+++ title = “Configure Grafana Docker image” description = “Guide for configuring the Grafana Docker image” keywords = [“grafana”, “configuration”, “documentation”, “docker”] aliases = [“/docs/grafana/latest/installation/configure-docker/”] weight = 200 +++

Configure a Grafana Docker image

If you are running Grafana in a Docker image, then you configure Grafana using [environment variables]({{< relref “../administration/configuration.md#configure-with-environment-variables” >}}) rather than directly editing the configuration file. If you want to save your data, then you also need to designate persistent storage or bind mounts for the Grafana container.

> **Note:** These examples use the Grafana Enterprise docker image. You can use the Grafana Open Source edition by changing the docker image to `grafana/grafana-oss`.

Save your Grafana data

If you do not designate a location for information storage, then all your Grafana data disappears as soon as you stop your container. To save your data, you need to set up persistent storage or bind mounts for your container.

Run Grafana container with persistent storage (recommended)

```
# create a persistent volume for your data in /var/lib/grafana (database and plugins)
docker volume create grafana-storage

# start grafana
docker run -d -p 3000:3000 --name=grafana -v grafana-storage:/var/lib/grafana grafana/grafan
```

Run Grafana container using bind mounts

You may want to run Grafana in Docker but use folders on your host for the database or configuration. When doing so, it becomes important to start the container with a user that is able to access and write to the folder you map into the container.

```
mkdir data # creates a folder for your data
ID=$(id -u) # saves your user id in the ID variable

# starts grafana with your user id and using the data folder
docker run -d --user $ID --volume "$PWD/data:/var/lib/grafana" -p 3000:3000 grafana/grafana-
```

Default paths

The following settings are hard-coded when launching the Grafana Docker container and can only be overridden using environment variables, not in `conf/grafana.ini`.

Setting	Default value
GF_PATHS_CONFIG	/etc/grafana/grafana.ini
GF_PATHS_DATA	/var/lib/grafana
GF_PATHS_HOME	/usr/share/grafana
GF_PATHS_LOGS	/var/log/grafana
GF_PATHS_PLUGINS	/var/lib/grafana/plugins
GF_PATHS_PROVISIONING	/etc/grafana/provisioning

Logging

Logs in the Docker container go to standard out by default, as is common in the Docker world. Change this by setting a different [log mode]({{< relref "../administration/configuration.md#mode" >}}).

Example:

```
# Run Grafana while logging to both standard out and /var/log/grafana/grafana.log
docker run -p 3000:3000 -e "GF_LOG_MODE=console file" grafana/grafana-enterprise
```

Configure Grafana with Docker Secrets

> Only available in Grafana v5.2 and later.

It's possible to supply Grafana with configuration through files. This works well with Docker Secrets as the secrets by default gets mapped into `/run/secrets/<name of secret>` of the container.

You can do this with any of the configuration options in conf/grafana.ini by setting `GF_<SectionName>_<KeyName>__FILE` to the path of the file holding the secret.

For example, you could set the admin password this way:

- Admin password secret: `/run/secrets/admin_password`
- Environment variable: `GF_SECURITY_ADMIN_PASSWORD__FILE=/run/secrets/admin_password`

Configure AWS credentials for CloudWatch Support

```
docker run -d \
-p 3000:3000 \
--name=grafana \
-e "GF_AWS_PROFILES=default" \
-e "GF_AWS_default_ACCESS_KEY_ID=YOUR_ACCESS_KEY" \
-e "GF_AWS_default_SECRET_ACCESS_KEY=YOUR_SECRET_KEY" \
-e "GF_AWS_default_REGION=us-east-1" \
grafana/grafana-enterprise
```

You may also specify multiple profiles to `GF_AWS_PROFILES` (e.g. `GF_AWS_PROFILES=default another`).

Supported variables:

- `GF_AWS_${profile}_ACCESS_KEY_ID`: AWS access key ID (required).
- `GF_AWS_${profile}_SECRET_ACCESS_KEY`: AWS secret access key (required).
- `GF_AWS_${profile}_REGION`: AWS region (optional).