Cloudera Streaming Analytics 1.4.0

Release Notes

Date published: 2019-12-17
Date modified: 2021-05-25

https://docs.cloudera.com/

Legal Notice

Contents

CVE-2021-45105 & CVE-2021-44832 remediation for CSA

Learn more about the CVE-2021-45105 and CVE-2021-44832 remediation for Flink and SQL Stream Builder in Cloudera Streaming Analytics (CSA).

Cloudera released maintenance versions for CSA on CDP Private Cloud Base to address CVE-2021-45105 and CVE-2021-44832 identified as critical vulnerability issues for Log4j2.

Cloudera encourages users to upgrade to the following CSA versions to avoid any possibility of exploitation:

- CVE-2021-45105: CSA 1.5.3 or higher version
- CVE-2021-44832: CSA 1.6.1

For more information about the impacts of CVE-2021-45105, see the TSB 2021-547: Critical vulnerability in log4j2 CVE-2021-45105 Knowledge Base article.

Related Information

CSA 1.5.3 Installation guide

CSA 1.6.1 Installation guide

What's new in Cloudera Streaming Analytics

Cloudera Streaming Analytics 1.4.0 covers new features beside the core streaming functionality of Apache Flink.

Cloudera Streaming Analytics 1.4.0 introduces the following key features:

- Oracle DB support
- SLES 12 support
- Flink SQL DDL and Catalog support
- Improved Kafka and Schema Registry integration
- Stream enrichment from Hive, Kudu and JDBC connectors
- Improved table management
- Custom connector support

Component support

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

CSA version	Component version
CSA 1.4.0	Apache Flink 1.12
CSA 1.3.0	
CSA 1.2.0	Apache Flink 1.10
CSA 1.1.0	Apache Flink 1.9.1

Related Information

CSA 1.3.0 Release Notes

CSA 1.2.0 Release Notes

CSA 1.1.0 Release Notes

Known issues and limitations

The following limitations apply to Cloudera Streaming Analytics 1.4.0

SQL Stream Builder

CSA-1023: SQL Stream jobs with large schemas fail when using MySQL

The SQL Stream jobs that have large schemas will fail when you configure SQL Stream Builder with MySQL database. The following error message appears when you run into this issue:

```
_mysql_connector.MySQLInterfaceError: Data too long for column '
sb_job_data' at row
        114:14
```

The MySQL ‘text’ data columns are limited to 64kb length. Make sure that the schema either does not exceed this value, or use the following workaround to change the ‘text’ data type to ‘longtext’ which has 4GB length.

1. Log in as the root user to MySQL:

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

2. Use the following ALTER TABLE command to change the data types into ‘longtext’:

   ```
   ALTER TABLE sb_jobs MODIFY sb_job_data LONGTEXT
   ```

3. Update the schemas in your SQL jobs:
   a. Open the Streaming SQL Console.
   b. Select Table tab.
   c. Search for the Kafka table where you want to update the schema.
   d. Click Edit.
   e. Select Schema tab.
   f. Click Detect schema.
   g. Click Save changes.
4. Restart your SQL jobs.

CSA-1232: Big numbers are incorrectly represented on the Streaming SQL Console UI

The issue impacts the following scenarios in Streaming SQL Console:

- When having integers bigger than 253-1 among your values, the Input transformations and User Defined Functions are considered unsafe and produce incorrect results as these numbers will lose precision during parsing.
- When having integers bigger than 253-1 among your values, sampling to the Streaming SQL Console UI produces incorrect results as these numbers will lose precision during parsing.

None

CSA-1378: Spring cleanup can cause exemptions and failure in SQL Stream Builder

Due to the cleanup mechanism of the Spring Boot framework used in SSB, the /tmp folder is cleared within certain dates on RHEL7 and Ubuntu. The cleanup removes every artifact stored by Spring. This can cause exemptions and job failures when using SSB.

None

CSA-1410: Restoring SSB job from savepoint fails when using MySQL

Restarting a SQL job from savepoint can fail when using MySQL database due to an issue of log creation.

None

CSA-1454: Timezone settings can cause unexpected behavior in Kafka tables

You must consider the timezone settings of your environment when using timestamps in a Kafka table as it can affect the results of your query. When the timestamp in a query is identified with from_unixtime, it returns the results based on the timezone of the system. If the timezone is not set in UTC+0, the timestamp of the query results will shift in time and will not be correct.

Change your local timezone settings to UTC+0.

CSA-1479: Incorrect Materialized View settings when loading SQL jobs

When editing an already existing SQL Stream job, the primary key and recreate table settings will not be correct and revert to default.

None

CSA-1499: Table name error for Materialized Views

When selecting data from a table, the Materialized View engine returns an error due to using the wrong table name in the execution.

None

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 CSA.
- 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.

Fixed Issues

Review the list of Flink and SQL Stream Builder issues that are resolved in Cloudera Streaming Analytics 1.4.0

SQL Stream Builder

CSA-1180: SSB generates directory in HDFS

The issues regarding the automatically created directory in HDFS has been fixed.

CSA-1037: Schema Detection fails with invalid key

The issue regarding the schema detection failure due to invalid key has been fixed.

Unsupported features

The following features are not supported in Cloudera Streaming Analytics 1.4.0

SQL Stream Builder

- Direct SQL Stream Builder upgrade from 1.3.0

  Important: This does not impact Flink, you can directly upgrade Flink from 1.3.0 to 1.4.0 as described in the documentation.

  For more information, see the Upgrading SQL Stream Builder section.
- INSERT INTO statements are not supported for SQL Stream Builder. Cloudera recommends to use sink tables instead of them.

Flink

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

Deprecation notices in Apache Flink

Certain features and functionality in Flink are deprecated or removed in Cloudera Streaming Analytics 1.4.0. You must review these changes along with the information about the features in Flink that will be removed or deprecated in a future release.

Removed

The containerized.heap-cutoff-ratio and containerized.heap-cutoff-min options have been removed in upstream Apache Flink. These options have no affect anymore.