

```
+++ title = "OpenTSDB" description = "Guide for using OpenTSDB
in Grafana" keywords = ["grafana", "opentsdb", "guide"] aliases =
["/docs/grafana/latest/features/opentsdb", "/docs/grafana/latest/features/datasources/opentsdb/"]
weight = 1100 +++
```

Using OpenTSDB in Grafana

Grafana ships with advanced support for OpenTSDB. This topic explains options, variables, querying, and other options specific to the OpenTSDB data source. Refer to [Add a data source]({{< relref "add-a-data-source.md" >}}) for instructions on how to add a data source to Grafana. Only users with the organization admin role can add data sources.

OpenTSDB settings

To access OpenTSDB settings, hover your mouse over the **Configuration** (gear) icon, then click **Data Sources**, and then click the OpenTSDB 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 will be pre-selected for new panels.
URL	The HTTP protocol, IP, and port of your OpenTSDB server (default port is usually 4242)
Allowed cookies	List the names of cookies to forward to the data source.
Version	Version = opentsdb version, either <=2.1 or 2.2
Resolution	Metrics from opentsdb may have datapoints with either second or millisecond resolution.
Lookup limit	Default is 1000.

Query editor

Open a graph in edit mode by click the title. Query editor will differ if the data source has version <=2.1 or = 2.2. In the former version, only tags can be used to query OpenTSDB. But in the latter version, filters as well as tags can be used to query opentsdb. Fill Policy is also introduced in OpenTSDB 2.2.

Note: While using OpenTSDB 2.2 data source, make sure you use either Filters or Tags as they are mutually exclusive. If used together, might give you weird results.

Auto complete suggestions

As soon as you start typing metric names, tag names and tag values , you should see highlighted auto complete suggestions for them. The autocomplete only works if the OpenTSDB suggest API is enabled.

Templating queries

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

Grafana’s OpenTSDB data source supports template variable queries. This means you can create template variables that fetch the values from OpenTSDB. For example, metric names, tag names, or tag values.

When using OpenTSDB with a template variable of **query** type you can use following syntax for lookup.

Query	Description
<code>metrics(prefix)</code>	Returns metric names with specific prefix (can be empty)
<code>tag_names(cpu)</code>	Returns tag names (i.e. keys) for a specific cpu metric
<code>tag_values(cpu, hostname)</code>	Returns tag values for metric cpu and tag key hostname
<code>suggest_tagk(prefix)</code>	Returns tag names (i.e. keys) for all metrics with specific prefix (can be empty)
<code>suggest_tagv(prefix)</code>	Returns tag values for all metrics with specific prefix (can be empty)

If you do not see template variables being populated in **Preview of values** section, you need to enable `tsd.core.meta.enable_realtime_ts` in the OpenTSDB server settings. Also, to populate metadata of the existing time series data in OpenTSDB, you need to run `tsdb uid metasync` on the OpenTSDB server.

Nested templating

One template variable can be used to filter tag values for another template variable. First parameter is the metric name, second parameter is the tag key for

which you need to find tag values, and after that all other dependent template variables. Some examples are mentioned below to make nested template queries work successfully.

Query	Description
<code>tag_values(cpu, hostname, env=\$env)</code>	Return tag values for cpu metric, selected env tag value and tag key hostname
<code>tag_values(cpu, hostname, env=\$env, region=\$region)</code>	Return tag values for cpu metric, selected env tag value, selected region tag value and tag key hostname

For details on OpenTSDB metric queries, check out the official OpenTSDB documentation

Configure the data source with provisioning

It's now possible to configure data sources using config files with Grafana's provisioning system. You can read more about how it works and all the settings you can set for data sources on the [\[provisioning docs page\]](#) ([{{< relref "../administration/provisioning/#datasources" >}}](#))

Here are some provisioning examples for this data source.

```
apiVersion: 1
```

```
datasources:
```

```
- name: OpenTsdb
  type: opentsdb
  access: proxy
  url: http://localhost:4242
  jsonData:
    tsdbResolution: 1
    tsdbVersion: 1
```