

## Securing Hue

**Date published:** 2020-07-28

**Date modified:** 2025-06-06

**CLOUDERA**

# Legal Notice

© Cloudera Inc. 2026. 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

<b>User management in Hue.....</b>	<b>4</b>
Understanding Hue users and groups.....	4
Finding the list of Hue superusers.....	5
Creating a Hue user in Cloudera Data Warehouse.....	6
Creating a group in Hue.....	6
Managing Hue permissions.....	7
<b>User authentication in Hue.....</b>	<b>7</b>
Authenticating Hue users with SAML.....	7
SAML properties.....	8
<b>Securing sessions.....</b>	<b>9</b>
<b>Specifying HTTP request methods.....</b>	<b>11</b>
<b>Restricting supported ciphers for Hue.....</b>	<b>12</b>
<b>Specifying domains or pages to which Hue can redirect users.....</b>	<b>12</b>
<b>Securing Hue from CWE-16.....</b>	<b>12</b>

## User management in Hue

Hue is a gateway to Cloudera 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 Cloudera service accessible within Hue.

A common configuration is for “Hue users” to be authenticated with an LDAP server and “Cloudera users” with Kerberos. These users can differ. For example, Cloudera 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 Cloudera 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.

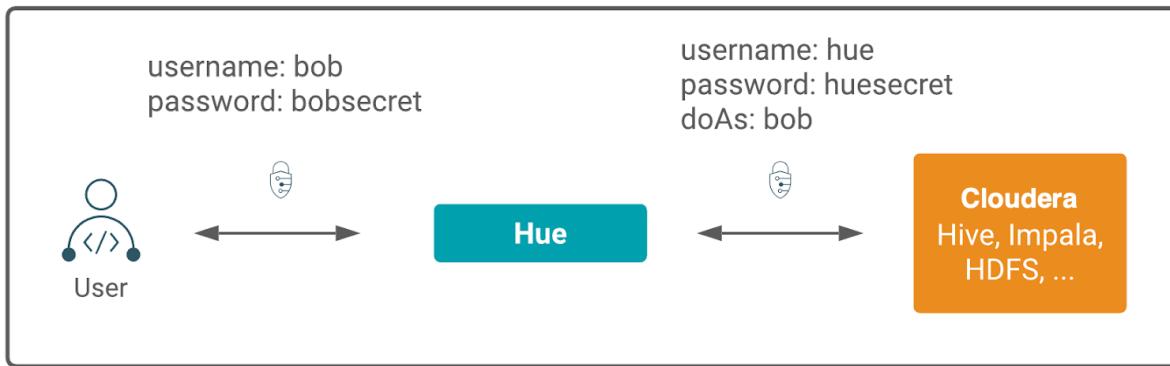
On a non-secure Cloudera cluster, the first user logging into Hue after the initial installation becomes the first superuser. Superusers have the permissions to perform the following administrative functions:

- 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 Cloudera 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 Cloudera services. Hue is considered as a single ‘hue’ user by the other services in the Cloudera 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 Cloudera. The HIVE\_ON\_TEZ service in Cloudera considers ‘bob’ as the owner of the query and not ‘hue’. This is illustrated in the following graphic:



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

### Running a SQL query on the auth\_user table to find Hue superusers

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

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

2. Run the following SQL query:

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

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

Sample output:

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

## Creating a Hue user in Cloudera Data Warehouse

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. Log in to Hue as a superuser.
2. From the left assist panel, point your cursor to the user profile icon and click Administer Users.
3. On the **User Admin** page, click Add user.
4. On the **Step 1. Credentials (required)** tab on the **Create user** page, specify the username and password and select the Create home directory option if you want to create a separate Hue home directory for this user.
5. Go to the Step 2. Profile and Groups tab.
6. Enter the first and last name of the user, their email address, and add them to the required groups.  
A user can be a part of more than one group.
7. Go to the Step 3. Advanced tab.
8. Ensure that the user is active by selecting the Active option.
9. If you want to make this user a superuser, then select the Superuser status option.
10. Click Add user.

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

## 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 Administer Users.  
The **User Admin** page is displayed.
3. 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 Administer 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.

## User authentication in Hue

Cloudera Data Warehouse supports authenticating users to Hue using SAML.

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 SAML

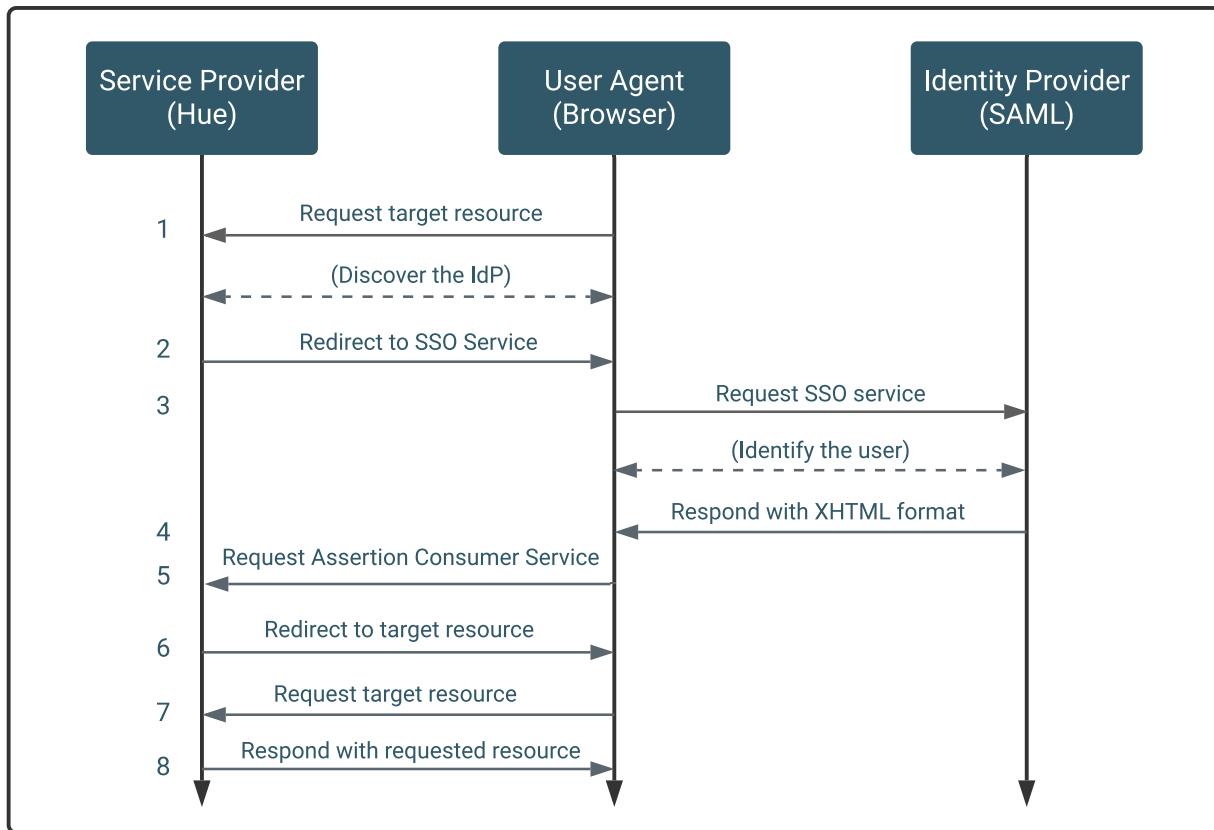
Hue supports SAML (Security Assertion Markup Language) for Single Sign-on (SSO) authentication.

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 1: 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.

In Cloudera Data Warehouse, SSO with SAML is automatically configured. You need not configure anything manually.

## SAML properties

In Cloudera Data Warehouse, SSO with SAML is automatically configured. However, if you need to configure a certain parameter, you can set the properties in the hue-safety-valve field.

**Table 1: 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.

SAML parameter	Description
logout_requests_signed	Boolean, that when True, signs Hue-initiated logout requests with an X.509 certificate.
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>\\V.\\*\$".
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.

## Securing sessions

When a Hue session expires, the screen blurs and the user is automatically logged out of the Hue web interface. 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 Cloudera Data Warehouse Virtual Warehouses Edit CONFIGURATIONS Hue Configuration files hue-safety-valve 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 Cloudera Data Warehouse Virtual Warehouses Edit CONFIGURATIONS Hue Configuration files hue-safety-valve 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\_time\_out=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 the [desktop] [[auth]] section in Cloudera Data Warehouse Virtual Warehouses Edit CONFIGURATIONS Hue Configuration files hue-safety-valve property as follows:

```
[desktop]
  [[auth]]
  [***SET-SESSION-LOGIN-PARAMETERS-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	<p>Valid values: true   false</p> <p>If this property is set to true, users must change their passwords on first login attempt.</p> <p>Example:</p> <pre>[desktop]   [[auth]]   change_default_password=true</pre> <p>To use this property, you must enable the AllowFirstUserDjangoBackend in Hue. For example:</p> <pre>[desktop]   [[auth]]   backend=desktop.auth.backend.AllowFirstUserDjangoBackend</pre>
expires_after	<p>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:</p> <pre>[desktop]   [[auth]]   expires_after=900</pre> <p>If you set this property to a negative value, user sessions never expire. For example, expires_after=-1.</p>
expire_superusers	<p>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.</p>
login_cooloff_time	<p>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.</p>
login_failure_limit	<p>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.</p>
login_lock_out_at_failure	<p>Valid values: true   false</p> <p>If set to true:</p> <ul style="list-style-type: none"> <li>The IP address that is attempting to log in is locked out after exceeding the limit set for login_failure_limit.</li> <li>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.</li> <li>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.</li> </ul>
login_lock_out_by_combination_user_and_ip	<p>Valid values: true   false</p> <p>If set to true, both the IP address and the user are locked out after exceeding the limit set for login_failure_limit.</p>

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 the [desktop] [[session]] section in Cloudera Data Warehouse Virtual Warehouses Edit CONFIGURATIONS Hue Configuration files hue-safety-valve property as follows:

```
[desktop]
  [[session]]
  [***SET-SESSION-COOKIE-PROPERTIES-HERE***]
```

Use the following properties to configure session cookie behavior:

secure	<p>Valid values: true   false If this property is set to true, the user session ID is secured.</p> <p><b>Important:</b> To use this property, HTTPS must be enabled.</p>  <p>Example:</p> <pre>[desktop]   [[session]]   secure=true</pre> <p>By default this property is set to false.</p>
http_only	<p>Valid values: true   false If this property is set to true, the cookie with the user session ID uses the HTTP only flag.</p> <p>Example:</p> <pre>[desktop]   [[session]]   http_only=true</pre> <p><b>Important:</b> If the HttpOnly flag is included in the HTTP response header, the cookie cannot be accessed through a client side script.</p>  <p>By default this property is set to true.</p>
expire_at_browser_close	<p>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.</p> <p>Example:</p> <pre>[desktop]   [[session]]   expire_at_browser_close=true</pre> <p>By default this property is set to false.</p>

## Specifying HTTP request methods

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

Use the http\_allowed\_methods property under the [desktop] section in Cloudera Data Warehouse Virtual Warehouses Edit CONFIGURATIONS Hue Configuration files hue-safety-valve 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 the [desktop] section in Cloudera Data Warehouse Virtual Warehouses Edit CONFIGURATIONS Hue Configuration files hue-safety-valve 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 Cryptography 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 the [desktop] section in Cloudera Data Warehouse Virtual Warehouses Edit CONFIGURATIONS Hue Configuration files hue-safety-valve 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=^\/.*$,^http:\/\/www.mydomain.com\/.*$
```

## Securing Hue from CWE-16

Hue may have allowed external domains such as doubleclick.net, .googletagmanager.com, or \*.google-analytics.com to run JavaScript scripts, for certain URLs in the Content Security Policy (CSP) headers. This may lead to Common Weakness Enumeration (CWE-16). To secure Hue from CWE-16 class of weaknesses, you can add the X-Content-Type-Options response HTTP header and prevent attacks based on MIME-type confusions in Hue's Advanced Configuration Snippet using Cloudera Manager.

### Procedure

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

```
[desktop]
# X-Content-Type-Options: nosniff This is an HTTP response header
```

```
# feature that helps prevent attacks based on MIME-type confusion.

secure_content_security_policy="script-src 'self' 'unsafe-inline' 'unsafe-eval' *.googletagmanager.com *.doubleclick.net data:;img-src 'self' *.doubleclick.net http://*.tile.osm.org *.tile.osm.org *.gstatic.com data:;style-src 'self' 'unsafe-inline' fonts.googleapis.com;connect-src 'self' *.google-analytics.com;frame-src *;child-src 'self' data: *.vimeo.com;object-src 'none'"
```

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