

# Grafana

Used to build dashboards!

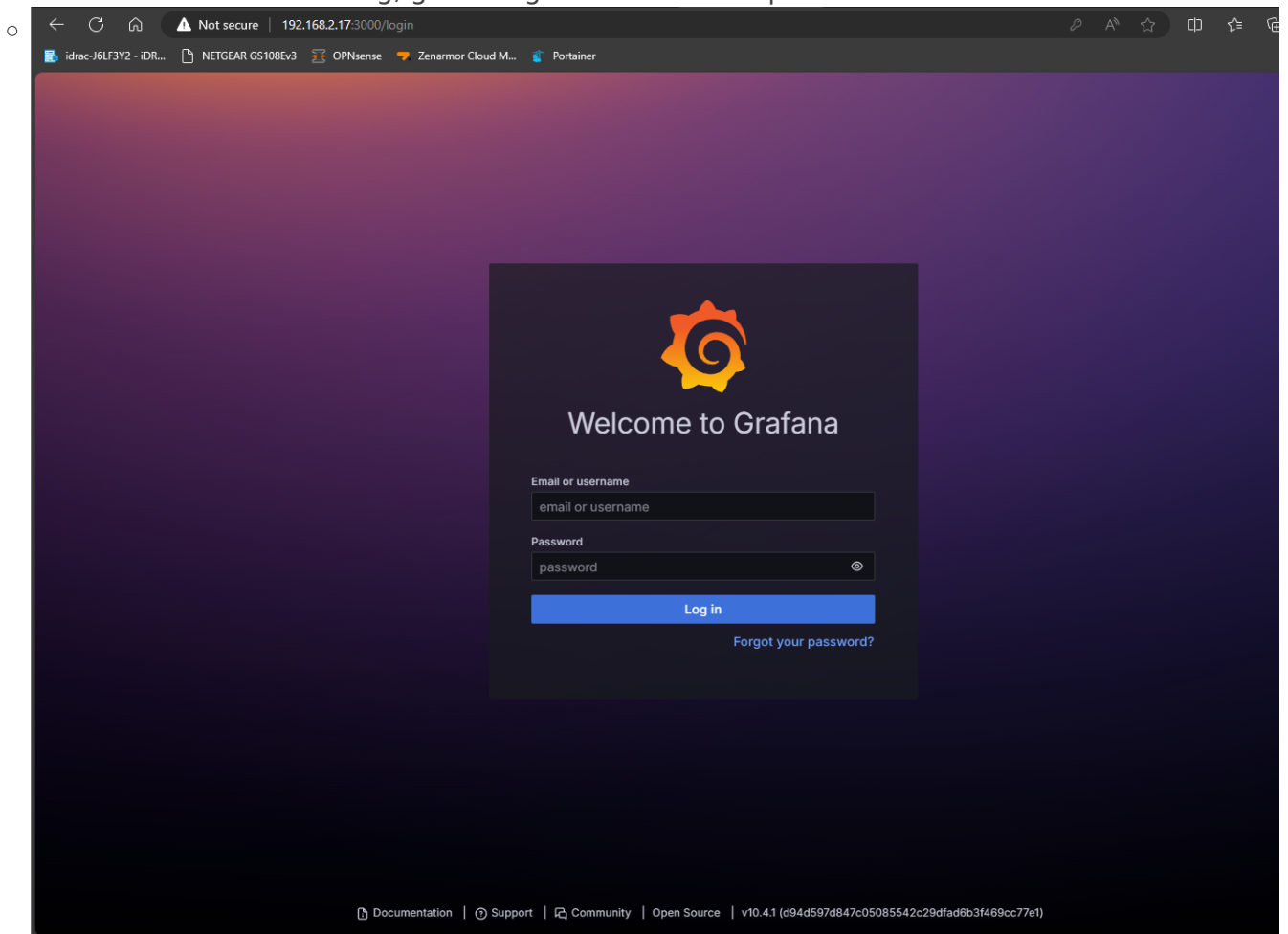
- [GUI Preview](#)

# GUI Preview

Download and run the Grafana container. You can get more details about how to do so @ [DockerHub - Grafana](#). Or, you can just run the following command:

- ```
docker run -d --name=grafana -p 3000:3000 grafana/grafana
```

  - 3000:3000 specifies it'll be accessible from port 3000 of the docker engine, which in my case is 192.168.2.17.
- Once the container is running, go through the initial set up wizard via the Web GUI:



With your Grafana container running, it can be accessed via port 3000 of your localhost. The default login username and password is **admin**. Once logged in, you should see a home page like the one below:

[Remove this page](#)

## Basic

The steps below will guide you to quickly finish setting up your Grafana installation.

### TUTORIAL DATA SOURCE AND DASHBOARDS

#### Grafana fundamentals

Set up and understand Grafana if you have no prior experience. This tutorial guides you through the entire process and covers the "Data source" and "Dashboards" steps to the right.

### DATA SOURCES

#### Add your first data source

[Learn how in the docs](#)

### DASHBOARDS

#### Create your first dashboard

[Learn how in the docs](#)

## Dashboards

### Starred dashboards

### Recently viewed dashboards

## Latest from the blog



Mar 22

### How shipping/third-party logistics companies reduce MTTR and increase uptime with the Grafana LGTM Stack

These days, everything can be tracked: transportation, deliveries, food orders . . . For consumers, knowing the location of a package or courier is a bonus, but for companies in the business of shipping, delivering, and third-party logistics, it's a necessity. And so is having the right observability system to ensure everything gets where it needs to go. After all, errors, downtime, or anything that causes delays will end up delivering unhappy customers and lost revenue.

Mar 21

### OpenTelemetry distributed tracing with eBPF: What's new in Grafana Beyla 1.3

Grafana Beyla, an open source eBPF auto-instrumentation tool, has been able to produce OpenTelemetry trace spans since