+++ title = “Internal Grafana metrics” description = “Internal metrics exposed by Grafana” keywords = [“grafana”, “metrics”, “internal metrics”] aliases = [“/docs/grafana/latest/admin/metrics/”] weight = 200 +++

Internal Grafana metrics

Grafana collects some metrics about itself internally. Grafana supports pushing metrics to Graphite or exposing them to be scraped by Prometheus.

For more information about configuration options related to Grafana metrics, refer to [metrics]({{< relref “../../administration/configuration/#metrics” >}}) and [metrics.graphite]({{< relref “../../administration/configuration/#metrics-graphite” >}}) in [Configuration]({{< relref “../../administration/configuration.md” >}}).

Available metrics

When enabled, Grafana exposes a number of metrics, including:

- Active Grafana instances
- Number of dashboards, users, and playlists
- HTTP status codes
- Requests by routing group
- Grafana active alerts
- Grafana performance

Pull metrics from Grafana into Prometheus

These instructions assume you have already added Prometheus as a data source in Grafana.

1. Enable Prometheus to scrape metrics from Grafana. In your configuration file (`grafana.ini` or `custom.ini` depending on your operating system) remove the semicolon to enable the following configuration options:

   ```
   # Metrics available at HTTP URL /metrics and /metrics/plugins/:pluginId
   [metrics]
   # Disable / Enable internal metrics
   enabled           = true

   # Disable total stats (stat_totals_*) metrics to be generated
   disable_total_stats = false
   ```

2. (optional) If you want to require authorization to view the metrics endpoints, then uncomment and set the following options:

   ```
   basic_auth_username =
   basic_auth_password =
   ```

3. Restart Grafana. Grafana now exposes metrics at http://localhost:3000/metrics.
4. Add the job to your prometheus.yml file. Example:

```
- job_name: 'grafana_metrics'

  scrape_interval: 15s
  scrape_timeout: 5s

  static_configs:
    - targets: ['localhost:3000']
```

5. Restart Prometheus. Your new job should appear on the Targets tab.
6. In Grafana, hover your mouse over the **Configuration** (gear) icon on the left sidebar and then click **Data Sources**.
7. Select the **Prometheus** data source.
8. On the Dashboards tab, **Import** the Grafana metrics dashboard. All scraped Grafana metrics are available in the dashboard.

View Grafana metrics in Graphite

These instructions assume you have already added Graphite as a data source in Grafana.

1. Enable sending metrics to Graphite. In your configuration file (`grafana.ini` or `custom.ini` depending on your operating system) remove the semicolon to enable the following configuration options:

```
# Metrics available at HTTP API Url /metrics
[metrics]
# Disable / Enable internal metrics
enabled           = true

# Disable total stats (stat_totals_*) metrics to be generated
disable_total_stats = false
```

2. Enable [metrics.graphite] options:

```
# Send internal metrics to Graphite
[metrics.graphite]
# Enable by setting the address setting (ex localhost:2003)
address = <hostname or ip>:<port#>
prefix = prod.grafana.%(instance_name)s.
```

3. Restart Grafana. Grafana now exposes metrics at http://localhost:3000/metrics and sends them to the Graphite location you specified.

Pull metrics from Grafana backend plugin into Prometheus

Any installed [backend plugin]({{< relref "../../developers/plugins/backend/_index.md" >}}) exposes a metrics endpoint through Grafana that you can configure Prometheus to scrape.

These instructions assume you have already added Prometheus as a data source in Grafana.

1. Enable Prometheus to scrape backend plugin metrics from Grafana. In your configuration file (`grafana.ini` or `custom.ini` depending on your operating system) remove the semicolon to enable the following configuration options:

   ```
   # Metrics available at HTTP URL /metrics and /metrics/plugins/:pluginId
   [metrics]
   # Disable / Enable internal metrics
   enabled           = true

   # Disable total stats (stat_totals_*) metrics to be generated
   disable_total_stats = false
   ```

2. (optional) If you want to require authorization to view the metrics endpoints, then uncomment and set the following options:

   ```
   basic_auth_username =
   basic_auth_password =
   ```

3. Restart Grafana. Grafana now exposes metrics at `http://localhost:3000/metrics/plugins/<plugin id>`, e.g. http://localhost:3000/metrics/plugins/grafana-github-datasource if you have the Grafana GitHub datasource installed.

4. Add the job to your prometheus.yml file. Example:

   ```
   - job_name: 'grafana_github_datasource'

     scrape_interval: 15s
     scrape_timeout: 5s
     metrics_path: /metrics/plugins/grafana-test-datasource

     static_configs:
       - targets: ['localhost:3000']
   ```

5. Restart Prometheus. Your new job should appear on the Targets tab.

6. In Grafana, hover your mouse over the **Configuration** (gear) icon on the left sidebar and then click **Data Sources**.

7. Select the **Prometheus** data source.

8. Import a Golang application metrics dashboard - for example Go Processes.