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

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

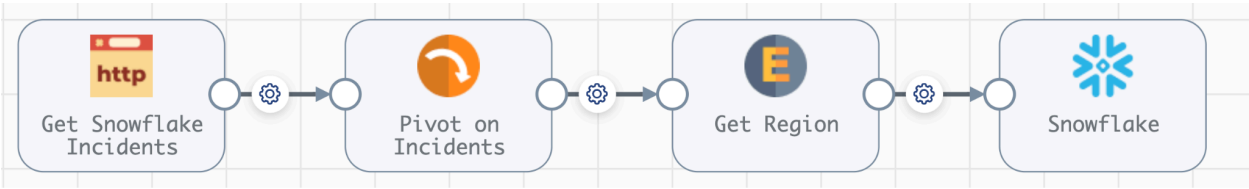
Import the Pipelines

- 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.

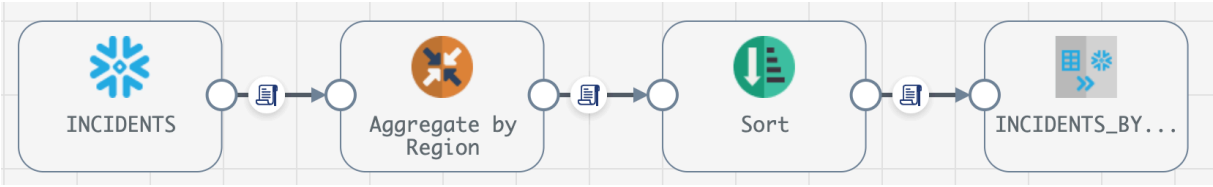
Get Snowflake Incidents (Data Collector pipeline)

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:



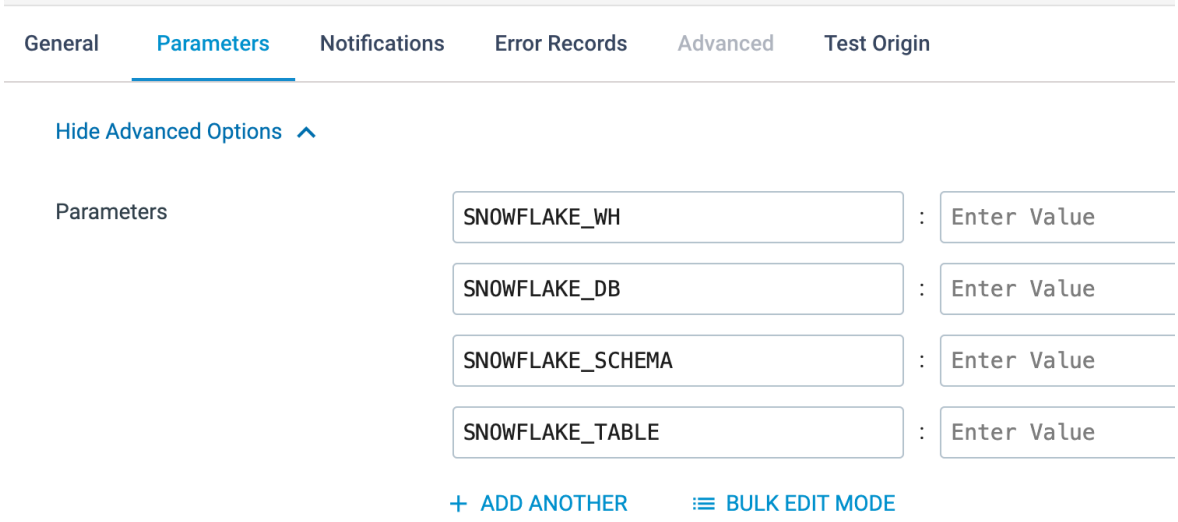
Snowflake Incidents by Region (Transformer for Snowflake pipeline)

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

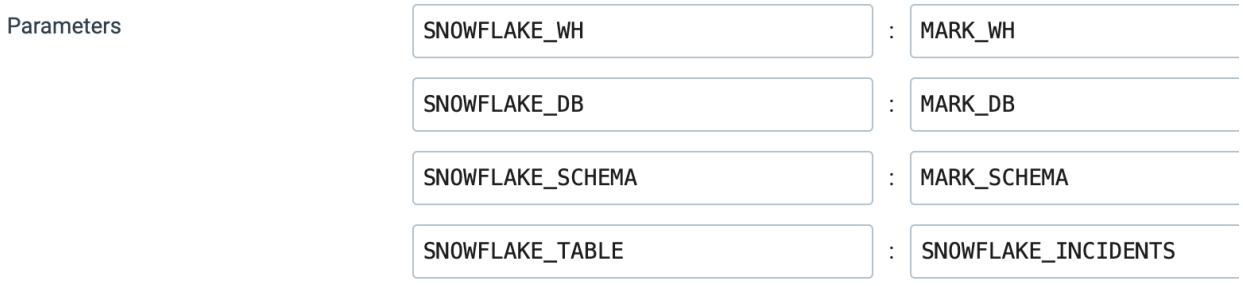


Set Parameters in the Data Collector Pipeline

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.

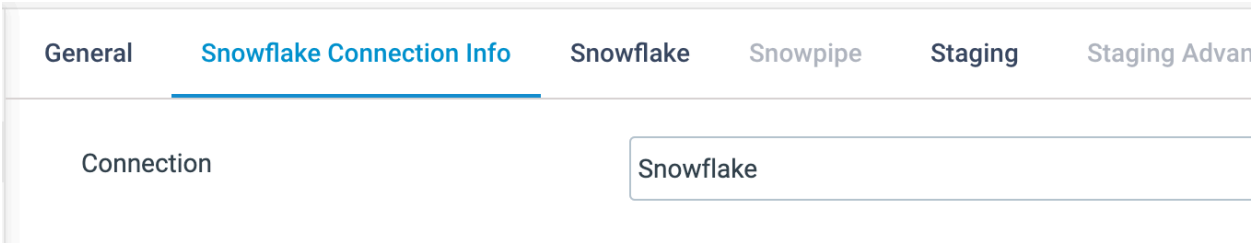


For example, in my environment:



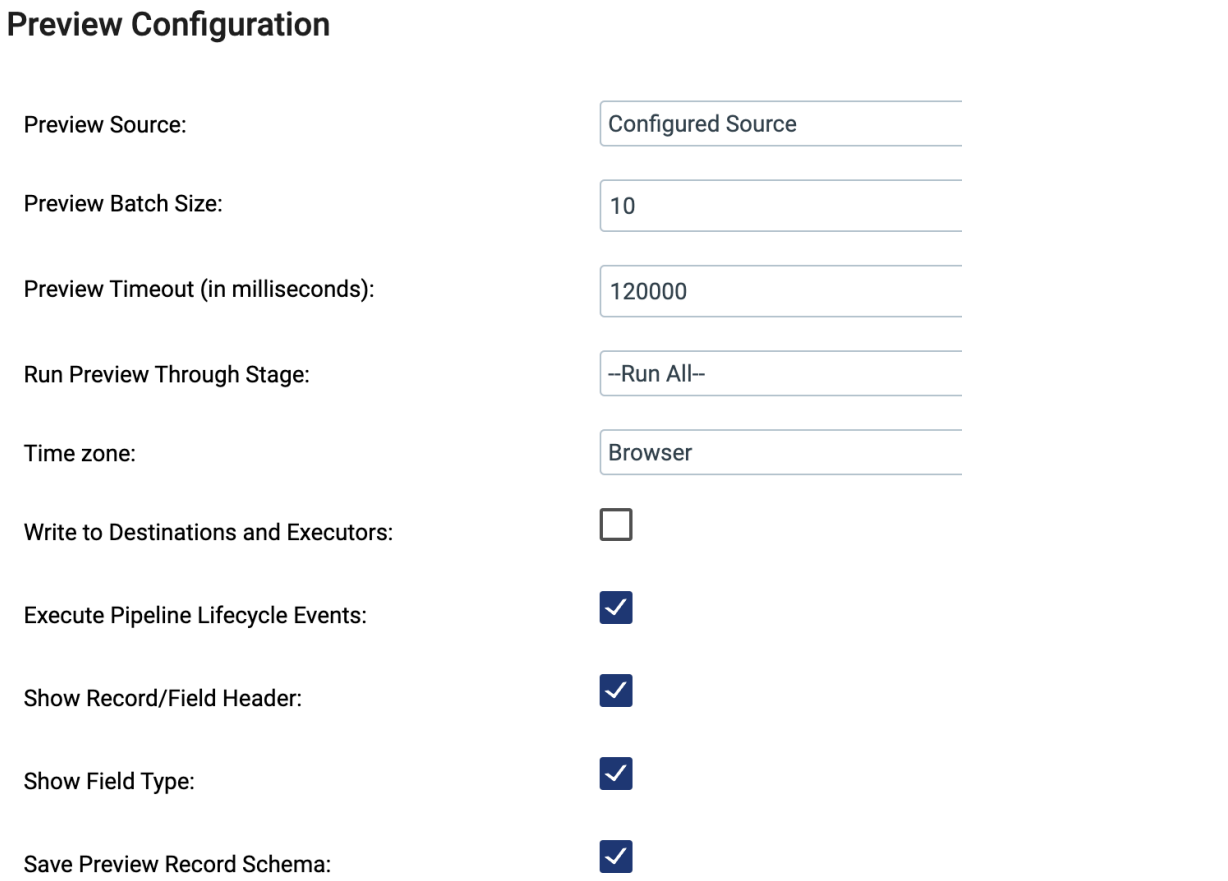
Set a Snowflake Connection in your Data Collector Pipeline

Set a Snowflake Connection in you Data Collector Pipeline:



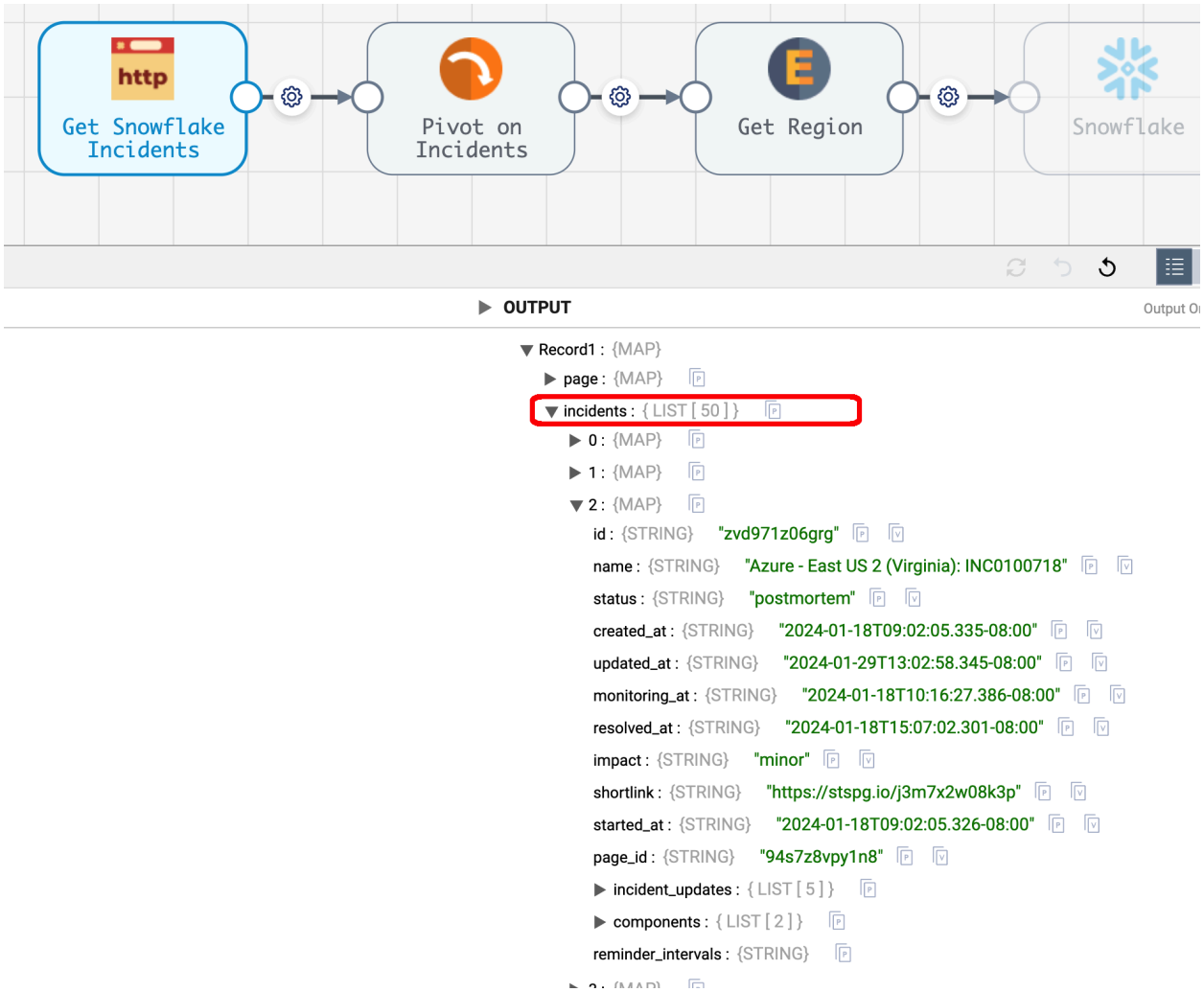
Preview the Data Collector pipeline

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:



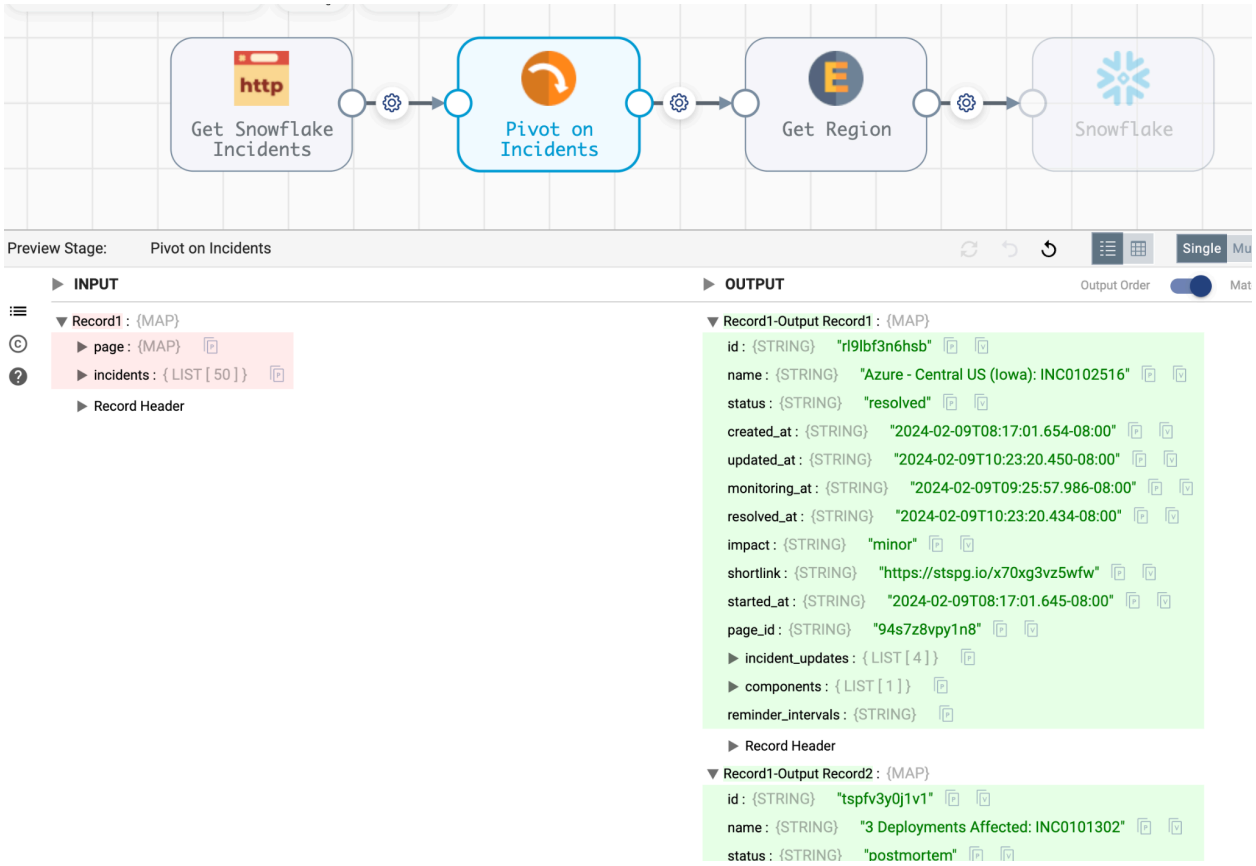
Preview data from the origin

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



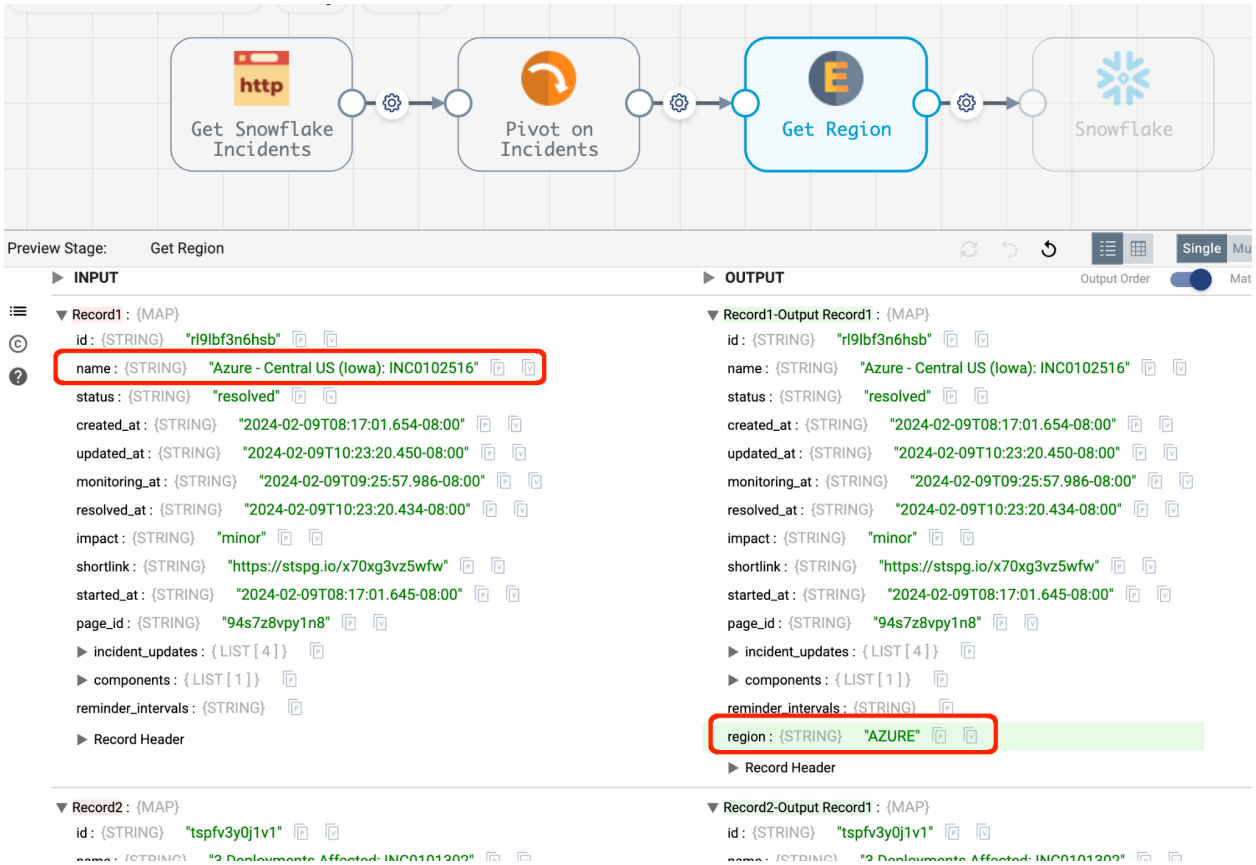
Preview the pivoted incidents

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



Preview the Get Region enrichment

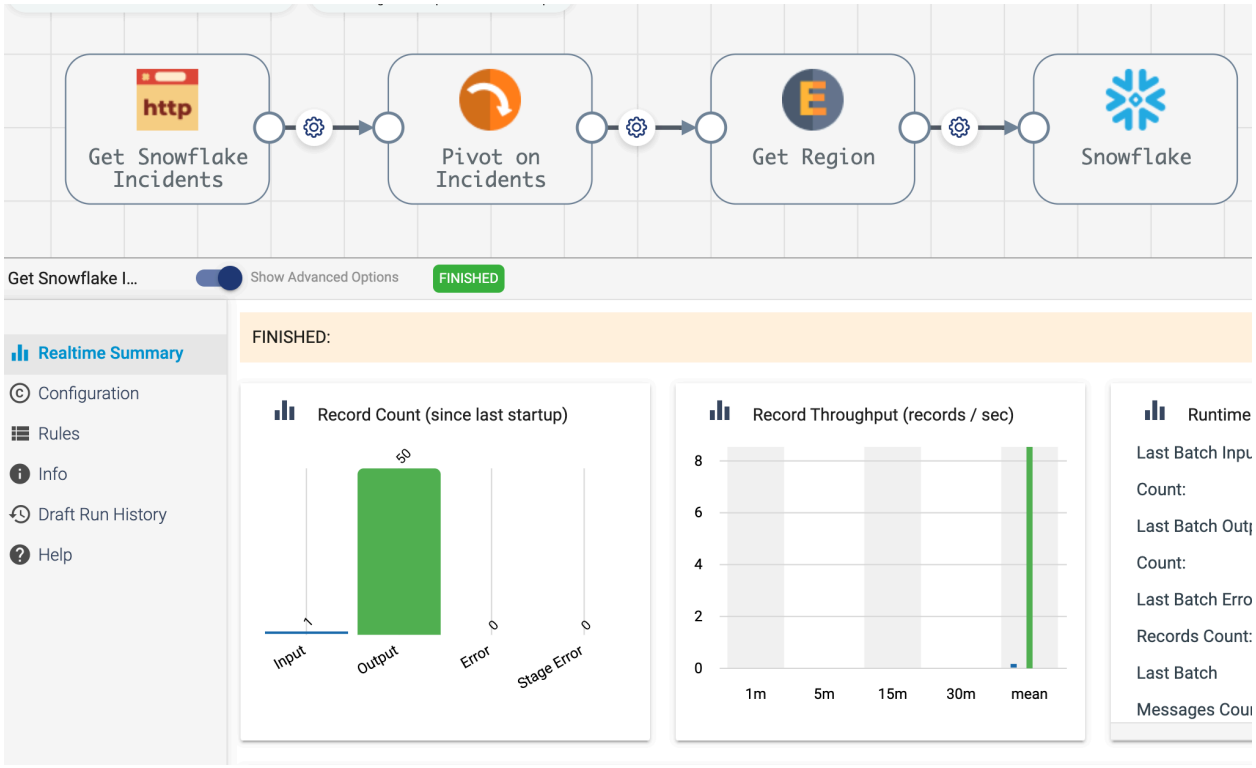
We can see how the region field was extracted from the start of the incident name. We'll use the region field for the Transformer for Snowflake aggregation:



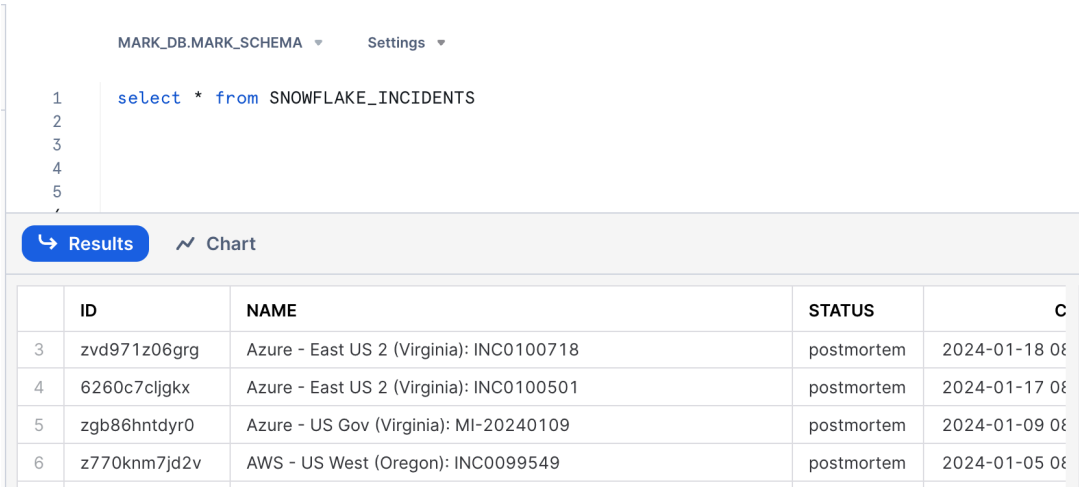
See the stage's config for the EL expression that performs the extraction.

Run the pipeline

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



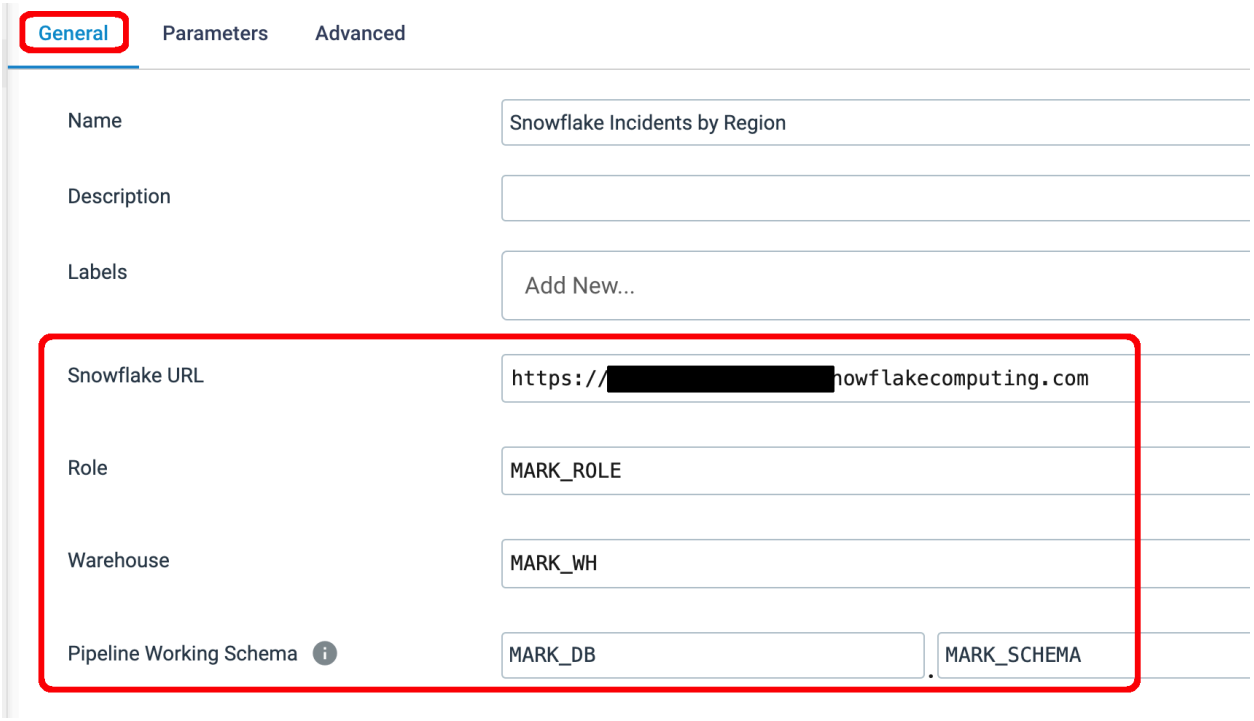
Confirm the Snowflake Table has been created and populated (the screenshot below does not show all the rows or columns):



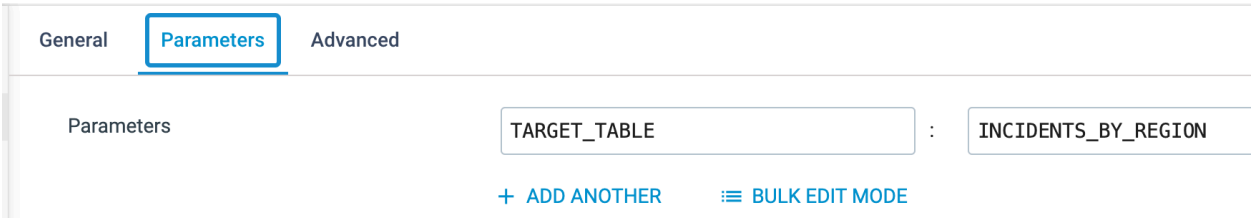
	ID	NAME	STATUS	C
3	zvd971z06grg	Azure - East US 2 (Virginia): INC0100718	postmortem	2024-01-18 08
4	6260c7cljgkx	Azure - East US 2 (Virginia): INC0100501	postmortem	2024-01-17 08
5	zgb86hntdyr0	Azure - US Gov (Virginia): MI-20240109	postmortem	2024-01-09 08
6	z770knm7jd2v	AWS - US West (Oregon): INC0099549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

Set Parameters in the Transformer for Snowflake Pipeline

Make sure all values here are set:

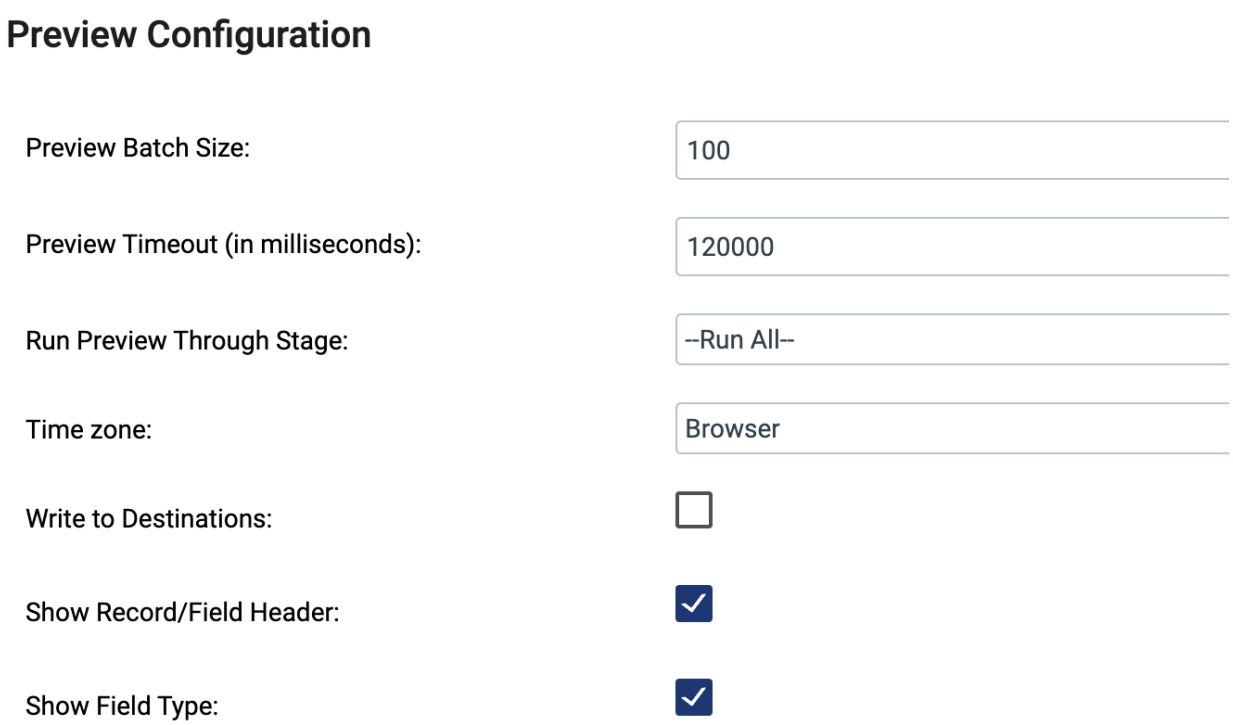


Set the name of a target table that does not yet exist for the results of transform in the pipeline's parameters. I have set the default name to `INCIDENTS_BY_REGION`:



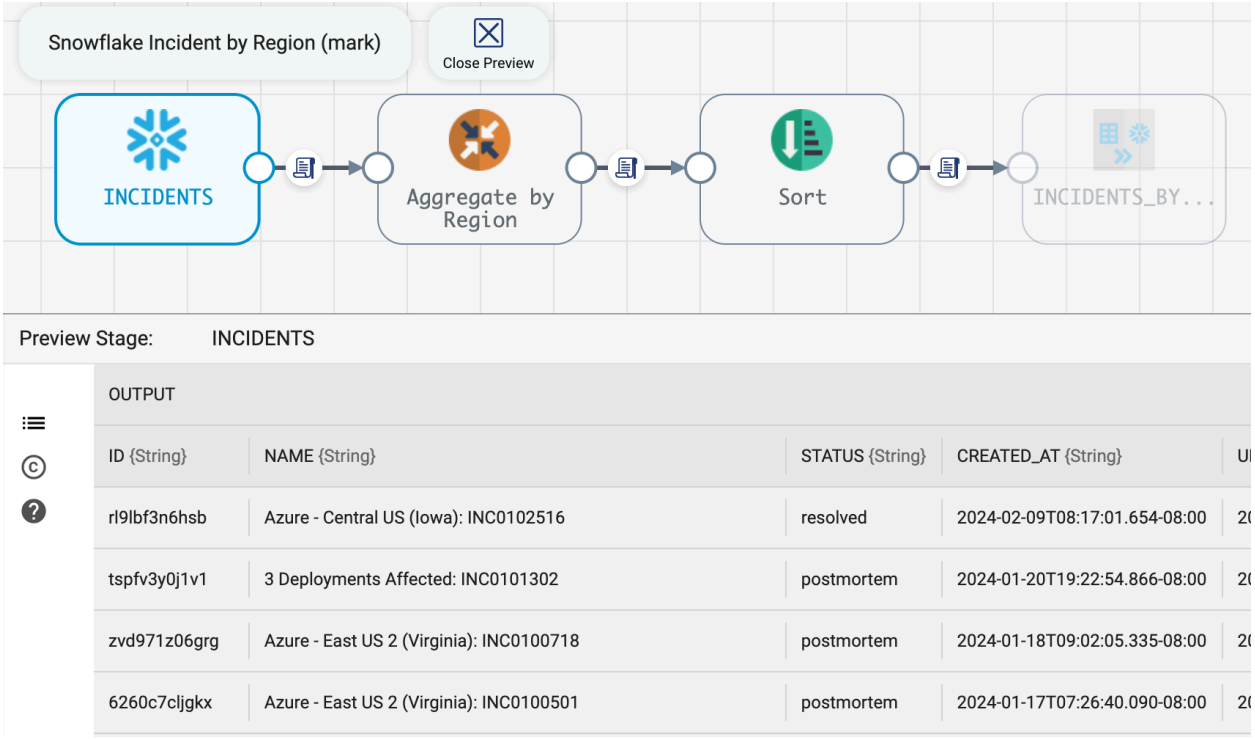
Preview the Transformer for Snowflake pipeline

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:



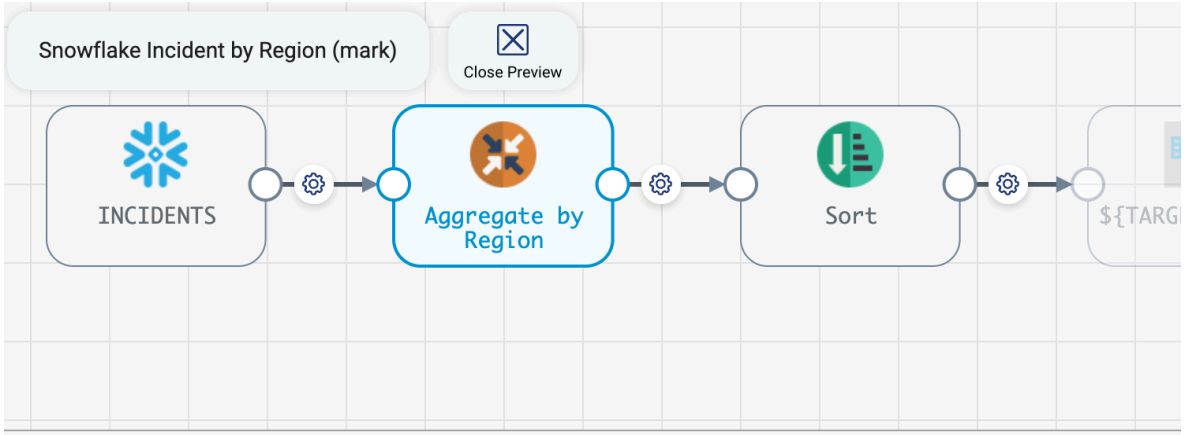
Preview the source records

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



Preview the aggregation

Preview the aggregation by region:

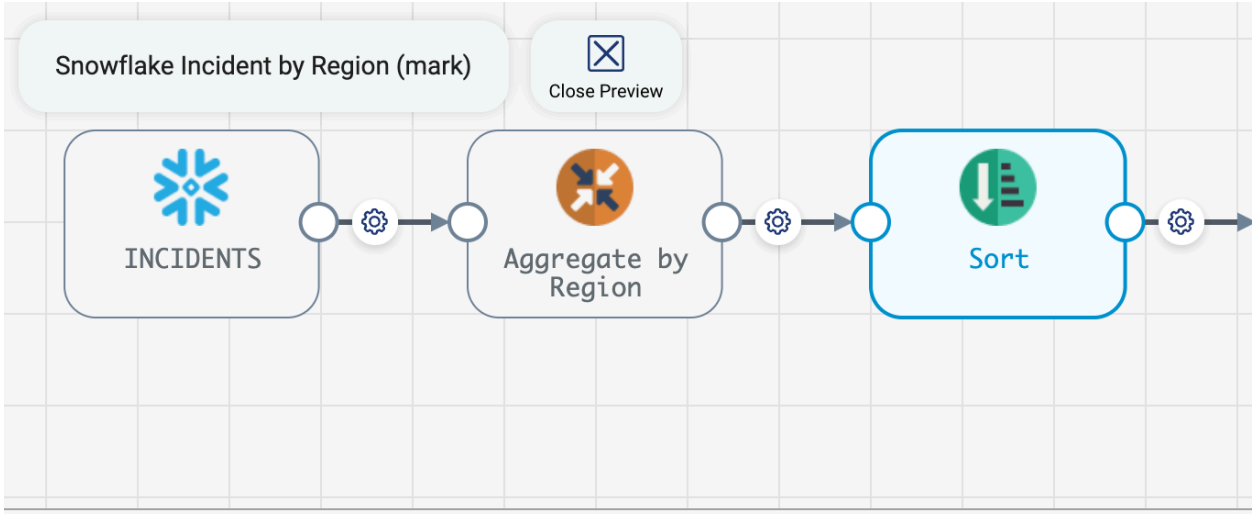


Preview Stage: Aggregate by Region

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

Preview the sort

Preview the sort:

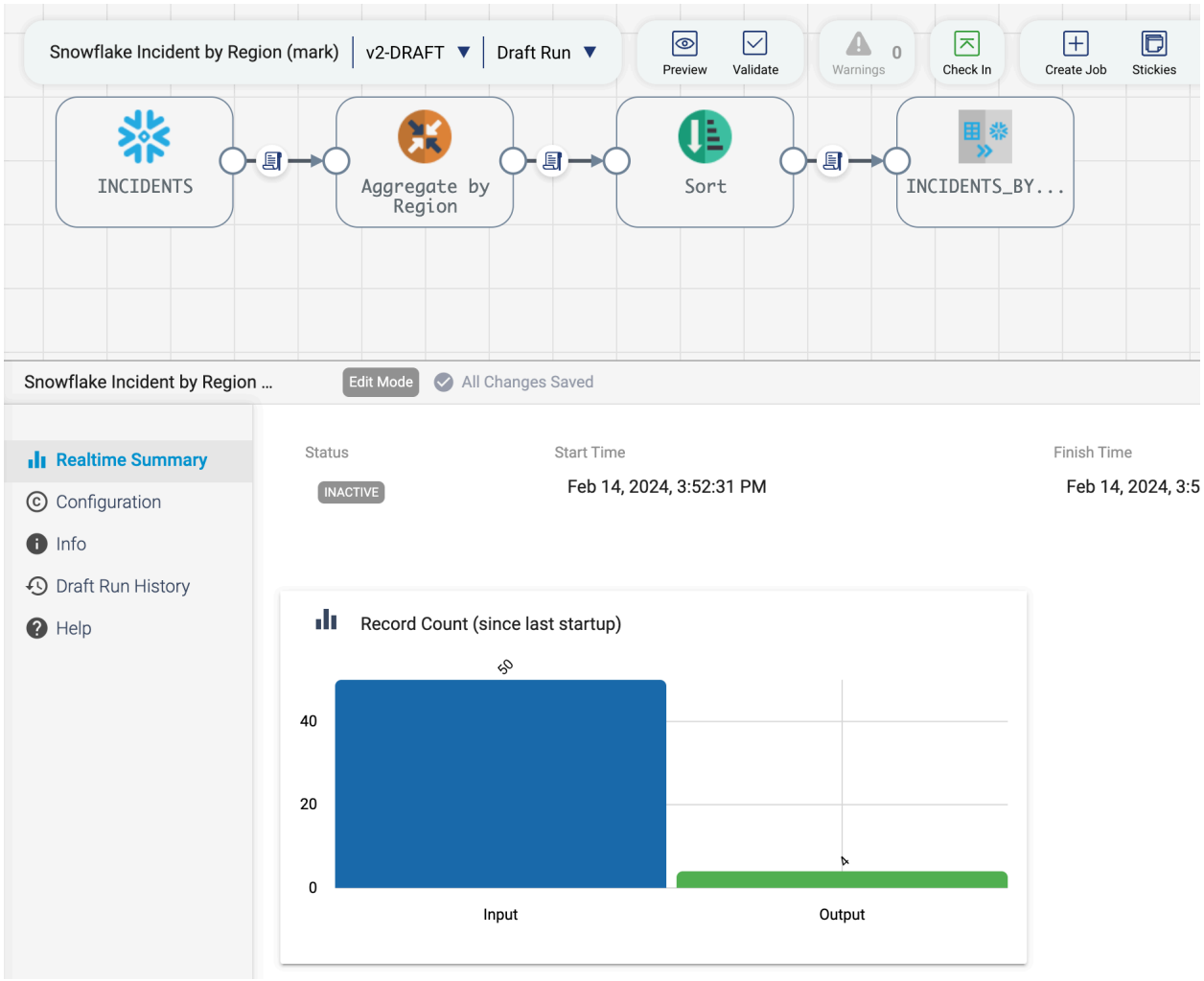


Preview Stage: Sort

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
```

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