Cloudera Runtime 7.1.1

Hue Troubleshooting

Date published: 2020-07-28
Date modified: 2020-08-10

https://docs.cloudera.com/

Legal Notice

© Cloudera Inc. 2025. 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

Unable to authenticate users in Hue using SAML..4

Unable to terminate Hive queries from the Hue Job Browser in a Kerberized cluster..4

Unable to view or create Oozie workflows in Hue on a Knox-secured cluster.. 5

MySQL: 1040, 'Too many connections' exception... 5

Unable to connect Oracle database to Hue using SCAN......................................6

Increasing the maximum number of processes for Oracle database...................7

UnicodeDecodeError: UTF-8 codec cannot decode byte in position................... 8

UnicodeDecodeError: 'ascii' codec cannot decode byte in position.....................8

Fixing authentication issues between HBase and Hue...9

Hue Load Balancer does not start due to lengthy BalancerMember Route length.. 10

Cannot alter compressed tables in Hue.. 11

Unable to authenticate users in Hue using SAML

If you have configured SAML to authenticate users, but your users are unable to log into Hue using Single Sign On (SSO), then it is possible that the RSA key format is not supported. To resolve this issue, you can use an unprotected private key and then specify the private key filename in the safety valve.

Procedure

1. Convert the .key file to an unprotected private key file by using the following command:

```
openssl rsa -in /opt/cloudera/security/<FILE NAME>.key -out /opt/cloudera/
security/<FILE NAME_UNPROTECTED>.key
```

```
openssl rsa -in /opt/cloudera/security/HADOOP-CPI-PROD.key -out /opt/cloud
era/security/HADOOP-CPI-PROD_UNPROTECTED.key
```

2. Update the advanced configuration snippet as shown in the following example:

```
[libsaml]
xmlsec_binary=/usr/bin/xmlsec1
metadata_file=/opt/cloudera/security/saml/idp-openam-metadata.xml
key_file=/opt/cloudera/security/HADOOP-CPI-PROD_UNPROTECTED.key
cert_file=/opt/cloudera/security/HADOOP-CPI-PROD.pem
```

Unable to terminate Hive queries from the Hue Job Browser in a Kerberized cluster

On a Kerberized cluster, if YARN does not have Kerberos authentication enabled for HTTP web consoles, then you may not be able to terminate Hive queries from the Hue Job Browser, and you may see the following error in the Hue role log runcpserver.log file: "The default static user cannot carry out this operation. (error 403)".

About this task

On a Kerberized cluster, YARN must have Kerberos authentication enabled for HTTP web consoles. If authentication is not enabled, then the user or application that is trying to access YARN using a REST API is identified as the default "dr.who" user. The default user does not have permissions to access the YARN UI and terminate the running jobs. As an immediate solution, you can terminate the job from the Hue query editor or from the YARN CLI using the following command:

```
yarn application -kill [***APPLICATION-ID***]
```

To enable terminating jobs and running queries from the Hue Job Browser, enable Kerberos authentication for HTTP web consoles for YARN as follows:

Procedure

1. Log in to Cloudera Manager as an Administrator.
2. Go to Clusters YARN Configuration and type enable kerberos in the search box.
3. Select Enable Kerberos Authentication for HTTP Web-Consoles.
4. Click Save Changes.
5. Restart the YARN service.

Unable to view or create Oozie workflows in Hue on a Knox-secured cluster

If you are unable to view Oozie workflow actions, such as "HiveServer2 Script" or "Shell Script" on the **Oozie Editor** page in Hue on a Knox-secured cluster, then check the Hue logs for the following warning: "POST /desktop/log_js_error HTTP/1.1" --- Referer checking failed - https://[***FQDN***]:[***PORT***]/oozie//editor/workflow/new does not match any trusted origins.. To fix this issue, add the Oozie server URL to the trusted_origins property in the Hue Advanced Configuration Snippet.

About this task

When you set up Knox on your CDP cluster, Knox authenticates access or requests from other services and applications. If you specify the Oozie server URL in the trusted_origins property, Knox can check that the incoming request is from a trusted source (Oozie) and approves access, allowing you to view and create Oozie workflows from Hue.

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. Add the Oozie server URL in the trusted_origins property under the desktop section as follows:

   ```
   [desktop]
   [[session]]
   # Comma-separated list of Oozie nodes and Oozie ports
   # for each Oozie instance
   trusted_origins=[***OOZIE-NODE1***]:[***OOZIE-PORT1***], [***OOZIE-NODE2**
   *]:[***OOZIE-PORT2***],...
   ```

   For example:

   ```
   [desktop]
   [[session]]
   # Comma-separated list of Oozie nodes and Oozie ports
   # for each Oozie instance
   trusted_origins=localhost:11000
   ```
4. Click Save Changes.
5. Restart the Hue service.

MySQL: 1040, 'Too many connections' exception

If Hue displays the "1040, Too many connections" exception, then it is possible that the Hue backend database is overloaded and out of maximum available connections. To resolve this issue, you can increase the value of the max_connections property for your database.

About this task

The 1040, 'Too many connections' exception occurs on a MySQL database when it runs out of maximum available connections. If you are using the Impala engine, you may see the following error message on the Hue web interface: OperationalError at /desktop/api2/context/computes/impala("1040: too many connections"). A similar error may be displayed for Hive. The exception is also captured in the Hue server logs.

Before you begin

The max_connections property defines the maximum number of connections that a MySQL instance can accept. Uncontrolled number of connections can crash the server. Following are some guidelines for tuning the value of the max_connections property:

- Set the value of the max_connections property according to the size of your cluster.
- If you have less than 50 hosts, then you can store more than one database (for example, both the Activity Monitor and Service Monitor) on the same host. If you have more than 50 hosts, then use a separate host for each database/host pair. The hosts need not be reserved exclusively for databases, but each database must be on a separate host.
- For less than 50 hosts:
  - Place each database on its own storage volume.
  - Allow 100 maximum connections for each database and then add 50 extra connections. For example, for two databases, set the maximum connections to 250. If you store five databases on one host (the databases for Cloudera Manager Server, Activity Monitor, Reports Manager, Atlas, and Hive MetaStore), then set the maximum connections to 550.

To increase the number of maximium available connections and to resolve the "1040, Too many connections" exception:

Procedure

1. Log in to Cloudera Manager and stop the Hue service.
2. SSH in to your database instance as a root user.
3. Check the number of available connections by running the following command:

   ```
   grep max_conn /etc/my.cnf
   ```

   /etc/my.cnf is the default location of the options file (my.cnf).
4. Set the new value of the max_connections property from the MySQL shell as per the guidelines provided above. For example:

   ```
   mysql> SET GLOBAL max_connections = 550;
   ```
5. Restart the Hue service.

Unable to connect Oracle database to Hue using SCAN

For high availability purposes, you may want Hue to stay connected to any Oracle database instances running in your cluster. Single Client Access Name (SCAN) serves as a cluster alias for databases in the cluster. Currently, Cloudera Manager does not provide an option to use SCAN to connect to the Oracle database. To use SCAN, you must temporarily install a MySQL database to create a Hue service and then specify Oracle settings in the Hue advanced configuration snippet.

About this task

It is possible that other clusters or services may be able to connect to the database using the Oracle SQL Developer. But you may see the following error when you try to add the Hue service using the Cloudera Manager Add Service wizard and specifying SCAN on the **Setup Database** page: “Able to find the Database server, but not the specified database. Please check if the database name is correct and make sure that the user can access the database.”

Procedure

1. Sign in to Cloudera Manager as an administrator.
2. Add the Hue service using a MySQL database.

3. Check whether the Hue service is added successfully by launching the web UI.
4. After the Hue service is running, go to Cloudera Manager Clusters Hue service Configuration .
5. Add the following lines in the Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini field:

```
[desktop]
[[database]]
port=0
engine=oracle
name=[***ORACLE-SCAN***]/[***SERVICE-NAME***]
user=[***HUE-DB-USER***]
```

Note: Specify port=0 because the port used for the Oracle database (1521) is part of the SCAN.

6. Enter the database password in the Hue Database Password field.
7. Click Save Changes.
8. Restart the Hue service.
9. Check whether you can access your databases from the Hue web UI.
10. Uninstall the MySQL database if no longer needed.

Increasing the maximum number of processes for Oracle database

While using Oracle as a backend database for Hue, if you face issues connecting to the Hue service after restarting the database, then it is possible that the Hue is not able to get a new database connection. The following error in the Hue logs indicates that the maximum number of connections have exhausted: "ORA-12519: TNS:no appropriate service handler found". This can be resolved by increasing the number of available processes.

After restarting the Oracle database, if you are not able to connect to the Hue service, check the Hue logs for the "ORA-12519: TNS:no appropriate service handler found" error. The Hue logs are present in the following directory:

opt/cloudera/parcels/CDH-[***VERSION***]/lib/hue/build/env/lib/python2.7/site-packages/Django-1.11.20-py2.7.egg/django/db/backends/oracle/base.py

If you see the above error in the logs, then work with your database administrator to check whether the maximum number of processes have exceeded. If the maximum number of processes have exceeded, then you see the following error: "ORA-00020: maximum number of processes exceeded". Increase the number of processes to resolve this issue.

How to calculate the number of database processes, transactions, and sessions?

Cloudera recommends that you allow 100 maximum connections for each service that requires a database and then add 50 extra connections. For example, for two services, set the maximum connections to 250. If you have five services that require a database on one host (the databases for Cloudera Manager Server, Activity Monitor, Reports Manager, Cloudera Navigator, and Hive metastore), set the maximum connections to 550.

From the maximum number of connections, you can determine the number of anticipated sessions using the following formula:

```
sessions = (1.1 * maximum_connections) + 5
```

For example, if a host has a database for two services, anticipate 250 maximum connections. If you anticipate a maximum of 250 connections, plan for 280 sessions.

Based on the number of sessions, you can determine the number of anticipated transactions using the following formula:

```
transactions = 1.1 * sessions
```

Continuing with the previous example, if you anticipate 280 sessions, you can plan for 308 transactions.

Work with your Oracle database administrator to apply these derived values to your system.

Using the sample values above, Oracle attributes would be set as follows:

```
alter system set processes=250;
alter system set transactions=308;
alter system set sessions=280;
```

UnicodeDecodeError: UTF-8 codec cannot decode byte in position

You may see an error such as the following while querying a table using the Impala editor in Hue: "'utf8' codec can't d ecode byte 0x91 in position 6: invalid start byte". This occurs because Hue does cannot handle columns containing non-UTF-8 data.

To resolve this issue, contact Cloudera Support to apply the following software patches: CDPD-22129 and CDPD-27412.

UnicodeDecodeError: 'ascii' codec cannot decode byte in position

You may see an error such as the following while downloading Impala query results in CSV format from Hue: "Unic odeDecodeError: 'ascii' codec can't decode byte 0xc2 in position 49: ordinal not in range(128)". This occurs when the query results contain special characters of non-ASCII encoding.

About this task

Note: You may not face this issue if you export the query result from Hue to Excel or if you are querying the table using the Hive editor.

In CDP, Hue is compatible with Python 2.7.x, but the Tablib library for Hue has been upgraded from 0.10.x to 0.14.x, which is generally used with the Python 3 release. If you try to download Impala query results having special characters in the result set in a CSV format, then the download may fail with the ASCII unicode decode error. To fix this issue, downgrade the Tablib library to 0.12.x as follows:

Procedure

1. SSH into the Hue server host.
2. Change directory to the following:

   ```
   cd /opt/cloudera/parcels/CDH-7.x/lib/
   ```

3. Back up the hue directory.

   ```
   cp -R hue hue_orginal
   ```

4. Change to the hue directory.

   ```
   cd hue
   ```
5. Install the Wheel package using pip.

   ```
   ./build/env/bin/pip install wheel
   ```

   The Wheel package is used to avoid recompiling your software during every install.
6. Install the Python Setuptools package for Hue:

   ```
   ./build/env/bin/pip install setuptools==44.1.0
   ```
7. Install Tablib version 0.12.1:

   ```
   ./build/env/bin/pip install tablib==0.12.1
   ```
8. Update the permissions of the Tablib library and other packages that were installed during the process by using the following command:

   ```
   chmod 0755 [***TABLIB-LIBRARY-NAME***]
   ```

   ⚠ **Attention:** The Tablib library and other packages should have the drwxr-sr-x permission. Failing to have this permission can result into an “tablib module not found” error.
9. Go to Cloudera Manager and restart the Hue service.

Fixing authentication issues between HBase and Hue

An HBase feature improvement to the Thrift Server (HBASE-19852) may cause authentication issues between HBase and Hue, and you may see the following error while accessing the HBase tables from Hue: “Failed to authenticate to HBase Thrift Server, check authentication configurations.”

About this task

You may also see the following error in the Hue logs: “RestException: Unable to authenticate <Response [401]>”. To fix this issue, update the HBase configurations using Cloudera Manager.

Procedure

1. Log in to Cloudera Manager as an Administrator.
2. Go to Clusters HBase service Instances and note the hostname of the host on which the HBase Thrift Server is running.

   If multiple Thrift Servers are configured, then find the one that Hue is configured to use.
3. Go to Configuration HBase Thrift Server Advanced Configuration Snippet (Safety Valve) for hbase-site.xml and add the following properties:

Field	Property
Name	hbase.thrift.spnego.principal
Value	HTTP/[***HOSTNAME-FROM-STEP2***]@REALM Substitute @REALM with the actual Kerberos realm.
Name	hbase.thrift.spnego.keytab.file
Value	hbase.keytab

4. Select the following options to enable the properties:

    - Enable HBase Thrift Http Server
    - Enable HBase Thrift Proxy Users

5. Deselect the following properties:

    - Enable HBase Thrift Server Compact Protocol
    - Enable HBase Thrift Server Framed Transport

6. If you have not enabled SSL on your cluster, but if you have Kerberized your cluster, then add the following HBase configurations, without which you may encounter a NullPointerException error while starting the HBase Thrift Server:

    a) Add the following properties in the HBase Client Advanced Configuration Snippet (Safety Valve) for hbase-site.xml field:

Field	Property
Name	hbase.thrift.ssl.enabled
Value	false

    b) Add the following properties in the HBase Thrift Server Advanced Configuration Snippet (Safety Valve) for hbase-site.xml field:

Field	Property
Name	hbase.thrift.ssl.enabled
Value	false

7. Click Save Changes.
8. Go to Clusters Hue service Configuration Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini and add the following lines:

    ```
    [hbase]
    thrift_transport=buffered
    ```

9. Click Save Changes.
10. Restart the HBase and Hue services to apply stale configurations.

Hue Load Balancer does not start due to lengthy BalancerMember Route length

The Hue Load Balancer may not start if the route name in the ROLES table exceeds 64 characters. You must manually reduce the length of the route name for each Hue server in the ROLES table to be less than or equal to 64 characters, and also ensure that it is unique.

About this task

Cloudera Manager creates the Role name in the following format, which may sometimes cause the route name to exceed 64 characters: "Service-name-Role-name-Cluster-name-Unique-hash". You may see the following error while starting the Hue Load Balancer:

```
BalancerMember Route length must be < 64 characters
+ '[' 1 '!=' 0 ']'
+ die '/var/run/cloudera-scm-agent/process/482-hue-HUE_LOAD_BALANCER/http
d.conf is invalid.'
+ echo '/var/run/cloudera-scm-agent/process/482-hue-HUE_LOAD_BALANCER/httpd.
conf is invalid.'
```

```
/var/run/cloudera-scm-agent/process/482-hue-HUE_LOAD_BALANCER/httpd.conf is
 invalid.
```

To resolve this issue, modify the route name column in the ROLES table in the scm database for every Hue server to be less than or equal to 64 characters, and ensure that it is unique.

Procedure

1. SSH into the database instance as an administrator.
2. Query the ROLES table to view table content:

   ```
   SELECT * FROM `ROLES`;
   ```

3. Update the values in the "NAME" column so that the route name is less than or equal to 64 characters:

   ```
   UPDATE ROLES SET NAME='[***HUE-ROLE-NAME***]' WHERE ROLE_ID=[***ROLE-ID-
   NUMBER***];
   ```

   ```
   UPDATE ROLES SET NAME='hue-6c02f47dbd7e181d293c078ea293f3da' WHERE ROLE_
   ID=10;
   ```

Cannot alter compressed tables in Hue

Due to a known bug in the Oracle database (12c and higher), you cannot perform ALTER TABLE operations (add, delete, drop, modify) on compressed tables. 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.

About this task

Even if you uncompress an existing table, you may not be allowed to alter the columns. To resolve this issue:

Procedure

1. SSH into the host on which you have installed the Oracle database.
2. Create a new uncompressed table with the same structure as the compressed table.
3. Copy the data from the compressed table to the new uncompressed table.
4. Rename or delete the compressed table.
5. Rename the uncompressed table with the name of the original compressed table.

Results

You should now be able to perform ALTER TABLE operations (add, delete, drop, modify) on the Hue tables.