

+++ title = "Google Cloud Monitoring" description = "Guide for using Google Cloud Monitoring in Grafana"  
keywords = ["grafana", "stackdriver", "google", "guide", "cloud", "monitoring"] aliases =  
["/docs/grafana/latest/features/datasources/stackdriver", "/docs/grafana/next/datasources/cloudmonitoring/",  
"/docs/grafana/next/features/datasources/cloudmonitoring/"] weight = 350 +++

## Using Google Cloud Monitoring in Grafana

Grafana ships with built-in support for Google Cloud Monitoring. Add it as a data source to build dashboards for your Google Cloud Monitoring metrics. For instructions on how to add a data source, refer to [Add a data source]({{< relref "../add-a-data-source.md" >}}). Only users with the organization admin role can add data sources.

**Note** Before Grafana v7.1, Google Cloud Monitoring was referred to as Google Stackdriver.

### Configure the Google Cloud Monitoring data source

To access Google Cloud Monitoring settings, hover your mouse over the **Configuration** (gear) icon, then click **Data Sources**, and click **Add data source**, then click the Google Cloud Monitoring data source.

Name	Description
Name	The data source name. This is how you refer to the data source in panels and queries.
Default	Default data source means that it is pre-selected for new panels.

For authentication options and configuration details, see the [Google authentication]({{< relref "google-authentication.md" >}}) documentation.

### Google Cloud Monitoring specific data source configuration

The following APIs need to be enabled first:

- [Monitoring API](#)
- [Cloud Resource Manager API](#)

Click on the links above and click the `Enable` button:

{{< figure src="/static/img/docs/v71/cloudmonitoring\_enable\_api.png" max-width="450px" class="docs-image--no-shadow" caption="Enable GCP APIs" >}}

### Using GCP Service Account Key File

The GCP Service Account must have the **Monitoring Viewer** role as shown in the image below:

{{< figure src="/static/img/docs/v71/cloudmonitoring\_service\_account\_choose\_role.png" max-width="600px" class="docs-image--no-shadow" caption="Choose role" >}}

### Using GCE Default Service Account

If Grafana is running on a Google Compute Engine (GCE) virtual machine, the service account in use must have access to the `Cloud Monitoring API` scope.

## Using the Query Editor

The Google Cloud Monitoring query editor allows you to build two types of queries - **Metric** and **Service Level Objective (SLO)**. Both types return time series data.

## Metric Queries

```
{{< figure src="/static/img/docs/google-cloud-monitoring/metric-query-builder-8-0.png" max-width= "400px" class="docs-image--right" >}}
```

The metric query editor allows you to select metrics, group/aggregate by labels and by time, and use filters to specify which time series you want in the results.

To create a metric query, follow these steps:

1. Choose the option **Metrics** in the **Query Type** dropdown
2. Choose a project from the **Project** dropdown
3. Choose a Google Cloud Platform service from the **Service** dropdown
4. Choose a metric from the **Metric** dropdown.
5. Use the plus and minus icons in the filter and group by sections to add/remove filters or group by clauses.  
This step is optional.

Google Cloud Monitoring supports different kinds of metrics like `GAUGE` , `DELTA`, and `CUMULATIVE` . They support different aggregation options, for example, reducers and aligners. The Grafana query editor displays the list of available aggregation methods for a selected metric and sets a default reducer and aligner when you select the metric.

### Filter

To add a filter, click the plus icon and choose a field to filter by and enter a filter value e.g. `instance_name = grafana-1` . You can remove the filter by clicking on the trash icon.

### Simple wildcards

When the operator is set to `=` or `!=` it is possible to add wildcards to the filter value field. E.g `us-*` will capture all values that starts with "us-" and `*central-a` will capture all values that ends with "central-a". `*-central-*` captures all values that has the substring of -central-. Simple wildcards are less expensive than regular expressions.

### Regular expressions

When the operator is set to `=~` or `!~=` it is possible to add regular expressions to the filter value field. E.g `us-central[1-3]-[af]` would match all values that starts with "us-central", is followed by a number in the range of 1 to 3, a dash and then either an "a" or an "f". Leading and trailing slashes are not needed when creating regular expressions.

### Pre-processing

Preprocessing options are displayed in the UI when the selected metric has a metric kind of `delta` or `cumulative` . If the selected metric has a metric kind of `gauge` , no pre-processing option will be displayed.

If you select `Rate` , data points are aligned and converted to a rate per time series. If you select `Delta` , data points are aligned by their delta (difference) per time series.

### Grouping

You can reduce the amount of data returned for a metric by combining different time series. To combine multiple time series, specify a grouping and a function.

### Group by

Group by resource or metric labels to reduce the number of time series and to aggregate the results by a group. For example, group by `instance_name` to view an aggregated metric for a Compute instance.

**METADATA LABELS**

Resource metadata labels contain information that can uniquely identify a resource in Google Cloud. Metadata labels are only returned in the time series response if they're part of the **Group By** segment in the time series request.

There's no API for retrieving metadata labels. As a result, you cannot populate the group by list with the metadata labels that are available for the selected service and metric. However, the **Group By** field list comes with a pre-defined set of common system labels.

User labels cannot be predefined, but you can enter them manually in the **Group By** field. If a metadata label, user label, or system label is included in the **Group By** segment, then you can create filters based on it and expand its value on the **Alias** field.

**Group by function**

Select a grouping function to combine the time series in the group into a single time series.

**Alignment**

The process of alignment consists of collecting all data points received in a fixed length of time, applying a function to combine those data points, and assigning a timestamp to the result.

**Alignment function**

During alignment, all data points are received in a fixed interval. Within each interval (determined by the alignment period) and for each time series, the data is aggregated into a single point. The value of that point is determined by the type of alignment function used. For more information on alignment functions, refer to [alignment metric selector](#).

**Alignment period**

The `Alignment Period` groups a metric by time if an aggregation is chosen. The default is to use the GCP Google Cloud Monitoring default groupings (which allows you to compare graphs in Grafana with graphs in the Google Cloud Monitoring UI). The option is called `cloud monitoring auto` and the defaults are:

- 1m for time ranges < 23 hours
- 5m for time ranges >= 23 hours and < 6 days
- 1h for time ranges >= 6 days

The other automatic option is `grafana auto`. This will automatically set the group by time depending on the time range chosen and the width of the time series panel. For more information about grafana auto, refer to the `[interval variable]({{< relref "../variables/variable-types/add-interval-variable.md" >}})`.

You can also choose fixed time intervals to group by, like `1h` or `1d`.

**Alias patterns**

The Alias By field allows you to control the format of the legend keys. The default is to show the metric name and labels. This can be long and hard to read. Using the following patterns in the alias field, you can format the legend key the way you want it.

**Metric type patterns**

Alias Pattern	Description	Example Result
<code>{{metric.type}}</code>	returns the full Metric Type	<code>compute.googleapis.com/instance/cpu/utilization</code>

<code>{{metric.name}}</code>	returns the metric name part	instance/cpu/utilization
<code>{{metric.service}}</code>	returns the service part	compute

### Label patterns

In the Group By dropdown, you can see a list of metric and resource labels for a metric. These can be included in the legend key using alias patterns.

Alias Pattern Format	Description	Alias Pattern Example	Example Result
<code>{{metric.label.xxx}}</code>	returns the metric label value	<code>{{metric.label.instance_name}}</code>	grafana-1-prod
<code>{{resource.label.xxx}}</code>	returns the resource label value	<code>{{resource.label.zone}}</code>	us-east1-b
<code>{{metadata.system_labels.xxx}}</code>	returns the meta data system label value	<code>{{metadata.system_labels.name}}</code>	grafana
<code>{{metadata.user_labels.xxx}}</code>	returns the meta data user label value	<code>{{metadata.user_labels.tag}}</code>	production

Example Alias By: `{{metric.type}} - {{metric.label.instance_name}}`

Example Result: `compute.googleapis.com/instance/cpu/usage_time - server1-prod`

It is also possible to resolve the name of the Monitored Resource Type.

Alias Pattern Format	Description	Example Result
<code>{{resource.type}}</code>	returns the name of the monitored resource type	<code>gce_instance</code>

Example Alias By: `{{resource.type}} - {{metric.type}}`

Example Result: `gce_instance - compute.googleapis.com/instance/cpu/usage_time`

### Deep linking from Grafana panels to the Metrics Explorer in Google Cloud Console

**Note:** Available in Grafana v7.1 and later versions.

{{< figure src="/static/img/docs/v71/cloudmonitoring\_deep\_linking.png" max-width="500px" class="docs-image--right" caption="Google Cloud Monitoring deep linking" >}}

Click on a time series in the panel to see a context menu with a link to View in Metrics Explorer in Google Cloud Console. Clicking that link opens the Metrics Explorer in the Google Cloud Console and runs the query from the Grafana panel there. The link navigates the user first to the Google Account Chooser and after successfully selecting an account, the user is redirected to the Metrics Explorer. The provided link is valid for any account, but it only displays the query if your account has access to the GCP project specified in the query.

### Automatic unit detection

Grafana issues one query to the Cloud Monitoring API per query editor row, and each API response includes a unit. Grafana will attempt to convert the returned unit into a unit that is understood by the Grafana time series panel. If the conversion was successful, then the unit will be displayed on the Y-axis on the panel. If the query editor rows returned different units, then the unit from the last query editor row is used in the time series panel.

### SLO (Service Level Objective) queries

**Note:** Available in Grafana v7.0 and later versions.

{{< figure src="/static/img/docs/google-cloud-monitoring/slo-query-builder-8-0.png" max-width= "400px" class="docs-image--right" >}}

The SLO query builder in the Google Cloud Monitoring data source allows you to display SLO data in time series format. To get an understanding of the basic concepts in service monitoring, please refer to Google Cloud Monitoring's [official docs](#).

#### How to create an SLO query

To create an SLO query, follow these steps:

1. Choose the option **Service Level Objectives (SLO)** in the **Query Type** dropdown.
2. Choose a project from the **Project** dropdown.
3. Choose an [SLO service](#) from the **Service** dropdown.
4. Choose an [SLO](#) from the **SLO** dropdown.
5. Choose a [time series selector](#) from the **Selector** dropdown.

The friendly names for the time series selectors are shown in Grafana. Here is the mapping from the friendly name to the system name that is used in the Service Monitoring documentation:

Selector dropdown value	Corresponding time series selector used
SLI Value	select_slo_health
SLO Compliance	select_slo_compliance
SLO Error Budget Remaining	select_slo_budget_fraction

#### Alias patterns for SLO queries

The Alias By field allows you to control the format of the legend keys for SLO queries too.

Alias Pattern	Description	Example Result
{{project}}	returns the GCP project name	myProject
{{service}}	returns the service name	myService
{{slo}}	returns the SLO	latency-slo
{{selector}}	returns the selector	select_slo_health

#### Alignment period/group by time for SLO queries

SLO queries use the same [alignment period functionality as metric queries]({{< relref "#metric-queries" >}}).

### MQL (Monitoring Query Language) queries

**Note:** Available in Grafana v7.4 and later versions.

The MQL query builder in the Google Cloud Monitoring data source allows you to display MQL results in time series format. To get an understanding of the basic concepts in MQL, refer to [Introduction to Monitoring Query Language](#).

### Create an MQL query

To create an MQL query, follow these steps:

1. In the **Query Type** list, select **Metrics**.
2. Click **<> Edit MQL** right next to the **Query Type** field. This will toggle the metric query builder mode so that raw MQL queries can be used.
3. Choose a project from the **Project** list.
4. Add the [MQL](#) query of your choice in the text area.

### Alias patterns for MQL queries

MQL queries use the same alias patterns as [metric queries]({{< relref "#metric-queries" >}}).

`{{metric.service}}` is not supported. `{{metric.type}}` and `{{metric.name}}` show the time series key in the response.

## Templating

Instead of hard-coding things like server, application and sensor name in your metric queries you can use variables in their place. Variables are shown as dropdown select boxes at the top of the dashboard. These dropdowns make it easy to change the data being displayed in your dashboard.

Check out the [Templating]({{< relref "../variables/\_index.md" >}}) documentation for an introduction to the templating feature and the different types of template variables.

### Query Variable

Variable of the type *Query* allows you to query Google Cloud Monitoring for various types of data. The Google Cloud Monitoring data source plugin provides the following `Query Types`.

Name	Description
Metric Types	Returns a list of metric type names that are available for the specified service.
Labels Keys	Returns a list of keys for <code>metric label</code> and <code>resource label</code> in the specified metric.
Labels Values	Returns a list of values for the label in the specified metric.
Resource Types	Returns a list of resource types for the specified metric.
Aggregations	Returns a list of aggregations (cross series reducers) for the specified metric.
Aligners	Returns a list of aligners (per series aligners) for the specified metric.
Alignment periods	Returns a list of all alignment periods that are available in Google Cloud Monitoring query editor in Grafana
Selectors	Returns a list of selectors that can be used in SLO (Service Level Objectives) queries
SLO Services	Returns a list of Service Monitoring services that can be used in SLO queries

Service Level Objectives (SLO)	Returns a list of SLO's for the specified SLO service
--------------------------------	---

## Using variables in queries

Refer to the [\[variable syntax documentation\]](#) for more information.

## Annotations

Annotations allow you to overlay rich event information on top of graphs. You add annotation queries via the Dashboard menu / Annotations view. Annotation rendering is expensive so it is important to limit the number of rows returned. There is no support for showing Google Cloud Monitoring annotations and events yet but it works well with [custom metrics](#) in Google Cloud Monitoring.

With the query editor for annotations, you can select a metric and filters. The `Title` and `Text` fields support templating and can use data returned from the query. For example, the Title field could have the following text:

```
{{metric.type}} has value: {{metric.value}}
```

Example Result: `monitoring.googleapis.com/uptime_check/http_status` has this value: 502

## Patterns for the Annotation Query Editor

Alias Pattern Format	Description	Alias Pattern Example	Example Result
<code>{{metric.value}}</code>	value of the metric/point	<code>{{metric.value}}</code>	555
<code>{{metric.type}}</code>	returns the full Metric Type	<code>{{metric.type}}</code>	<code>compute.googleapis.com/instance/cpu/utilization</code>
<code>{{metric.name}}</code>	returns the metric name part	<code>{{metric.name}}</code>	<code>instance/cpu/utilization</code>
<code>{{metric.service}}</code>	returns the service part	<code>{{metric.service}}</code>	<code>compute</code>
<code>{{metric.label.xxx}}</code>	returns the metric label value	<code>{{metric.label.instance_name}}</code>	<code>grafana-1-prod</code>
<code>{{resource.label.xxx}}</code>	returns the resource label value	<code>{{resource.label.zone}}</code>	<code>us-east1-b</code>

## Configure the data source with provisioning

You can provision CloudWatch data source by modifying Grafana's configuration files. To learn more about provisioning and all the settings you can set, see the [\[provisioning documentation\]](#).

```
"../administration/provisioning/#data-sources" >}})
```

Here is a provisioning example using the JWT (Service Account key file) authentication type.

```
apiVersion: 1

datasources:
- name: Google Cloud Monitoring
  type: stackdriver
  access: proxy
  jsonData:
    tokenUri: https://oauth2.googleapis.com/token
    clientEmail: stackdriver@myproject.iam.gserviceaccount.com
    authenticationType: jwt
    defaultProject: my-project-name
  secureJsonData:
    privateKey: |
      -----BEGIN PRIVATE KEY-----
      POSEvQIBADANBgkqhkiG9w0BAQEFAASCBCwggSjAgEAAoIBAQCblu1Srw8ICYHS
      ...
      yA+23427282348234=
      -----END PRIVATE KEY-----
```

Here is a provisioning example using GCE Default Service Account authentication.

```
apiVersion: 1

datasources:
- name: Google Cloud Monitoring
  type: stackdriver
  access: proxy
  jsonData:
    authenticationType: gce
```