Dataflow SQL and Pub/Sub

Overview

An introduction to Dataflow SQL with Cloud Pub/Sub public dataset in a global GCP topic.

Introduction

What is Dataflow?

Dataflow is a managed service for executing a wide variety of data processing patterns. The documentation on this site shows you how to deploy your batch and streaming data processing pipelines using Dataflow, including directions for using service features.

The Apache Beam SDK is an open source programming model that enables you to develop both batch and streaming pipelines. You create your pipelines with an Apache Beam program and then run them on the Dataflow service. The Apache Beam documentation provides in-depth conceptual information and reference material for the Apache Beam programming model, SDKs, and other runners.

Streaming data analytics with speed

Dataflow enables fast, simplified streaming data pipeline development with lower data latency.

Simplify operations and management

Allow teams to focus on programming instead of managing server clusters as Dataflow's serverless approach removes operational overhead from data engineering workloads.

Reduce total cost of ownership

Resource autoscaling paired with cost-optimized batch processing capabilities means Dataflow offers virtually limitless capacity to manage your seasonal and spiky workloads without overspending.

Key features

Automated resource management and dynamic work rebalancing

Dataflow automates provisioning and management of processing resources to minimize latency and maximize utilization so that you do not need to spin up instances or reserve them by hand. Work partitioning is also automated and optimized to dynamically rebalance lagging work. No need to chase down "hot keys" or preprocess your input data.

Horizontal autoscaling

Horizontal autoscaling of worker resources for optimum throughput results in better overall price-to-performance.

Flexible resource scheduling pricing for batch processing

For processing with flexibility in job scheduling time, such as overnight jobs, flexible resource scheduling (FlexRS) offers a lower price for batch processing. These flexible jobs are placed into a queue with a guarantee that they will be retrieved for execution within a six-hour window.

What you will run as part of this

In this codelab, you're going to begin using Dataflow SQL by submitting a SQL statement through the Dataflow SQL UI. You will then explore the pipeline running by using the Dataflow monitoring UI.

What you'll learn

How to submit a SQL statement as a Dataflow job in the Dataflow SQL UI.

How to navigate to the Dataflow Pipeline.

Explore the Dataflow graph created by the SQL statement.

Explore monitoring information provided by the graph.

What you'll need

A Google Cloud Platform project with Billing enabled.

Google Cloud Dataflow and Google Cloud PubSub enabled. You can check those service consoles, but we've used those services previously. Ensure they're enabled.

Using Dataflow SQL

The page explains how to use Dataflow SQL and create Dataflow SQL jobs.

To create a Dataflow SQL job, you must write and run a Dataflow SQL query.

Using the Dataflow SQL editor

The Dataflow SQL editor is a page in the Google Cloud Console where you write and run queries for creating Dataflow SQL jobs.

To access the Dataflow SQL editor, follow these steps:

In the Cloud Console, go to the **Dataflow SQL Editor** page.

Go to Dataflow SQL editor

You can also access the Dataflow SQL editor from the Dataflow monitoring interface by following these steps:

In the Cloud Console, go to the Dataflow **Jobs** page.

[Go to Jobs]

Click **Create job from SQL**.

Writing Dataflow SQL queries

Dataflow SQL queries use the [Dataflow SQL query syntax]. The Dataflow SQL query syntax is similar to [BigQuery standard SQL].

You can use the [Dataflow SQL streaming extensions] to aggregate data from continuously updating Dataflow sources like Pub/Sub.

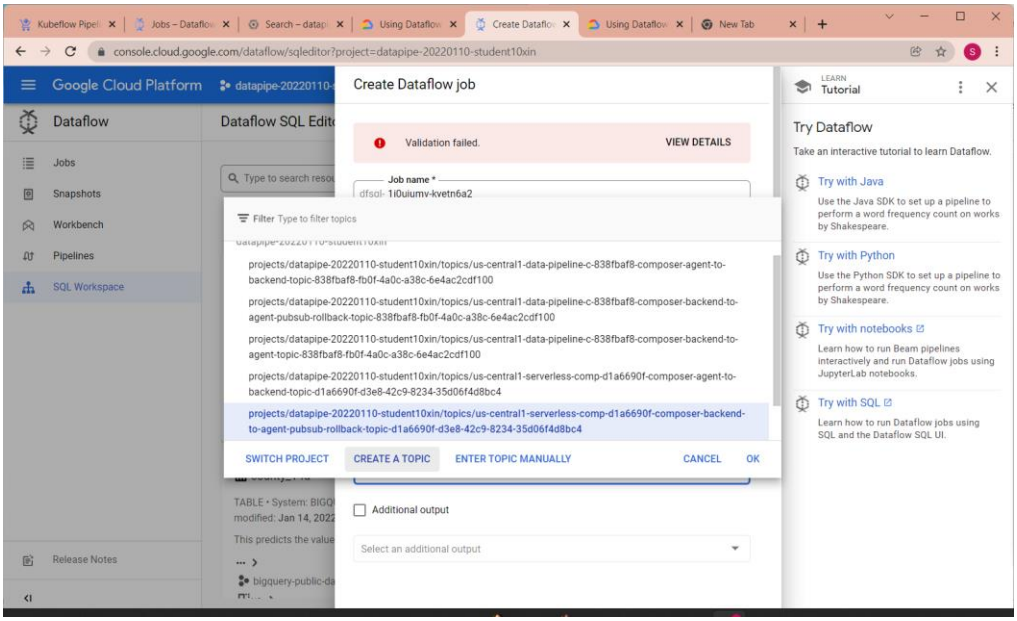
For example, the following query counts the passengers in a Pub/Sub stream of taxi rides every minute:

```
SELECT
  TUMBLE_START('INTERVAL 1 MINUTE') as period_start,
  SUM(passenger_count) AS pickup_count
FROM pubsub.topic.`pubsub-public-data`.`taxirides-realtime`
WHERE
  ride_status = "pickup"
GROUP BY
  TUMBLE(event_timestamp, 'INTERVAL 1 MINUTE')
```

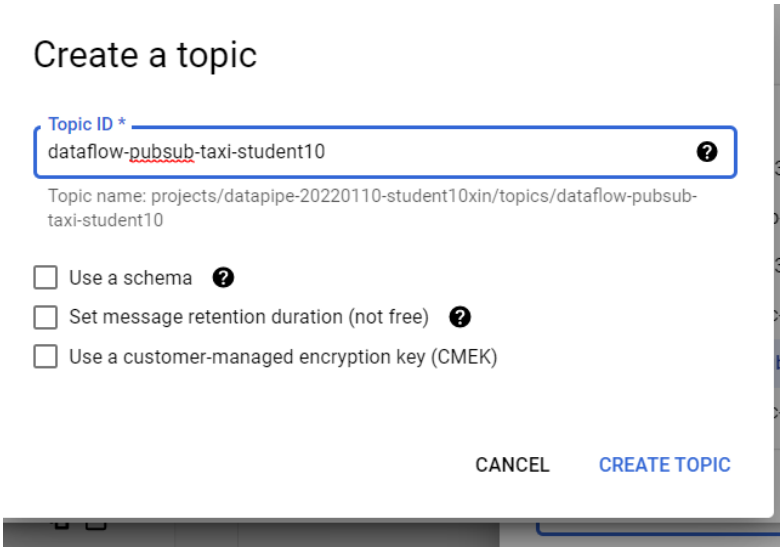
From the Dataflow SQL console

Select **CREATE JOB**

Select PubSub for the output



Choose **Create Topic**



Enter dataflow-pubsub-taxi-student###

Using your Student Number for the session

Choose **CREATE TOPIC**

Choose **CREATE JOB**

Running Dataflow SQL queries

When you run a Dataflow SQL query, Dataflow turns the query into an Apache Beam pipeline and runs the pipeline.

You can run a Dataflow SQL query using the Cloud Console or gcloud command-line tool.

Consolegcloud

To run a Dataflow SQL query, use the gcloud dataflow sql query command. The following is an example SQL query that creates

```
gcloud dataflow sql query \
  --job-name=JOB_NAME \
  --region=REGION \
  --bigquery-table=BIGQUERY_TABLE \
  --bigquery-dataset=BIGQUERY_DATASET \
  --bigquery-project=BIGQUERY_PROJECT \
'SQL_QUERY'
```

Replace the following:

JOB_NAME: a name for your Dataflow SQL job

REGION: the regional endpoint for deploying your Dataflow job

BIGQUERY_TABLE: the name of the BigQuery table to which you want to write the output

BIGQUERY_DATASET: the BigQuery dataset ID that contains the output table

BIGQUERY_PROJECT: the Cloud project ID that contains the output BigQuery table

SQL_QUERY: your Dataflow SQL query

Note: Starting a Dataflow SQL job might take several minutes. You cannot update a Dataflow SQL job after creating it.

For more information about querying data and writing Dataflow SQL query results, see Using data sources and destinations.

Setting pipeline options

You can set Dataflow pipeline options for Dataflow SQL jobs. Dataflow pipeline options are execution parameters that configure how and where to run Dataflow SQL queries.

To set Dataflow pipeline options for Dataflow SQL jobs, specify the following parameters when you run a Dataflow SQL query.

Consolegcloud

Flag	Type	Description	Default value
--region	String	The region to run the query in. Dataflow SQL queries can be run in regions that have a Dataflow regional endpoint.	If not set, throws an error.
--max-workers	int	The maximum number of Compute Engine instances available to your pipeline during execution.	If unspecified, Dataflov automatically determines a appropriate number o workers.
--num-workers	int	The initial number of Compute Engine instances to use when executing your pipeline. This parameter determines how many workers Dataflow starts up when your job begins.	If unspecified, Dataflov automatically determines a appropriate number o workers.

--worker-region	String	The Compute Engine region for launching worker instances to run your pipeline. The Compute Engine worker region can be in a different region than the Dataflow regional endpoint. You can specify one of --worker-region or --worker-zone.	If not set, defaults to th specified Dataflow regiona endpoint.
--worker-zone	String	The Compute Engine zone for launching worker instances to run your pipeline. The Compute Engine zone can be in a different region than the Dataflow regional endpoint. You can specify one of --worker-region or --worker-zone.	If not set, defaults to a zone i the specified Dataflow regiona endpoint.
--worker-machine-type	String	The Compute Engine machine type that Dataflow uses when starting workers. You can use any of the available Compute Engine machine type families as well as custom machine types. For best results, use n1 machine types. Shared core machine types, such as f1 and g1 series workers, are not supported under the Dataflow Service Level Agreement. Note that Dataflow bills by the number of vCPUs and GB of memory in workers. Billing is independent of the machine type family.	If not set, Dataflov automatically chooses th machine type.
--service-account-email	String	The email address of the controller service account with which to run the pipeline. The email address must be in the form my-service-account-name@<project-id>.iam.gserviceaccount.com.	If not set, Dataflow worker use the Compute Engin service account of the curren project as the controlle service account.
--disable-public-ips	boolean	Specifies whether Dataflow workers use public IP addresses. If set, Dataflow workers use private IP addresses for all communication.	If not set, Dataflow worker use public IP addresses.
--network	String	The Compute Engine network to which workers are assigned.	If not set, defaults to th network default.
--subnetwork	String	The Compute Engine subnetwork to which workers are assigned. The subnetwork must be in the form regions/***region***/subnetworks/***subnetwork***.	If not set, Dataflov automatically determine subnetwork.
--dataflow-kms-key	String	The customer-managed encryption key (CMEK) used to encrypt data at rest. You can control the encryption key through Cloud KMS. The key must be in the same location as the job.	If unspecified, Dataflow use the default Google Clou encryption instead of a CMEK.

For more information, see the gcloud dataflow sql query command reference.

Note: Dataflow SQL jobs use autoscaling and Dataflow automatically chooses the execution mode (batch or streaming). You cannot control this behavior for Dataflow SQL jobs.

Stopping Dataflow SQL jobs

To stop a Dataflow SQL job, you must cancel it. Stopping a Dataflow SQL job with the drain option is not supported.

Congratulations!

In this experiment, you created and ran a Dataflow SQL pipeline using a global public Cloud Pub/Sub dataset.