

# Instant Snowflake Demo

Here's an easy and fast StreamSets / Snowflake demo that uses both Data Collector and Transformer for Snowflake, and requires no infrastructure setup other than being able to run a Data Collector.

The demo retrieves incident records from [Snowflake's Incident API](#), auto-creates and populates a Snowflake table with the incident records, and then aggregates the incidents by region.

## Table of Contents

<b>Prerequisites</b>	<b>2</b>
<b>Import the Pipelines</b>	<b>2</b>
Get Snowflake Incidents (Data Collector pipeline)	2
Snowflake Incidents by Region (Transformer for Snowflake pipeline)	2
<b>Set Parameters in the Data Collector Pipeline</b>	<b>3</b>
<b>Set a Snowflake Connection in your Data Collector Pipeline</b>	<b>4</b>
<b>Preview the Data Collector pipeline</b>	<b>4</b>
Preview data from the origin	5
Preview the pivoted incidents	6
Preview the Get Region enrichment	7
<b>Run the pipeline</b>	<b>8</b>
<b>Set Parameters in the Transformer for Snowflake Pipeline</b>	<b>9</b>
<b>Preview the Transformer for Snowflake pipeline</b>	<b>10</b>
Preview the source records	11
Preview the aggregation	12
Preview the sort	13
<b>Run the pipeline</b>	<b>14</b>
<b>Resetting the Demo</b>	<b>15</b>

## Prerequisites

- A deployed Data Collector Engine
- A Snowflake Connection defined in Data Collector
- A Snowflake account (a trial account is fine)
- The ability to run Transformer for Snowflake pipelines

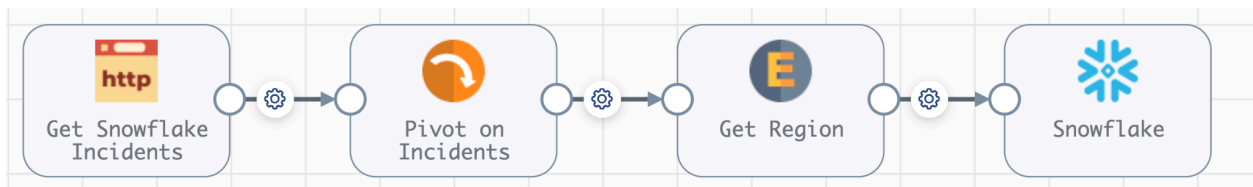
## Import the Pipelines

- Download the pipeline archive as a zip file from [here](#) (click the page's download link)
- Import the pipeline as a pipeline archive (docs on how to do that are [here](#)).

You should see the following two pipelines have been added to your pipeline list.

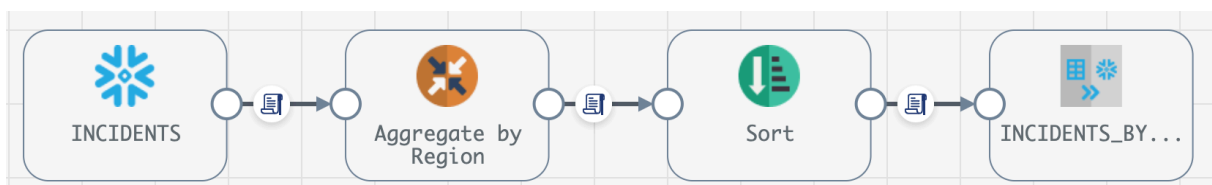
### Get Snowflake Incidents (Data Collector pipeline)

Here is the `Get Snowflake Incidents` Data Collector pipeline that gets the incidents from the REST API, does some in-flight transformation, and writes them to Snowflake:



### Snowflake Incidents by Region (Transformer for Snowflake pipeline)

Here is the `Snowflake Incidents by Region` Transformer for Snowflake pipeline that aggregates incidents by region:



# Set Parameters in the Data Collector Pipeline

Set these pipeline parameters to match your Snowflake environment. For the `SNOWFLAKE_TABLE`, specify the name of a table that does not yet exist as the pipeline will create it.

---

[General](#)   [Parameters](#)   [Notifications](#)   [Error Records](#)   [Advanced](#)   [Test Origin](#)

---

[Hide Advanced Options](#) ^

Parameters	<input type="text" value="SNOWFLAKE_WH"/>	:	<input type="text" value="Enter Value"/>
	<input type="text" value="SNOWFLAKE_DB"/>	:	<input type="text" value="Enter Value"/>
	<input type="text" value="SNOWFLAKE_SCHEMA"/>	:	<input type="text" value="Enter Value"/>
	<input type="text" value="SNOWFLAKE_TABLE"/>	:	<input type="text" value="Enter Value"/>

[+ ADD ANOTHER](#)   [≡ BULK EDIT MODE](#)

For example, in my environment:

Parameters	<input type="text" value="SNOWFLAKE_WH"/>	:	<input type="text" value="MARK_WH"/>
	<input type="text" value="SNOWFLAKE_DB"/>	:	<input type="text" value="MARK_DB"/>
	<input type="text" value="SNOWFLAKE_SCHEMA"/>	:	<input type="text" value="MARK_SCHEMA"/>
	<input type="text" value="SNOWFLAKE_TABLE"/>	:	<input type="text" value="SNOWFLAKE_INCIDENTS"/>

# Set a Snowflake Connection in your Data Collector Pipeline

Set a Snowflake Connection in you Data Collector Pipeline:

General	<b>Snowflake Connection Info</b>	Snowflake	Snowpipe	Staging	Staging Advan
Connection		Snowflake			

## Preview the Data Collector pipeline

Preview the pipeline to see how it works. Use these settings in the Preview Config (with `Write to Destinations` and `Executors` unset) to avoid creating the Snowflake table just yet:

### Preview Configuration

Preview Source:	Configured Source
Preview Batch Size:	10
Preview Timeout (in milliseconds):	120000
Run Preview Through Stage:	--Run All--
Time zone:	Browser
Write to Destinations and Executors:	<input type="checkbox"/>
Execute Pipeline Lifecycle Events:	<input checked="" type="checkbox"/>
Show Record/Field Header:	<input checked="" type="checkbox"/>
Show Field Type:	<input checked="" type="checkbox"/>
Save Preview Record Schema:	<input checked="" type="checkbox"/>

## Preview data from the origin

Here we can see the full payload returned by the API call, with a list of 50 incidents:

**OUTPUT**

```
▼ Record1 : {MAP}
  ► page : {MAP}
  ▼ incidents : {LIST[50]}
    ► 0 : {MAP}
    ► 1 : {MAP}
    ▼ 2 : {MAP}
      id : {STRING} "zvd971z06grg"
      name : {STRING} "Azure - East US 2 (Virginia): INC0100718"
      status : {STRING} "postmortem"
      created_at : {STRING} "2024-01-18T09:02:05.335-08:00"
      updated_at : {STRING} "2024-01-29T13:02:58.345-08:00"
      monitoring_at : {STRING} "2024-01-18T10:16:27.386-08:00"
      resolved_at : {STRING} "2024-01-18T15:07:02.301-08:00"
      impact : {STRING} "minor"
      shortlink : {STRING} "https://stspg.io/j3m7x2w08k3p"
      started_at : {STRING} "2024-01-18T09:02:05.326-08:00"
      page_id : {STRING} "94s7z8vpy1n8"
      ► incident_updates : {LIST[5]}
      ► components : {LIST[2]}
      reminder_intervals : {STRING}
```

## Preview the pivoted incidents

We can see the incidents list has been split into individual incident records:

Preview Stage: Pivot on Incidents

**INPUT**

- Record1: {MAP}
  - page: {MAP}
  - incidents: {LIST[50]}
  - Record Header

**OUTPUT**

Record1-Output Record1: {MAP}

- id: {STRING} "rl9lbf3n6hsb"
- name: {STRING} "Azure - Central US (Iowa): INC0102516"
- status: {STRING} "resolved"
- created\_at: {STRING} "2024-02-09T08:17:01.654-08:00"
- updated\_at: {STRING} "2024-02-09T10:23:20.450-08:00"
- monitoring\_at: {STRING} "2024-02-09T09:25:57.986-08:00"
- resolved\_at: {STRING} "2024-02-09T10:23:20.434-08:00"
- impact: {STRING} "minor"
- shortlink: {STRING} "https://stspg.io/x70xg3vz5wfw"
- started\_at: {STRING} "2024-02-09T08:17:01.645-08:00"
- page\_id: {STRING} "94s7z8vpy1n8"
- incident\_updates: {LIST[4]}
- components: {LIST[1]}
- reminder\_intervals: {STRING}
- Record Header

Record1-Output Record2: {MAP}

- id: {STRING} "tspfv3y0j1v1"
- name: {STRING} "3 Deployments Affected: INC0101302"
- status: {STRING} "postmortem"

## Preview the Get Region enrichment

We can see how the region field was extracted from the start of the incident name. We'll use the region field for the Transformer for Snowflake aggregation:

Preview Stage: Get Region

**INPUT**

Record1: {MAP}

- id: {STRING} "r19lbf3n6hsb"
- name: {STRING} "Azure - Central US (Iowa): INC0102516"
- status: {STRING} "resolved"
- created\_at: {STRING} "2024-02-09T08:17:01.654-08:00"
- updated\_at: {STRING} "2024-02-09T10:23:20.450-08:00"
- monitoring\_at: {STRING} "2024-02-09T09:25:57.986-08:00"
- resolved\_at: {STRING} "2024-02-09T10:23:20.434-08:00"
- impact: {STRING} "minor"
- shortlink: {STRING} "https://stspg.io/x70xg3vz5wfw"
- started\_at: {STRING} "2024-02-09T08:17:01.645-08:00"
- page\_id: {STRING} "94s7z8vpy1n8"
- incident\_updates: {LIST[4]}
- components: {LIST[1]}
- reminder\_intervals: {STRING}

Record Header

**OUTPUT**

Record1-Output Record1: {MAP}

- id: {STRING} "r19lbf3n6hsb"
- name: {STRING} "Azure - Central US (Iowa): INC0102516"
- status: {STRING} "resolved"
- created\_at: {STRING} "2024-02-09T08:17:01.654-08:00"
- updated\_at: {STRING} "2024-02-09T10:23:20.450-08:00"
- monitoring\_at: {STRING} "2024-02-09T09:25:57.986-08:00"
- resolved\_at: {STRING} "2024-02-09T10:23:20.434-08:00"
- impact: {STRING} "minor"
- shortlink: {STRING} "https://stspg.io/x70xg3vz5wfw"
- started\_at: {STRING} "2024-02-09T08:17:01.645-08:00"
- page\_id: {STRING} "94s7z8vpy1n8"
- incident\_updates: {LIST[4]}
- components: {LIST[1]}
- reminder\_intervals: {STRING}
- region: {STRING} "AZURE"

Record Header

Record2: {MAP}

- id: {STRING} "tspfv3y0j1v1"
- name: {STRING} "3 Deployments Affected: INC0101202"

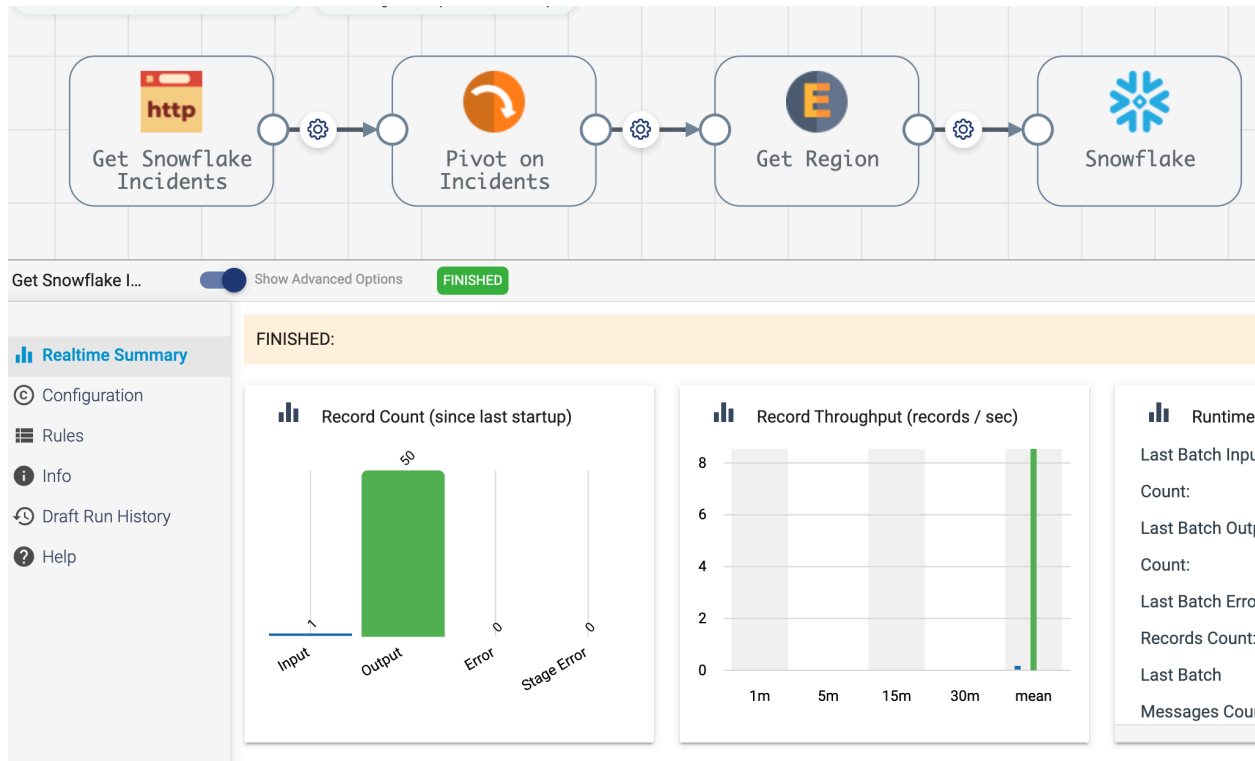
Record2-Output Record1: {MAP}

- id: {STRING} "tspfv3y0j1v1"
- name: {STRING} "3 Deployments Affected: INC0101202"

See the stage's config for the EL expression that performs the extraction.

# Run the pipeline

You should see one record read from the REST API that is pivoted into 50 records written to Snowflake:



Confirm the Snowflake Table has been created and populated (the screenshot below does not show all the rows or columns):

MARK\_DB.MARK\_SCHEMA Settings

```
1 select * from SNOWFLAKE_INCIDENTS
2
3
4
5
6
```

**Results** Chart

	ID	NAME	STATUS	C
3	zvd971z06grg	Azure - East US 2 (Virginia): INC0100718	postmortem	2024-01-18 08
4	6260c7cljgkx	Azure - East US 2 (Virginia): INC0100501	postmortem	2024-01-17 08
5	zgb86hntdyr0	Azure - US Gov (Virginia): MI-20240109	postmortem	2024-01-09 08
6	z770knm7jd2v	AWS - US West (Oregon): INC0099549	postmortem	2024-01-05 08



# Set Parameters in the Transformer for Snowflake Pipeline

Make sure all values here are set:

General Parameters Advanced

Name Snowflake Incidents by Region

Description

Labels Add New...

Snowflake URL https://[redacted]snowflakecomputing.com

Role MARK\_ROLE

Warehouse MARK\_WH

Pipeline Working Schema *i* MARK\_DB MARK\_SCHEMA

Set the name of a target table that does not yet exist for the results of transform in the pipeline's parameters. I have set the default name to `INCIDENTS_BY_REGION`:

General Parameters Advanced

Parameters TARGET\_TABLE : INCIDENTS\_BY\_REGION

+ ADD ANOTHER BULK EDIT MODE

# Preview the Transformer for Snowflake pipeline

Preview the pipeline to see how it works. Use these settings in the Preview Config (with `Write to Destinations` unset) to avoid creating the target table just yet:

## Preview Configuration

Preview Batch Size:

Preview Timeout (in milliseconds):

Run Preview Through Stage:

Time zone:

Write to Destinations:

☐

Show Record/Field Header:

☒

Show Field Type:

☒

# Preview the source records

Preview the source records that were created by the Data Collector pipeline:

Snowflake Incident by Region (mark)

Close Preview

INCIDENTS

Aggregate by Region

Sort

INCIDENTS\_BY...

Preview Stage: INCIDENTS

OUTPUT

ID {String}	NAME {String}	STATUS {String}	CREATED_AT {String}	UI
rl9lbf3n6hsb	Azure - Central US (Iowa): INC0102516	resolved	2024-02-09T08:17:01.654-08:00	20
tspfv3y0j1v1	3 Deployments Affected: INC0101302	postmortem	2024-01-20T19:22:54.866-08:00	20
zvd971z06grg	Azure - East US 2 (Virginia): INC0100718	postmortem	2024-01-18T09:02:05.335-08:00	20
6260c7cljgkx	Azure - East US 2 (Virginia): INC0100501	postmortem	2024-01-17T07:26:40.090-08:00	20

## Preview the aggregation

Preview the aggregation by region:

Snowflake Incident by Region (mark)

Close Preview

INCIDENTS

Aggregate by Region

Sort

#{TARGET}

Preview Stage: Aggregate by Region

≡

©

?

OUTPUT	
REGION {String}	NUM_INCIDENTS {Long}
AZURE	27
OTHER	7
AWS	14
GCP	2

## Preview the sort

Preview the sort:

Snowflake Incident by Region (mark) Close Preview

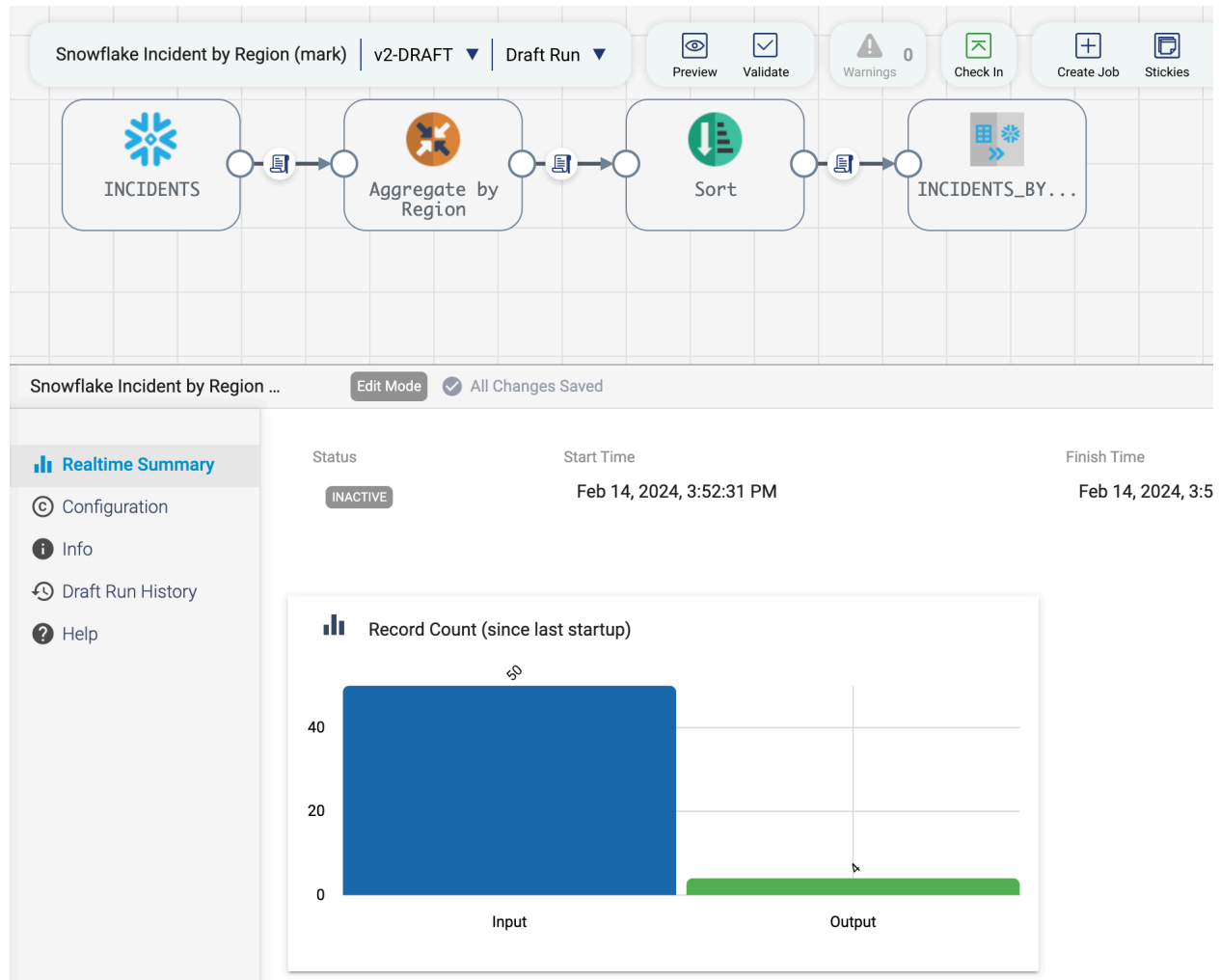
```
graph LR; A[INCIDENTS] --> B[Aggregate by Region]; B --> C[Sort];
```

Preview Stage: Sort

OUTPUT	
REGION {String}	NUM_INCIDENTS {Long}
AZURE	27
AWS	14
OTHER	7
GCP	2

# Run the pipeline

You should see the 50 records read from the source tables and the four records written to the target:



Confirm the results in Snowflake:

```
1 select * from INCIDENTS_BY_REGION
2
3
4
5
6
```

Results		Chart	
	REGION	...	NUM_INCIDENTS
1	AZURE		27
2	AWS		14
3	OTHER		7
4	GCP		2

## Resetting the Demo

To reset the demo, drop the two Snowflake tables