StreamSets Data Collector 3.3.0 Release Notes

May 24, 2018

We're happy to announce a new version of StreamSets Data Collector. This version contains several new features, enhancements, and some important bug fixes.

This document contains important information about the following topics for this release:

- Upgrading to Version 3.3.0
- New Features and Enhancements
- Fixed Issues
- Known Issues

Upgrading to Version 3.3.0

You can upgrade previous versions of Data Collector to version 3.3.0. For complete instructions on upgrading, see the Upgrade Documentation.

Upgrade to Spark 2.1 or Later

Data Collector version 3.3.0 introduces cluster streaming mode with support for Kafka security features such as SSL/TLS and Kerberos authentication using Spark 2.1 or later and Kafka 0.10.0.0 or later.

However, this means that using Spark 1.x for cluster streaming mode, the Spark Evaluator processor, and the Spark executor was deprecated as of version 3.2.0.0. Support for Spark 1.x is now removed in version 3.3.0. If you are using cluster streaming mode, the Spark Evaluator processor, or the Spark executor, you must upgrade to Spark 2.1 or later in version 3.3.0. In addition, if you are using cluster streaming mode for Kafka, you must also upgrade to Kafka 0.10.0.0 or later.

Note: You can continue to use Kafka 0.9.0.0 in standalone pipelines. Or you can continue to use an earlier version of Data Collector to use Kafka 0.9.0.0 in cluster pipelines until you can upgrade Kafka.

Since Spark 1.x is no longer supported and since Kafka 0.9.0.0 is no longer supported in cluster pipelines, the following stage libraries have changed:

Category	Stage Library
New stage libraries	The following new stage libraries include the Kafka Consumer origin for cluster mode pipelines: • streamsets-datacollector-cdh-spark_2_1-lib • streamsets-datacollector-cdh-spark_2_2-lib • streamsets-datacollector-cdh-spark_2_3-lib

Changed stage libraries	The following stage library no longer includes the Kafka Consumer origin for cluster mode pipelines: ● streamsets-datacollector-hdp_2_4-lib The following stage libraries were upgraded to use Spark 2.1: ● streamsets-datacollector-hdp_2_6-lib ● streamsets-datacollector-mapr_5_2-lib ● streamsets-datacollector-mapr_6_0-mep4-lib
Removed stage libraries	The following stage libraries are removed: ● streamsets-datacollector-cdh_5_8-cluster-cdh_kafka_2_0-lib ● streamsets-datacollector-cdh_5_9-cluster-cdh_kafka_2_0-lib ● streamsets-datacollector-cdh_5_10-cluster-cdh_kafka_2_1-lib ● streamsets-datacollector-cdh_5_11-cluster-cdh_kafka_2_1-lib ● streamsets-datacollector-cdh_5_12-cluster-cdh_kafka_2_1-lib ● streamsets-datacollector-cdh_5_13-cluster-cdh_kafka_2_1-lib ● streamsets-datacollector-cdh_5_14-cluster-cdh_kafka_2_1-lib During the upgrade process, these removed stage libraries are replaced with the new streamsets-datacollector-cdh-spark_2_1-lib stage library.
Removed legacy stage libraries	The following legacy stage libraries are removed: ● streamsets-datacollector-cdh_5_4-cluster-cdh_kafka_1_2-lib ● streamsets-datacollector-cdh_5_4-cluster-cdh_kafka_1_3-lib ● streamsets-datacollector-cdh_5_5-cluster-cdh_kafka_1_3-lib ● streamsets-datacollector-cdh_5_7-cluster-cdh_kafka_2_0-lib
Changed legacy stage libraries	The following legacy stage libraries no longer include the Spark Evaluator processor: ● streamsets-datacollector-cdh_5_4-lib ● streamsets-datacollector-cdh_5_5-lib

To continue to use cluster streaming mode, you must upgrade to a newer Cloudera CDH or Hortonworks Hadoop distribution and to Kafka 0.10.0.0 or later in Data Collector version 3.3.0. The major Hadoop distribution vendors provide a means for Spark 1.x and Spark 2.x to coexist on the same cluster, so you can use both versions in your clusters. Data Collector supports the following Spark 2.x versions for the Hadoop distribution vendors:

- **Cloudera** - Cloudera Distribution of Spark 2.1 release 1 or later is supported. For more information, see Spark 2 Requirements.
- **Hortonworks** - Hortonworks Data Platform (HDP) 2.6 or later includes Spark 2.2.0. For more information, see the HDP 2.6 Release Notes.
- **MapR** - MapR with MapR Expansion Pack 3.0 or later is supported. For more information, see MEP Support by MapR Core Version.

Then, you must configure upgraded pipelines to work with the upgraded system, as described in Working with Upgraded External Systems.

In addition to selecting the upgraded stage library version for each stage that connects to the upgraded CDH, HDP, or Kafka system, you might need to perform additional tasks for the following stages:

- **Spark Evaluator processor** - If the Spark application was previously built with Spark 2.0 or earlier, you must rebuild it with Spark 2.1. Or if you used Scala to write the custom Spark class, and the application was compiled with Scala 2.10, you must recompile it with Scala 2.11.
- **Spark executor** - If the Spark application was previously built with Spark 2.0 or earlier, you must rebuild it with Spark 2.1 and Scala 2.11.

New Features and Enhancements

This version includes new features and enhancements in the following areas.

Cluster Pipelines

When using Spark 2.1 or later and Kafka 0.10.0.0 or later in a cluster pipeline that reads from a Kafka cluster on YARN, you can now enable the pipeline to use Kafka security features such as SSL/TLS and Kerberos authentication.

Origins

- **WebSocket Client origin enhancement** - You can now configure the origin to send an initial message or command after connecting to the WebSocket server.

Processors

- **New SQL Parser processor** - A processor that parses SQL queries. For example, if you set the Parse SQL Query property to false in the Oracle CDC origin, the origin writes the SQL query to an “sql” field that can be parsed by the SQL Parser.
- **Field Zip processor enhancement** - The Continue option for the Field Does Not Exist property is now named Include without Processing.

Pipelines

- **Notifications** - You can now configure a pipeline to send an email or webhook when the pipeline changes to the Stop_Error state.

- **Preview** - The default value of the Preview Timeout property has been increased to 30,000 milliseconds. Previously the default was 10,000 milliseconds.

Edge Pipelines

- **Sensor Reader origin enhancement** - This development stage can now generate records with thermal data such as that generated by BCM2835 onboard thermal sensors.

Stage Libraries

This version of Data Collector includes several new, changed, and removed stage libraries because of the introduction of cluster streaming mode with support for Kafka security features using Spark 2.1 or later and Kafka 0.10.0.0 or later.

For more information about the changed stage libraries, see Upgrade to Spark 2.1 or Later.

Fixed Issues

The following table lists some of the known issues that are fixed with this release.

For the full list, click here.

JIRA	Description
SDC-8955	Control Hub jobs might fail to restart because Data Collector can retain stale pipelineState.json files in the runInfo directory.
SDC-8897	Cluster mode pipelines fail when configured to use a credential store.
SDC-8892	The Value Replacer processor performs slower than it did in version 2.5.1.1.
SDC-8877	Support bundle thread dumps are truncated.
SDC-8874	The protobuf data format does not correctly handle default values for the byte data type.
SDC-8848	When a cluster mode pipeline is configured to skip lines before reading delimited data, data parsing errors occur.
SDC-8808	If a pipeline encounters parsing errors when converting the SDC Record data format to Avro due to incompatible data types or invalid values, the pipeline fails.
SDC-8580	Azure Event Hub libraries need to be upgraded.
SDC-8793	For the Vault credential store, if you update the secret ID value in the file configured for the `credentialStore.vault.config.secret.id` property, Data Collector does not use the updated value for currently running pipelines.
SDC-8195	The SFTP/FTP Client origin fails if the origin starts reading a file with the same name but a later timestamp, such as the file in the last stored offset.
SDC-7903	Do not use cluster streaming pipelines with MapR and Spark 2.x.
SDC-5531	When the HTTP Client processor receives an HTTP 500 error response, the pipeline fails instead of sending the record to error handling.

Known Issues

Please note the following known issues with this release.

For a full list of known issues, check out our JIRA.

JIRA	Description
SDC-9029	When pagination is enabled for the HTTP Client origin and the server responds with a relative URL, the origin does not correctly resolve the relative URL.
SDC-8959	The HDFS Standalone origin cannot successfully process the whole file data format.
SDC-8893	The Field Type Converter processor performs slower than it did in version 2.5.1.1.
SDC-8876	When you configure a metric rule with a meter metric type and a 1 minute rate for the metric element, the rule does not wait until 1 minute passes before evaluating the condition and sending the alert. Instead, the alert is sent when the pipeline starts and stops.
SDC-8855	The MySQL Binary Log origin does not start reading from the offset specified in the Initial Offset property after a pipeline restart.
SDC-8731	Pipelines with the Google Pub/Sub Subscriber origin might hang and display the following error in the Data Collector log: `io.grpc.StatusRuntimeException: UNAVAILABLE: The service was unable to fulfill your request. Please try again. [code=8a75]`
SDC-8697	Starting multiple pipelines concurrently that run a Jython import call can lead to retry errors and cause some of the pipelines to fail.
SDC-8680	The Azure Data Lake Store destination does not properly roll files based on the “roll” record header attribute.
SDC-8660	The MapR Streams destination cannot write to topics that are auto-created in MapR Streams when the destination is enabled for runtime resolution. Workaround: To write to an auto-created topic, disable runtime resolution.

SDC-8598	Upon starting, Data Collector writes the following messages to the log file about the Cloudera CDH 5.14 stage library:
	
	`The following stages have invalid classpath: cdh_5_14_lib`
	`Detected colliding dependency versions: <additional information>`
	
	These messages are written in error and can be safely ignored.

SDC-8514	The Data Parser processor sends a record to the next stage for processing even when the record encounters an error. Workaround: Use a Stream Selector processor after the Data Parser. Define a condition for the Stream Selector that checks if the fields in the record were correctly parsed. If not parsed correctly, send the record to a stream that handles the error.

SDC-8474	The Data Parser processor loses the original record when the record encounters an error.
SDC-8320	Data Collector inaccurately calculates the Record Throughput statistics for cluster mode pipelines when some Data Collector workers have completed while others are still running.
SDC-8078	The HTTP Server origin does not release the ports that it uses after the pipeline stops. Releasing the ports requires restarting Data Collector.
SDC-7761	The Java keystore credential store implementation fails to work for a Data Collector installed through Cloudera Manager. The jks-cs command creates the Java keystore file in the Data Collector configuration directory defined for the parcel. However, for Data Collector to access the Java keystore file, the file must be outside of the parcel directory. The CyberArk and Vault credential store implementations do work with a Data Collector installed through Cloudera Manager.
SDC-7645	The Data Collector Docker image does not support processing data using another locale. Workaround: Install Data Collector from the tarball or RPM package.
SDC-6554	When converting Avro to Parquet on Impala, Decimal fields seem to be unreadable. Data Collector writes the Decimal data as variable-length byte arrays. And due to Impala issue IMPALA-2494, Impala cannot read the data.
SDC-5357	The Jython Evaluator and stages that can access the file system - such as the Directory and File Tail origins - can access all pipeline JSON files stored in the $SDC_DATA directory. This allows users to access pipelines that they might not have permission to access within Data Collector. Workaround: To secure your pipelines, complete the following tasks: 1. Remove the Jython stage library and use the Groovy Evaluator or JavaScript Evaluator processor instead of the Jython Evaluator. 2. Update the Data Collector security policy file, `$SDC_CONF/sdc-security.policy`, so that Data Collector stages do not have `AllPermission` access to the file system. Update the security policy for the following code bases: streamsets-libs-extras, streamsets-libs, and streamsets-datacollector-dev-lib. Use the policy file syntax to set the security policies.
SDC-5141	Due to a limitation in the Javascript engine, the Javascript Evaluator issues a null pointer exception when unable to compile a script.
SDC-4212	If you configure a UDP Source or UDP to Kafka origin to enable multithreading after you have already run the pipeline with the option disabled, the following validation error displays: `Multithreaded UDP server is not available on your platform.` Workaround: Restart Data Collector.

SDC-3944	The Hive Streaming destination using the MapR library cannot connect to a MapR cluster that uses Kerberos or username/password login authentication.
SDC-2374	A cluster mode pipeline can hang with a CONNECT_ERROR status. This can be a temporary connection problem that resolves, returning the pipeline to the RUNNING status. If the problem is not temporary, you might need to manually edit the pipeline state file to set the pipeline to STOPPED. Edit the file only after you confirm that the pipeline is no longer running on the cluster or that the cluster has been decommissioned. To manually change the pipeline state, edit the following file: `$SDC_DATA/runInfo/<cluster pipeline name>/<revision>/pipelineState.json` In the file, change CONNECT_ERROR to STOPPED and save the file.

Contact Information

For more information about StreamSets, visit our website: https://streamsets.com/.

Check out our Documentation page for doc highlights, what's new, and tutorials: streamsets.com/docs

Or you can go straight to our latest documentation here:
https://streamsets.com/documentation/datacollector/latest/help

To report an issue, to get help from our Google group, Slack channel, or Ask site, or to find out about our next meetup, check out our Community page: https://streamsets.com/community/.

For general inquiries, email us at info@streamsets.com.