



# Dataflow Monitoring

Jérémie Gomez  
Data consultant, Google Cloud  
[medium.com/@foup](https://medium.com/@foup)





# Metrics for Dataflow

We can think of metrics for Dataflow as different types.

- Native metrics
- Worker metrics
- Monitoring agent metrics
- Custom metrics

They all show up in Cloud Monitoring. The most important ones are also shown in the Dataflow UI.



# Native metrics (1)

## Frequently used ones include:

- `dataflow.googleapis.com/job/is_failed`
- `dataflow.googleapis.com/job/system_lag`
- `dataflow.googleapis.com/job/per_stage_system_lag` (per stage)
- `dataflow.googleapis.com/job/current_num_vcpus`
- `dataflow.googleapis.com/job/element_count` (per PCollection)

## Some metrics are influenced by time, such as:

- `dataflow.googleapis.com/job/total_memory_usage_time`
- `dataflow.googleapis.com/job/total_vcpu_time`



## Native metrics (2)

**Some additional metrics are available if your job reads from Pub/Sub, such as:**

- `dataflow.googleapis.com/job/pubsub/read_latencies`

**Some additional metrics about the amount of logs your jobs write, such as:**

- `logging.googleapis.com.byte_count`

Full list: [https://cloud.google.com/monitoring/api/metrics\\_gcp#gcp-dataflow](https://cloud.google.com/monitoring/api/metrics_gcp#gcp-dataflow)



# Worker metrics

## Frequently used metrics include:

- `compute.googleapis.com/instance/cpu/utilization`
- `compute.googleapis.com/instance/disk/write_bytes_count`
- `compute.googleapis.com/guest/disk/bytes_used`
- `compute.googleapis.com/instance/memory/balloon/ram_used` (only for E2 machines)

Full list: [https://cloud.google.com/monitoring/api/metrics\\_gcp#gcp-compute](https://cloud.google.com/monitoring/api/metrics_gcp#gcp-compute)



# Monitoring agent metrics (1)

## Enabling the agent

To monitor persistent disk, CPU, network, and process metrics from your Cloud Dataflow worker instances, use the pipeline option

```
--experiments=enable_stackdriver_agent_metrics
```

These metrics are chargeable.



# Monitoring agent metrics (2)

**Some useful metrics from the agent include:**

- `agent.googleapis.com/cpu/utilization`
- `agent.googleapis.com/disk/bytes_used`
- `agent.googleapis.com/memory/percent_used` (not to be confused with `agent.googleapis.com/agent/memory_usage`)

Full list: [https://cloud.google.com/monitoring/api/metrics\\_agent](https://cloud.google.com/monitoring/api/metrics_agent)



# Custom metrics

**Counter:** Metric that can be incremented and decremented.

**Distribution:** Metric that records various statistics about the distribution of reported values.

```
class SomeDoFn extends DoFn<String, String> {  
  
    private Counter counter = Metrics.counter(SomeDoFn.class, "my-counter");  
    @ProcessElement  
    public void processElement(ProcessContext c) {  
        counter.inc();  
        Metrics.counter(SomeDoFn.class, "my-counter2").inc();  
    }  
}
```





# Which metrics for common use cases? (1)

## **Has my job failed?**

`job/is_failed > 0`, filter by `job_name`

## **Is there lag ?**

`job/system_lag`, filter by `job_name` or `job/per_stage_system_lag`, filter by `job_name` and `stage`

## **Is there a spike in processing?**

`/job/current_num_vcpus` to know if the job has scaled, `/job/element_count` or `job/elements_produced_count` (throughput) on an upstream PCollection, **pubsubtool**

## **Is data processed fresh?**

`/job/per_stage_data_watermark_age` or `/job/data_watermark_age` (a.k.a data freshness)



## Which metrics for common use cases? (2)

### **What is my CPU utilization?**

`compute.googleapis.com/instance/cpu/utilization` or `agent.googleapis.com/cpu/utilization`

### **Is my memory close to full?**

`dataflow.googleapis.com/job/total_memory_usage_time` (not easy to alert on)

`compute.googleapis.com/instance/memory/balloon/ram_used` (only for E2 machines)

`agent.googleapis.com/memory/bytes_used`

`agent.googleapis.com/memory/percent_used`

### **Is a dependency failing?**

Use metrics specialized for your dependency (e.g. Memorystore). Use your custom metric (e.g. number of times that connecting to your dependency failed), or use a log-based metric.



# Setting up alerts



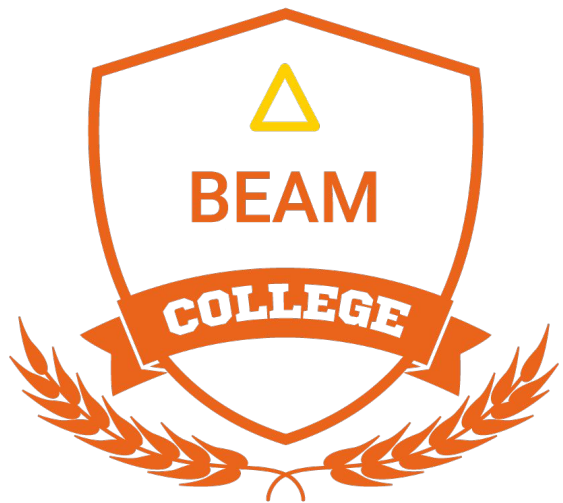
## Choose your policy

- Add conditions: e.g. mean data freshness has been above 10 minutes for 2 minutes.
- Choose policy trigger: ALL conditions are met, ANY condition is met, ALL conditions are met for different resources for each condition.



## Choose your notification channel

- Choose between PagerDuty, Slack, Webhooks, Email, SMS or Cloud Pub/Sub.
- The Pub/Sub channel will enable you to integrate with other GCP components, or with third-party systems.



# Demo

Cloud Monitoring & Dataflow UI



Thank you!

Q & A

# Demo ideas

## Cloud Monitoring

- Show an important metric of each type in the metrics explorer in Cloud Monitoring
- Show an already-made dashboard
- Show how to create an alert (and maybe what it looks like?)

## Dataflow UI

- Show job list, click “running”, sort function, filter function
- Show job info panel
- Show job graph, click steps and substeps, show the associated metrics & stage fusion
- Show job metrics. Use time selector. Show “create alert policy”.

