Cloudera Runtime 7.1.6

Securing Hue

Date published: 2019-07-28
Date modified: 2021-03-19

https://docs.cloudera.com/

Legal Notice

© Cloudera Inc. 2024. All rights reserved.

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

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

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

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

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

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

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

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

Contents

User management in Hue 5
- Understanding Hue users and groups 5
- Finding the list of Hue superusers 6
- Creating a Hue user 7
- Restricting user login 8
- Creating a group in Hue 8
- Managing Hue permissions 9
- Resetting Hue user password 10
- Assigning superuser status to an LDAP user 11

User authentication in Hue 11
- Authenticating Hue users with Kerberos 11
- Authenticating Hue users with LDAP 12
  - Import and synchronize users and groups with an LDAP server in Hue 15
  - Configuring authentication with LDAP and Search Bind 17
  - Configuring authentication with LDAP and Direct Bind 19
  - Configuring Hue for authentication against multiple LDAP/Active Directory servers 20
  - Testing the LDAP configuration 21
  - Configuring group permissions 22
  - Enabling LDAP authentication with HiveServer2 and Impala 22
  - LDAP properties 23
  - Configuring LDAP on unmanaged clusters 25
- Authenticating Hue users with SAML 26
  - Configuring SAML authentication on managed clusters 26
  - Manually configuring SAML authentication 27
  - Integrating your identity provider's SAML server with Hue 29
  - SAML properties 29
  - Troubleshooting SAML authentication 31
- Authenticating Hue users with Knox SSO 31

Applications and permissions reference 34

Securing Hue passwords with scripts 35

Configuring TLS/SSL for Hue 36
- Creating a truststore file in PEM format 37
- Configuring Hue as a TLS/SSL client 37
- Enabling Hue as a TLS/SSL client 37
- Configuring Hue as a TLS/SSL server 38
- Enabling Hue as a TLS/SSL server using Cloudera Manager 38
- Enabling TLS/SSL for Hue Load Balancer 39
- Enabling TLS/SSL communication with HiveServer2 39
- Enabling TLS/SSL communication with Impala 40
- Securing database connections with TLS/SSL 40

Enforcing TLS version 1.2 for Hue **41**

Securing sessions **42**

Specifying HTTP request methods **45**

Restricting supported ciphers for Hue **45**

Specifying domains or pages to which Hue can redirect users **45**

Setting Oozie permissions **46**

Configuring secure access between Solr and Hue **47**

User management in Hue

Hue is a gateway to CDP cluster services and both have completely separate permissions. Being a Hue superuser does not grant access to HDFS, Hive, and so on.

Users who log on to the Hue UI must have permission to use Hue and to each CDP service accessible within Hue.

A common configuration is for “Hue users” to be authenticated with an LDAP server and “CDP users” with Kerberos. These users can differ. For example, CDP services do not authenticate each user who logs on to Hue. Rather, they authenticate “Hue” and trust that Hue has authenticated “its” users.

Once Hue is authenticated by a service such as Hive, Hue impersonates the user requesting use of that service. For example, to create a Hive table. The service uses Apache Ranger to ensure the group to which that user belongs is authorized for that action.

Hue user permissions are at the application level only. For example, a Hue superuser can filter Hue user access to a CDP service but cannot authorize the use of its features. Again, Ranger does that.

Understanding Hue users and groups

There are two types of users in Hue - superusers and general users referred to as users, each with specific privileges. These users can be a part of certain groups. Groups enable you to control which Hue applications and features your users can view and access when they log into Hue.

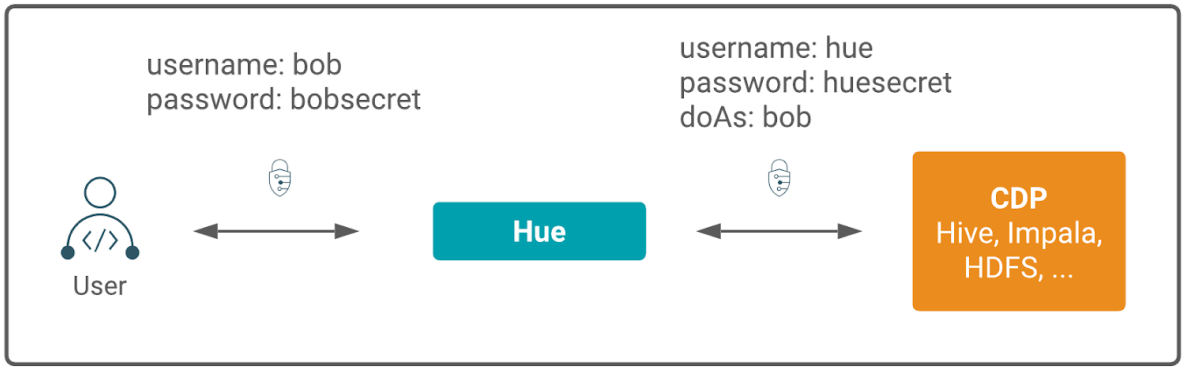
On a non-secure CDP cluster, the first user logging into Hue after the initial installation becomes the first superuser. Superusers have the permissions to perform the following administrative functions:

- Add and delete users
- Add and delete groups
- Assign permissions to groups
- Change a user into a superuser
- Import users and groups from an LDAP server

If a user is part of the superuser LDAP group in Hue, then that user is also a part of the group of superusers in Hue.

Users can only change their name, e-mail address, and password. They can log in to Hue and run Hue applications, subject to the permissions provided by the Hue groups to which they belong. This is different from how CDP perceives the Hue application when you submit a Hive or an Impala query from the Hue user interface (UI). Hue is a server between the users and the CDP services. Hue is considered as a single ‘hue’ user by the other services in the CDP cluster.

For example, when a user ‘bob’ submits a query from Hue, Hue also sends the username of this user to the corresponding service in CDP. The HIVE_ON_TEZ service in CDP considers ‘bob’ as the owner of the query and not ‘hue’. This is illustrated in the following graphic:



Hue is a gateway to CDP cluster services and both have separate permissions. A Hue superuser is not granted access to HDFS, Hive, and other CDP cluster services. Apache Ranger governs access to the CDP cluster services.

Note: Groups in Hue are different from groups in Ranger.

Hue user permissions are at the application level only. For example, a Hue superuser can filter Hue user access to a CDP service but cannot authorize the use of its features. Users who log on to the Hue UI must have permission to use Hue and to each CDP service accessible within Hue.

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>]>
```

Runing 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;
```

Sample output:

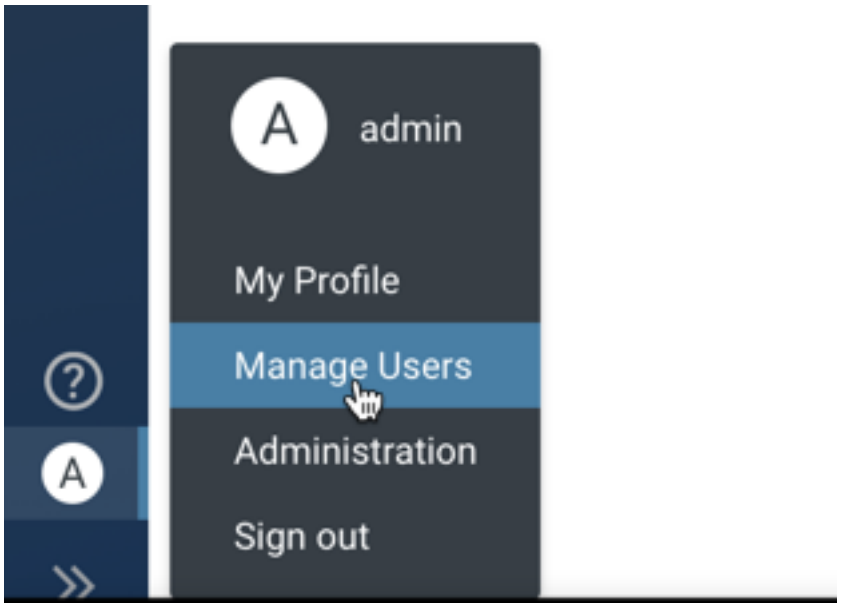
```
----------------------+
username is_superuser
----------------------+

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

Creating a Hue user

You can create new Hue users and superusers from the Hue web UI and assign them to groups so that they can view and access Hue as per the permissions granted to them.

Procedure

1. Sign in to the Hue UI as a superuser.
2. From the left assist panel, point your cursor to the user profile icon and click Manage Users.



The **User Admin** page is displayed.

3. On the **User Admin** page, click Add User.

   The **Create user** page is displayed.
4. Enter the username and password for the user that you are adding on the Credentials tab.

   To create a separate Hue home directory for the user, select the Create home directory option.

   Click Next.
5. On the Profile and Group tab, create a profile for the user by entering the details such as name and email address.

   At this point, if you have already created a group(s) that you want to assign to the user, then select it from the list displayed in the Groups field.

   A user can be a part of more than one group.

   Click Next.

6. (Optional) On the Advanced tab, select the Superuser status option to make this user a superuser and click Add user.

   The new user is displayed on the **Users** page.

Restricting users who can log in to Hue

Admins can control users who can access Hue by creating different LDAP login groups, adding users needing access to these login groups, and then specifying the login groups in the Advanced Configuration Snippet in Cloudera Manager.

Before you begin

Ensure that you have added users needing access to Hue to a login group.

Procedure

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

   ```
   [desktop]
   [[ldap]]
   login_groups=[***LDAP-GRP1***], [***LDAP-GRP2***], [***LDAP-GRP3***]
   ```

   Where, *[***LDAP-GRP1***]*, *[***LDAP-GRP2***]*, *[***LDAP-GRP3***]* are the login groups containing users needing access to Hue.
3. Click Save Changes.
4. Restart the Hue service.

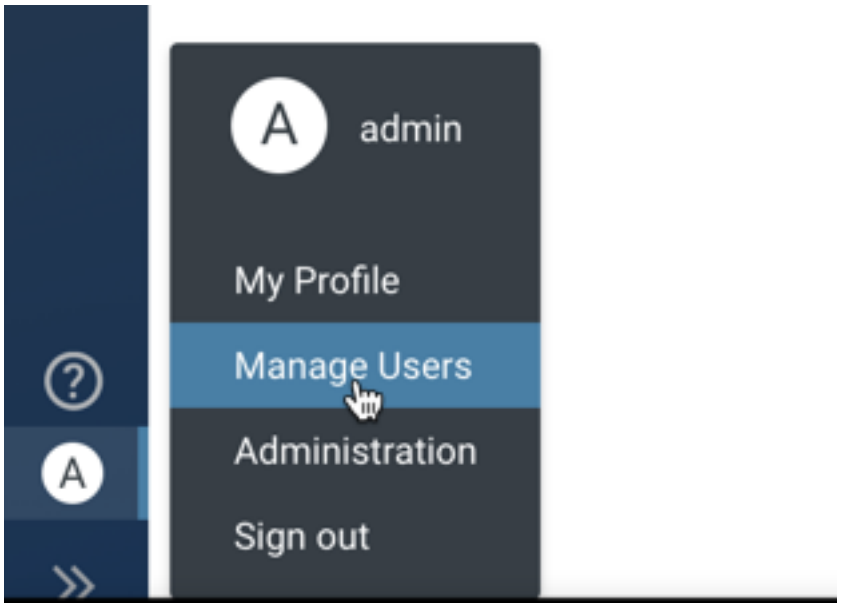
Creating a group in Hue

By creating groups, you can club certain permissions that you want to assign to specific users in your organization.

Procedure

1. Sign in to the Hue UI as a superuser.

2. From the left assist panel, point your cursor to the user profile icon and click Manage Users.



The **User Admin** page is displayed.

3. From the **User Admin** page, go to the Groups tab.

   The **Groups** page displays the list of existing groups, if any.

4. Click Add group.
5. On the **Create group** page, specify a name for your group.
6. (Optional) You can select the users that you want to add to this group.
7. Select the permissions that you want to associate with the group and click Add group.

   The newly added group is displayed on the **Groups** page along with the list of members and permissions associated with it.

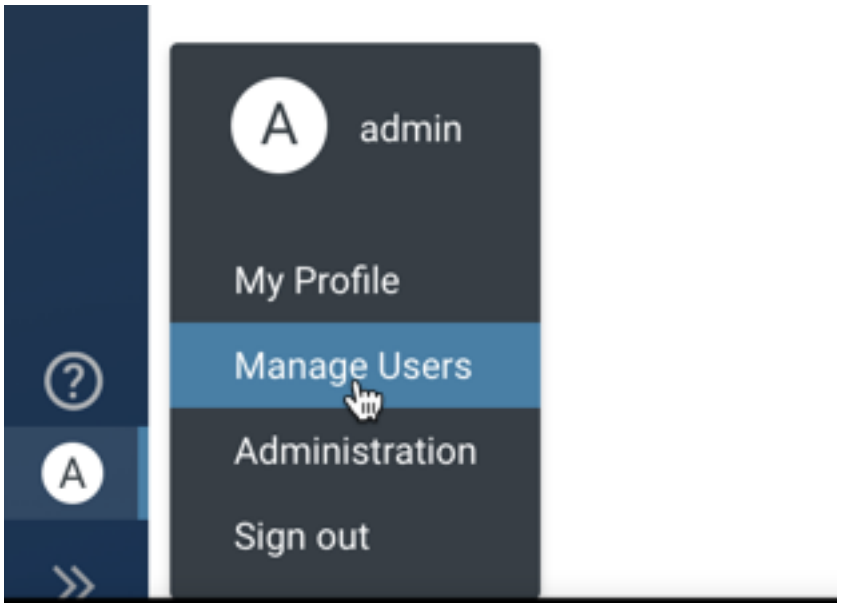
Managing Hue permissions

Permissions for Hue applications are granted to groups, with users gaining permissions based on their group membership. Group permissions define the Hue applications visible to group members when they log in to Hue and the application features available to them. There is a fixed set of Hue permissions. You cannot add or modify permissions. However, you can apply permission to group(s).

Procedure

1. Sign in to the Hue UI as a superuser.

2. From the left assist panel, point your cursor to the user profile icon and click Manage Users.



The **User Admin** page is displayed.

3. From the **User Admin** page, go to the Permissions tab.

   The **Permissions** page displays the list of all the available permissions.

4. Click a permission that you want to assign to a group(s).

   The **Edit [permission name]** page is displayed.

5. Select the group(s) on which you want to apply the permission and click Update permission.

   The “Permission information updated successfully” message is displayed.

Resetting Hue user password

The first user logging into Hue after its initial installation becomes the first superuser. Even if a user does not log into the Hue UI, the first security scan may log in creating an initial user and therefore resulting in an unknown username and password. You can change the password for a user if you know the username or you can create a new superuser user and then use it to log in to Hue and change the password for a user.

Procedure

1. Sign in to the Hue server as the root user and go to the Hue home directory.
2. If you know the user ID of the currently logged in user, then reset the password by running the following command:

   ```
   build/env/bin/hue changepassword [***USER-ID***] --cm-managed
   ```

   Replace the *USER-ID* with the actual ID of the user.

3. If you do not know the user ID of the user whose password you want to change, then create a new Hue admin user by running the following command:

   ```
   build/env/bin/hue createsuperuser --cm-managed
   ```

   After creating a new admin user, log in to Hue and reset the password for a given user ID.

Assigning superuser status to an LDAP user

The Hue User Admin application provides two levels of privileges: users and superusers. The superusers have administrative privileges.

About this task

Users can change their name, email address, and password. They can log in to Hue and run Hue applications according to their group permissions.

Superusers can perform administrative functions such as:

- Add and delete users and groups
- Import and sync users and groups from an LDAP server
- Assign group permissions
- Promote users to superusers and vice versa.

Hue superusers have no special privileges to the underlying CDP cluster services. Ranger is used to add those privileges.

Important: On a non-secure cluster, the first user to log in to Hue without LDAP authentication becomes the first superuser.

Procedure

In a secure cluster with LDAP deployed, there are three ways to assign superuser status to a user:

- With desktop.auth.backend.AllowAllBackend set for the Authentication Backend property in Cloudera Manager temporarily enabled, assign superuser status and synchronize one user to the LDAP server.
- With desktop.auth.backend.LdapBackend set for the Authentication Backend property in Cloudera Manager, run a Hue shell command to apply superuser status.
- Enable multiple backends so that the first user to log on still works when integrated with LDAP.

User authentication in Hue

CDP services do not authenticate each user that logs in to Hue. The CDP services authenticate Hue and trust that Hue has authenticated its users. In a most typical configuration, Hue users can be authenticated with an LDAP server and the CDP users can be authenticated with Kerberos. You can also use SAML for Single Sign-on (SSO) authentication.

After Hue is authenticated by a service such as Hive, Hue impersonates the user requesting the use of that service, for example, to create a Hive table. In this case, the Hive service uses Apache Ranger to ensure that the group to which the user belonged is authorized for that action (to create a Hive table).

Note: By default, the Hue session uses a secure cookie protocol.

Authenticating Hue users with Kerberos

For Hue to work properly with a CDP cluster that uses Kerberos for authentication, the Kerberos Ticket Renewer role must be added to the Hue service.

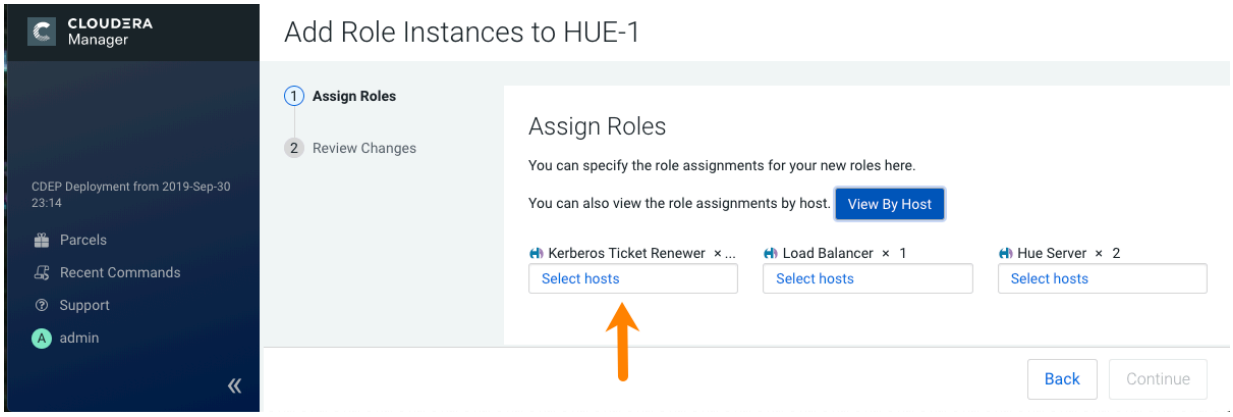
About this task

Use the Cloudera Manager Admin Console to add the Kerberos Ticket Renewer role to each host with a Hue Server role instance. The Hue Kerberos Ticket Renewer renews only those tickets created for the Hue service principal:

hue/*hostname@REALM-NAME*. The Hue principal impersonates other users for applications within Hue such as the Job Browser, File Browser, and so on. Other services, such as HDFS and MapReduce, do not use the Hue Kerberos Ticket Renewer. Instead these other services handle ticket renewal as needed by using their own mechanisms.

Procedure

1. On the Cloudera Manager home page, select the Hue service.
2. On the Hue service page, click the Instances tab.
3. On the Instances page, click Add Role Instances on the right side of the page. This launches the Add Role Instances wizard.
4. To add a Kerberos Ticket Renewer role instance to the same host that has the Hue server on your CDP cluster, click Select hosts under Kerberos Ticket Renewer:



   To check which host has the Hue Server role instance, click View By Host, which launches a table that lists all the hosts in your CDP cluster and shows all the roles each host already has.
5. In the host selection dialog box, after selecting the host where you want to add the Kerberos Ticket Renewer role instance, click OK, and Cloudera Manager adds the role instance.
6. After processing the request to add the role instance, Cloudera Manager returns you to the Instances page and prompts you to restart the service. Click the Restart the service (or the instance)... link so the configuration change can take effect.
7. After the services have restarted, click Finish to return to the Instances page.

   Repeat these steps for each Hue Server role on your cluster.

What to do next

Troubleshooting the Kerberos Ticket Renewer:

If the Hue Kerberos Ticket Renewer does not start, check the configuration of your Kerberos Key Distribution Center (KDC). Look at the ticket renewal property, maxrenewlife, to ensure that the principals, hue/<*host_name*> and krbtgt, are renewable. If these principals are not renewable, run the following commands on the KDC to enable them:

```
kadmin.local: modprinc -maxrenewlife 90day krbtgt/<YOUR_REALM.COM>
kadmin.local: modprinc -maxrenewlife 90day +allow_renewable hue/
<host_name>@<YOUR_REALM>
```

Authenticating Hue users with LDAP

Configuring Hue for Lightweight Directory Access Protocol (LDAP) enables you to import users and groups from a directory service, synchronize group membership manually or automatically at login, and authenticate with an LDAP server.

Hue supports Microsoft Active Directory (AD) and open standard LDAP such as OpenLDAP and Forgerock OpenDJ Directory Services.

Integrating Hue with LDAP

When Hue is integrated with LDAP, users can use their existing credentials to authenticate and inherit their existing groups transparently. There is no need to save or duplicate any employee password in Hue.

When authenticating using LDAP, Hue validates login credentials against an LDAP directory service if Hue is configured with the LDAP authentication backend (desktop.auth.backend.LdapBackend) in Cloudera Manager.

The LDAP authentication backend automatically creates users that do not exist in Hue by default. Hue needs to import users to properly perform the authentication. Passwords are never imported when importing users. You can disable automatic import of users by setting the create_users_on_login property in the Cloudera Manager Clusters Hue service Configuration Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini field to false.

```
[desktop]
[[ldap]]
create_users_on_login=false
```

The purpose of disabling the automatic import is to allow only a predefined list of manually imported users to login.

Preserving the case of the usernames

If you are using mixed case, upper case, or Camel case for usernames in your LDAP directory, then you must add the following configurations in Hue's Advanced Configuration Snippet (Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini), so that the user names are not over written in lower case in the Hue's database. Set the values of these properties to true or false depending on your requirement:

```
[desktop]
[[auth]]
ignore_username_case=true
force_username_uppercase=true/false
force_username_lowercase=true/false
[[ldap]]
ignore_username_case=true
force_username_uppercase=true/false
force_username_lowercase=true/false
```

Binding Hue with LDAP

There are two ways to bind Hue with an LDAP directory service:

Search Bind

The search bind mechanism for authenticating will perform an ldapsearch against the directory service and bind using the found distinguished name (DN) and password provided. This is the default method of authentication used by Hue with LDAP.

You can restrict the search process by configuring the following two properties under the Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini [desktop] > [[ldap]] > [[[users]]] section.

Property	Description
user_filter	General LDAP filter to restrict the search. Default: "objectclass=*"
user_name_attr	The attribute that will be considered the username to be searched against. Typical attributes to search for include: uid, sAMAccountName. Default: sAMAccountName

With the above configuration, the LDAP search filter takes the following form:

```
(&(objectClass=*)(sAMAccountName=[***USERNAME-ENTERED-BY-USER***
]))
```

Note: Setting search_bind_authentication=true tells Hue to perform an LDAP search using the bind credentials specified for the bind_dn and bind_password configuration properties. Hue will start searching the subtree starting from the base DN specified for the base_dn property. It will then search the base DN for an entry whose attribute, specified in user_name_attr, has the same value as the short name provided on login. The search filter, defined in user_filter will also be used to limit the search.

Direct Bind

The direct bind mechanism for authenticating binds to the LDAP server using the username and password provided at login.

Hue authenticates (without searching) in one of two ways:

- NT Domain (nt_domain): (Only for use with Microsoft Active Directory) Hue binds to the AD with username@domain using the User Principal Names (UPN) to bind to the LDAP service. This AD-specific property allows Hue to authenticate with AD without having to follow LDAP references to other partitions. This typically maps to the email address of the user or the user's ID in conjunction with the domain. Default: mycompany.com.
- Username Pattern (ldap_username_pattern): Bind to open standard LDAP with full path of directory information tree (DIT). It provides a template for the DN that is ultimately sent to the directory service when authenticating. The [***USERNAME***] parameter is replaced with the username provided at login.

  Default:

  ```
  "uid=[***USERNAME***],ou=People,dc=mycompany,dc=com"
  ```

  Note: Setting search_bind_authentication=false tells Hue to perform a direct bind to LDAP using the credentials provided (not bind_dn and bind_password specified in the Advanced Configuration Snippet). There are two ways direct bind works depending on whether the nt_domain property is specified in the Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini :

  - nt_domain is specified: This is used to connect to an Active Directory service. In this case, the UPN is used to perform a direct bind. Hue forms the UPN by concatenating the short name provided at login with the nt_domain. For example, [***SHORT-NAME***]@[***NT-DOMAIN***]. The ldap_username_pattern property is ignored.
  - nt_domain is not specified: This is used to connect to all other directory services (can handle AD, but nt_domain is the preferred way for AD). In this case, ldap _username_pattern is used and takes the following form:

    ```
    cn=[***USERNAME-PROVIDED-AT-LOGIN***],dc=example,dc=
    com
    ```

Note: Username pattern does not work with AD because AD inserts spaces into the UID which Hue cannot process.

Encryption

To prevent credentials from transmitting in the clear, encrypt with LDAP over SSL, using the LDAPS protocol on the LDAPS port, which uses port 636 by default. An alternative, is to encrypt with the StartTLS operation using

the standard LDAP protocol, which uses port 389 by default. Cloudera recommends LDAPS. You must have a CA Certificate in either case.

Table 1: Hue Supported LDAP authentication and encryption methods

LDAP Auth Action	Encrypted (LDAPS)	Encrypted (LDAP+TLS)	Not Encrypted (LDAP)
Search Bind	AD, LDAP	AD, LDAP	AD, LDAP
Direct Bind - NT Domain	AD	AD	AD
Direct Bind - User Pattern	LDAP	LDAP	LDAP

Prerequisites

To authenticate Hue users with LDAP, you must have:

- LDAP server
- Bind account (or support for anonymous binds)
- Cloudera Manager access with Full Administrator permissions
- [optional] LDAP server with LDAPS or StartTLS encryption.

Important: To authenticate securely, configure your LDAP server with either LDAP over SSL (LDAPS) or StartTLS encryption. Both methods require a Certificate Authority (CA) chain in a .pem file.

Import and synchronize users and groups with an LDAP server in Hue

Configuring Hue for Lightweight Directory Access Protocol (LDAP) enables you to import users and groups from a directory service, synchronize group membership manually or automatically at login, and authenticate users with LDAP.

About this task

To synchronize your Hue users and groups with your LDAP server:

- Hue must be configured to authenticate with LDAP.
- The logged in user must have Hue superuser permissions.

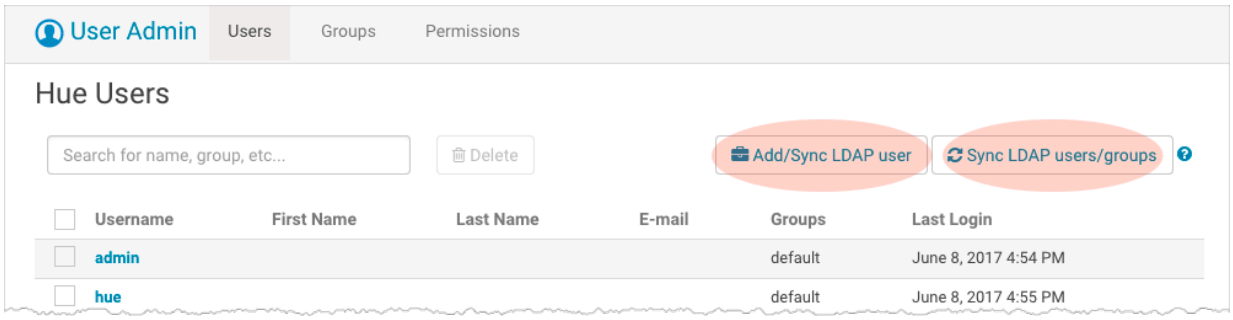
There are four LDAP import and sync options in Hue:

LDAP Sync Action	Description
Add/Sync LDAP user	Import and synchronize one user at a time
Sync LDAP users/groups	Synchronize user memberships in all groups
Add/Sync LDAP group	Import and synchronize all users in one group
sync_groups_at_login	Automatically synchronize group membership at login

Note: Hue does not support importing all groups at once.

Procedure

1. Import and synchronize LDAP users in Hue:



To import and synchronize one LDAP user in Hue:

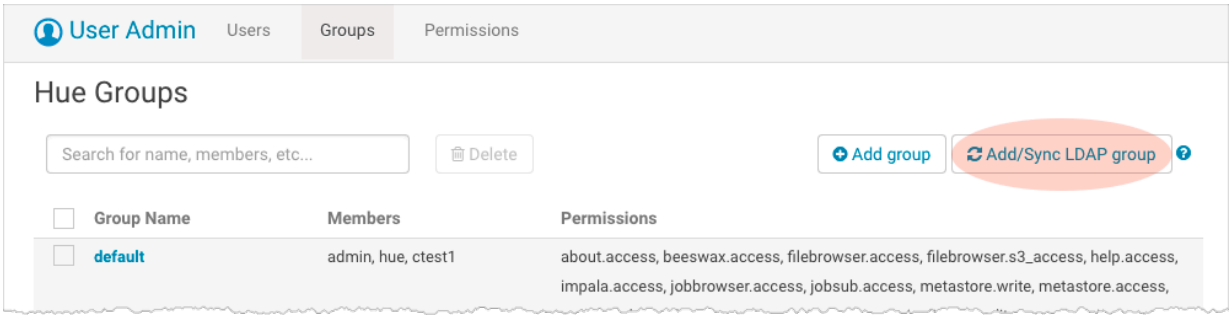
a. Log on to the Hue UI as a superuser.
b. Go to User AdminUsers.
c. Click Add/Sync LDAP user.
d. Add a username, check Create home directory, and click Add/Sync user.

To synchronize group memberships for LDAP users who have already been imported to Hue:

Note: This synchronizes group memberships with the LDAP server.

a. Log on to the Hue UI as a superuser.
b. Go to User AdminUsers.
c. Click Sync LDAP users/groups.
d. Check Create home directories, and click Sync.

2. Import and synchronize LDAP groups in Hue:



To import and synchronize one LDAP group containing its users:

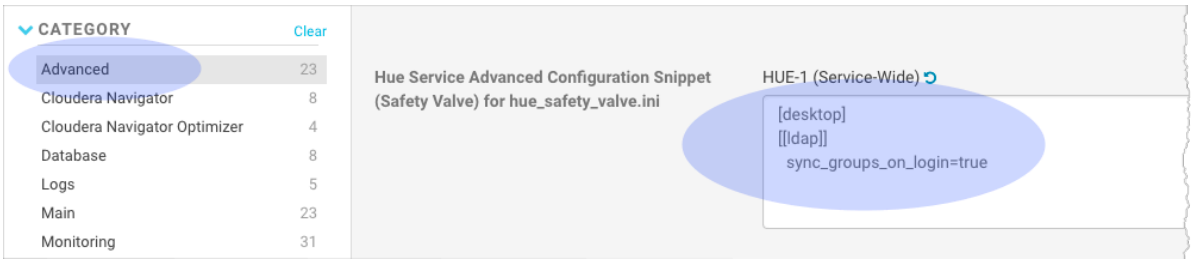
a. Log on to the Hue UI as a superuser.
b. Go to User AdminGroups.
c. Click Add/Sync LDAP group.
d. Check Create home directories, and click Sync.

To configure Hue to automatically synchronize LDAP groups and their users when they log in to Hue:

Note: LDAP sync_groups_at_login only works with search bind authentication.

a. Log on to Cloudera Manager and click Hue.
b. Click the Configuration tab and filter by scope=Service-wide and category=Advanced.
c. Enter the following text in the Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini text box:

```
[desktop]
[[ldap]]
  sync_groups_on_login=true
```



d. Click Save Changes and Restart Hue.

Configuring authentication with LDAP and Search Bind

Search Bind authentication executes ldapsearch against one or more directory services and binds with the distinguished name (DN) and password. Hue searches the subtree from the base distinguished name. If the LDAP Username Attribute is set, Hue looks for an entry whose attribute has the same value as the short name given at login.

About this task

Important: Search Binding works with all directory service types. It is also the only method that allows synchronizing groups at login (set with sync_groups_on_login in a safety-valve).

Video: Authenticate Hue with LDAP and Search Bind

Figure 1: Video: Authenticate Hue with LDAP and Search Bind

Procedure

1. Log on to Cloudera Manager and click Hue.
2. Click the Configuration tab and filter by scope=Service-wide and category=Security.
3. Set the following required properties:

Authentication Backend	desktop.auth.backend.LdapBackend
LDAP URL	• ldaps://<ldap_server>:636 if using Secure LDAP • ldap://<ldap_server>:389 if not using encryption Note: If ldaps:// is specified in the LDAP URL, then do not set LDAP TLS.
Enable LDAP TLS	• TRUE if not using Secure LDAP (LDAPS) but want to establish a secure connection using TLS • FALSE if using LDAPS or not encrypting
LDAP Server CA Certificate	/path_to_certificate/cert.pem
LDAP Search Base	DC=mycompany,DC=com
LDAP Bind User Distinguished Name	username@domain
LDAP Bind Password	bind_user_password
Use Search Bind Authentication	TRUE
Create LDAP users on login	TRUE

Note: To encrypt with TLS, set LDAP URL to ldap://<ldap_server>:389 and check Enable LDAP TLS. For a proof of concept without encryption, use ldap://<ldap_server>:389, remove the value for LDAP Server CA Certificate, and uncheck Enable LDAP TLS.

4. You can optionally improve search performance with attributes and filters:

LDAP User Filter	objectclass=user (default = *)
LDAP Username Attribute	sAMAccountName (AD default), uid (LDAP default)
LDAP Group Filter	objectclass=group (default = *)
LDAP Group Name Attribute	cn (default)
LDAP Group Membership Attribute	member (default)

Note: With the user settings in the table above, the LDAP search filter has the form: (&(objectClass=user)(sAMAccountName=<user entered username>)).

5. Add any valid user and/or valid group to quickly test your LDAP configuration:

LDAP Username for Test LDAP Configuration	Any valid user
LDAP Group Name for Test LDAP Configuration	Any valid group

6. Click Save Changes.
7. Test your LDAP configuration, and when successful click Restart Hue.

Note: The syntax of Bind Distinguished Name differs per bind method:

- Search Bind: username@domain
- Direct Bind with NT Domain: username
- Direct Bind with Username Pattern: DN string (full DIT path)

Do not use if anonymous binding is supported.

You can test ldapsearch at the command line as follows:

```
LDAPTLS_CACERT=/<path_to_cert>/<ca_certificate> ldapsearch -H ldaps://<l
dap_server>:636 \
```

```
-D "<bind_dn>" -w <bind_password> -b <base_dn> "samaccountname=<user>"
```

Note: To run ldapsearch with a CA certificate, you may need to install ldap_utils on Debian/Ubuntu and openldap-clients on RHEL/CentOS.

Configuring authentication with LDAP and Direct Bind

To authenticate with Direct Binding, Hue needs either the User Principal Name (UPN) for Active Directory, or the full path to the LDAP user in the Directory Information Tree (DIT) for open standard LDAP.

About this task

Important: Direct binding only works with one domain. For multiple directories, use search bind.

Video: Authenticate Hue with LDAP and Direct Bind

Figure 2: Video: Authenticate Hue with LDAP and Direct Bind

To directly bind to an Active Directory/LDAP server with NT domain:

Procedure

1. Log in to Cloudera Manager and click Hue.
2. Click the Configuration tab and filter by scope=Service-wide and category=Security.
3. Set the following LDAP properties:

Authentication Backend	desktop.auth.backend.LdapBackend
LDAP URL	• ldaps://<ldap_server>:636 if using Secure LDAP • ldap://<ldap_server>:389 if not using encryption Note: If ldaps:// is specified in the LDAP URL, then do not set LDAP TLS.
Enable LDAP TLS	• TRUE if not using Secure LDAP (LDAPS) but want to establish a secure connection using TLS • FALSE if using LDAPS or not encrypting
LDAP Server CA Certificate	/path_to_certificate/cert.pem
LDAP Search Base	DC=mycompany,DC=com
LDAP Bind User Distinguished Name	<username> Only the username is required for Direct Bind. There is no need to specify the domain.
LDAP Bind Password	bind_user_password
Active Directory Domain	<your NT domain>
Use Search Bind Authentication	FALSE
Create LDAP users on login	TRUE

4. Click Save Changes

5. Test your LDAP configuration, and when successful, click Restart Hue.

   To directly bind to an open standard LDAP server with a username pattern:

   a. Remove the value for the Active Directory Domain.
   b. Set both LDAP Username Pattern and LDAP Bind User Distinguished Name to a DN string that represents the full path of the directory information tree, from UID to top level domain.

   Note: When using Direct Bind, set the LDAP Search Base property. This is not for authentication because you can log in to Hue without it, but to synchronize Hue with the LDAP server.

Configuring Hue for authentication against multiple LDAP/Active Directory servers

Some organizations have more than one LDAP or Active Directory (AD), and users in one LDAP/AD may not be present in the other. Hue supports the ability to authenticate users against multiple LDAP/AD servers.

Before you begin

Before attempting LDAP authentication against multiple servers, ensure you have configured LDAP synchronization for each server, as described in the previous sections. As long as users and groups are synchronised across LDAP and Hue, authentication should work.

Procedure

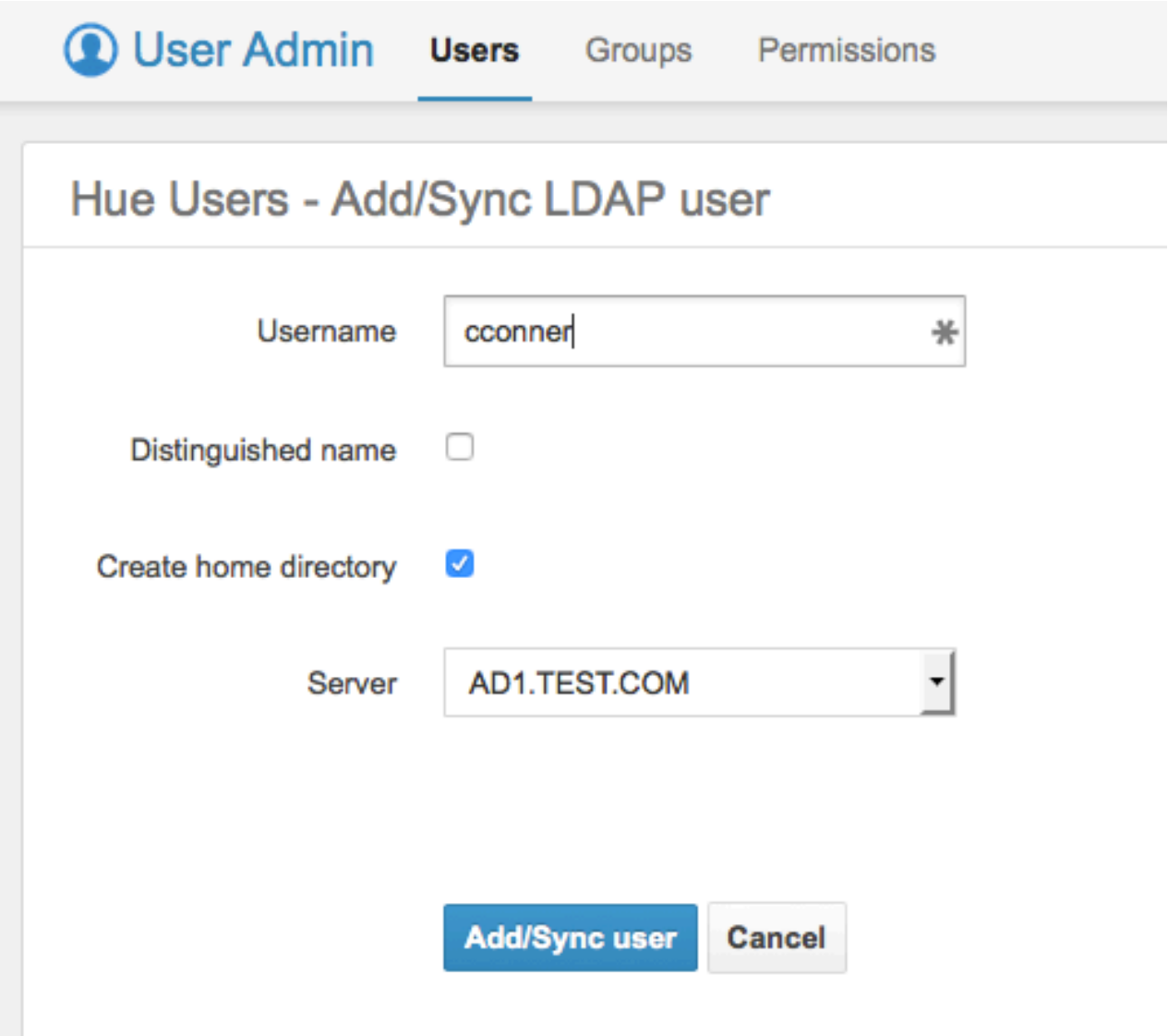
1. Log in to Cloudera Manager as an Administrator.
2. Go to Clusters Hue service Configuration Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini , and add the following lines for each LDAP server:

```
[desktop]
[[ldap]]
[[[ldap_servers]]]

[[[[AD1.TEST.COM]]]]
ldap_url=ldap://w2k8-ad1
search_bind_authentication=true
create_users_on_login=true
base_dn="cn=users,dc=ad1,dc=test,dc=com"
bind_dn="cn=Administrator,cn=users,dc=ad1,dc=test,dc=com"
bind_password="[***PASSWORD***]"
[[[[AD2.TEST.COM]]]]
ldap_url=ldap://w2k8-ad2
search_bind_authentication=true
create_users_on_login=true
base_dn="cn=users,dc=ad2,dc=test,dc=com"
bind_dn="cn=Administrator,cn
bind_password="[***PASSWORD***]"
```

   The above code snippet is for a demo LDAP server called "AD1.TEST.COM" and "AD2.TEST.COM".
3. Click Save Changes.
4. Restart the Hue service.

Results

When you log into Hue or while synchronizing LDAP users from the Hue web interface, you need to select the LDAP/AD server from the dropdown list.



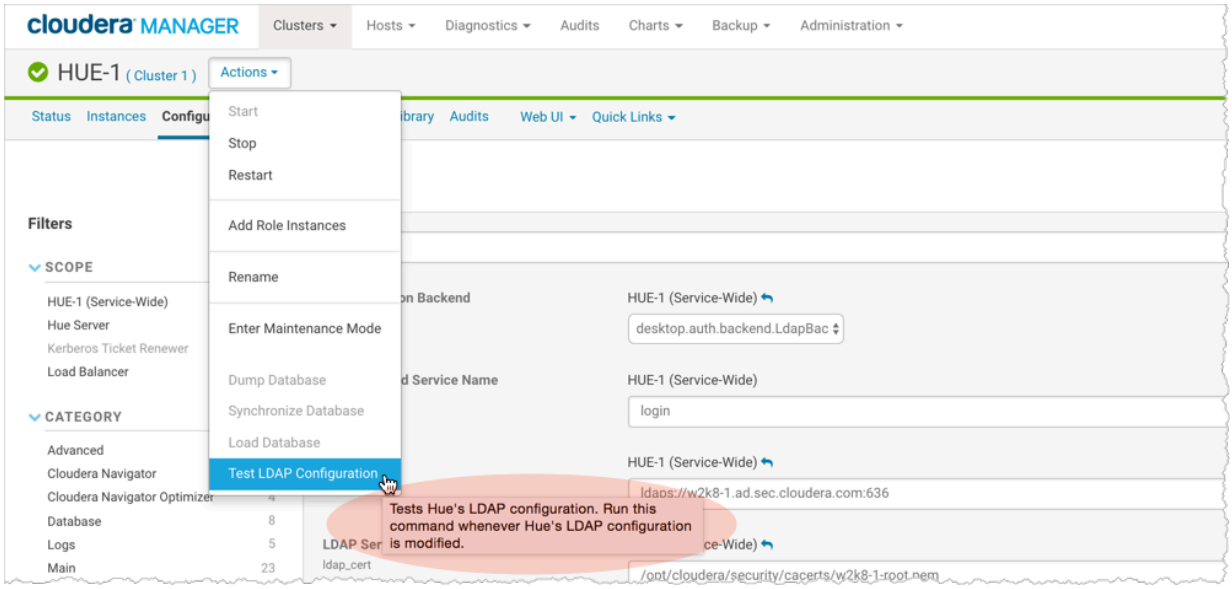
Testing the LDAP configuration

You can test your Hue LDAP configuration without restarting the Hue service. Add the values and save your changes.

Procedure

1. Configure Hue to authenticate with LDAP by using search bind or direct bind.
2. Add a user and group name for Test LDAP Configuration.
3. Click Save Changes.
4. Select ActionsTest LDAP Configuration.

5. Click Test LDAP Configuration:

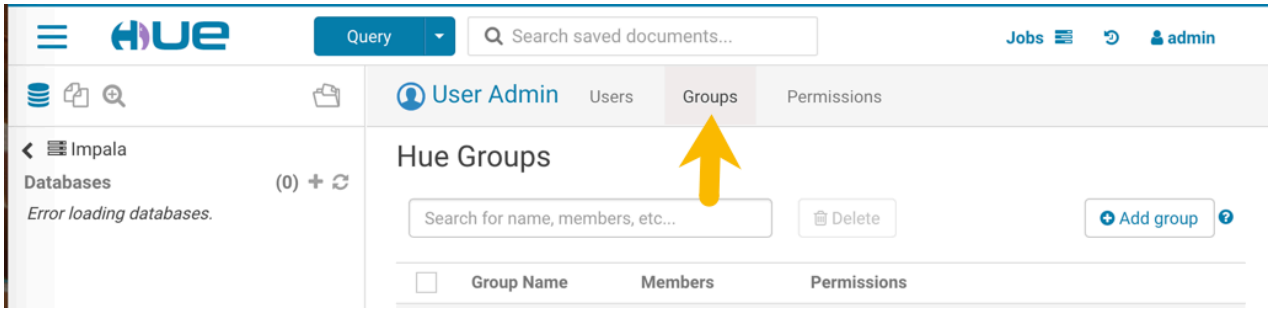


6. Click Restart Hue. When the test succeeds, log in to the Hue Web UI.

Configuring group permissions

You can configure permissions for members of groups on the Groups tab of the Hue User Admin application.

About this task



Note: A best practice is to remove all permissions from the default group and assign permissions as appropriate to your own groups.

Procedure

1. Log on to the Hue UI as a superuser.
2. Go to User AdminGroups.
3. Click the name of the group you want to alter.
4. Deselect any users that you do not want to change (all users in the group are selected by default).
5. Select or deselect the permissions you want to apply or remove.
6. Click Update Group.

Enabling LDAP authentication with HiveServer2 and Impala

LDAP authentication with HiveServer2 and Impala can be enabled by setting the auth_username and auth_password properties under the [beeswax] section for Hive and the [impala] section for Impala in a Cloudera Manager safety valve configuration property.

Important: Set these properties in the Cloudera Manager Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini property.

auth_username	LDAP username of the Hue user to be authenticated to the server.
auth_password	LDAP password of the Hue user to be authenticated to the server.

For example:

```
[beeswax]
   auth_username=<hiveserver2_ldap_user_name>
   auth_password=<hiveserver2_ldap_password>
[impala]
   auth_username=<impala_ldap_user_name>
   auth_password=<impala_ldap_password>
```

These login details are only used by Impala and Hive to authenticate to the LDAP server. The Impala and Hive services trust Hue to have already validated the user being impersonated instead of passing on the credentials.

LDAP properties

These are the properties you can use to configure LDAP for Hue in Cloudera Manager or in the hue.ini file for unmanaged clusters.

Property Name	Description and Syntax
General Hue LDAP Properties	
Authentication Backend backend	Authentication Mode. Select desktop.auth.backend.LdapBackend. Multiple backends are allowed. Create a list and add it to the Hue safety-valve.
LDAP URL ldap_url	URL for the LDAP server. Syntax: ldaps://<ldap_server>:<636> or ldap://<ldap_server>:<389> Important: To prevent usernames and passwords from transmitting in the clear, use ldap s:// or ldap:// + "Enable LDAP TLS".
Create LDAP users on login create_users_on_login	Flag to create new LDAP users at Hue login. If true, any user who logs into Hue is automatically created. If false, only users that exist in useradmin can log in.
Direct Bind Properties	
Active Directory Domain nt_domain	For direct binding with Microsoft Active Directory only. Typically maps to the user email address or ID in conjunction with the domain. Allows Hue to authenticate without having to follow LDAP references to other partitions. Hue binds with User-Principal-N ame (UPN) if provided. Example: ad.<mycompany>.com Important: Do not use nt_domain with LDAP Username Pattern or Search Bind.
LDAP Username Pattern ldap_username_pattern	For direct binding with LDAP (non-Active Directory) only (because AD uses UPNs which have a space in them). Username Pattern finds the user attempting to login into LDAP by adding the username to a predefined DN string. Use <username> to reference the user logging in. An example is "uid=<username>,ou=people,dc=mycompany,dc=com".
Search Bind Properties	
Use Search Bind Authentication search_bind_authentication	Flag to enable/disable Search Bind.
LDAP Search Base base_dn	Distinguished name to use as a search base for finding users and groups. Syntax: dc=ad, dc=sec, dc=mycompany,dc=com

Property Name	Description and Syntax
Encryption Properties	
LDAP Server CA Certificate ldap_cert	Full path to .pem file with Certificate Authority (CA) chain used to sign the LDAP server certificate. If left blank, all certificates are trusted and otherwise encrypted usernames and passwords are vulnerable to attack.
Enable LDAP TLS use_start_tls	Flag to enable/disable encryption with the StartTLS operation.
Import / Sync Properties	
LDAP Bind User Distinguished Name bind_dn	Bind user. Only use if LDAP/AD does not support anonymous binds. (Typically, LDAP supports anonymous binds and AD does not.) Bind User differs per auth type: • Search Bind: username@domain • Direct Bind with NT Domain: username • Direct Bind with Username Pattern: DN string (and same as LDAP Username Pattern)
LDAP Bind Password bind_password	Bind user password.
Filter Properties	
LDAP User Filter user_filter	General LDAP text search filter to restrict search of valid users. Only used by Search Bind authentication and LDAP Sync. The default is objectclass=* but can differ. For example, some LDAP environments support Posix objects for *nix authentication and the user filter might need to be objectclass=posixAccount.
LDAP Username Attribute user_name_attr	Username to search against (the attribute in LDAP that contains the username). Typical attributes include sAMAccountName (default for AD/LDAP) and uid (LDAP default). Maintain case sensitivity when setting attributes for AD/LDAP.
LDAP Group Filter group_filter	General LDAP text search filter to restrict search of valid groups. Only used by LDAP Sync (not authentication). If left blank, no filtering is used and all groups in LDAP are synced. The default is objectclass=* but can differ. For example, some LDAP environments support Posix objects for *nix authentication and the user filter might need to be objectclass=posixGroup.
LDAP Group Name Attribute group_name_attr	Group name to search against (the attribute in LDAP that contains the groupname). If left blank, the default is "cn" (common name), that typically works with AD/LDAP. Maintain case sensitivity when setting attributes for AD/LDAP.
LDAP Group Membership Attribute group_member_attr	Attribute in the group that contains DNs of all the members.(Optional) - If left blank, the default is "memberOf" or "member", that typically works with Active Directory/LDAP.
Test Properties	
LDAP Username for Test LDAP Configuration test_ldap_user	Any user (ideally with low privileges) used to verify the LDAP configuration.
LDAP Group Name for Test LDAP Configuration test_ldap_group	Any group (and not necessarily one that includes the test user) used to verify the LDAP configuration.

Configuring LDAP on unmanaged clusters

If your clusters are not managed with Cloudera Manager, you must manually set the LDAP configuration properties in the hue.ini file.

Refer to the following examples of LDAP configurations in the hue.ini file:

Example of a Search Bind configuration encrypted with LDAPS:

```
[[custom]]
[[auth]]
backend=desktop.auth.backend.LdapBackend

[[ldap]]
ldap_url=ldaps://<hostname>.ad.sec.<domain_name>.com:636
search_bind_authentication=true
ldap_cert=/<path_to_cacert>/<cert_filename>.pem
use_start_tls=false
create_users_on_login=true
base_dn="DC=ad,DC=sec,DC=<domain_name>,DC=com"
bind_dn="<username>@ad.sec.<domain_name>.com"
bind_password_script=<path_to_password_script>/<script.sh>
test_ldap_user="testuser1"
test_ldap_group="testgroup1"

[[[users]]]
user_filter="objectclass=user"
user_name_attr="sAMAccountName"

[[[groups]]]
group_filter="objectclass=group"
group_name_attr="cn"
group_member_attr="member"
```

Example of a Direct Bind configuration for Active Directory encrypted with LDAPS:

```
[[ldap]]
ldap_url=ldaps://<hostname>.ad.sec.<domain_name>.com:636
search_bind_authentication=false
nt_domain=ad.sec.<domain_name>.com
ldap_cert=/<path_to_cacert>/<cert_filename>.pem
use_start_tls=false
create_users_on_login=true
base_dn="DC=ad,DC=sec,DC=<domain_name>,DC=com"
bind_dn="<username>"
bind_password_script=<path_to_password_script>/<script.sh>
...
```

Example of a Direct Bind configuration for Active Directory encrypted with StartTLS:

```
[[ldap]]
ldap_url=ldap://<hostname>.ad.sec.<domain_name>.com:389
search_bind_authentication=false
nt_domain=ad.sec.<domain_name>.com
ldap_cert=/opt/cloudera/security/cacerts/<cert_filename>.pem
use_start_tls=true
create_users_on_login=true
base_dn="DC=ad,DC=sec,DC=<domain_name>,DC=com"
bind_dn="<username>"
bind_password_script=<path_to_password_script>/<script.sh>
...
```

Authenticating Hue users with SAML

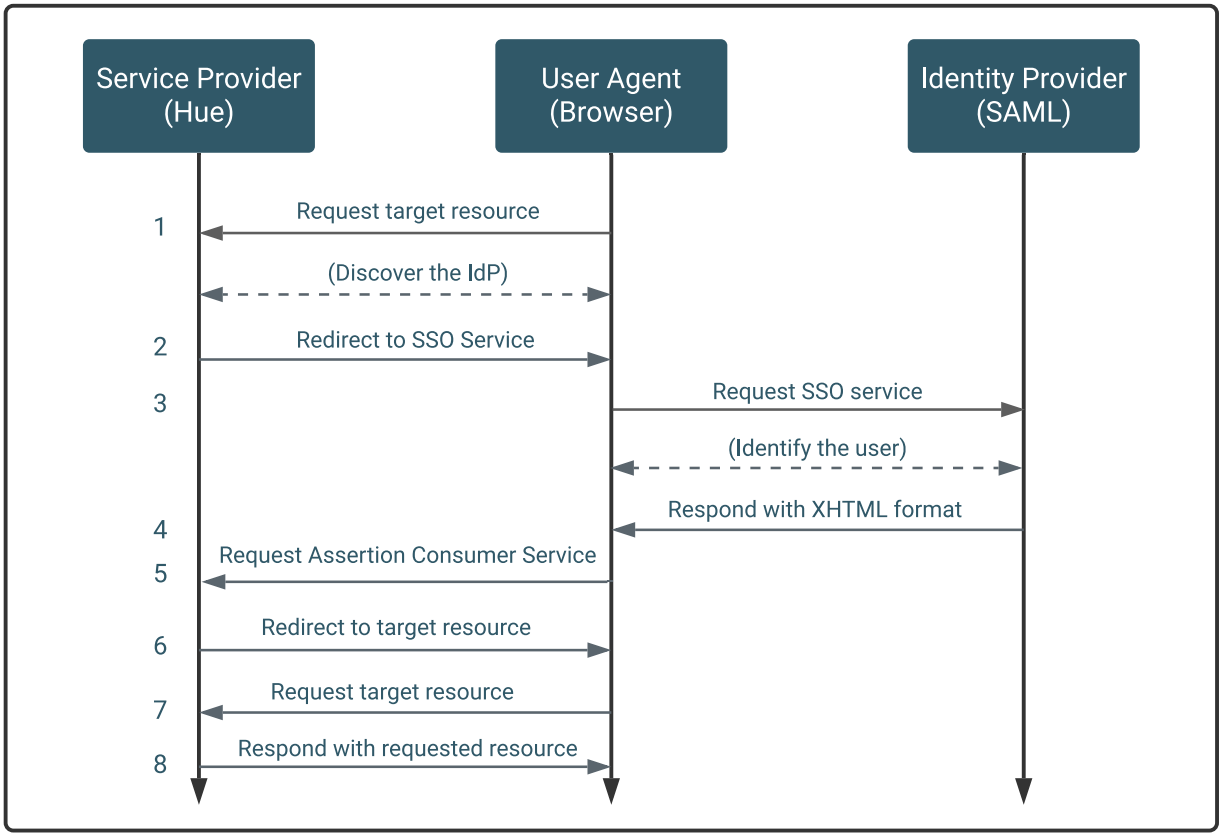
Hue supports SAML (Security Assertion Markup Language) for Single Sign-on (SSO) authentication.

The SAML 2.0 Web Browser SSO profile has three components:

- User Agent - Browser that represents you, the user, seeking resources.
- Service Provider (SP) - Service (Hue) that sends authentication requests to SAML.
- Identity Provider (IdP) - SAML service that authenticates users.

When a user requests access to an application, the Service Provider (Hue) sends an authentication request from the User Agent (browser) to the identity provider. The identity provider authenticates the user, sends a response, and redirects the browser back to Hue as shown in the following diagram:

Figure 3: SAML SSO protocol flow in a web browser



The Service Provider (Hue) and the identity provider use a metadata file to confirm each other's identity. Hue stores metadata from the SAML server, and the identity provider stores metadata from the Hue server.

Configuring SAML authentication on managed clusters

To configure Hue for SAML authentication on managed clusters, you must add the SAML authentication properties to the Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini in Cloudera Manager.

Before you begin

These instructions assume that you have an Identity Provider set up and running. You can use any identity provider of your choice. For example, Okta, Ping Identity, and OpenAM.

Procedure

1. Log on to Cloudera Manager and go to HueConfiguration.
2. In the search text box, enter hue_safety_valve.ini to locate the Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini.
3. Enter the SAML parameters into the Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini text box. For example:

```
## Example Settings using Open AM:
[desktop]
redirect_whitelist="^\/.*$,^https:\/\/idp.example.com:8080\/.*$"
[[auth]]
backend=libsaml.backend.SAML2Backend
[libsaml]
want_response_signed=True
want_assertions_signed=True
xmlsec_binary=/usr/bin/xmlsec1
metadata_file=/opt/cloudera/security/saml/idp-metadata.xml
key_file=/opt/cloudera/security/saml/host.key
cert_file=/opt/cloudera/security/saml/host.pem
key_file_password=Config(
  key="key_file_password",
  help=_t("key_file_password password of the private key"),
  default=None) ## If using encrypted private key
username_source=nameid
name_id_format="urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified"
entity_id=[***HOST-BASE-NAME***]
logout_enabled=false
```

   If you are using an encrypted private key file, then you must specify the password in the key_file_password property. Or you can use an unencrypted private key file.

   To create an unencrypted private key file from an encrypted key:

   a. SSH into a terminal as a root user.
   b. Change to the directory where you have stored the ssl certificate key.
   c. Run the following command:

```
openssl rsa -in ssl_certificate.key -out ssl_certificate-nocrypt.key
```

   d. When prompted, enter the password that you use to access the ssl_certificate.key file.

      The output file (ssl_certificate-nocrypt.key) is an unencrypted PEM-formatted key.

   Note: For SLES distributions, the xmlsec binary may be located in /usr/local/bin/. If so:

   - Set xmlsec_binary=/usr/local/bin/xmlsec1 in the Hue Service Advanced Configuration Snippet.
   - Set LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/lib/ in the Hue Service Environment Advanced Configuration Snippet.

4. Click Save Changes, then select, ActionsRestart Hue.

Manually configuring SAML authentication

To manually configure Hue for SAML authentication on unmanaged clusters, you must add GCC Python libraries and install xmlsec1 tools on all the hosts in your cluster.

Before you begin

These instructions assume that you have an Identity Provider set up and running. You can use any identity provider of your choice. For example, Okta, Ping Identity, and OpenAM.

Important: You may need to disable cipher algorithms before manually configuring Hue for SAML authentication.

Procedure

1. Install the following libraries on all hosts in your cluster:

```
## RHEL/CentOS
yum install git gcc python-devel swig openssl
```

```
## Ubuntu/Debian
apt-get install git gcc python-dev swig openssl
```

```
## SLES
zypper install git gcc python-devel swig openssl make libxslt-devel libl
tdl-devel
```

2. Install xmlsec1 and xmlsec1-openssl on all hosts in the cluster:

Important: Ensure that the xmlsec1 package is executable by the user, hue.

```
## RHEL/CentOS
yum install xmlsec1 xmlsec1-openssl
```

Note: If xmlsec libraries are not available, use the appropriate epel repository:

```
## For RHEL/CentOS 7
wget http://dl.fedoraproject.org/pub/epel/7/x86_64/e/epel-relea
se-7-6.noarch.rpm
rpm -ivh epel-release-7-6.noarch.rpm
```

```
## Ubuntu/Debian
apt-get install xmlsec1 libxmlsec1-openssl
```

```
## SLES (get latest version)
wget http://www.aleksey.com/xmlsec/download/xmlsec1-1.2.24.tar.gz
tar -xvzf xmlsec1-1.2.24.tar.gz
cd xmlsec1-1.2.24
./configure && make
make install
```

3. Copy metadata from your identity provider's SAML server and save it as an XML file on every host with a Hue server. For example, if your identity provider is Shibboleth, visit https://<idp_host>:8443/idp/shibboleth, copy the metadata content, and paste it into an XML file. Read the documentation of your identity provider for details on how to copy the XML of the SAML server metadata.

Note: You may have to edit the copied metadata; for example, the identity provider's port number (8443) might be missing from its URL.

```
mkdir -pm 755 /opt/cloudera/security/saml/
cd /opt/cloudera/security/saml/
```

```
vim idp-<your idp provider>-metadata.xml
# Paste IdP SAML here and save
```

4. Add the files that the key_file and cert_file SAML properties point to for encrypted assertions and make sure you add this properties to the hue.ini configuration file.
   - The key_file parameter points to the location of the private key that is used to encrypt metadata. Its file format must be <file_name>.PEM.
   - The cert_file parameter points to the location of the X.509 certificate that is sent with encrypted metadata. Its file format must be <file_name>.PEM.

Warning:

Add the key and cert files even if you are not encrypting assertions. Hue checks for the existence and validity of these files even if they are not needed. They cannot be empty files. This is a known issue. If necessary, create a valid self-signed certificate:

```
openssl req -x509 -newkey rsa:2048 -sha256 -days 3560 -nodes -keyout
 host.key -out host.pem -subj '/CN=Hue SAML'
```

Integrating your identity provider's SAML server with Hue

After Hue is configured for SAML authentication and restarted, copy the metadata that is generated by the Hue server and send it to your identity provider so they can configure the SAML server.

Before you begin

Ensure that you have configured Hue for SAML authentication and restarted it before you integrate your identity provider's SAML server with Hue.

Procedure

1. Ensure Hue is configured, restarted, and running.
2. Go to http://<hue_fqdn>:8889/saml2/metadata.
3. Copy the metadata and send it to your identity provider.
4. Ensure that your identity provider configures the SAML server with the Hue metadata. It is the same process you used to configure the Hue server with SAML metadata.

SAML properties

These SAML properties can be set in the hue.ini file for unmanaged clusters. A subset of them can be set in the Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini for managed clusters.

Table 2: Table of SAML parameters

SAML parameter	Description
authn_requests_signed	Boolean, that when True, signs Hue-initiated authentication requests with X.509 certificate.
backend	Hard-coded value set to SAML backend library packaged with Hue (libsaml.backend.SAML2Backend).
base_url	URL that SAML Identity Provider uses for responses. Typically used in Load balanced Hue environments.
cert_file	Path to X.509 certificate sent with encrypted metadata. File format must be .PEM.
create_users_on_login	Boolean, that when True, creates users from OpenId, upon successful login.
entity_id	Service provider ID. Can also accept pattern where '<base_url>' is replaced with server URL base.
key_file	Path to private key used to encrypt metadata. File format must be .PEM.
key_file_password	Password used to decrypt the X.509 certificate in memory.
logout_enabled	Boolean, that when True, enables single logout.
logout_requests_signed	Boolean, that when True, signs Hue-initiated logout requests with an X.509 certificate.

SAML parameter	Description
metadata_file	Path to readable metadata XML file copied from Identity Provider.
name_id_format	Format of NameID that Hue requests from SAML server.
optional_attributes	Comma-separated list of optional attributes that Hue requests from Identity Provider.
required_attributes	Comma-separated list of required attributes that Hue requests from Identity Provider. For example, uid and email.
redirect_whitelist	Fully qualified domain name of SAML server: "^\/.*$,^https:\/\/<SAML_server_FQDN>\/.*$".
user_attribute_mapping	Map of Identity Provider attributes to Hue django user attributes. For example, {'uid':'username', 'email':'email'}.
username_source	Declares source of username as nameid or attributes.
want_response_signed	A boolean parameter, when set to True, requires SAML response wrapper returned by an IdP to be digitally signed by the IdP. The default value is False.
want_assertions_signed	A boolean parameter, when set to True, requires SAML assertions returned by an IdP to be digitally signed by the IdP. The default value is False.
xmlsec_binary	Path to xmlsec_binary that signs, verifies, encrypts/decrypts SAML requests and assertions. Must be executable by user, hue.

SAML properties that can be set for managed clusters

- redirect_whitelist [desktop]

  Set to the fully qualified domain name of the SAML server so that Hue can redirect to the SAML server for authentication.

  ```
  [desktop]
  redirect_whitelist=^\/.$,^https:\/\/<SAML_server_fully_qualified_domain_name>\/.$
  ```

  Note: Hue uses redirect_whitelist to protect itself from redirecting to unapproved URLs.

- backend [desktop]>[[auth]]

  Point to the SAML backend that is packaged with Hue:

  ```
  backend=libsaml.backend.SAML2Backend
  ```

- xmlsec_binary [libsaml]

  Point to the xmlsec1 library path:

  ```
  xmlsec_binary=/usr/bin/xmlsec1
  ```

  Note: To find the path, run: which xmlsec1

- metadata_file [libsaml]

  Point to the path of the XML file you created from the identity provider's metadata:

  ```
  metadata_file=/path/to/<your_idp_metadata_file>.xml
  ```

- key_file and cert_file [libsaml]

  To encrypt communication between Hue and the Identity Provider (IdP), you need a private key and certificate. The private key signs requests sent to the IdP, and decrypts messages from the IdP. The certificate is used to encrypt messages to Hue from the IdP, and must be provided to the IdP. Typically, the cert_file is shared by

providing Hue's Service Provider metadata XML to the IdP admins, but you may also share a copy of the cert_file itself.

The SAML certificate and private key must be the same on all Hue Server hosts, and can be self-signed, obtained from a commercial CA vendor, or from your internal PKI administrators. Both key_file and cert_file must be in PEM format.

Users with password-protected certificates can set the property, key_file_password in the hue.ini file. Hue uses the password to decrypt the SAML certificate in memory and passes it to xmlsec1 through a named pipe. The decrypted certificate never touches the disk. This only works for POSIX-compatible platforms.

Troubleshooting SAML authentication

Before troubleshooting your SAML authentication configuration in Hue, enable DEBUG for the Hue Django logs that are located in /var/log/hue. In the Hue Web UI, go to the Home page, select Server Logs, and check Force Debug Level. For managed clusters, you can use Cloudera Manager to enable DEBUG by navigating to the Hue service, selecting the Configuration tab, check Enable Django Debug mode, click Save Changes, and then Restart.

SAML SSL error

OpenSSL might fail with this message:

```
SSLError: [Errno bad handshake] [('SSL routines', 'SSL3_CHECK_CERT_AND_ALGOR
ITHM', 'dh key too small')]
```

To resolve, append the following code to the file, /usr/java/<your_jdk_version>-cloudera/jre/lib/securi ty/java.security:

```
jdk.tls.disabledAlgorithms=MD5, RC4, DH
```

SAML decrypt error

The following error is an indication that you are using a slightly different SAML protocol from what Hue expects:

```
Error: ('failed to decrypt', -1)
```

To resolve:

1. Download and rename the fix-xmlsec1.txt Python script.

   ```
   wget https://www.cloudera.com/documentation/other/shared/fix-xmlsec1.txt -
   O fix-xmlsec1.py
   ```

2. Change permissions as appropriate, for example:

   ```
   chmod 755 fix-xmlsec1.py
   ```

3. In hue.ini, set xmlsec_binary=<path_to_script>/fix-xmlsec1.py.
4. Run fix-xmlsec1.py.

This script repairs the known issue whereby xmlsec1 is not compiled with RetrievalMethod and cannot find the location of the encrypted key. SAML2 responses would sometimes place EncryptedKey outside of the EncryptedData tree. This script moves EncryptedKey under EncryptedData.

Authenticating Hue users with Knox SSO

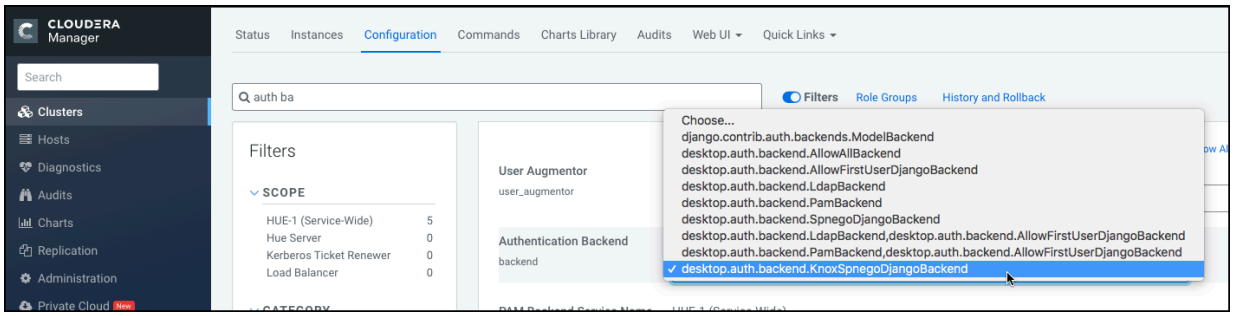
You can use the Apache Knox Gateway to interact with Hue REST APIs and the Hue user interface, along with other CDP components and services. To set up Knox Single Sign-on (SSO) to authenticate users, you must configure the KnoxSpnegoDjangoBackend property using Cloudera Manager.

Before you begin

To authenticate users using Knox SSO, you must have Knox installed on your CDP cluster, also known as a secure cluster.

Procedure

1. Sign in to Cloudera Manager as an Administrator.
2. Go to Clusters Hue service Configurations and search for the Authentication Backend field.
3. Select desktop.auth.backend.KnoxSpnegoDjangoBackend from the dropdown.

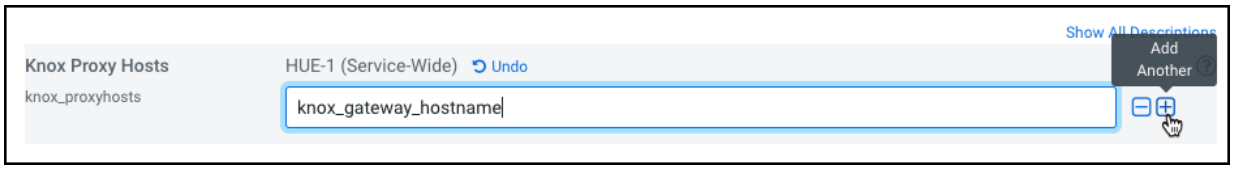


4. Click Save Changes.
5. Go to Clusters $Knox service Instances and note down the hostnames of the Knox Gateways.

   You must provide these details in the next step.

   If you have set up Knox in High-Availablity (HA) mode, then you can see more than one Knox Gateways listed on the Instances tab.
6. Go back to Clusters Hue service Configurations and search for the Knox Proxy Hosts field.
7. Enter the hostname of the Knox Gateway that you noted earlier.

   If you have set up Knox HA, then click + to add another hostname.



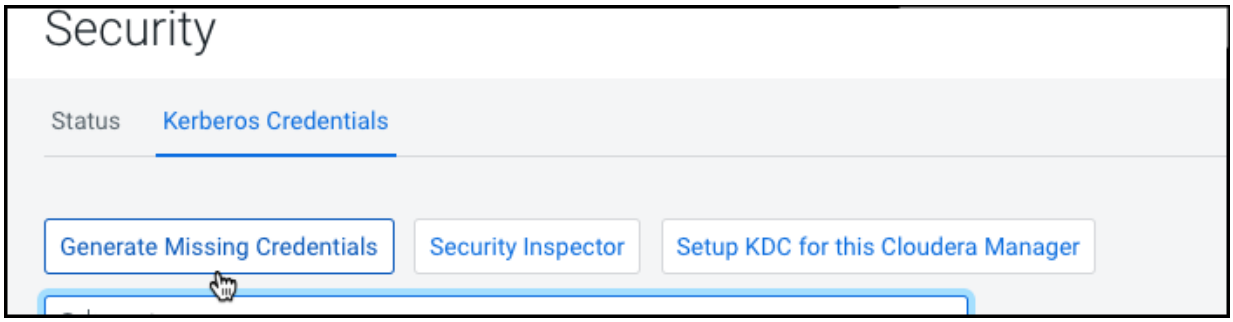
8. Click Save Changes.

   You would see the following warning:

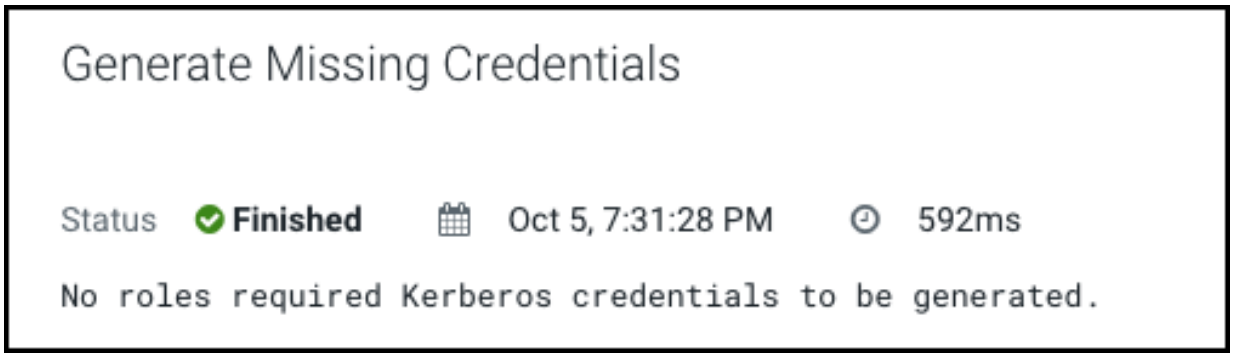
```
Role is missing Kerberos keytab. Go to the Kerberos Credentials page and
 click the Generate Missing Credentials button.
```

9. Click Administration on the Cloudera Manager left navigation panel and select Security.

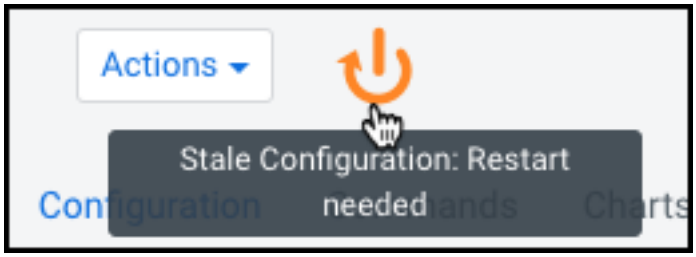
10. Go to the Kerberos Credentials tab and click Generate Missing Credentials.



A pop-up showing the status is displayed.



11. Go to Clusters Hue service and click Restart next to Actions.



12. On the **Stale Configurations** page, click Restart Stale Services.

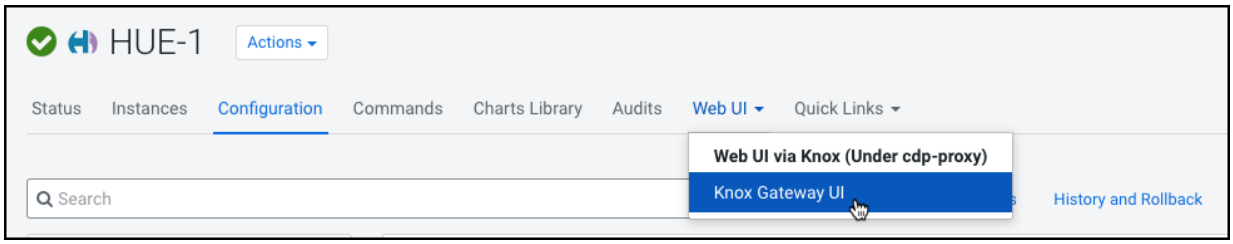
    The **Restart Stale Services** wizard is displayed.

13. On the Review Changes page, select Redeploy client configuration, and click Restart Now.

    The **Command Details** page shows the live status as the service restarts.

    When all the steps are complete, click Finish.

14. From the Hue service page, click Web UI Knox Gateway UI .

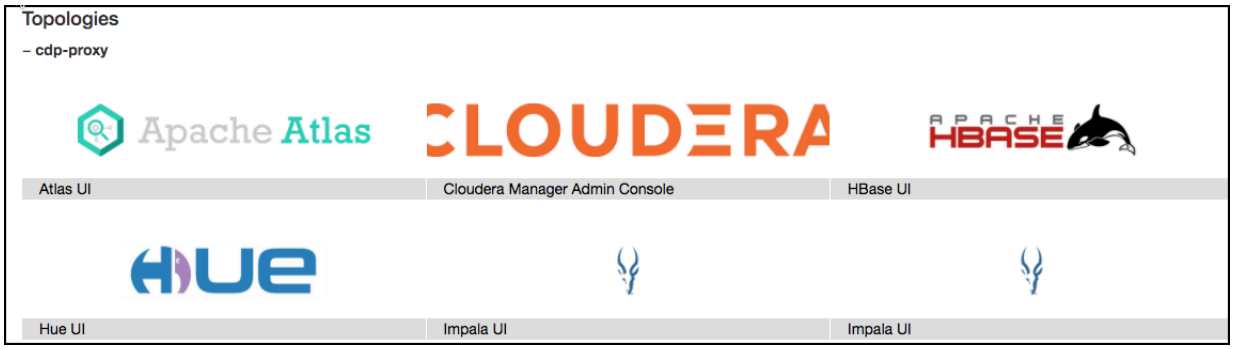


    The Knox Gateway UI is displayed.

15. On the **General Proxy Information** page, expand the CDP Proxy topology by clicking + cdp-proxy under Topologies.

    The list of services that are configured with the cdp-proxy topology is displayed.

16. Click on the Hue logo.



You should be able to log in to the Hue web UI.

You can also log into Hue using the following URL:

```
https://[***HOSTNAME***]:[***PORT***]/gateway/cdp-proxy/hue/
```

Applications and permissions reference

Hue is a web-based UI for several cluster services that you can access by using Hue applications and their associated permissions.

Hue applications

These CDP services are available in Hue. Currently, Spark is only available in the upstream version of Hue.

Hue Application	Application Dependencies
HBase	HBase Browser
HDFS	Core, File Browser
Hive	Metastore Tables, Hive Editor
Impala	Metastore Tables, Impala Editor
MapRed / YARN	Job Browser, Job Designer, Oozie, Hive Editor, Pig, Sqoop
Oozie	Job Designer, Oozie Editor/Dashboard
Solr (Search)	Hadoop Security
Spark	Spark

Hue application permissions

Hue application permissions are composed of name.permission:action.

For example, filebrowser.access:Launch this application(3)

In this example:

- filebrowser = Hue application name
- access = Execute permissions
- Launch this application = Action that is enabled
- (3) = Process ID of the filebrowser application in the Hue database

Hue Application	Permission	Read/Write/Execute	Action Description
about	access	execute	Launch this application

Hue Application	Permission	Read/Write/Execute	Action Description
beeswax	access	execute	Launch this application
dashboard	access	execute	Launch this application
filebrowser	access	execute	Launch this application
filebrowser	s3_access	execute	Access to S3 from filebrowser and filepicker
filebrowser	adls_access	execute	Access to ADLS from filebrowser and filepicker
filebrowser	abfs_access	execute	Access to ABFS from filebrowser and filepicker
filebrowser	gs_access	execute	Access to GS from filebrowser and filepicker
help	access	execute	Launch this application
hive	access	execute	Launch this application
impala	access	execute	Launch this application
indexer	access	execute	Launch this application
jobbrowser	access	execute	Launch this application
jobsub	access	execute	Launch this application
kafka	access	execute	Launch this application
metadata	access	execute	Launch this application
metadata	write	write	Allow edition of metadata like tags
metastore	access	execute	Launch this application
metastore	write	write	Allow DDL operations. Need the app access too
notebook	access	execute	Launch this application
oozie	access	execute	Launch this application
oozie	dashboard_jobs_access	execute	Oozie Dashboard read-only user for all jobs
oozie	disable_editor_access	execute	Disable Oozie Editor access
proxy	access	execute	Launch this application
rdbms	access	execute	Launch this application
search	access	execute	Launch this application
useradmin	access_view:useradmin:edit_user	read/write/execute	Access to profile page on User Admin
useradmin	access_view:useradmin:view_user	read/write/execute	Access to any profile page on User Admin
useradmin	access	execute	Launch this application

Securing Hue passwords with scripts

You can secure passwords in Hue by using one consolidated script, or multiple individual scripts. Hue runs each password script at startup and extracts passwords from stdout.

About this task

Store scripts in a directory that only Hue can read, write, and execute. You can choose password script names but you cannot change hue.ini property names to which you assign those scripts. Add the suffix _script to any password property and set it equal to the script name.

Procedure

1. At the command line, create one or more password scripts. For example, create a consolidated script named my_passwords_script.sh:

```
#!/bin/bash
SERVICE=$1
if [[ ${SERVICE} == "ldap_password" ]]
then
 echo "<your_ldap_password>"
fi

if [[ ${SERVICE} == "ssl_password" ]]
then
 echo "<your_ssl_password>"
fi

if [[ ${SERVICE} == "bind_password" ]]
then
 echo "<your_bind_password>"
fi

if [[ ${SERVICE} == "db_password" ]]
then
 echo "<your_database_password>"
fi
```

2. Log on to Cloudera Manager and go to HueConfiguration.
3. Search on Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini.
4. Add script properties. In the following example, the required _script is added to the password property:

```
[desktop]
ldap_username=hueservice
ldap_password_script="/var/lib/hue/password_script.sh ldap_password"
ssl_password_script="/var/lib/hue/password_script.sh ssl_password"

[[ldap]]
bind_password_script="/var/lib/hue/password_script.sh bind_password"
[[database]]
db_password_script="/var/lib/hue/password_script.sh db_password"
```

5. Click Save Changes and Restart Hue.

Configuring TLS/SSL for Hue

You can independently enable TLS/SSL for Hue.

Cloudera recommends that your cluster and the Hue service use Kerberos for authentication. If you enable TLS/SSL for a cluster that has not been configured to use Kerberos, a warning is displayed. You should integrate the cluster with your Kerberos deployment before proceeding.

Creating a truststore file in PEM format

Server certificates are stored in Java KeyStore (JKS) format and must be converted to Privacy Enhanced Mail (PEM) format. You must create a PEM file before configuring Hue as a TLS/SSL client or a TLS/SSL server.

About this task

To create the Hue truststore, extract each certificate from its keystore with the Java keytool, convert the certificate to PEM format with the OpenSSL.org openssl tool, and then add it to the Hue truststore:

Procedure

1. Extract the certificate from the keystore of each TLS/SSL-enabled server with which Hue communicates.
   For example, if you have hadoop-server.keystore that contains a server certificate, foo-1.example.com with a password of example123, you would use the following keytool command:

   ```
   keytool -exportcert -keystore hadoop-server.keystore -alias foo-1.exampl
   e.com -storepass example123 -file foo-1.cert
   ```

2. Convert each certificate into a PEM file. Here is what the openssl tool command looks like for the foo-1.cert file that was extracted in Step 1:

   ```
   openssl x509 -inform der -in foo-1.cert > foo-1.pem
   ```

3. Concatenate all the PEM certificates you extracted and converted from the server truststore into one PEM file:

   ```
   cat foo-1.pem foo-2.pem foo-n.pem ... > hue_truststore.pem
   ```

   Concatenate the certificate files in the following order: SSL certificate followed by intermediate certificate, followed by the root CA certificate.

   Important: Ensure the final PEM truststore is deployed in a location that is accessible by the Hue service.

Configuring Hue as a TLS/SSL client

Hue acts as a TLS/SSL client when communicating with other services, such as core Hadoop, HBase, Oozie, and cloud providers like Amazon S3 or Azure.

To act as a TLS/SSL client, Hue must authenticate HDFS, MapReduce, YARN daemons, the HBase Thrift server, and so on. To do this, Hue needs to have the certificate chains of these components' hosts in the Hue trust store.

The Hue truststore is a single PEM (Privacy Enhanced Mail) file that contains the certificate authority (CA) root certificate and all intermediate certificates to authenticate the certificate installed on each TLS/SSL-enabled server. These servers host the services with which Hue communicates.

Note: A certificate is specific to a host. It is signed by a CA and tells the requesting client, which is Hue in this case, that the host is the same one as is represented by the host public key. Hue uses a chain of signing authority in its truststore to validate the CA that signed the host certificate.

Enabling Hue as a TLS/SSL client

After you create a Hue truststore file in PEM format, you can configure Hue as a TLS/SSL client by using Cloudera Manager.

Procedure

1. Log in to Cloudera Manager as an Administrator.
2. Go to Clusters Hue service Configuration Hue TLS/SSL Server CA Certificate (PEM Format) ssl_cacerts and add the path to the *hue_truststore*.pem file on the host that is running the Hue web server.
3. Click Save Changes.
4. Restart the Hue service.

Configuring Hue as a TLS/SSL server

Hue and other Python-based services expect certificates and keys to be stored in PEM (Privacy Enhanced Mail) format.

Before you enable TLS/SSL for the Hue server, you must generate a private key and certificate by using the openssl command-line tool and reuse a host's existing Java keystore by converting it to the PEM format.

Enabling Hue as a TLS/SSL server using Cloudera Manager

You can use Cloudera Manager to enable TLS/SSL for the Hue server.

Procedure

1. Log in to Cloudera Manager as an Administrator.
2. Go to Clusters Hue service Configuration and filter by SCOPE Hue Server and CATEGORY Security .
3. Edit the following Hue TLS/SSL properties according to your cluster configuration:
   - Enable TLS/SSL for Hue: Select the check box to encrypt communication between clients and Hue with TLS/SSL.
   - Hue TLS/SSL Server Certificate File (PEM Format) ssl_certificate: Specifies the path to the TLS/SSL certificate on the host that is running the Hue web server.

     Ensure that you include the complete chain in the ssl_certificate PEM file.

     The order of the certificates should be as follows from the top to bottom: server, intermediate, root.

     If there are multiple intermediate CA certificates, then you must add them in the correct order. For example:

     ```
     Subject: CN=Hue Server Certificate
     Issuer: CN=Intermediate 2

       Subject: CN=Intermediate 2
       Issuer: CN=Intermediate 1

         Subject: CN=Intermediate 1
         Issuer: CN=RootCA

           Subject: CN=RootCA
           Issuer: CN=RootCA
     ```
   - Hue TLS/SSL Server Private Key File (PEM Format) ssl_private_key: Specifies the path to the TLS/SSL private key on the host running the Hue web server.
   - Hue TLS/SSL Private Key Password ssl_password: Specifies the password for the private key in the Hue TLS/SSL Server Certificate and Private Key file.
   - Hue TLS/SSL Server CA Certificate (PEM Format) ssl_cacerts: Specifies the path to the TLS/SSL certificate authority root certificate on the host that is running the Hue web server.

4. Add the path to the certificate chain PEM file in [desktop] section of the Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini field:

```
[desktop]
ssl_certificate_chain=/[***PATH***]/[***TO***]/[***FULL-CHAIN***].pem
```

5. Click Save Changes.
6. Select ActionsRestart to restart the Hue service.

What to do next

Change the permissions for Hue to read the certificates after you have enabled TLS/SSL as follows:

```
chmod 644 [***PATH-WHERE-SSL-FILES-FOR-HUE-ARE-LOCATED***]
```

Enabling TLS/SSL for Hue Load Balancer

To configure the Hue Load Balancer to use HTTPS or operate as a TLS/SSL server, you need a self-signed SSL certificate and a private key file. If the private key file is password protected, then you must configure the Hue Load Balancer to use the corresponding key password.

Procedure

1. Log in to Cloudera Manager as an Administrator.
2. Go to Clusters Hue service Configuration Scope Load Balancer .
3. Enter the location of the file that contains the server certificate key for TLS/SSL on the host running Hue Load Balancer in the Hue Load Balancer TLS/SSL Server Certificate File (PEM Format) field.

   The certificate file must be in the Privacy-Enhanced Mail (PEM) format.
4. Enter the location of the TLS/SSL file that contains the private key used for TLS/SSL on the host running Hue Load Balancer, in the Hue Load Balancer TLS/SSL Server Private Key File (PEM Format) field.

   The certificate file must be in PEM format.
5. (Optional) If the private key file is password protected perform the following steps:
   a) 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.
   b) Set the file ownership and permissions as shown in the following example:

      ```
      chown hue:hue password.txt
      chmod 700 password.txt
      ```

   c) 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.
6. Click Save Changes.
7. Restart the Hue service.

Enabling TLS/SSL communication with HiveServer2

For Hue to communicate with HiveServer2 using TLS/SSL, Hue needs the Hive certificate and certificate chain.

To enable TLS/SSL communication with HiveServer2, add the following properties in the [beeswax] section under [[ssl]] in the Cloudera Manager Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini configuration property:

Property	Description

`[beeswax]`
`[[ssl]]`
`enabled`
	Valid values: true | false
	Enables or disables TLS/SSL communication for this server.
	Default setting: false
	Example: enabled=true

`[beeswax]`
`[[ssl]]`
`cacerts`
	Valid values: directory path
	Specifies the path to the Certificate Authority certificates.
	Default setting: /etc/hue/cacerts.pem
	Example: cacerts=/opt/cloudera/security/CAcerts/cacerts

`[beeswax]`
`[[ssl]]`
`validate`
	Valid values: true | false
	Specifies whether Hue validates certificates received from the server.
	Default setting: true
	Example: validate=true

Enabling TLS/SSL communication with Impala

For Hue to communicate with Impala using TLS/SSL, Hue needs the Impala certificate and certificate chain.

To enable TLS/SSL communication with Impala, add the following properties in the [impala] section under [[ssl]] in the Cloudera Manager Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini configuration property:

Property	Description

`[impala]`
`[[ssl]]`
`enabled`
	Valid values: true | false
	Enables or disables TLS/SSL communication for this server.
	Default setting: false
	Example: enabled=true

`[impala]`
`[[ssl]]`
`cacerts`
	Valid values: directory path
	Specifies the path to the Certificate Authority certificates.
	Default setting: /etc/hue/cacerts.pem
	Example: cacerts=/opt/cloudera/security/CAcerts/cacerts

`[impala]`
`[[ssl]]`
`validate`
	Valid values: true | false
	Specifies whether Hue validates certificates received from the server.
	Default setting: true
	Example: validate=true

Securing database connections with TLS/SSL

Hue uses different clients to communicate with each database internally. Client-specific options, such as secure connectivity can be configured using Cloudera Manager.

Procedure

1. Log in to Cloudera Manager as an administrator.

2. Go to Clusters Hue service Configuration and add the following section in the Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini field:

```
[desktop]
  [[database]]
    …
    options={"ssl":{"ca":"/tmp/ca-cert.pem"}}
```

This identifies the Certificate Authority (CA) certificate for the backend database. You can also identify public and private keys as follows:

```
options='{"ssl": {"ca": "/tmp/newcerts2/ca.pem", "key": "/tmp/newcerts2/
client-key.pem", "cert": “/tmp/newcerts2/client-cert.pem"}}'
```

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

Enforcing TLS version 1.2 for Hue

CDP Data Hub cluster components and services such as the Cloudera Manager web UI, the Hue web UI, and the Impala web UI communicate with each other using TLS 1.2 as the default TLS protocol, and TLS 1.1 or 1.0 if a client requests it. You can enforce these services to only use TLS 1.2 by specifying the SSL protocol in Cloudera Manager.

About this task

Note: The TLS version that is auto-applied depends on the Python version. If your installed Python version is higher than 2.7.9, then both the client and the server use the latest TLS. But if your installed Python version is lower than 2.7.9, then TLS 1.0 is used.

Note: The following steps do not apply to the connections between Hue and its backend database or external identity services such as LDAP and Active Directory.

Procedure

1. Sign in to Cloudera Manager as an Administrator.
2. Go to Clusters Hue service Configuration Load Balancers Advanced and add the following line in the SSL Protocol field:

```
-all +TLSv1.2
```

3. Click Save Changes.
4. Restart the Hue service.
5. Verify that TLS version 1.2 is used for encryption and all the ciphers used are “strong” by using a security scanner such as Nmap.
   a) Open a CLI console on a machine in your cluster.
   b) Run the following command:

```
nmap -sV --script +ssl-enum-ciphers -p 8889 [***HOSTNAME***] -f
```

Replace [***HOSTNAME***] with the actual name of the host.

The following is a sample output. It shows that only TLS 1.2 is available for the handshake and that all the ciphers are “strong”:

```
Starting Nmap 7.80 ( http://nmap.org ) at 2020-30-10 11:16 PDT
Nmap scan report for hostname.example.com (a.b.c.d)
```

```
Host is up (-1800s latency).
PORT STATE SERVICE VERSION
8889/tcp open ssl/http Apache httpd 2.4.6 ((CentOS) OpenSSL/1.0.2k-fips)
| ssl-enum-ciphers:
| SSLv3: No supported ciphers found
| TLSv1.2:
| ciphers:
| TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA - strong
| TLS_DHE_RSA_WITH_AES_128_CBC_SHA - strong
| TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 - strong
| TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 - strong
| TLS_DHE_RSA_WITH_AES_256_CBC_SHA - strong
| TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 - strong
| TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 - strong
| TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA - strong
| TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA - strong
| TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 - strong
| TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 - strong
| TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA - strong
| TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 - strong
| TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 - strong
| TLS_RSA_WITH_3DES_EDE_CBC_SHA - strong
| TLS_RSA_WITH_AES_128_CBC_SHA - strong
| TLS_RSA_WITH_AES_128_CBC_SHA256 - strong
| TLS_RSA_WITH_AES_128_GCM_SHA256 - strong
| TLS_RSA_WITH_AES_256_CBC_SHA - strong
| TLS_RSA_WITH_AES_256_CBC_SHA256 - strong
| TLS_RSA_WITH_AES_256_GCM_SHA384 - strong
| compressors:
| NULL
|_ least strength: strong

Service detection performed. Please report any incorrect results at http
://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 22.43 seconds
You have new mail in /var/spool/mail/root
```

Note: You must perform steps 2 through 5 every time you upgrade Cloudera Manager.

6. Set the SSL_CIPHER_LIST property for the Hue Server in Cloudera Manager.
   a) Sign in to Cloudera Manager as an Administrator.
   b) Go to Clusters Hue service Configuration Hue Server and specify the following in the Hue Server Advanced Configuration Snippet (Safety Valve) for hue_safety_valve_server.ini field:

   ```
   [desktop]
   ssl_cipher_list=DEFAULT:!aNULL:!eNULL:!LOW:!EXPORT:!SSLv2:!SSLv3:!TLSv1
   ```

   The SSL_CIPHER_LIST property is a list of one or more cipher suite strings separated by colons. This restricts the use of the default cipher suite before establishing an encrypted SSL connection.
   c) Click Save Changes.
   d) Restart the Hue service.

Securing sessions

When a Hue session expires, the screen blurs and the user is automatically logged out of the Hue Web UI. Logging back on returns the user to the same location in the application.

Session timeout

User sessions are controlled with the ttl (time-to-live) property, which is set in the Cloudera Manager Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini property as follows:

```
[desktop]
     [[session]]
     ttl=<number_of_seconds>
```

The default setting for ttl is 1,209,600 seconds, which equals two weeks. The ttl property determines the length of time that the cookie with the user's session ID lives before expiring. After the ttl setting is reached, the user's session expires whether it is active or not.

Idle session timeout

Idle sessions are controlled with the idle_session_timeout property, which is set in the Cloudera Manager Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini property as follows:

```
[desktop]
     [[auth]]
     idle_session_timeout=<number_of_seconds>
```

Sessions expire that are idle for the number of seconds set for this property. For example, if you set idle_session_timeout=900, sessions expire after being idle for 15 minutes. You can disable the property by setting it to a negative value, like idle-session_timeout=-1.

Secure session login

Session login properties are set under [desktop] > [[auth]] in the Cloudera Manager Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini property as follows:

```
[desktop]
     [[auth]]
     <set_session_login_properties_here>
```

Note: These configuration settings are based on django-axes 1.5.0.

Use the following properties to configure session login behavior:

change_default_password	Valid values: true | false
	If this property is set to true, users must change their passwords on first login attempt.
	Example:
	`[desktop]`
	`[[auth]]`
	`change_default_password=true`
	To use this property, you must enable the AllowFirstUserDjangoBackend in Hue. For example:
	`[desktop]`
	`[[auth]]`
	`backend=desktop.auth.backend.AllowFirstUserDjangoBackend`

expires_after	Use this property to configure the number of seconds after logout that user accounts are disabled. For example, user accounts are disabled 900 seconds or 15 minutes after logout with the following configuration:
	`[desktop]`
	`[[auth]]`
	`expires_after=900`
	If you set this property to a negative value, user sessions never expire. For example, expires_after=-1.

expire_superusers	Use to expire superuser accounts after the specified number of seconds after logout. For example, expire_superusers=900 causes superuser accounts to expire 15 minutes after logging out.
login_cooloff_time	Sets the number of seconds after which failed logins are forgotten. For example, if you set login_cooloff_time=900, a failed login attempt is forgotten after 15 minutes.
login_failure_limit	Sets the number of login attempts allowed before a failed login record is created. For example, if you set login_failure_limit=3, a failed login record is created after 3 login attempts.
login_lock_out_at_failure	Valid values: true | false If set to true: • The IP address that is attempting to log in is locked out after exceeding the limit set for login_failure_limit. • If login_lock_out_by_combination_user_and_ip is also set to true, both the IP address and the user are locked out after exceeding the limit set for login_failure_limit. • If login_lock_out_use_user_agent is also set to true, both the IP address and the agent application (such as a browser) are locked out after exceeding the limit set for login_failure_limit.
login_lock_out_by_combination_user_and_ip	Valid values: true | false If set to true, both the IP address and the user are locked out after exceeding the limit set for login_failure_limit.
login_lock_out_use_user_agent	Valid values: true | false If set to true, the agent application (such as a browser) is locked out after exceeding the limit set for login_failure_limit.

Secure session cookies

Session cookie properties are set under [desktop] > [[session]] in the Cloudera Manager Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini property as follows:

```
[desktop]
     [[session]]
     <set_session_cookie_properties_here>
```

Use the following properties to configure session cookie behavior:

secure	Valid values: true | false
	If this property is set to true, the user session ID is secured.
	Important: To use this property, HTTPS must be enabled.
	Example:
	`[desktop]`
	`[[session]]`
	`secure=true`
	By default this property is set to false.

http_only	Valid values: true | false
	If this property is set to true, the cookie with the user session ID uses the HTTP only flag.
	Example:
	`[desktop]`
	`[[session]]`
	`http_only=true`
	Important: If the HttpOnly flag is included in the HTTP response header, the cookie cannot be accessed through a client side script.
	By default this property is set to true.

expire_at_browser_close	Valid values: true | false
	If this property is set to true, only session-length cookies are used. Users are automatically logged out when the browser window is closed.
	Example:
	`[desktop]`
	`[[session]]`
	`expire_at_browser_close=true`
	By default this property is set to false.

Specifying HTTP request methods

You can specify the HTTP request methods that the Hue server responds to.

Use the http_allowed_methods property under [desktop] in the Cloudera Manager Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini property.

By default, the http_allowed_methods property is set to options, get, head, post, put, delete, connect.

Restricting supported ciphers for Hue

You can configure the list of ciphers that Hue supports with HTTPS.

Use the ssl_cipher_list property under [desktop] in the Cloudera Manager Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini property:

```
[desktop]
ssl_cipher_list=<list_of_accepted_ciphers>
```

By default, the ssl_cipher_list property is set to !aNULL:!eNULL:!LOW:!EXPORT:!SSLv2. Specify ciphers using the cipher list format described at OpenSSL Cryptograpy and SSL/TLS Toolkit Manpages by selecting the SSL version, and then going to Commands ciphers .

Specifying domains or pages to which Hue can redirect users

You can restrict the domains or pages to which Hue can redirect users.

Use the redirect_whitelist property under [desktop] in the Cloudera Manager Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini property:

```
[desktop]
redirect_whitelist=<redirect_URL>
```

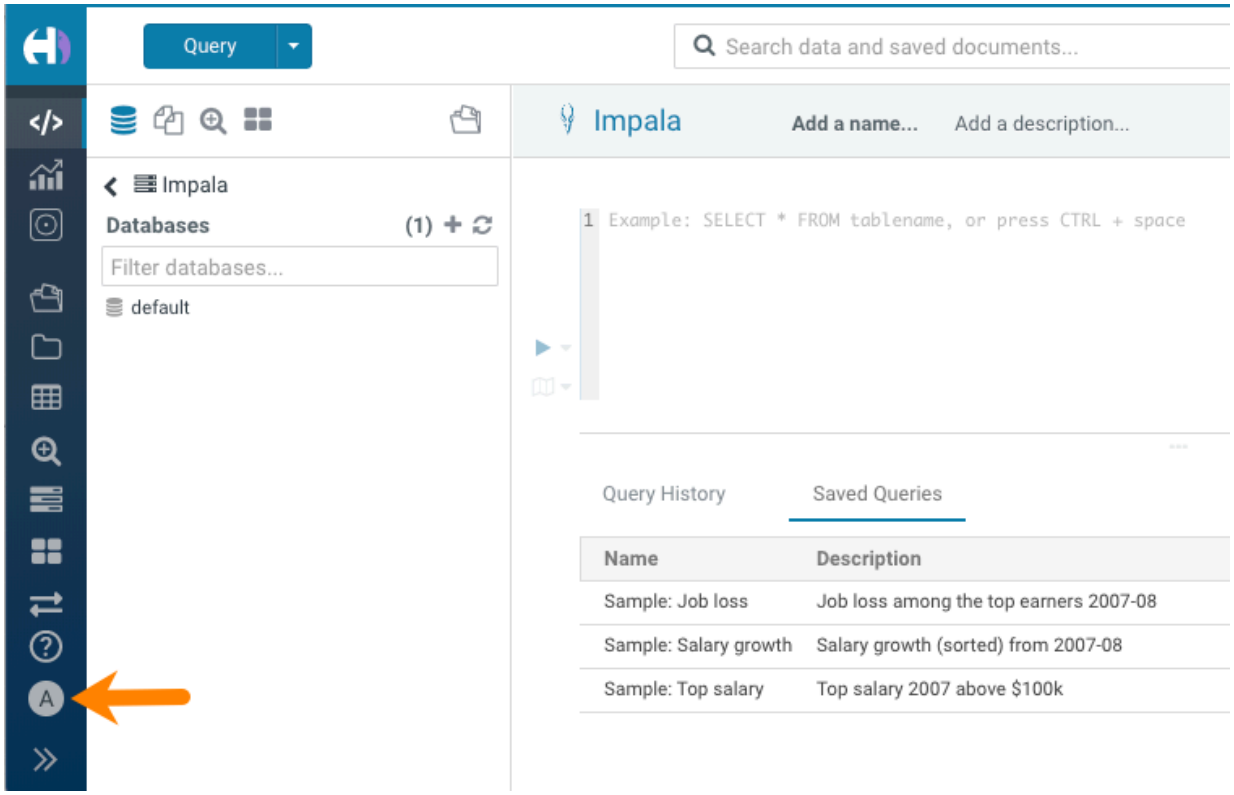
Specify the redirect_whitelist value with a comma-separated list of regular expressions that match the redirect URL. For example, to restrict redirects to your local domain and fully-qualified domain name (FQDN), use the following value:

```
redirect_whitelist=^\/.*$,^http:\/\/www.mydomain.com\/.*$
```

Setting Oozie permissions

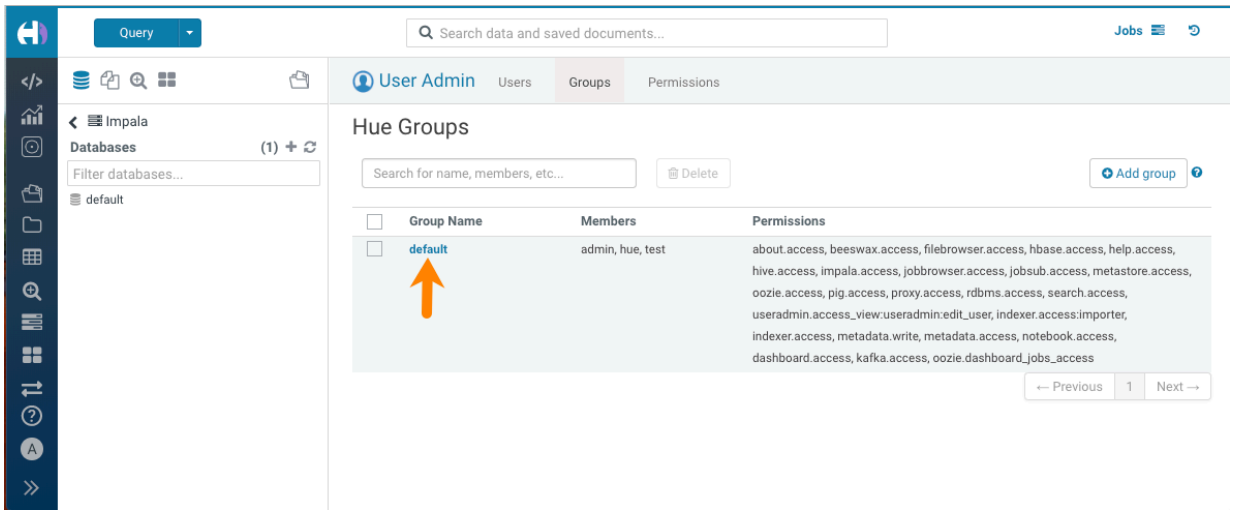
You can control access to the Oozie dashboard and editor by using controls in the Hue Web UI.

1. On the Cloudera Manager home page, click the Hue service.
2. On the Hue service page, select Web UIHue Load Balanced - recommended .
3. Log in to the Hue Web UI.
4. In the Hue Web UI, click the admin menu icon in the lower part of the left menu and select Manage Users:

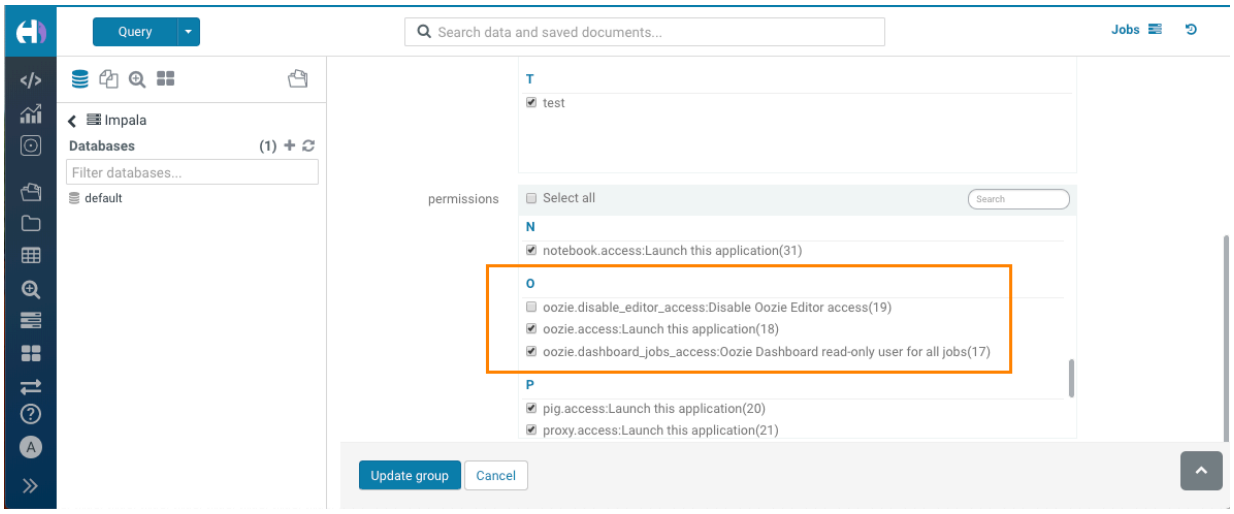


5. On the User Admin page, click the Groups tab.

6. On the Hue Groups page, in the Group Name column, click the default group.



7. On the Hue Groups - Edit group page, scroll down to locate the list of permissions and then scroll further to locate the Oozie permissions:



Groups property in UI	Description
oozie.disable_editor_access	Disables access to the Oozie editor for the selected groups Default setting: Unchecked, which disables this permission.
oozie.access	Enables access to the Oozie editor in Hue. Default setting: Checked, which enables access to the Oozie editor.
oozie.dashboard_jobs_access	Enables read-only access for all jobs in the Oozie dashboard. Default setting: Check, which enables this permission.

8. Check or uncheck the permissions as needed and then click Update group to save the permission change.

Configuring secure access between Solr and Hue

If you are using Solr (search) to build dynamic search dashboards and explore data from the Hue web UI, and have secured your CDP cluster using Kerberos, then you must enable secure access between the Solr service and Hue by turning on the security_enabled parameter in the hue_safety_valve.ini file.

About this task

The security_enabled parameter is set to false, by default. To enable secure access between Solr and Hue:

Procedure

1. Sign in to Cloudera Manager as an Administrator.
2. Go to Clusters Hue service Configuration and search for the Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini field.
3. Go to the [search] section and set the value of the security_enabled parameter to true, as shown in the following sample:

```
[search]
# Requires FQDN in solr_url if enabled
security_enabled=true
# URL of the Solr Server
solr_url=http://[***FQDN-SOLR-HOST***]:8983/solr
```

   Note: If you set security_enabled=true, then you must provide the fully-qualified domain name of the Solr host in the Solr URL.

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