Cloudera Streaming Analytics 1.4.1

Manage SSB Security

Date published: 2019-12-17
Date modified: 2021-07-20

CLOUDERA

https://docs.cloudera.com/

Legal Notice

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

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

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

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

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

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

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

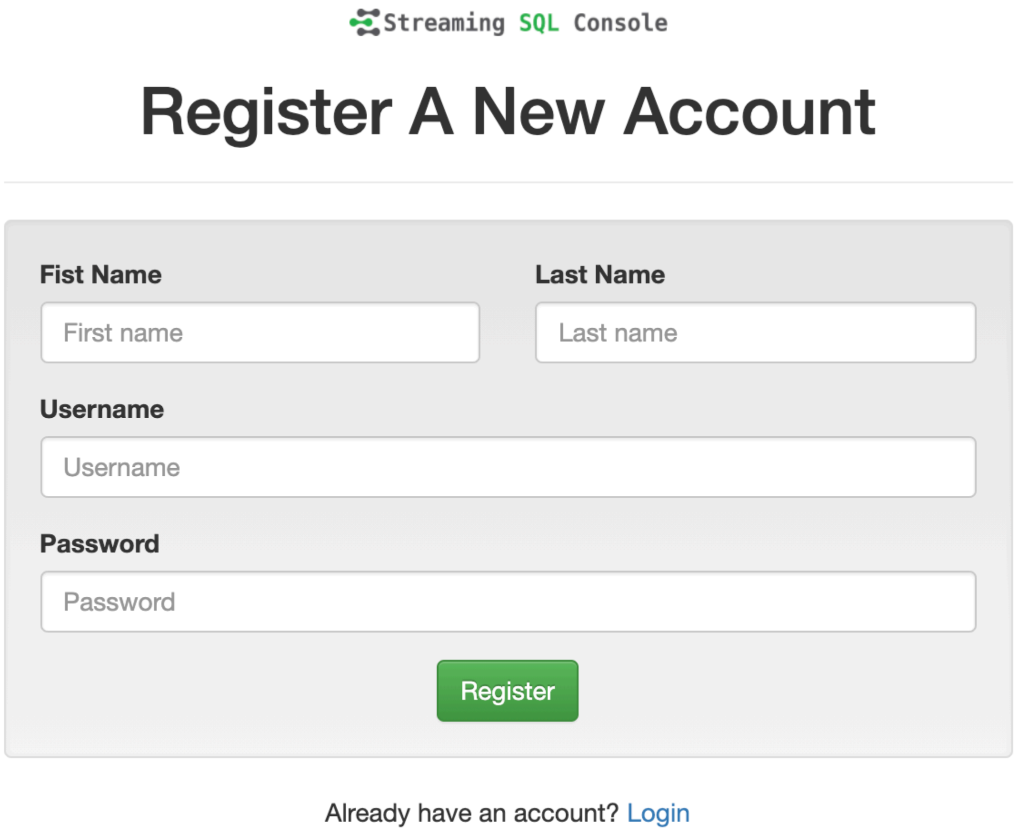
Disclaimer: EXCEPT AS EXPRESSLY PROVIDED IN A WRITTEN AGREEMENT WITH CLOUDERA, CLOUDERA DOES NOT MAKE NOR GIVE ANY REPRESENTATION, WARRANTY, NOR COVENANT OF ANY KIND, WHETHER EXPRESS OR IMPLIED, IN CONNECTION WITH CLOUDERA TECHNOLOGY OR RELATED SUPPORT PROVIDED IN CONNECTION THEREWITH. CLOUDERA DOES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

Contents

Authentication in SSB

You can authenticate users to access the Streaming SQL Console using Kerberos or Knox authentication.

On an unsecured cluster, you need to register a new account to access the SSB Console. Provide your Name, Username and Password to create an account.



Streaming SQL Console

Register A New Account

Fist Name	Last Name
First name	Last name

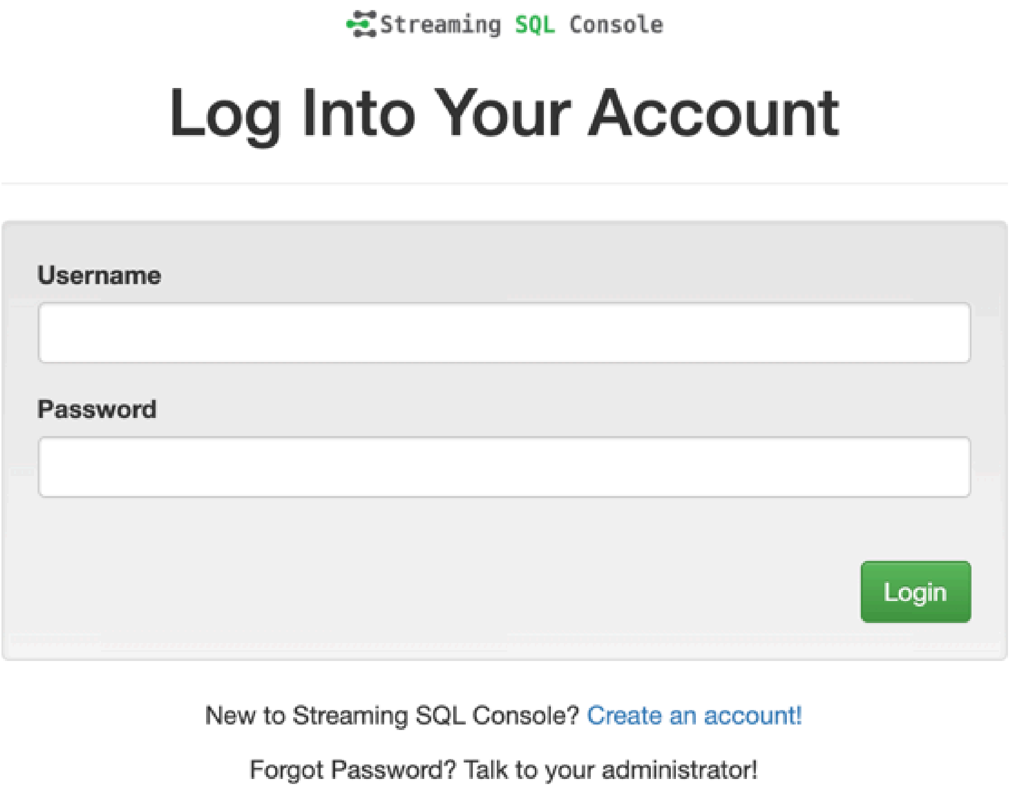
Username

Username

Password

Password

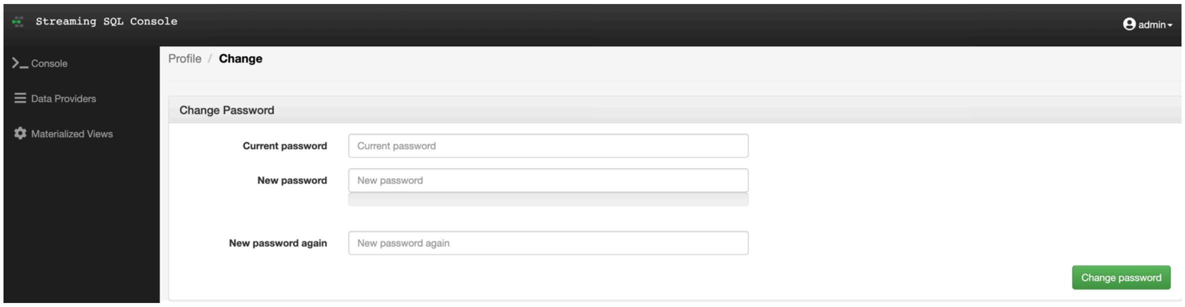
After creating an account, you can log in to the Streaming SQL Console by providing the registered Username and Password.



New to Streaming SQL Console? Create an account!

Forgot Password? Talk to your administrator!

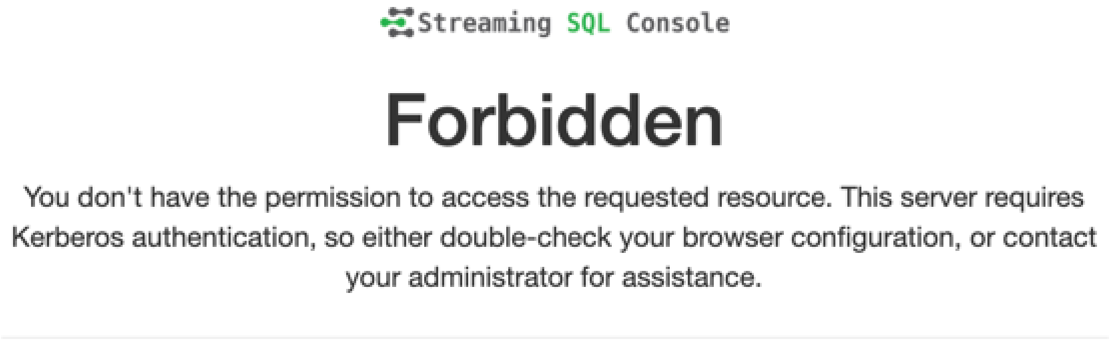
Note: You can later change your password by clicking on your username at the right top of the Streaming SQL Console, and by selecting the Profile tab.



Enabling Kerberos authentication

You need to enable Kerberos authentication in Cloudera Manager as well as directly for your browser to securely reach the Streaming SQL Console, and to use Knox authentication.

When Kerberos authentication is set up for SSB, and an unauthorized user wants to reach the Streaming SQL Console, the following error message appears:



Forbidden

You don't have the permission to access the requested resource. This server requires Kerberos authentication, so either double-check your browser configuration, or contact your administrator for assistance.

1. Go to your cluster in Cloudera Manager.
2. Click SQL Stream Builder from the list of services.
3. Go to the Configuration tab.
4. Select Category > Security .
5. Type *kerberos* in the search field.
6. Select the Enable Kerberos authentication setting.
7. Open a terminal window.

8. Configure your browser for Kerberos authentication:

   - Mozilla Firefox

     a. Load the about:config page to open the low level Firefox configuration.
     b. Search for network.negotiate-auth.trusted-uris preference.
     c. Open the network.negotiate-auth.trusted-uris preference.
     d. Enter the hostnames of the SQL Stream Console are protected by Kerberos HTTP SPNEGO.
     e. Click Ok.
   - Internet Explorer

     a. Configure the Local Intranet Domain

        1. Click on the Settings icon in Internet Explorer.
        2. Go to Internet options > Security.
        3. Select Local Intranet zone.
        4. Click on Sites.
        5. Review that the following options are checked:

           - Include all local (intranet) sites not listed in other zones
           - Include all sites that bypass the proxy server are checked
        6. Click Advanced.
        7. Enter the hostnames and domains of the SQL Stream Console that are protected by Kerberos HTTP SPNEGO.
        8. Click Ok.
     b. Configure the Intranet Authentication

        1. Click on the Settings icon in Internet Explorer.
        2. Go to Internet options > Security.
        3. Select Local Intranet zone.
        4. Click on Custom level.

           The Security Settings - Local Intranet Zone dialog box opens.
        5. Scroll down to the User Authentication options.
        6. Select Automatic logon only in Intranet Zone.
        7. Click Ok.
     c. Verify the Proxy Settings

        1. Make sure that you enabled a proxy server.
        2. Click on the Settings icon in Internet Explorer.
        3. Go to Internet options > Connections.
        4. Select LAN settings.
        5. Confirm that the proxy server Address and Port number are correct.
        6. Click Advanced.

           The Proxy Settings dialog box opens.
        7. Add the Streaming SQL Console domains that are protected by Kerberos to the Exceptions field.
        8. Click Ok.
   - Google Chrome

     - Windows

       a. Open Control Panel > Internet Options > Security.
       b. Perform the steps from the Internet Explorer configuration.
     - MacOS

       a. Open a terminal window.

b. Copy and paste the following command:

```
defaults write com.google.Chrome AuthServerAllowlist
"*<host_domain>,*<host_domain1>,*<host_domain2>"
sudo scp <your_hostname>:/etc/krb5.conf /etc/krb5.conf
kinit <username>
```

Configuring custom Kerberos principal for SQL Stream Builder

The Kerberos principal for SQL Stream Builder is configured by default to use the same service principal as the default process user. However, you can change the default setting by providing a custom principal in Cloudera Manager.

Procedure

1. Go to your Cluster in Cloudera Manager.
2. Select SQL Stream Builder from the list of services.
3. Go to the Configuration tab.
4. Search for the Kerberos principal by entering "kerberos" in the search field.
5. Provide a custom name to the Kerberos Principal property.
6. Click Save Changes.
7. Click Actions > Restart next to the SQL Stream Builder service name to restart the service

Enabling Knox authentication

The auto-discovery function in Cloudera Manager does not support SSB in this release, you need to manually configure Knox and SSB to enable Knox authentication.

Apache Knox Gateway is used to help ensure perimeter security for SSB. With Knox, enterprises can confidently extend the SSB UI and API endpoints to new users without Kerberos complexities. Knox provides a central gateway and has varying degrees of authorization, authentication, SSL, and SSO capabilities to enable a single access point for SSB.

Before you begin

- Install and configure Knox on your cluster. For more information, see the Installing Apache Knox documentation.
- Enable Kerberos authentication for SSB. For more information, see Enabling Kerberos authentication section.

Cloudera Manager does not support simplified topology management (also known as auto-discovery) for the current SSB release. You need to manually configure Knox to be able to integrate with SSB, this consist of the following high level steps:

Extending Knox with the SSB service definitions

You must create service definitions for the SSB Console and Materialized Views in the Knox Admin UI.

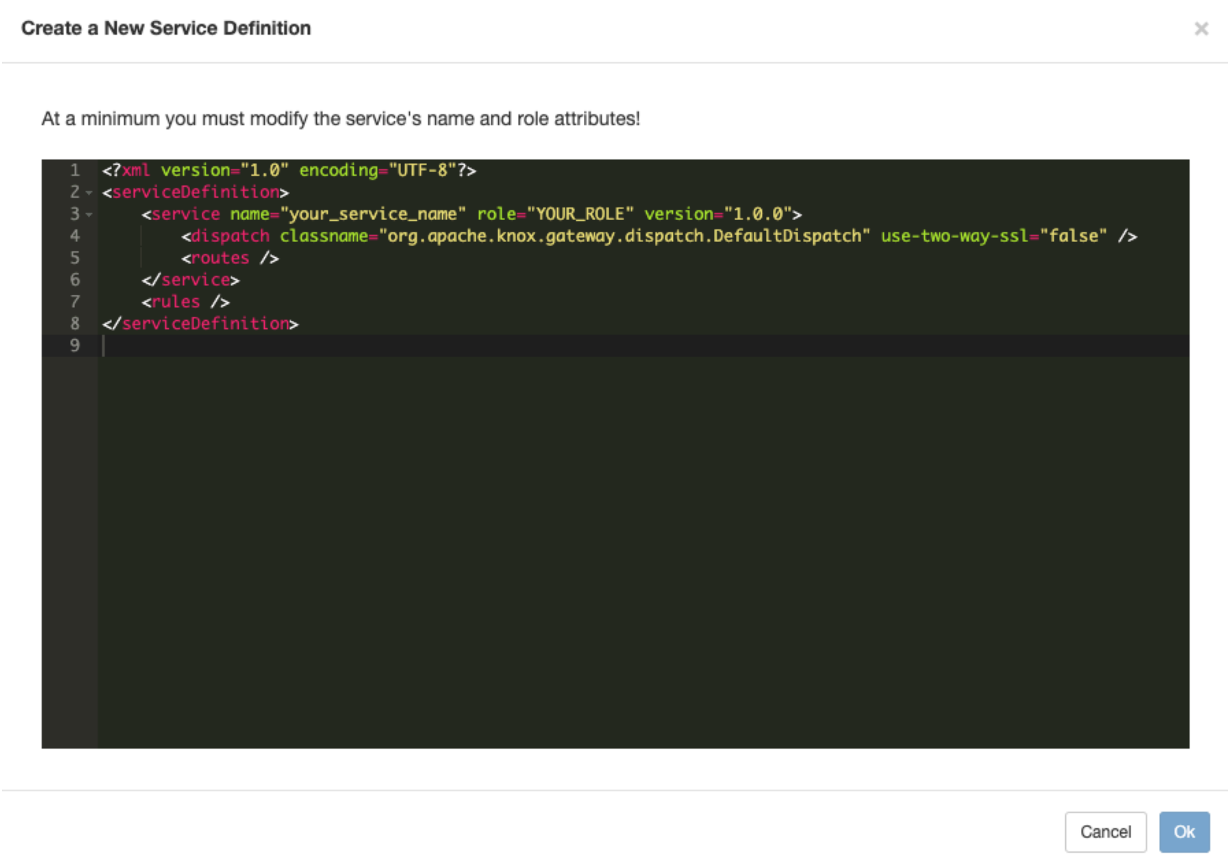
Procedure

1. Go to your cluster in Cloudera Manager.
2. Select Knox from the list of services.
3. Select Knox Gateway Home.
4. Open the General Proxy Information.
5. Click Admin UI URL.

   You are redirected to the Knox Manager page.
6. Click Service Definitions under Resource Types.

7. Click on the plus icon to add the SSB service definitions.

   The Create a New Service Definition window appears.



At a minimum you must modify the service's name and role attributes!

```
<?xml version="1.0" encoding="UTF-8"?>
<serviceDefinition>
    <service name="your_service_name" role="YOUR_ROLE" version="1.0.0">
        <dispatch classname="org.apache.knox.gateway.dispatch.DefaultDispatch" use-two-way-ssl="false" />
        <routes />
    </service>
    <rules />
</serviceDefinition>
```

8. Delete the default text from the window.
9. Create the service definitions for SSB.
   a) Copy the following XML entry for the SSB-MVE-API service definition:

```
<?xml version="1.0" encoding="UTF-8"?>
<serviceDefinitions>
   <serviceDefinition>
      <service name="ssb-mve-api" role="SSB-MVE-API" version="1.4.0.0">
         <metadata>
            <context>/ssb-mve-api</context>
            <description>Streaming SQL Builder - Materialized View Engine API</description>
            <shortDesc>SSB - MVE API</shortDesc>
            <type>API</type>
         </metadata>
         <routes>
            <route path="/ssb-mve-api/**">
               <rewrite apply="SSB-MVE-API/ssb-mve-api/path" to="request.url"/>
            </route>
         </routes>
      </service>
      <rules>
         <rule dir="IN" name="SSB-MVE-API/ssb-mve-api/path" pattern="*://*:*/**/ssb-mve-api/{path=**}?{**}">
            <rewrite template="{$serviceUrl[SSB-MVE-API]}/{path=**}?{**}"/>
         </rule>
      </rules>
   </serviceDefinition>
</serviceDefinitions>
```

b) Paste it to the New Service Definition window.
c) Click Ok.
d) Click on the plus icon to create a new service definition.
e) Copy the following XML entry for the SSB-MVE-API-PUBLIC service definition:

```
<?xml version="1.0" encoding="UTF-8"?>
<serviceDefinitions>
   <serviceDefinition>
      <service name="ssb-mve-api-public" role="SSB-MVE-API-PUBLIC" ve
rsion="1.4.0.0">
         <metadata>
            <context>/ssb-mve-api-public</context>
            <description>Streaming SQL Builder - Materialized View Engin
e Public API</description>
            <shortDesc>SSB - MVE Public API</shortDesc>
            <type>API</type>
         </metadata>
         <routes>
            <route path="/ssb-mve-api-public/api/**">
               <rewrite apply="SSB-MVE-API-PUBLIC/ssb-mve-api-public/pa
th" to="request.url"/>
            </route>
         </routes>
      </service>
      <rules>
         <rule dir="IN" name="SSB-MVE-API-PUBLIC/ssb-mve-api-public/pa
th" pattern="*://*:*/**/ssb-mve-api-public/api/{path=**}?{**}">
            <rewrite template="{$serviceUrl[SSB-MVE-API-PUBLIC]}/api/{pa
th=**}?{**}"/>
         </rule>
      </rules>
   </serviceDefinition>
</serviceDefinitions>
```

f) Paste it to the New Service Definition window.
g) Click Ok.
h) Copy the following XML entry for the SSB-SSC-UI service definition:

```
<?xml version="1.0" encoding="UTF-8"?>
<serviceDefinitions>
   <serviceDefinition>
      <service name="ssb-ssc-ui" role="SSB-SSC-UI" version="1.4.0.0">
         <dispatch classname="org.apache.knox.gateway.dispatch.Configur
ableDispatch" use-two-way-ssl="false">
            <param>
               <name>responseExcludeHeaders</name>
               <value>Www-Authenticate</value>
            </param>
         </dispatch>
         <metadata>
            <context>/ssb-ssc-ui</context>
            <description>SQL Streaming Builder - Streaming SQL Console -
 UI</description>
            <shortDesc>SSB Console</shortDesc>
            <type>UI</type>
         </metadata>
         <routes>
            <route path="/ssb-ssc-ui/">
```

```
               <rewrite apply="SSB-SSC-UI/filter/outbound/headers" to=
"response.headers"/>
               <rewrite apply="SSB-SSC-UI/filter/outbound/body" to="res
ponse.body"/>
            </route>
            <route path="/ssb-ssc-ui/**">
               <rewrite apply="SSB-SSC-UI/filter/outbound/headers" to="
response.headers"/>
               <rewrite apply="SSB-SSC-UI/filter/outbound/body" to="resp
onse.body"/>
            </route>
            <route path="/ssb-ssc-ui/**?**">
               <rewrite apply="SSB-SSC-UI/filter/outbound/headers" to
="response.headers"/>
               <rewrite apply="SSB-SSC-UI/filter/outbound/body" to="re
sponse.body"/>
            </route>
         </routes>
      </service>
      <rules>
         <rule dir="IN" name="SSB-SSC-UI/rule/inbound/query" pattern="
*://*:*/**/ssb-ssc-ui/{path=**}?{**}">
            <rewrite template="{$serviceUrl[SSB-SSC-UI]}/{path=**}?{*
*}"/>
         </rule>
         <rule dir="IN" name="SSB-SSC-UI/rule/inbound/path" pattern="
*://*:*/**/ssb-ssc-ui/{**}">
            <rewrite template="{$serviceUrl[SSB-SSC-UI]}/{**}"/>
         </rule>
         <rule dir="IN" name="SSB-SSC-UI/rule/inbound/root" pattern="
*://*:*/**/ssb-ssc-ui/">
            <rewrite template="{$serviceUrl[SSB-SSC-UI]}/"/>
         </rule>
         <rule dir="OUT" flow="OR" name="SSB-SSC-UI/rule/outbound/hea
der/links">
            <match pattern="*://*:*/{path=**}?{**}">
               <rewrite template="{$frontend[url]}/ssb-ssc-ui/{path=**}
"/>
            </match>
            <match pattern="*://*:*/{path=**}">
               <rewrite template="{$frontend[url]}/ssb-ssc-ui/{path=**}
"/>
            </match>
            <match pattern="*://*:*/">
               <rewrite template="{$frontend[url]}/ssb-ssc-ui/"/>
            </match>
         </rule>
         <rule dir="OUT" name="SSB-SSC-UI/rule/outbound/html/static" pat
tern="/static/{**}">
            <rewrite template="{$frontend[url]}/ssb-ssc-ui/static/{**}"/
>
         </rule>
         <rule dir="OUT" flow="OR" name="SSB-SSC-UI/rule/outbound/js/api
">
            <match pattern="/api/v1/query"/>
            <match pattern="/api/v1">
               <rewrite template="{$frontend[path]}/ssb-ssc-ui/api/v1"/>
            </match>
         </rule>
         <rule dir="OUT" name="SSB-SSC-UI/rule/outbound/js/ws">
            <match pattern="/socket.io">
               <rewrite template="{$frontend[path]}/ssb-ssc-ws/socket.
io"/>
            </match>
         </rule>
         <rule dir="OUT" flow="OR" name="SSB-SSC-UI/rule/outbound/html/l
ink">
            <match pattern="/{path=**}?{**}">
               <rewrite template="{$frontend[url]}/ssb-ssc-ui/{path=**}?
{**}"/>
            </match>
            <match pattern="/{**}">
               <rewrite template="{$frontend[url]}/ssb-ssc-ui/{**}"/>
            </match>
         </rule>
         <filter name="SSB-SSC-UI/filter/outbound/headers">
            <content type="application/x-http-headers">
               <apply path="Location" rule="SSB-SSC-UI/rule/outbound/he
ader/links"/>
            </content>
         </filter>
         <filter name="SSB-SSC-UI/filter/outbound/body">
            <content type="*/javascript">
               <apply path="/api/v1/query|/api/v1" rule="SSB-SSC-UI/r
ule/outbound/js/api"/>
               <apply path="/socket.io" rule="SSB-SSC-UI/rule/outbound/
js/ws"/>
            </content>
            <content type="*/html">
               <apply path="/static" rule="SSB-SSC-UI/rule/outbound/ht
ml/static"/>
               <apply path="/api/v1" rule="SSB-SSC-UI/rule/outbound/htm
l/link"/>
               <apply path="/ui.*" rule="SSB-SSC-UI/rule/outbound/html/
link"/>
            </content>
         </filter>
      </rules>
   </serviceDefinition>
</serviceDefinitions>
```

i) Paste it to the New Service Definition window.

j) Click Ok.

k) Click on the plus icon to create a new service definition.

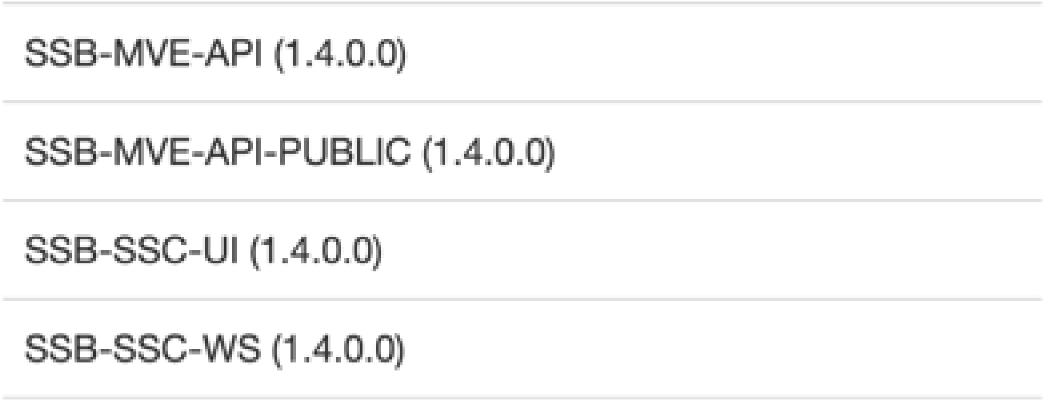
l) Copy the following XML entry for the SSB-SSC-WS service definition:

```
<?xml version="1.0" encoding="UTF-8"?>
<serviceDefinitions>
   <serviceDefinition>
      <service name="ssb-ssc-ws" role="SSB-SSC-WS" version="1.4.0.0">
         <routes>
            <route path="/ssb-ssc-ws/socket.io/">
               <rewrite apply="SSB-SSC-WS/ssb-ssc-ws/inbound1" to="requ
est.url"/>
            </route>
            <route path="/ssb-ssc-ws/socket.io/**">
               <rewrite apply="SSB-SSC-WS/ssb-ssc-ws/inbound2" to="reque
st.url"/>
            </route>
         </routes>
      </service>
      <rules>
         <rule dir="IN" name="SSB-SSC-WS/ssb-ssc-ws/inbound1" pattern="
*://*:*/**/ssb-ssc-ws/socket.io/">
            <rewrite template="{$serviceUrl[SSB-SSC-WS]}/socket.io/"/>
         </rule>
         <rule dir="IN" name="SSB-SSC-WS/ssb-ssc-ws/inbound2" pattern
="*://*:*/**/ssb-ssc-ws/socket.io/{**}">
            <rewrite template="{$serviceUrl[SSB-SSC-WS]}/socket.io/{**}
"/>
         </rule>
      </rules>
   </serviceDefinition>
</serviceDefinitions>
```

m) Paste it to the New Service Definition window.

n) Click Ok.

10. In the list of Service definitions, you should be able to see the following entries:



SSB-MVE-API (1.4.0.0)

SSB-MVE-API-PUBLIC (1.4.0.0)

SSB-SSC-UI (1.4.0.0)

SSB-SSC-WS (1.4.0.0)

Adding SSB services to the default topologies

After creating the service definitions, you must add the SSB services to the Knox default topologies in Cloudera Manager.

Procedure

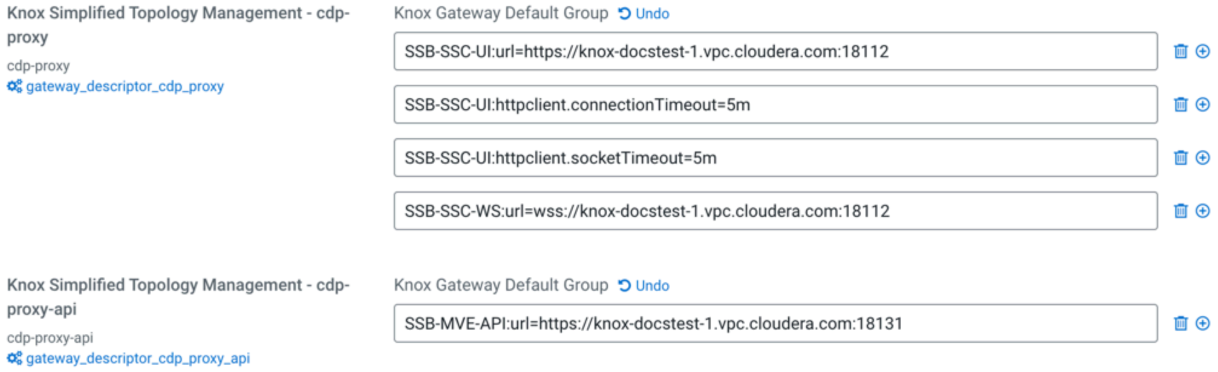
1. Go to your cluster in Cloudera Manager.
2. Click on Knox from the list of Services.
3. Select Configuration.
4. Search for Knox Simplified Topology Management.
5. Add the following entries to the Knox Simplified Topology Management - cdp-proxy:

```
SSB-SSC-UI:url=https://<ssb_console_host>:18112
SSB-SSC-UI:httpclient.connectionTimeout=5m
SSB-SSC-UI:httpclient.socketTimeout=5m
SSB-SSC-WS:url=wss://<ssb_console_host>:18112
```

6. Add the following entry to the Knox Simplified Topology Management - cdp-proxy-api:

```
SSB-MVE-API:url=https://<ssb_mv_host>:18131
```

You need to add the hostname to the entries as shown in the following example:



Knox Simplified Topology Management - cdp-proxy
cdp-proxy
gateway_descriptor_cdp_proxy

Knox Gateway Default Group Undo

SSB-SSC-UI:url=https://knox-docstest-1.vpc.cloudera.com:18112
SSB-SSC-UI:httpclient.connectionTimeout=5m
SSB-SSC-UI:httpclient.socketTimeout=5m
SSB-SSC-WS:url=wss://knox-docstest-1.vpc.cloudera.com:18112

Knox Simplified Topology Management - cdp-proxy-api
cdp-proxy-api
gateway_descriptor_cdp_proxy_api

Knox Gateway Default Group Undo

SSB-MVE-API:url=https://knox-docstest-1.vpc.cloudera.com:18131

7. Click Save changes.

   The Refresh needed indicator appears beside the Knox service name.
8. Refresh Knox.

Defining the external URLs in SSB

You must provide the cdp-proxy external URLs in Cloudera Manager to authenticate the user when accessing the Materialized Views and the Resource Manager through the Streaming SQL Console.

Procedure

1. Go to your cluster in Cloudera Manager.
2. Click on SQL Stream Builder from the list of Services.
3. Select Configuration.
4. Search for external_url.
5. Add the following URL to Materialized View Engine External API URL:

```
https://<ssb_mv_host>:8443/gateway/cdp-proxy-api/ssb-mve-api
```

6. Add the following URL to Yarn resource manager external URL override:

```
https://<yarn_rm_host>:8443/gateway/cdp-proxy/yarn-ui-v2
```

7. Restart the Knox service.

Reaching SSB through Knox

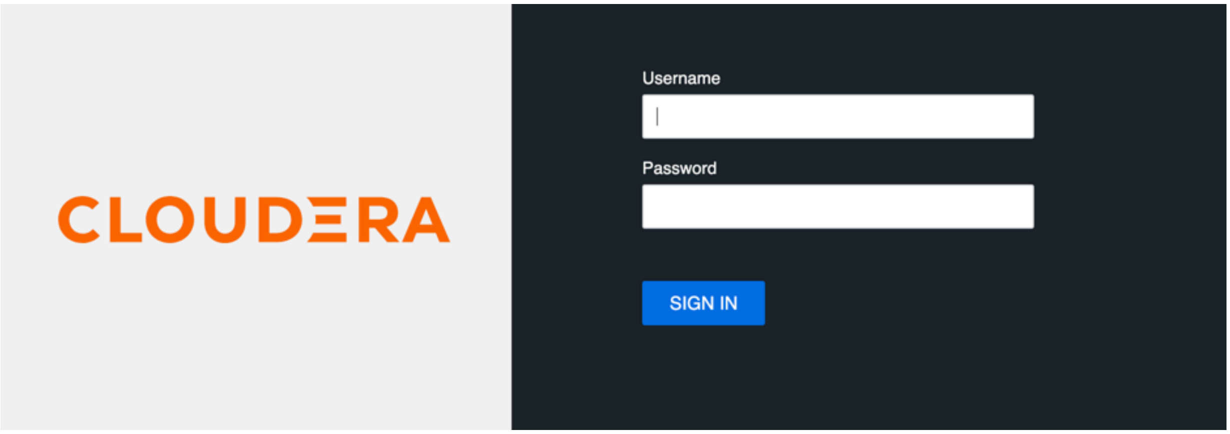
After manually configuring Knox and SSB, you should check if the SSO authentication works for the Streaming SQL Console.

Procedure

1. Go to your cluster in Cloudera Manager.
2. Click on Knox from the list of Services.

3. **3.** Select Knox Gateway Home.

   You will be prompted to provide your username and password.



4. **4.** Click cdp-proxy under Topologies.

   SSB Console should be listed under the cdp-proxy.

5. **5.** Click SSB Console.

   You are redirected to the Streaming SQL Console page.

   Note: In case SSB Console does not appear under the cdp-proxy topologies, try to clean up the deployments folder of Knox with the following command:

   ```
   rm -rf /var/lib/knox/gateway/data/deployments/*
   ```

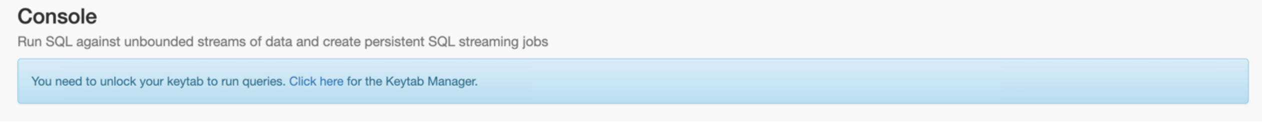
   After running the command, restart the Knox service.

Unlocking keytabs in SSB

After setting Kerberos or Knox authentication for SSB, you need to unlock the user specific keytabs on the Streaming SQL Console by providing your keytab passowrd or uploading the keytab file.

About this task

The following message is displayed on the Console in case the keytabs are still locked:



Console

Run SQL against unbounded streams of data and create persistent SQL streaming jobs

You need to unlock your keytab to run queries. Click here for the Keytab Manager.

Before you begin

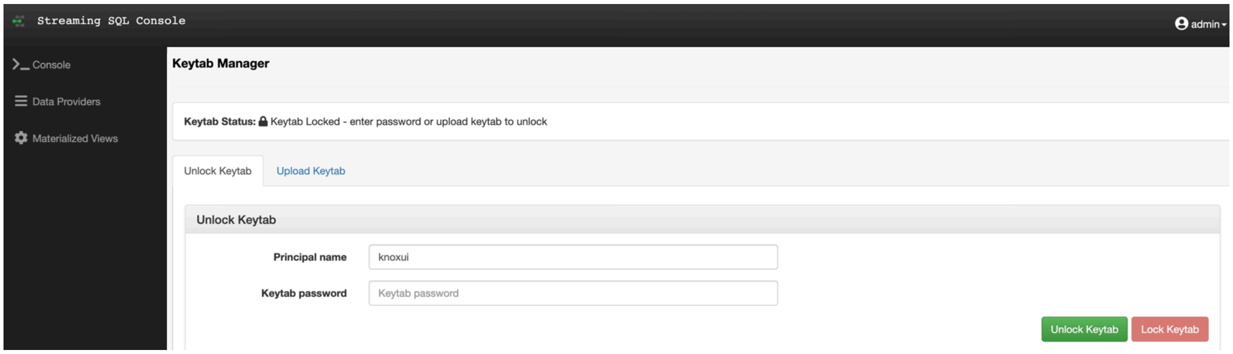
Before unlocking the keytab, you need to authenticate your username

Procedure

1. **1.** Click your username at the right top corner of the Streaming SQL Console.

2. Click Manage keytab.

   The Keytab Manager page appears.



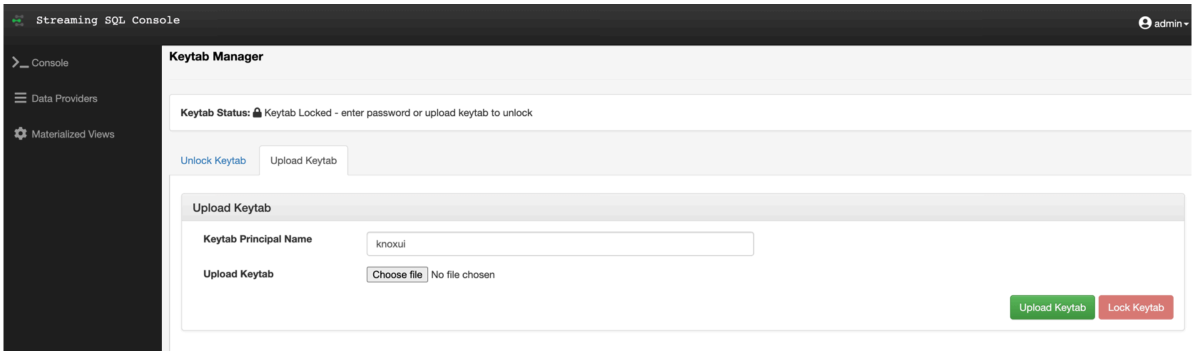
Keytab Status: 🔒 Keytab Locked - enter password or upload keytab to unlock

Principal name: knoxui

Keytab password: Keytab password

   You can choose between the following steps:

   - To unlock the keytab, provide the Keytab password for the principal.
   - Click Upload Keytab tab, and upload your keytab file directly to the Console.



Keytab Status: 🔒 Keytab Locked - enter password or upload keytab to unlock

Keytab Principal Name: knoxui

Upload Keytab: No file chosen

3. Click Unlock keytab.

Encryption in SSB

When auto-TLS is disabled for the SQL Stream Builder (SSB) service, you must manually set the TLS properties for SSB in Cloudera Manager.

Before you begin

Ensure that you have set up Transport Layer Security (TLS) (formerly known as Secure Socket Layer (SSL)) for Cloudera Manager:

- Generated TLS certificates
- Configured TLS for Admin Console and Agents
- Enabled server certificate verification on Agents
- Configured agent certificate authentication
- Configured TLS encryption on the agent listening port

For more information, see the Cloudera Manager documentation.

Note: You can also configure the security parameters when adding SQL Stream Builder as a service using the Add Service Wizard.

Procedure

1. Click SQL Builder service on your Cluster.

2. Go to the Configuration tab.
3. Select Category > Security.
   All the security related properties are displayed.
4. Edit the security properties according to the cluster configuration.

   Note: You need to provide the keystore and truststore information for the Materialized View Engine, the SQL Stream Engine, and for the Streaming SQL Console.

Materialized View Engine	
Enable TLS/SSL for Materialized View Engine	Select the option to encrypt communication between clients and Materialized View Engine using Transport Layer Security (TLS) (formerly known as Secure Socket Layer (SSL)).
Materialized View Engine TLS/SSL Server JKS Keystore File Location	Path to the TLS/SSL keystore file containing the server certificate and private key used for TLS/SSL. Used when Materialized View Engine is acting as a TLS/SSL server. The keystore must be in JKS format.
Materialized View Engine TLS/SSL Server JKS Keystore File Password	Password for the Materialized View Engine JKS keystore file.
Materialized View Engine TLS/SSL Server JKS Keystore Key Password	Password that protects the private key contained in the JKS keystore.
Materialized View Engine TLS/SSL Client Trust Store File	Location of the truststore on disk. The truststore must be in JKS format. If this parameter is not provided, the default list of known certificate authorities is used instead.
Materialized View Engine TLS/SSL Client Trust Store Password	The password for the Materialized View Engine TLS/SSL Certificate truststore file. This password is not mandatory to access the truststore; this field is optional. This password provides optional integrity checking of the file. The contents of truststores are certificates, and certificates are public information.

Streaming SQL Console	
Enable TLS/SSL for Streaming SQL Console	Select the option to encrypt communication between clients and Streaming SQL Console using Transport Layer Security (TLS) (formerly known as Secure Socket Layer (SSL)).
Streaming SQL Console TLS/SSL Server Private Key File (PEM Format)	Path to the TLS/SSL file containing the private key used for TLS/SSL. Used when Streaming SQL Console is acting as a TLS/SSL server. The certificate file must be in PEM format.
Streaming SQL Console TLS/SSL Server Certificate File (PEM Format)	Path to the TLS/SSL file containing the server certificate key used for TLS/SSL. Used when Streaming SQL Console is acting as a TLS/SSL server. The certificate file must be in PEM format.
Streaming SQL Console TLS/SSL Server CA Certificate (PEM Format)	Path to the TLS/SSL file containing the certificate of the certificate authority (CA) and any intermediate certificates used to sign the server certificate. Used when Streaming SQL Console is acting as a TLS/SSL server. The certificate file must be in PEM format, and is usually created by concatenating all of the appropriate root and intermediate certificates.
Streaming SQL Console TLS/SSL Private Key Password	Password for the private key in the Streaming SQL Console TLS/SSL Server Certificate and Private Key file. If left blank, the private key is not protected by a password.

Streaming SQL Console	
Streaming SQL Console TLS/SSL Certificate Trust Store File	Location on disk of the truststore, in .pem format, used to confirm the authenticity of TLS/SSL servers that Streaming SQL Console might connect to. This is used when Streaming SQL Console is the client in a TLS/SSL connection. This truststore must contain the certificate(s) used to sign the service(s) connected to. If this parameter is not provided, the default list of known certificate authorities is used instead.

SQL Stream Engine	
Enable TLS/SSL for Streaming SQL Engine	Select the option to encrypt communication between clients and Streaming SQL Engine using Transport Layer Security (TLS) (formerly known as Secure Socket Layer (SSL)).
Streaming SQL Engine TLS/SSL Server JKS Keystore File Location	Path to the TLS/SSL keystore file containing the server certificate and private key used for TLS/SSL. Used when Streaming SQL Engine is acting as a TLS/SSL server. The keystore must be in JKS format.
Streaming SQL Engine TLS/SSL Server JKS Keystore File Password	Password for the Streaming SQL Engine JKS keystore file.
Streaming SQL Engine TLS/SSL Server JKS Keystore Key Password	Password that protects the private key contained in the JKS keystore used when Streaming SQL Engine is acting as a TLS/SSL server.
Streaming SQL Engine TLS/SSL Client Trust Store File	Location on disk of the truststore, in .jks format, used to confirm the authenticity of TLS/SSL servers that Streaming SQL Engine might connect to. This is used when Streaming SQL Engine is the client in a TLS/SSL connection. This truststore must contain the certificate(s) used to sign the service(s) connected to. If this parameter is not provided, the default list of known certificate authorities is used instead.
Streaming SQL Engine TLS/SSL Client Trust Store Password	Password for the Streaming SQL Engine TLS/SSL Certificate Trust Store file. This password is not mandatory to access the trust store; this field is optional. This password provides optional integrity checking of the file. The contents of truststores are certificates, and certificates are public information.

5. Click Save Changes.

Enabling TLS for database connection

You can enable encrypted communication between SSB and the configured databases using the Database connection property in Cloudera Manager.

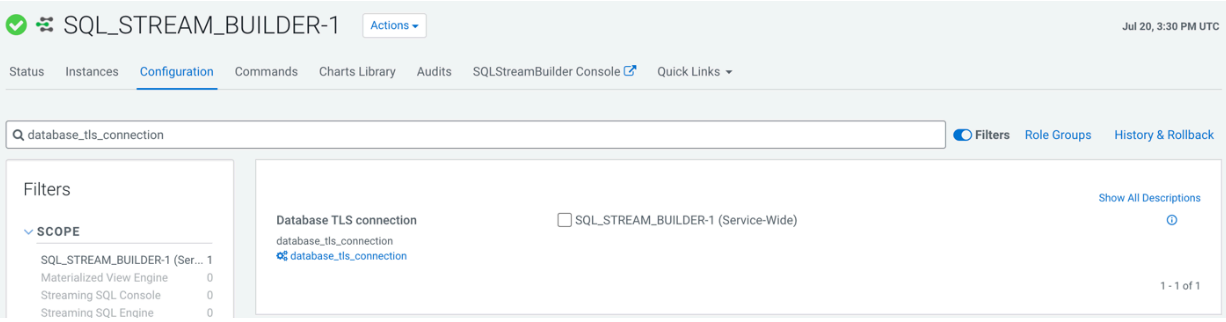
About this task

Note: When you are using Auto-TLS, the configuration for the Database TLS connection is not enabled automatically.

Procedure

1. Go to your cluster in Cloudera Manager.
2. Click SQL Stream Builder from the list of services.
3. Select Configuration tab.

4. Add database_tls_connection to the Search box.



database_tls_connection

Database TLS connection
database_tls_connection
database_tls_connection

□ SQL_STREAM_BUILDER-1 (Service-Wide)

5. Select Database TLS connection to enable encryption for the databases.
6. Click Save changes.
7. Restart the SQL Stream Builder service.
   a) Click Action > Restart next to the SQL Stream Builder service name.

Results

Encrypted communication is enabled between SSB and the configured databases.

What to do next

You must rotate the Certificate Authority and Host Certificates to update the security requirements of your cluster. For more information, see the Rotate Auto-TLS Certificate Authority and Host Certificates in Cloudera Manager documentation.

Configuring Ranger policies for SSB

You must add SQL Stream Builder (SSB) users to the Ranger policies that are used by Kafka, Schema Registry, Hive and Kudu to provide access to topics, schemas and tables provided by the components.

Before you begin

- Install Apache Ranger on your cluster. For more information, see the Production Installation documentation.
- Add the required Ranger policies for Flink. For more information, see Configuring Ranger policies for Flink documentation.

You can reach the Ranger User Interface through Cloudera Manager:

1. Go to your cluster in Cloudera Manager.
2. Select Ranger from the list of services.
3. Click on Ranger Admin Web UI.

   You are redirected to the Ranger Admin Web UI.

Adding SSB user to Kafka policies

You must add the SSB user to the following policies:

- all-consumergroup
- all-topic
- all-transationalid
- all-cluster
- all-delegationtoken

1. Select cm_kafka from the Service Manager home page on the Ranger Admin Web UI.

   You are redirected to the list of Kafka policies page.
2. Click on the edit button of the all-consumergroup policy.
3. Add the SSB user to the Select User field under the Allow Conditions setting.
4. Click Save.

   You are redirected to the list of Kafka policies page
5. Click on + More… to check if the SSB user is listed under the Users for the consumergroup policy.
6. Add the SSB user to the following policy with the above steps as well:

   - all-topic
   - all-transationalid
   - all-cluster
   - all-delegationtoken

Adding SSB user to Schema Registry policies

You must add the SSB user to the following policy:

- all-schema-group, schema-metadata, schema-branch, schema-version

1. Select cm_schema-registry from the Service Manager home page on the Ranger Admin Web UI.

   You are redirected to the list of Schema Registry policies page.
2. Click on the edit button of the all-schema-group, schema-metadata, schema-branch, schema-version policy.
3. Add the SSB user to the Select User field under the Allow Conditions setting.
4. Click Save.

   You are redirected to the list of Schema Registry policies page.
5. Click on + More… to check if the SSB user is listed under the Users for the schema-group, schema-metadata, schema-branch, schema-version policy.

Adding SSB user to Hive policies

You must add the SSB user to the following policy:

- all-global
- all-database, table, column
- all-database, table
- all-database
- all-hiveservice
- all-database, udf
- all-url

1. Select cm_hadoopsql from the Service Manager home page on the Ranger Admin Web UI.

   You are redirected to the list of Hadoop SQL policies page.
2. Click on the edit button of the all-global policy.
3. Add the SSB user to the Select User field under the Allow Conditions setting.
4. Click Save.

   You are redirected to the list of Hadoop SQL policies page.
5. Click on + More… to check if the SSB user is listed under the Users for the all-global policy.

6. Add the SSB user to the following policy with the above steps as well:

   - all-database, table, column
   - all-database, table
   - all-database
   - all-hiveservice
   - all-database, udf
   - all-url

Adding SSB user to Kudu policies

You must create a policy to grant access to Kudu tables for the SSB user.

1. Select cm_kudu from the Service Manager home page on the Ranger Admin Web UI.

   You are redirected to the Create Policy page.
2. Click on Add New Policy.
3. Provide a name to the Policy Name field.
4. Provide a prefix for the Databases you want to add to the policy or select all by typing *.
5. Provide a prefix for the table you want to add to the policy or select all by typing *.
6. Provide a prefix for the column you want to add to the policy or select all by typing *.
7. Add the SSB user to the Select User field under the Allow Conditions setting.
8. Click on the plus icon to Add Permissions to the Permissions field.
9. Click on the specific permissions or Select All.
10. Click on Add at the bottom of the page.

    You are redirected to the list of Kudu policies page where the created policy should be listed.
11. Click on + More… to check if the SSB user is listed under the Users for the created policy.

Managing teams in Streaming SQL Console

You can manage your team, team members and invite new team members under the Teams menu on the SQL Stream Builder console.

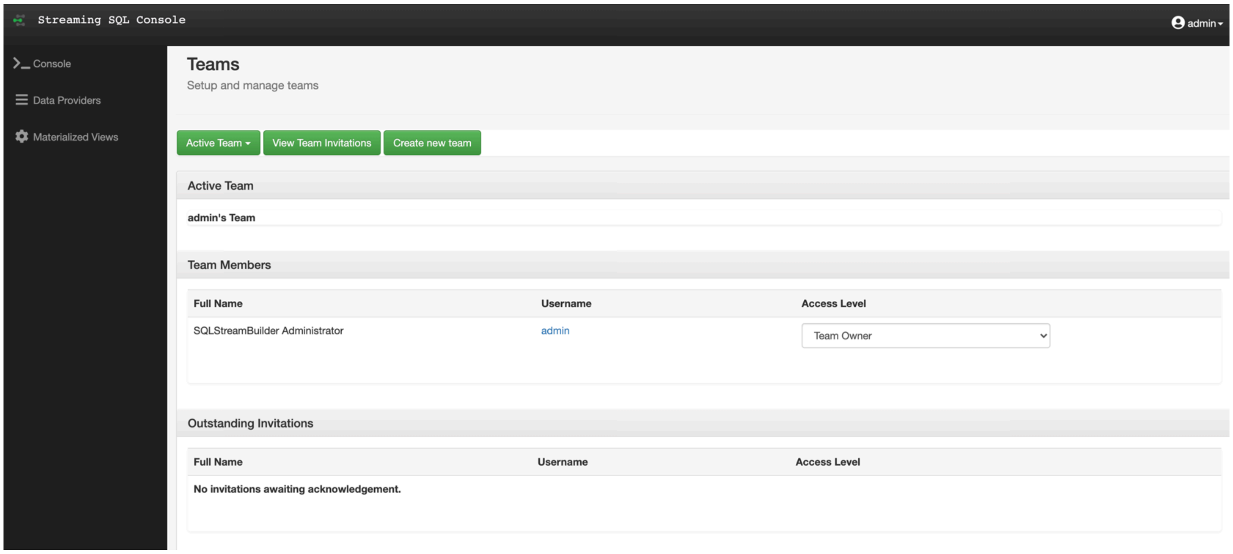
You can access the Teams menu using Streaming SQL Console:

1. Go to your cluster in Cloudera Manager.
2. Click SQL Stream Builder from the list of services.
3. Click SQLStreamBuilder Console.

   The Streaming SQL Console opens in a new window
4. Click on your username.

5. Click Teams.

You are redirected to the Teams page.



Teams

Setup and manage teams

Active Team

admin's Team

Team Members

Full Name	Username	Access Level
SQLStreamBuilder Administrator	admin	Team Owner

Outstanding Invitations

Full Name	Username	Access Level
No invitations awaiting acknowledgement.		

By default, a new user is assigned to the SQLStreamBuilder team which is a default team for the administrator within SSB. Every user who can access the Streaming SQL Console on the same cluster, is automatically added and listed as Team Members in the SQLStreamBuilder team. Only the administrator has the privilege to change the access level for a team member, and inactivate-activate a team member from the SQLStreamBuilder team. Team members can create their own team. In this case, only the Team Owner can delete their team. A team can be deleted by the Team Owner of that certain team. A team cannot be deleted if it is a primary team of a user or it is the default team (SQLStreamBuilder).

Every team member in a team (SQLStreamBuilder team or user created team) can access the created Virtual Tables, User Defined Functions, Materialized Views and API keys within a team. A team member can also view the jobs submitted in a team they are a member of. A user can be a part of multiple teams, and can switch between them. On the Streaming SQL Console, the currently selected team is shown under the Active Team header.

A team member can invite other members to join their team. The invitation can be accepted or ignored. To view the invitation within a team click the View Team Invitations button.