

# 部署开源项目管理工具 Masterlab

作者: [luofeng0603](#)

原文链接: <https://ld246.com/article/1682390027583>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



## 前言

新项目要开工，需要一个项目管理软件，目前一些成品话的项目管理软件不是太重就是需要企业资质对于小团队来讲，其实不是很好~所以找了一个MasterLab。比较清爽，小团队用用，问题不大。

## 私有部署

Masterlab是开源的，可以私有部署。Docker项目地址：<https://hub.docker.com/r/gopeak/masterlab>

推荐使用的是docker-composer部署。那么就按官方文档操作一下。

## 第一步：安装git和unzip

拿到的机器是centos8的，直接：

```
yum install -y git
yum install -y unzip
```

## 第二步：获取Docker项目代码

```
git clone https://gitee.com/firego/masterlab-docker.git
# 海外网络请用 https://github.com/gopeak/masterlab-docker.git
```

## 第三步：获取Masterlab程序

```
cd ./masterlab-docker/www
git clone https://gitee.com/firego/masterlab.git
# 海外网络请用 https://github.com/gopeak/masterlab.git
```

```
cd ./masterlab
unzip ./vendor.zip // 解压依赖的类库
```

## 第四部：运行容器编排

首先应配置镜像加速器（可选）：

Linux操作系统针对Docker客户端版本大于 1.10.0 的用户

您可以通过修改daemon配置文件/etc/docker/daemon.json来使用加速器

```
mkdir -p /etc/docker
tee /etc/docker/daemon.json <<-'EOF'
{
  "registry-mirrors": ["https://c9jzs6ju.mirror.aliyuncs.com"]
}
EOF
systemctl daemon-reload
systemctl restart docker
```

加速镜像配置完毕后，启用服务，第一次需要构建镜像

```
cd ../../ // 回到masterlab-docker的根目录
docker-compose up -d // 容器编排命令
```

容器编排成功后，以后可以使用以下命令控制服务

```
docker-compose ps // 查看运行的容器
docker-compose stop // 停止服务
docker-compose start // 启动服务
docker-compose restart // 重启服务
docker-compose start nginx // 单独启动nginx服务
docker-compose stop nginx // 单独停止nginx服务
docker-compose logs // 查看服务日志输出
docker-compose kill nginx // 通过发送 SIGKILL 信号来停止指定服务的容器
docker-compose logs // 查看服务日志输出
docker-compose up // 构建并启动服务
docker-compose build // 构建或者重新构建服务
docker network ls // 查看docker的网络配置
```

## 第五步：赋予权限

```
# 查看运行已经运行的容器,找到php的容器id如ee84df733af6
docker ps
# 进入php
docker exec -it ee84df733af6 /bin/bash
# 赋予读写权限, 执行完 ctrl + d 退出
chown -R www-data:www-data ./
# 按键 ctrl + d 退出
```

## 第六步：访问 /install 进行图形安装界面

访问: 域名/install, 这个域名得提前配置好, 服务器开放80端口, 或者就是直接访问服务器的外网 ip install 也行。

进去之后, 就是php项目的配置引导, 一共4步, 我忘记截图了。

注意其中有一部, 配置redis, 可以勾选, redis的ip需要手动查看一下。端口是默认的6379, 点击下一步的时候, 会测试redis连接是否正常。

```
docker ps #查看redis的 id
docker inspect redisId #进去查看分配的网络ip
```

注: 连接数据库的地址, 用户名,密码分别为

```
172.100.0.5 root 123456
```

如果是通过ip/域名安装的, 安装结束后 将 ./conf/nginx/conf.d/masterlab.conf 里的地址改为你自己的域名即可。

安装这一步非常的耗时, 不知道等了多久~

## 第七步: 启动Swoole服务

这一步, 跟它Gitee上的步骤略有出入, Gitee多一个步骤:

- 修改masterlab的配置文件 config.yml,找到 `socket/host`节点, 将`127.0.0.1`修改为`0.0.0.0`

```
socket:
  host: '0.0.0.0'
  port: '9002'
  port_websocket: 9003
```

- 首先, 自定义构建 `PHP-CLI` 镜像, 安装 `Git`, `Composer`, `Swoole` 等扩展和工具

```
# 构拉取php-cli镜像
docker pull gopeak/masterlab:php-cli-74
```

- 启动 `Swoole` 进程

```
docker run -d -it --rm --name www-data --network masterlabdocker_docker_net --ip 172.1
0.0.8 \
  -p 9002:9002 \
  -v "$PWD"/www/masterlab:/usr/workspaces/project \
  -w /usr/workspaces/project \
  gopeak/masterlab:php-cli-74 \
  php ./bin/swoole_server.php
```

# 注意: 此时你所在的目录应该是在mastlab这个文件夹下

启动之后, 使用docker ps 观察下启动的容器, 正常的话, 应该是5个容器, docker-composer启动4个, 最后我们手动启动了一个php-cli 所以一共是5个。