

# Docker Dockerfile 使用方法

[Dockerfile 介绍](#)

[创建Dockerfile文件](#)

[构建 Docker 镜像](#)

[查看已下载的镜像](#)

[运行 mysql 命令](#)

## Dockerfile 介绍

当使用Docker构建容器化应用程序时，Dockerfile是一个用于定义容器镜像的文本文件。它包含了一系列指令，告诉Docker如何从基础镜像（通常是官方或自定义的操作系统镜像）构建出最终的镜像，以及如何配置容器中的环境、文件和应用程序。

Dockerfile 的编写是构建容器的基础，它允许您定义容器的构建步骤、环境和配置。通过合理使用各种指令，您可以构建出一个满足应用程序需求的定制化镜像，从而实现应用的容器化部署。

## 创建Dockerfile文件

并将以下内容粘贴到文件中，然后将文件命名为 **Dockerfile**：

```
1  # 使用官方的 MySQL 镜像作为基础镜像
2  FROM mysql:latest
3
4  # MySQL密码
5  ENV MYSQL_ROOT_PASSWORD=123456
6
7  # 暴露 MySQL 的默认端口
8  EXPOSE 3306
```

(1) 使用 **FROM** 关键字指定了基础镜像。在这个示例中，我们选择了官方的 MySQL 镜像，并使用 **:latest** 标签，表示我们将使用最新的 MySQL 版本作为基础。

(2) 使用 **ENV** 关键字设置了一个环境变量 **MYSQL\_ROOT\_PASSWORD**，用于指定 MySQL 数据库的 root 用户的密码。

(3) 使用 **EXPOSE** 关键字指定了容器将要监听的端口。在这里，我们暴露了容器内部的 3306 端口，这是 MySQL 数据库的默认端口。

## 构建 Docker 镜像

打开终端，并导航到包含 Dockerfile 的目录，执行以下命令来构建 Docker 镜像：

```
1 docker build -t my-mysql-container .
```

其中，**my-mysql-container** 是你为镜像指定的名称，**.** 表示 Dockerfile 在当前目录。

## 查看已下载的镜像

使用 **docker images** 命令可以列出所有本地已下载的 Docker 镜像：

```
1 docker images
```

## 运行 mysql 命令

```
1 docker run -itd --name mysql-test -p 3306:3306 my-mysql-containe
```

其中，**my-mysql-container** 是你为镜像指定的名称。