Cloudera Streaming Analytics 1.3.0

Manage SSB Security

Date published: 2019-12-17
Date modified: 2021-03-25

https://docs.cloudera.com/

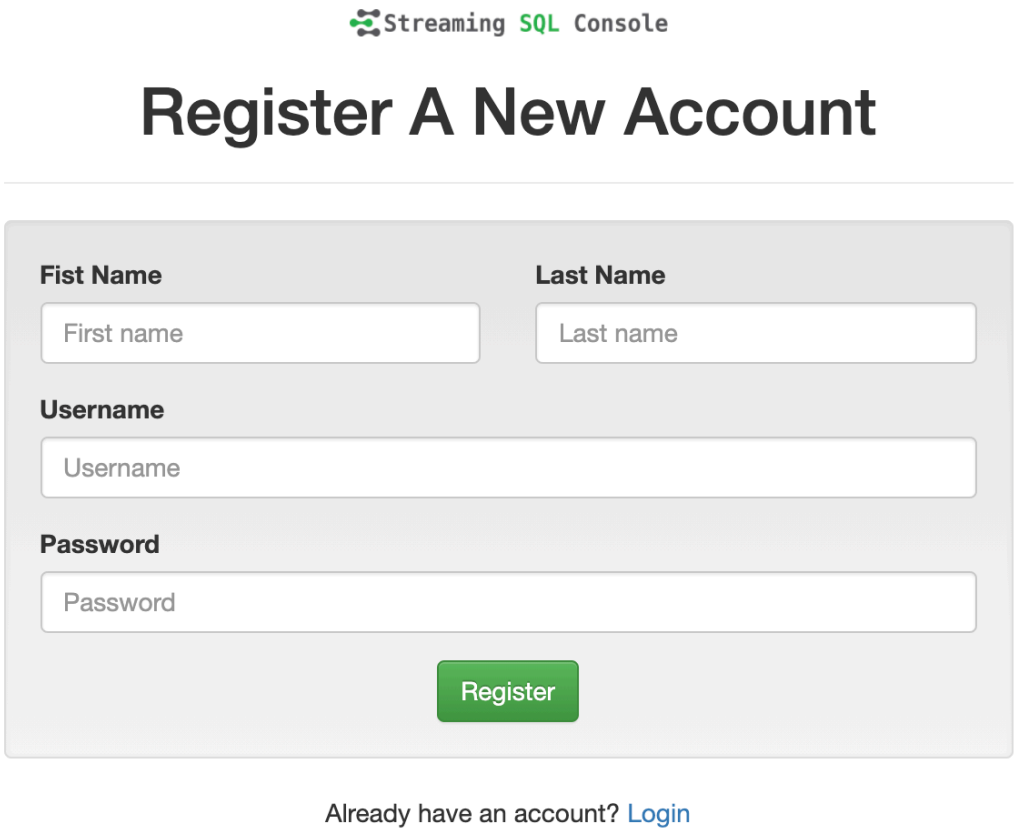
Legal Notice

Contents

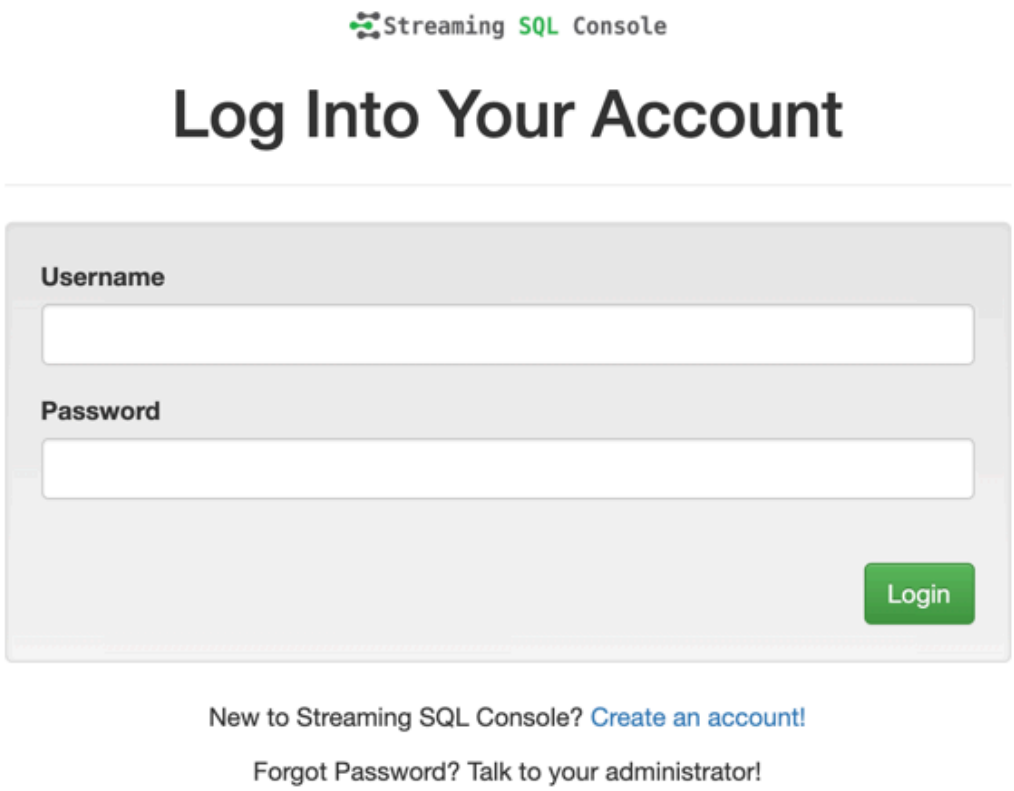
Authentication in SSB

You can use Kerberos authentication to secure your environment and your data in SQL Stream Builder (SSB).

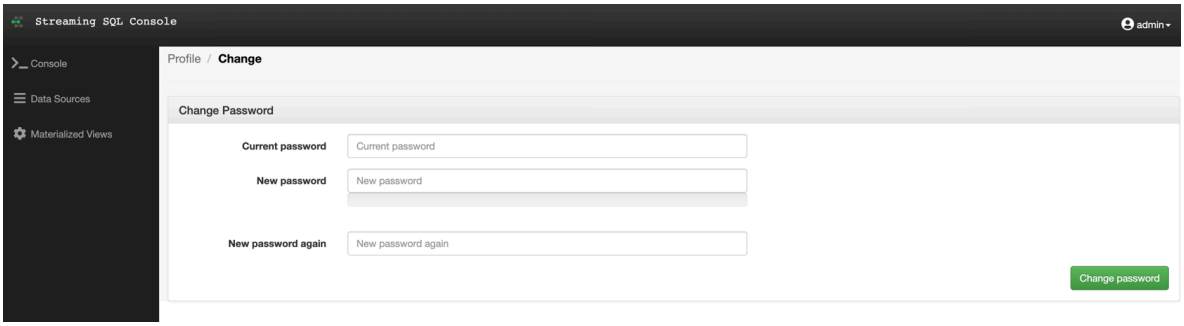
On an unsecured cluster, you need to register a new account to access the SSB Console. Provide your Name, Username and Password to create an account.



After creating an account, you are able to login to the Streaming SQL Console by providing the registered Username and Password.



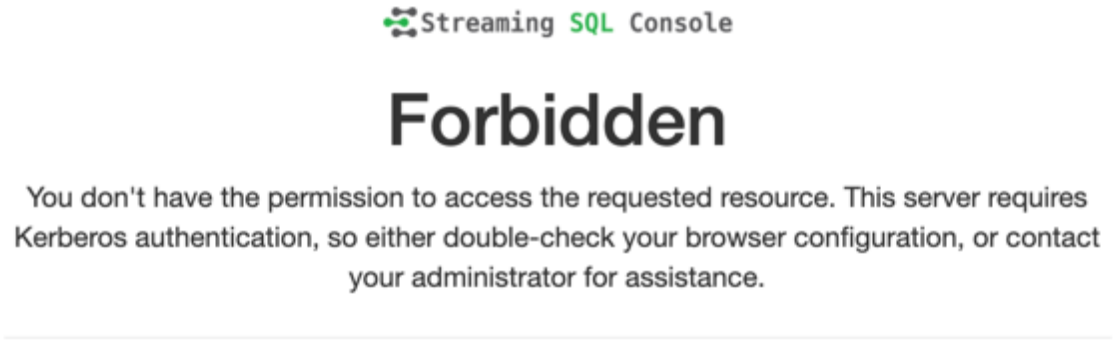
Note: You can later change your password by clicking on your username at the right top of the Streaming SQL Console, and under the Profile tab.



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.

When Kerberos authentication is set up for SSB, and a not authorized user wants to reach the Streaming SQL Console, the following error message appears:



Enabling Kerberos for authentication

1. Go to your cluster in Cloudera Manager.
2. Click on SQL Stream Builder from the list of Services.
3. Click the Configuration tab.
4. Select Category > Security .
5. Type kerberos to the search field.
6. Select the Enable Kerberos authentication setting.
7. Open a terminal window.
8. Copy and paste the following command:

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

Enabling keytabs

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

   The Streaming SQL Console opens up in a new window
4. Click on your username.
5. Click on Profile.
6. Click Unlock keytab.

Encryption in SSB

You need to configure the TLS/SSL properties for the SQL Stream Builder.

Before you begin

Ensure that you have set up TLS for Cloudera Manager:

- Generate TLS certificates
- Configure TLS for Admin Console and Agents
- Enable server certificate verification on Agents
- Configure agent certificate authentication

- Configure 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. Click 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 as well.

Materialized View Engine	
Enable TLS/SSL for Materialized View Engine	Select the checkbox 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 Trust Store on disk. The Trust Store must be in JKS format. If this parameter is not provided, the default list of well-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 Trust Store File. This password is not required to access the trust store; this field can be left blank. This password provides optional integrity checking of the file. The contents of trust stores are certificates, and certificates are public information.

Streaming SQL Console	
Enable TLS/SSL for Streaming SQL Console	Select the checkbox 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	
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 TLS/SSL Certificate Trust Store File	Location on disk of the trust store, 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 trust store must contain the certificate(s) used to sign the service(s) connected to. If this parameter is not provided, the default list of well-known certificate authorities is used instead.

SQL Stream Engine	
Enable TLS/SSL for Streaming SQL Engine	Select the checkbox 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 trust store, 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 trust store must contain the certificate(s) used to sign the service(s) connected to. If this parameter is not provided, the default list of well-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 required to access the trust store; this field can be left blank. This password provides optional integrity checking of the file. The contents of trust stores are certificates, and certificates are public information.

5. Click Save Changes.

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.

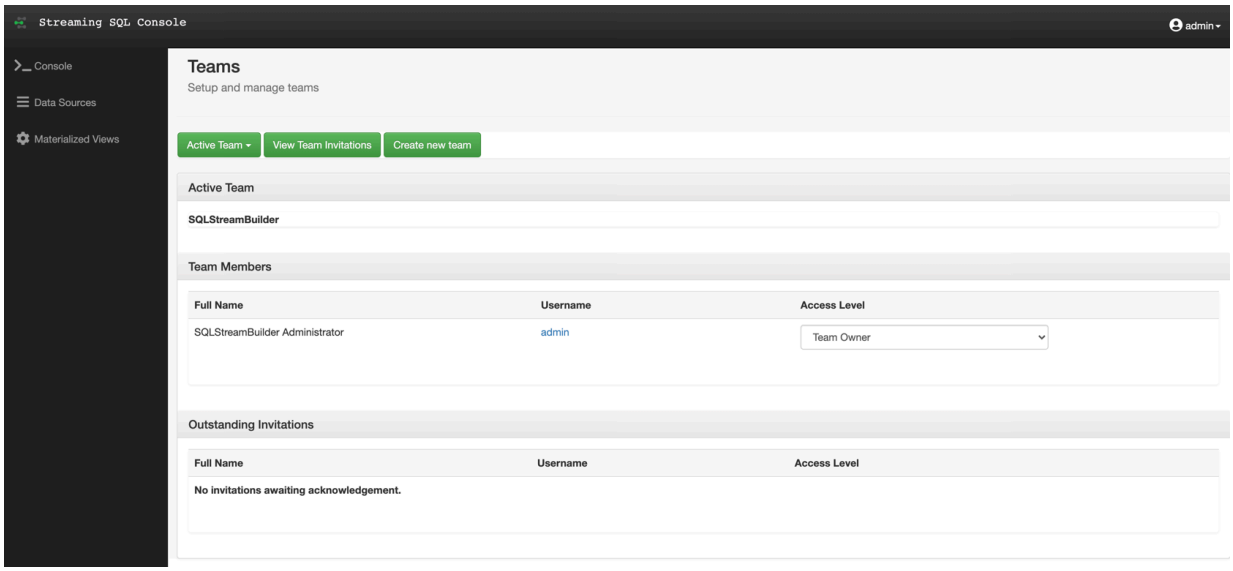
You can access the Teams menu thourhg Streaming SQL Console:

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

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

5. Click on Teams.

   You are redirected to the Teams page.



By default, a new user is assigned to the SQLStreamBuilder team which is a default team for the administrator within SSB. Every user that can access the Streaming SQL Console on the same cluster, will be 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 (this can be the 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 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 on the View Team Invitations button.