Cloudera Streaming Analytics 1.16.0

Release Notes

Date published: 2019-12-17
Date modified: 2025-10-24

CLOUDERA

Legal Notice

Contents

What's new in Cloudera Streaming Analytics

Cloudera Streaming Analytics 1.16.0 covers new features beside the core streaming functionality of Apache Flink and SQL Stream Builder.

Cloudera platform support

Cloudera Streaming Analytics 1.16.0 is supported on Cloudera on premises 7.3.1.500 (SP3). Ensure that you review the 7.3.1.500 (SP3) Release Notes and the Support Matrix to understand which operating systems, databases, and JDK versions are supported for Cloudera Streaming Analytics as well.

Python UDFs enabled by default

The Python User Defined Functions feature is enabled by default in new Cloudera Streaming Analytics deployments, simplifying Python environment setup across all nodes.

For more information about working with Python UDFs, see Adding Python UDFs.

Removed v1 API from Cloudera SQL Stream Builder

The v1 REST API for Cloudera SQL Stream Builder is removed from Cloudera Streaming Analytics. Migrate integrations to the v2 API to continue managing SQL Stream Builder resources.

See the Cloudera SQL Stream Builder REST API reference for v2 API details.

Fixed issues

Review the list of Flink and Cloudera SQL Stream Builder issues that are resolved in Cloudera Streaming Analytics 1.16.0.

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

CSA-5782 - AnalyzeCatalogWrapper throws false errors when applying functions to the GenericInMemoryCatalog

CSA-5816 - Cannot export a project that has secrets

CSA-5847 - Default database name of Kudu Catalogs break existing queries after upgrade

CSA-5851 - Users can execute jobs when Kerberos is enabled without a keytab

CSA-5852 - User can execute jobs with CMD + Enter when keytab is not unlocked

CSA-5854 - SSB fails to delete MV endpoint

CSA-5858 - V2 execute API shouldn't allow session mode execution

CSA-5879 - TableProviderService uses the wrong project when getting table filters

Deprecation notices in Cloudera Streaming Analytics 1.16.0

Certain features and functionalities have been removed or deprecated in Cloudera Streaming Analytics 1.16.0. 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
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 are deprecated in Cloudera Streaming Analytics. Cloudera recommends that you create new functionality with Python UDFs and start migrating existing JavaScript UDFs to Python.

> Support in Python UDFs for table transformations and the webhook connector is available for Cloudera SQL Stream Builder and Flink.

Removed
Cloudera SQL Stream Builder v1 REST API

> The v1 REST API for Cloudera SQL Stream Builder is removed in Cloudera Streaming Analytics 1.16.0 and later releases. Migrate scripts and integrations to the v2 API.

> See the Cloudera SQL Stream Builder REST API reference for v2 API details.

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 1.16.0.

Cloudera SQL Stream Builder
CSA-5873 - Cloudera SQL Stream Builder environment variables cannot be used through the V2 API

> Environment variables are not being added to the session after activation.

> 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.

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
CSA-5542 - Enabled Python UDFs by default

Python User Defined Functions are enabled by default, replacing JavaScript UDFs. This streamlines the setup of a functional Python environment across all nodes.

CSA-5765 - Ensure OpenTelemetry dependencies are included in the Cloudera Streaming Analytics parcel

All dependencies that are required to configure the OpenTelemetry exporter ship with the parcels, so you no longer need additional application or environment dependencies.

CSA-5809 - Added discriminator mappings to all classes with @JsonSubTypes

The @Schema annotation is added to classes that use @JsonSubTypes so that Python code generation can deserialize API responses correctly.

Behavioral changes
Summary:

CSA-5372 - Use Apache Kerby's KrbConfig for more robust encryption type configuration and refactored keytab validation

Previous behavior:

The final Kerberos encryption type that Cloudera SQL Stream Builder used when generating a keytab followed this order:

1. Values extracted from the keytab provided in the source.enctype.keytab configuration property.
2. Default values defined in Cloudera SQL Stream Builder.

New behavior:

The final encryption type is now determined in the following order:

1. Values defined in the krb5.conf file.
2. Values extracted from the keytab provided in the source.enctype.keytab configuration property.

> **Note:** This method is deprecated and will be removed in a future release.

3. Default values defined in Cloudera SQL Stream Builder.

The default encryption types in Cloudera SQL Stream Builder are:

- aes256-cts-hmac-sha1-96
- aes128-cts-hmac-sha1-96
- des3-cbc-sha1
- arcfour-hmac
- des-hmac-sha1
- des-cbc-md5
- des-cbc-crc

Unsupported features

Some Apache Flink and Cloudera SQL Stream Builder features exist in Cloudera Streaming Analytics 1.16.0, 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 1.16.0	Apache Flink 1.20.1
Cloudera Streaming Analytics 1.15.1	Apache Flink 1.20.0
Cloudera Streaming Analytics 1.15.0	Apache Flink 1.20.0
Cloudera Streaming Analytics 1.14.1	Apache Flink 1.19.1
Cloudera Streaming Analytics 1.14.0	Apache Flink 1.19.1
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 version	Component version
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 1.15.1 Release Notes
Cloudera Streaming Analytics 1.15.0 Release Notes
Cloudera Streaming Analytics 1.14.1 Release Notes
Cloudera Streaming Analytics 1.14.0 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	1.5.2

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
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

---

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

Component	Service	Port	Configuration property
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-csa1.16.0.0</version>
</dependency>
```

Confluent Registry

```
<dependency>
   <groupId>org.apache.flink</groupId>
   <artifactId>flink-connector-confluent-registry</artifactId>
   <version>1.0-csa1.15.0.0</version>
</dependency>
```

CSV

```
<dependency>
   <groupId>org.apache.flink</groupId>
   <artifactId>flink-csv</artifactId>
   <version>1.20.1-csa1.16.0.0</version>
</dependency>
```

Hive

```
<dependency>
   <groupId>org.apache.flink</groupId>
   <artifactId>flink-connector-hive_2.12</artifactId>
   <version>1.20.1-csa1.16.0.0</version>
</dependency>
```

HBase

```
<dependency>
   <groupId>org.apache.flink</groupId>
   <artifactId>flink-connector-hbase-1.4</artifactId>
   <version>3.0-csa1.15.0.0</version>
</dependency>
```

```
<dependency>
   <groupId>org.apache.flink</groupId>
   <artifactId>flink-connector-hbase-2.4</artifactId>
```

```
  <version>4.0-csa1.16.0.0</version>
</dependency>
```

Iceberg

```
<dependency>
  <groupId>org.apache.iceberg</groupId>
  <artifactId>iceberg-flink-runtime-1.16</artifactId>
  <version>1.5.2.7.3.1.500-182</version>
</dependency>
```

JDBC

```
<dependency>
  <groupId>org.apache.flink</groupId>
  <artifactId>flink-connector-jdbc</artifactId>
  <version>3.2-csa1.15.0.0</version>
</dependency>
```

JSON

```
<dependency>
  <groupId>org.apache.flink</groupId>
  <artifactId>flink-json</artifactId>
  <version>1.20.1-csa1.16.0.0</version>
</dependency>
```

Kafka

```
<dependency>
  <groupId>org.apache.flink</groupId>
  <artifactId>flink-connector-kafka</artifactId>
  <version>3.3-csa1.15.0.0</version>
</dependency>
```

Kudu

```
<dependency>
  <groupId>org.apache.bahir</groupId>
  <artifactId>flink-connector-kudu_2.12</artifactId>
  <version>1.1.0-csa1.15.0.0</version>
</dependency>
```

Schema Registry

```
<dependency>
  <groupId>org.apache.flink</groupId>
  <artifactId>flink-connector-cloudera-registry</artifactId>
  <version>1.0-csa1.15.0.0</version>
</dependency>
```

Table API

```
<dependency>
  <groupId>org.apache.flink</groupId>
  <artifactId>flink-table-api-java-bridge</artifactId>
  <version>1.20.1-csa1.16.0.0</version>
</dependency>
<dependency>
  <groupId>org.apache.flink</groupId>
  <artifactId>flink-table-planner-loader-bundle</artifactId>
```

```
    <version>1.20.1-csa1.16.0.0</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).