Introduction to Cloud Monitoring

This document provides an overview of Cloud Monitoring, which is one part of Google Cloud's operations suite. Cloud Monitoring is integrated with most Google Cloud services, and it automatically collects and stores performance information about those services. It can also collect system and application metrics from third-party applications. The data visualization and analysis tools provided by Cloud Monitoring help you answer important questions like the following:

- What is the load on my service?
- Is my website responding correctly?
- Is my service performing well?

Cloud Monitoring provides Google Cloud console and API support for most of its services, and the Cloud Monitoring API reference pages, such as the page `alertPolicies.list` (/monitoring/api/ref_v3/rest/v3/projects.alertPolicies/list), let you experiment with API calls directly from the reference page.

This document is intended for developers and system administrators who need to monitor the performance of a service or system.

Monitor the load on a service

To understand the current load on a service, or to view the performance data of your service for the past month, use the charts and dashboards (/monitoring/dashboards) tools. You can chart and monitor any (numeric) metric data that your Google Cloud project collects, including the following:

- System metrics generated by Google Cloud services. These metrics provide information about how the service is operating. For example, Compute Engine reports more than 25 unique metrics for each virtual machine (VM) instance. For a complete list of metrics, see Google Cloud metrics (/monitoring/api/metrics_gcp).
- System and application metrics (/monitoring/api/metrics_agent#agent-agent) that the Cloud Monitoring agent (/monitoring/agent) gathers. These metrics provide additional information about system resources and applications running on Compute Engine instances and on Amazon Elastic Compute Cloud (Amazon EC2) instances. Optionally, you can configure the agent to collect metrics from third-party plugins

(/monitoring/agent/plugins) such as Apache or Nginx web servers, or MongoDB or PostgreSQL databases.

- Custom metrics (/monitoring/custom-metrics) that your service writes by using the Cloud Monitoring API (/monitoring/docs/apis) or by using a library like OpenCensus.
- Logs-based metrics (/logging/docs/logs-based-metrics), which collect numeric information about the logs written to Cloud Logging (/logging/docs). Google-defined logs-based metrics include counts of errors that your service detects and the total number of log entries received by your Google Cloud project. You can also define logs-based metrics. For example, you might create a metric that counts the number of `404 Not Found` errors for an application deployed to App Engine.

To visualize your data to see trends, identify outliers, and view other details about your data, you can use the following tools:

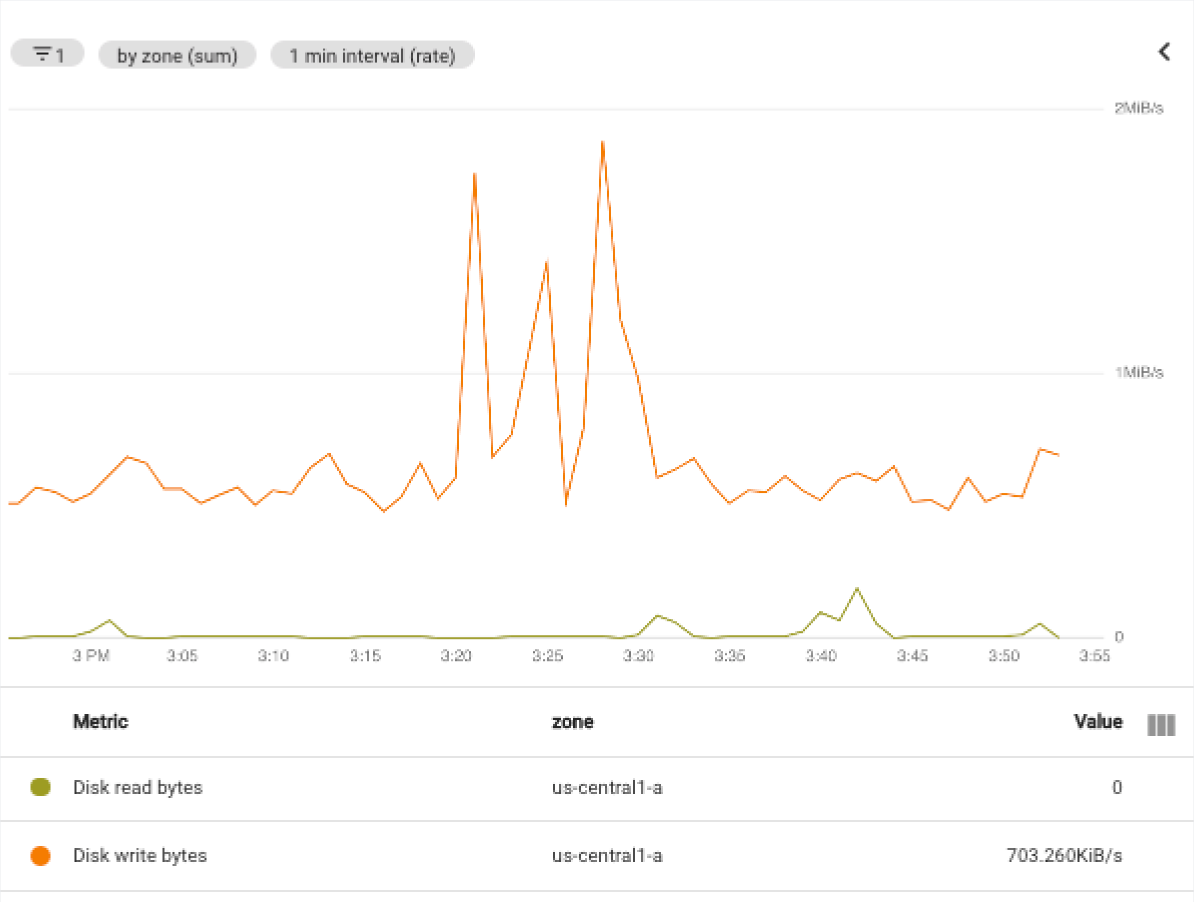
- Google Cloud dashboards (/monitoring/charts/predefined-dashboards): Cloud Monitoring automatically creates these dashboards based on the resources used by your Google Cloud project.

  For example, when a Google Cloud project contains Compute Engine VM instances, dashboards for those VM instances and disks are created automatically. By using the **VM instances** dashboard, you can view details such as memory and disk usage, identify IP addresses, and identify which VMs are dropping network packets. This dashboard also displays information about your usage of the Cloud Monitoring agent and provides suggestions for instrumentation.

- Custom dashboards (/monitoring/charts/dashboards): You create or install (/monitoring/dashboards/dashboard-templates) these dashboards. Custom dashboards let you define what data you want to view and how to view that data. For example, you can display metric data, alerting policies, and logs stored in your Google Cloud project. You can display time-series data on a chart, with a gauge or scorecard, or in tabular form. Dashboards also support text widgets. You can create a custom dashboard with the Dashboards API (/monitoring/dashboards/api-dashboard) or with the Google Cloud console (/monitoring/charts/dashboards).
- Charts: You can add charts to a custom dashboard or you can use Metrics Explorer (/monitoring/charts/metrics-explorer), which is a charting tool designed to let you quickly chart and explore time-series data. You can save charts created with Metrics Explorer (/monitoring/charts/metrics-explorer#save) to a custom dashboard.

When you create a chart, you select the time-series data that you want to view. For example, you can configure a chart to display data for Compute Engine VM instances that are located in the **us-east-1d** zone (/compute/docs/regions-zones).

The chart settings let you compare current data to previous data, display outliers and percentiles, and display multiple metrics. For example, the following screenshot shows a chart that displays the number of bytes both read and written by a single VM:



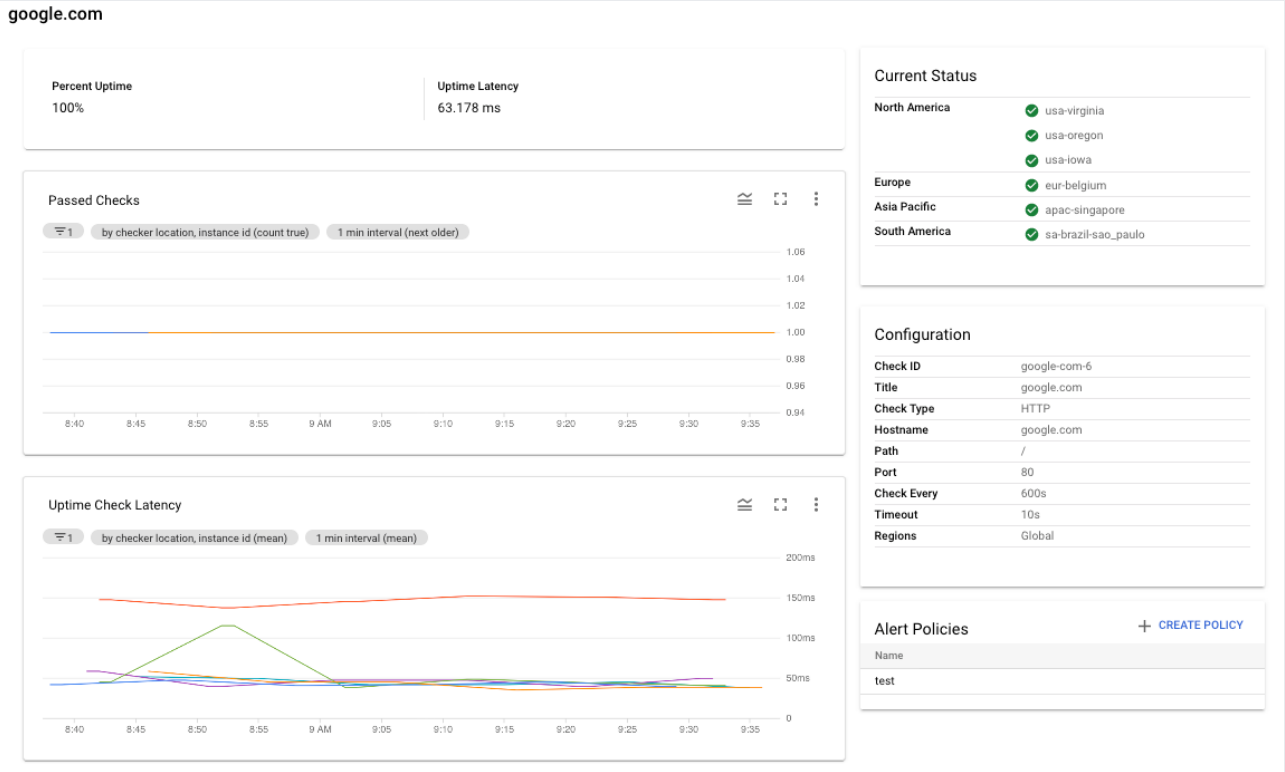
	Metric	zone	Value
●	Disk read bytes	us-central1-a	0
●	Disk write bytes	us-central1-a	703.260KiB/s

For more information about viewing time-series data, see Using dashboards and charts (/monitoring/dashboards).

Monitor website availability

To monitor whether a website is responding, configure an uptime check (/monitoring/uptime-checks). These checks periodically probe your service in a way that mimics how your customers access your service, and then they record the success and latency of the probe.

To view information about your uptime checks, Cloud Monitoring provides a dashboard that summarizes the status of each uptime check, and for each check, it provides a dashboard with detailed information. The detail view for an uptime check displays the success or failure of the response and the latency of the response, along with details about the uptime check:



google.com

Percent Uptime
100%

Uptime Latency
63.178 ms

Passed Checks

Uptime Check Latency

Current Status

North America	✓ usa-virginia ✓ usa-oregon ✓ usa-iowa
Europe	✓ eur-belgium
Asia Pacific	✓ apac-singapore
South America	✓ sa-brazil-sao_paulo

Configuration

Check ID	google-com-6
Title	google.com
Check Type	HTTP
Hostname	google.com
Path	/
Port	80
Check Every	600s
Timeout	10s
Regions	Global

Alert Policies

Name
test

For more information about this topic, see Managing uptime checks (/monitoring/uptime-checks).

Get notified when a service isn't performing well

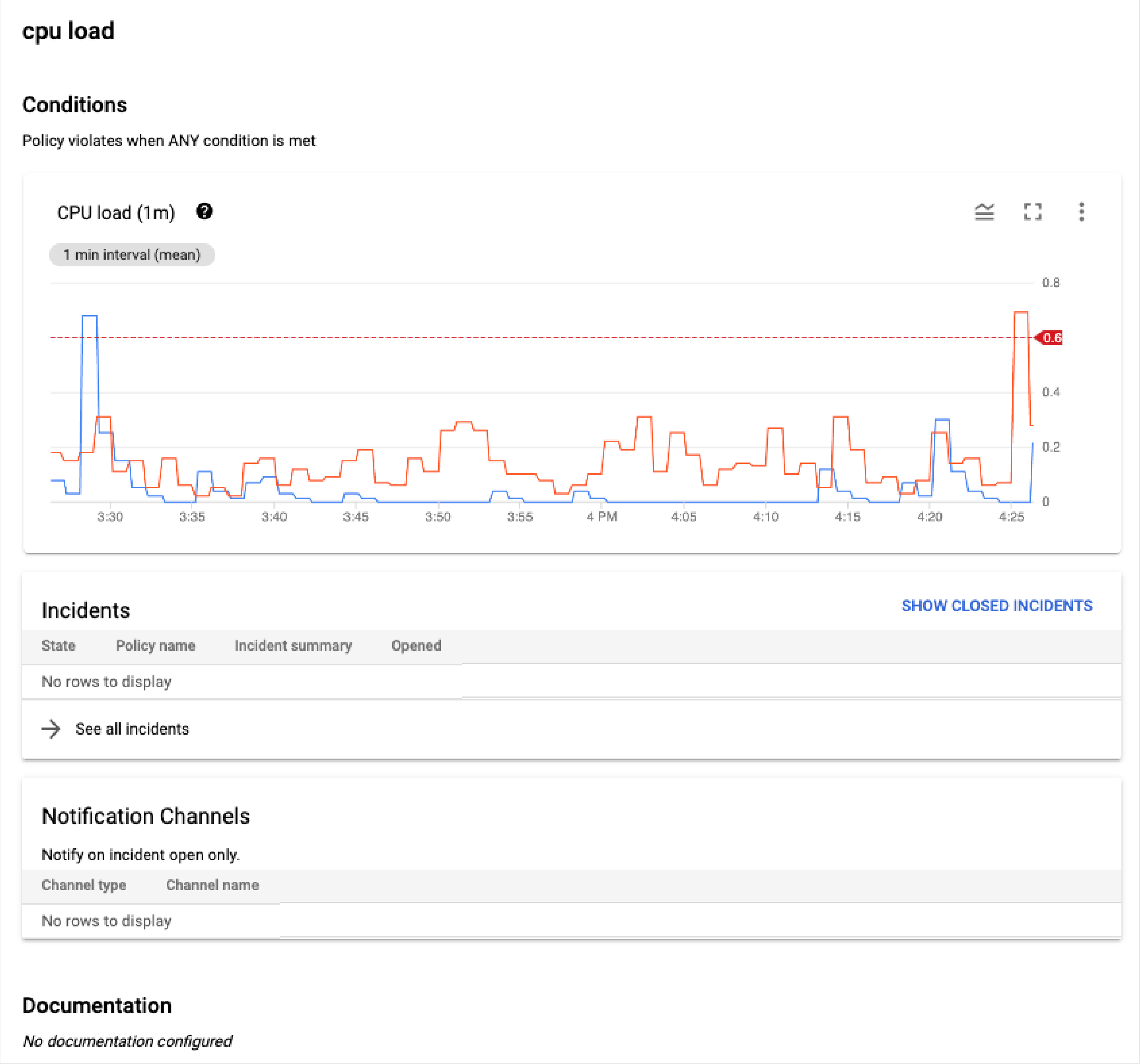
To be notified when the performance of a service doesn't meet criteria you define, create an alerting policy (/monitoring/alerts). For example, you can create an alerting policy that notifies your on-call team when the 90th percentile of the latency of HTTP `200` responses from your service exceeds 100 ms. Similarly, you can be notified when an uptime check (/monitoring/uptime-checks) fails.

Alerting policies let you configure whether a single time series can cause a condition to be met, or whether multiple time series must satisfy the condition before it is met. Alerting policies can be simple or complex. For example:

- Notify me when any uptime check to the domain `example.com` fails for at least three minutes.
- Notify the on-call team when the 90th percentile of HTTP `200` responses exceeds a latency of 100 ms for 3 or more web servers in 2 Google Cloud locations, provided there are fewer than 15 QPS on the server.

- Notify me when the CPU load of any VM instances in my Google Cloud project is greater than a threshold of 0.6.

Cloud Monitoring provides a dashboard that summarizes the status of your alerting policies, and for each policy, provides a dashboard with detailed information. As shown by the following screenshot, the detailed dashboard for an alert displays the data being monitored, the alert threshold, the notification channels, a list of incidents, and any user-defined documentation that is included in a notification:



cpu load

Conditions

Policy violates when ANY condition is met

Incidents

State	Policy name	Incident summary	Opened
No rows to display			

Notification Channels

Notify on incident open only.

Channel type	Channel name
No rows to display	

Documentation

No documentation configured

Conditions are the core component of an alerting policy. A *condition* describes a potential problem with your system that you want Cloud Monitoring to watch for. For example, you might describe conditions like the following:

- Any uptime check to the domain `example.com` fails for at least three minutes.
- The free space of any monitored VM instance is less than 10%.

When the conditions of an alerting policy are met, for example when every uptime check to the domain `example.com` fails for three minutes, Cloud Monitoring opens an incident and issues notifications.

An *incident* is a persistent record that stores information about the resource being monitored. For example, an alerting policy monitoring CPU utilization would store information about the VM whose utilization causes the condition to be met. When the condition stops being met, the incident is automatically closed. You can view all incidents, open and closed, by using the alerting dashboard.

You specify who is to be notified when you configure an alerting policy. Monitoring supports common notifications channels, including email, Cloud Mobile App, and services such as PagerDuty or Slack. For a full list of notification channels, see Notification options (/monitoring/support/notification-options).

For more information about alerting policies, see Introduction to alerting (/monitoring/alerts).

Monitor large systems

This section describes how you can manage resources as a collection and how you can monitor metrics that are stored in multiple Google Cloud projects.

Manage resources as a collection

To manage your resources as a collection instead of individually, create a resource group (/monitoring/groups#creating_groups). A *resource group* is a dynamic collection of resources that satisfy some criteria that you provide. As you add and remove resources, for example by adding Compute Engine VM instances to your Cloud project, the membership in the group automatically changes. The following are examples of resource groups:

- Compute Engine instances whose names start with the string `prod-`.
- Resources with the tag `test-cluster`.
- Amazon EC2 instances in region A or region B.

After you define a resource group (/monitoring/groups#creating_groups), you can monitor the group as if it were a single resource. For example, you can configure an uptime check (/monitoring/uptime-checks) to monitor a resource group. For charts and alerting policies, you can also filter based on the group name.

For more information about this topic, see Using resource groups (/monitoring/groups).

Monitor metrics for multiple Cloud projects

To view and monitor the time-series data for multiple Google Cloud projects and AWS accounts through a single interface, configure a multi-project metrics scope (/monitoring/settings#concept-scope).

By default, Cloud Monitoring pages in the Google Cloud console provide access only to the time series stored in the *scoping project*. The scoping project is the project that you selected with the Google Cloud console project picker. The scoping project stores the alerts (/monitoring/alerts), uptime checks (/monitoring/uptime-checks), dashboards (/monitoring/dashboards), and monitoring groups (/monitoring/groups) that you configure.

The scoping project also hosts a metrics scope. The *metrics scope* defines the projects and accounts whose metrics are visible to the scoping project. You can configure the metrics scope to include time-series data from other Google Cloud projects and from AWS accounts. For information about how to modify a metrics scope, see Modifying your project's Cloud Monitoring configuration (/monitoring/settings#modifying).

Cloud Monitoring data model

This section introduces the Cloud Monitoring data model:

- A *metric* describes something that is measured. Examples of metrics include a VM's CPU utilization and the percentage of a disk that is used.
- A *time series* is a data structure that contains time-stamped measurements of a metric and information about the source and meaning of those measurements.

For example, the following illustrates a time series:

```
"timeSeries": [
  {
    "points": [
      {
        "interval": {
          "startTime": "2020-07-27T20:20:21.597143Z",
          "endTime": "2020-07-27T20:20:21.597143Z"
        },
        "value": {
          "doubleValue": 0.473005
        }
      },
      {
        "interval": {
          "startTime": "2020-07-27T20:19:21.597239Z",
          "endTime": "2020-07-27T20:19:21.597239Z"
        },
        "value": {
          "doubleValue": 0.473025
        }
      },
    ],
    "resource": {
      "type": "gce_instance",
      "labels": {
        "instance_id": "2708613220420473591",
        "zone": "us-east1-b",
        "project_id": "sampleproject"
      }
    },
    "metric": {
      "labels": {
        "device": "sda1",
        "state": "free"
      },
      "type": "agent.googleapis.com/disk/percent_used"
    },
    "metricKind": "GAUGE",
    "valueType": "DOUBLE",

  },
```

Here are some details about what a time series contains:

- The `points` array contains the time-stamped measurements.

  In the previous example, the `points` array contains two values:

  ```
  "points": [
    {
      "interval": {
        "startTime": "2020-07-27T20:20:21.597143Z",
        "endTime": "2020-07-27T20:20:21.597143Z"
      },
      "value": {
        "doubleValue": 0.473005
      }
    },
  ```

```
  {
    "interval": {
      "startTime": "2020-07-27T20:19:21.597239Z",
      "endTime": "2020-07-27T20:19:21.597239Z"
    },
    "value": {
      "doubleValue": 0.473025
    }
  },
],
```

To understand the meaning of a value, you need to refer to the other data included in the time series and to the definitions of that data.

- The `resource` field describes the hardware or software component that is being monitored. In Cloud Monitoring, the hardware or software component is referred to as the *monitored resource*. Examples of monitored resources include Compute Engine instances and App Engine applications. For a complete list of monitored resources, see the Monitored resource list (/monitoring/api/resources).

  In the previous example, the `resource` field is as shown:

  ```
  "resource": {
    "type": "gce_instance",
    "labels": {
      "instance_id": "2708613220420473591",
      "zone": "us-east1-b",
      "project_id": "sampleproject"
    }
  ```

  - The `type` field lists the monitored resource as a `gce_instance` (/monitoring/api/resources#tag_gce_instance), which indicates that these measurements are taken on a Compute Engine VM instance.
  - The `labels` field contains key-value pairs that provide additional information about the monitored resource. For a `gce_instance` type, the labels identify the VM instance that is being monitored.

- The `metric` field describes what is being measured.

  In the previous example, the `metric` field is as shown:

```
"metric": {
  "labels": {
    "device": "sda1",
    "state": "free"
  },
  "type": "agent.googleapis.com/disk/percent_used"
},
```

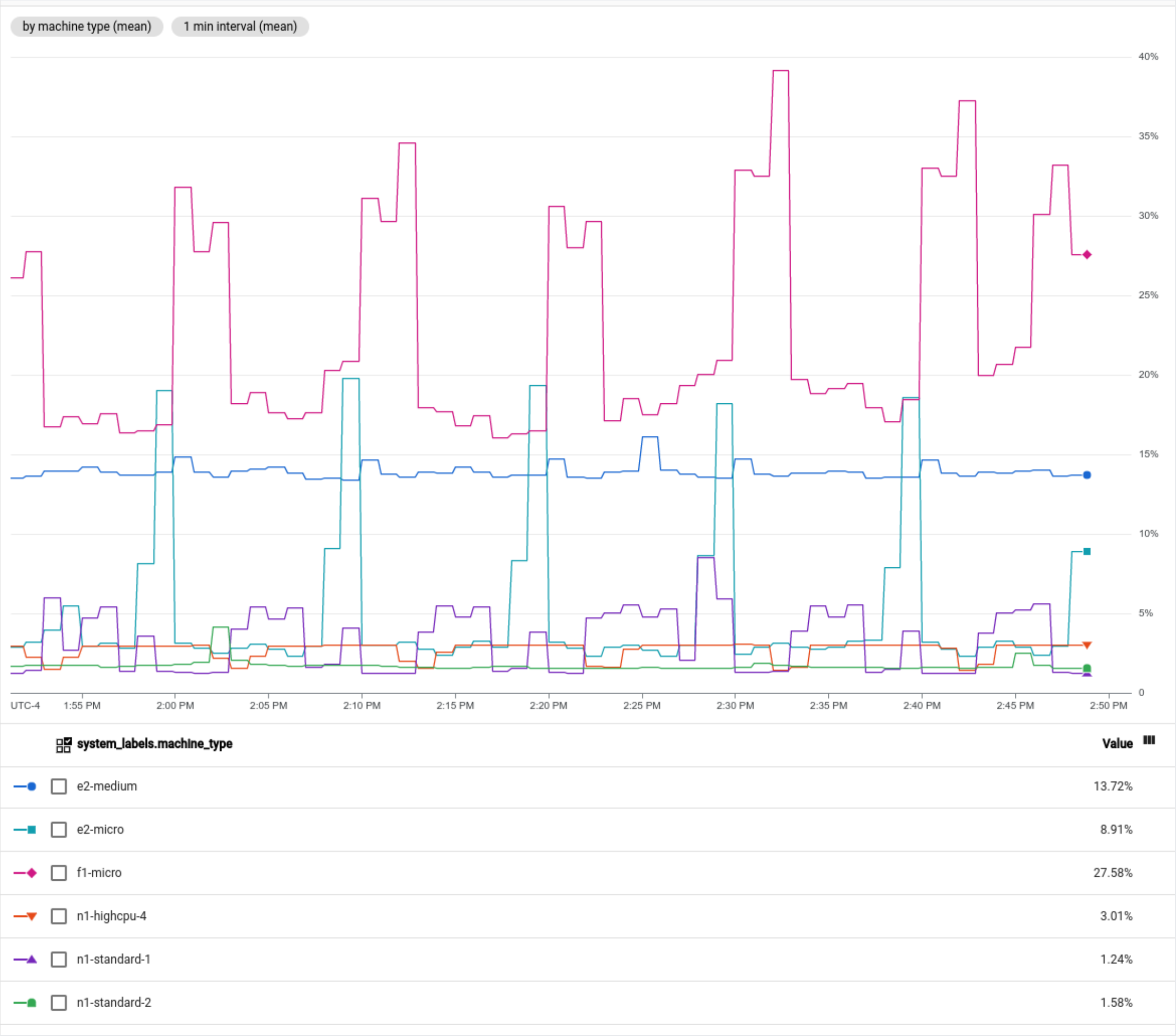
  - For Google services, the `type` field specifies the service and what is being monitored. In this example, the Cloud Monitoring agent is the service, and it's measuring the percentage of the disk that is used. When the `type` field begins with `custom` or `external`, the metric is either a custom metric or one defined by a third party.
  - The `labels` field contains key-value pairs that provide additional information about the measurement. These labels are defined as part of the `MetricDescriptor` (/monitoring/api/ref_v3/rest/v3/projects.metricDescriptors#MetricDescriptor), which is a data structure that defines the attributes of the measured data. The `MetricDescriptor` for the metric `agent.googleapis.com/disk/percent_used` (/monitoring/api/metrics_agent#disk/percent_used) includes the labels `device` and `state`.

- The `metricKind` (/monitoring/api/ref_v3/rest/v3/projects.metricDescriptors#MetricKind) field describes the relationship between adjacent measurements within a time series:
  - `GAUGE` metrics store the value of the thing being measured at a given moment in time—for example, an hourly temperature record.
  - `CUMULATIVE` metrics store the accumulated value of the thing being measured at a given moment in time—for example, an odometer in a vehicle.
  - `DELTA` metrics store the change in the value of the thing being measured over a specified period—for example, a stock summary that shows the stock's gains or losses.
- The `valueType` (/monitoring/api/ref_v3/rest/v3/projects.metricDescriptors#ValueType) field describes the data type for the measurement: `INT64`, `DOUBLE`, `BOOL`, `STRING`, or `DISTRIBUTION` (/monitoring/api/ref_v3/rest/v3/TypedValue#Distribution).

Cloud Monitoring writes one time series for each combination of resource and metric label values. You can use these labels to group and to filter time series. For example, when a

Google Cloud project contains multiple Compute Engine VM instances, the CPU utilization for each VM instance is a unique time series. Here are a few of the ways that you can display this data:

- You can show the CPU utilization of every VM instance.
- You can show the CPU utilization for a specific VM instance by filtering the time series for a single value of the `instance_id` label.
- You can group by the VM instances by the `machine_type` label, and then display the average CPU utilization. The following screenshot illustrates a chart with this configuration:



by machine type (mean) 1 min interval (mean)

	system_labels.machine_type	Value
—●	□ e2-medium	13.72%
—■	□ e2-micro	8.91%
—◆	□ f1-micro	27.58%
—▼	□ n1-highcpu-4	3.01%
—▲	□ n1-standard-1	1.24%
—●	□ n1-standard-2	1.58%

What's next

- To explore Cloud Monitoring, try the Quickstart for monitoring a Compute Engine instance (/monitoring/monitor-compute-engine-virtual-machine).

- For information about how to configure our Google Cloud project to view metrics for multiple Google Cloud projects and AWS accounts, see Configure Cloud Monitoring (/monitoring/settings).
- For information about the Cloud Monitoring API, see APIs and reference (/monitoring/docs/apis).
- For lists of metrics and monitored resources, see Metrics list (/monitoring/api/metrics) and Monitored resource list (/monitoring/api/resources).
- For information about pricing, quotas, and limits, see Resources (/monitoring/docs/resources).

Last updated 2022-10-04 UTC.