Running Cloud Dataflow Pipelines Using Templates

Janani Ravi
CO-FOUNDER, LOONYCORN

www.loonycorn.com

Overview

Using built-in Dataflow templates for pipeline creation and execution

Creating custom pipeline templates from Dataflow jobs

Configuring and running custom pipeline templates

Dataflow Templates

Dataflow is very hard for non-developers to use - Dataflow Templates attempt to change that

Dataflow Templates

Dataflow can be intimidating to get started with

- Needs dependencies to be set up

- Code to be written

Dataflow Templates are a great way to get started quickly

Dataflow Templates

Start with Google-provided templates for common tasks

Do not recompile code

Use runtime parameters to customize execution

Dataflow Templates

No need to configure dependencies

Have developers define templates

Non-developers can use those templates without writing any code

Traditional Dataflow Jobs

Configure Dependencies

Apache Beam SDK

Done by developer

Execute Pipeline

Still within dev environment, usually done by developer

Apache Beam SDK creates job request on GCS and submits to Dataflow

Write Code for Pipeline

Python or Java code

Done by developer

Templated Dataflow Jobs

Configure Dependencies

Apache Beam SDK

Done by developer

Create Template by Executing Pipeline

Still within dev environment, done by developer

Apache Beam SDK creates template file and merely stores in GCS

Write Code for Pipeline

Python or Java code

Done by developer

Execute Pipeline

Using gcloud, web console or REST API

Can be done by non-developer

Google-provided Templates

BigTable to Cloud Storage

Pub/Sub to BigQuery

Cloud Storage to Pub/Sub

...

Many more intra-GCP transfers

Demo

Executing built-in templates in Dataflow for common processing operations

Demo

Creating and executing custom
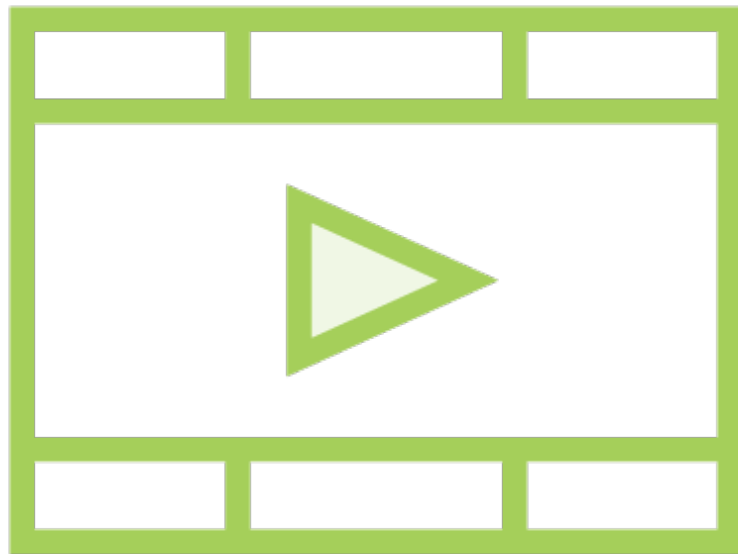Dataflow templates

Summary

Using built-in Dataflow templates for pipeline creation and execution

Creating custom pipeline templates from Dataflow jobs

Configuring and running custom pipeline templates

Related Courses

Handling Streaming Data with GCP Dataflow

Conceptualizing the Processing Model for Apache Flink