Configure a Grafana Docker image

 › [Administration](https://grafana.com/docs/grafana/latest/administration/) › Configure Docker image

If you are running Grafana in a Docker image, then you configure Grafana using [environment variables](https://grafana.com/docs/grafana/latest/administration/configuration/#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.

如果你是在Docker镜像中运行Grafana的话，那么你使用环境变量来配置Grafana比编辑配置文件更加直接。如果你想要去保存你的数据，那么你需要去设计持久化存储或为Grafana窗口绑定加载的文件系统。

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 image. To save your data, you need to set up persistent storage or bind mounts for your container.

如果你没有为信息存储设计一个位置，那么你一停止你的镜像，你的所有数据都会消失。为了保存你的数据，你需要去安装一个持久化存储或为你的窗口绑定加载的文件系统。

Run Grafana container with persistent storage (recommended)

使用持久化存储运行Grafana容器（建议）

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

# 在/var/lib/grafana中为你的数据创建一个持久化卷（数据库和插件）

docker volume create grafana-storage

# start grafana

docker run -d -p 3000:3000 --name=grafana -v grafana-storage:/var/lib/grafana grafana/grafana

Bash

Run Grafana container using bind mounts

绑定加载的文件系统去运行Grafana容器

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.

你可能想要在Docker中运行Grafana，为了你的数据库或配置来使用你在主机上的文件夹。当这样做的时候，使用有权限的用户去运行容器，然后映射文件夹到容器中，这是很重要的。

mkdir data # creates a folder for your data 为你的数据创建一个文件夹

ID=$(id -u) # saves your user id in the ID variable 在ID变量中保存你的用户id

# starts grafana with your user id and using the data folder

# 使用你的用户id和数据文件夹来开始你的Grafana

docker run -d --user $ID --volume "$PWD/data:/var/lib/grafana" -p 3000:3000 grafana/grafana:5.1.0

Bash

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.

当运行Grafana的Docker容器时，下面的设置是被写死的，你只能使用环境变量来重写它们，还不是在conf/grafana.ini中。

| 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 |

Configure Grafana with Docker Secrets

Only available in Grafana v5.2 and later.

只在Grafana 5.2及以后的版本中可用。

It’s possible to supply Grafana with configuration through files. This works well with [Docker Secrets](https://docs.docker.com/engine/swarm/secrets/) as the secrets by default gets mapped into /run/secrets/<name of secret> of the container.

可以通过配置文件来供给Grafana。使用Docker隐私作为隐私并映射到窗口的/run/secrets/<name of secret>。

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.

你也可以在conf/grafana.ini中通过设置GF\_<SectionName>\_<KeyName>\_\_FILE的文件路径来解决隐私。

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

例如，你可以使用这种方法来设置管理员密码：

* Admin password secret: /run/secrets/admin\_password

管理员密码隐私：/run/secrets/admin\_password

* Environment variable: GF\_SECURITY\_ADMIN\_PASSWORD\_\_FILE=/run/secrets/admin\_password

环境变量： GF\_SECURITY\_ADMIN\_PASSWORD\_\_FILE=/run/secrets/admin\_password

Configure AWS credentials for CloudWatch Support

为了CloudWatch的支持配置AWS认证

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

Bash

You may also specify multiple profiles to GF\_AWS\_PROFILES (e.g. GF\_AWS\_PROFILES=default another).

你也可以指定多个配置文件到GF\_AWS\_PROFILES(如：GF\_AWS\_PROFILES=default another)

Supported variables:

* GF\_AWS\_${profile}\_ACCESS\_KEY\_ID: AWS access key ID (required).

AWS访问密钥ID（要求）

* GF\_AWS\_${profile}\_SECRET\_ACCESS\_KEY: AWS secret access key (required).

AWS隐私访问密钥（要求）

* GF\_AWS\_${profile}\_REGION: AWS region (optional).

AWS域（可选）