Cloudera Streaming Analytics 1.13.0

Release Notes

Date published: 2019-12-17
Date modified: 2024-08-07

https://docs.cloudera.com/

Legal Notice

Contents

What's new in Cloudera Streaming Analytics

Cloudera Streaming Analytics 1.15.1 covers new features beside the core streaming functionality of Apache Flink and SQL Stream Builder.

Cloudera platform support

Cloudera Streaming Analytics 1.15.1 is supported as a Long Term Support (LTS) version on Cloudera on premises 7.3.1.100 (CHF1). Ensure that you review the 7.3.1.100 (CHF1) Release Notes and Support Matrix to understand which operating system, database, and JDK versions are supported for Cloudera Streaming Analytics as well.

Rebase to Flink 1.20

Apache Flink 1.20.1 is supported in Cloudera Streaming Analytics.

For more information on what is included in the Apache Flink 1.20 version, see the Apache Flink 1.20 Release Announcement and Release Notes.

Support BATCH mode for PROD deployments in Cloudera SQL Stream Builder

Users can select and use batch as the runtime mode for production execution mode jobs that are running in an isolated Flink cluster. For more information, see Executing SQL jobs in production mode.

New OpenTelemetry Metrics Reporter [Technical Preview]

Cloudera Streaming Analytics includes, in Technical Preview, a new OpenTelemetry metrics reporter that makes it easier and more efficient to aggregate jobs' metrics to a central service like Prometheus or any OpenTelemetry-compatible service using open standards.

Refer to the documentation pages to learn more about using the OpenTelemetry reporter with Flink and Cloudera SQL Stream Builder Apache Flink documentation

Using Python UDFs for table transformations and webhook connector

Support in Python UDFs for table transformations and the webhook connector have been added to Cloudera SQL Stream Builder and Flink.

See Webhook connector.

Fixed issues

Review the list of Flink and Cloudera SQL Stream Builder issues that are resolved in Cloudera Streaming Analytics 1.15.1.

CSA-4889 - [ssb][flink] TUMBLE_ROWTIME column is not recognized as row time in subsequent query

CSA-5078 - Exclude `kafka-clients` from `flink-metrics-kafka`

CSA-5198 - The SSE Kafka AdminClient needs to configure truststore password to work with JDK 11 and above in secure clusters

CSA-5352 - flink hbase connector should not relocate Exception class

CSA-5473 - Search input gets focus by double shift even when editing queries

CSA-5561 - [ssb] Fix server sent events indefinitely holding db connections

CSA-5564 - Unlocking keytab may fail in SSB

CSA-5580 - Deletion of a running job is not handled properly.

CSA-5588 - Gracefully handle malformed UDFs

CSA-5591 - Renaming UDF does not work

CSA-5592 - SSB accepts invalid names for UDFs

CSA-5703 - Secure environment variables are broken

CSA-5711 - Environment importing loader is stuck when there is an error with importing

CSA-5727 - Cannot drop table via SQL if data source does not exist

CSA-5418 - Remove the "loadbalancer" entry from the SSB service keytab

CSA-5586 - 3rd party dependency upgrades

Deprecation notices in Cloudera Streaming Analytics 1.15.1

Certain features and functionalities have been removed or deprecated in Cloudera Streaming Analytics 1.15.1. You must review these items to understand whether you must modify your existing configuration. You can also learn about the features that will be removed or deprecated in the future release to plan for the required changes.

Terminology

Items in this section are designated as follows:

Deprecated

Technology that Cloudera is removing in a future Cloudera Streaming Analytics release. Marking an item as deprecated gives you time to plan for removal in a future Cloudera Streaming Analytics release.

Moving

Technology that Cloudera is moving from a future Cloudera Streaming Analytics release and is making available through an alternative Cloudera offering or subscription. Marking an item as moving gives you time to plan for removal in a future Cloudera Streaming Analytics release and plan for the alternative Cloudera offering or subscription for the technology.

Removed

Technology that Cloudera has removed from Cloudera Streaming Analytics and is no longer available or supported as of this release. Take note of technology marked as removed since it can potentially affect your upgrade plans.

Deprecation Notices for Cloudera SQL Stream Builder

Certain features and functionality are deprecated or removed in Cloudera SQL Stream Builder. You must review these changes along with the information about the features in Cloudera SQL Stream Builder that will be removed or deprecated in a future release.

Deprecated

v1 REST API

The v1 REST API for Cloudera SQL Stream Builder has been deprecated and will be removed in a future version of Cloudera Streaming Analytics.

Customers are advised to migrate to the v2 API, available for Cloudera SQL Stream Builder.

For more information on the v2 API, see Cloudera SQL Stream Builder REST API reference.

Support for JavaScript UDFs

Due to the deprecation of the Nashorn engine used in JDK 8 and 11, User-Defined Functions (UDFs) written in JavaScript has been deprecated in Cloudera Streaming Analytics 1.13.0. Cloudera recommends that customers start using Python UDFs for all new developments, and start migrating their JavaScript UDFs to Python to prepare for future upgrades.

Important: Due to the deprecation of Javascript for Cloudera Streaming Analytics, support in Python UDFs for table transformations and the webhook connector have been added to Cloudera SQL Stream Builder and Flink.

Known issues and limitations

Learn about the known issues in Flink and Cloudera SQL Stream Builder, the impact or changes to the functionality, and the workaround in Cloudera Streaming Analytics 1.15.1.

Cloudera SQL Stream Builder

CSA-5873 - Cloudera SQL Stream Builder environment variables cannot be used for Data Source validation/creation through the V2 API

Environment variables are not available for Data Source creation after an Environment is activated through the V2 API.

None. Data Sources that are using environment variables has to be created in the UI or through the V1 API.

CSA-5816 - Can't export a project that has secrets

When exporting a project to Github using Source ControlPush, if the project has a secret a warning displays, but selecting Force Push only shows an empty dialog and exporting the project isn't possible.

None.

CSA-5743 - Materialized View created from UDFs throws Invalid request error

Newly created UDFs are not recognized by Materialized View widget to analyze SQL operation. Clicking on the widget in Cloudera SQL Stream Builder displays SQL validation failed. Cannot instantiate user-defined function error if the UDF has not been added to the user session (starting a job with the UDF).

After creating a UDF, start a job with it to add it to the session.

CSA-5733 - Chart or diagram type dashboard widgets do not work when the label field is the same as the data field

When creating a diagram type widget in Cloudera SQL Stream Builder, setting the label and data fields to the same value causes the graph to disappear.

None.

CSA-5732 - MV widget not fetching mv data when authenticated via spnego

When using Cloudera SQL Stream Builder with SPNEGO authentication, creating a Materialized View widget fails with a Data Source Error.

None. Users are advised to authenticate through KNOX.

KNOX SSEDispatch does not work if HA is enabled for the service

By default, Cloudera SQL Stream Builder has a HA proxy configuration applied in KNOX, which is applied even when only one Cloudera SQL Stream Builder role instance exists. This breaks the async behavior of job sampling, because it uses a regular KNOX dispatch implementation.

Two workaround options exist:

1. Execute the following command on every KNOX node:

```
find /opt/cloudera/parcels/CDH/lib/knox/data/services -type
f -exec fgrep -l 'classname="org.apache.knox.gateway.sse.SSE
Dispatch"' {} \; | \
xargs sed -i.bak 's/classname="org.apache.knox.gateway.sse.
SSEDispatch"/classname="org.apache.knox.gateway.sse.SSEDispa
tch" ha-classname="org.apache.knox.gateway.sse.SSEDispatch"/'
```

2. Modify the SSB-SSE-UI service.xml on every KNOX instance via the KNOX Admin UI to have the below content:

```
<service name="ssb-sse-ui" role="SSB-SSE-UI" version="1.13.0">
   <metadata>
   <type>UI</type>
   <context>/ssb-sse-ui</context>
   <shortDesc>SQL Stream Builder UI</shortDesc>
   <description>Cloudera's SQL Stream Builder is an IDE and ma
nager tool for Flink SQL jobs.</description>
   </metadata>
   <dispatch classname="org.apache.knox.gateway.dispatch.Con
figurableDispatch" use-two-way-ssl="false">
   <param>
      <name>responseExcludeHeaders</name>
      <value>CONTENT-LENGTH,WWW-AUTHENTICATE</value>
   </param>
   <param>
      <name>httpclient.connectionTimeout</name>
      <value>5m</value>
   </param>
   <param>
      <name>httpclient.socketTimeout</name>
      <value>5m</value>
   </param>
   </dispatch>
   <routes>
   <route path="/ssb-sse-ui/"/>
   <route path="/ssb-sse-ui/**"/>
   <route path="/ssb-sse-ui/**?**"/>
   <route path="/ssb-sse-ui/swagger/**">
      <rewrite apply="SSB-SSE-UI/filter/outbound/swagger/body"
 to="response.body"/>
   </route>
   <route path="/ssb-sse-ui/**/event-stream/**">
      <rewrite apply="SSB-SSE-UI/ssb-sse-ui/inbound/event-st
ream" to="request.url"/>
```

Note: With the above workaround options sampling can be fixed for one instance, but in case of multiple instances, load-balancing will not work properly.

CSA-5747 - Upgrade causes attribute conversion failure when there are data sources containing secret properties

Due to the secret property encryption for data sources introduced in Cloudera Streaming Analytics 1.14.0, after upgrading from a version without the encryption (1.13.x or lower), data sources with secret properties may cause Cloudera Streaming Analytics not starting, or the Explorer Cloudera SQL Stream Builder failing to load resources properly.

1. Connect to the configured Cloudera SQL Stream Builder admin database. The database information can be found in Cloudera ManagerSQL Stream Builder ServiceConfiguration and filtering for database.

2. In the database, select the records of the data_sources table.
3. Remove all data_source entries that has properties containing keys matching the following keywords: secret, password, pwd, credentials, token, user-info, user.info.
4. Verify that the Cloudera SQL Stream Builder loads the resources properly.
5. Re-register the previously deleted data sources.

Note: Cloudera recommends that users delete data sources containing secrets **before upgrading** Cloudera Streaming Analytics, and re-register them after the upgrade is completed.

ENGESC-23078 - Job not found after successful job creation

After successfully creating a job in Cloudera SQL Stream Builder, the SQL job is not found due to tables having empty values. This issue is indicated with the following error message in the log files:

```
java.lang.IllegalArgumentException: argument "content" is null
```

The issue only applies when upgrading from a CSA version lower than 1.9.0.

Update the empty values with null string in the mv_config and checkpoint_config fields as shown in the following example:

```
UPDATE jobs SET mv_config = 'null' WHERE mv_config IS NULL;
UPDATE jobs SET checkpoint_config = 'null' WHERE checkpoint_conf
ig IS NULL;
```

CSA-4858 - Kerberos encryption type detection does not always work correctly for Cloudera SQL Stream Builder

SSB detects no supported encryption types even though there is a list of allowed encryption types in the krb5.conf file. This causes an error when generating keytabs from the principal and password pair.

1. Run ktutil on your cluster.
2. Change the configuration with the following commands:

```
addent -password -p [***USERNAME***] -k 1 -e aes256-cts
wkt /tmp/new_keytab.keytab
```

3. Upload the new keytab to Cloudera SQL Stream Builder.

Auto discovery is not supported for Apache Knox

You need to manually configure Knox with Cloudera SQL Stream Builder to enable Knox authentication. Complete the configuration based on the Cloudera Base on premises version you use. For more information, see the Enabling Knox authentication for Cloudera SQL Stream Builder documentation.

Flink

In Cloudera Streaming Analytics, the following SQL API features are in preview:

- Match recognize
- Top-N
- Stream-Table join (without rowtime input)

DataStream conversion limitations

- Converting between Tables and POJO DataStreams is currently not supported in Cloudera Streaming Analytics.
- Object arrays are not supported for Tuple conversion.

- The java.time class conversions for Tuple DataStreams are only supported by using explicit TypeInformation: LegacyInstantTypeInfo, LocalTimeTypeInfo.getInfoFor(LocalDate/LocalDateTime/LocalTime.class).
- Only java.sql.Timestamp is supported for rowtime conversion, java.time.LocalDateTime is not supported.

Kudu catalog limitations

- CREATE TABLE
  - Primary keys can only be set by the kudu.primary-key-columns property. Using the PRIMARY KEY constraint is not yet possible.
  - Range partitioning is not supported.
- When getting a table through the catalog, NOT NULL and PRIMARY KEY constraints are ignored. All columns are described as being nullable, and not being primary keys.
- Kudu tables cannot be altered through the catalog other than simply renaming them.

Schema Registry catalog limitations

- Currently, the Schema Registry catalog / format only supports reading messages with the latest enabled schema for any given Kafka topic at the time when the SQL query was compiled.
- No time-column and watermark support for Registry tables.
- No CREATE TABLE support. Schemas have to be registered directly in the SchemaRegistry to be accessible through the catalog.
- The catalog is read-only. It does not support table deletions or modifications.
- By default, it is assumed that Kafka message values contain the schema id as a prefix, because this is the default behaviour for the SchemaRegistry Kafka producer format. To consume messages with schema written in the header, the following property must be set for the Registry client: store.schema.version.id.in.header: true.

Improvements and Behavioral changes

Learn about the change in certain functionality of Flink and Cloudera SQL Stream Builder that has resulted in a change in behavior from the previously released version of Cloudera Streaming Analytics.

Improvements

Replaced the bahir-flink Kudu connector with the externalized flink-connector-kudu

The new Kudu connector is updated to the current Flink APIs, and also includes new features.

Highlights:

- Updated existing Table API stack, and DataStream Sink code
- Source API implementation that includes streaming mode support
- SQL option to configure Kudu FlushMode
- PRIMARY KEY definition as part of the SQL code

To see all the included changes and learn more about the Kudu connector, see the Apache release notes.

Added aes256-cts-hmac-sha1-96 enctype as default to Keytab service

The keytab generation through Cloudera SQL Stream Builder works out of the box in environments where the Kerberos environment is configured to use the aes256-cts encryption type.

Unified default Flink job log4j rolling configuration

For jobs deployed through both Cloudera SQL Stream Builder and Flink, the log4j configuration now by default:

- Applies compression
- Keeps the last 5 entries
- Triggers rollover when size reaches 100MB

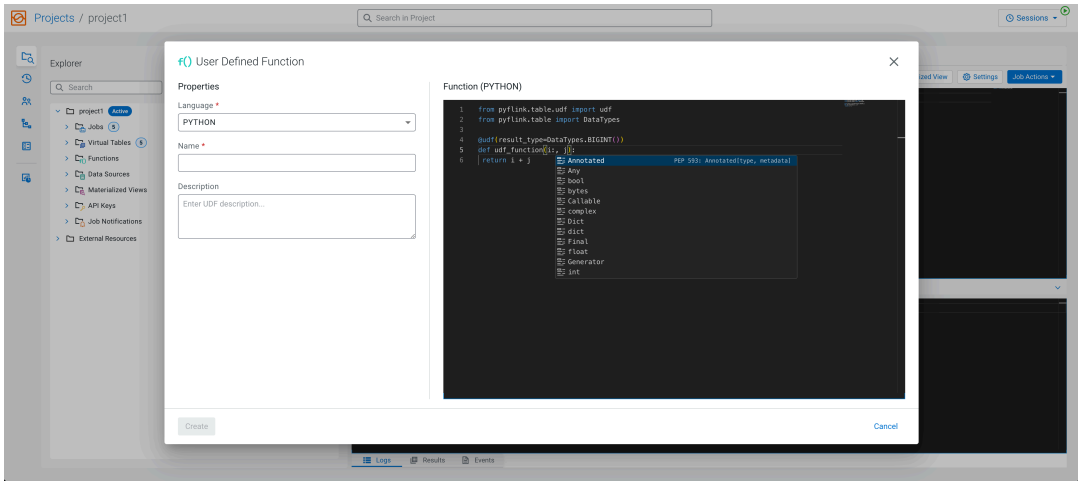
For more information on configuring logging, see Adjusting logging configuration in Advanced Settings.

Added metric reporter jars to parcel

All metrics reporter implementations available in the core vanilla Flink distribution are shipped with Cloudera Streaming Analytics. Customers can use them by modifying flink-conf.yaml. For more information, refer to Metric Reporters in the Apache Flink documentation.

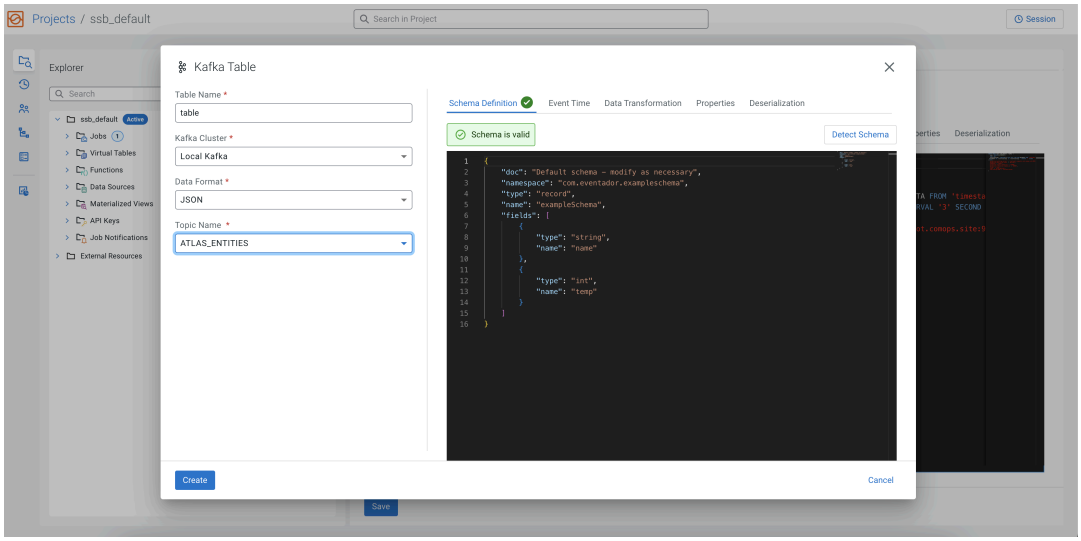
Enhanced Python editor in the UI

The code editor in Cloudera SQL Stream Builder UI is extended for better Python syntax checking and codestyle support.



Refactored schema detect UX for Kafka wizard tables

Users can refresh the detected schema using the UI. The new Avro schema is displayed after the new detection was successful.



Behavioral changes

Summary:

CSA-5473 - Search input gets focus by double shift even when editing queries

Previous behavior:

Pressing the shift key twice while editing SQL queries caused the search input to become active.

New behavior:

Pressing shift twice while the text input is in focus does not trigger autofocus on the search input.

Summary:

CSA-5616 - Increase metadata Column Size in MySQL Database to Support Larger Strings

Previous behavior:

The text type was used to store table metadata in MySQL.

New behavior:

Table metadata is now stored as longtext to accommodate longer strings.

Summary:

CSA-5694 - Modify text columns to longtext in the mysql schema of Cloudera SQL Stream Builder

Previous behavior:

The text type was used to store data in MySQL.

New behavior:

To avoid issues caused by too long strings, MySQL text columns now use the longtext type.

Unsupported features

Some Apache Flink and Cloudera SQL Stream Builder features exist in Cloudera Streaming Analytics 1.15.1, but are not supported by Cloudera. These features are not ready for production deployment, but Cloudera encourages you to explore them in non-production environments and provide feedback on your experiences through the Cloudera Community Forums.

Cloudera SQL Stream Builder

- Virtual environments for Python are not supported.
- Direct Cloudera SQL Stream Builder upgrade from 1.3.0

Important: This does not impact Flink, you can directly upgrade Flink as described in the documentation.

For more information, see the Upgrading SQL Stream Builder in the 1.3.0 documentation.

Flink

- Apache Flink batch (DataSet) API
- GPU Resource Plugin
- SQL Client
- The following features are not supported in SQL and Table API:
  - HBase Table Connector
  - Old Planner
  - Non-windowed (unbounded) joins, distinct

Support Matrix

Before installing Cloudera Streaming Analytics, review the supported components, databases, connectors and the default ports in use for Flink and Cloudera SQL Stream Builder.

Component support

Learn more about which Apache Flink component version is supported in the Cloudera Streaming Analytics releases.

Cloudera Streaming Analytics version	Component version
Cloudera Streaming Analytics 1.15.1	Apache Flink 1.20.1
Cloudera Streaming Analytics 1.14.0	Apache Flink 1.19.2
Cloudera Streaming Analytics 1.13.0	Apache Flink 1.19.1
Cloudera Streaming Analytics 1.12.0	Apache Flink 1.18.0
Cloudera Streaming Analytics 1.11.0	Apache Flink 1.16.2
Cloudera Streaming Analytics 1.10	Apache Flink 1.16.1
Cloudera Streaming Analytics 1.9.0	Apache Flink 1.15.1
Cloudera Streaming Analytics 1.8.0	
Cloudera Streaming Analytics 1.7.0	Apache Flink 1.14
Cloudera Streaming Analytics 1.6.2	
Cloudera Streaming Analytics 1.6.1	
Cloudera Streaming Analytics 1.6.0	
Cloudera Streaming Analytics 1.5.3	Apache Flink 1.13
Cloudera Streaming Analytics 1.5.1	
Cloudera Streaming Analytics 1.5.0	
Cloudera Streaming Analytics 1.4.1	Apache Flink 1.12
Cloudera Streaming Analytics 1.4.0	
Cloudera Streaming Analytics 1.3.0	
Cloudera Streaming Analytics 1.2.0	Apache Flink 1.10
Cloudera Streaming Analytics 1.1.0	Apache Flink 1.9.1

Related Information

Cloudera Streaming Analytics 1.16.0 Release Notes
Cloudera Streaming Analytics 1.15.0 Release Notes
Cloudera Streaming Analytics 1.15.0 Release Notes
Cloudera Streaming Analytics 1.14.0 Release Notes
Cloudera Streaming Analytics 1.13.2 Release Notes
Cloudera Streaming Analytics 1.13.2 Release Notes
Cloudera Streaming Analytics 1.13.2 Release Notes
Cloudera Streaming Analytics 1.13.1 Release Notes
Cloudera Streaming Analytics 1.13.0 Release Notes
Cloudera Streaming Analytics 1.12.0 Release Notes
Cloudera Streaming Analytics 1.11.0 Release Notes

Cloudera Streaming Analytics 1.10.0 Release Notes
Cloudera Streaming Analytics 1.9.0 Release Notes
Cloudera Streaming Analytics 1.8.0 Release Notes
Cloudera Streaming Analytics 1.7.0 Release Notes
Cloudera Streaming Analytics 1.6.2 Release Notes
Cloudera Streaming Analytics 1.6.1 Release Notes
Cloudera Streaming Analytics 1.6.0 Release Notes
Cloudera Streaming Analytics 1.5.3 Release Notes
Cloudera Streaming Analytics 1.5.1 Release Notes
Cloudera Streaming Analytics 1.5.0 Release Notes
Cloudera Streaming Analytics 1.4.1 Release Notes
Cloudera Streaming Analytics 1.4.0 Release Notes
Cloudera Streaming Analytics 1.3.0 Release Notes
Cloudera Streaming Analytics 1.2.0 Release Notes
Cloudera Streaming Analytics 1.1.0 Release Notes

System Requirements

Before installing Cloudera Streaming Analytics, you should verify that you meet the system requirements. Other than Cloudera Base on premises, you should also check the latest supported version of the needed components.

For detailed information about the supported versions of Cloudera Base on premises, operating systems and databases, see the Cloudera Support Matrix.

Apache Flink support	**1.20.1**
Cloudera Runtime component support in Cloudera Base on premises	**7.3.1**
Atlas	3.0.0
HBase	2.4.17
HDFS	3.1.1
Hive	3.1.3
Kafka[1]	3.4.1
Kudu	1.17.0
Schema Registry	0.10.0
Streams Messaging Manager	2.3.0
Apache Iceberg	1.3.0

Connector support	
JDBC PostgreSQL	9.6-16
JDBC MySQL	5.7, 8
JDBC Hive	3.1.3
JDBC Oracle	19, 19c, 21c, 23ai
JDBC Db2	11.5
JDBC SQL Server	2007-2022
CDC PostgreSQL	9.6-16

[1] Connecting to Kafka that is running on remote CDH6 or HDP3 is also supported.

Connector support	
CDC MySQL	5.7, 8
CDC Oracle	19, 19c, 21c, 23ai
CDC Db2	11.5
CDC SQL Server	2007-2022
Apache Iceberg	1.3.0

Default ports for Flink and Cloudera SQL Stream Builder

You need to use the default ports of Flink and Cloudera SQL Stream Builder when you need to reach or connect to their services. The default port are set in Cloudera Manager, but can be changed if required.

The following table lists the default ports and the corresponding property file names for Flink and Cloudera SQL Stream Builder. The ports are set by default in Cloudera Manager. You can change the ports as required using the configuration properties.

Component	Service	Port	Configuration property
Flink	Flink Dashboard	18211	historyserver.web.port
Cloudera SQL Stream Builder	Streaming SQL Engine	18121	server.port
	Materialized View Engine	18131	server.port
Cloudera SQL Stream Builder with Load Balancer	Streaming SQL Engine	8080	ssb.sse.loadbalancer.server.port
	Secured Streaming SQL Engine	8445	ssb.sse.loadbalancer.server.secure.port
	Materialized View Engine	8081	ssb.mve.loadbalancer.server.port
	Secured Materialized View Engine	8444	ssb.mve.loadbalancer.server.secure.port

For the default port list of the Cloudera Runtime components, see the *Ports Used by Cloudera Runtime Components* document.

Maven dependencies in Flink

Review the list of Maven dependencies to ensure the correct connector versions in your Flink applications.

Avro

```
<dependency>
  <groupId>org.apache.flink</groupId>
  <artifactId>flink-avro</artifactId>
  <version>1.20.1-csa1.15.1.0</version>
</dependency>
```

Confluent Registry

```
<dependency>
  <groupId>org.apache.flink</groupId>
  <artifactId>flink-connector-confluent-registry</artifactId>
  <version>1.0-csa1.15.1.0</version>
</dependency>
```

CSV

```
<dependency>
```

```
  <groupId>org.apache.flink</groupId>
  <artifactId>flink-csv</artifactId>
  <version>1.20.1-csa1.15.1.0</version>
</dependency>
```

Hive

```
<dependency>
  <groupId>org.apache.flink</groupId>
  <artifactId>flink-connector-hive_2.12</artifactId>
  <version>1.20.1-csa1.15.1.0</version>
</dependency>
```

HBase

```
<dependency>
  <groupId>org.apache.flink</groupId>
  <artifactId>flink-connector-hbase-1.4</artifactId>
  <version>3.0-csa1.15.1.0</version>
</dependency>
```

```
<dependency>
  <groupId>org.apache.flink</groupId>
  <artifactId>flink-connector-hbase-2.4</artifactId>
  <version>3.0-csa1.15.1.0</version>
</dependency>
```

Iceberg

```
<dependency>
  <groupId>org.apache.iceberg</groupId>
  <artifactId>iceberg-flink-runtime-1.16</artifactId>
  <version>1.3.0.7.1.9.1-158</version>
</dependency>
```

JDBC

```
<dependency>
  <groupId>org.apache.flink</groupId>
  <artifactId>flink-connector-jdbc</artifactId>
  <version>3.2-csa1.15.1.0</version>
</dependency>
```

JSON

```
<dependency>
  <groupId>org.apache.flink</groupId>
  <artifactId>flink-json</artifactId>
  <version>1.20.1-csa1.15.1.0</version>
</dependency>
```

Kafka

```
<dependency>
  <groupId>org.apache.flink</groupId>
  <artifactId>flink-connector-kafka</artifactId>
  <version>3.3-csa1.15.1.0</version>
</dependency>
```

Kudu

```
<dependency>
  <groupId>org.apache.bahir</groupId>
  <artifactId>flink-connector-kudu_2.12</artifactId>
  <version>1.1.0-csa1.15.1.0</version>
</dependency>
```

Schema Registry

```
<dependency>
  <groupId>org.apache.flink</groupId>
  <artifactId>flink-connector-cloudera-registry</artifactId>
  <version>1.0-csa1.15.1.0</version>
</dependency>
```

Table API

```
<dependency>
  <groupId>org.apache.flink</groupId>
  <artifactId>flink-table-api-java-bridge</artifactId>
  <version>1.20.1-csa1.15.1.0</version>
</dependency>
<dependency>
  <groupId>org.apache.flink</groupId>
  <artifactId>flink-table-planner_2.12</artifactId>
  <version>1.20.1-csa1.15.1.0</version>
</dependency>
```

For more information about how to use Maven in Flink, see the Apache documentation.

Flink API Support

Cloudera Streaming Analytics offers support for three fundamental layers of the Apache Flink API. You can use DataStream API, the ProcessFunction API and a selected subset of the SQL API to develop your Flink streaming applications.

From the DataStream and ProcessFunction APIs, the following are supported based on the support annotations provided by the Apache Flink community.

Stable (@Public)	Evolving (@PublicEvolving)
• DataStream API	• ProcessFunction • Stream Join • Interval Join • Stateful operators • FsStatebackend with HDFS • RocksDBStateBackend with HDFS

Note: Cloudera Streaming Analytics does not support batch processing (DataSet API).