+++ title = “Enable diagnostics” weight = 200 +++

Enable diagnostics

You can set up the `grafana-server` process to enable certain diagnostics when it starts. This can be helpful when investigating certain performance problems. It's *not* recommended to have these enabled by default.

Turn on profiling

The `grafana-server` can be started with the arguments `-profile` to enable profiling, `-profile-addr` to override the default HTTP address (`localhost`), and `-profile-port` to override the default HTTP port (`6060`) where the `pprof` debugging endpoints are available. For example:

```
./grafana-server -profile -profile-addr=0.0.0.0 -profile-port=8080
```

Note that `pprof` debugging endpoints are served on a different port than the Grafana HTTP server.

You can configure or override profiling settings using environment variables:

```
export GF_DIAGNOSTICS_PROFILING_ENABLED=true
export GF_DIAGNOSTICS_PROFILING_ADDR=0.0.0.0
export GF_DIAGNOSTICS_PROFILING_PORT=8080
```

Refer to Go command pprof for more information about how to collect and analyze profiling data.

Use tracing

The `grafana-server` can be started with the arguments `-tracing` to enable tracing and `-tracing-file` to override the default trace file (`trace.out`) where trace result is written to. For example:

```
./grafana-server -tracing -tracing-file=/tmp/trace.out
```

You can configure or override profiling settings using environment variables:

```
export GF_DIAGNOSTICS_TRACING_ENABLED=true
export GF_DIAGNOSTICS_TRACING_FILE=/tmp/trace.out
```

View the trace in a web browser (Go required to be installed):

```
go tool trace <trace file>
2019/11/24 22:20:42 Parsing trace...
2019/11/24 22:20:42 Splitting trace...
2019/11/24 22:20:42 Opening browser. Trace viewer is listening on http://127.0.0.1:39735
```

For more information about how to analyze trace files, refer to Go command trace.