Monitoring Jobs in Cloud Dataflow

Janani Ravi
CO-FOUNDER, LOONYCORN

www.loonycorn.com

Overview

Autoscaling of workers in Dataflow jobs

Observing job metrics on the Dataflow monitoring interface

Monitoring jobs using the command-line

Monitoring jobs using Cloud Monitoring

Monitoring Dataflow Jobs

Monitoring Dataflow Jobs

Dataflow monitoring interface

Command-line interface

Cloud Monitoring

Logging messages

Dataflow Monitoring Interface

Google Cloud Console to view job information

A list of all jobs run in the last 30 days

Graphical representation of pipeline

Job status, execution, and SDK version

Links to cloud services running your pipeline

Errors or warnings

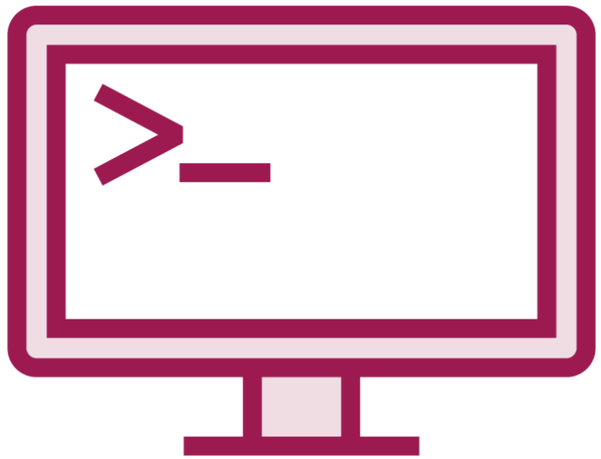
Dataflow Monitoring Interface

Step-level visibility to metrics

Helps identify slow stages and pipeline lag

I/O metrics to identify bottlenecks in sources and sinks

Command-line Interface



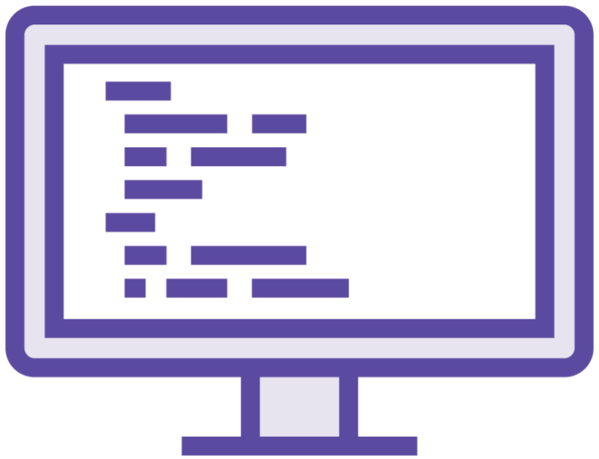
Part of the `gcloud` command line tool

View jobs and their status

View logs

View metrics

Cloud Monitoring



Provides logging and diagnostic tools for all GCP services

Allows access to Dataflow metrics such as:

- Job status
- Element counts
- System lag

Alerting capabilities to send notifications for errors or failed jobs

Logging Messages

Use the Apache Beam SDK's built-in logging infrastructure

Recommended library is the SLF4J (Simple Logging Facade for Java)

Different log types available

job-message, worker, worker-startup, shuffler, docker, and kubelet logs

Demo

Executing a Apache Beam pipeline on Cloud Dataflow and monitoring it using the Dataflow Monitoring Interface

Demo

Configuring the number of workers for autoscaling

Demo

Enabling job execution using the streaming engine

Demo

Monitoring jobs using the command-line interface

Demo

Logging messages in Dataflow using SLF4J

Demo

Using the Cloud Monitoring to view metrics and set up alerts

Summary

Autoscaling of workers in Dataflow jobs

Observing job metrics on the Dataflow monitoring interface

Monitoring jobs using the command-line

Monitoring jobs using Cloud Monitoring

Up Next:

Optimizing Cloud Dataflow Pipelines