+++ title = “Alerting on numeric data” aliases = [“/docs/grafana/latest/alerting/metrics/”] weight = 116 +++

Alerting on numeric data

This topic describes how Grafana managed alerts are evaluated by the backend engine as well as how Grafana handles alerting on numeric rather than time series data.

- Alert evaluation
- Alerting on numeric data

Alert evaluation

Grafana managed alerts query the following backend data sources that have alerting enabled:

- built-in data sources or those developed and maintained by Grafana: `Graphite`, `Prometheus`, `Loki`, `InfluxDB`, `Elasticsearch`, `Google Cloud Monitoring`, `Cloudwatch`, `Azure Monitor`, `MySQL`, `PostgreSQL`, `MSSQL`, `OpenTSDB`, `Oracle`, and `Azure Monitor`
- community developed backend data sources with alerting enabled (`backend` and `alerting` properties are set in the [plugin.json]({{< relref “../../../developers/plugins/metadata.md” >}}))

Metrics from the alerting engine

The alerting engine publishes some internal metrics about itself. You can read more about how Grafana publishes [internal metrics]({{< relref “../../../administration/view-server/internal-metrics.md” >}}). See also, [View alert rules and their current state]({{< relref “../alerting-rules/rule-list.md” >}}).

Metric Name	Type	Description
`grafana_alerting_alerts`	gauge	How many alerts by state
`grafana_alerting_request_duration_seconds`	histogram	Histogram of requests to the Alerting API
`grafana_alerting_active_configurations`	gauge	The number of active, non default Alertmanager configurations for grafana managed alerts
`grafana_alerting_rule_evaluations_total`	counter	The total number of rule evaluations
`grafana_alerting_rule_evaluation_failures_total`	counter	The total number of rule evaluation failures
`grafana_alerting_rule_evaluation_duration`	summary	The duration for a rule to execute
`grafana_alerting_rule_group_rules`	gauge	The number of rules

Alerting on numeric data

Among certain data sources numeric data that is not time series can be directly alerted on, or passed into Server Side Expressions (SSE). This allows for more processing and resulting efficiency within the data source, and it can also simplify alert rules. When alerting on numeric data instead of time series data, there is no need to reduce each labeled time series into a single number. Instead labeled numbers are returned to Grafana instead.

Tabular Data

This feature is supported with backend data sources that query tabular data:

- SQL data sources such as MySQL, Postgres, MSSQL, and Oracle.
- The Azure Kusto based services: Azure Monitor (Logs), Azure Monitor (Azure Resource Graph), and Azure Data Explorer.

A query with Grafana managed alerts or SSE is considered numeric with these data sources, if:

- The "Format AS" option is set to "Table" in the data source query.
- The table response returned to Grafana from the query includes only one numeric (e.g. int, double, float) column, and optionally additional string columns.

If there are string columns then those columns become labels. The name of column becomes the label name, and the value for each row becomes the value of the corresponding label. If multiple rows are returned, then each row should be uniquely identified their labels.

Example

For a MySQL table called "DiskSpace":

Time	Host	Disk	PercentFree
2021-June-7	web1	/etc	3
2021-June-7	web2	/var	4
2021-June-7	web3	/var	8
...	...	...	...

You can query the data filtering on time, but without returning the time series to Grafana. For example, an alert that would trigger per Host, Disk when there is less than 5% free space:

```
SELECT Host, Disk, CASE WHEN PercentFree < 5.0 THEN PercentFree ELSE 0 END FROM (
  SELECT
      Host,
      Disk,
```

```
    Avg(PercentFree)
FROM DiskSpace
Group By
  Host,
  Disk
Where __timeFilter(Time)
```

This query returns the following Table response to Grafana:

Host	Disk	PercentFree
web1	/etc	3
web2	/var	4
web3	/var	0

When this query is used as the **condition** in an alert rule, then the non-zero will be alerting. As a result, three alert instances are produced:

Labels	Status
{Host=web1,disk=/etc}	Alerting
{Host=web2,disk=/var}	Alerting
{Host=web3,disk=/var}	Normal