

Installation

Date published: 2019-12-17

Date modified: 2021-05-25



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

Installing CSD and parcel.....	4
Adding Flink as a Service.....	5
Setting up your HDFS Home directory.....	9
Setting the Java executable for the Flink client.....	9
Configuring databases for SSB.....	10
Setting up MySQL/MariaDB database for SSB.....	10
Installing the MySQL JDBC connector.....	10
Creating MySQL/MariaDB database for SSB.....	11
Installing the MySQL Python connector.....	11
Setting up PostgreSQL database for SSB.....	12
Installing the PostgreSQL JDBC connector.....	12
Creating PostgreSQL database for SSB.....	12
Installing the PostgreSQL Python connector.....	13
Setting up Oracle database for SSB.....	13
Installing the Oracle JDBC connector.....	14
Creating Oracle database for SSB.....	14
Installing the Oracle Python connector.....	14
Adding SSB as a service.....	15

Installing CSD and parcel

For installing Cloudera Streaming Analytics (CSA), you need to upload the downloaded Flink and SQL Stream Builder (SSB) Custom Service Descriptor (CSD) files to the default CSD directory, and add the CSA parcel to your cluster using Cloudera Manager.

Before you begin

- Download the Flink and SSB CSD and parcel files.

For more information about download Flink and SSB artifacts, see the [Download location](#) section.

- Install CDP Private Cloud Base.

For more information about installing CDP Private Cloud Base and Cloudera Manager, see the [CDP Private Cloud Base documentation](#).

Procedure

1. Place the CSD files in the `/opt/cloudera/csd/` folder (default CSD directory).
Cloudera Manager automatically detects the CSD files.

```
wget -P /opt/cloudera/csd/ https://user:password@archive.cloudera.com/p/csa/1.4.0.0/csd/FLINK-1.12-csa1.4.0.0-cdh7.1.6.0-297-13947709.jar
https://user:password@archive.cloudera.com/p/csa/1.4.0.0/csd/SQL_STREAM_B
UILDER-1.12-csa1.4.0.0-cdh7.1.6.0-297-13947709.jar
```

2. Change the ownership of the CSD files.

```
chown cloudera-scm:cloudera-scm FLINK-1.12-csa1.4.0.0-cdh7.1.6.0-297-139
47709.jar
```

```
chown cloudera-scm:cloudera-scm SQL_STREAM_BUILDER-1.12-csa1.4.0.0-cdh7.
1.6.0-297-13947709.jar
```

3. Restart Cloudera Manager and CMS services for the changes to take effect.

```
systemctl restart cloudera-scm-server
```

4. Log into Cloudera Manager.
5. Select Parcels on the Home > Hosts tab in the main navigation bar.
6. Click on Parcel Repositories & Network Settings tab.
7. Add the new Remote Parcel Repository URL for CSA.

```
https://user:password@archive.cloudera.com/p/csa/1.4/parcels/
```



Note: Make sure that the Remote Parcel Repository URL uses HTTPS link. To install a different version of the parcel, you can change the URL as needed.

8. Enter your download credentials to HTTP authentication username override for Cloudera Repositories and HTTP authentication password override for Cloudera Repositories.
9. Click Save & Verify Configuration to commit the change.
10. Click Close.
You are redirected to the Parcels page.
11. Search for Flink, and click Download to download the parcel to the local repository.
12. After the download is completed, click Distribute to distribute the parcel to all clusters.

13. After the parcel is distributed, click Activate to activate the parcel.

14. Click OK when confirmation is required.

For more details, follow the standard [procedure](#) from the GUI or the API.

Results

You have added the Flink and SSB CSD files, and the parcel to your cluster.

What to do next

Add Flink as a service in Cloudera Manager. After adding Flink as a service, you are able to add SSB service to your cluster.

Related Information

[Installing CDP Private Cloud Base](#)

[Add Flink as a Service](#)

Adding Flink as a Service

You need to use the Add Service wizard in Cloudera Manager to include the Flink service on your cluster. When assigning roles, you must install Flink, HDFS, and YARN roles on the same node from where the Flink jobs are submitted.

Before you begin

- Make sure that the Flink CSD file is in the `/opt/cloudera/csd` folder.
- Make sure that the CSA parcel is added to Cloudera Manager.
- Check that the following components are installed on your cluster:

Mandatory components	Optional components
YARN	Kafka
HDFS	HBase
Zookeeper	Schema Registry
	Streams Messaging Manager
	Kudu
	Hive
	Atlas

Procedure

1. Open Cloudera Manager.
2. On the Home screen, select the drop-down menu to the right of your cluster.
3. Select Add Service.
4. From the list, select Flink as the type of service, then click Continue.
The Add Service wizard is displayed.
5. Choose HBase and Hive as Optional dependency if needed for the source and sink solution, then click Continue.

6. Assign the Flink Dashboard role to one of the hosts, and the Flink Gateway role to every host.

**Note:**

Make sure that you installed Flink, HDFS, and YARN Gateway roles on the same machine that will be used to submit Flink jobs. The Flink Dashboard role also depends on having HDFS client configurations on the same machine. The HDFS client configurations can either be provided by an HDFS daemon role implicitly or can be deployed by an HDFS Gateway role explicitly.

7. Click Continue.

8. Review the changes needed for your service.

You can leave this page blank as the settings can be configured after the Flink service is added. You can configure the security settings at [Flink > Configuration > Security](#).

9. Click Continue and wait until the first run of the Flink service is completed.

10. Click Continue and then Finish.



CLOUDERA
Manager

Add Service to Clus

Results

You have added Flink as a service in Cloudera Manager.

Related Information

[Installing CSD and parcel](#)

[Add SSB as a Service](#)

Setting up your HDFS Home directory

You need an HDFS Home directory to store temporary logs and data of your application to run a Flink job. You must set up the HDFS Home directory for your user to avoid error when using Flink.

About this task

To run a Flink job, your HDFS Home directory has to exist. If it does not exist, you receive an error message similar to the following:

```
Permission denied: user=$USER_NAME, access=WRITE, inode="/user".
```

Procedure

Create HDFS Home directory. Ask your HDFS administrator to perform the following (or obtain [HDFS administrator role](#)).

Options**Command****Kerberos enabled**

```
> kinit hdfs
> hdfs dfs -mkdir /user/$USER_NAME
> hdfs dfs -chown $USER_NAME:$USER_NAME /user/$USER_NAME
```

Kerberos disabled

```
> HADOOP_USER_NAME=hdfs hdfs dfs -mkdir /user/$USER_NAME
> HADOOP_USER_NAME=hdfs hdfs dfs -chown $USER_NAME:$USER_NAME /user/$USER_NAME
```

In case of an enterprise environment, you can use Hue to set up the Home directory by enabling automatic synchronization for users. For more information, see the Cloudera Runtime [documentation](#).

Setting the Java executable for the Flink client

To avoid error when using Flink, you must set the Java_home environment manually through the command line for the Flink clients. The configuration in Cloudera Manager only applies to services, and not to clients.

About this task

Cloudera Manager offers a configuration for the JAVA_HOME environment variable under **Hosts > All Hosts > Configuration**. However, this only applies to services (for example YARN NodeManager or Flink HistoryServer) and does not propagate to clients such as the JVM created locally by the Flink executable. JVM uses the Bigtop utility under `/usr/bin/bigtop-detect-javahome` to automatically detect the JAVA_HOME.

Procedure

Set a fixed value for JAVA_HOME:

```
> cat /etc/default/bigtop-utils
export JAVA_HOME=/usr/java/default
```



Note:

Cloudera recommends providing the same value set in Cloudera Manager. It is also recommended to set it uniformly on all the nodes to avoid confusion. This is a known issue in the [Cloudera Community](#).

Configuring databases for SSB

Before adding SQL Stream Builder (SSB) as a service to your cluster, you need to manually configure the databases to use SSB. The databases are used to store the metadata information of the Streaming SQL Console and the Materialized Views.

SSB supports the following databases:

Database	Streaming SQL Console	Materialized View Engine
MySQL/MariaDB	Supported	Not supported
PostgreSQL	Supported	Supported
Oracle DB	Supported	Not supported

For more information about the supported versions of the databases, see the CSA-specific [System Requirements](#).

Setting up MySQL/MariaDB database for SSB

After installing MySQL/MariaDB server, you must rename the JDBC connector, and create a database with credentials for SQL Stream Builder (SSB) to be able to install the service on your cluster. You also must install the MySQL Python connector to integrate with the Streaming SQL Console.

Before you begin

You need to install and configure MySQL or MariaDB based on which one you plan to use, before setting up the databases for SSB. To install and configure MySQL or MariaDB, you must complete the basic steps mentioned in the Private Cloud Base documentation.

- [Install and configure MySQL for Cloudera Software](#)
- [Install and configure MariaDB for Cloudera Software](#)

Installing the MySQL JDBC connector

1. Download the MySQL JDBC Driver from the [MySQL website](#).
2. Extract the JDBC driver JAR file from the downloaded file with the following command:

```
tar zxvf mysql-connector-java-5.1.48.tar.gz
```

3. Rename the JDBC jar file to mysql-connector-java.jar.

```
mv mysql-connector-java-5.1.48-bin.jar mysql-connector-java.jar
```

4. Copy the MySQL JDBC jar file to /usr/share/java folder using the following command.

```
sudo mkdir -p /usr/share/java
```

```
sudo cp <location>/mysql-connector-java.jar /usr/share/java
```

5. Check if the MySQL connector is in the folder with ls command.

Creating MySQL/MariaDB database for SSB

1. Log in as the root user to MySQL:

```
mysql -u root -p  
Enter password:
```

2. Create databases for the Streaming SQL Console:

```
CREATE DATABASE ssb_admin DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_general_ci;
```

3. Grant all privileges for the database:

```
GRANT ALL ON ssb_admin.* TO 'ssb_admin'@'%' IDENTIFIED BY '<password>';
```

4. Confirm that you have created the Streaming SQL Console database:

```
SHOW DATABASES;
```

Installing the MySQL Python connector

Use the following sample commands to install the correct version for MySQL:

RHEL

1. Install the python-pip package:

```
sudo yum install python3-pip
```

2. Create a directory where you install the package:

```
mkdir -p /usr/share/python3
```

3. Install the 8.0.23 version for psycopg2 using pip:

```
sudo pip3 install mysql-connector-python==8.0.23 -t /usr/share/python3
```

Ubuntu

1. Install the python-pip package:

```
sudo apt-get install python3-pip
```

2. Create a directory where you install the package:

```
mkdir -p /usr/share/python3
```

3. Install the 8.0.23 version for psycopg2 using pip:

```
sudo pip3 install mysql-connector-python==8.0.23 -t /usr/share/python3
```

SLES 12

1. Install the python-pip package:

```
sudo zypper install python3-pip
```

2. Create a directory where you install the package:

```
mkdir -p /usr/share/python3
```

3. Install the 8.0.23 version for psycopg2 using pip:

```
sudo pip3 install mysql-connector-python==8.0.23 -t /usr/share/python3
```

Setting up PostgreSQL database for SSB

After installing PostgreSQL server, you must rename the JDBC connector, and create a database with credentials for SQL Stream Builder (SSB) before installing the service on your cluster. You also must install the PostgreSQL Python connector to integrate with the Streaming SQL Console.

Before you begin

You need to install and configure PostgreSQL, before setting up the databases for SSB. To install and configure PostgreSQL, you must complete the basic steps mentioned in the Private Cloud Base documentation.

- [Install and configure PostgreSQL for Cloudera Software](#)

Installing the PostgreSQL JDBC connector

1. Download the PostgreSQL JDBC Driver from the [PostgreSQL website](#).
2. Rename the JDBC jar file to postgresql-connector-java.jar.

```
mv postgresql-jdbc.jar postgresql-connector-java.jar
```

3. Copy the PostgreSQL JDBC jar file to /usr/share/java folder using the following command:

```
sudo mkdir -p /usr/share/java
sudo cp <location>/postgresql-connector-java.jar /usr/share/java
```

4. Check if the PostgreSQL connector is in the folder with ls command.

Creating PostgreSQL database for SSB

1. Connect to PostgreSQL:

```
sudo -u postgres psql
```

2. Create databases for the Streaming SQL Console:

```
CREATE ROLE ssb_admin LOGIN PASSWORD '<password>';
CREATE DATABASE ssb_admin OWNER ssb_admin ENCODING 'UTF8';
```

3. Create databases for the Materialized View Engine:

```
CREATE ROLE ssb_mve LOGIN PASSWORD '<password>';
CREATE DATABASE ssb_mve OWNER ssb_mve ENCODING 'UTF8';
```

4. Confirm that you have created the Streaming SQL Console and Materialized View Engine database using the \l command.

Installing the PostgreSQL Python connector

Use the following sample commands to install the correct version for PostgreSQL:

RHEL

1. Install the python-pip package:

```
sudo yum install python3-pip
```

2. Create a directory where you install the package:

```
mkdir -p /usr/share/python3
```

3. Install the 2.8.5 version for psycopg2 using pip:

```
sudo pip3 install psycopg2-binary==2.8.5 -t /usr/share/python3
```

Ubuntu

1. Install the python-pip package:

```
sudo apt-get install python3-pip
```

2. Create a directory where you install the package:

```
mkdir -p /usr/share/python3
```

3. Install the 2.8.5 version for psycopg2 using pip:

```
sudo pip3 install psycopg2-binary==2.8.5 -t /usr/share/python3
```

SLES 12

1. Install the python-pip package:

```
sudo zypper install python3-pip
```

2. Create a directory where you install the package:

```
mkdir -p /usr/share/python3
```

3. Install the 2.8.5 version for psycopg2 using pip:

```
sudo pip3 install psycopg2-binary==2.8.5 -t /usr/share/python3
```

Setting up Oracle database for SSB

After installing Oracle database server, you must rename the JDBC connector, and create a database with credentials for SQL Stream Builder (SSB) before installing the service on your cluster. You also must install the Oracle Python connector to integrate with the Streaming SQL Console.

Before you begin

You need to install and configure Oracle database, before setting up the databases for SSB. To install and configure Oracle DB, you must complete the basic steps mentioned in the Private Cloud Base documentation.

- [Install and configure Oracle Database for Cloudera Software](#)

Installing the Oracle JDBC connector

1. Download the Oracle JDBC Driver from the [Oracle website](#).
2. Rename the JDBC jar file to oracle-connector-java.jar.

```
mv ojdbc8-19.3.0.0.jar oracle-connector-java.jar
```

3. Copy the Oracle JDBC jar file to /usr/share/java folder using the following command:

```
sudo mkdir -p /usr/share/java
sudo cp <location>/oracle-connector-java.jar /usr/share/java
sudo chmod 644 /usr/share/java/oracle-connector-java.jar
```

4. Check if the Oracle connector is in the folder with ls command.

Creating Oracle database for SSB

1. Log in to the Oracle client:

```
sqlplus system@localhost
Enter password: *****
```

2. Create a user and schema for Streaming SQL Console:

```
create user ssb_admin identified by <password> default tablespace ssb_ad
min;
grant CREATE SESSION to ssb_admin;
grant CREATE TABLE to ssb_admin;
grant CREATE SEQUENCE to ssb_admin;
grant EXECUTE on sys.dbms_lob to ssb_admin;
```

3. Grant a quota on the ssb_admin tablespace where the tables will be created:

```
ALTER USER ssb_admin quota 100m on ssb_admin;
```

You can also create unlimited space with the following command:

```
ALTER USER ssb_admin quota unlimited on ssb_admin;
```

4. Confirm that you have created the Streaming SQL Console and Materialized View Engine database using the \l command.

Installing the Oracle Python connector

Use the following sample commands to install the correct version for Oracle:

RHEL

1. Install the python-pip package:

```
sudo yum install python3-pip
```

2. Create a directory where you install the package:

```
mkdir -p /usr/share/python3
```

3. Install the 2.8.5 version for pycopg2 using pip:

```
sudo pip3 install pycopg2-binary==8.1.0 -t /usr/share/python3
```

Ubuntu

1. Install the python-pip package:

```
sudo apt-get install python3-pip
```

2. Create a directory where you install the package:

```
mkdir -p /usr/share/python3
```

3. Install the 2.8.5 version for pycopg2 using pip:

```
sudo pip3 install pycopg2-binary==8.1.0 -t /usr/share/python3
```

SLES 12

1. Install the python-pip package:

```
sudo zypper install python3-pip
```

2. Create a directory where you install the package:

```
mkdir -p /usr/share/python3
```

3. Install the 2.8.5 version for pycopg2 using pip:

```
sudo pip3 install pycopg2-binary==8.1.0 -t /usr/share/python3
```

Adding SSB as a service

You need to use the Add Service wizard in Cloudera Manager to have the SQL Stream Builder (SSB) service on your cluster.

Before you begin

- Make sure that the SSB CSD file is in the /opt/cloudera/csd folder.
- Make sure that the CSA parcel is added to Cloudera Manager.
- Make sure that you have installed and configured the SSB databases correctly, and installed the required Python and Java drivers as well.
- Check that the following components are installed on your cluster:

Mandatory components	Optional components
Flink	Schema Registry
Kafka	Kudu
	Hive

Procedure

1. Open Cloudera Manager.
2. On the Home screen, select the drop-down menu to the right of your cluster.
3. Select Add Service.
4. From the list, select SQL Stream Builder as the type of service, then click Continue.
The Add Service wizard launches.

5. Assign the SQL Stream Engine, Materialized View Engine, and SQL Stream Console service roles to hosts, then click Continue.

You need to assign the service roles based on where you have created the databases, and where you have assigned the Flink and Kafka roles.

6. Connect the SSB service to a database.



Important: You must install and configure MySQL/MariaDB or PostgreSQL database before adding SSB as a service. In case you did not set up any database for your cluster, see the [Configuring databases for SSB](#) documentation.

- a) Select MySQL, PostgreSQL or Oracle as the type of database.



Important: If you plan to use Materialized Views in SSB, you must install PostgreSQL database.

- b) Choose the host on which you want to add the database.

You must add the databases to the same host where you have assigned the service roles.

- c) Provide a name to the database.
- d) Provide the user and password of the created database.
- e) Click Test connection.

Setup Database

Configure and test database connections. If using custom databases, create the databases first according to the **Installing and Configuring an External Database** section of the [Installation Guide](#).

SQL Stream Builder

✓ Successful

Type	Database Hostname	Database Name
<input type="text" value="PostgreSQL"/>	<input type="text" value="docs-test-1.vpc.cloudera.com"/>	<input type="text" value="ssb_admin"/>
Username	Password	
<input type="text" value="ssb_admin"/>	<input type="password" value="*****"/>	

☐ Show Password

Test Connection

7. Review the changes needed for your service.

In case you are using MySQL or Oracle for Streaming SQL Console, and PostgreSQL for Materialized Views, you will be prompted to provide information about the database for Materialized View Engine. Provide the database hostname with the default port, your user and password in the Materialized View Engine fields.

DB Connector Jar Directory db.connector.jar.dir db.connector.jar.dir	SQL Stream Builder (Service-Wide) ⓘ <input type="text" value="/usr/share/java/"/>
Database URL (JDBC) ssb.mve.datasource.url ssb.mve.datasource.url	Materialized View Engine Default Group ↻ Undo ⓘ <input type="text" value="jdbc:postgresql://docs-test-1.vpc.cloudera.com:5432/ssb_mve"/>
Database User ssb.mve.datasource.username ssb.mve.datasource.username	Materialized View Engine Default Group ↻ Undo ⓘ <input type="text" value="ssb_mve"/>
Database Password ssb.mve.datasource.password ssb.mve.datasource.password	Materialized View Engine Default Group ↻ Undo ⓘ <input type="password" value="*****"/>
Streaming SQL Console External Lib Path console.external.python.lib.path console.external.python.lib.path	Streaming SQL Console Default Group ⓘ <input type="text" value="/usr/share/python3"/>



Note: You can configure the security properties for SQL Stream Builder in this step, or after adding the service in the Configuration page. For more information about configuring security, see the Manage security documentation for SQL Stream.

8. Click Continue and wait until the first run of the SSB service is completed.

9. Click Continue and then Finish.

Related Information

[Installing CSD and parcel](#)

[Add Flink as a Service](#)

[Install and configure MySQL for Cloudera Software](#)

[Install and configure MariaDB for Cloudera Software](#)

[Install and configure PostgreSQL for Cloudera Software](#)

[Install and configure Oracle database for Cloudera Software](#)

[Configuring MySQL/MariaDB for SSB](#)

[Configuring PostgreSQL for SSB](#)

[Configuring Oracle database for SSB](#)