

+++ title = “Flux support in Grafana” description = “Guide for Flux in Grafana”
weight = 200 +++

Flux query language in Grafana

Grafana supports Flux running on InfluxDB 1.8+. See 1.8 compatibility for more information and connection details.

Name	Description
Name	The data source name. This is how you refer to the data source in panels and queries. We recommend something like InfluxDB-Flux .
Default	Default data source means that it will be pre-selected for new panels.
URL	The HTTP protocol, IP address and port of your InfluxDB API. InfluxDB 2.0 API port is by default 8086.
Organization	Influx organization that will be used for Flux queries. This is also used to for the v.organization query macro.
Token	The authentication token used for Flux queries. With Influx 2.0, use the influx authentication token to function. For influx 1.8, the token is username:password .
Default bucket	(Optional) The Influx bucket that will be used for the v.defaultBucket macro in Flux queries.
Min time interval	(Optional) Refer to Min time interval({{< relref “#min-time-interval” >}}).
Max series	(Optional) Limits the number of series/tables that Grafana processes. Lower this number to prevent abuse, and increase it if you have lots of small time series and not all are shown. Defaults to 1000.

Min time interval

A lower limit for the auto group by time interval. Recommended to be set to write frequency, for example **1m** if your data is written every minute. This option can also be overridden/configured in a dashboard panel under data source options. It’s important to note that this value **needs** to be formatted as a number followed by a valid time identifier, e.g. **1m** (1 minute) or **30s** (30 seconds). The following time identifiers are supported:

Identifier	Description
y	year
M	month
w	week
d	day
h	hour

Identifier	Description
m	minute
s	second
ms	millisecond

You can use the Flux query and scripting language. Grafana's Flux query editor is a text editor for raw Flux queries with Macro support.

Supported macros

The macros support copying and pasting from Chronograf.

Macro exam- ple	Description
<code>v.timeRangeStart</code>	Replaced by the start of the currently active time selection. For example, <i>2020-06-11T13:31:00Z</i>
<code>v.timeRangeStop</code>	Replaced by the end of the currently active time selection. For example, <i>2020-06-11T14:31:00Z</i>
<code>v.windowPeriod</code>	Replaced with an interval string compatible with Flux that corresponds to Grafana's calculated interval based on the time range of the active time selection. For example, <i>5s</i>
<code>v.defaultBucket</code>	Replaced with the data source configuration's "Default Bucket" setting
<code>v.organization</code>	Replaced with the data source configuration's "Organization" setting

For example, the following query will be interpolated as the query that follows it, with interval and time period values changing according to active time selection):

Grafana Flux query:

```
from(bucket: v.defaultBucket)
  |> range(start: v.timeRangeStart, stop: v.timeRangeStop)
  |> filter(fn: (r) => r["_measurement"] == "cpu" or r["_measurement"] == "swap")
  |> filter(fn: (r) => r["_field"] == "usage_system" or r["_field"] == "free")
  |> aggregateWindow(every: v.windowPeriod, fn: mean)
  |> yield(name: "mean")
```

Interpolated query send to Influx:

```
from(bucket: "grafana")
  |> range(start: 2020-06-11T13:59:07Z, stop: 2020-06-11T14:59:07Z)
  |> filter(fn: (r) => r["_measurement"] == "cpu" or r["_measurement"] == "swap")
  |> filter(fn: (r) => r["_field"] == "usage_system" or r["_field"] == "free")
```

```
|> aggregateWindow(every: 2s, fn: mean)
|> yield(name: "mean")
```

You can view the interpolated version of a query with the query inspector. For more information, refer to [\[Navigate the Query Inspector\]](#)(({< relref “../panels/working-with-panels/navigate-inspector-panel.md” >}}).