Cloudera Streaming Analytics - Kubernetes Operator 1.3.0 Release Notes

Date published: 2024-06-15
Date modified: 2024-02-28

CLOUDERA

Legal Notice

Contents

Cloudera Streaming Analytics - Kubernetes Operator 1.3.0

Learn about the new features, improvements, known and fixed issues, limitations, and unsupported features in this release of Cloudera Streaming Analytics - Kubernetes Operator.

What's new

Learn about the new features and notable changes in this release of Cloudera Streaming Analytics - Kubernetes Operator.

Cloudera Streaming Analytics - Kubernetes Operator 1.3.0

To learn more about Cloudera Streaming Analytics - Kubernetes Operator and its typical deployment architecture, see Overview and for the installation instructions, see Installation.

This release of the Cloudera Streaming Analytics - Kubernetes Operator is based on the Apache Flink Kubernetes Operator (Flink Operator) 1.11.0 and Apache Flink (Flink) 1.20. For more information about the supported versions, see the following upstream resources:

- Flink Operator 1.11.0
- Flink 1.20

Flink

Rebase to Flink 1.20

Apache Flink 1.20 is supported in Cloudera Streaming Analytics - Kubernetes Operator 1.3.0

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.

Rebase to Flink Operator 1.11.0

Apache Flink Operator 1.11.0 is supported in Cloudera Streaming Analytics - Kubernetes Operator 1.3.0

For more information on what is included in the Apache Flink Operator 1.11.0 version, see the Apache Flink 1.11.0 Release Announcement and Release Notes.

Cloudera SQL Stream Builder

Flink OpenTelemetry Metrics Reporter (Technical Preview)

Cloudera Streaming Analytics - Kubernetes Operator 1.3.0 includes, in Technical Preview, a new OpenTelemetry metrics reporter that makes it easier and more efficient to aggregate job metrics to a central service like Prometheus or any OpenTelemetry-compatible service to aggregate metrics using open standards.

To learn more about using the OpenTelemetry reporter with Flink, see Using the OpenTelemetry Collector [Technical Preview] and the Apache Flink documentation.

Flink image with Hadoop, Hive, Iceberg, and Kudu connectors

Cloudera Streaming Analytics - Kubernetes Operator now offers an alternative image that includes the Hadoop, Hive, Iceberg, and Kudu connectors, as well as all required dependencies, in the Flink image.

For more information, refer to Installation.

Related Information
LDAP authentication
Routing with ingress

Fixed issues and improvements

Learn about the fixed issues, improvements, and changes functionality in this release of Cloudera Streaming Analytics - Kubernetes Operator.

Fixed issues

CSA-5622: Project sync doesn't sync Kubernetes config for jobs

Fixes an issue where the project synchronization mechanism in Cloudera Streaming Analytics - Kubernetes Operator did not correctly sync Kubernetes configurations for Flink jobs.

CSA-5751: Bug in SecuredPropertyConverter

Fixes an issue where the SecuredPropertyConverter class caused an infinite encryption loop when its convertToDatabaseColumn method mutated the attribute argument, resulting in exceptions when the data became too long.

CSA-5770: Fix intermittent Job Not Found error while deploying Flink jobs

Fixes an issue with prematurely failed Flink jobs due to recoverable errors to ensure jobs only fail if they reach a globally terminal state or encounter an unrecoverable error.

CSA-5780: Upgrade jackson-databind in flink-cdc to version 2.13.4.2 or above

Upgrades outdated jackson-databind versions in transitive dependencies for Flink artifacts.

CSA-5703: Secure environment variables are broken

Fixes an issue with secure environment variables inadvertently overwriting their original values masking process, leading to data source validation failures and infinite encryption loops.

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

Fixes an issue where the environment importing loader remained stuck after an import operation failed and users confirmed the error pop-up.

CSA-5658: Postgresql does not start when installing the operator on RKE2

Fixes an issue where the Postgresql init container failed to start for non-root users when installing Cloudera Streaming Analytics - Kubernetes Operator operator on RKE2 due to lack of permissions for the ssb_admin user.

Improvements

Unify default Flink job log4j rolling configuration

⚠️ **Warning:** Causes behavior change.

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.

Configurable checkpoint and savepoint directories in the Cloudera SQL Stream Builder UI

The savepoint and checkpoint paths are displayed and editable in the Job Settings window.

Ensure OpenTelemetry dependencies are included in Cloudera Streaming Analytics - Kubernetes Operator

Cloudera Streaming Analytics - Kubernetes Operator includes all required dependencies to configure the OTel exporter.

To enable the Flink OpenTelemetry Metrics Exporter for a working Flink Java application, launch the application with the following additional FlinkDeployment options:

```
spec:
  flinkConfiguration:
    metrics.reporters: otel
    metrics.reporter.otel.factory.class: org.apache.flink.metric
s.otel.OpenTelemetryMetricReporterFactory
    metrics.reporter.otel.exporter.endpoint: [*** ENDPOINT
 ***]:[*** PORT ***]
    metrics.reporter.otel.service.name: [*** SERVICE NAME ***]
    metrics.reporter.otel.service.version: [*** VERSION ***]
```

To enable the Flink OpenTelemetry Metrics Exporter for a job in Cloudera SQL Stream Builder, add the following SET commands to the job:

```
set metrics.reporters=otel;
set metrics.reporter.otel.factory.class=org.apache.flink.metrics
.otel.OpenTelemetryMetricReporterFactory;
set metrics.reporter.otel.exporter.endpoint=[*** ENDPOINT
 ***]:[*** PORT ***];
set metrics.reporter.otel.service.name=[*** SERVICE NAME ***];
set metrics.reporter.otel.service.version=[*** VERSION ***];
```

For more information, see Using the OpenTelemetry Collector [Technical Preview].

More flexible error handling when executing jobs in Cloudera SQL Stream Builder

Implementing more flexible error handling for jobs executing on Kubernetes. Unrecoverable error checks parse errors to determine if the Flink job's main method was the source of an exception. Cloudera SQL Stream Builder jobs are only marked as failed if their state reaches a globally terminal state or if the error explicitly indicates that they're unrecoverable.

Configurable Kafka sampling settings at runtime in Cloudera SQL Stream Builder

⚠️ **Warning:** Causes behavior change.

Users can add the settings of the sampling Kafka instance on the global configuration UI in Cloudera SQL Stream Builder, previously only configurable through configuration files, which will be passed to Flink jobs.

The settings are only configurable in Cloudera SQL Stream Builder UI, and settings added to the configuration YAML files will be ignored.

To learn how to configure sampling in Cloudera SQL Stream Builder, see Sampling.

Temporary table and view support

Users can create and manage temporary tables and views in Cloudera SQL Stream Builder to define temporary objects suitable for session-scoped operations or intermediate results. Temporary tables and views simplify complex SQL queries and making job definitions more modular without affecting the global catalog.

To create a temporary table:

```
CREATE TEMPORARY TABLE [*** TABLE NAME ***] (
    id INT,
    name STRING,
    event_time TIMESTAMP(3)
) WITH (
    'connector' = 'datagen',
    'rows-per-second' = '1',
    'fields.id.kind' = 'sequence',
    'fields.id.start' = '1',
    'fields.id.end' = '100',
    'fields.name.length' = '10'
);
```

To create a temporary view:

```
CREATE TEMPORARY VIEW [*** VIEW NAME ***] AS
SELECT
    order_id,
    SUM(amount) AS total_amount
FROM
    [*** PERSISTENT TABLE NAME ***]
GROUP BY
    order_id;
```

Known issues and limitations

Learn about the known issues and limitations in this release of Cloudera Streaming Analytics - Kubernetes Operator.

Stuck session jobs in Cloudera Streaming Analytics - Kubernetes Operator

Session jobs stop running if the session cluster's Job Manager is restarted without High Availability configured. However, because of a Flink bug, such stopped jobs get stuck in RECONCILING/ STABLE state and cannot be restarted or deleted.

In such cases, the following is seen when using the kubectl get FlinkSessionJobs -n flink command:

```
kubectl get FlinkSessionJobs -n flink
NAME                          JOB STATUS     LIFECYCLE STATE
ssb-ssbdefault-testjobname    RECONCILING    STABLE
```

1. Remove the `FlinkDeployment` resource related to SSB:

```
kubectl -n [***NAMESPACE***] delete flinkdeployment [***DEPL
OYMENT NAME***]
```

2. Open the SSB configurations to edit:

```
kubectl -n [***NAMESPACE***] edit cm ssb-config
```

3. Add the highlighted line to the security-context.yaml entry:

```
allowPrivilegeEscalation: false
  capabilities:
    drop:
    - ALL
  runAsNonRoot: true
  seccompProfile:
```

```
      type: RuntimeDefault
    runAsUser: 9999
```

It can take a couple of minutes for the changes to take effect after modifying the `ConfigMap`.

4. Execute the SQL job from Cloudera SQL Stream Builder again.

FLINK-33536: S3 filesystem sink and CSV format throws error

When using the Flink Table API CSV streaming sink with the S3 filesystem, the operation fails with IOException: Stream closed.

Cloudera SQL Stream Builder

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

Cloudera SQL Stream Builder 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 on Streaming SQL Console.

Flink

DataStream conversion limitations

- Converting between Tables and POJO DataStreams is currently not supported in Cloudera Streaming Analytics - Kubernetes Operator.
- 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.

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 behavior 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.

Unsupported features

Learn what features are unsupported in this release of Cloudera Streaming Analytics - Kubernetes Operator.

Some Apache Flink and Cloudera SQL Stream Builder features exist in Cloudera Streaming Analytics - Kubernetes Operator, but are not supported by Cloudera. The following features are not ready for production deployment.

Cloudera encourages you to explore these features in non-production environments and provide feedback on your experiences through the Cloudera Community Forums.

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

Cloudera SQL Stream Builder

- Materialized Views are not supported
- Temporary views and tables are not supported
- Job names over 45 characters (including all unique tags, for example ssb-*[*** PROJECT NAME ***]-[*** JOB NAME ***]*) are not supported
- Virtual environments for Python are not supported

Deprecation notices in Cloudera Streaming Analytics - Kubernetes Operator

Certain features and functionalities have been removed or deprecated in Cloudera Streaming Analytics - Kubernetes Operator. 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 - Kubernetes Operator release. Marking an item as deprecated gives you time to plan for removal in a future Cloudera Streaming Analytics - Kubernetes Operator release.

Moving

> Technology that Cloudera is moving from a future Cloudera Streaming Analytics - Kubernetes Operator 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 - Kubernetes Operator release and plan for the alternative Cloudera offering or subscription for the technology.

Removed

> Technology that Cloudera has removed from Cloudera Streaming Analytics - Kubernetes Operator 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 - Kubernetes Operator.

> 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 - Kubernetes Operator 1.2.0. 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.

Component support

You can review the Cloudera Streaming Analytics - Kubernetes Operator components and their versions shipped in this release of theCloudera Streaming Analytics - Kubernetes Operator.

Table 1: Cloudera Streaming Analytics - Kubernetes Operator component versions

Component	Version
Flink	1.20-csaop1.3.0-b40
Flink Operator	1.11-csaop1.3.0-b40
Cloudera SQL Stream Builder	1.20-csaop1.3.0-b40
OpenJDK	11.0.24

Supported Flink versions

Cloudera Streaming Analytics - Kubernetes Operator supports the following Flink versions:

Table 2: Supported Flink versions

Latest (default)	Other
1.20-csaop1.3.0-b40	None

The default version is the Cloudera-recommended current and latest supported Flink version. This version is used by default to deploy clusters if an explicit version is not provided in your `FlinkDeployment` resource.

Notice that the Flink versions are specific to Cloudera. Their version numbers consist of two parts: the first three digits specify the Apache Flink version, and the following specify the major and minor version of Cloudera Streaming Analytics - Kubernetes Operator. When deploying a cluster, you must use the Cloudera versions for Flink listed here. Specifying upstream versions is not supported.

Component support

Flink file system support

> By default, Cloudera Streaming Analytics - Kubernetes Operator only supports local and locally mounted NFS/SAN file systems for Flink. Pluggable file systems (for example S3, HDFS, etc.) can be used by adding plugins to the Apache Flink Operator. For more information and a list of supported pluggable file systems, see *Using pluggable file systems*.

Cloudera SQL Stream Builder database support

> You can use the following databases with Cloudera SQL Stream Builder:

- MySQL/MariaDB
- Oracle
- PostgreSQL

Cloudera SQL Stream Builder connector support

> With Cloudera SQL Stream Builder, you will get the following connectors installed by default:

- Kafka
- JDBC
- CDC (MySQL, Oracle, Postgres, Db2, SqlServer)
- Amazon S3
- Azure Blob Storage
- Google Cloud Storage
- HDFS (when using the alternative image, see Installation overview)
- Hive (when using the alternative image, see Installation overview)
- Iceberg (when using the alternative image, see Installation overview)
- Kudu (when using the alternative image, see Installation overview)

> You will also get the following formats installed with Cloudera SQL Stream Builder:

- JSON
- Avro
- ORC
- Parquet

Related Information

Using pluggable file systems

System requirements

Cloudera Streaming Analytics - Kubernetes Operator requires a Kubernetes cluster environment that meets the provided system requirements and prerequisites. You must ensure to meet these requirements to be able to install and use the Cloudera Streaming Analytics - Kubernetes Operator or its components.

- A Kubernetes 1.25 or later cluster:

  - OpenShift 4.12 or later
  - Any Cloud Native Computing Foundation (CNCF) certified Kubernetes distribution. For more information, see cncf.io.
- Administrative rights on the Kubernetes cluster.
- Access to kubectl or oc. These command line tools must be configured to connect to your running cluster.
- Access to helm.

- Log collection is enabled for the Kubernetes cluster.

  Cloudera requires that the logs of Cloudera Streaming Analytics - Kubernetes Operator components are stored long term for diagnostic and supportability purposes. Review Log collection.