Run Grafana Docker image

 › [Installation](https://grafana.com/docs/grafana/latest/installation/) › Run Docker image

You can install and run Grafana using the official Docker container. The official Grafana Docker image comes in two variants: Alpine and Ubuntu.

你可以使用正式的Docker container来安装和运行Grafana。正式的Grafana Docker image有两种：Alpine和Ubuntu。

This page also contains important information about [migrating from earlier Docker container versions](https://grafana.com/docs/grafana/latest/installation/docker/#migrate-from-previous-docker-containers-versions).

本节也包含了关于从老的Docker container版本进行迁移的重要信息。

Alpine image (recommended)

Alpine镜像（建议）

grafana/grafana:<version>

This is the default image. This image is based on the popular [Alpine Linux project](http://alpinelinux.org/), available in [the alpine official image](https://hub.docker.com/_/alpine). Alpine Linux is much smaller than most distribution base images, and thus leads to slimmer and more secure images.

这是一个默认的镜像。这个镜像基于很受欢迎的Alpine Linux项目，在alpine正式镜像是可用的。Alpine Linux对于其他大多数的基本镜像来说是非常小的，这样使得更轻量更安全。

This 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 does use [musl libc](http://www.musl-libc.org/) instead of [glibc and friends](http://www.etalabs.net/compare_libcs.html), so certain software might run into issues depending on the depth of their libc requirements. However, most software doesn’t have an issue with this, so this variant is usually a very safe choice.

当安全性和最终镜像的大小有要求时，非常建议使用这种镜像。最需要注意的是它使用了musl\_libc代替了glibc\_and\_friends，因此，某些软件可能会遇到问题，这取决于它们对于libc的依赖程度。但是大多数的软件是没有这个问题的，所以这种镜像是一个很安全的选择。

**Note:** The grafana/grafana:<version> image was based on [Ubuntu](https://ubuntu.com/) before version 6.4.0.

注意：这个grafana/grafana:<version>镜像是基本于Ubuntu 6.4.0之前的版本。

Ubuntu image

Ubuntu镜像

grafana/grafana:<version>-ubuntu

This image is based on [Ubuntu](https://ubuntu.com/), available in [the Ubuntu official image](https://hub.docker.com/_/ubuntu). This is an alternative image for those who prefer an [Ubuntu](https://ubuntu.com/) based image and/or who are dependent on certain tooling not available for Alpine.

这个镜像是基于Ubuntu的，在Ubuntu正式镜像上可用。这对于那些倾向于使用Ubuntu或当前所依赖的工具不能在Alpine情况来说，是另一种选择。

**Note:** The grafana/grafana:<version>-ubuntu image is available for Grafana version 6.5.0 and later.

注意：这个grafana/grafana:<version>-ubuntu镜像可用于Grafana 6.5.0版本或之后中。

Run Grafana

运行Grafana

You can run the latest Grafana version, run a specific version, or run an unstable version based on the master branch of the [grafana/grafana GitHub repository](https://github.com/grafana/grafana).

你可以运行最新版的Grafana，运行指定的版本，或者运行基于主分支上的非稳定的版本。

Run the latest stable version of Grafana

运行最新的稳定版的Grafana

**Note:** If you are on a Linux system, you might need to add sudo before the command.

注意：如果你是Linux系统，你可能需要在命令前添加sudo。

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

Bash

Run a specific version of Grafana

运行指定版本的Grafana

**Note:** If you are on a Linux system, you might need to add sudo before the command.

注意：如果你是Linux系统，你可能需要在命令前添加sudo。

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

Bash

**Example:**

docker run -d -p 3000:3000 --name grafana grafana/grafana:6.5.0

Bash

Run the Grafana master branch

运行Grafana的主分支

For every successful build of the master branch, we update the grafana/grafana:master and grafana/grafana:master-ubuntu tags. Additionally, two new tags are created, grafana/grafana-dev:master-<commit hash> and grafana/grafana-dev:master-<commit hash>-ubuntu, which includes the hash of the Git commit that was built. Use these to get access to the latest master builds of Grafana.

为了每一个主分支的成功Build，我们更新了grafana/grafana:master和grafana/grafana:master-ubuntu两个标签。另外的，有两个新的标签被创建，grafana/grafana-dev:master-<commit hash>和grafana/grafana-dev:master-<commit hash>-ubuntu，Build时包含了Git的提交Hash。使用这些可以去访问最新的Grafana的Master Build。

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

当在生产环境运行Grafana的Master时，我们强烈建议你使用grafana/grafana-dev:master-<commit hash>标签。这个标签保证您使用的是Grafana的一个特定版本，而不是当时最近提交的那个版本。

For a list of available tags, check out [grafana/grafana](https://hub.docker.com/r/grafana/grafana/tags/) and [grafana/grafana-dev](https://hub.docker.com/r/grafana/grafana-dev/tags/).

可以check out grafana/grafana和grafana/grafana-dev来查看所有可用的列表。

Install plugins in the Docker container

在Docker container中安装插件

You can install official and community plugins listed on the Grafana [plugins page](https://grafana.com/grafana/plugins) or from a custom URL.

你可以从Grafana插件页面或从自定义URL上，安装正式版或社区版的插件。

Install official and community Grafana plugins

安装正式版或社区版的Grafana插件

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.

将你想要安装的插件以GF\_INSTALL\_PLUGINS为参数，并以逗号为分隔。当Grafana开始时，会发送每个插件的名字给grafana-cli plugins install ${plugin}，并安装它们。

docker run -d \

-p 3000:3000 \

--name=grafana \

-e "GF\_INSTALL\_PLUGINS=grafana-clock-panel,grafana-simple-json-datasource" \

grafana/grafana

Bash

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 will be assumed. For example: -e "GF\_INSTALL\_PLUGINS=grafana-clock-panel 1.0.1,grafana-simple-json-datasource 1.3.5".

如果你需要指定插件的版本，你可以把它们添加到GF\_INSTALL\_PLUGINS环境变量中。否则的话，默认会安装最新的版本。例如：-e "GF\_INSTALL\_PLUGINS=grafana-clock-panel 1.0.1,grafana-simple-json-datasource 1.3.5"。

Install plugins from other sources

从其他地方安装插件

Only available in Grafana v5.3.1 and later.

只支持Grafana 5.3.1版本或之后的版本。

You can install plugins from custom URLs by specifying the URL like this: GF\_INSTALL\_PLUGINS=<url to plugin zip>;<plugin name>.

你可以指定URL来安装插件，例如：GF\_INSTALL\_PLUGINS=<url to plugin zip>;<plugin name>.

docker run -d \

-p 3000:3000 \

--name=grafana \

-e "GF\_INSTALL\_PLUGINS=http://plugin-domain.com/my-custom-plugin.zip;custom-plugin" \

grafana/grafana

Bash

Build and run a Docker image with pre-installed plugins

使用预安装的插件构建和运行Docker镜像

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.

你可以构建自己的自定义镜像，其中包括插件。如果你创建多个镜像，并且希望它们在构建时都安装相同的插件，这将节省时间。

In the [Grafana GitHub repository](https://github.com/grafana/grafana/tree/master/packaging/docker) there is a folder called 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.

在Grafana Github库中，有一个叫custom的文件夹，里面有两个Dockerfile，Dockerfile和ubuntu.Dockerfile，可以用来构建自定义的Grafana镜像。它可以接收GRAFANA\_VERSION, GF\_INSTALL\_PLUGINS, 和GF\_INSTALL\_IMAGE\_RENDERER\_PLUGIN 作为构建参数。

Build with pre-installed plugins

使用预安装的插件进行构建

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"

如果你需要指定插件的版本，你可以把它们添加到GF\_INSTALL\_PLUGINS环境变量中。否则的话，默认会安装最新的版本。例如：-e "GF\_INSTALL\_PLUGINS=grafana-clock-panel 1.0.1,grafana-simple-json-datasource 1.3.5"。

Example of how to build and run:

怎样去构建并运行的例子：

cd 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

Bash

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

使用ubuntu.Dockerfile去替换上面例子中的Dockerfile，可以构建自定义的Ubuntu镜像（Grafana 6.5版本以上适用）

Build with Grafana Image Renderer plugin pre-installed

使用预安装的插件构建Grafana镜像渲染

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

只有Grafana 6.5版本及以后的版本有效。这是一个实验性的功能。

The [Grafana Image Renderer plugin](https://grafana.com/docs/grafana/latest/administration/image_rendering/#grafana-image-renderer-plugin) does not currently work if it is installed in 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.

如果被安装在了Grafana Docker镜像中，那么Grafana Image Renderer Plugin当前是不能工作的。你可以使用GF\_INSTALL\_IMAGE\_RENDERER\_PLUGIN构建参数来构建一个你自定义的镜像。它会额外安装Grafana镜像渲染的依赖去运行。

Example of how to build and run:

怎样构建并运行的例子:

cd 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

Bash

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

使用ubuntu.Dockerfile去替换上面例子中的Dockerfile，可以构建自定义的Ubuntu镜像（Grafana 6.5版本以上适用）

Migrate from previous Docker containers versions

迁移以前的Docker Containers版本

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

如果你想要迁移以前的Grafana Container到现在的Container，那么这个章节是非常重要的。

Migrate to v6.5 or later

迁移到6.5版本或以后的版本

Grafana Docker image now comes in two variants, one [Alpine](http://alpinelinux.org/) based and one [Ubuntu](https://ubuntu.com/) based, see [Image Variants](https://grafana.com/docs/grafana/latest/installation/docker/#image-variants) for details.

Grafana Docker镜像现在有两种类型，一种基于Alpine，一种基于Ubuntu，查看Image Variants来了解详情。

Migrate to v6.4 or later

迁移到6.4版本或以后的版本

Grafana Docker image was changed to be based on [Alpine](http://alpinelinux.org/) instead of [Ubuntu](https://ubuntu.com/).

Grafana Docker镜像已经被更改为基于Alpine来替代Ubuntu.

Migrate to v5.1 or later

迁移到5.1版本或以后的版本

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

Grafana已经对5.1版本的Docker Container进行了重写。

**Important changes**

**重要的更改**

* File ownership is no longer modified during startup with chown.

文件的所有权不会在使用chown创建时进行修改

* Default user ID is now 472 instead of 104.

默认的用户ID现在是472，而不是104

* Removed the following implicit volumes:

移除了下面暗藏的Volume：

* + /var/lib/grafana
  + /etc/grafana
  + /var/log/grafana

Removal of implicit volumes

移除暗藏的Volume

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.

在Dockerfile中，/var/lib/grafana, /etc/grafana 和/var/log/grafana 之前被定义为Volume。这就使得无论你使用需要使用它们，它们都会在Grafana Container开始的时候就被创建。

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.

你应该时刻小心的去定义你自己的存储Volume，如果你依赖去这些volume，那么你应该小心，如果你升级了Container后，可能它们就不在了。

**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.

警告：当你使用Docker Compose和暗藏Volume，从更早的版本迁移到5.1版本或以后的版本时，你需要使用docker inspect去找到你映射的Volume，以便于你在升级后可以再次映射到你的Container。你也可以修改如下记录的文件所有权（或文件的用户）。

User ID changes

更改用户ID

In Grafana v5.1, we changed the ID of the Grafana user. 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.

在Grafana 5.1版本中，我们更改了Grafana的用户ID。不幸的是，这意味着你在5.1版本期间创建的文件，在之后的版本中不能获取到正常的权限。我们去做了这个更改，是为了让用户ID对Grafana来说是唯一的。例如，在Ubuntu 16.04版本中，104已经被分配给了syslog用户。

| Version | User | User ID |
| --- | --- | --- |
| < 5.1 | grafana | 104 |

= 5.1 | grafana | 472

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.

这里有两种解决方案来解决这个问题。你可以使用Root用户来启动Container并更改文件所有权从104到472，或者你升级Container用户为104。

Run Docker as a different user

作为不同的用户来运行Docker

docker run --user 104 --volume "<your volume mapping here>" grafana/grafana:5.1.0

Bash

Specify a user in docker-compose.yml

在docker-compose.yml中指定用户

version: "2"

services:

grafana:

image: grafana/grafana:5.1.0

ports:

- 3000:3000

user: "104"

YAML

Modify permissions

修改权限

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.

使用下面的命令可以将你的Volume映射到你的Grafana Container当中。这就有可能去修改文件的所有权来匹配新的Container。当你修改权限的时候都要小心一些。

$ docker run -ti --user root --volume "<your volume mapping here>" --entrypoint bash grafana/grafana:5.1.0

# 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

Bash

Next steps

下一步

Refer to the [Getting Started](https://grafana.com/docs/grafana/latest/getting-started/getting-started/) guide for information about logging in, setting up data sources, and so on.

到Getting Started指南去了解登录，创建数据源等。

Configure Docker image

配置Docker镜像

Refer to [Configure a Grafana Docker image](https://grafana.com/docs/grafana/latest/installation/configure-docker/) page for details on options for customizing your environment, logging, database, and so on.

到Configure a Grafana Docker image页面去详细了解如果自定义你的环境，登录，数据库等。

Configure Grafana

配置Grafana

Refer to the [Configuration](https://grafana.com/docs/grafana/latest/administration/configuration/) page for details on options for customizing your environment, logging, database, and so on.

到Configuration页面去了解自定义环境，登录，数据库等。