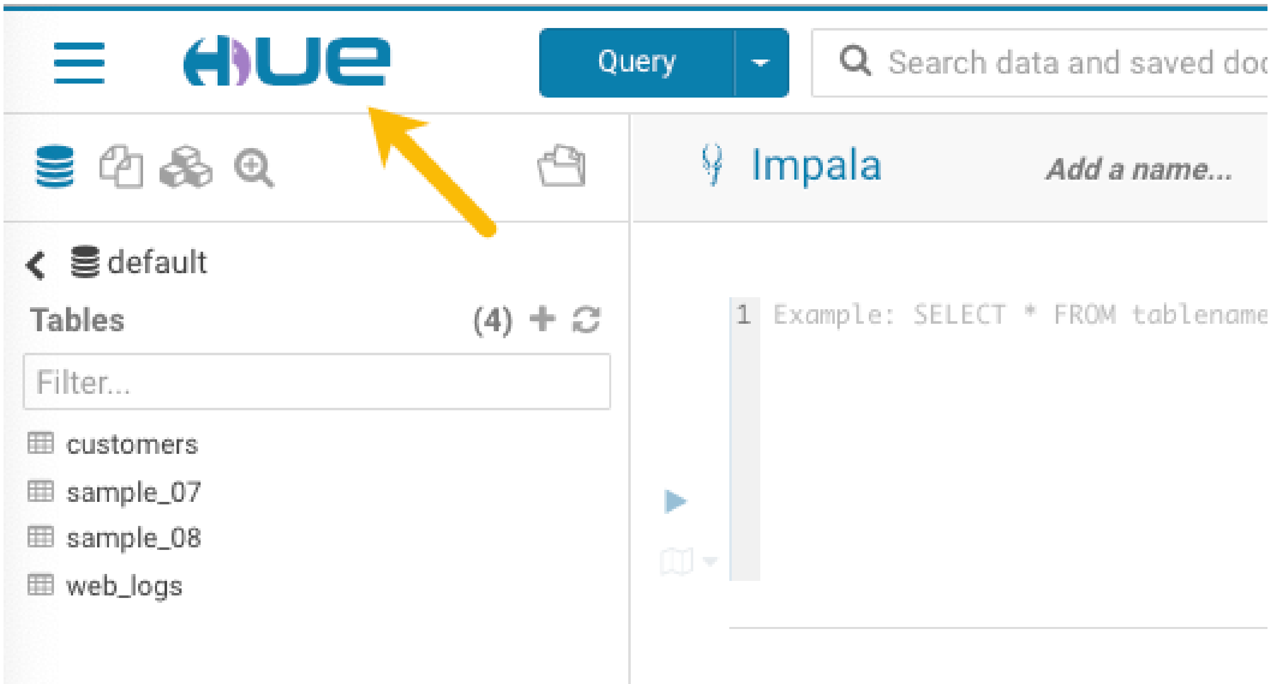
1. Log in to Cloudera Manager as an Administrator.
2. Go to Clusters Hue service Configuration and add the banner test in the Top Banner Custom HTML field as follows:
   <H1>Your company's custom Hue Web UI banner</H1>
3. Click Save Changes.
4. Restart the Hue service.

Changing the page logo

You can replace the Hue web interface logo with a custom log that is created with an SVG code. Add any type of logo you want, but your custom logo should be designed to fit into a 160 x 40 pixel space.

About this task

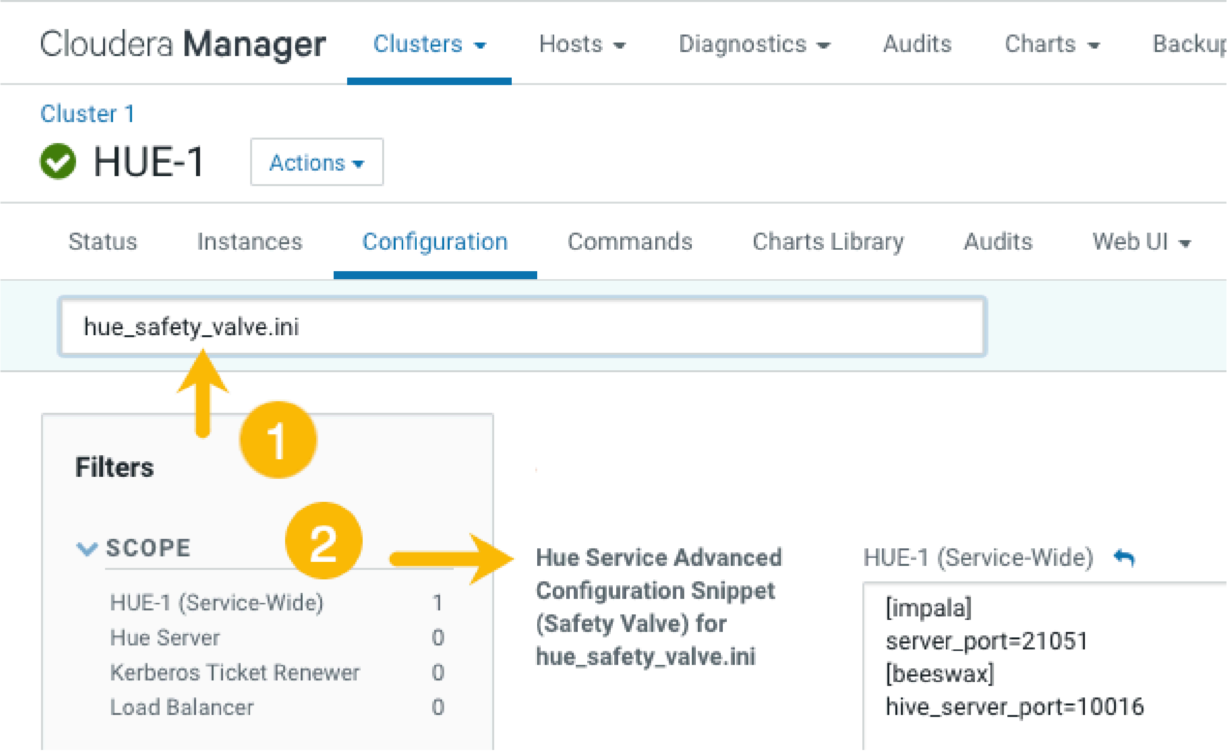
For example, here is the Hue logo shown in the following image:



You can change this Hue logo by adding the appropriate SVG code to the logo_svg property under the [desktop] [[custom]] section in the Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini configuration parameter in Cloudera Manager.

Procedure

1. Log in to Cloudera Manager as an Administrator.
2. Go to Clusters Hue Configuration
3. In the Search text box, type hue_safety_valve.ini to locate the configuration parameter:



hue_safety_valve.ini

Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini

```
[impala]
server_port=21051
[beeswax]
hive_server_port=10016
```

4. Add the following parameters with your custom logo SVG code to the Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini configuration parameter:

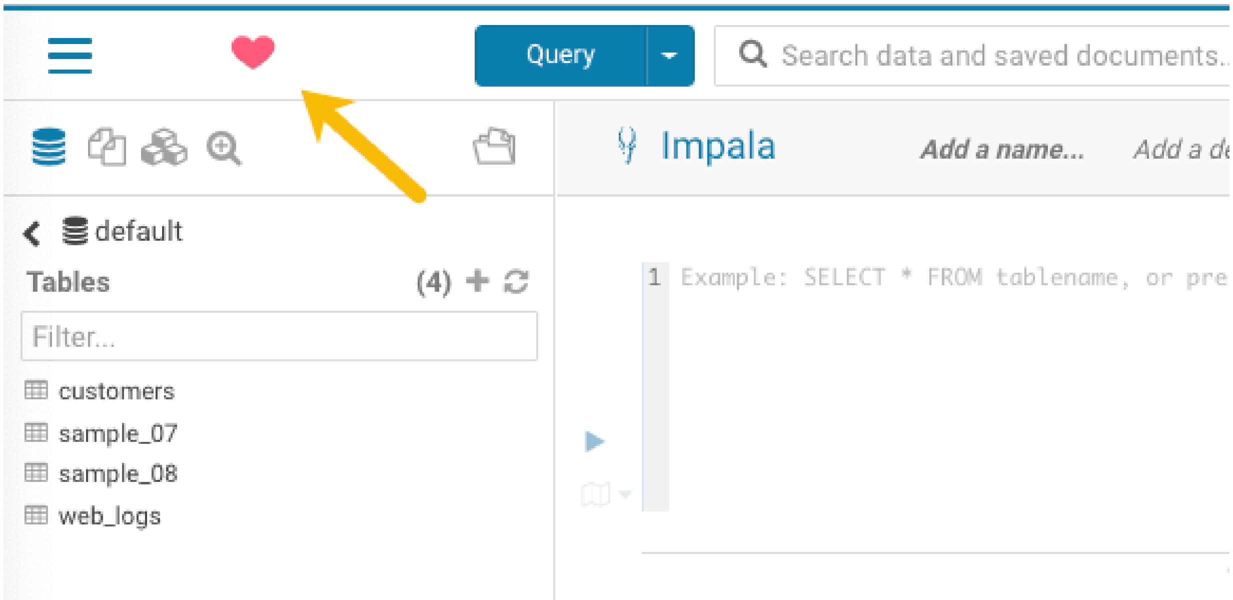
   ```
   [desktop]
   [[custom]]
   logo_svg='[***SVG-CODE-FOR-CUSTOM-LOGO***]'
   ```

   For example, the following SVG code replaces the Hue logo with a red heart:

   ```
   [desktop]
   [[custom]]
   logo_svg='<g><path stroke="null" id="svg_1" d="m44.41215,11.43463c-4.05
   017,-10.71473
   -17.19753,-5.90773 -18.41353,-0.5567c-1.672,-5.70253 -14.497,-9.95663
   -18.411,0.5643c-4.35797,11.71793 16.891,22.23443 18.41163,23.95773c1.518
   1,-1.36927 22.7696,-12.43803
   18.4129,-23.96533z" fill="#ffffff"/> <path stroke="null" id="svg_2"
   d="m98.41246,10.43463c-4.05016,-10.71473 -17.19753,-5.90773 -18.41353,-
   0.5567c-1.672,-5.70253
   -14.497,-9.95663 -18.411,0.5643c-4.35796,11.71793 16.891,22.23443 18.4116
   4,23.95773c1.5181,-1.36927
   22.76959,-12.43803 18.41289,-23.96533z" fill="#FF5A79"/> <path stroke="nu
   ll" id="svg_3"
   d="m154.41215,11.43463c-4.05016,-10.71473 -17.19753,-5.90773 -18.41353,-0
   .5567c-1.672,-5.70253
   -14.497,-9.95663 -18.411,0.5643c-4.35796,11.71793 16.891,22.23443 18.41164
   ,23.95773c1.5181,-1.36927 22.76959,-12.43803 18.41289,-23.96533z" fill="
   #ffffff"/> </g>'
   ```

5. Click Save Changes.
6. Restart the Hue service.
7. In the Hue configuration page of Cloudera Manager, select Web UI Hue Load Balanced to load Hue and view your custom logo.

   If you added the sample SVG code that defines a red heart as the logo, your Hue web UI looks like this:



8.
9.
10.

Related Information

Scalable Vector Graphics

Setting the cache timeout

Enable Hue UI caching by setting a timeout value in milliseconds. The default is 10 days or 864000000 milliseconds. Set the timeout to 0 to disable caching.

About this task

When you browse tables using the left assist panel or run queries, Hue caches this information for fetching information faster and query autocompletion. You can configure the time for which you want to Hue to cache this information by setting the value of the cacheable_ttl property under the [desktop][[custom]] section in the Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini configuration property in Cloudera Manager.

Procedure

1. Log in to Cloudera Manager as an Administrator.
2. Go to Clusters Hue Configuration .
3. In the Search text box, type hue_safety_valve.ini to locate the Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini configuration parameter.
4. Add the following parameters with the cache timeout value to the Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini configuration parameter:

   ```
   [desktop]
   [[custom]]
   cacheable_ttl=[***VALUE-IN-MILLISECONDS***]
   ```

   For example, the following configuration sets the cache timeout to the default value of 86400000 milliseconds:

   ```
   [desktop]
   [[custom]]
   cacheable_ttl=86400000
   ```

5. Click Save Changes.
6. Restart the Hue service.

Enabling or disabling anonymous usage date collection

Hue tracks anonymized pages and application versions to gather information about application usage levels. The data collected does not include hostnames or IDs. For example, the data collected has the format /2.3.0/pig or /2.5.0/beeswax/execute.

About this task

To enable or disable anonymous usage data collection:

Procedure

1. In the Cloudera Manager Admin Console, select ClustersHueConfiguration to navigate to the configuration page for Hue.
2. In the Search text box, type usage to locate the Enable Usage Data Collection check box:
   - To enable anonymous data collection, check the box, which is the default setting.
   - To disable anonymous data collection, clear the check box.

3. Enter a Reason for change..., and then click Save Changes at the bottom of the page to save the configuration change.
4. Refresh the browser page and click the restart icon ⏻ at the top of the page so the new configuration changes can be read by the server and the new data collection setting takes effect.

Configuring the number of objects displayed in Hue

By default, a specific number of databases, tables, and columns are displayed in the Hue web interface. You can increase this number by specifying a value for the MAX_CATALOG_SQL_ENTRIES property in the Hue Advanced Configuration Snippet using Cloudera Manager.

About this task

5000 objects (databases, tables, and columns) are displayed by default on the left assist panel, auto-complete drop-down list, and on the Table Browser page in the Hue web interface. Setting the value of the MAX_CATALOG_SQL_ENTRIES property higher than the default value increases the number of objects but can downgrade performance.

Procedure

1. Log in to Cloudera Manager as an Administrator.
2. Go to Clusters Hue Configuration Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini and add the following lines:

```
[beeswax]
max_catalog_sql_entries=[***NUMBER-OF-OBJECTS***]
```

3. Click Save Changes.
4. Restart the Hue service.

Using Oracle database with Hue

You can use the Oracle database to store Hue metadata, job and query history, and account information. In CDP 7.1.x, Hue supports Oracle database versions 12c and 19c.

To use an Oracle database with Hue, you need:

- Oracle Instant Client library: The Oracle Instant Client connects Hue to the backend Oracle database. The Oracle Client libraries provide the necessary network connectivity and tools such as SQL*Plus and Oracle Data Pump to access the data.
- cx_Oracle Python extension module: The cx_Oracle module loads the Oracle Client libraries. The cx_Oracle module and the Oracle Client libraries communicate with the data over Oracle Net.

Hue on CDP 7.1.8 and higher uses Python version 3.8.12 and Django version 3.2.13. The minimum compatible cx_Oracle version that you can use with Hue is 6.4.1. cx_Oracle 8.3.0 is distributed with CDP. If you are on an older version of CDP, or if you are upgrading to CDP from CDH 5.x or 6.x, then you must upgrade cx_Oracle from 5.2.1 to 6.4.1.

Attention: Hue on 7.1.8 and higher supports Oracle Database Server versions 12.2 and higher and cx_Oracle (the Python interface for Oracle database) versions 6.0 and higher.

To use Oracle as a backend database for Hue, you must download and install the Oracle Instant Client library and cx_Oracle Python extension module, and then set the database parameters in Cloudera Manager Hue configurations.

You must download the Oracle Instant Client library 19.16 (both basic and SDK clients) from the Oracle website.

Installing and configuring the Oracle server

Follow the Oracle documentation to install the database. This step is required only if you do not have the Oracle database installed in your data center/environment. If you are using an existing Oracle database, then you can skip this step.

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

Related Information

Oracle documentation

(Optional) Configuring the character set

AL16UTF16 is the default value of the National Character Set (NLS_NCHAR_CHARACTERSET) for the Oracle database. Cloudera recommends that you use the default NLS_NCHAR_CHARACTERSET. However, if needed, you can configure the National Character Set to UTF-8.

Procedure

1. Log in as the oracle user:

   ```
   su - oracle
   ```

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

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

3. Log in to SQL*Plus:

   ```
   sqlplus / as sysdba
   ```

4. Check whether the 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
   ```

Creating Hue Schema in Oracle database

Hue requires a SQL database to store small amounts of data, including user account information as well as the history of queries and sharing permissions. Ensure that tables in the Hue schema are not compressed.

Before you begin

Due to a known bug in the Oracle database (12c and higher), you cannot alter a table if it is compressed. If you have compressed tables in the Hue schema, then you may see the “ORA-39726: unsupported add/drop column operation on compressed tables” error. Even if you uncompress an existing table, you may not be allowed to alter the columns. To resolve this, follow the below steps:

1. Create a new uncompressed table with the same structure as the compressed table.
2. Copy the data from the compressed table to the new uncompressed table.
3. Rename or delete the compressed table.
4. Rename the uncompressed table with the name of the original compressed table.

Procedure

1. Create the Hue schema, set quotas, and grant select permissions (do not grant all).

   Note: Oracle 12c users must ALTER session set to avoid creating a common user with the 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
   ```

2. Verify that you can connect to Hue by running the following command:

   ```
   sqlplus hue/[***HUE-PASSWORD***]
   ```

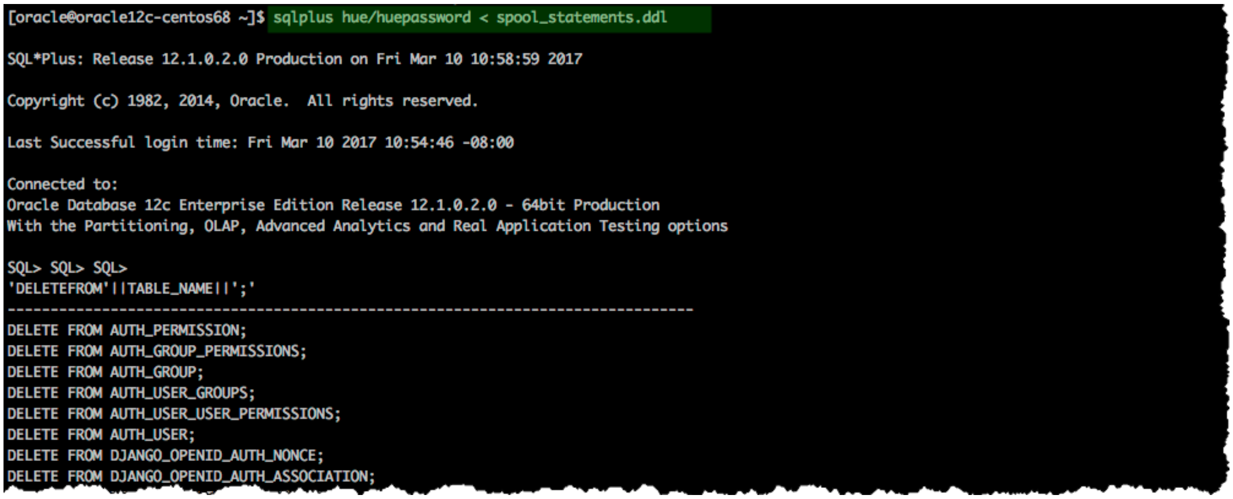
3. Clean all Hue user tables. Create a script to spool delete statements into a new file.

   ```
   delete_from_tables.ddl
   vi spool_statements.ddl
   ```

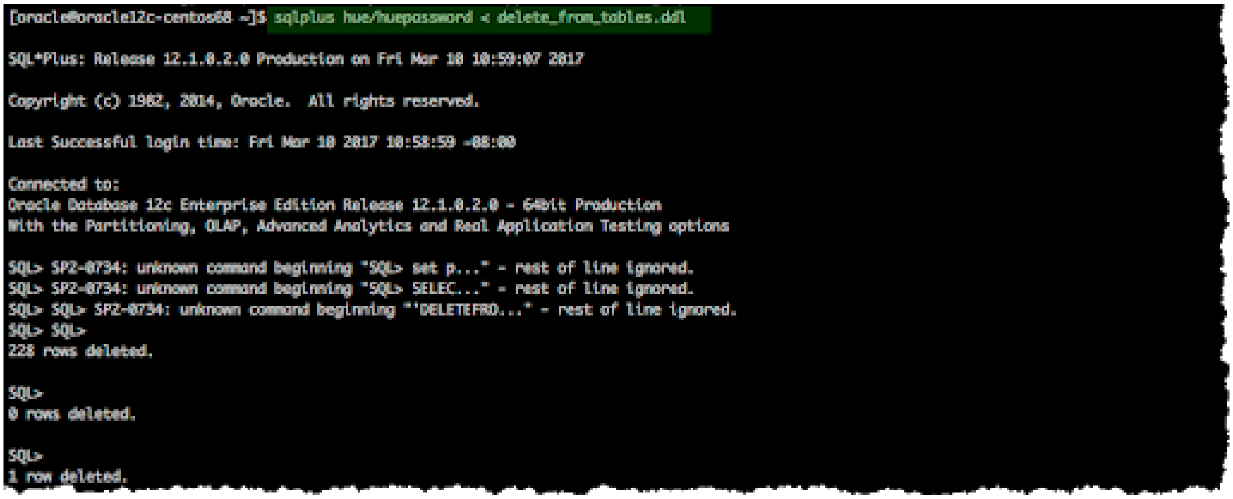
   ```
   ## Save in spool_statements.ddl (which generates delete_from_tables.ddl)
   spool delete_from_tables.ddl
   set pagesize 100;
   SELECT 'DELETE FROM ' || table_name || ';' FROM user_tables;
   ```

```
commit;
spool off
quit
```

```
## Create delete_from_tables.ddl
sqlplus hue/[***HUE-PASSWORD***] < spool_statements.ddl

## Run delete_from_tables.ddl
sqlplus hue/[***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

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> 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.

SQL>
0 rows deleted.

SQL>
1 row deleted.
```

Downloading, staging, and activating the Oracle Instant Client parcel

Cloudera distribution does not contain the Oracle Instant Client parcel. You must download the Oracle Instant Client library 19.16 (both basic and SDK clients) from the Oracle website.

About this task

Note: Hue on CDP 7.1.8 and higher supports Oracle Database Server versions 12.2 and higher.

Before you begin

You must install the libaio1 package on all hosts that run the Hue server.

On Ubuntu, run the following command to install the libaio1 package:

```
sudo apt-get install libaio1
```

Procedure

1. SSH into the Hue server host as a root user.
2. Download the zip files for the Instant Client Package, both Basic and SDK (with headers)
3. Change directory to /usr/share/oracle.
4. Decompress the downloaded basic and SDK files as follows:

   ```
   unzip /tmp/instantclient-basiclite-linux.x64-19.16.0.0.0dbru.zip -d .
   ```

   ```
   unzip /tmp/instantclient-sdk-linux.x64-19.16.0.0.0dbru.zip -d .
   ```

5. Arrange the client libraries to mirror the tree structure as shown in the following example:

   ```
   # 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/instantc
   lient_[***VER***]/
   unzip '*.zip' -d /usr/share/oracle/instantclient/
   # Move lib files from instantclient_[***VER***] to /usr/share/oracle/in
   stantclient/lib/
   mv /usr/share/oracle/instantclient/`ls -l /usr/share/oracle/instantclie
   nt/ | grep instantclient_ | awk '{print $9}'`/lib* /usr/share/oracle/ins
   tantclient/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/instantcl
   ient/

   # Create symbolic links. Remember to edit version numbers as necessary
   cd /usr/share/oracle/instantclient/lib
   ln -s libclntsh.so.[***VER***].1 libclntsh.so
   ln -s libocci.so.[***VER***].1 libocci.so
   ```

   Where, [***VER***] is the version of the Instant Client Package. Replace [***VER***] with the actual version of the Instant Client Package.

6. Set the ORACLE_HOME and LD_LIBRARY_PATH environment variables as follows:

   ```
   export ORACLE_HOME=/usr/share/oracle/instantclient;
   ```

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

Configuring Oracle as backend database for Hue

You can configure Oracle as the backend database for Hue after you have downloaded, distributed, activated the Oracle Instant Client parcel either using Oracle Service Name or SID.

About this task

Use the instructions in this topic to connect Hue to a regular Oracle database. To connect Hue to a TCPS-enable Oracle database, see Set up and configure TLS 1.2 for Hue.

Procedure

1. Log into Cloudera Manager as an Administrator.
2. Go to Clusters Hue service Configuration Category Database .

   All the database configurations for Hue are displayed.
3. Select Oracle as the Hue Database Type.
4. Specify the host on which you have installed the Oracle database in the Hue Database Hostname field.
5. Enter 1521 in the Hue Database Port field.
6. Enter user name and password in Hue Database Username and Hue Database Password fields.
7. Specify the database name in the Hue Database Name field.

   If you are using Oracle Service Name instead of SID, then specify the database name in the following format:

   ```
   [***HUE-DB-HOST***]:1521/[***SERVICE-NAME***]
   ```

   You can also configure additional parameters by entering the following in the Hue Database Name field:

   ```
   (DESCRIPTION=(LOAD_BALANCE=off)(FAILOVER=on)(CONNECT_TIMEOUT=5)(TRANSPOR
   T_CONNECT_TIMEOUT=3)(RETRY_COUNT=3)(ADDRESS=(PROTOCOL=TCP)(HOST=[***HUE-
   DB-HOST***])(PORT=1521))(CONNECT_DATA=(SERVICE_NAME=[***SERVICE-NAME***]
   )))
   ```
8. Go to Clusters Hue service Configuration Hue Service Environment Advanced Configuration Snippet (Safety Valve) and add the following parameters:

   Key: LD_LIBRARY_PATH

   Value: /opt/cloudera/parcels/ORACLE_INSTANT_CLIENT/[***INSTANT-CLIENT-WITH-VERSION***]

   Key: ORACLE_HOME

   Value: /opt/cloudera/parcels/ORACLE_INSTANT_CLIENT/[***INSTANT-CLIENT-WITH-VERSION***]

   For example: /opt/cloudera/parcels/ORACLE_INSTANT_CLIENT/instantclient_12_2

   You can obtain the value of the Instant Client version by running the following command from the Cloudera Manager server terminal /opt/cloudera/parcels/ directory:

   ```
   ls -al ORACLE_INSTANT_CLIENT/
   ```

   If you have manually set the values for LD_LIBRARY_PATH and ORACLE_HOME, then specify those.
9. Add support for a multi-threaded environment by adding the following lines in the Hue Service Environment Advanced Configuration Snippet (Safety Valve) field under the [desktop] section:

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

   A multi-threaded environment is recommended for using the Oracle database.

   If you are using Oracle Service Name instead of SID, then add the following line under the [desktop] section in the Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini field:

   ```
   [desktop]
   [[database]]
   port=0
   ```
10. Click Save Changes and restart the Hue service.

Results

If the database setup is successful and Hue starts without errors, then you should be able to log into the Hue web interface and view the database/tables.

Configuring high availability support for Oracle RAC database

For seamless connectivity and high availability, you can use Oracle Real Application Clusters (RAC) database as Hue’s backend database.

Before you begin

- You must have Oracle RAC installed on your CDP cluster.
- You must have configured Oracle as Hue’s backend database.

Procedure

1. Log in to Cloudera Manager as an Administrator.
2. Go to Clusters Hue Configuration Category Database .

   The database configurations are displayed.
3. Specify the database name in the following format in the Hue Database Name field:

```
(DESCRIPTION=(LOAD_BALANCE=off)(FAILOVER=on)(CONNECT_TIMEOUT=5)(TRANSPOR
T_CONNECT_TIMEOUT=3)(RETRY_COUNT=3)(ADDRESS=(PROTOCOL=TCP)(HOST=[***HOST
NAME-OR-SCAN-IP***])(PORT=[***PORT***]))(CONNECT_DATA=(SERVICE_NAME=[***
SERVICE-NAME***])))
```

4. Click Save Changes.
5. Restart the Hue service.

Using MySQL database with Hue

Hue is packaged with a lightweight embedded PostgreSQL database for proof-of-concept deployments with one Hue server. However, you can install and use a custom MySQL database as the backend database to store Hue metadata, job and query history, and account information.

Note: MySQL 5.7 has reached its end of life (EOL) and is now covered under Oracle Lifetime Sustaining Support. Per Oracle's Lifetime Support policy, as of October 25, 2023, MySQL 5.7 is covered under Oracle Sustaining Support. You are encouraged to upgrade to MySQL 8.0.

Related Information

MySQL documentation

Downloading and installing MySQL database

Download and install MySQL database depending on your CDP and Operating System (OS) version.

About this task

Note: MySQL 5.7 has reached its end of life (EOL) and is now covered under Oracle Lifetime Sustaining Support. Per Oracle's Lifetime Support policy, as of October 25, 2023, MySQL 5.7 is covered under Oracle Sustaining Support. You are encouraged to upgrade to MySQL 8.0.

Procedure

1. Download the MySQL distribution from https://dev.mysql.com/downloads/ based on your OS version.

   Use the Cloudera Support Matrix for finding the compatibile versions.

2. Install the database using the following commands depending on your operating system:

   On RHEL/CentOS:

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

   On SLES:

   ```
   sudo zypper install mysql
   ```

   On Ubuntu:

   ```
   sudo apt-get install mysql-server    #set root psswd when prompted
   ```

   On Debian:

   ```
   sudo apt-get install mysql-server    #set root psswd when prompted
   ```
3. Start the database server using the following commands depending on your operating system:

   On RHEL/CentOS:

   ```
   sudo service mysqld start
   ```

   On SLES:

   ```
   sudo rcmysql start
   ```

   On Ubuntu:

   ```
   sudo service mysql start
   ```

   On Debian:

   ```
   sudo service mysql start
   ```
4. Secure your installation by running the following command:

   ```
   sudo /usr/bin/mysql_secure_installation
   ```

   mysql_secure_installation is a shell script available on Unix systems, and enables you to improve the security of your MySQL installation in the following ways:

   ```
   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
   ```

Configuring MySQL server

The MySQL server, mysqld, has many command options and system variables that can be set at startup to configure its operation. The configurations are stored in the my.cnf file.

Procedure

1. SSH into the database server host as a root user.
2. Open the my.cnf file for editing in an editor such as vi or nano:

   On RHEL/CentOS/SLES:

   ```
   vi /etc/my.cnf
   ```

   On Debian/Ubuntu:

   ```
   vi /etc/mysql/my.cnf
   ```
3. Ensure that the bind-address property is set to 0.0.0.0 or is commented out if case of a default value.

   ```
   bind-address=0.0.0.0
   ```
4. Ensure that the default-storage-engine property is set to innodb, which is the default storage engine in MySQL versions 5.5 and higher.

   ```
   default-storage-engine=innodb
   ```
5. Ensure that the sql_mode property is set to STRICT_ALL_TABLES to avoid columns being truncated during migration.

   ```
   sql_mode=STRICT_ALL_TABLES
   ```
6. Save the file and exit.
7. Restart the database server.
8. Enable the server to automatically start on boot:

   On RHEL/CentOS:

   ```
   sudo chkconfig mysqld on
   ```

   On SLES:

   ```
   sudo chkconfig mysql on
   sudo rcmysql status
   ```

   On Ubuntu, MySQL starts automatically after installation. Run the following command to check the status:

   ```
   sudo service mysql status
   ```

   On Debian, MySQL starts automatically after installation. Run the following command to check the status:

   ```
   sudo service mysql status
   ```

Installing and configuring MySQL on RHEL 8

The steps for installing and configuring MySQL version 8.0.25 on RHEL 8 on a CDP cluster host are different. Make sure that you copy the corresponding my__config_8h_source.html content to the my_config.h file corresponding to your MySQL server version.

About this task

Note: MySQL 5.7 has reached its end of life (EOL) and is now covered under Oracle Lifetime Sustaining Support. Per Oracle's Lifetime Support policy, as of October 25, 2023, MySQL 5.7 is covered under Oracle Sustaining Support. You are encouraged to upgrade to MySQL 8.0.

Procedure

1. SSH into the Hue server host.
2. Install the driver for connecting to the MySQL database server:

```
yum install perl-DBD-MySQL
```

3. Configure the RHEL 8 OS repository (/etc/yum.repos.d/mysql8.repo) to point to http://repo.mysql.com/yum/mysql-8.0-community/el/8/x86_64/.
4. Install MySQL community server:

```
yum install mysql-community-server-8.0.25 mysql-community-common-8.0.25
mysql-community-devel-8.0.25
```

5. Copy the my__config_8h_source.html configuration file content to /usr/include/mysql/my_config.h.
6. Install the MySQL Python interface:

```
pip install MySQL-python --force-reinstall --ignore-installed
```

7. Grant read and execute permissions to all users and write access to the admin user:

```
chmod 755 chmod -R 755 /usr/lib64/python2.7/site-packages/MySQLdb /usr/lib64/python2.7/site-packages/MySQL_python* /usr/lib64/python2.7/site-packages/_mysql*
```

What to do next

After installing the MySQL server, create the Hue database and configure MySQL as the backend database for Hue.

Installing MySQL client for MySQL databases

To use MySQL as a backend database for Hue, you must install the MySQL client and other required dependencies on all the Hue hosts based on your operating system.

Attention: The version 2.2.0 of the MySQL client requires that you set the values of the MYSQLCLIENT_CFLAGS and MYSQLCLIENT_LDFLAGS environment variables, as follows:

```
$ export MYSQLCLIENT_CFLAGS=`pkg-config mysqlclient --cflags`
$ export MYSQLCLIENT_LDFLAGS=`pkg-config mysqlclient --libs`
```

The version 2.2.0 of the MySQL client also requires you to install the Python 3 and MySQL development headers and libraries, as follows on Debian or Ubuntu operating systems:

```
sudo apt-get install python3-dev default-libmysqlclient-dev build-essential
```

or as follows on CentOS and RHEL operating systems:

```
sudo yum install python3-devel mysql-devel
```

Alternatively, you can install and use the version 2.1.1 of the MySQL client, as follows, which does not have this requirement:

```
pip3 install mysqlclient==2.1.1
```

For Cent OS

1. SSH into the Hue host as a root user.
2. Download the MySQL yum repository as follows:

   ```
   curl -sSLO https://dev.mysql.com/get/mysql80-community-release-el7-5.noarch.rpm
   ```
3. Install the package as follows:

   ```
   rpm -ivh mysql80-community-release-el7-5.noarch.rpm
   ```
4. Install the required dependencies as follows:

   ```
   yum install mysql-devel
   yum install -y xmlsec1  xmlsec1-openssl
   ```

   For MySQL version 8.0.27, add the mysql-community-client-8.0.25 client package as follows:

   ```
   yum install mysql-community-client-8.0.25
   ```
5. Add the path where you installed the MySQL client and packages to the PATH environment variable as follows:

   ```
   export PATH=/usr/local/bin:$PATH
   ```
6. Install the MySQL client as follows:

   ```
   pip3.8 install mysqlclient
   ```

For RHEL

1. SSH into the Hue host as a root user.

2. Download the MySQL yum repository as follows:

   (RHEL 7)

   ```
   curl -sSLO https://dev.mysql.com/get/mysql80-community-release-el7-5.noa
   rch.rpm
   ```

   (RHEL 8)

   ```
   curl -sSLO https://dev.mysql.com/get/mysql80-community-release-el8-8.noa
   rch.rpm
   ```

   (RHEL 9)

   ```
   curl -sSLO https://dev.mysql.com/get/mysql80-community-release-el9-4.noa
   rch.rpm
   ```
3. Install the package as follows:

   (RHEL 7)

   ```
   rpm -ivh mysql80-community-release-el7-5.noarch.rpm
   ```

   (RHEL 8)

   ```
   rpm -ivh mysql80-community-release-el8-8.noarch.rpm
   ```

   (RHEL 9)

   ```
   rpm -ivh mysql80-community-release-el9-4.noarch.rpm
   ```
4. Install the required dependencies as follows:

   ```
   yum install mysql-devel
   yum install -y xmlsec1  xmlsec1-openssl
   ```
5. Add the path where you installed the MySQL client and packages to the PATH environment variable as follows:

   ```
   export PATH=/usr/local/bin:$PATH
   ```
6. Install the MySQL client as follows:

   (RHEL 8)

   ```
   pip3.8 install mysqlclient
   ```

   (RHEL 9)

   ```
   pip3.9 install mysqlclient
   ```

For SLES

1. SSH into the Hue host as a root user.
2. Install the required packages and dependencies as follows:

   ```
   zypper install libmysqlclient-devel
   zypper install xmlsec1
   zypper install xmlsec1-devel
   zypper install xmlsec1-openssl-devel
   ```

3. Add the path where you installed the packages to the PATH environment variable as follows:

```
export PATH=/usr/local/bin:$PATH
```

4. Install the MySQL client as follows:

```
pip3.8 install mysqlclient
```

For Ubuntu

1. SSH into the Hue host as a root user.
2. Install the required packages and dependencies as follows:

```
apt-get install libmysqlclient-dev
apt-get install -y xmlsec1
apt-get install libxmlsec1-openssl
```

3. Add the path where you installed the packages to the PATH environment variable as follows:

```
export PATH=/usr/local/bin:$PATH
```

4. Install the MySQL client as follows:

```
pip3.8 install mysqlclient
```

Creating the Hue database

Hue requires a SQL database to store small amounts of data, including user account information as well as the history of queries and sharing permissions. You must create a database for Hue with UTF-8 character set and grant user privileges.

Procedure

1. SSH into the database server host as the root user.
2. Create a database for Hue with UTF-8 collation and grant user privileges:

```
CREATE DATABASE [***HUE-DATABASE-NAME***] DEFAULT CHARACTER SET utf8 DEF
AULT COLLATE utf8_general_ci;
GRANT ALL ON [***HUE-DATABASE-NAME***].* TO '[***HUE-DATABASE-NAME***]'@'%
' IDENTIFIED BY '[***PASSWORD***]';
```

   Replace [***HUE-DATABASE-NAME***] with the actual Hue database name such as "hue".

   Ensure Hue uses UTF-8 collation and character set.

```
# To create (use utf8_general_ci or utf8mb4_general_ci):
CREATE DATABASE [***HUE-DATABASE-NAME***] 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-DATABASE-NAME***] COLLATE = 'utf8_general_ci';
```

3. Verify the connection to the Hue database by running the following command:

```
mysql -u [***HUE-DATABASE-NAME***] -p
Enter password: [***PASSWORD***]
```

Configuring MySQL as the backend database for Hue

After installing and configuring the MySQL database, you can configure Hue to use it as its backend database.

Procedure

1. Log into Cloudera Manager as an Administrator.
2. Go to Clusters Hue service Configuration and filter by Database category.
3. Select the Hue Database Type as MySQL.
4. Set the following database parameters:

   Hue Database Hostname: FQDN of host running MySQL server

   Hue Database Port: 3306

   Hue Database Username: [***USERNAME***]

   Hue Database Password: [***PASSWORD***]

   Hue Database Name: [***HUE-DATABASE-NAME***] or SID
5. Click Save Changes.
6. Restart the Hue service.

Results

You can click Hue Web UI to log on to Hue with a custom MySQL database.

Configuring MySQL server enforced with TLS 1.2 or higher to connect to Hue

If your MySQL server is configured to use TLS version 1.2 or higher for an encrypted connection with Hue, then you may not be able to start the Hue server and the Kerberos ticket renewer. One solution to fix this problem is to upgrade the MySQL client to version 8.

About this task

One of the errors you may see while connecting to Hue if your MySQL server is configured with TLS 1.2 or higher is: “The health test result for HUE_SERVER_SCM_HEALTH has become bad: This role's process failed to start.” The error might be because the MySQL-Python module for Hue that is shipped with CDP uses an older libmysqlclient library which does not support TLS 1.2 and higher.

Important: You must install MySQL database version 8.x to use TLS 1.2-enforced MySQL server with Hue because it uses a newer libmysqlclient library, which supports TLS 1.2 and higher.

Procedure

1. SSH into the Hue server host as a root user.
2. Change directory to /etc/cloudera-scm-agent/ and take a backup of the config.ini file.
3. Run the following command to get a list of all the installed MySQL packages:

   ```
   rpm -qa | grep -i mysql*
   ```

   Save the list of all the installed MySQL packages for future reference.

4. Remove all MySQL Community RPM packages and the Cloudera Manager agent:

   ```
   yum erase mysql-community*|cloudera-manager-agent
   ```

   Uninstalling the Cloudera Manager agent also uninstalls the MySQL-python library. Save the list of all the MySQL packages that are uninstalled for future reference.
5. Reinstall the Cloudera Manager agent as follows:

   ```
   yum -y install cloudera-manager-agent
   ```

   This installs all the required packages:

   ```
   Package Arch Version Repository Size
   ===============================================================
   Installing:
   cloudera-manager-agent x86_64 7.2.4-7594142.el7 cloudera-manager 12 M
   Installing for dependencies:
   MySQL-python x86_64 1.2.5-1.el7 base 90 k
   mysql-community-client-plugins x86_64 8.0.24-1.el7 mysql80-community 189 k
   mysql-community-common x86_64 8.0.24-1.el7 mysql80-community 614 k
   mysql-community-libs x86_64 8.0.24-1.el7 mysql80-community 4.0 M
   mysql-community-libs-compat x86_64 8.0.24-1.el7 mysql80-community 1.2 M

   Transaction Summary
   ===============================================================
   ```
6. Install the mysql-devel (the development header files and libraries for MySQL database client applications):

   ```
   yum install mysql-devel
   ```

   ```
   ===============================================================
   Package Arch Version Repository Size
   ===============================================================
   Installing:
   mysql-community-devel x86_64 8.0.24-1.el7 mysql80-community 6.5 M

   Transaction Summary
   ===============================================================
   ```
7. Install the Python developer package:

   ```
   yum install python-devel.x86_64 0:2.7.5-90.el7
   ```

   ```
   ===============================================================
   Package Arch Version Repository Size
   ===============================================================
   Removing:
   python-devel x86_64 2.7.5-90.el7 @updates 1.0 M

   Transaction Summary
   ===============================================================
   ```
8. Remove the existing _mysql.so file:

   ```
   rm -rf /opt/cloudera/parcels/CDH-[***7.x-VERSION***]/lib/hue/build/env/l
   ib/python2.7/site-packages/_mysql.so
   ```

9. Change directory to /usr/include/mysql and check whether the my_config.h file is present.

   If the my_config.h file is present, then move to the next step.

   If the my_config.h file is not present, then:

   a) Go to https://dev.mysql.com/doc/dev/mysql-server/latest/my__config_8h_source.html and copy the contents of the my_config.h file.
   b) Create a new my_config.h file using the following command:

      ```
      vi /usr/include/mysql/my_config.h
      ```

   c) Paste the content that you copied earlier in the /usr/include/mysql/my_config.h file and save it.
10. Install the MySQL Python library as follows:

    ```
    /opt/cloudera/parcels/CDH/lib/hue/build/env/bin/pip install MySQL-python
     --force-reinstall --ignore-installed
    ```

11. Copy the _mysql.so (MySQL client library) to Cloudera's local parcel directory:
    a) Locate the _mysql.so file:

       locate _mysql.so

       Output: /usr/lib64/python2.7/site-packages/_mysql.so
    b) Copy the _mysql.so file to Cloudera's local parcel directory:

       ```
       cp /usr/lib64/python2.7/site-packages/_mysql.so /opt/cloudera/parcels/CD
       H-[***7.x-VERSION***]/lib/hue/build/env/lib/python2.7/site-packages/MySQ
       L_python-1.2.5-py2.7-linux-x86_64.egg/_mysql.so
       ```

12. Log in to Cloudera Manager and restart the Hue service.

Using MariaDB database with Hue

Hue is packaged with a lightweight embedded PostgreSQL database for proof-of-concept deployments with one Hue server. However, you can install and use a custom MariaDB database as the backend database to store Hue metadata, job and query history, and account information.

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

Related Information

MariaDB product website

MariaDB documentation

Downloading and installing MariaDB database

Download and install MariaDB database on a host in your CDP cluster depending on your CDP and OS version.

Procedure

1. Download the MariaDB distribution from https://mariadb.org/download/ based on your OS version.

   Use the Cloudera Support Matrix for finding the compatibile versions.

2. Install the database using the following commands depending on your operating system:

   On RHEL/CentOS:

   ```
   sudo yum install mariadb-server
   ```

   On SLES:

   ```
   sudo zypper install mariadb
   ```

   On Ubuntu:

   ```
   sudo apt-get install mariadb-server    #set root psswd when prompted
   ```

   On Debian:

   ```
   sudo apt-get install mariadb-server    #set root psswd when prompted
   ```
3. Start the database server using the following commands depending on your operating system:

   On RHEL/CentOS:

   ```
   sudo systemctl start mariadb
   ```

   On SLES:

   ```
   sudo systemctl start mariadb
   ```

   On Ubuntu, MariaDB starts automatically after installation. Run the following command to check the status:

   ```
   sudo systemctl status mariadb
   ```

   On Debian, MariaDB starts automatically after installation. Run the following command to check the status:

   ```
   sudo systemctl status mariadb
   ```
4. Secure your installation by running the following command:

   ```
   sudo /usr/bin/mysql_secure_installation
   ```

   mysql_secure_installation is a shell script available on Unix systems, and enables you to improve the security of your MariaDB installation in the following ways:

   ```
   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
   ```

Configuring MariaDB server

You can custom configure the MariaDB database by configuring the MariaDB server with the option files. The default MariaDB option file is called my.cnf on Unix-like operating systems.

Procedure

1. SSH into the database server host as a root user.
2. Open the my.cnf file for editing in an editor such as vi or nano:

   On RHEL/CentOS/SLES:

   ```
   vi /etc/my.cnf
   ```

   On Debian/Ubuntu:

   ```
   vi /etc/mysql/my.cnf
   ```
3. Ensure that the bind-address property is set to 0.0.0.0 or is commented out in case of a default value.

   ```
   bind-address=0.0.0.0
   ```
4. Ensure that the default-storage-engine property is set to innodb.

   ```
   [mysqld]
   default-storage-engine=innodb
   ```
5. Ensure that the sql_mode property is set to STRICT_ALL_TABLES to avoid columns being truncated during migration.

   ```
   sql_mode=STRICT_ALL_TABLES
   ```
6. Save the file and exit.
7. Restart the database server.
8. Enable the database server to automatically start on system boot:

   On RHEL/CentOS:

   ```
   sudo systemctl enable mariadb
   ```

   On SLES:

   ```
   sudo systemctl enable mariadb
   ```

   On Ubuntu, MariaDB starts automatically after installation. Run the following command to check the status:

   ```
   sudo systemctl status mariadb
   ```

   On Debian, MariaDB starts automatically after installation. Run the following command to check the status:

   ```
   sudo systemctl status mariadb
   ```

Installing and configuring MariaDB on RHEL 8

The steps for installing MariaDB versions 10.3 and 10.4 on RHEL 8 on a CDP cluster host are different.

Procedure

1. SSH into the Hue server host.
2. Remove the MariaDB GSSAPI server.

   ```
   yum remove mariadb-gssapi-server
   ```

3. Download the MariaDB repository:

   ```
   curl -LsS https://downloads.mariadb.com/MariaDB/mariadb_repo_setup | bash
    -s – --mariadb-server-version="mariadb-[***VERSION***]"
   ```

   Replace *[***VERSION***]* with the actual MariaDB version.
4. Install the MariaDB client and server:

   ```
   yum install MariaDB-server MariaDB-client MariaDB-common MariaDB-devel m
   ariadb-libs
   ```

5. Run the following stream editor command to filter and transform the text:

   ```
   sed '/st_mysql_options options;/a unsigned int reconnect;' /usr/include/
   mysql/mysql.h
   ```

6. Install the MySQL Python interface:

   ```
   pip install MySQL-python --force-reinstall --ignore-installed
   ```

7. Grant read and execute permissions to all users and write access to the admin user:

   ```
   chmod -R 755 /usr/lib64/python2.7/site-packages/MySQLdb /usr/lib64/pytho
   n2.7/site-packages/MySQL_python*; chmod 755 /usr/lib64/python2.7/site-pa
   ckages/_mysql*
   ```

8. After installing the CDP parcels, copy the site packages:

   ```
   copy /usr/lib64/python2.7/site-packages/_mysql.so /opt/cloudera/parcels/
   CDH/lib/hue/build/env/lib/python2.7/site-packages/MySQL_python-1.2.5-py2
   .7-linux-x86_64.egg/
   ```

What to do next

After installing the MariaDB server, create the Hue database and configure MariaDB as the backend database for Hue.

Installing MySQL client for MariaDB databases

To use MariaDB as a backend database for Hue, you must install the MySQL client and other required dependencies on all the Hue hosts based on your operating system.

For Cent OS

1. SSH into the Hue host as a root user.
2. Install the required dependencies as follows:

   ```
   yum install -y xmlsec1  xmlsec1-openssl
   ```

3. Add the path where you installed the packages to the PATH environment variable as follows:

   ```
   export PATH=/usr/local/bin:$PATH
   ```

4. Install the MySQL client as follows:

```
pip3.8 install mysqlclient
```

For RHEL

1. SSH into the Hue host as a root user.
2. Install the required dependencies as follows:

```
yum install mysql-devel
yum install -y xmlsec1  xmlsec1-openssl
```

3. Add the path where you installed the packages to the PATH environment variable as follows:

```
export PATH=/usr/local/bin:$PATH
```

4. Install the MySQL client as follows:

   (RHEL 8)

```
pip3.8 install mysqlclient
```

   (RHEL 9)

```
pip3.9 install mysqlclient
```

For SLES

1. SSH into the Hue host as a root user.
2. Install the required packages and dependencies as follows:

```
zypper install libmysqlclient-devel
zypper install xmlsec1
zypper install xmlsec1-devel
zypper install xmlsec1-openssl-devel
```

> ⚠ **Attention:** While installing the mysql-devel and libmysqlclient-devel packages on SLES15, use the "--replacefiles" zypper switch or manually enter yes on the interactive pop-up that you see when the files are being overwritten. Else, you may see an error such as: File /usr/bin/mariadb_config from install of MariaDB-devel-<version>.x86_64 conflicts with file from install of libmariadb-devel-3.1.21-150000.3.33.3.x86_64 (SLES Module Server Applications Updates).

3. Add the path where you installed the packages to the PATH environment variable as follows:

```
export PATH=/usr/local/bin:$PATH
```

4. Install the MySQL client as follows:

```
pip3.8 install mysqlclient
```

For Ubuntu

1. SSH into the Hue host as a root user.
2. Install the required packages and dependencies as follows:

```
apt-get install libmysqlclient-dev
apt-get install -y xmlsec1
apt-get install libxmlsec1-openssl
```

3. Add the path where you installed the packages to the PATH environment variable as follows:

   ```
   export PATH=/usr/local/bin:$PATH
   ```
4. Install the MySQL client as follows:

   ```
   pip3.8 install mysqlclient
   ```

Creating the Hue database

Hue requires a SQL database to store small amounts of data, including user account information as well as the history of queries and sharing permissions. You must create a database for Hue with UTF-8 character set and grant user privileges.

Procedure

1. SSH into the database server host as the root user.
2. Create a database for Hue with UTF-8 collation and grant user privileges:

   ```
   CREATE DATABASE [***HUE-DATABASE-NAME***] DEFAULT CHARACTER SET utf8 DEF
   AULT COLLATE utf8_general_ci;
   GRANT ALL ON [***HUE-DATABASE-NAME***].* TO '[***HUE-DATABASE-NAME***]'@'%
   ' IDENTIFIED BY '[***PASSWORD***]';
   ```

   Replace [***HUE-DATABASE-NAME***] with the actual Hue database name such as "hue".

   Ensure Hue uses UTF-8 collation and character set.

   ```
   # To create (use utf8_general_ci or utf8mb4_general_ci):
   CREATE DATABASE [***HUE-DATABASE-NAME***] 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-DATABASE-NAME***] COLLATE = 'utf8_general_ci';
   ```
3. Verify the connection to the Hue database by running the following command:

   ```
   mysql -u [***HUE-DATABASE-NAME***] -p
   Enter password: [***PASSWORD***]
   ```

Configuring MariaDB as the backend database for Hue

After installing and configuring the MariaDB database,you can configure Hue to use it as its backend database.

Procedure

1. Log into Cloudera Manager as an Administrator.
2. Go to Clusters Hue service Configuration and filter by Database category.
3. Select the Hue Database Type as MySQL.

4. Set the following database parameters:

   Hue Database Hostname: FQDN of host running MariaDB server

   Hue Database Port: 3306

   Hue Database Username: [***USERNAME***]

   Hue Database Password: [***PASSWORD***]

   Hue Database Name: [***HUE-DATABASE-NAME***] or SID
5. Click Save Changes.
6. Restart the Hue service.

Results

You can click Hue Web UI to log on to Hue with a custom MariaDB database.

Using PostgreSQL database with Hue

Hue is packaged with a lightweight embedded PostgreSQL database for proof-of-concept deployments with one Hue server. In production environments, you can install and use a custom PostgreSQL database as the backend database to store Hue metadata, job and query history, and account information.

Note: In 7.1.8, the Query Processor service requires the backend PostgreSQL database to not be SSL-enabled.

Download and install PostgreSQL

RHEL and CentOS 7 include PostgreSQL version 9.2. However, on these OS versions, Hue is compatible only with PostgreSQL versions 10, 11.x, 12, 13 and 14. Therefore, you must download and install the PostgreSQL distribution separately.

Before you begin

For supported PostgreSQL versions and compatibility information, see https://supportmatrix.cloudera.com/.

Procedure

1. Download the PostgreSQL distribution from https://www.postgresql.org/download/ based on your OS version.
2. Install the PostgreSQL server using the following command:

```
sudo yum install postgresql-server
```

Configure the PostgreSQL server

By default, PostgreSQL only accepts connections on the loopback interface. You must reconfigure PostgreSQL to accept connections from the fully qualified domain names (FQDN) of the hosts hosting the services for which you are configuring databases. If you do not make these changes, the services cannot connect and use the database on which they depend.

Procedure

1. Make sure that LC_ALL is set to en_US.UTF-8 and initialize the database as follows:

```
echo 'LC_ALL="en_US.UTF-8"' >> /etc/locale.conf
sudo su -l postgres -c "postgresql-setup initdb"
```

2. Enable MD5 authentication. Edit the pg_hba.conf file typically located at /var/lib/pgsql/data or /etc/postgresql/[***VERSION***]>/main. Add the following line:

   ```
   host all all 127.0.0.1/32 md5
   ```

3. If the default pg_hba.conf file contains the "host all   all 127.0.0.1/32 ident" line, then the host line specifying MD5 authentication shown above must be inserted before this line. Failure to do so may cause an authentication error when running the scm_prepare_database.sh script. You can modify the contents of the MD5 line shown above to support different configurations. For example, if you want to access PostgreSQL from a different host, replace 127.0.0.1 with your IP address and update the postgresql.conf file, which is typically found in the same place as pg_hba.conf, to include the following line:

   ```
   listen_addresses = '*'
   ```

4. Configure settings to ensure your system performs as expected. Update these settings in the /var/lib/pgsql/data/postgresql.conf or /var/lib/postgresql/data/postgresql.conf file based on cluster size and resources as follows:

   Small to mid-sized clusters: Consider the following settings as starting points. If resources are limited, consider reducing the buffer sizes and checkpoint segments further. Ongoing tuning may be required based on each host's resource utilization. For example, if the Cloudera Manager server is running on the same host as other roles, the following values may be acceptable:

- max_connection: In general, allow each database on a host a maximum of 100 connections and then add 50 extra connections. You may have to increase the system resources available to PostgreSQL.
- shared_buffers: 256 MB
- wal_buffers: 8 MB
- checkpoint_segments: 16. Replace configuration parameter checkpoint_segments with min_wal_size and max_wal_size.

  If you previously adjusted checkpoint_segments, the following formula will give you an approximately equivalent setting:

  ```
  max_wal_size = (3 * checkpoint_segments) * 16MB
  ```

  Note:

  The checkpoint_segments setting is removed in PostgreSQL 9.5 and higher. This setting is replaced by min_wal_size and max_wal_size.

  The default setting for max_wal_size is much higher than the default setting for checkpoint_segments. So, adjusting it might no longer be necessary.
- checkpoint_completion_target: 0.9

Large clusters: Can contain up to 1000 hosts. Consider the following settings as starting points:

- max_connection: for large clusters, allow each database on a host a maximum of 100 connections and then add 50 extra connections. You may have to increase the system resources that is available to PostgreSQL.
- shared_buffers: 1024 MB. This requires that the operating system can allocate sufficient shared memory.
- wal_buffers: 16 MB. This value is derived from the shared_buffers value. Setting wal_buffers to be approximately 3% of shared_buffers, up to a maximum of approximately 16 MB is sufficient in most cases.
- checkpoint_segments: 128. The PostgreSQL Tuning Guide recommends values between 32 and 256 for write-intensive systems, such as this one. Replace configuration parameter checkpoint_segments with min_wal_size and max_wal_size.

  If you previously adjusted checkpoint_segments, the following formula will give you an approximately equivalent setting:

  ```
  max_wal_size = (3 * checkpoint_segments) * 16MB
  ```

  Note:

  The checkpoint_segments setting is removed in PostgreSQL 9.5 and higher. This setting is replaced by min_wal_size and max_wal_size.

  The default setting for max_wal_size is much higher than the default setting for checkpoint_segments. So, adjusting it might no longer be necessary.
- checkpoint_completion_target: 0.9

5. Configure the PostgreSQL server to start when the system boots.

OS	Command
RHEL 7 Compatible, SLES, and Ubuntu	`sudo systemctl enable postgresql`

6. Restart the PostgreSQL database.

OS	Command
RHEL 7 Compatible, SLES, and Ubuntu	`sudo systemctl restart postgresql`

Configure PostgreSQL as the backend database for Hue

After installing and configuring the PostgreSQL database, you can configure Hue to use it as its backend database.

Procedure

1. Log in to Cloudera Manager as an Administrator.
2. Go to Clusters Hue service and stop the Hue service by clicking ACTIONS Stop .
3. Go to Clusters Hue service Configuration Category Database .

   All the database configurations for Hue are displayed.
4. Select PostgreSQL as the Hue Database Type.
5. Specify the fully qualified domain name (FQDN) of the host on which you have installed the PostgreSQL database in the Hue Database Hostname field.
6. Enter 5432 in the Hue Database Port field.
7. Enter user name and password in Hue Database Username and Hue Database Password fields.
8. Specify the database name in the Hue Database Name field.
9. Click Save Changes and restart the Hue service.

Results

If the database setup is successful and Hue starts without errors, then you should be able to log into the Hue web interface and view the database and tables.

Disabling the share option in Hue

Hue allows you to share documents, queries, and workflows with other users, either for viewing only or viewing and modifying. Sharing is enabled by default in the Hue UI. For added privacy and control, you can disable sharing by setting the enable_sharing property to false in the Advanced Configuration Snippet.

About this task

The sharing option is always available to the admin users. To disable the share option:

Procedure

1. Log in to Cloudera Manager as an Administrator.
2. Go to Clusters Hue service Configuration Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini and set the value of the enable_sharing property to false in the [desktop] section:

   ```
   [desktop]
   enable_sharing=false
   ```
3. Click Save Changes.
4. Restart the Hue service.

Enabling Hue applications with Cloudera Manager

Most Hue applications are configured by default, based on the services you have installed. Cloudera Manager selects the service instance that Hue depends on. If you have more than one service, you may want to verify or change the service dependency for Hue. If you add a service such as Oozie after you have set up Hue, you must set the dependency because it is not done automatically.

About this task

To add a dependency in Hue:

Procedure

1. In the Cloudera Manager Admin Console, select ClustersHueConfiguration to navigate to the configuration page for Hue.
2. Filter by ScopeHue (Service-Wide) and CategoryMain .
3. Select the <*service_name*> Service property that you want to set a dependency for. Select none to remove a dependency.
4. Enter a Reason for change..., and then click Save Changes at the bottom of the page to save the configuration change.
5. Refresh the browser page and click the restart icon ⏻ at the top of the page so the new configuration changes can be read by the server and the new setting takes effect.

Running shell commands

You can run shell commands to administer Hue programmatically. For example, to reset the superuser password or to assign an LDAP user superuser permissions.

About this task

To run Hue shell commands:

Procedure

1. 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_SERVER | tail -1 | awk '{print
    $9}'`"
   echo $HUE_CONF_DIR
   ```

2. Set environment variables used to run the Hue webserver:

   - CentOS/RHEL:

     ```
     for line in `strings /proc/$(lsof -i :8888|grep -m1 python|awk '{ print
      $2 }')/environ|egrep -v "^HOME=|^TERM=|^PWD="`;do export $line;done
     ```

   - Ubuntu:

     ```
     for line in `strings /proc/$(lsof -i :8888|grep -m1 hue|awk '{ print $2
      }')/environ|egrep -v "^HOME=|^TERM=|^PWD="`;do export $line;done
     ```

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.

For CDH parcel deployments:

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

  ```
  HUE_IGNORE_PASSWORD_SCRIPT_ERRORS=1 /opt/cloudera/parcels/CDH/lib/hue/
  build/env/bin/hue shell
  ```

  Or with the database password:

  ```
  HUE_IGNORE_PASSWORD_SCRIPT_ERRORS=1 HUE_DATABASE_PASSWORD=<your db passw
  ord> /opt/cloudera/parcels/CDH/lib/hue/build/env/bin/hue shell
  ```
- Change a user password

  ```
  HUE_IGNORE_PASSWORD_SCRIPT_ERRORS=1 /opt/cloudera/parcels/CDH/lib/hue/bu
  ild/env/bin/hue changepassword admin
  ```
- Promote Hue user to superuser

  ```
  HUE_IGNORE_PASSWORD_SCRIPT_ERRORS=1 /opt/cloudera/parcels/CDH/lib/hue/bu
  ild/env/bin/hue shell
  ```

  ```

  from django.contrib.auth.models import User
  a = User.objects.get(username='gwen')
  a.is_superuser = True
  a.save()
  ```
- Count all of the documents of a certain user:

  ```

  from django.contrib.auth.models import User
  from desktop.models import Document2

  user=User.objects.get(username='demo')
  Document2.objects.documents(user=user).count()

  Out[X]: 1167
  ```
- List available subcommands

  ```
  HUE_IGNORE_PASSWORD_SCRIPT_ERRORS=1 /opt/cloudera/parcels/CDH/lib/hue/bu
  ild/env/bin/hue
  ```

For CDH package deployments:

-
  ```
  HUE_IGNORE_PASSWORD_SCRIPT_ERRORS=1 /usr/lib/hue/build/env/bin/hue shell
  ```

Downloading and exporting data from Hue

Hue enables you to download or export data from Hue to HDFS or to an external storage location from Hue Editor, Hue Dashboard, and the Hue File browser. You can limit the number of rows or bytes that are downloaded or disable the export feature altogether so that you do not run out of storage space.

For a service-wide change, go to Cloudera Manager Clusters Hue service Configuration and specify the configurations in the Hue Service Advanced Configuration Snippet (Safety valve) for hue_safety_valve.ini field.

By default, Hue users can download the query results from the Hue Editor, the Hue Dashboard, and the File browser.

Limiting the number of rows to download

Specify the following in the Hue Service Advanced Configuration Snippet (Safety valve) for hue_safety_valve.ini to limit the number of rows that can be downloaded from a query before it is truncated:

```
[beeswax]
download_row_limit=x
```

x represents the number of rows that you can download.

By default, there is no download limit, and you can configure this by setting the value to “-1”:

```
[beeswax]
download_row_limit=-1
```

Limiting the number of bytes to download

Specify the following in the Hue Service Advanced Configuration Snippet (Safety valve) for hue_safety_valve.ini to limit the number of bytes that can be downloaded from a query before it is truncated:

```
[beeswax]
download_bytes_limit=x
```

x represents the number of bytes that you can download.

By default, there is no download limit, and you can configure this by setting the value to “-1”:

```
[beeswax]
download_bytes_limit=-1
```

Disabling the data download feature

Specify the following in the Hue Service Advanced Configuration Snippet (Safety valve) for hue_safety_valve.ini field to disable your users from downloading query results:

```
[desktop]
enable_download=false
```

Specify the following in the Hue Service Advanced Configuration Snippet (Safety valve) for hue_safety_valve.ini field to hide the Download button from the Hue File browser:

```
[filebrowser]
show_download_button=false
```

Backing up the Hue database

Cloudera Manager provides you (the Hue service administrator) an option to easily back up your database without the need to engage your database administrators or perform this operation manually.

About this task

You may have the need to back up the Hue database before upgrading or migrating to a new database, or for any other purposes. You can do so from Cloudera Manager as follows:

Procedure

1. Log in to Cloudera Manager as Administrator.
2. Go to Clusters Hue service .
3. Stop the Hue service by clicking Actions Stop .
4. After the Hue service has successfully stopped, click Actions Dump Databse .
5. On the **Dump Database** popup, click Dump Databse to confirm the operation.

   The database dump is written to the /tmp/hue_database_dump.json file on the Hue server host.
6. Start the Hue service by clicking Actions Start .

Enabling a multi-threaded environment for Hue

A multi-threaded environment can help reduce the time it takes to migrate data from one database to other. By default, operations such as migrating data run on a single thread. For example, if you are switching from MySQL as the backend database for Hue to Oracle, then enabling a multi-threaded environment significantly reduces the data transfer time.

Procedure

1. Log in to Cloudera Manager as an Administrator.
2. Go to Clusters Hue service Configuration Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini .
3. Locate the [[database]] section under [desktop] and set threaded to true:

   ```
   [desktop]
   [[database]]
   options={"threaded":true}
   ```
4. Click Save Changes.
5. Restart the Hue service.

Moving the Hue service to a different host

After adding new hosts to your CDP cluster if you need to move the Hue service from one host to another, then either add a new Hue service and manually configure it on the new host or add new role instances for Hue server, Hue Load Balancer, and Kerberos Ticket Renewer on new hosts.

Adding and configuring a new Hue service on a new host

To migrate the Hue service from one host to another within the same cluster, you can add a new Hue service on a new host using the Add Service wizard and manually configure the Hue service as needed.

Procedure

1. Sign in to Cloudera Manager as an Administrator.
2. Go to Clusters Hue service and stop the Hue service by clicking Actions Stop .
3. Go to Clusters [***CLUSTER-NAME***] and click Add Service from the menu.
4. On the **Add Service** wizard, select Hue.
5. On the **Select Dependencies** page, select the Optional Dependencies.

   Important: In CDH 6 and earlier, the Hive service included the Hive Metastore and HiveServer2. In Cloudera Runtime 7.0 and later, this service only includes the Hive Metastore. HiveServer2 and other components of the Hive execution engines are part of the HIVE_ON_TEZ service.

   If you need to run Hive queries, then select an optional dependency containing the HIVE_ON_TEZ service. If you need to run only Impala queries, then the HIVE_ON_TEZ service is not required.
6. On the **Assign Roles** page, click the field below Hue Server and Load Balancer roles. This launches a dialog box where you can select hosts where you want to add the role.

   Click OK after making your selection.
7. On the **Setup Database** page, specify the details of your existing Hue database.

   a) Select your database vendor from the Type field and specify the Database Hostname, Database Name, Username, and Password of your existing database.
   b) Click Test Connection, and when the success message appears, click Continue.

   The wizard configures and starts Hue and its dependent services.
8. After the new Hue service is in the running state, configure the Hue Advanced Configuration Snippet as it is in the existing Hue service on the older host.
9. Click Save Changes and restart the Hue service.
10. Verify that the new Hue service functions as expected.
11. Delete the Hue service from the old host by clicking Actions Delete .

Adding new role instances for Hue server, Hue Load Balancer, and Kerberos Ticket Renewer on new hosts

To migrate the Hue service from one host to another within the same cluster, you can add new hosts to your CDP cluster and then add the Hue Server, Hue Load Balancer, and Kerberos Ticket Renewer to the new host while inheriting the configurations from the existing roles.

Procedure

1. Sign in to Cloudera Manager as an Administrator.
2. Add new hosts for Hue server, Hue Load Balancer, and Kerberos Ticket Renewer to your cluster by going to Hosts Add Hosts .
3. Go to Clusters Hue service Instances and click Add Role Instances.
4. On the **Assign Roles** page, specify the role assignments for Kerberos Ticket Renewer, Load Balancer, and Hue Server by clicking Select hosts.
5. Select the new hosts that you have added to your CDP cluster for each of these roles.
6. Click Continue, review your changes, and exit the wizard.

7. Go back to the Instances tab and start the new role instances by selecting the newly added roles and click Actions for Selected Start .

   The newly configured roles inherit settings and configurations from the existing roles, including the configuration in the Advanced Configuration Snippet.
8. Verify that the new roles have the same configuration as the existing roles and the Hue service functions as expected.

   Important: If you have applied host-specific configuration, then you must manually apply the same configuration to your new host.
9. Delete the Hue roles from the old host by selecting them and clicking Actions for Selected Delete .
10. Restart the Hue service.

Adding Query Processor service to a cluster

The Query Processor service indexes Hive and Tez events and provides APIs to access them. It is required if you want to view the Queries tab (query history and query details) on the Hue Job Browser. You must install the Query Processor service on your CDP Private Cloud Base clusters manually.

About this task

Attention: This task is applicable only if you are upgrading to CDP 7.1.8 and higher. If you are upgrading to CDP 7.1.7 SP2, 7.1.7 SP1, 7.1.7, or lower, then you can skip this task.

You can either install the Query Processor service as a dependency while adding the Hue service or after adding the Hue service.

Before you begin

This task assumes that you already have a database installed on a host in your cluster. On CDP, Hue Query Processor supports PostgreSQL, MySQL, and Oracle databases. The supported PostgreSQL database version for Hue Query Processor is 9.6 and higher.

Next, you need to create a database for the Query Processor service with the required roles. To create the Query Processor database:

1. SSH into your database host as a root user.
2. Start the database terminal by running the following commands:

   (PostgreSQL)

   ```
   sudo -u postgres psql
   ```

   (MySQL)

   ```
   systemctl start mysqld
   ```

   (Oracle)

   ```
   sqlplus / as sysdba
   ```
3. Run the following statement while specifying the username, password, and a database name for the Query Processor:

   (PostgreSQL)

   ```
   CREATE ROLE [***QP-USER***] WITH LOGIN PASSWORD '[***QP-PASSWORD***]';
   ALTER ROLE [***QP-USER***] CREATEDB;
   ```

```
CREATE DATABASE [***QP-DATABASE***];
GRANT ALL PRIVILEGES ON DATABASE [***QP-DATABASE***] TO [***QP-USER***];
```

(MySQL)

```
CREATE USER [***QP-USER***] IDENTIFIED BY '[***QP-PASSWORD***]';
CREATE DATABASE [***QP-DATABASE***];
USE [***QP-DATABASE***];
GRANT ALL PRIVILEGES ON [***QP-DATABASE***].* TO [***QP-USER***];
```

(Oracle)

```
CREATE USER [***QP-USER***] IDENTIFIED BY [***QP-PASSWORD***];
GRANT ALL PRIVILEGES TO [***QP-USER***];
```

Note: When you create a user in the Oracle database, the associated schema is created automatically. This schema becomes the default schema for that user.

Procedure

1. Log in to Cloudera Manager as an Administrator.
2. Go to Clusters ⋮ Add Service .
3. Select Query Processor on the **Add Service to Cluster** page and click Continue.

   The required dependencies are automatically selected.
4. Select the host on which you want to install the Query Processor by clicking View By Host. Then click Continue.
5. Select the database type, and specify the database hostname, database name, and username and password to access the database on the **Setup Database** page and click Test Connection.

   Attention:

   - (For MySQL and Oracle databases) Ensure that you specify the same name in the Database Name and Database Schemas fields.
   - (For Oracle database) You must specify the value of the query_processor_database_schema property (Database Schema) in upper case.

   After the connection is verified, click Continue.
6. On the **Review Changes** page, accept the default settings and click Continue.

   The **Command Details** page is displayed, which shows the status of the installation and configuration.
7. Click Continue if the operation is successful.
8. Click Finish on the **Summary** page.

Results

You are redirected to the Cloudera Manager home page. You can now see the Query Processor service listed within your cluster.

What to do next

(Optional) To view the **Queries** tab on the **Job Browser** page in Hue, you must enable the Query Processor Service option in Hue configurations using Cloudera Manager.

Removing Query Processor service from cluster

You can remove the Query Processor service using Cloudera Manager when you no longer need it.

Before you begin

Because Hue has a dependency on the Query Processor service, deselect the Query Processor Service option in the Hue configurations before proceeding to remove service dependencies.

Procedure

1. Log in to Cloudera Manager as an Administrator.
2. Go to Clusters Query Processor service ⋮ Delete .

   If you have not removed the service dependencies before deleting the Query Processor, then a **Remove Service Dependencies** dialog box is displayed.

   a) Click Configure Service Dependency.
   b) Deselect the Query Processor Service option on the **Configuration** page and click Save Changes.
   c) Go to Clusters Query Processor service ⋮ Delete .

Enabling the Query Processor service in Hue

The Queries tab on Hue's Job Browser page allows you to view Hive query history and query details. You need to enable the Query Processor service from the Hue configurations to enable the Queries tab.

About this task

The Query Processor service is automatically enabled in Hue if you select it as a dependency on the **Select Dependencies** page while adding the Hue service to your cluster. But if you did not select it as a dependency while adding the Hue service, you can add the Query Processor service to your cluster (if not installed already) and then enable the Query Processor Service option in Hue configurations.

Procedure

1. Log in to Cloudera Manager as an Administrator.
2. Go to Clusters Hue service Configuration and select the Query Processor Service option.
3. Click Save Changes.
4. Restart the Hue service.

Results

You can now view Hive query history and query details under the **Queries** tab on Hue's **Job Browser** page.

Adding Query Processor Administrator users and groups on Private Cloud Base clusters

Query Processor Administrators have special privileges that enable them to view and monitor queries from all users, including the ones that are submitted from query interfaces, such as Beeline, Hive Warehouse Connector (HWC), Tableau, and so on.

About this task

This task is applicable only if you are running Hive queries.

Procedure

1. Log in to Cloudera Manager as an Administrator.

2. Go to Clusters Query Processor Configuration and enter the list of users in the Extra Auth Options field as follows:

   "adminUsers": "hive, [***USER-1***], [***USER-2***]"

   You can also add a list of admin groups as follows:

   "adminGroups": "admin-group, [***GROUP-1***], [***GROUP-2***]"
3. Click Save Changes.
4. Restart the Query Processor service.

Ways to clean up old queries from the Query Processor tables

Learn how to schedule a query clean-up and how to use the API to manually clean up queries from the following Query Processor tables: vertex_info, dag_details, dag_info, query_details, hive_query, tez_app_info.

Scheduling query clean-up

By default, the Hue Query Processor automatically clears queries from the Query Processor tables that are 30 days old at 2 AM every day as per the timezone set on your server. You can configure the following parameters in Cloudera Manager Query Processor service Configurations Query Processor Extra Configurations to set up a schedule based on your requirements:

- hue.query-processor.event-pipeline.cleanup.cron.expression
- hue.query-processor.event-pipeline.cleanup-interval-secs

For example:

```
“hue.query-processor.event-pipeline.cleanup.cron.expression” : “0 0 2 * * ?”
,
“hue.query-processor.event-pipeline.cleanup-interval-secs” : “2592000”
```

Manually cleaning up queries using an API

The ability to clean up queries manually in addition to the scheduled clean-up routines is useful when you have a high load of queries in a particular week that are hogging resources that you must free up. The API also runs a VACUUM command on the Query Processor table to reclaim storage that is occupied by dead tuples.

You can send an API request using tools such as cURL or Postman.

API format: `[***QUERY-PROCESSOR-ADDRESS***]/api/admin/cleanup/[***EPOCH-TIME***]`

Where,

- [***QUERY-PROCESSOR-ADDRESS***] is the query processor host address
- [***EPOCH-TIME***] is the Unix epoch time in seconds

Queries that were run before the specified epoch time are purged.

For example:

```
curl "http://machine1.company.com:30700/api/admin/cleanup/1670006742"
```

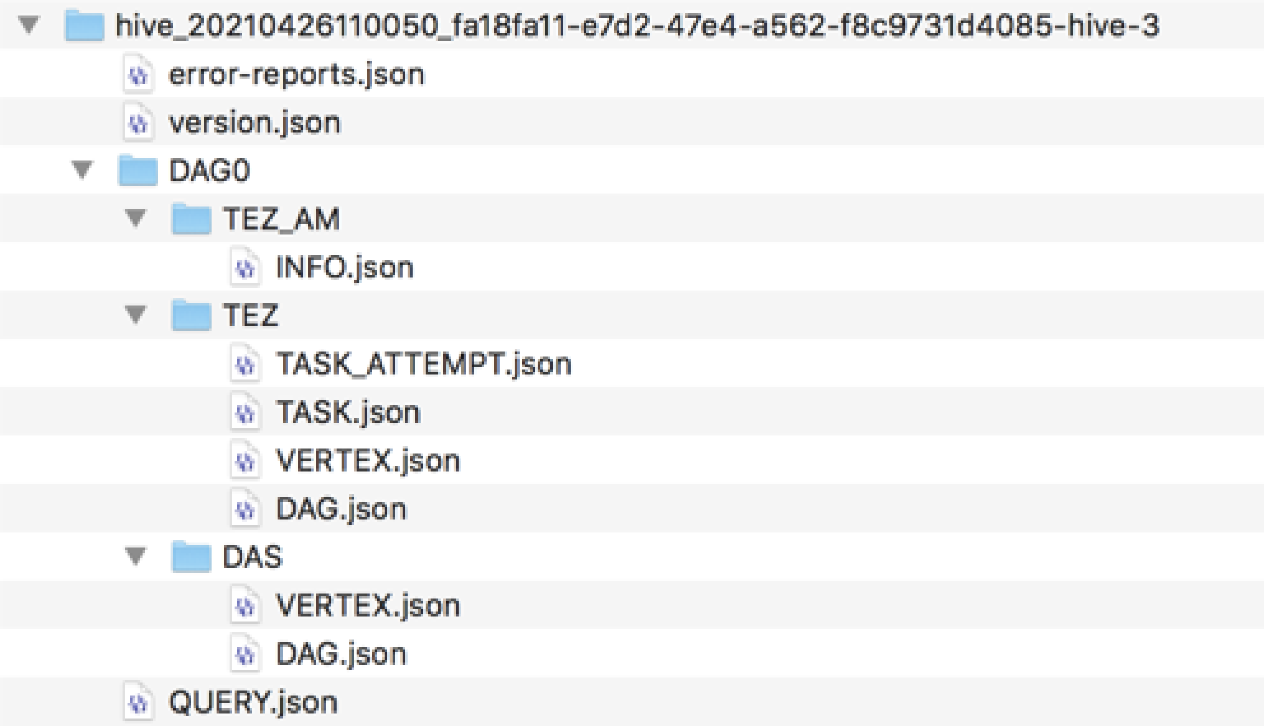
Downloading debug bundles

The debug bundle is a ZIP file that contains the query details in JSON format and an error-reports.json file, which is created only if an error occurs while the query is run.

About this task

Note: This feature is available only for Hive queries.

If Tez is used to run a query, then the debug bundle also contains DAG and Tez JSON files, as shown in the following image:



```
▼ hive_20210426110050_fa18fa11-e7d2-47e4-a562-f8c9731d4085-hive-3
    error-reports.json
    version.json
  ▼ DAG0
    ▼ TEZ_AM
        INFO.json
    ▼ TEZ
        TASK_ATTEMPT.json
        TASK.json
        VERTEX.json
        DAG.json
    ▼ DAS
        VERTEX.json
        DAG.json
    QUERY.json
```

Procedure

1. Log in to the Hue web interface.
2. Click Jobs from the left assist panel.

   The **Job Browser** page is displayed.
3. Click Queries.

   The Hive queries that were run are displayed.
4. Select a query for which you want to download the debug bundle.
5. Click Download and save the ZIP file on your computer.

   The filename is in the following format:

   ```
   hive_[***HIVE-QUERY-ID***]_[***USER-ID***]_[***UNIQUE-INDEX***]
   ```

Enabling DSL search for Hue

Domain-Specific Language (DSL) calls used for polling Atlas use the LIKE operator and hence they are inefficient, resource-intensive, and expensive. By default, Hue uses a basic search to look up data in Atlas. If needed, you can enable DSL for polling Atlas entities by setting the search_cluster property in the Advanced Configuration Snippet.

Before you begin

Obtain the Atlas metadata namespace from the Atlas application properties file. Search the file for the search_cluster property and note its value.

Procedure

1. Log in to Cloudera Manager as an Administrator.
2. Go to Clusters Hue Configuration .
3. Set the value of the search_cluster property in the Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini field:

   ```
   [metadata]
   [[catalog]]
   api_url=https://[***HOSTNAME***]:[***PORT***]
   interface=atlas
   search_cluster=cm ([***ATLAS.METADATA.NAMESPACE***])
   kerberos_enabled=true
   enable_basic_search=false
   ```

   Replace [***ATLAS.METADATA.NAMESPACE***] with the value of the search_cluster property you obtained from the Atlas application properties file.
4. Click Save Changes.
5. Restart the Hue service.

Configuring Hue to handle HS2 failover

Hue can handle HiveServer2 (HS2) failovers using ZooKeeper without the need to set up a load balancer.

Procedure

1. Log in to Cloudera Manager as an Administrator.
2. Go to Clusters Hue Configuration and add the following lines in the Hue Service Environment Advanced Configuration Snippet (Safety Valve) field:

   ```
   [beeswax]
   hive_discovery_hs2=true
   hive_discovery_hiveserver2_znode=/hiveserver2
   ```

   Note: hiveserver2 is the default value of the hive_discovery_hiveserver2_znode property. If this value is different from the Hive-On-Tez configuration, then you must obtain the information from the hive.server2 .zookeeper.namespace property from Hive-On-Tez configuration.
3. Click Save Changes.
4. Restart the Hue service.

Enabling Spark 3 engine in Hue

Hue leverages Apache Livy 3 to support Spark SQL queries in Hue on the Apache Spark 3 engine. To enable the Spark 3 engine, install the Spark 3 binaries on your cluster, add “Livy for Spark 3” and “Spark 3” services, and configure settings for these services using Cloudera Manager.

Before you begin

Note: Livy v0.6 supports Python versions upto Python 3.7. If you install Python 3.8 or higher, then you may see the following error: TypeError: required field "type_ignores" missing from Module.

Procedure

1. Log in to Cloudera Manager as an Administrator.
2. Add the following services by going to Clusters Actions and clicking Add Service:
    - Livy for Spark 3
    - Spark 3

    Important:

    - While adding the Livy for Spark 3 service to your cluster, you must select Hive as optional dependency on the **Select Dependencies** page.
    - While adding the Spark 3 service to your cluster, you must select an optional dependency containing Kafka on the **Select Dependencies** page and select the Spark Authentication option on the **Review Changes** page.
3. Go to Clusters HDFS Configuration and add the following lines in the Cluster-wide Advanced Configuration Snippet (Safety Valve) for core-site.xml field:

```
<property>
 <name>hadoop.proxyuser.hue.groups</name>
 <value>*</value>
<name>hadoop.proxyuser.hue.hosts</name>
 <value>*</value>
 <name>hadoop.proxyuser.spark.groups</name>
 <value>*</value>
 <name>hadoop.proxyuser.spark.hosts</name>
 <value>*</value>
 <name>hadoop.proxyuser.livy.groups</name>
 <value>*</value>
 <name>hadoop.proxyuser.livy.hosts</name>
 <value>*</value>
</property>
```

4. Click Save Changes.
5. Go to Clusters Livy for Spark 3 service Configuration and add the following configurations:
    a) Add the hue user in the Admin Users (livy.superusers) field.
    b) Go to the HMS Service field and select Hive.
    c) Click Save Changes.
6. Go to Clusters SPARK_ON_YARN Configuration Admin Users , add hue to the list of admin users (spark.history.ui.admin.acls) and click Save Changes.
7. Go to Clusters SPARK Configuration Admin Users , add hue to the list of admin users (spark.history.ui.admin.acls) and click Save Changes.
8. Go to Clusters SPARK 3 Configuration Admin Users , add hue to the list of admin users (spark.history.ui.admin.acls) and click Save Changes.

9. Go to Clusters Hue Configuration and enter the following lines in the Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini field and click Save Changes:

```
[desktop]
app_blacklist=zookeeper, pig #custom list of blocked apps
[spark]
#This is not a thrift server port
#If this TLS/SSL is enabled then check to see whether the livy url is on h
ttps or http and modify the url accordingly.
livy_server_url=http(s)://[***LIVY-FOR-SPARK3-SERVER-HOST***]:[***LIVY-
FOR-SPARK3-SERVER-PORT***]
ssl_cert_ca_verify=false
security_enabled=true
[notebook]
[[interpreters]]
[[[sparksql]]]
name=Spark SQL
interface=livy
```

Attention:

- Ensure that the Spark application is not on the blocked list.
- Set ssl_cert_ca_verify=false if an SSL certificate is not present in Hue's truststore. ssl_cert_ca_verify=true if your environment is a secured environment.
- Set security_enabled=true for Kerberized clusters.

10. Restart the affected services.

Results

You can now select the “Spark SQL” dialect on the Hue editor and run Spark queries from Hue.

Note: Starting a Livy session can take 30-45 seconds. If you cancel a running Spark SQL query and rerun it, the Livy session expires, and a new session is created.

Enabling the Phoenix SQL editor in Hue

To query Apache HBase tables from Hue using Phoenix SQL, configure the Phoenix interpreter settings in the Hue Advanced Configuration Snippet.

Before you begin

Add {USER} to the HBASE (cm_hbase) Ranger policy that governs access to all tables, columns, column-family, and so on. Grant {USER} the required permissions such as read, write, create, execute, admin, and so on.

Procedure

1. Log in to Cloudera Manager as an Administrator.
2. Go to Clusters Hue Configuration and add the following lines in the Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini field:

```
[notebook]
[[interpreters]]
[[[phoenix]]]
name=phoenix
interface=sqlalchemy
```

```
options='{"url": "phoenix://[***PHOENIX-URL***]:[***PHOENIX-PORT***]"}'
```

⚠ **Attention:** If you are using security, then specify the following in the options property:

```
options='{"url": "phoenix://[***PHOENIX-URL***]:[***PHOENIX-PORT***]
", "tls": true, "connect_args": "{\"authentication\": \"SPNEGO\", \"
verify\": false }", "has_impersonation": true}'
```

3. Click Save Changes.
4. Restart the Hue service.

Using Hue scripts

Hue scripts help you to share workflows, set up default editors, change ownership of documents, and test the backend processes using cURL commands. The Hue scripts are included with CDP.

Sharing workflows with users and groups

Use the following command to share workflows with users and groups:

```
./build/env/bin/hue share_all_workflows --owner [***OWNER-NAME***] --sharegr
oups [***GROUP1/USER1***],[***GROUP2/USER2***] --permissions read,write
```

To share users, use the --shareusers option instead of the --sharegroups option.

Changing the document owner

Use the following command to change the owner of the Hue documents:

```
./build/env/bin/hue change_owner_of_docs --olduser [***OLD-USER-NAME***] --n
ewuser [***NEW-USER-NAME***]
```

Testing the backend using cURL

Use the following command to get the syntax for the backend cURL commands or to test the endpoints:

```
./build/env/bin/hue get_backend_curl --showcurl --testname [***SERVICE-NAME-
WHERE-TEST-NEEDS-TO-BE-PERFORMED***] --service [***COMMA-SEPARATED-SERVICES-
TO-TEST]
```

Replace [***SERVICE-NAME***] with the name of the service where you want to perform the test. Replace [***COMMA-SEPARATED-SERVICES-TO-TEST***] with options or services such as all, httpfs, solr, oozie, rm, jhs, sparkhs, and so on.

Setting the default editor in Hue

Use the following command to set the default editor in Hue:

```
./build/env/bin/hue set_default_editor --hive/impala --username [***USERNAME
***]
```

The default value for the username is all.

Configurations for submitting a Hive query to a dedicated queue

When you run a Hive query from Hue or Beeline, HiveServer2 submits the query in YARN in the queue mapped from an end user instead of a hive user. If you run a query while specifying doAs=false, then the queries are submitted and run as a hive user, using impersonation. This is the default behaviour. You can change this behaviour by configuring the queue using "yarn_queue_manager" and "application tag" parameters in YARN.

If you see go to the **Job Browser** page in Hue and see that the Hive queries are appearing under the default queue, and not the actual user who ran the query, then this means that the queries are not being run in their dedicated queue in YARN. To submit a Hive query in its dedicated queue, you must enable application-tag based scheduling by configuring queue mapping and per queue properties.

Related Information

Configuring queue mapping to use the user name from the application tag using Cloudera Manager

Configure Per Queue Properties

Enabling browsing Ozone from Hue on CDP Private Cloud Base

Hue can read and write files on the Ozone filesystem. To access Ozone from Hue, you must add the necessary configurations in the Advanced Configuration Snippet in Cloudera Manager.

About this task

You must perform this task on all Hue instances in which you want to enable the Ozone File Browser.

Before you begin

- You must have an Ozone parcel containing the HttpFS service. For installation, see Ozone 718.2.x installation documentation.
- Ensure that the Ozone HttpFS Gateway role is running in a healthy state.
- Go to Cloudera Manager Ozone Configuration and add the following entries in the HttpFS Gateway Advanced Configuration Snippet (Safety Valve) for ozone-conf/httpfs-site.xml field:

Field name	Value
Name	httpfs.proxyuser.[***PRINCIPAL-NAME***].hosts
Value	*
Name	httpfs.proxyuser.[***PRINCIPAL-NAME***].groups
Value	*

Replace the [***PRINCIPAL-NAME***] with the actual Kerberos principal name. For example, hive. You can obtain the Kerberos principal name from the Hive service configuration.

- Obtain the following values from the Ozone service. You need it to construct the fs_defaultfs and webhdfs_url URLs:
  - HttpFS Gateway host name
  - Ozone HttpFS Gateway TTP Web UI Port (ozone.httpfs.http-port)

    9778 is the default.
  - Ozone Service ID (ozone.service.id). The Ozone service ID must be specified when Ozone is configured in HA mode.

Procedure

1. Log in to Cloudera Manager as an Administrator.
2. Go to Clusters Hue Configuration and add the following lines in the Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini field:

   ```
   [desktop]
   [[ozone]]
   [[[default]]]
   fs_defaultfs=ofs://[***SERVICE-ID***] \\ozone1 by default
   webhdfs_url=http://[***OZONE-HTTPFS-HOST***]:[***OZONE-HTTPFS-PORT***]/w
   ebhdfs/v1 \\https for secure cluster
   ssl_cert_ca_verify=true
   security_enabled=true
   ```

   Specify https in the webhdfs_url property if SSL is enabled on your cluster.

   If Ozone is not configured in HA mode, then specify the URL for the Ozone manager in the fs_defaultfs property.
3. Click Save Changes.
4. Restart the Hue service.

Limitations in browsing Ozone from Hue

While Hue fully supports browsing the Ozone filesystem, there are some known limitations. Review these limitations and learn about the workarounds.

File-size limit for uploads

The maximum supported file size that you can upload using the Hue File Browser to the Ozone filesystem is 2 GB. The default file size is 64MB. You can change the permissible upload limit by setting the value of the upload_chunk_size parameter as follows in the Hue Advanced Configuration Snippet (Hue safety valve):

```
[hadoop]
upload_chunk_size=2000000000
```

File-size limitations for copy operations

There is a limit on the size of files you can copy from the source path to the destination using the Hue File Browser present on the Ozone filesystem. Hue automatically skips copying the files larger than the chunk size value (default is 64MB) and copies rest of the files recursively. You can change the upload chunk size by setting the value of the upload_chunk_size parameter in bytes as follows in the Hue Advanced Configuration Snippet (Hue safety valve):

```
[hadoop]
upload_chunk_size=[***SIZE-IN-BYTES***]
```

The following sample configuration sets the upload chunk size to 120 MB:

```
[hadoop]
upload_chunk_size=125829120 \\or 120*1024*1024
```

User home directory

All users default to the ofs:// path when they access the Ozone filesystem from Hue.

Volume and bucket limitations

You can only create default volume and buckets in Ozone using Hue. Operations other than create are not supported.

You cannot create Erasure Coding (EC)-enabled buckets from Hue, unless they are set by default in Ozone. However, you can create the EC-enabled buckets using the Ozone CLI and then browse it normally from Hue. If EC-enabled buckets are already present on Ozone, then you can browse them from Hue.

Hue can also display the content of the encrypted buckets present on Ozone.

Configuring timezone for Hue

You can configure the timezone for Hue using Cloudera Manager. The default value is America/Los_Angeles.

Procedure

1. Log in to Clouder Manager as an Administrator.
2. Go to Clusters Hue Configuration Time Zone and set a desired value.

   The default value is “America/Los_Angeles”. You can refer to https://timezonedb.com/time-zones for information on the timezones.
3. Click Save Changes.
4. Restart the Hue service.