StreamSets

TECHNICAL GUIDE

MySQL Change Data Capture to Snowflake Guide

Contents

Introduction . . . 3

Prerequisites . . . 3

Guided Walkthrough . . . 4

MySQL CDC to Snowflake Pipeline Overview . . . 4

MySQL CDC Client Origin . . . 4

Field Masker Processor . . . 5

Snowflake Destination . . . 5

MySQL CDC to Snowflake Sample Pipeline . . . 6

Import Pipeline . . . 6

Select Authoring Data Collector . . . 7

Pipeline Validation, Preview and Test Run . . . 8

Create and Run Job . . . 9

Monitor Job . . . 11

Introduction

Change Data Capture is a design pattern to determine, track, capture, and deliver changes made to enterprise data sources--typically relational databases like MySQL, Oracle, and PostgreSQL.This is critical because when these changes occur at the source, actions must be taken using the changed data in data warehouses for downstream analysis.

In this guide, you will learn how to process Change Data Capture (CDC) data from MySQL to Snowflake in StreamSets DataOps Platform.

Prerequisites

- Access to StreamSets DataOps Platform account
  - Setup Environment
  - Setup Deployment with engine type Data Collector
    - Once a deployment has been successfully activated, the Data Collector engine should be up and running before you can create pipelines and run jobs.
- Access to Snowflake account
- Access to MySQL database
  - Check versions of MySQL supported for CDC

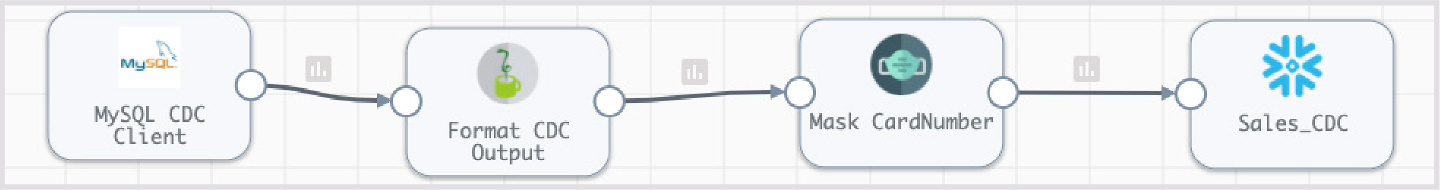
Complete MySQL CDC prerequisites

Guided Walkthrough

Once all of the above prerequisites have been satisfied and you have a Data Collector engine up and running, follow along and learn how to process Change Data Capture (CDC) data from MySQL to Snowflake.

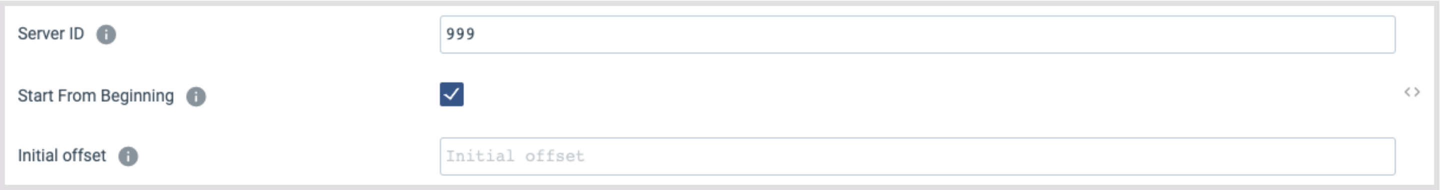
MySQL CDC to Snowflake Pipeline Overview

A data pipeline describes the flow of data from origin to destination systems and defines how to process the data along the way. Pipelines can access multiple types of external systems, including cloud data lakes, cloud data warehouses, and storage systems installed on-premises such as relational databases.



MySQL CDC Client Origin

MySQL CDC Bin Log Client origin will enable you to capture Create, Update, and Delete (CRUD) operations across various tables in your MySQL data warehouse so that your Snowflake Data Cloud can be kept in sync.



Server ID	999
Start From Beginning	☑
Initial offset	Initial offset

Key configuration on **MySQL Binary Log** tab for this setup:

- **Server ID:** Unless otherwise required you can keep the default value
- **Start From Beginning:** Unless you want to start from a specific offset checking this option will capture everything from the start
- **Initial Offset:** if you have GTID enabled you can use that value to start from which ever transaction you want and then going forward.

For other configuration details such as JDBC connection string, limiting CDC operation to specific tables instead of all tables, GTID (Global Transaction ID), etc., refer to the detailed configuration section.

In StreamSets DataOps Platform, it is really easy to optionally apply any number of transformations to data while it's in motion flowing through the pipeline. Here is an example using Field Masker processor.

Field Masker Processor

This processor will enable us to "mask" PII in configured fields. In this case, it is configured to mask customer Credit Card Numbers before sending it over to Snowflake.

Key configuration on **Mask** tab for this setup:

- Set **Fields to Mask** to */CardNumber*
- Set **Mask Type** to *Variable Length*

Snowflake Destination

Snowflake destination uses the MERGE command to write CDC data that's being captured from MySQL. Note that this provides real-time access to data as it is written into Snowflake tables.

Key configuration on **Snowflake Connection Info** tab for this setup:

- Set **Snowflake Region, Account, User, and Password**
    - ***Note:*** You can also take advantage of Snowflake Connection so these attributes can be used across multiple pipelines, shared with team members and any changes to credentials can be made in a centralized location.

Key configuration on **Snowflake** tab for this setup:

- Set **Warehouse, Database, Schema, and Table**
    - ***Note:*** Setting **Table** to *${record:attribute('table')}* will dynamically get the table name from the record header attribute generated by the MySQL CDC Client origin
- Enable **Table Auto Create** -- this will automatically create the table(s) if they don't already exist in Snowflake

Key configuration on **Data** tab for this setup:

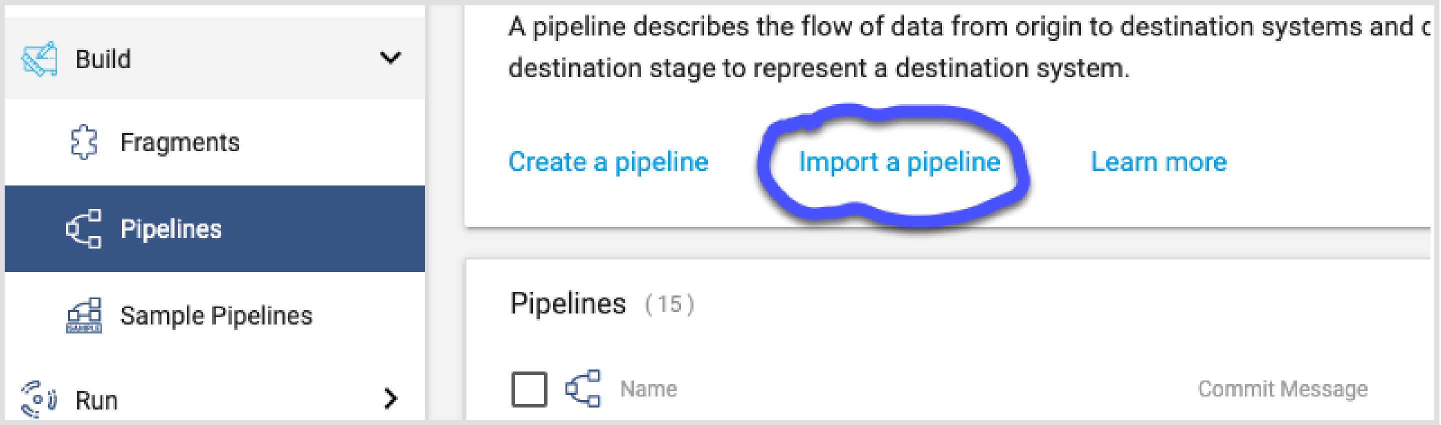
- Enable **Processing CDC Data (Use MERGE)**
- Set **Table Key Columns** for all the tables you'd like to capture and sync changes. For example, in this case we'd like to sync records from the following tables:
    - Table: **SALES**; Key Columns: *InvoiceId,SSN*

For other configuration details such as **Staging, Snowflake File Format**, defaults for missing fields, etc. refer to the configuration section.

MySQL CDC to Snowflake Sample Pipeline

After you download the sample pipeline from GitHub, use the **Import a pipeline** feature to create an instance of the pipeline in your StreamSets DataOps Platform account.

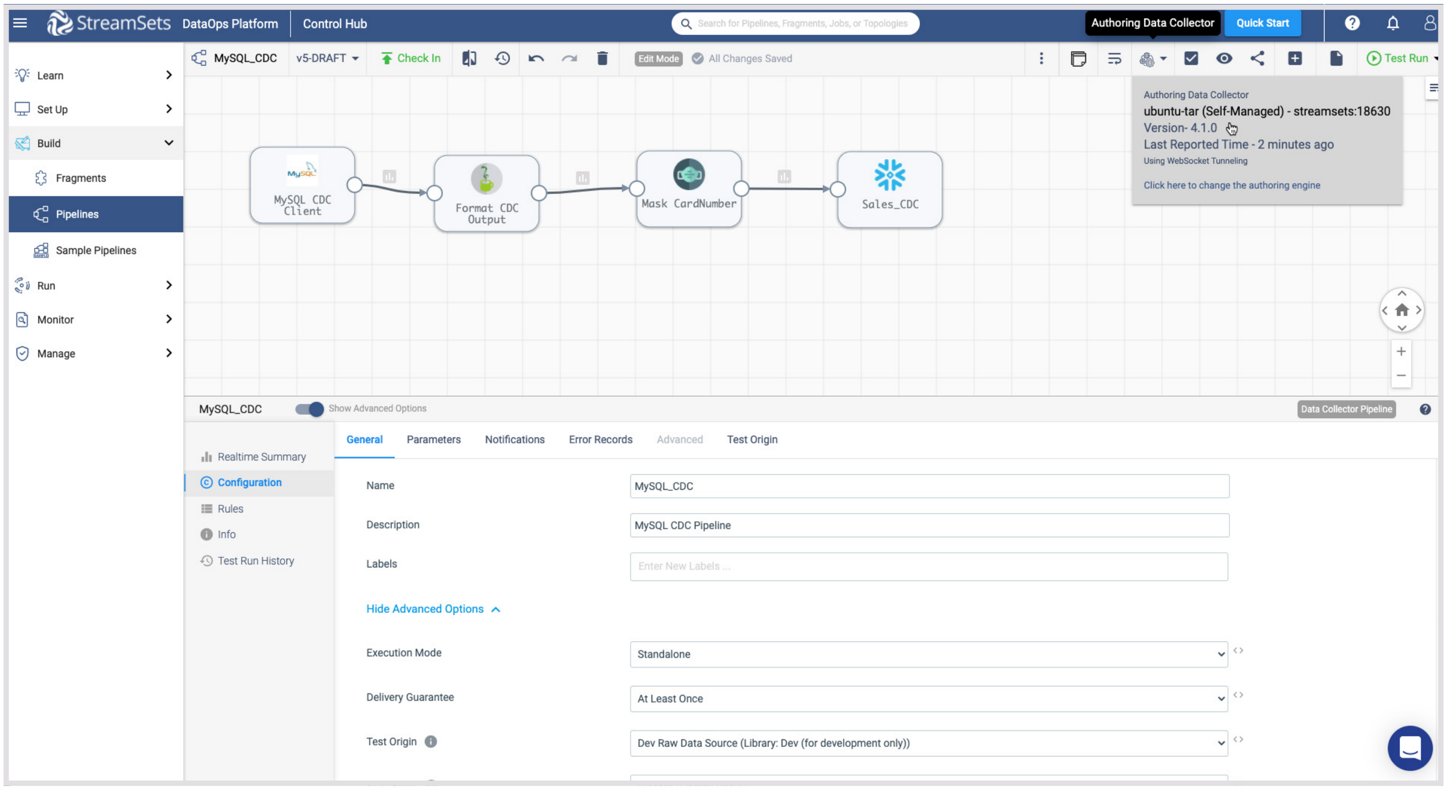
Import Pipeline



A pipeline describes the flow of data from origin to destination systems and c destination stage to represent a destination system.

Select Authoring Data Collector

Once the pipeline has been imported, open it in the pipeline canvas and select **Authoring Data Collector** -- this is the Data Collector engine that would have been deployed once your deployment was successfully activated.



Authoring Data Collector
ubuntu-tar (Self-Managed) - streamsets:18630
Version- 4.1.0
Last Reported Time - 2 minutes ago
Using WebSocket Tunneling

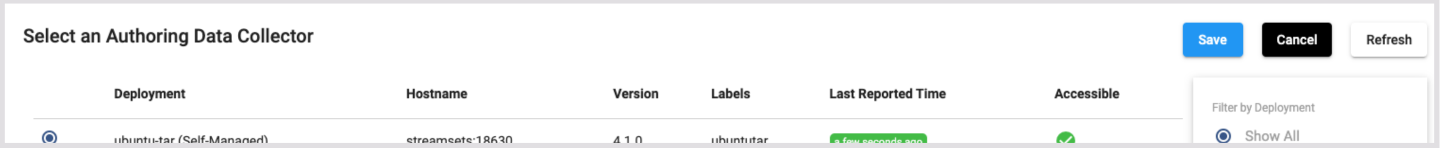
Click here to change the authoring engine

MySQL_CDC

Name	MySQL_CDC
Description	MySQL CDC Pipeline
Labels	Enter New Labels ...

Hide Advanced Options

Execution Mode	Standalone
Delivery Guarantee	At Least Once
Test Origin	Dev Raw Data Source (Library: Dev (for development only))

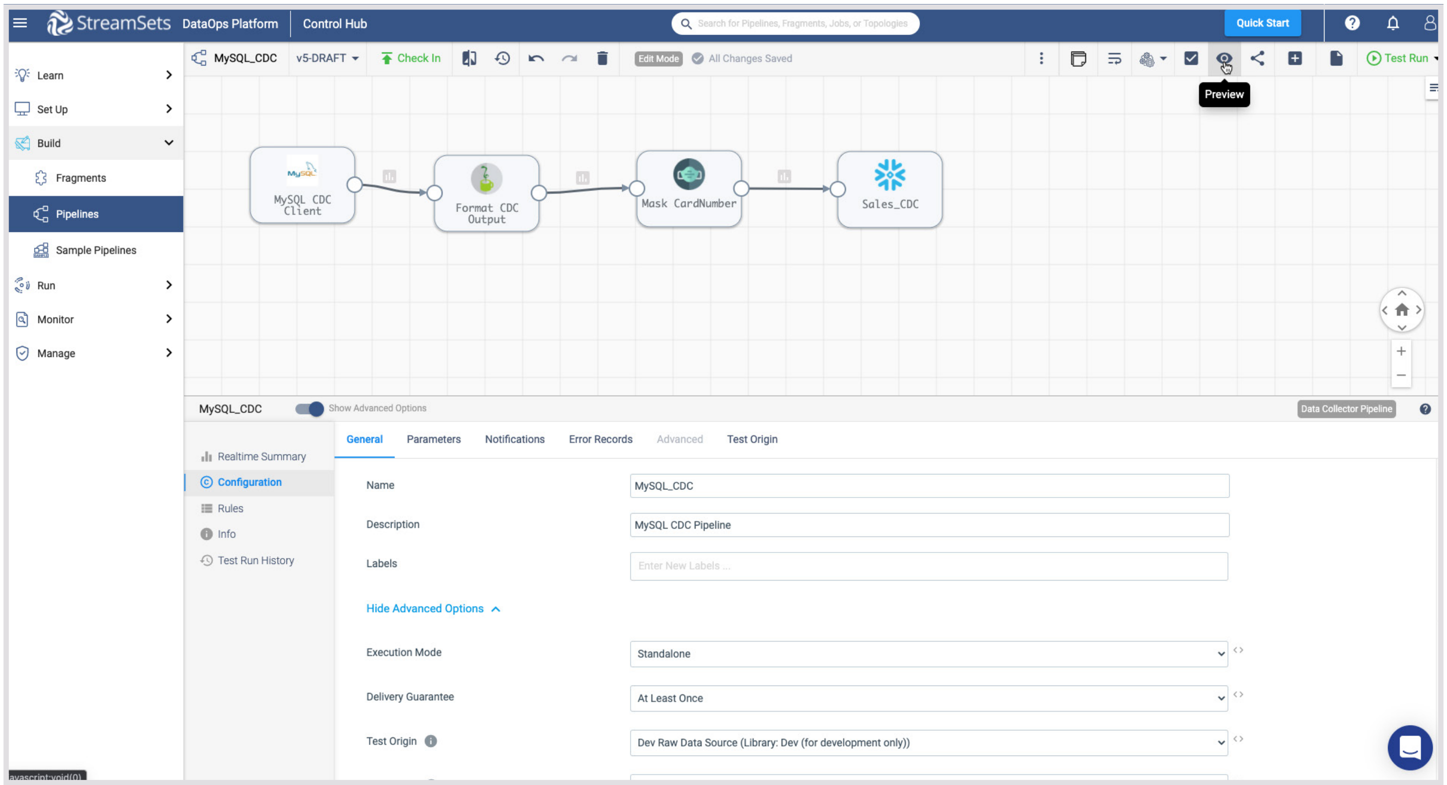


Select an Authoring Data Collector

	Deployment	Hostname	Version	Labels	Last Reported Time	Accessible
●	ubuntu-tar (Self-Managed)	streamsets:18630	4.1.0	ubuntutar	a few seconds ago	✓

Pipeline Validation, Preview and Test Run

Once you've updated the pipeline parameters, you can validate it to make sure the credentials are correct, preview the data to make sure the transformations are accurate and also test run the pipeline to ensure the data is being ingested into Snowflake correctly.



Name: MySQL_CDC

Description: MySQL CDC Pipeline

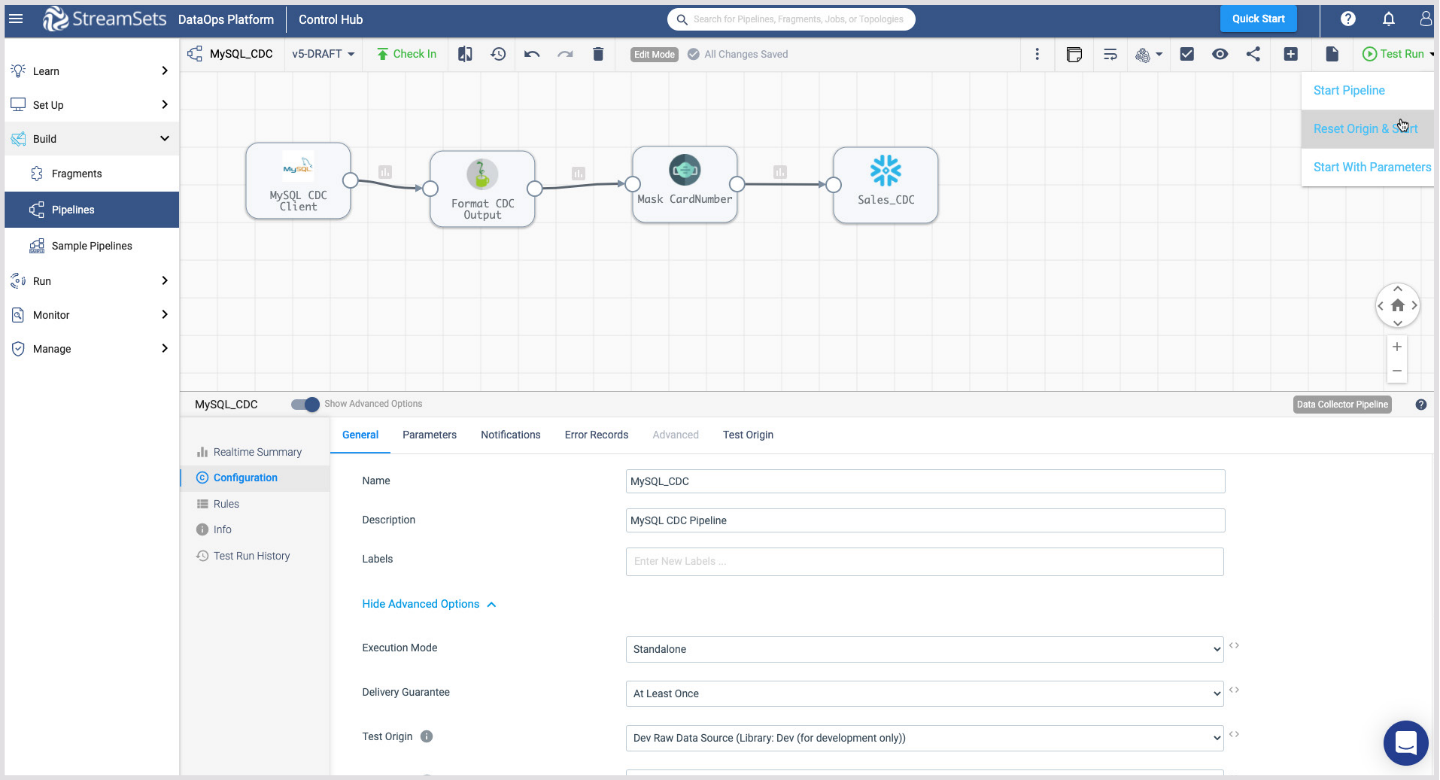
Labels: Enter New Labels ...

Execution Mode: Standalone

Delivery Guarantee: At Least Once

Test Origin: Dev Raw Data Source (Library: Dev (for development only))

Make sure to **Reset Origin & Run**



Name: MySQL_CDC

Description: MySQL CDC Pipeline

Labels: Enter New Labels ...

Execution Mode: Standalone

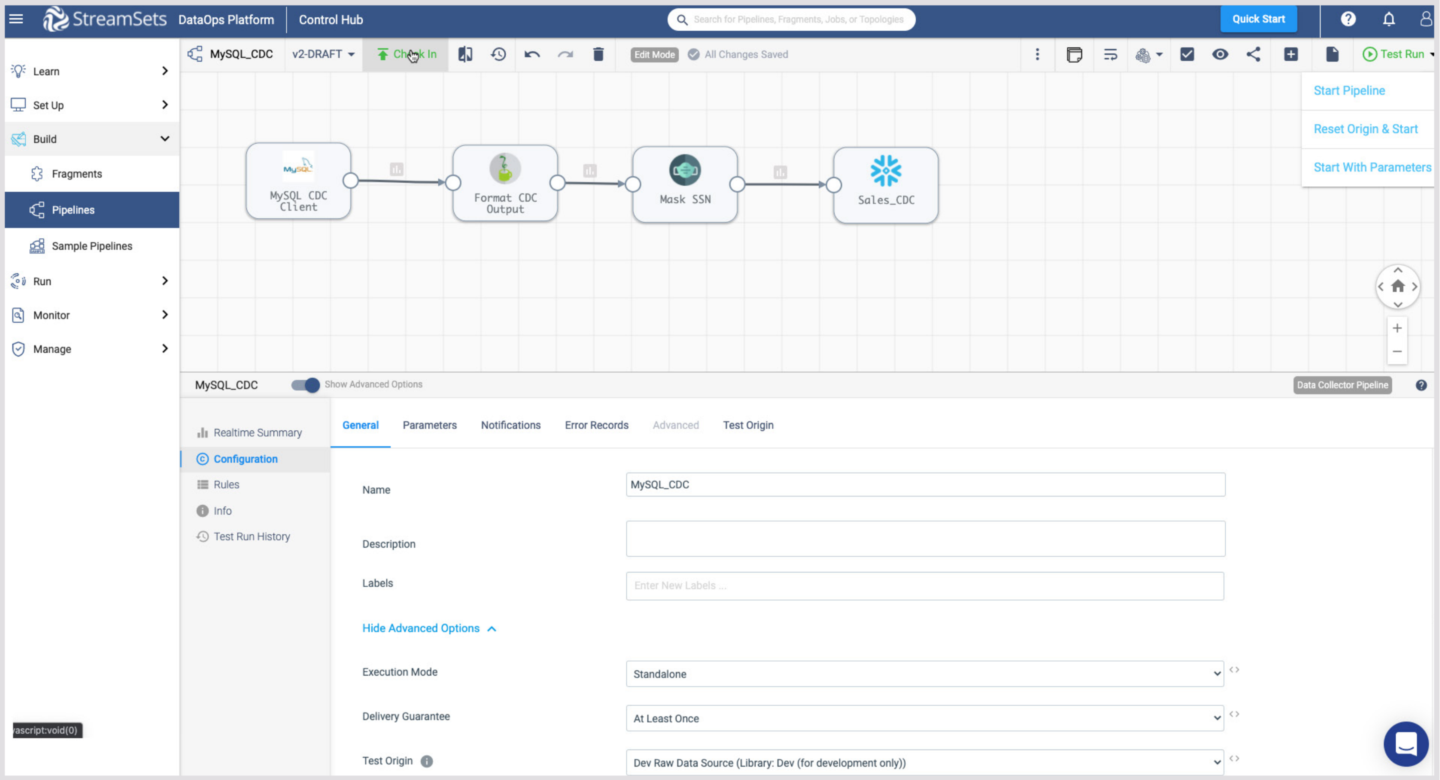
Delivery Guarantee: At Least Once

Test Origin: Dev Raw Data Source (Library: Dev (for development only))

Create and Run Job

Once you've successfully executed a pipeline test run, you can create a job to run the pipeline after you've checked-in the pipeline.

Jobs enable you to execute, manage and orchestrate data pipelines that run across multiple engines. You can increase the number of pipeline instances that run for a job, or you can enable a job for pipeline failover to minimize downtime due to unexpected failures.



Name: MySQL_CDC

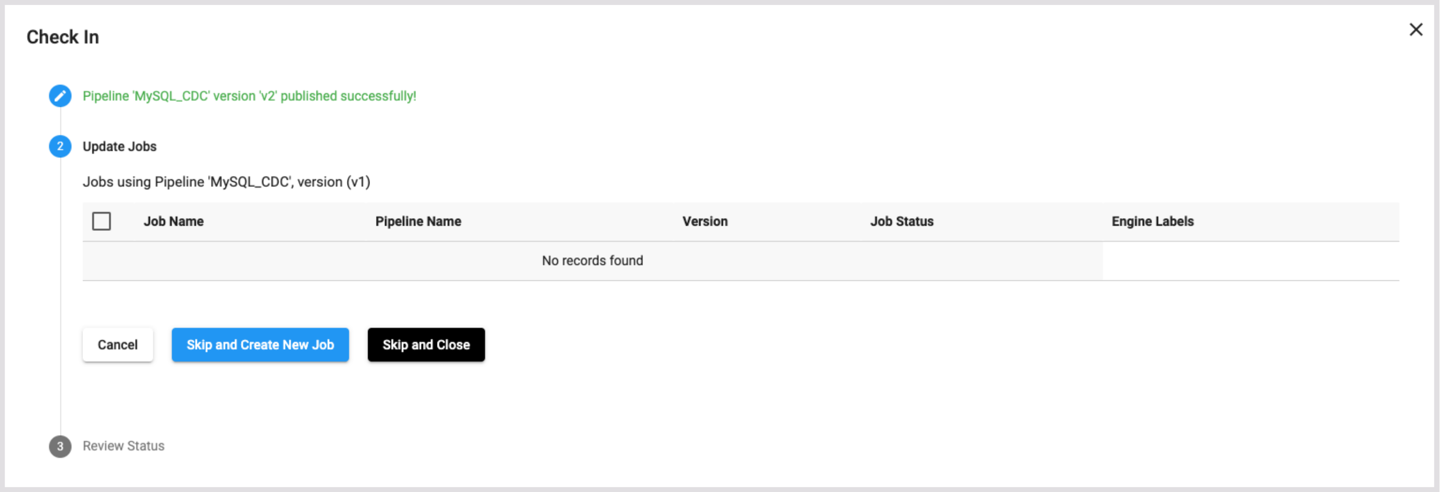
Description:

Labels: Enter New Labels ...

Execution Mode: Standalone

Delivery Guarantee: At Least Once

Test Origin: Dev Raw Data Source (Library: Dev (for development only))



Check In

Pipeline 'MySQL_CDC' version 'v2' published successfully!

2 Update Jobs

Jobs using Pipeline 'MySQL_CDC', version (v1)

□	Job Name	Pipeline Name	Version	Job Status	Engine Labels
		No records found			

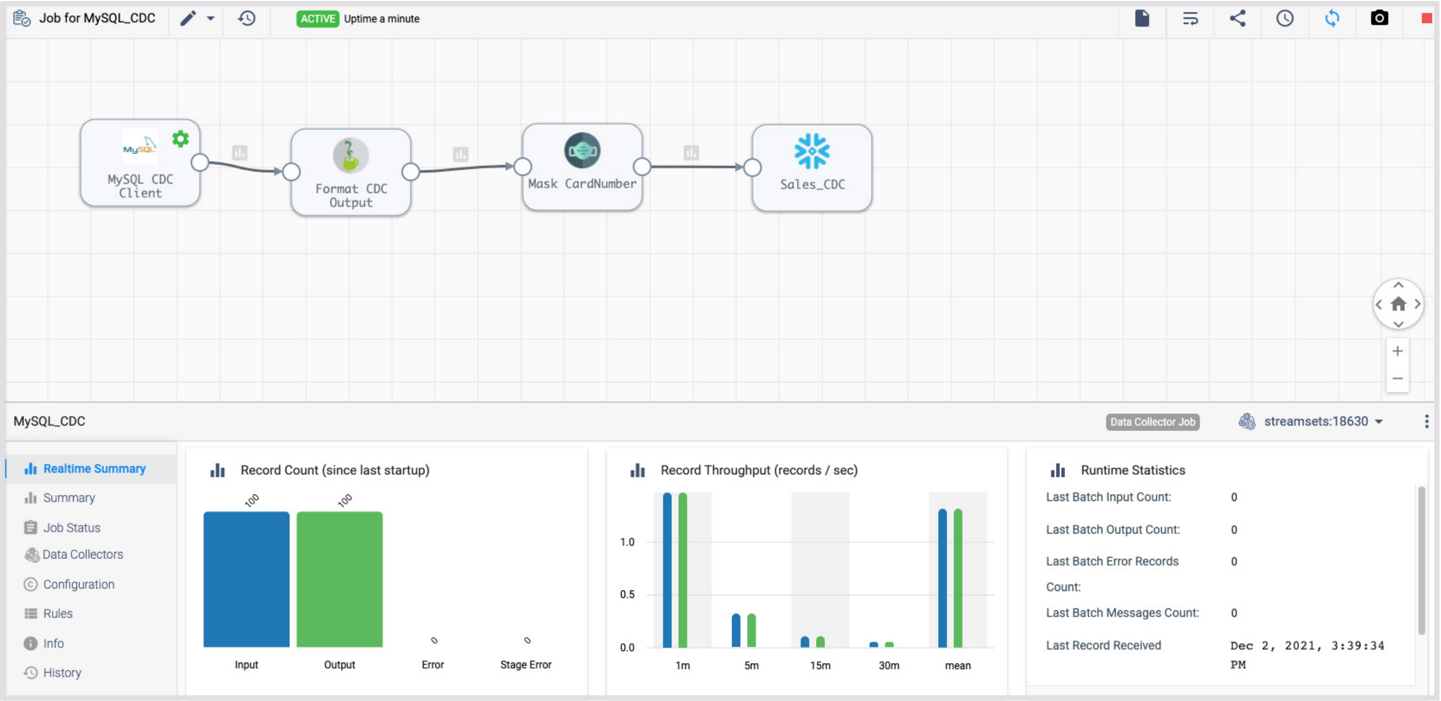
3 Review Status

For more information on jobs, refer to the documentation.

Monitor Job

When you start a job, Control Hub sends the pipeline to the engines. The engine runs the pipeline, sending status updates and metrics back to Control Hub.

As the job runs, click the **Realtime Summary** tab in the monitor panel to view the real-time statistics for the job.



Last Batch Input Count: 0
Last Batch Output Count: 0
Last Batch Error Records Count: 0
Last Batch Messages Count: 0
Last Record Received Dec 2, 2021, 3:39:34 PM

For more information on monitoring jobs, refer to the documentation.

About StreamSets

StreamSets' core mission is to make data engineering teams wildly successful. The StreamSets DataOps Platform empowers engineers to build and run the smart data pipelines needed to power DataOps across hybrid and multi-cloud architectures. That's why the largest companies in the world trust StreamSets to power millions of data pipelines for modern analytics, AI/ML and smart applications.

Try Now

Get up and running with StreamSets in minutes - free.

Start Now