



TECHNICAL GUIDE

# Oracle to Snowflake Guide

# Introduction

In this guide, you will learn how to migrate data from Oracle to Snowflake in StreamSets DataOps Platform.

## Prerequisites

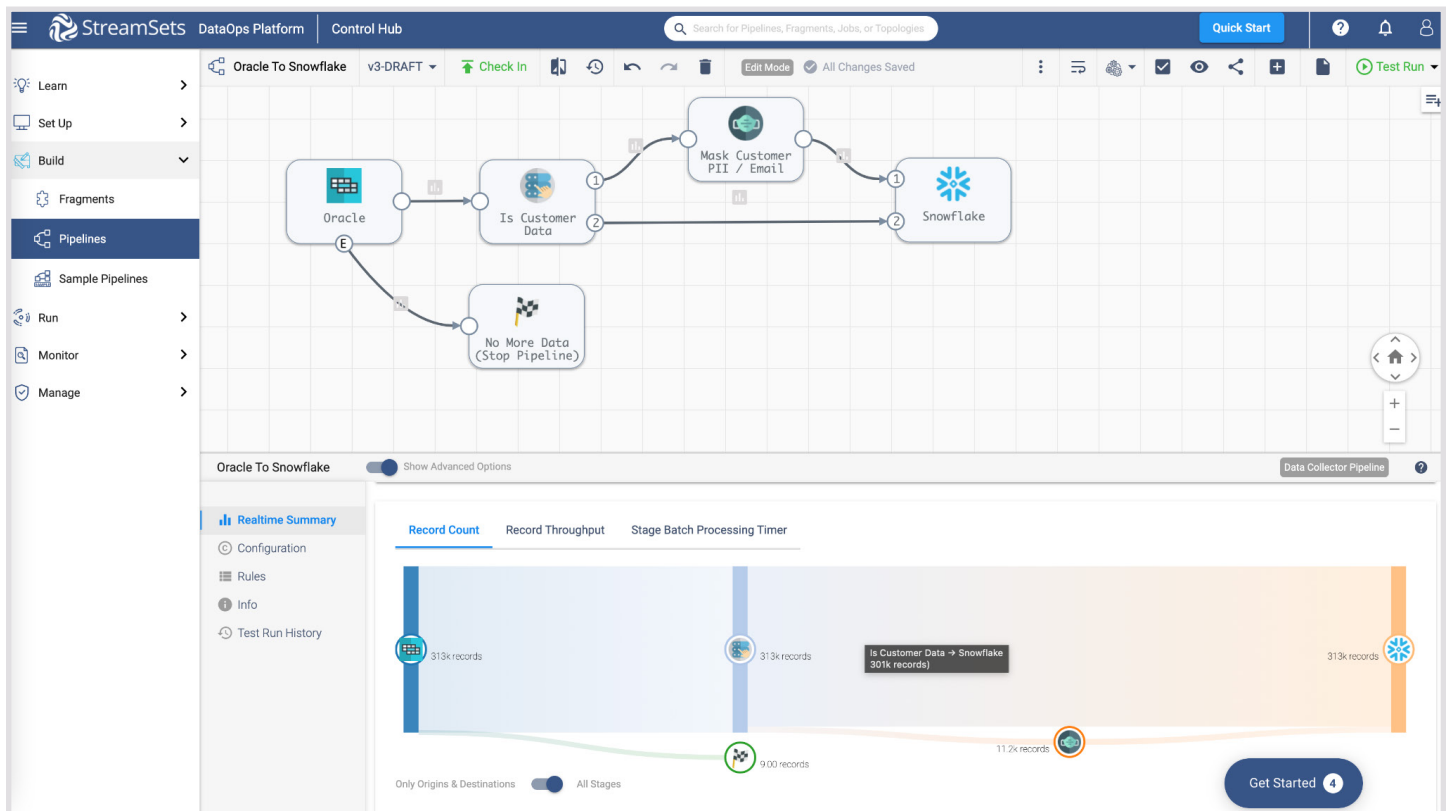
- Access to [StreamSets DataOps Platform](#) account
  - Setup [Environment](#)
  - Setup [Deployment](#) with engine type [Data Collector](#)
    - Once a deployment has been successfully activated, the Data Collector engine should be up and running before you can create pipelines and run jobs.
- Access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

# 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 migrate data from Oracle to Snowflake.

## Oracle 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.



## Origin

[JDBC Multitable Consumer](#) origin will enable you to migrate data across various tables in your Oracle data warehouse into Snowflake Data Cloud.

Key configuration on the **General** tab for this setup:

- Enable **Produce Events** -- this will instruct the pipeline to automatically stop the pipeline once all of the records have been processed.
  - This is made possible using the [Pipeline Finisher](#) executor. In this executor, add a precondition under **Preconditions** and set the condition to `${record:eventType() == 'no-more-data'}` and set **On Record Error** to **Discard** so that events other than 'no-more-data' are ignored.

Key configuration on the **JDBC** tab for this setup:

- Set **Max Batch Size (Records)** -- depending on how many records there are across all tables to be migrated, set this to a value that makes sense. For example, in my case, I have about ~300K total records to migrate from Oracle and setting this value to 25,000 takes about a minute to process all the records on a m4.16xlarge AWS instance.

Key configuration on the **Tables** tab for this setup:

- Set **Schema** to the one that would like to migrate tables off your Oracle data warehouse
- Set **Table Name** Pattern to **"%"** -- this wildcard will migrate all tables in your Oracle data warehouse

For other configuration details such as **JDBC Connection String**, limiting migration of specific tables using **Table Name Pattern** or **Table Exclusion Pattern** instead of all tables, etc., refer to the detailed [configuration section](#).

In the 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 are a couple of examples using [Stream Selector](#) and [Field Masker](#) processors.

## Stream Selector Processor

This processor will conditionally route records based on user-defined conditions. For instance, in this case, we'd like to protect customer email addresses from being ingested (in plaintext) in Snowflake.

Key configuration on the Conditions tab for this setup:

- Set **Condition 1** to expression `${str:toLower(record:attribute('jdbc.tables')) == str:toLower('customers')}`  
-- this will route records being read from 'customers' table through **Field Masker**; all other records will flow directly into Snowflake.

## Field Masker Processor

This processor will enable us to “mask” PII in configured fields. In this case, it is configured to mask customer email addresses before sending it over to Snowflake.

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

- Set **Fields to Mask** to `/CUSTOMER_EMAIL`
- Set **Mask Type** to *Custom*
- Set **Custom Mask** to `XXXXXXXX`

## Snowflake Destination

[Snowflake](#) destination in this case will use the COPY command, the default load method, to perform a bulk synchronous load, treating all records as INSERTS.

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

- Set **Warehouse, Database, Schema, and Table**

**Note:** Setting **Table** to `${record:attribute('jdbc.tables')}` will dynamically get the table name from the record header attribute generated by the [JDBC Multitable Consumer](#) origin

- Enable **Table Auto Create** -- this will automatically create the tables if they don't already exist in Snowflake

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

# Oracle 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

The screenshot shows the StreamSets DataOps Platform Control Hub. On the left is a navigation menu with options: Learn, Set Up, Build, Fragments, Pipelines, Sample Pipelines, Run, Monitor, and Manage. The 'Pipelines' section is selected. The main area displays 'What are Pipelines?' with a description and three links: 'Create a pipeline', 'Import a pipeline' (circled in blue), and 'Learn more'. Below this is a table of pipelines.

| Name  | Commit Message   | Version  | Last Modified On | Last Modified By    |
|---|------------------|----------|------------------|---------------------|
| DwD: Oracle CDC To Snowflake                  | v1               | v3       | 2 hours ago      | dash@streamsets.com |
| DwD: Salesforce To Cloud Storage              | my second commit | v3-DRAFT | 4 hours ago      | dash@streamsets.com |
| DwD: MySQL CDC To S3                          | v1               | v4-DRAFT | 4 days ago       | dash@streamsets.com |
| DwD: ML Train Model (MLflow   DB   Delta L... | v1               | v4-DRAFT | 4 days ago       | dash@streamsets.com |
| DwD: GCS To BigQuery (Existing Dataproc)      | v1               | v4-DRAFT | 10 days ago      | dash@streamsets.com |
| DwD: Streamino Data To S3 (Fragment)          | v1               | v1-DRAFT | 11 days ago      | dash@streamsets.com |

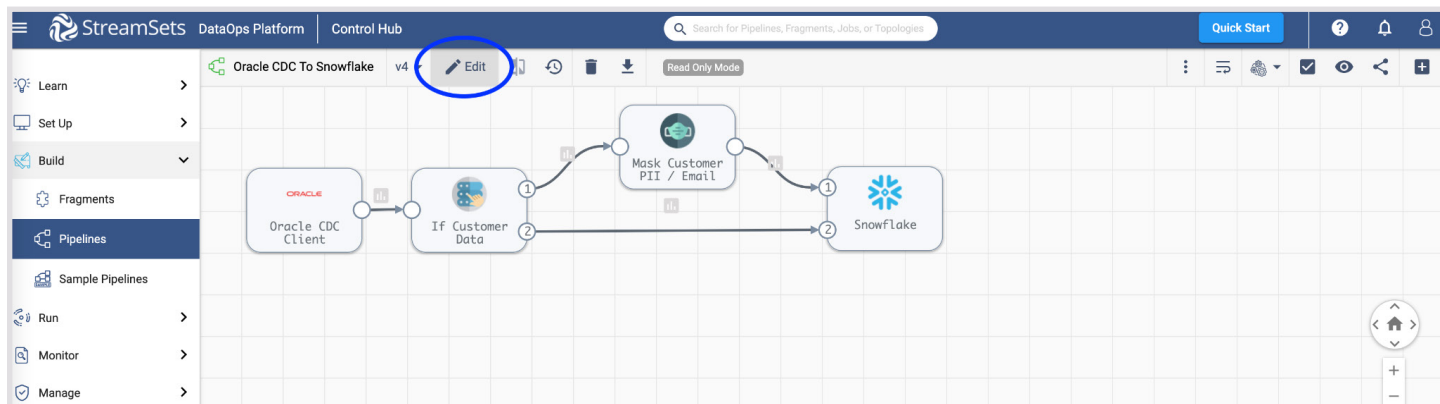
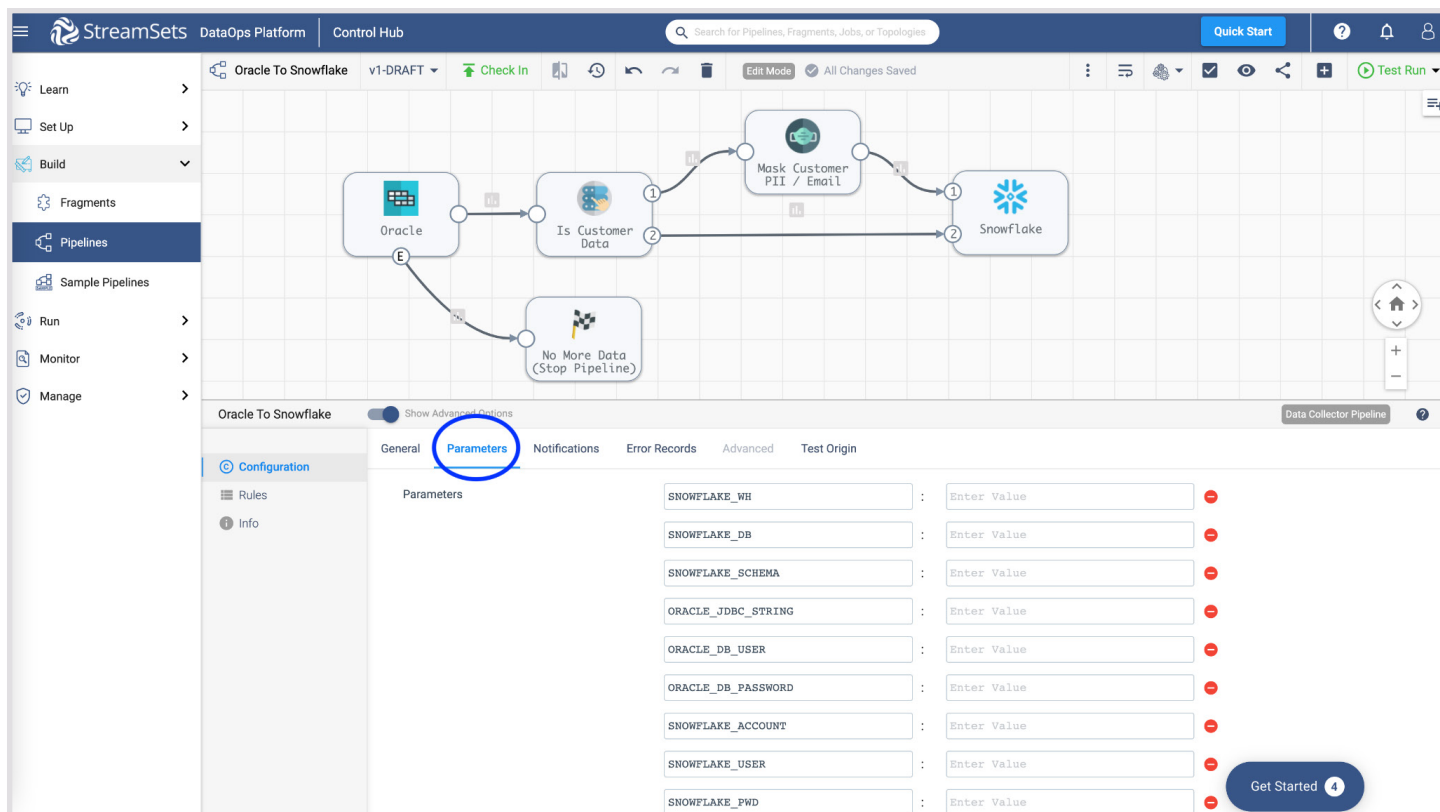
## Select Authoring Data Collector

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.

The screenshot shows the StreamSets DataOps Platform pipeline canvas. The pipeline is named 'Oracle To Snowflake' and is in 'v1-DRAFT' mode. The canvas displays a flow starting from an 'Oracle' connector, passing through an 'Is Customer Data' processor, and ending at a 'No More Data (Stop Pipeline)' connector. The 'Authoring Data Collector' engine is highlighted with a blue circle in the top right corner of the canvas. A tooltip for the 'Authoring Data Collector' is visible, showing details: 'SDC 4.0.2 (Amazon EC2) - ip-10-10-48-65.us-west-2.compute.internal:18630', 'Version: 4.0.2', 'Last Reported Time - a minute ago', and 'Using WebSocket Tunneling'. The tooltip also includes a link to 'Click here to change the authoring engine'.

## Edit Pipeline and Set Pipeline Parameters

After selecting the authoring Data Collector engine, click on the **Edit** button and update the following pipeline parameters.

| Parameter          | Value       | Status   |
|--------------------|-------------|----------|
| SNOWFLAKE_WH       | Enter Value | Required |
| SNOWFLAKE_DB       | Enter Value | Required |
| SNOWFLAKE_SCHEMA   | Enter Value | Required |
| ORACLE_JDBC_STRING | Enter Value | Required |
| ORACLE_DB_USER     | Enter Value | Required |
| ORACLE_DB_PASSWORD | Enter Value | Required |
| SNOWFLAKE_ACCOUNT  | Enter Value | Required |
| SNOWFLAKE_USER     | Enter Value | Required |
| SNOWFLAKE_PWD      | Enter Value | Required |

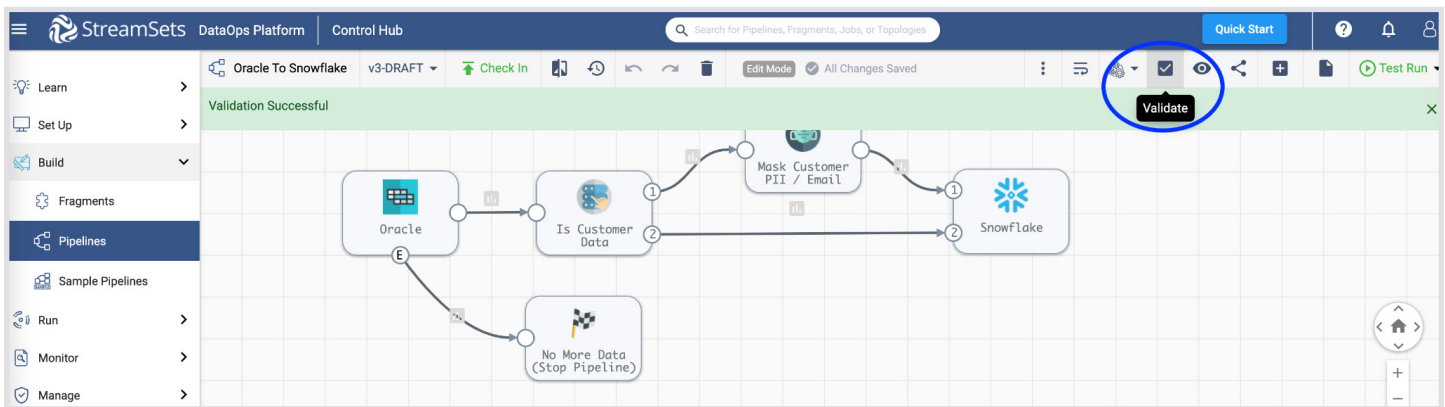
Get Started 4

Pipeline parameters to update.

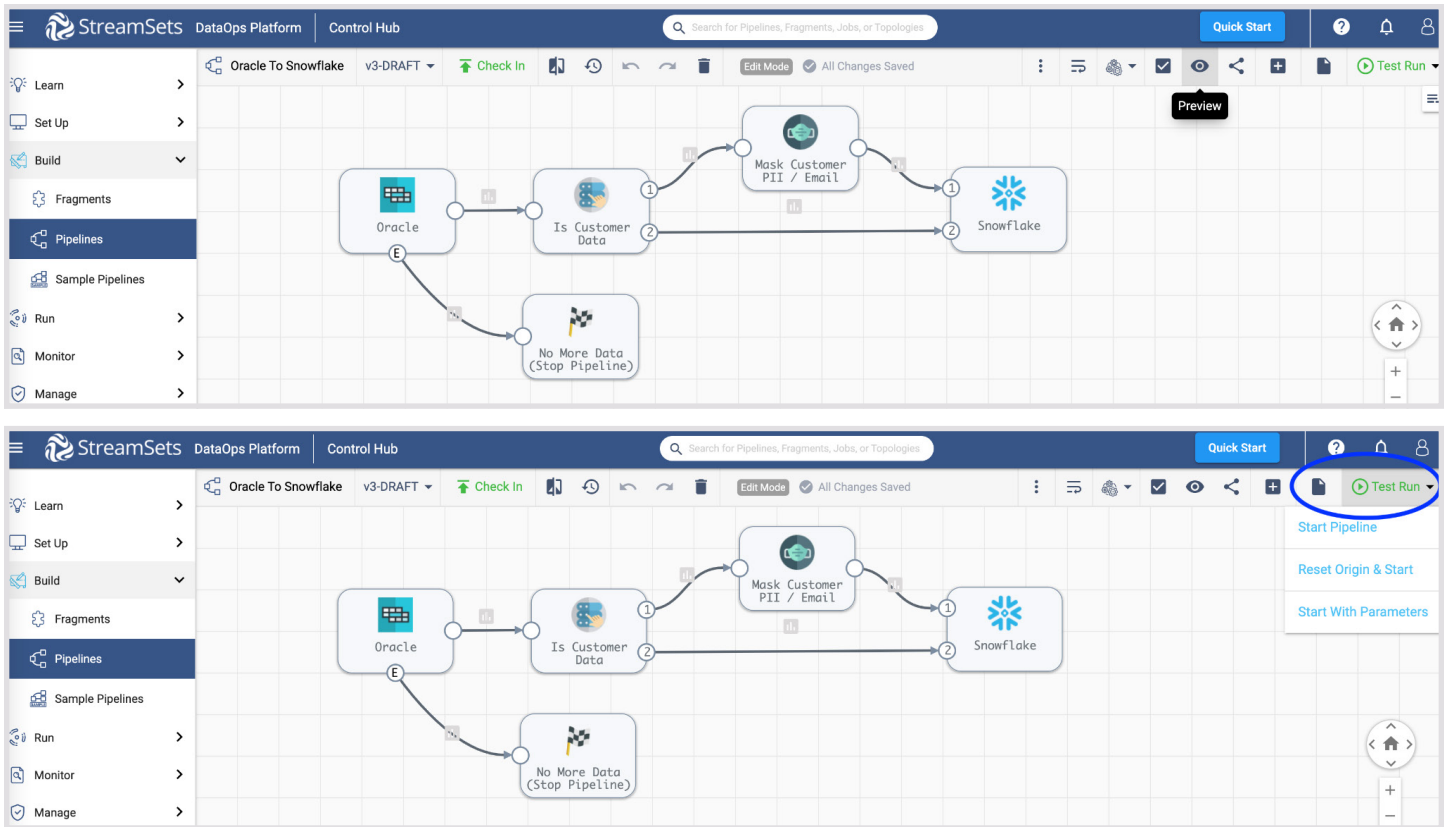
- `SNOWFLAKE_WH`
- `SNOWFLAKE_DB`
- `SNOWFLAKE_SCHEMA`
- `SNOWFLAKE_ACCOUNT`
- `SNOWFLAKE_USER`
- `SNOWFLAKE_PWD`
- `ORACLE_JDBC_URL`
- `ORACLE_JDBC_USERNAME`
- `ORACLE_JDBC_PASSWORD`

## 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.



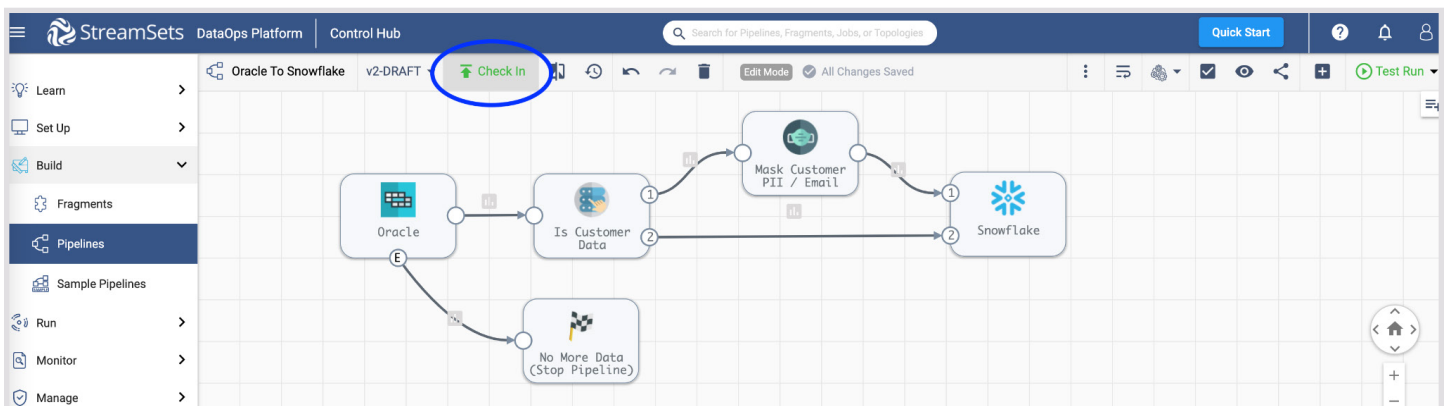


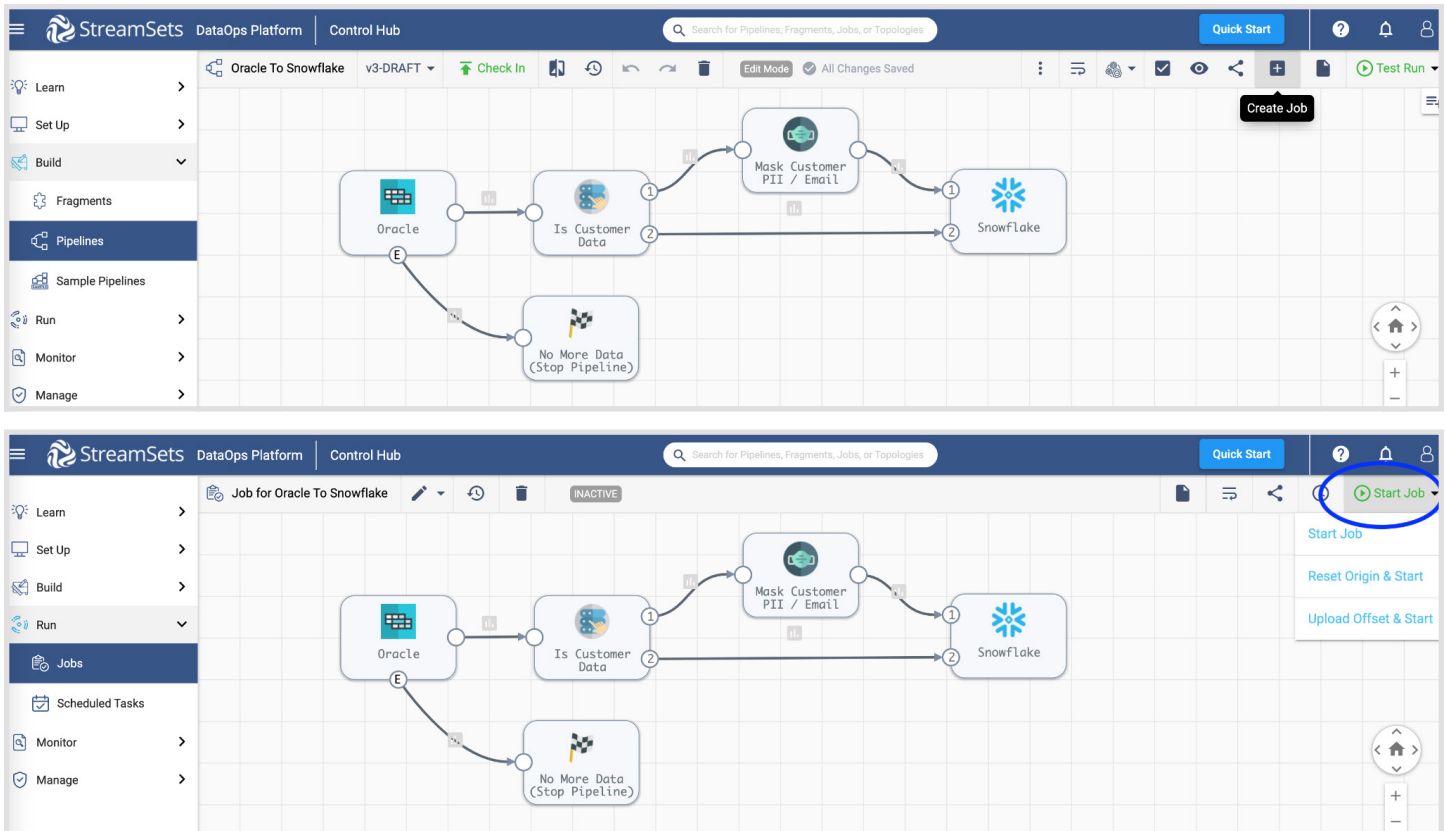


## Create and Run Job

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.





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 the Control Hub.

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



For more information on jobs, refer to the [documentation](#).

To learn more about StreamSets, please visit [www.streamsets.com](http://www.streamsets.com).

## 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](http://www.streamsets.com)