Cloudera Runtime 7.3.1

Hue Troubleshooting

Date published: 2020-07-28
Date modified: 2024-12-10

https://docs.cloudera.com/

Legal Notice

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

The Hue load balancer not distributing users evenly across various Hue servers

The Hue load balancer redirects the new users to the newly added Hue servers and the existing users to the existing Hue servers on your cluster. Even though you add more Hue servers to meet the growing user base, the resources might not be utilized effectively.

About this task

The Hue load balancer is tasked to evenly distribute users across the available Hue servers for effective resource utilization. However, due to session persistence, it does not distribute users evenly. To overcome this issue, you can refresh the cookies from Cloudera Manager.

The load balancer uses the cookie ROUTEID with a random string from your browser and is stored in the hue.conf file. This random string is used to redirect a user to the Hue server. To refresh the cookie and set a new random string every time you add a new Hue server, do the following:

Procedure

1. Go to Cloudera Manager Clusters Hue Configuration .
2. Click Scope Load balancer and select the Hue Load Balancer Cookie Refresh checkbox.

   This refreshes the cookie value in the hue.conf file to rebalance the Hue backend connections.
3. In the Instances tab, select all the Hue services and roles, and click Action for Services Restart .

   This creates a new random string for the cookie which the load balancer can now use to evenly distribute users.

Results

Whenever you restart the Hue server, the load balancer will redistribute users evenly based on the server load.

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
   ```

Cleaning up old data to improve performance

Some tables in Hue retain data indefinitely resulting in slower performance or application crashes. Hue does not automatically clean up data from these tables. You can configure Hue to retain the data for a specific number of days and then schedule a cron job to clean up these tables at regular intervals for improved performance.

About this task

Consider cleaning up old data from the backend Hue database if you face the following problems while using Hue:

- Upgrade times out
- Performance is slower than expected
- Long time to log in to Hue
- SQL query shows a large number of documents in tables
- Hue crashes while trying to access saved documents

⚠ **Important:** The clean-up steps only deletes the unsaved documents and workflows. Saved data and information is not cleaned up.

Before you begin

Back up your database before starting the cleanup activity. Check the saved documents such as Queries and Workflows for a few users to prevent data loss. You can also note the sizes of the tables you want to clean up as a reference by running the following queries:

```
select count(*) from desktop_document;
select count(*) from desktop_document2;
select count(*) from beeswax_session;
select count(*) from beeswax_savedquery;
select count(*) from beeswax_queryhistory;
select count(*) from oozie_job;
```

Note: The optimal number of documents that can be stored in a table is less than or equal to 30,000. Consider this number while specifying the cleanup interval.

Procedure

1. SSH into an active Hue instance.
2. Change to the Hue home directory:

   ```
   cd /opt/cloudera/parcels/CDH/lib/hue
   ```

3. Run the following command as the root user:

   ```
   ./build/env/bin/hue desktop_document_cleanup --keep-days X --cm-managed
   ```

   The --keep-days property is used to specify the number of days for which Hue will retain the data in the backend database.

   (Optional) Specify DESKTOP_DEBUG=True if you want to log information for troubleshooting purposes.

   ```
   DESKTOP_DEBUG=True ./build/env/bin/hue desktop_document_cleanup --keep-d
   ays 30 --cm-managed
   ```

   In this case, Hue will retain data for 30 days.

   The logs are displayed on the console because DESKTOP_DEBUG is set to True. Alternatively, you can view the logs from the following location:

   /var/log/hue/desktop_document_cleanup.log

   The first run can typically take around 1 minute per 1000 entries in each table.
4. Check whether the table size has decreased by running a query as follows:

   ```
   select count(*) from desktop_document;
   ```

   If the desktop_document_cleanup command has run successfully, the table size should decrease.

What to do next

Set up a cron job that runs at regular intervals to automate the database cleanup. For example, you can set up a cron job to run daily and it purges data older than *X* number of days.

Unable to connect to database with provided credential

Cloudera Manager tests the database connection when you add the Hue service to a cluster. The “Test Database Connection” does not work for Oracle databases that require service name instead of the Oracle System ID (SID). This could stop you from adding the Hue service to your cluster.

About this task

If you encounter the following error while adding the Hue service through Cloudera Manager, then follow the workaround as mentioned in this topic:

```
Unable to connect to database with provided      credential. Able to find the Database server, but not the specified dat
abase.      Please check if the database name is correct and make sure that the user can      access the database.
```

Procedure

1. Install a different database instance, such as MySql temporarily to use with Hue.

   This is referred to as the Hue database.
2. Add the Hue service from Cloudera Manager and specify the Hue database details that you created in the previous step.

   This will allow you to get past the Add Service wizard and add the Hue service to your cluster.

3. Modify the Hue instance to use the actual Oracle database as follows:
    a) Navigate to Cloudera Manager Clusters Hue service Configuration Category Database .

       The database configuration fields are displayed.
    b) Set up the Oracle database by configuring the following fields:
        1. Select Oracle as the Hue Database Type.
        2. In the Hue Database Hostname field, specify the Fully Qualified Domain Name (FQDN) of the host on which you have installed the Oracle database.
        3. In the Hue Database Port field, specify the port on the host on which the Oracle databse is running. Typically, this value is 1521.
        4. In the Hue Database Username field, specify the username to log in to the Oracle database.
        5. In the Hue Database Password field, specify the database password.
        6. In the Hue Database Name field, specify the name of the Hue database in the following format:

           ```
           <HUE_DB_HOST>:1521/<SERVICENAME>
           ```
    c) Navigate to Cloudera Manager Clusters Hue service Configuration Category Advanced and specify the following in the Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini field:

       ```
       [desktop]
       [[database]]
       port=0
       ```
4. Click Save Changes.
5. Restart the Hue service by clicking Actions Restart .

Activating Hive query editor on Hue UI

You may not see the Hive query editor on the Hue user interface if you have not installed and selected the HIVE_ON_TEZ service on your cluster. The HIVE_ON_TEZ service is needed to configure and use Hive with Hue.

Note:

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

About this task

To enable the Hive query editor on the Hue web UI:

Procedure

1. Sign in to Cloudera Manager as an Administrator.
2. Check whether the HIVE_ON_TEZ service is installed on your cluster.

   If it is not installed already, add it as a service.
3. Go to Clusters Hue service Configuration .

   The list of Hue configurations is displyed.
4. Search for the HiveServer2 Service field and select the HIVE_ON_TEZ service.
5. Save the changes and restart Hue.

Results

The Hive query editor should now be available on the Hue user interface.

Query execution finished in Hue but shows as executing on Cloudera Manager Impala Queries Page

Cloudera Manager and the Impala demon web page may show a query in an “executing” or “In Flight” state even though the query has finished executing on the Hue web UI. This can happen due to various reasons.

The three main reasons why the completed Hue query still shows as "executing" are:

- Hue does not close the connection to Impala until you click on the **Results** page.

  Clicking the **Results** page in Hue executes the fetchresults call to Impala.
- Impala queries are client-driven. Therefore, the query still remains in a running state until the client sends a fetch command to complete fetching the entire result set.
- If a query has not been closed or unregistered, Impala shows the same in the **In Flight** section on its web UI. Cloudera Manager shows all In Flight queries in the “Executing” state.

Impala query life cycle

When you submit Impala queries, they are first registered by the system. The system identifies the queries with the help of a coordinator. They also have a state, such as CREATED, INITIALIZED, RUNNING, FINISHED, EXCEPTION, and some metadata.

- FINISHED implies that the rows are available but not all rows are ready to be fetched. It is possible that Impala daemons are still executing the query.
- EXCEPTION implies that an error has occurred. For example, if the system runs out of memory, then the query transitions to the EXCEPTION state.

  The query can also go into an EXCEPTION state if it is cancelled.

Query cancellations may be triggered explicitly with a HiveServer2/Beeswax call or if the query times out. Query time-out may be set through a process-wide impalad argument or with a per-query option.

Currently, Impala does not have a state that explicitly indicates whether all Impala daemons have finished executing the query and that all results have been fetched. Let us call it as End of Statement (EOS), temporarily.

When a query is in the EOS (FINISHED) or EXCEPTION state, the query is not doing any more processing, but the query remains registered. It needs to remain registered because clients may need to access the state.

The query is unregistered only in the following two cases:

- The query is explicitly closed by a Close() API call
- The session associated with the query is closed explicitly or the session time-out is set and the session times out

Note: Hue does not close a query until you explicitly close it. When you close a browser tab on which you are running the query in Hue, the browser sends a JavaScript Close() callback request to close the query. If you leave the query unattended, for example by closing the laptop through which you are accessing Hue or if the browser crashes, then the Close() call is never sent to Hue. The query may eventually time out, but because it was not cancelled explicitly, it does not properly clean up the resources.

To optimize resource utilization, configure the Impala daemon to stop the idle sessions by setting the session timeout value in the --idle_session_timeout impalad argument:

1. Sign in to Cloudera Manager as an Administrator.
2. Go to Clusters Impala service Configuration .

3. Specify the following in the Impala Command Line Argument Advanced Configuration Snippet (Safety Valve) field:

   ```
   --idle_session_timeout=<maximum lifetime of your queries in seconds>
   ```

   For example,

   ```
   --idle_session_timeout=3600
   ```

   In this case, the query will time out after one hour.

Finding the list of Hue superusers

You can fetch the list of superusers by using the Hue shell with Python code or by running a SQL query on the auth _user table.

Using the Hue shell and Python code to find Hue superusers

1. Connecting to Hue shell by running the following command:

   ```
   /opt/cloudera/parcels/CDH/lib/hue/build/env/bin/hue shell --cm-managed
   ```

2. Enter the Python code as follows:

   ```
   from django.contrib.auth.models import User
   print "%s" % User.objects.filter(is_superuser = True)
   ```

   Sample output:

   ```
   <QuerySet [<User: admin>]>
   ```

Running a SQL query on the auth_user table to find Hue superusers

1. Connect to Hue database shell by running the following command:

   ```
   /opt/cloudera/parcels/CDH/lib/hue/build/env/bin/hue dbshell --cm-managed
   ```

2. Run the following SQL query:

   ```
   select username, is_superuser from auth_user where is_superuser=1;
   ```

   The superuser status is stored as a boolean value, though its representation varies by database: 1 for true and 0 for false, or t for true and f for false.

   Sample output:

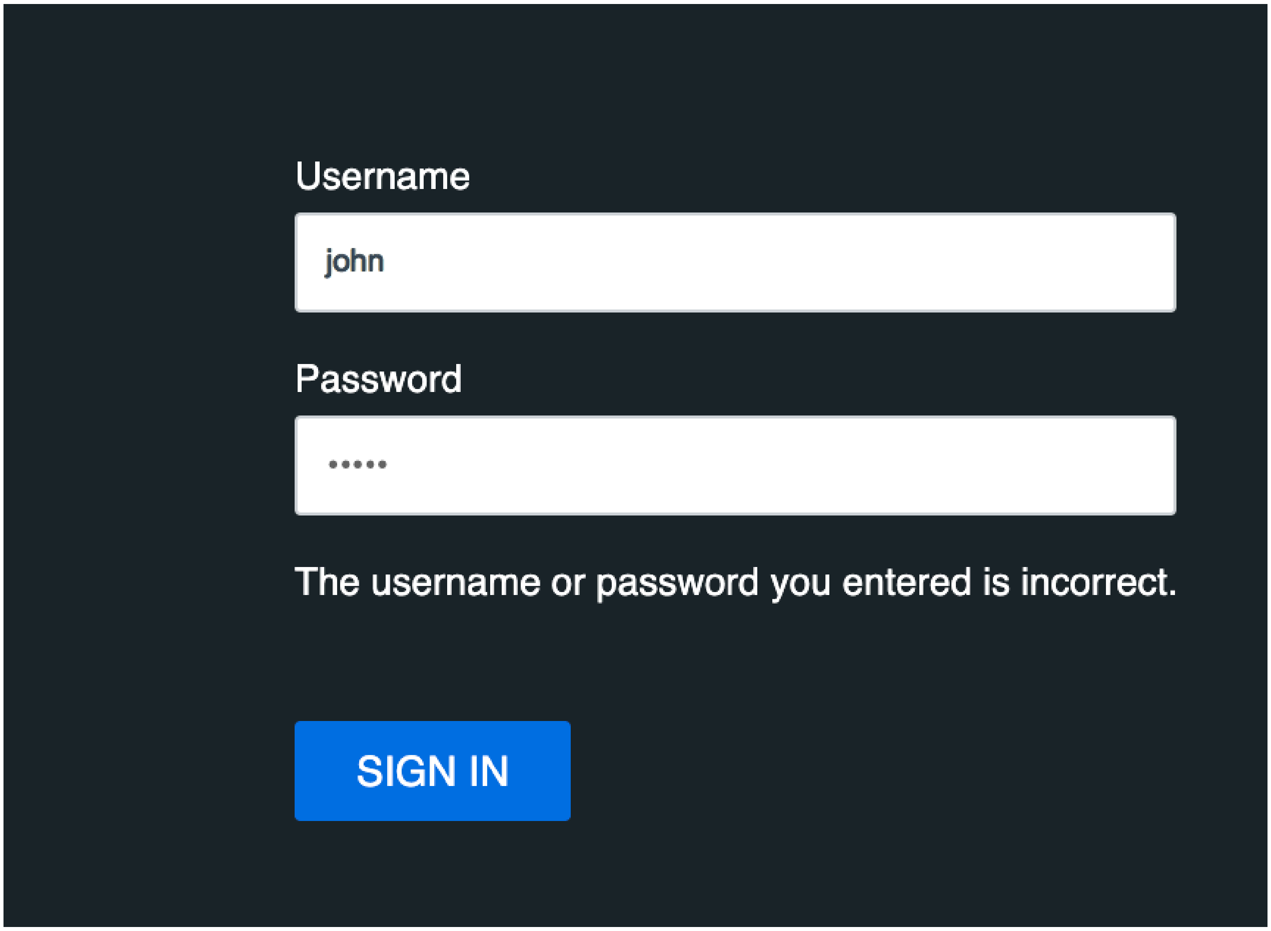
   ```
   ----------------------+
   username is_superuser
   ----------------------+

   admin 1
   ----------------------+
   1 row in set (0.00 sec)
   ```

Username or password is incorrect while accessing Hue through Knox

If an error such as "The username or password you entered is incorrect" appears when you try to log in to Hue using the Knox UI, then you can verify your credentials by logging in to the Knox Gateway using the command-line interface.

About this task

Figure 1: Knox Gateway UI: Incorrect username or password



Username: john

Password: •••••

The username or password you entered is incorrect.

Procedure

1. Open a terminal session.
2. SSH into the Knox Gateway host by entering the following command:

```
ssh [***KNOX-USERNAME***]@[***KNOX-HOST***].[***DOMAIN***].site
```

Replace [***KNOX-HOST***].[***DOMAIN***].site with the Fully-Qualified Domain Name (FQDN) of your Knox Gateway host.

For example:

```
ssh john@abc-1.example.com
```

3. Enter the password that you used on the Knox Gateway web UI.

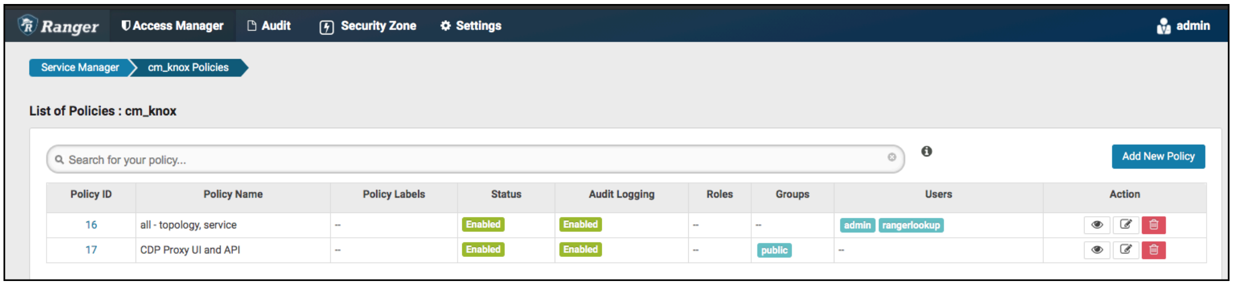
   If you are able to log in using these credentials, then you should be able to log into the Knox Gateway UI.

HTTP 403 error while accessing Hue UI from Knox

When an HTTP 403 error appears while accessing the Hue UI from the Knox Gateway, then the user or the group that the user belongs to may not have the required permissions.

Procedure

1. Sign in to Cloudera Manager as an Administrator.
2. Go to ClustersRanger service Ranger Admin Web UI and sign in to the Ranger Admin web UI as an Administrator.
3. Click on cm_knox.
4. The cm_knox policies are displayed.

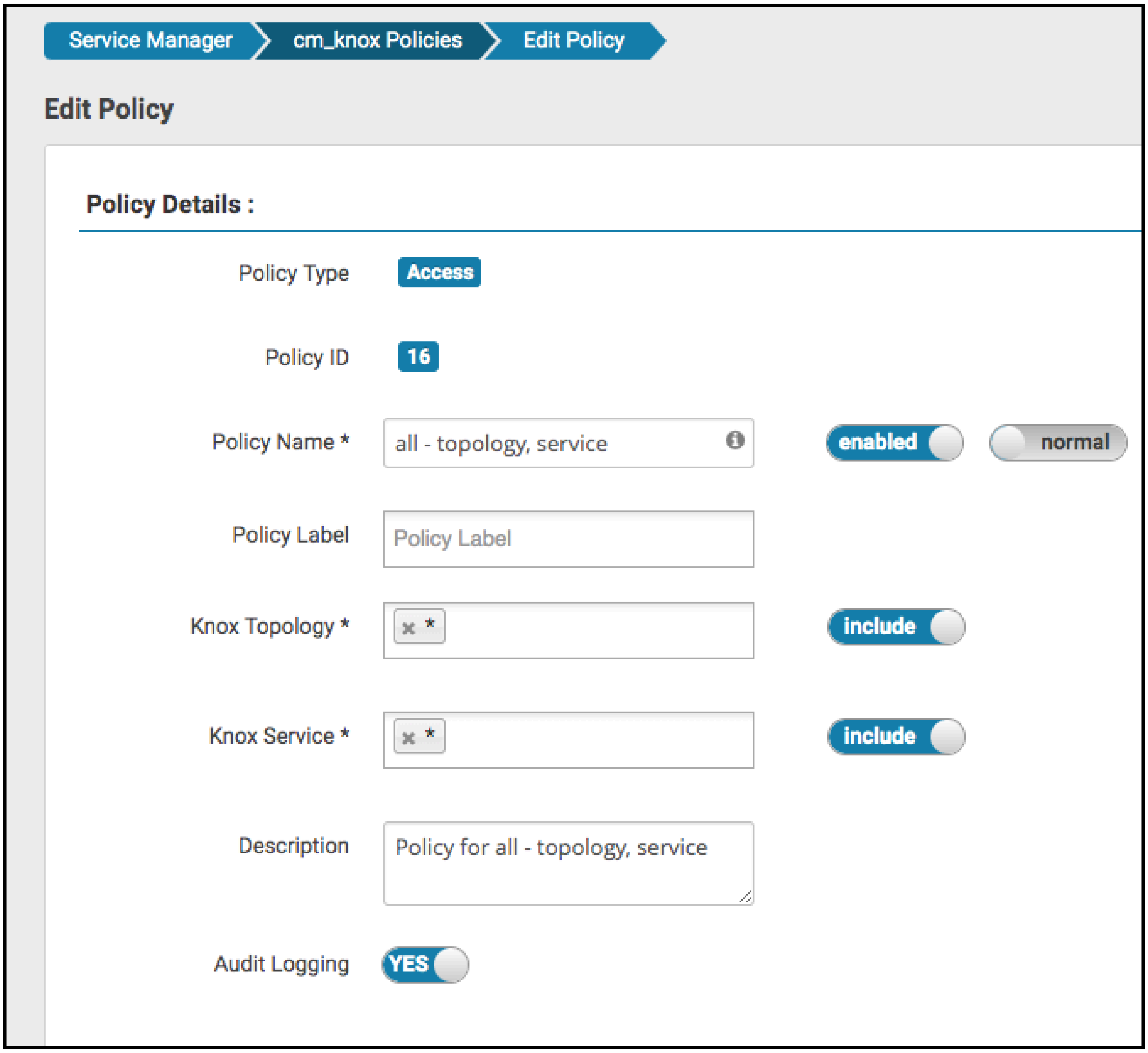


List of Policies : cm_knox

Policy ID	Policy Name	Policy Labels	Status	Audit Logging	Roles	Groups	Users	Action
16	all - topology, service	--	Enabled	Enabled	--	--	admin rangerlookup	👁 ✎ 🗑
17	CDP Proxy UI and API	--	Enabled	Enabled	--	public	--	👁 ✎ 🗑

5. Click on the policy ID 16 (all - topology, service).

   The **Edit Policy** page is displayed.



Service Manager > cm_knox Policies > Edit Policy

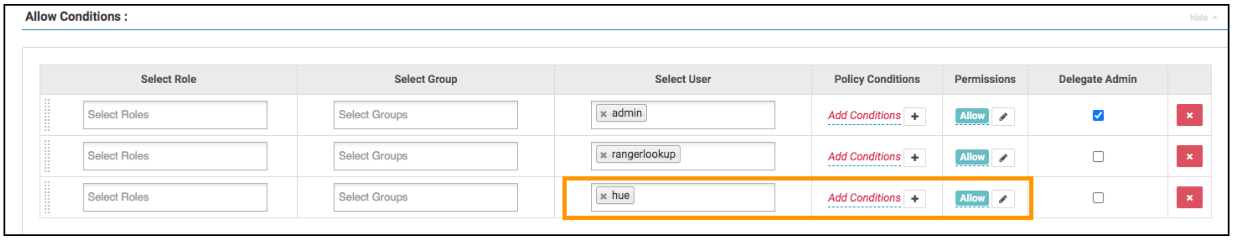
Edit Policy

Policy Details :

Policy Type	Access	
Policy ID	16	
Policy Name *	all - topology, service	enabled normal
Policy Label	Policy Label	
Knox Topology *	× *	include
Knox Service *	× *	include
Description	Policy for all - topology, service	
Audit Logging	YES	

6. Verify that the user and the group to which the user belongs has the required permissions.

   If the user or group does not have the required permission, then add the user or the user group to the Select User field and select Allow under Permissions.



Allow Conditions :

Select Role	Select Group	Select User	Policy Conditions	Permissions	Delegate Admin	
Select Roles	Select Groups	× admin	Add Conditions +	Allow	☑	×
Select Roles	Select Groups	× rangerlookup	Add Conditions +	Allow	☐	×
Select Roles	Select Groups	× hue	Add Conditions +	Allow	☐	×

'Type' error while accessing Hue from Knox Gateway

Condition

If you are using KnoxSpenego as an authentication mechanism, and if Knox is set up with Kerberos (HadoopAuth), then you may see the "type" error on the Hue web interface when you try to open Hue from the Knox Gateway UI

Cause

This is because of KNOX-2865.

Solution

Open a Knox support case with Cloudera to request for a hotfix.

Referer checking failed because domain does not match any trusted origins

You may see a "Referer checking failed" error in the Hue access.log file if the Knox Gateway DNS does not resolve properly when accessing Hue using the Knox Gateway UI.

About this task

Follow the steps listed in this topic if you see the following error in the Hue access.log file:

```
”POST /accounts/login HTTP/1.1" — Referer checking failed -https://<ip_addr
ess>:<knoxui_port>/gateway/cdp-proxy/hue/hue/accounts/login?next=%2F%253FdoA
s%253Dknoxui does not match any trusted origins.”
```

Before you begin

The Knox Gateway DNS may not resolve due to various reasons. Verify the following before proceeding:

- Verify that the Knox proxy user configuration is set properly in the Hue configuration
- Check whether the hostnames specified in the Hue service Configuration Knox Proxy Hosts fields are spelled correctly
- Make sure that the DNS is set up correctly by logging into the DNS server and pinging a few hosts
- Make sure that the /etc/hosts file has the correct mapping of the IP address and the hosts

Procedure

1. Go to Clusters $Knox service Instances and click on the Knox Gateway hostname.
2. Note down the IP address of the Knox Gateway host from the Details section.
3. Go to Clusters Hue service Configurations and search for the Knox Proxy Hosts field.
4. Enter the IP address of the Knox Gateway host that you noted earlier.
5. Click Save Changes.
6. Verify whether you can access the Knox Gateway through the following URL:

```
https://[***IP_ADDRESS***]:[***KNOXUI_PORT***]/gateway/cdp-proxy/hue/hue/
accounts/login?
```

Unable to view Snappy-compressed files

You must install the python-snappy library on your cluster to view files compressed with Snappy using the Hue File Browser and the HBase Browser. Post-installation, Hue automatically detects and displays the Snappy-compressed files.

Before you begin

The python-snappy library is incompatible with the python library called snappy. You must uninstall snappy if it is present on your cluster.

Run the following command to check whether the snappy library is installed on your cluster:

```
/usr/bin/pip show snappy
```

No output on the console indicates that the snappy library is not installed on your cluster. If you get any results for snappy, then uninstall it by running the following command:

```
/usr/bin/pip uninstall snappy
```

Next, check whether you have the python-snappy library is installed on your cluster by running the following command:

```
/usr/bin/pip show python-snappy
```

Sample output:

```
Name: python-snappy
Version: 0.5.4
Location: /usr/lib64/python2.7/site-packages
```

Procedure

1. Sign in to Cloudera Manager as an Administrator.
2. Stop the Hue service by going to Cluster Hue service Action .
3. Change to the following directory depending on whether you have used parcels or packages to set up your CDH cluster.

   For parcels:

   ```
   cd /opt/cloudera/parcels/CDH/lib/hue
   ```

   For package:

   ```
   cd /usr/lib/hue
   ```

4. Install the python-snappy package by running the following commands:

   ```
   yum install gcc gcc-c++ python-devel snappy-devel
   ./build/env/bin/pip install -U setuptools
   ./build/env/bin/pip install python-snappy
   ```

5. Verify that the python-snappy library is readable by all users by running the following commands:

```
ls -lart `locate snappy.py`
```

The output should be similar to the following:

```
-rw-r--r-- 1 root root 11900 Sep  1 12:25 /usr/lib64/python2.7/site-pack
ages/snappy.py
-rw-r--r-- 1 root root 10344 Sep  1 12:26 /usr/lib64/python2.7/site-packa
ges/snappy.pyc
```

6. Start the Hue service by going to Cluster Hue service Action .
7. Verify that the python-snappy library is working for Hue by running the following command:

```
sudo -u hue /bin/bash -c "echo 'import snappy' | python"
```

If the python-snappy library is working as expected, then no output is displayed for this command.

Results

You should be able to view Snappy-compressed files on the Hue File Browser and the HBase Browser using the Hue web interface.

"Unknown Attribute Name" exception when enabling SAML

You may see an "Unknown Attribute Name" exception when a SAML Identity Provider (IdP) returns the 'uid' profile attribute and Hue which uses pysaml2 cannot interpret this attribute. To resolve this, you must create an attribute mapping file and then reference it in the libsaml configuration of Hue.

Procedure

1. SSH into a Hue server as a root user.
2. Create an attribute mapping directory as follows:

```
mkdir -p /opt/cloudera/security/saml/attribute_mapping
```

3. Create an attribute mapping file as follows:

```
vi /opt/cloudera/security/saml/attribute_mapping/saml_uri.py
```

4. Add the following lines in the saml_uri.py file:

```
MAP = {
    "identifier": "urn:oasis:names:tc:SAML:2.0:attrname-format:uri",
    "fro": {
        'uid': 'uid',
        },
    "to": {
        'uid': 'uid',
    }
}
```

5. Repeat steps 1 to 4 on all the Hue hosts.
6. Sign in to Cloudera Manager as an Administrator.
7. Go to Clusters Hue service Configuration .

8. Add the following lines in the Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini field:

   ```
   [libsaml]
   xmlsec_binary=/usr/bin/xmlsec1
   metadata_file=/opt/certs/saml/FederationMetadata.xml
   key_file=/opt/certs/hue.key
   cert_file=/opt/certs/hue.crt
   entity_id=[***HOST-BASE-URL***]
   logout_enabled=false
   username_source=attributes
   attribute_map_dir=/opt/cloudera/security/saml/attribute_mapping
   #user_attribute_mapping='{"uid":"username"}'
   ```
9. Click Save Changes.
10. Restart the Hue service by clicking Actions Restart .

Results

The users should now be able to authenticate to Hue through SAML.

Impala query fails with invalid query handle error

You encounter an "Invalid query handle" error when running Impala queries from the Hue web interface because the connection between Impala Thrift server and the Hue Load Balancer times out. This is governed by the server_conn_timeout property.

About this task

The default value of the server_conn_timeout property is 30 minutes. You can increase the timeout limit by updating the Hue configuration using Cloudera Manager.

Procedure

1. Log into 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. Increase the value of the server_conn_timeout property in the Impala section as follows:

   ```
   [impala]
      server_host=[***SERVER-HOST***]
      server_port=[***PORT***]
      server_conn_timeout=[***TIMEOUT-IN-SECONDS***]
   ```

   You can increase the value of the server_conn_timeout property to 2 hours (7200 seconds).
4. Click Save Changes.
5. Restart the Hue service.

Results

You should be able to run Impala queries from the Hue web interface successfully.

Load balancing between Hue and Impala

Condition

You see the following error when running Impala queries from Hue: “"Invalid query handle error or Result have expired , rerun the query if needed"”. You also see either of the following errors in the runcpserver.log file:

- “Invalid query handle”
- “Invalid session id”

Cause

Hue uses a TCP connection pool (10 connections) for all Thrift traffic to Impala. This means that each Impala session is not guaranteed to use the same TCP connection. Load balancers send a single TCP connection to a single Impalad, but without correct persistence, Impala sessions can be sent to the wrong backend server, causing the errors you see.

Solution

Procedure

To solve this issue, you must configure your load balancer that is between Hue and Impala to use Source IP persistence. This is not the load balancer in front of Hue on port 8888/8889, this is the load balancer for Impala, defined in the Impala configuration in Cloudera Manager as Impala Daemons Load Balancer. In addition to Source IP persistence, you must also set the timeout in the load balancer for these connections to a bigger value, otherwise the load balancer can close these connections even though Hue is using them and thinks they are active. Cloudera recommends a minium of 6 hours as the timeout value. 12 hours is ideal.

Cloudera also recommends that you split the VIP configurations into 3 different ports, 21000 for impala-shell users, 21050 for JDBC users and then 21051 for Hue instances. This way you only have to configure the high timeout and Source IP persistence for the Hue port 21051.

Configure the HA Proxy as follows:

1. Open a terminal session and SSH in to the Impala Daemon.
2. Download and install an HAproxy service by running a command based on your operating system. For example:

```
yum install haproxy
```

3. Configure HAProxy for each role as follows:

```
vi /etc/haproxy/haproxy.cfg
```

```
# For impala-shell users on port 21000.
#---------------------------------------------------------------------
# main frontend which proxys to the backends
#---------------------------------------------------------------------
frontend  impala_front
    bind                 *:21000 ssl crt /opt/cloudera/security/x509/certk
eynopw.pem
    mode                 tcp
    option               tcplog
    default_backend      impala-shell

#---------------------------------------------------------------------
# round robin balancing between the various backends
#---------------------------------------------------------------------
backend impala-shell
    balance              leastconn
    mode                 tcp
    server impalad1 impalad-1.example.com:21000 check ssl ca-file /opt/cl
oudera/security/truststore/ca-truststore.pem
    server impalad2 impalad-2.example.com:21000 check ssl ca-file /opt/
cloudera/security/truststore/ca-truststore.pem
    server impalad3 impalad-3.example.com:21000 check ssl ca-file /opt/cl
oudera/security/truststore/ca-truststore.pem


# For JDBC or ODBC version 2.x driver, use port 21050 instead of 21000.
#---------------------------------------------------------------------
# main frontend which proxys to the backends
#---------------------------------------------------------------------
frontend  impala_front
    bind                 *:21050 ssl crt /opt/cloudera/security/x509/ce
rtkeynopw.pem
    mode                 tcp
    option               tcplog
    default_backend      impala-jdbc
#---------------------------------------------------------------------
# round robin balancing between the various backends
#---------------------------------------------------------------------
backend impala-jdbc
    balance              leastconn
    mode                 tcp
    server impalad1 impalad-1.example.com:21050 check ssl ca-file /opt/c
loudera/security/truststore/ca-truststore.pem
    server impalad2 impalad-2.example.com:21050 check ssl ca-file /opt/clo
udera/security/truststore/ca-truststore.pem
    server impalad3 impalad-3.example.com:21050 check ssl ca-file /opt/c
loudera/security/truststore/ca-truststore.pem


# Setup for Hue or other JDBC-enabled applications.
# In particular, Hue requires SOURCE IP PERSISTANCE
# The application connects to load_balancer_host:21051, and HAProxy bala
nces
# connections to the associated hosts, where Impala listens for JDBC
# requests on port 21050.
# Notice the timeouts below that do not exist in the other configs
# these are to stop the connections from being killed even though
# hue is using them
#---------------------------------------------------------------------
# main frontend which proxys to the backends
#---------------------------------------------------------------------
frontend  impalajdbc_front
    bind                 *:21051 ssl crt /opt/cloudera/security/x509/cert
keynopw.pem
    mode                 tcp
    option               tcplog
  timeout client    720m
  timeout server    720m
    default_backend      impala-hue

#---------------------------------------------------------------------
# source balancing between the various backends
#---------------------------------------------------------------------
backend impala-hue
    balance              source
    mode                 tcp
    server impalad1 impalad-1.example.com:21050 check ssl ca-file /opt/clo
udera/security/truststore/ca-truststore.pem
    server impalad2 impalad-2.example.com:21050 check ssl ca-file /opt/c
loudera/security/truststore/ca-truststore.pem
```

```
    server impalad3 impalad-3.example.com:21050 check ssl ca-file /opt/clo
udera/security/truststore/ca-truststore.pem
```

4. Create a persistence profile which enables source IP persistence in the F5. Make sure the timeout value is between 6 and 12 hours.

```
ltm persistence source-addr source_addr_12h_idle_timeout {
  app-service none
  defaults-from source_addr
  hash-algorithm default
  map-proxies enabled
  mask none
  match-across-pools disabled
  match-across-services disabled
  match-across-virtuals disabled
  override-connection-limit disabled
  timeout 43200
}
```

5. Create virtual profiles for each VIP, impala-shell, impala JDBC, and impala Hue.

```
impala-shell profile:

ltm virtual vs-impala-21000 {
  description "Impala 21000 for Shell"
  destination <VIPIP>:21000
  ip-protocol tcp
  mask 255.255.255.255
  pool pool-impala-21000
  profiles {
    fastL4 { }
  }
  source 0.0.0.0/0
  source-address-translation {
    type automap
  }
  translate-address enabled
  translate-port enabled
  vs-index 30
}

impala JDBC profile:
ltm virtual vs-impala-jdbc-21050 {
  description "Impala 21050 for JDBC"
  destination <VIPIP>:21050
  ip-protocol tcp
  mask 255.255.255.255
  pool pool-impala-21050
  profiles {
    fastL4 { }
  }
  source 0.0.0.0/0
  source-address-translation {
    type automap
  }
  translate-address enabled
  translate-port enabled
  vs-index 31
}
impala Hue profile:

ltm virtual vs-impala-hue-21051 {
  description "Impala 21051 for Hue"
  destination <VIPIP>:21051
  ip-protocol tcp
  mask 255.255.255.255
  persist {
    source_addr_12h_idle_timeout {
      default yes
    }
  }
  pool pool-impala-21050
  profiles {
    fastL4 { }
  }
  source 0.0.0.0/0
  source-address-translation {
    type automap
  }
  translate-address enabled
  translate-port enabled
  vs-index 32
}
```

Services backed by PostgreSQL fail or stop responding

When the number of connections between the Cloudera services and the PostgreSQL database exceeds the preset connection limit, it is possible that a new connection fails, Cloudera stops responding, and you cannot log into Hue. The logs show a “FATAL: remaining connection slots are reserved for non-replication superuser connections” error.

About this task

The number of connections between the Cloudera services and the PostgreSQL database is governed by the max_connections setting. By default, the maximum number of available connections to your PostgreSQL database is 115. 15 connections are reserved for the superuser to maintain the state and integrity of your database, and 100 connections are available for Cloudera and other services.

Note: There is a higher probability for this issue to occur in Hue because Cloudera Manager starts the Hue service after starting other Cloudera services. Therefore, the Hue service gets relatively fewer connections to PostgreSQL as compared to other services sharing the same database.

Procedure

1. Check the number of available and idle connections:
   a) SSH into the PostgreSQL database from the command-line client psql as an admin user.
   b) Run the following query to check the number of idle connections:

   ```
   SELECT datname, count(datname) FROM pg_stat_activity WHERE state = 'idle
   ' GROUP BY datname;
   ```

   c) Run the following query to check the number of connections currently in use:

   ```
   SELECT datname, count(datname) FROM pg_stat_activity GROUP BY datname;
   ```

   d) Run the following command to view the maximum number of connections:

   ```
   show max_connections;
   ```

   e) Run the following query to know where the connections are going:

   ```
   SELECT datname, numbackends FROM pg_stat_database;
   ```

2. If most connections are idle and the max_connections value is less than 100, then increase the max_connections value in the postgresql.conf file:
   a) Log into Cloudera Manager and stop all services that use the PostgreSQL database.
   b) SSH into the host on which the PostgreSQL server is running.
   c) Open the postgresql.conf file for editing.

      The postgresql.conf file is typically present in the /var/lib/pgsql/data directory. But this may vary depending on where you have installed the database.
   d) Increase the value of max_connections as per the following recommendation:

      Allow a maximum of 100 connections for each database and 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, Apache Atlas, and Hive Metastore), set the maximum connections to 550.
   e) Save the changes and exit.
   f) Restart the PostgreSQL database by running the following command:

   ```
   pg_ctl restart
   ```

   g) Restart all the affected services from Cloudera Manager.

What to do next

If increasing the connection limit does not solve your problem and you see a need to scale up, then add new PostgreSQL instances on other hosts and migrate the services to those hosts with the help of your Database Administrator (DBA).

Error validating LDAP user in Hue

Hive can use LDAP and Kerberos both, if they are enabled on your Cloudera cluster. By default, Hive uses LDAP to authenticate the Hue service instead of Kerberos. As a result, you may see the following error after logging in to Hue web interface or while trying to access the Hive editor: Bad status: 3 (PLAIN auth failed: Error validating LDAP user).

About this task

Additionally, you may not be able to view databases or Hive tables. To resolve this issue, you can enforce the client connections (between Hive and Hue) to use Kerberos instead of LDAP by configuring the value of the hive.server2 .authentication property to KERBEROS in the Hue hive-site.xml file.

Procedure

1. Log into Cloudera Manager as an Administrator.
2. Go to Clusters Hue service Configuration Hue Server Advanced Configuration Snippet (Safety Valve) for hive-site.xml .
3. Click View as XML and add the following lines in the text box:

```
<property>
    <name>hive.server2.authentication</name>
    <value>KERBEROS</value>
</property>
```

   Alternatively, you can click + to enable the Editor mode and specify hive.server2.authentication in the Name field and KERBEROS in the Value field.
4. Click Save Changes.
5. Restart the Hue service.

Results

The hive.server2.authentication property is appended to the hive/conf/hive-site.xml file. From now on, Hive will use Kerberos to authenticate access requests from the Hive editor within Hue and LDAP when you access Hive using Beeline.

502 Proxy Error while accessing Hue from the Load Balancer

When you access Hue from the Hue Load Balancer and encounter the "502 Proxy Error Proxy Error The proxy server received an invalid response from an upstream server. The proxy server could not handle the request POST /desktop/api/search/entities." error message, then increase the proxy timeout value for the Hue Load Balancer using Cloudera Manager.

Procedure

1. Log into Cloudera Manager as an Administrator.
2. Go to Clusters Hue service Configuration Scope Load Balancer Load Balancer Advanced Configuration Snippet (Safety Valve) for httpd.conf .
3. Add the following line in the Load Balancer Advanced Configuration Snippet (Safety Valve) for httpd.conf text box:

```
ProxyTimeout 600
```

   Note: If you are seeing the following error, then considering increasing the proxy timeout value to 1000 seconds:Proxy Error Proxy Error The proxy server received an invalid response from an upstream s erver. The proxy server could not handle the request POST /notebook/api/get_logs.
4. Click Save Changes.
5. Restart the Hue service.

Invalid method name: 'GetLog' error after submitting Hive queries

The Invalid method name: 'GetLog' (code THRIFTAPPLICATION): None error can occur after you submit Hive queries from the Hue editor and while Hue tries to fetch the result set. If you encounter this error, then set the beeswax use_get_log_api property to false using Cloudera Manager.

Procedure

1. Log into Cloudera Manager as an Administrator.
2. Go to Clusters Hue service Configuration .
3. Add/update the beeswax section as follows in the Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini field:

   ```
   [beeswax]
   use_get_log_api=false
   ```

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

"Authorization Exception" error on submitting queries in Hue

If you have secured your cluster using Ranger, then you must grant the required permissions to your users and groups from the Ranger web UI. If your users do not have proper permissions, then they may not be able to access certain databases or tables from the Hue editor.

About this task

If your users see the "Authorization Exception: User does not have privileges to execute..." error on submitting queries from the Hue editor, then grant them the proper permissions using the Ranger web UI.

Procedure

1. Log into Cloudera Manager as an administrator.
2. Go to Clusters Ranger service Instances tab and note down the hostname corresponding to the “Ranger Usersync” role type.
3. Open the Ranger web UI by clicking Ranger Admin Web UI.
4. SSH into the Ranger Usersync host that you noted in step 2 and add the user or the group as follows:

   ```
   ssh root@example.domain.site useradd [***USERNAME/GROUP-NAME***] passwd
   [***PASSWORD***]
   ```

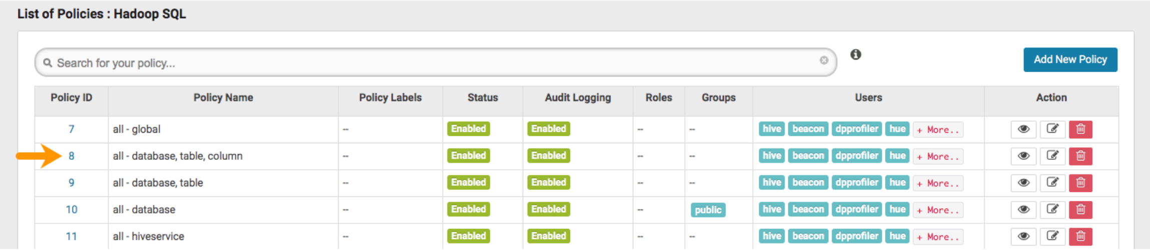
5. On the Ranger web UI, click Hadoop SQL listed under the HADOOP SQL service.

   The **Hadoop SQL Policies** page is displayed.

6. On the **Hadoop SQL Policies** page, you can grant the new user access to all the databases or to specific databases by adding a new policy.

- To grant the permission on all databases:

  a. Click the policy ID corresponding to "all - database, table, column".

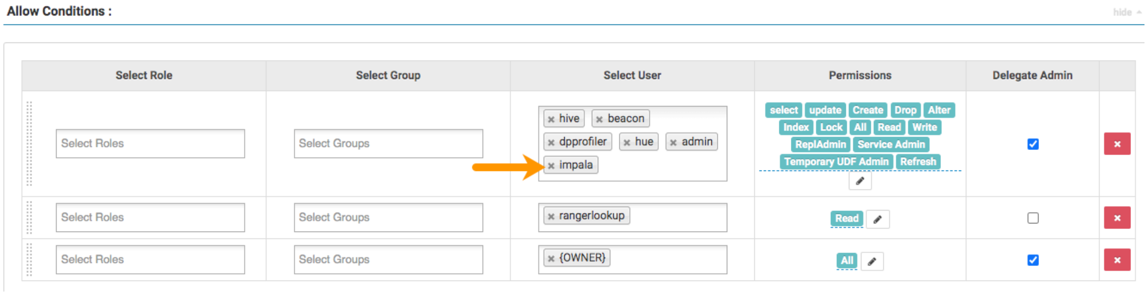


List of Policies : Hadoop SQL

Search for your policy...

Policy ID	Policy Name	Policy Labels	Status	Audit Logging	Roles	Groups	Users	Action
7	all - global	--	Enabled	Enabled	--	--	hive beacon dpprofiler hue + More..	
8	all - database, table, column	--	Enabled	Enabled	--	--	hive beacon dpprofiler hue + More..	
9	all - database, table	--	Enabled	Enabled	--	--	hive beacon dpprofiler hue + More..	
10	all - database	--	Enabled	Enabled	--	public	hive beacon dpprofiler hue + More..	
11	all - hiveservice	--	Enabled	Enabled	--	--	hive beacon dpprofiler hue + More..	

  b. On the **Edit Policy** page, add the user whom you want to grant the permission in the Select User field under the Allow Conditions section as shown in the following image:



Allow Conditions :

Select Role	Select Group	Select User	Permissions	Delegate Admin	
Select Roles	Select Groups	hive beacon dpprofiler hue admin impala	select update Create Drop Alter Index Lock All Read Write ReplAdmin Service Admin Temporary UDF Admin Refresh	☑	×
Select Roles	Select Groups	rangerlookup	Read	□	×
Select Roles	Select Groups	{OWNER}	All	☑	×

  To grant permissions to a group, enter the group name in the Select Group field.

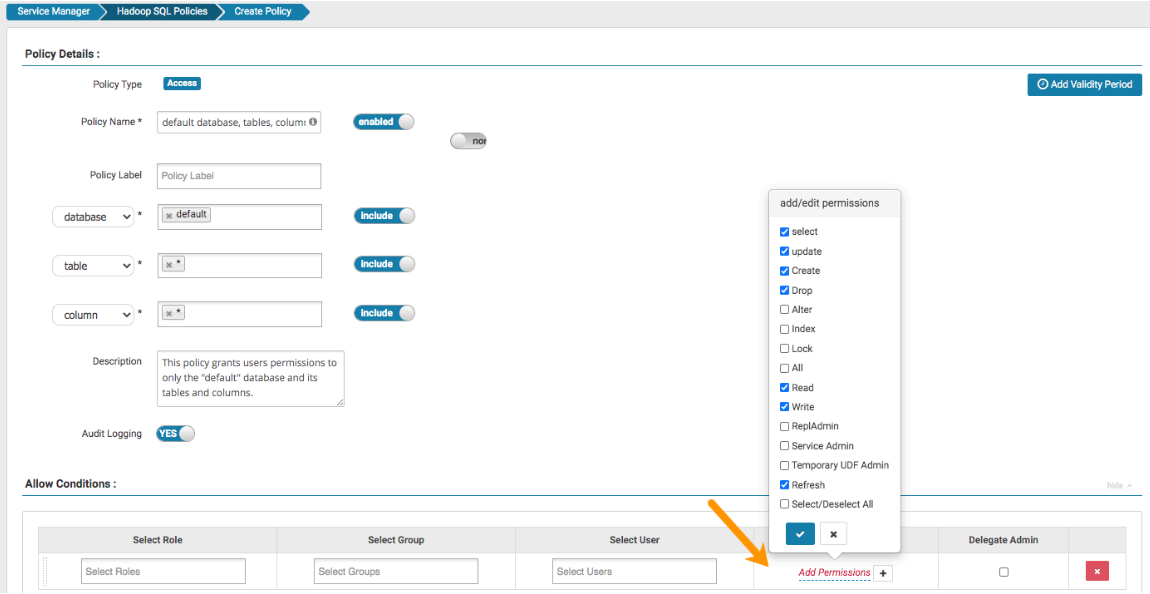
  c. Click Save.

- To grant permission on specific database:

  a. Click Add New Policy.

  The **Create Policy** page is displayed.

  b. Under the Policy Details section, specify the policy name and select the database, table, and column that you want your user to access as shown in the following image:



Policy Details :

Policy Type: Access

Policy Name *: default database, tables, columi — enabled

Policy Label: Policy Label

database *: default — include

table *: * — include

column *: * — include

Description: This policy grants users permissions to only the "default" database and its tables and columns.

Audit Logging: YES

add/edit permissions

☑ select
☑ update
☑ Create
☑ Drop
□ Alter
□ Index
□ Lock
□ All
☑ Read
☑ Write
□ ReplAdmin
□ Service Admin
□ Temporary UDF Admin
☑ Refresh
□ Select/Deselect All

Allow Conditions :

Select Role	Select Group	Select User	Permissions	Delegate Admin	
Select Roles	Select Groups	Select Users	Add Permissions +	□	×

  c. Under the Allow Conditions section, enter the username in the Select User field and click Add Permissions and select the permissions that your user must have.

To grant permissions to a group, enter the group name in the Select Group field.

   d. Click Add.

7. Start the Hue service from Cloudera Manager.

Results

The user or the group should be able to run any query on any entities as defined in the policy.

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.

Connection failed error when accessing the Search app (Solr) from Hue

If you are using Solr with Hue to generate interactive dashboards and for indexing data, and if you have deployed two Solr services on your cluster and selected the second one as a dependency for Hue, then Cloudera Manager assigns the hostname of the first Solr service and the port number of the second Solr service resulting in an incorrect Solr URL in the search section of the hue.ini file. As a result, you may see a “Connection failed” error when you try to access the Search app from the Hue web UI.

About this task

For example, consider two Solr services with the following configuration:

Solr-1, hostname=solr1, port:2345

Solr-2, hostname=solr2, port=4567

If you select Solr-2 as a dependent service for Hue, then Cloudera Manager updates the search section of the hue.ini file as follows:

```
[search]
# URL of the Solr Server
 solr_url=http://solr2:2345/solr/
```

As a result, you may not be able to access the Search app from the Hue web UI. To resolve this issue:

Procedure

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

   ```
   [search]
   # URL of the Solr Server
   solr_url=http://[***HOSTNAME***]:[***PORT***]/solr/
   ```

   For example:

   ```
   solr_url=http://solr2:4567/solr/
   ```
3. Click Save Changes.
4. Restart the Hue service.

Downloading query results from Hue takes time

If downloading query results from the Hue web UI takes time or if the operation exits with the "Invalid query handle" message, then you can improve the processing speed by increasing the number of threads used by the Hue web server.

Procedure

1. Sign in to Cloudera Manager as an Administrator.
2. Go to Clusters Hue service Configuration and search the cherrypy_server_threads property.

   The Hue Web Server Threads field is displayed.
3. Increase the thread count to 100 or a higher value.

   The default Hue web server thread count is 50.
4. Click Save Changes.
5. Restart the Hue service.

Hue Load Balancer does not start after enabling TLS

The Hue Load Balancer reads the private key file that is defined in the Hue Load Balancer TLS/SSL Server Private Key File (PEM Format) configuration property to start. Because the private key files are usually encrypted, the Hue Load Balancer must be configured to use the corresponding key password, without which it cannot start.

About this task

If you have enabled TLS for the Hue service on your cluster, and if the private key file is password protected (encrypted), then you may see the following error in the Hue Load Balancer log file (/var/log/hue-httpd/error_log):

```
AH02312: Fatal error initialising mod_ssl, exiting.
```

The following message is also logged in the /var/run/cloudera-scm-agent/process/[***XXX-HUE_LOAD_BALANCER***]/logs/stdout.log file:

```
CLOUDERA_HTTPD_USE_SSL=true
Apache/2.4.6 mod_ssl (Pass Phrase Dialog)
Some of your private key files are encrypted for security reasons.
In order to read them you have to provide the pass phrases.
```

```
Server example.test.com:443 (RSA)
Enter pass phrase:
```

To resolve this issue:

Procedure

1. Create a password file in your chosen security directory and insert the private key password as shown in the following example:

   ```
   echo "abc123" > /etc/security/password.txt
   ```

   Where abc123 is the private key password and password.txt is the password file.
2. Set the file ownership and permissions as shown in the following example:

   ```
   chown hue:hue password.txt
   chmod 700 password.txt
   ```
3. Enter the path to the file containing the passphrase used to encrypt the private key of the Hue Load Balancer server in the Hue Load Balancer TLS/SSL Server SSLPassPhraseDialog field.

   In this case, /etc/security/password.txt.
4. Click Save Changes.
5. Restart the Hue service.

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 Cloudera 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 service 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. If you see the ORA-12519: TNS:no appropriate service handler found 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;
```

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

Field	Property
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;
```

Enabling access to HBase browser from Hue

If HBase impersonation is not allowed in HDFS, then you may get an API error while trying to access the HBase browser from Hue. To resolve this issue, you must allow proxy users on the Thrift gateway, and also allow all groups form all hosts in HDFS to impersonate the hbase user.

Procedure

1. Log in to Cloudera Manager as an Administrator.
2. Go to Clusters HBase service Configuration and search for the hbase.thrift.support.proxyuser property.
3. Enable the Enable HBase Thrift Proxy Users option.
4. Click Save Changes.
5. Go to Clusters HDFS service Configuration .
6. Enter the following lines in the Cluster-wide Advanced Configuration Snippet (Safety Valve) for core-site.xml field:

```
<property>
<name>hadoop.proxyuser.hbase.hosts</name>
<value>*</value>
</property>
<property>
<name>hadoop.proxyuser.hbase.groups</name>
<value>*</value>
</property>
```

7. Click Save Changes.
8. Restart the HBase and HDFS services.

Unable to use pip command in Cloudera

You may not be able to use the pip command in Cloudera Base on premises releases 7.1.7 and above and may see the following error when using pip in a command: “ImportError: cannot import name chardet”.

About this task

To fix this issue and use the pip command, you must download pip from pypi.org and copy the chardet package to the target directory.

Procedure

1. SSH into the Hue host as a root user.
2. Change to the following directory:

```
cd /opt/cloudera/parcels/CDH/lib/hue
```

3. Check the pip version by running the following command:

   ```
   ./build/env/bin/pip --version
   ```

4. Download the pip package by running the following command:

   ```
   wget https://files.pythonhosted.org/packages/53/7f/55721ad0501a9076dbc35
   4cc8c63ffc2d6f1ef360f49ad0fbcce19d68538/pip-20.3.4.tar.gz -O ~/pip-20.3.
   4.tar.gz
   ```

5. Extract the tar file by running the following command:

   ```
   tar xvzf ~/pip-20.3.4.tar.gz -C ~
   ```

6. Copy the chardet package to the target directory by running the following command:

   ```
   cp -R ~/pip-20.3.4/src/pip/_vendor/chardet build/env/lib/python2.7/site-
   packages/pip/_vendor/.
   ```

7. Verify that you can run a pip command without any error by running any of the following sample commands:

   ```
   ./build/env/bin/pip freeze
   ```

   or

   ```
   ./build/env/bin/pip list
   ```

Related Information

pypi.org

Hue load balancer does not start after enabling TLS

If the Hue load balancer does not start after you enable TLS, then check whether your TLS private key file is password protected. To resolve this issue, you must configure the Hue load balancer to use the corresponding key password using Cloudera Manager.

About this task

At startup, the Hue load balancer reads the private key file as defined in the Cloudera Manager Clusters Hue service Configuration Hue Load Balancer TLS/SSL Server Private Key File (PEM Format) configuration property. If the TLS private key file is password protected, then you may see error such as Fatal error initialising mod_ssl, exiting and Some of your private key files are encrypted for security reasons. In order to read them you have to provide the pass phrases. in the /var/log/hue-httpd/error_log Hue load balancer log file.

Procedure

1. SSH into the Hue host as an Administrator.
2. Create a password file in your chosen security directory and insert the private key password, as shown in the following example:

   ```
   # echo "abc123" > /etc/security/password.txt
   # chown hue:hue password.txt
   # chmod 700 password.txt
   ```

3. Log in to Cloudera Manager as an Administrator.

4. Go to Clusters Hue service Configuration and enter the file path and the filename of the password file in the Hue Load Balancer TLS/SSL Server SSLPassPhraseDialog field.
5. Click Save Changes.
6. Restart the Hue service.

Unable to log into Hue with Knox

Opening Hue from Cloudera Manager or Knox Gateway Home may fail if the Kerberos principal name for the Knox service is different from the default Kerberos principal defined in the hue.ini file. To resolve this issue, you must set the value of the knox_principal property to your custom Kerberos principal name for Knox in the Hue Advanced Configuration Snippet, so that Hue's login mechanism (KnoxSpenegoDjangoBackend) can authenticate using Knox's Kerberos principal name.

About this task

If logging into Hue fails due to a mismatch between the custom Kerberos principal name for Knox and the default Kerberos principal name defined in the hue.ini file, then you may see the following error in the access.log file: Failed to verify provided username set(['*KNOX-PRINCIPAL-NAME*']) with set(['knox']).

Procedure

1. Log in to Cloudera Manager as an Administrator.
2. Note the Kerberos principal name for the Knox service from Clusters Knox service Configuration Kerberos Principal .
3. Go to Clusters Hue service Configuration Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini and enter the following lines:

   ```
   [desktop]
   [[knox]]
   knox_principal=[***KNOX-PRINCIPAL-NAME***]
   ```

   Replace [***KNOX-PRINCIPAL-NAME***] with the Kerberos principal name for the Knox service that you obtained earlier.
4. Click Save Changes.
5. Restart the Hue service.

Results

You should be able to log into Hue from the Knox Gateway UI cdp-proxy topology.

LDAP search fails with invalid credentials error

LDAP authentication fails with an "Invalid credentials" error, even if you input valid login credentials on the Hue login page, and you are unable to log into Hue. To resolve this issue, verify and update the LDAP Bind User credentials using Cloudera Manager.

About this task

This issue may happen if the credentials for the LDAP Bind User for Hue configured in Cloudera Manager are invalid. The invalid credentials could either be the “LDAP Bind Password" or "LDAP Bind User Distinguished Name". If the credentials are valid and the issue persists, verify that LDAP Search Base option in Cloudera Manager Hue Configurations is valid. The LDAP search base should be similar to 'dc=hadoop,dc=mycompany,dc=com'.

Before you begin

This task assumes that the Use Search Bind Authentication option is enabled in Cloudera Manager Hue Configurations . Search Bind Authentication connects to the LDAP server using the credentials provided in the 'bind_dn' and 'bind_password' configurations. If these configurations are not set, then an anonymous search is performed.

If the Use Search Bind Authentication option is not enabled in Cloudera Manager Hue Configurations , then do not set the LDAP Bind User credentials as described in this task. You must use the LDAP Username Pattern field for configuring the LDAP credentials, and verify whether the authentication works as expected.

Procedure

1. Log in to Cloudera Manager as an Administrator.
2. Go to Clusters Hue Configurations .
3. Set the LDAP Bind User credentials in the following fields:

   - LDAP Bind User Distinguished Name
   - LDAP Bind Password

   You can specify the LDAP Bind User Distinguished Nameeither in the generic LDAPv3 Distinguished Name ("CN=binduser,OU=users,DC=Example,dc=com") format or the Active Directory style (binduser@EXAMPLE.COM) format.
4. Click Save Changes.
5. Restart the Hue service.

Unable to execute queries due to atomic block

Condition

You may see the following error after submitting a query from Hue, while logging into Hue, or while saving documents and workflows: "TransactionManagementError: An error occurred in the current transaction. You can't execute queries until the end of the 'atomic' block."

Cause

If there is a load on Hue's backend database and slower processing of operations such as saving documents or workflows, running INSERT or UPDATE queries from multiple users, then the database puts an atomic block on the table rows. If you are using MySQL or MariaDB as the backend database for Hue, then you can increase the value of the innodb-lock-wait-timeout parameter along with a few other timeout parameters in the my.cnf file.

Solution

Procedure

1. SSH into the database host as an Administrator.
2. Back up the my.cnf configuration file as follows:

   ```
   cp /etc/my.cnf  /[***BACKUP-DIRECTORY***]
   ```
3. Open the my.cnf file for editing and add the following two lines under the [mysqld] section:

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

   ```
   [mysqld]
   wait_timeout = 28800
   ```

```
interactive_timeout = 28800
```

4. Save the file and exit from the editor.
5. Restart the database server.

Hue service does not start after a fresh installation or upgrade

Condition

Hue service fails during startup with the following error after a fresh installation or after performing an upgrade:

```
File "/cs/cloudera/opt/cloudera/parcels/CDH-7.1.8-1.cdh7.1.8.p0.30990532/lib
/hue/build/env/lib/python3.8/site-packages/gunicorn/util.py", line 173, in c
hown
    os.chown(path, uid, gid)
PermissionError: [Errno 1] Operation not permitted: '/tmp/wgunicorn-qfm_jl
ch'
```

Cause

This happens because the Hue process user and group specified in the Hue configuration in Cloudera Manager are different than the default information present in the hue.ini file.

Solution

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 update the values of server_user and server_group properties as follows:

   ```
   [desktop]
   server_user=[***CUSTOM-HUE-USER***]
   server_group=[***CUSTOM-HUE-GROUP***]
   ```

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

Query Process fails to start intermittently due to access issues in Java 9 and later

Condition

When running a Hue service on Java 9 or later, you may encounter IllegalAccessError exceptions related to module access restrictions. These errors occur because certain internal Java classes (e.g., sun.net.util.IPAddressUtil) are not accessible by default due to Java’s module system.

Example Error Messages:

```
[Guice/ErrorInjectingConstructor]: IllegalAccessError: class SecurityUtil$Qu
alifiedHostResolver (in unnamed module @0x31c88ec8)
cannot access class IPAddressUtil (in module java.base)
```

```
because module java.base does not export sun.net.util to unnamed module @0
x31c88ec8
```

Cause

Starting with Java 9, the Java Platform Module System restricts access to internal Java classes. If your application relies on these internal classes, the JVM throws an IllegalAccessError unless the module explicitly exports the necessary packages.

Solution

Procedure

1. Log in to Cloudera Manager as an Administrator.
2. Go to Clusters Hue service Configuration query_processor_java_opts option.
3. Add JVM arguments to allow access.

   ```
   --add-exports java.base/sun.net.dns=ALL-UNNAMED --add-exports java.base/
   sun.net.util=ALL-UNNAMED
   ```

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

Unable to access Hue after upgrading

Condition

You are not able to access Hue after upgrading to CDP Private Cloud Base 7.1.8 on a cluster secured using Knox.

Cause

The Hue Load Balancer is not installed, and the SetEnv proxy-sendcl configuration is missing. When integrating Hue with Knox, it is essential to deploy a Hue Load Balancer and specify its hostname in the Knox Proxy Hosts field. Without this configuration, Hue may not be accessible after the upgrade.

Solution

Procedure

1. Log in to Cloudera Manager as an Administrator.
2. Add the Hue Load Balancer role instance.
3. Go to the Configuration tab and add the following line in the Load Balancer Advanced Configuration Snippet (Safety Valve) for httpd.conf field:

   ```
   SetEnv proxy-sendcl 1
   ```

4. Click Save Changes.
5. Restart the Hue Load Balancer and Hue service.

Unable to run the freeze command

Condition

You see the /usr/lib/hue/build/env/bin/python: No such file or directory error when you run the following command:

```
build/env/bin/pip3.8 freeze
```

Cause

This issue is caused because of an incorrect path in Hue's pip3.8 file.

Solution

Procedure

- Run the freeze command by specifying the paths of python3.8 and pip3.8 as follows:

  ```
  /opt/cloudera/parcels/CDH/lib/hue/build/env/bin/python3.8 /opt/cloudera/
  parcels/CDH/lib/hue/build/env/bin/pip3.8 freeze
  ```

Disabling the web metric collection for Hue

Condition

You see the following error in the Hue server logs: DatabaseError: ORA-01000: maximum open cursors exceeded, when you are using Oracle as the backend database for Hue.

Cause

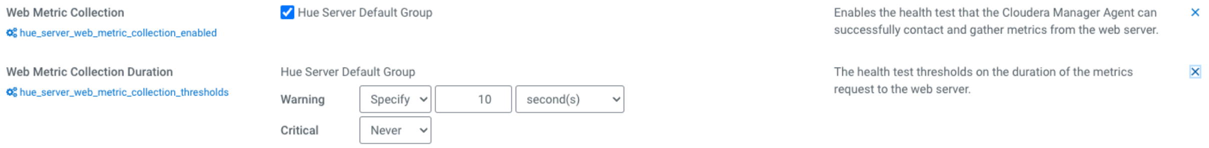
Cloudera Manager Agent collects web metrics from the Hue server to monitor its health. At times, the number of metrics that are collected can exhaust the number of available cursors on the Oracle database, resulting in the ORA-01000: maximum open cursors exceeded error.

Solution

Procedure

1. Log in to Cloudera Manager as an Administrator.
2. Go to Clusters Hue Configuration and deselect the Web Metric Collection option.

   Note: Restarting the Hue service from time to time can also clear the cursors, but temporarily. For a long-term resolution, consider turning off the Web Metric Collection option or reducing the frequency of the metric collection using the Web Metric Collection Duration option in Cloudera Manager.

   Web Metric Collection
   hue_server_web_metric_collection_enabled
   ☑ Hue Server Default Group
   Enables the health test that the Cloudera Manager Agent can successfully contact and gather metrics from the web server.

   Web Metric Collection Duration
   hue_server_web_metric_collection_thresholds
   Hue Server Default Group
   Warning Specify 10 second(s)
   Critical Never
   The health test thresholds on the duration of the metrics request to the web server.

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

Resolving "The user authorized on the connection does not match the session username" error

Condition

In Cloudera, if you try to open two Hue sessions on different ports on the same browser window, then you may see the “The user authorized on the connection does not match the session username” error on the Hue web interface and in the Hue logs.

Cause

On CDH, if you opened Hue web URL on port 8889 on one tab and on port 8888 on another tab, you would be logged out of Hue on the port 8889, and you would be forced to refresh the page. In Cloudera, Hue depends on the ZooKeeper quorum for how it connects to Impala or Hive and how the Hue load balancer distributes the connections to the Hue server nodes and controls switching between the different sesions.

Solution

After migrating/upgrading from CDH to Cloudera, note this change in behaviour and always open Hue web UI using the Hue Load Balanced - recommended option from Cloudera Manager.

Requirements for compressing and extracting files using Hue File Browser

Downloading multiple files from the Hue File Browser is not supported directly. If you want to download multiple files, you can compress them into a zip file and download the zip file. To use the compress and extract function, you must have the Oozie service installed on your cluster, along with other zip and unzip utilities.

Condition

You do not see the Compress option on the **File Browser** page in Hue.

Cause

This can happen because you may not have the Oozie service installed on your cluster, or it is not configured to be used with Hue.

Solution

You must have the Oozie service installed on your cluster. After installing Oozie, go to Cloudera Manager Clusters Hue service Configuration , and select the Oozie Service option. Click Save Changes and restart the Hue service.

You must also install the following utilities for compressing and extracting files on your cluster:

- zip
- unzip
- tar
- bzip2

Note: The option to compress files is not available in Cloudera Data Warehouse.

Fixing a warning related to accessing non-optimized Hue

Condition

You see the following messgae when you open Hue: “You are accessing a non-optimized Hue, please switch to one of the available addresses”.

Cause

This message is displayed when you have a Load Balancer role, but none of the Hue instances are linked to it.

Solution

Procedure

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

   If you are using an external load balancer, then you must add the list of the load balancers under the [desktop] section in the Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini field as follows:

   ```
   [desktop]
   hue_load_balancer=[***LOAD-BALANCER-HOST-1***],[***LOAD-BALANCER-HOST-
   2***]
   ```

   Note: If you have added more than one Hue load balancers (Httpd), then restart the Hue service. The hue_load_balancer configuration is automatically updated after restarting Hue.
3. Click Save Changes.
4. Restart the Hue service.

Fixing incorrect start time and duration on Hue Job Browser

Some users may notice incorrect values in the "Started" and "Duration" columns on the Job Browser page in Hue. This could be because the user is unauthorized to view the application according to the Job ACLs. Learn more about this issue and how to fix it.

Condition

Some users see that the "Started" and "Duration" columns show wrong or unexpected values. For example, the Started column displays a value such as: "1 January 1970 00:00" and Duration column displays a value such as: "0s". However, all users do not face this issue. For example, the admin user can see the values fine.

Cause

This issue occurs when Hue accesses the Resource Manager (RM) when Job ACLs are enabled. To confirm this, run the following command and check the output:

```
curl -kiv --negotiate -u : http://[***RM-HTTP-ADDRESS***]:8088/ws/v1/clus
ter/apps/[***APP-ID***]
```

Sample output:

```
{"app":{"id":"appid","user":"testuser","name":"PySparkShell","queue":"root.u
sers.testuser",
"state":"FINISHED","finalStatus":"SUCCEEDED","progress":100.0,
"trackingUI":"History","trackingUrl":"http://[***RM-HTTP-ADDRESS***]:8088/pr
oxy/[***APP-ID***]/",
"diagnostics":"","clusterId":12345678,"applicationType":"SPARK","application
Tags":"",
"startedTime":0,"finishedTime":0,"elapsedTime":0,"allocatedMB":0,"allocate
dVCores":0,
"runningContainers":0,"memorySeconds":175016,"vcoreSeconds":170,
"preemptedResourceMB":0,"preemptedResourceVCores":0,
"numNonAMContainerPreempted":0,"numAMContainerPreempted":0}}
```

"startedTime"=0 in the output indicates that the user is not authorized to view the application according to the following two Job ACLs:

- application VIEW acl (i.e. mapreduce.job.acl-view-job in combination with mapreduce.cluster.acls.enabled)
- queue ADMIN acl (aclAdministerApps for the FS)

Solution

Procedure

1. Log in to Cloudera Manager as an Administrator.
2. Go to Clusters YARN Configuration and add the following lines in the YARN Service MapReduce Advanced Configuration Snippet (Safety Valve) and MapReduce Client Advanced Configuration Snippet (Safety Valve) for mapred-site.xml field:

   ```
   <property>
   <name>mapreduce.cluster.acls.enabled</name>
   <value>true</value>
   </property>
   <property>
   <name>mapreduce.job.acl-view-job</name>
   <value>mapred,hue group1,group2</value>
   </property>
   ```
3. Restart the Job History Service and run Deploy Client Configuration.
4. Restart all affected services.

Hue fails to start due to missing Oracle Client Library

Condition

Installation of Oracle fails due to missing shared libraries, resulting in an error related to loading specific library files. This issue is encountered while loading shared libraries libnsl.so.1 in an application environment.

Example Error Message:

```
Installing Oracle.19.0.0.0.0
replace /u01/app/19.0.0/grid/instantclient/libsglplusic.s0? [y]es, [n]o, [A
]ll, [N]one, [r]ename:y
config? [y]es, [n]o, [A]ll, [N]one, [r]ename:y
replace /u01/app/19.0.0/grid/opmn/adnin/libons. def? [y]es, [n]o, [A]ll,
[N]one, [r]ename:A
replace /u01/app/oracle/product/19.0.0/db_1/drdaas/admin/drdasqtt_transla
tor_setup.sql? [y]es, [n]o, [A]ll, [N]one, [r]ename:A
```

```
u01/app/19.0.0/grid/perl/bin/perl: error while loading shared libraries: l
ibnsl.so.1: cannot open shared object file: No such file or directory
Fresh install failed.. deleting /tmp/aveksa/install, /home/oracle/Aveksa_S
ystem.cfg and /export/home/oracle/setDeployEnv.sh as part of cleanup.
step failed! See /tmp/aveksa-install.log for more information.
```

Cause

The required shared library, libnsl.so.1, is not installed by default in certain operating system versions. Many modern Linux distributions have deprecated or removed some legacy libraries, requiring manual installation to maintain compatibility with older applications.

Solution

Procedure

1. Install libnsl manually using the following command.

   ```
   sudo dnf install libnsl*
   ```

2. Confirm that the library libnsl.so.1 is installed correctly in /usr/lib64/ using the following command.

   ```
   find / -iname libnsl.so.1
   ```

3. Restart Hue.

Hue SSL-enabled MySQL connection issues

Condition

When using Hue with SSL-enabled MySQL, connection issues may occur due to unsupported TLS protocols in newer MySQL client versions.

Example Error Message:

```
ERROR: django.db.utils.OperationalError: (2026, 'SSL connection error: error
:14077102:SSL routines:SSL23_GET_SERVER_HELLO:unsupported protocol')
```

Cause

The error occurs because MySQL client version 8.0.28 and later has removed support for TLSv1 and TLSv1.1. As a result, SSL connections using these older protocols fail.

Solution

Procedure

1. Locate the my.cnf configuration file for your MySQL client.
2. To ensure secure connections, configure your supported TLS versions similar to the example below:

   ```
   [mysqld]
   tls_version = TLSv1.2,TLSv1.3 \\ Example setting
   ```

   Replace the TLS versions configured in MySQL with the versions supported by your database system.
3. Restart the MySQL service to apply the changes.

Hue is unable to create a Spark3 session with Livy3

Condition

This issue occurs when trying to open the Spark3 editor through Hue and receiving a pop-up displaying "Extra data: line 1 column 5."

Sample command to verify:

```
export LIVY_HOST=https://$(hsotname )-f:28998
curl -k --negotiate -u : -X POST --data '{"kind": "spark"}' -H "Content-Ty
pe: application/json" ${LIVY_HOST}/sessions
```

Example exception seen for the above command:

```
org.apache.hadoop.security.authorize.AuthorizationException: User:livy not a
llowed to do 'DECRYPT_EEK' on 'cdp_kms_key'
```

Cause

The issue arises due to insufficient permissions for the Livy user to perform theDECRYPT_EEK operation on the cdp_kms_key, which prevents the Livy session from initializing.

Solution

Grant the DECRYPT_EEK permission to the Livy user in Ranger. Once this permission is added, the Spark3 editor works correctly.

java.io.EOFException when reading DAG or Hive proto data files

Condition

You might encounter a java.io.EOFException error when reading DAG proto data files during processing with the Hue Query processor or observability tools. These errors typically occur intermittently or after an abrupt ApplicationMaster (AM) termination or Out-of-Memory (OOM) event.

When this happens, proto files used in data pipelines fail to load correctly, leading to job failures or pipeline interruptions.

Example Error Message

```


Caused by: java.io.EOFException
    at java.base/java.io.DataInputStream.readFully(DataInputStream.java:202)
    at org.apache.hadoop.io.DataOutputBuffer$Buffer.write(DataOutputBuffe
r.java:70)
    at org.apache.hadoop.io.DataOutputBuffer.write(DataOutputBuffer.java:
120)
    at org.apache.hadoop.io.SequenceFile$Reader.next(SequenceFile.java:2505)
    at org.apache.hadoop.io.SequenceFile$Reader.next(SequenceFile.java:2637)
    at org.apache.hadoop.mapred.SequenceFileRecordReader.next(SequenceFil
eRecordReader.java:82)
    at org.apache.hadoop.hive.ql.io.protobuf.ProtobufMessageInputFormat$1
.next(ProtobufMessageInputFormat.java:124)
```

```
    at org.apache.hadoop.hive.ql.io.protobuf.ProtobufMessageInputFormat$1
.next(ProtobufMessageInputFormat.java:84)
    at org.apache.hadoop.hive.ql.io.HiveContextAwareRecordReader.doNext(Hi
veContextAwareRecordReader.java:365)
    ... 24 more
```

Cause

- DAG proto files become empty or partially written due to an abrupt AM termination or OOM event.
- These incomplete files cannot be identified without attempting to read them.
- Reading these partially written files results in an EOFException error.

Solution

Procedure

1. Remove empty or partially written proto files.

   Manually identify and delete zero-byte or abnormally small DAG or Hive proto files before initiating processing. This prevents the EOFException error during read operations.
2. Build or use a proto file validation tool.

   Hive proto files can become corrupt, and no direct method exists to identify which file is affected. To address this, you can use or develop a validation tool that processes proto files and flags corrupted ones.
3. Validate proto files using the Hue Query Processor.

   The Hue Query Processor includes a command-line utility to process Hive proto files.

   a. Log in to an active Hue instance through SSH.

      ```
      ssh user@[***hue-host***]
      ```
   b. Create a temporary directory with appropriate permissions for processing.

      ```
      mkdir /tmp/proto
      sudo chmod 1777 /tmp/proto
      sudo chown -R hive:hive /tmp/proto
      ```
   c. Copy the proto file from HDFS to the local temporary directory.

      ```
      sudo -u hive hdfs dfs -get /warehouse/tablespace/managed/hive/sys.db/que
      ry_data/date=YYYY-MM-DD/hive_[***file_id***] /tmp/proto/
      ```

      Replace [***file_id***] with the actual file id name.
   d. (Optional) Transfer the proto file to your local machine if further processing is required.

      ```
      scp user@backup-server:/backup/location/tmp/proto/hive_[***file_id***] /
      tmp
      ```
   e. Convert the proto file to JSON format using the Hue query processor event processor tool.

      ```
      java -jar data_analytics_studio-event-processor-<version>.jar protodump
      -type hive -src /tmp/proto/hive_[***file_id***] -dest /tmp/json/output.j
      son
      ```
4. Automate validation and cleanup.

   Integrate a preprocessing validation step in your data ingestion or maintenance workflows to automatically identify and remove corrupted proto files before they cause processing failures.