+++ title = "Run Grafana Docker image" description = "Guide for running Grafana using Docker" keywords = ["grafana", "configuration", "documentation", "docker"] weight = 600

+++

Run Grafana Docker image

You can install and run Grafana using the official Docker images. Our docker images come in two editions:

Grafana Enterprise: `grafana/grafana-enterprise`

Grafana Open Source: `grafana/grafana-oss`

Each edition is available in two variants: Alpine and Ubuntu. See below.

This topic also contains important information about migrating from earlier Docker image versions.

> ***Note:*** *You can use Grafana Cloud to avoid the overhead of installing, maintaining, and scaling your observability stack. The free forever plan includes Grafana, 10K Prometheus series, 50 GB logs, and more.Create a free account to get started.*

Alpine image (recommended)

Grafana Enterprise edition: `grafana/grafana-enterprise:<version>`

Grafana Open Source edition: `grafana/grafana-oss:<version>`

The default images are based on the popular Alpine Linux project, available in the Alpine official image. Alpine Linux is much smaller than most distribution base images, and thus leads to slimmer and more secure images.

The Alpine variant is highly recommended when security and final image size being as small as possible is desired. The main caveat to note is that it uses musl libc instead of glibc and friends, so certain software might run into issues depending on the depth of their libc requirements. However, most software don't have an issue with this, so this variant is usually a very safe choice.

> ***Note:*** *Grafana docker images were based on Ubuntu prior to version 6.4.0.*

Ubuntu image

Grafana Enterprise edition: `grafana/grafana-enterprise:<version>-ubuntu`

Grafana Open Source edition: `grafana/grafana-oss:<version>-ubuntu`

These images are based on Ubuntu, available in the Ubuntu official image. It is an alternative image for those who prefer an Ubuntu based image and/or are dependent on certain tooling not available for Alpine.

Run Grafana

You can run the latest Grafana version, run a specific version, or run an unstable version based on the main branch of the grafana/grafana GitHub repository.

Run the latest stable version of Grafana

> ***Note:*** *If you are on a Linux system, you might need to add* `sudo` *before the command or add your user to the* `docker` *group.*

```
docker run -d -p 3000:3000 grafana/grafana-enterprise
```

Run a specific version of Grafana

> ***Note:*** *If you are on a Linux system, you might need to add* `sudo` *before the command or add your user to the* `docker` *group.*

```
docker run -d -p 3000:3000 --name grafana grafana/grafana-enterprise:<version
number>
```

Example:

```
docker run -d -p 3000:3000 --name grafana grafana/grafana-enterprise:8.2.0
```

Run the Grafana main branch

For every successful build of the main branch, we update the `grafana/grafana-oss:main` and `grafana/grafana-oss:main-ubuntu` tags. Additionally, two new tags are created, `grafana/grafana-oss-dev:<version>-<build ID>pre` and `grafana/grafana-oss-dev:<version>-<build ID>pre-ubuntu`, where *version* is the next version of Grafana and *build ID* is the ID of the corresponding CI build. Use these to get access to the latest main builds of Grafana.

When running Grafana main in production, we *strongly* recommend that you use the `grafana/grafana-oss-dev:<version>-<build ID>pre` tag. This tag guarantees that you use a specific version of Grafana instead of whatever was the most recent commit at the time.

For a list of available tags, check out grafana/grafana-oss and grafana/grafana-oss-dev.

Install plugins in the Docker container

You can install official and community plugins listed on the Grafana plugins page or from a custom URL.

Install official and community Grafana plugins

Pass the plugins you want installed to Docker with the `GF_INSTALL_PLUGINS` environment variable as a comma-separated list. This sends each plugin name to `grafana-cli plugins install ${plugin}` and installs them when Grafana starts.

```
docker run -d \
  -p 3000:3000 \
  --name=grafana \
  -e "GF_INSTALL_PLUGINS=grafana-clock-panel,grafana-simple-json-datasource" \
  grafana/grafana-enterprise
```

> ***Note:*** *If you need to specify the version of a plugin, then you can add it to the* `GF_INSTALL_PLUGINS` *environment variable. Otherwise, the latest is used. For example:* `-e "GF_INSTALL_PLUGINS=grafana-`

> *`clock-panel 1.0.1,grafana-simple-json-datasource 1.3.5"`.*

Install plugins from other sources

> *Only available in Grafana v5.3.1 and later.*

You can install a plugin from a custom URL by specifying the URL like this: `GF_INSTALL_PLUGINS=<url to plugin zip>;<plugin install folder name>`.

```
docker run -d \
  -p 3000:3000 \
  --name=grafana \
  -e "GF_INSTALL_PLUGINS=http://plugin-domain.com/my-custom-plugin.zip;custom-
plugin,grafana-clock-panel" \
  grafana/grafana-enterprise
```

Build and run a Docker image with pre-installed plugins

You can build your own customized image that includes plugins. This saves time if you are creating multiple images and you want them all to have the same plugins installed on build.

In the Grafana GitHub repository there is a folder called `packaging/docker/custom/`, which includes two Dockerfiles, `Dockerfile` and `ubuntu.Dockerfile`, that can be used to build a custom Grafana image. It accepts `GRAFANA_VERSION`, `GF_INSTALL_PLUGINS`, and `GF_INSTALL_IMAGE_RENDERER_PLUGIN` as build arguments.

Build with pre-installed plugins

> *If you need to specify the version of a plugin, you can add it to the `GF_INSTALL_PLUGINS` build argument. Otherwise, the latest will be assumed. For example: `--build-arg "GF_INSTALL_PLUGINS=grafana-clock-panel 1.0.1,grafana-simple-json-datasource 1.3.5"`*

Example of how to build and run:

```
cd packaging/docker/custom
docker build \
  --build-arg "GRAFANA_VERSION=latest" \
  --build-arg "GF_INSTALL_PLUGINS=grafana-clock-panel,grafana-simple-json-
datasource" \
  -t grafana-custom -f Dockerfile .

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

Build with pre-installed plugins from other sources

You can build a Docker image with plugins from other sources by specifying the URL like this: `GF_INSTALL_PLUGINS=<url to plugin zip>;<plugin install folder name>`.

```
cd packaging/docker/custom
docker build \
  --build-arg "GRAFANA_VERSION=latest" \
  --build-arg "GF_INSTALL_PLUGINS=http://plugin-domain.com/my-custom-
plugin.zip;custom-plugin,grafana-clock-panel" \
  -t grafana-custom -f Dockerfile .

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

Replace `Dockerfile` in above example with `ubuntu.Dockerfile` to build a custom Ubuntu based image (Grafana v6.5+).

Build with Grafana Image Renderer plugin pre-installed

> *Only available in Grafana v6.5 and later. This is experimental.*

The [Grafana Image Renderer plugin]({{< relref "../image-rendering/#grafana-image-renderer-plugin" >}}) does not currently work if it is installed in a Grafana Docker image. You can build a custom Docker image by using the `GF_INSTALL_IMAGE_RENDERER_PLUGIN` build argument. This installs additional dependencies needed for the Grafana Image Renderer plugin to run.

Example of how to build and run:

```
cd packaging/docker/custom
docker build \
  --build-arg "GRAFANA_VERSION=latest" \
  --build-arg "GF_INSTALL_IMAGE_RENDERER_PLUGIN=true" \
  -t grafana-custom -f Dockerfile .

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

Replace `Dockerfile` in above example with `ubuntu.Dockerfile` to build a custom Ubuntu-based image (Grafana v6.5+).

Migrate from previous Docker containers versions

This section contains important information if you want to migrate from previous Grafana container versions to a more current one.

Migrate to v7.3 or later

The Grafana Docker image runs with the `root` group (id 0) instead of the `grafana` group (id 472), for better compatibility with OpenShift. If you extend the official Docker image you may need to change your scripts to use the `root` group instead of `grafana`.

Migrate to v6.5 or later

Grafana Docker image now comes in two variants, one Alpine based and one Ubuntu based, see Image Variants for details.

Migrate to v6.4 or later

Grafana Docker image was changed to be based on Alpine instead of Ubuntu.

Migrate to v5.1 or later

The Docker container for Grafana has seen a major rewrite for 5.1.

Important changes

- File ownership is no longer modified during startup with `chown`.
- Default user ID is now `472` instead of `104`.
- Removed the following implicit volumes:
  - `/var/lib/grafana`
  - `/etc/grafana`
  - `/var/log/grafana`

Removal of implicit volumes

Previously `/var/lib/grafana`, `/etc/grafana` and `/var/log/grafana` were defined as volumes in the `Dockerfile`. This led to the creation of three volumes each time a new instance of the Grafana container started, whether you wanted it or not.

You should always be careful to define your own named volume for storage, but if you depended on these volumes, then you should be aware that an upgraded container will no longer have them.

Warning: When migrating from an earlier version to 5.1 or later using Docker compose and implicit volumes, you need to use `docker inspect` to find out which volumes your container is mapped to so that you can map them to the upgraded container as well. You will also have to change file ownership (or user) as documented below.

User ID changes

In Grafana v5.1, we changed the ID and group of the Grafana user and in v7.3 we changed the group. Unfortunately this means that files created prior to v5.1 won't have the correct permissions for later versions. We made this change so that it would be more likely that the Grafana users ID would be unique to Grafana. For example, on Ubuntu 16.04 `104` is already in use by the syslog user.

Version	User	User ID	Group	Group ID
< 5.1	grafana	104	grafana	107
>= 5.1	grafana	472	grafana	472
>= 7.3	grafana	472	root	0

There are two possible solutions to this problem. Either you start the new container as the root user and change ownership from `104` to `472`, or you start the upgraded container as user `104`.

Run Docker as a different user

```
docker run --user 104 --volume "<your volume mapping here>" grafana/grafana-
enterprise:8.2.0
```

Specify a user in docker-compose.yml

```
version: '2'

services:
  grafana:
    image: grafana/grafana-enterprise:8.2.0
    ports:
      - 3000:3000
    user: '104'
```

Modify permissions

The commands below run bash inside the Grafana container with your volume mapped in. This makes it possible to modify the file ownership to match the new container. Always be careful when modifying permissions.

```
$ docker run -ti --user root --volume "<your volume mapping here>" --entrypoint bash
grafana/grafana-enterprise:8.2.0

# in the container you just started:
chown -R root:root /etc/grafana && \
chmod -R a+r /etc/grafana && \
chown -R grafana:grafana /var/lib/grafana && \
chown -R grafana:grafana /usr/share/grafana
```

Next steps

Refer to the [Getting Started]({{< relref "../getting-started/getting-started/" >}}) guide for information about logging in, setting up data sources, and so on.

Configure Docker image

Refer to [Configure a Grafana Docker image]({{< relref "../administration/configure-docker.md" >}}) page for details on options for customizing your environment, logging, database, and so on.

Configure Grafana

Refer to the [Configuration]({{< relref "../administration/configuration.md" >}}) page for details on options for customizing your environment, logging, database, and so on.