

🔗 动手实战学Docker (/courses/498)

# 使用 Docker 运行 Wordpress

## 1. 课程说明

课程为纯动手实验教程，为了能说清楚实验中的一些操作会加入理论内容。理论内容我们不会写太多，已经有太多好文章了，会精选最值得读的文章推荐给你，在动手实践的同时扎实理论基础。

实验环境中可以联网，不受实验楼网络限制。

## 2. 学习方法

实验楼的 Docker 课程包含 14 个实验，每个实验都提供详细的步骤和截图，适用于有一定 Linux 系统基础，想快速上手 Docker 的同学。

学习方法是多实践，多提问。启动实验后按照实验步骤逐步操作，同时理解每一步的详细内容。

如果实验开始部分有推荐阅读的材料，请务必先阅读后再继续实验，理论知识是实践必要的基础。

## 3. 本节内容简介

在本实验里我们将通过完成一个 Wordpress 的容器来学习 Dockerfile 及 Docker 的运行机制。

本节中，我们需要依次完成下面几项任务：

1. Wordpress 的安装及配置
2. Dockerfile 的编写
3. 从 Dockerfile 构建镜像

本次实验的需求是完成一个 Dockerfile，通过该 Dockerfile 创建一个 Wordpress 应用。尽管 Dockerhub 上已经提供了官方的 Wordpress 镜像，本实验仅仅用于学习 Dockerfile 及 Docker 机制。

WordPress 是一种使用 PHP 语言开发的博客平台，用户可以在支持 PHP 和 MySQL 数据库的服务器上架设属于自己的网站。也可以把 WordPress 当作一个内容管理系统（CMS）来使用。

- 百度百科

## 4 实验准备

动手实战 Docker (/courses/498)

### 4.1 实验分析

在本实验中，除了部署 Wordpress 之外，我们需要安装 Nginx 提供对外的外部服务，同时安装一个 ssh 服务提供便捷的管理。

Wordpress 依赖的包比较多，至少需要安装 php，mysql 等依赖软件。所以为了提高 docker build 速度，我们直接使用阿里云的 Ubuntu 源。因此要在 Dockerfile 开始位置增加下面一句命令：

```
RUN echo "deb http://mirrors.cloud.aliyuncs.com/ubuntu/ trusty main universe" > /etc/apt/sources.list
```

**注意：**如果在自己的电脑操作，请将 mirrors.cloud.aliyuncs.com 改为 mirrors.aliyun.com

### 4.2 创建 Dockerfile 文件

首先，需要创建一个目录来存放 Dockerfile 文件，目录名称可以任意，在目录里创建 Dockerfile 文件：

```
cd /home/shiyanlou
mkdir shiyanlouwordpress
cd shiyanlouwordpress
touch Dockerfile
```

```
shiyanlou@iZ23zslbpciZ: ~
$ cd /home/shiyanlou

shiyanlou@iZ23zslbpciZ: ~
$ mkdir shiyanlouwordpress

shiyanlou@iZ23zslbpciZ: ~
$ cd shiyanlouwordpress

shiyanlou@iZ23zslbpciZ: ~/shiyanlouwordpress
$ touch Dockerfile

shiyanlou@iZ23zslbpciZ: ~/shiyanlouwordpress
$
```



使用 vim/gedit 编辑 Dockerfile 文件，根据我们的需求输入内容。

## 5. Dockerfile 基本框架

按照上一节学习的内容，我们先完成 Dockerfile 基本框架。

依次输入下面的基本框架内容：

```
# Version 0.1

# 基础镜像
FROM ubuntu:latest

# 维护者信息
MAINTAINER shiyanlou@shiyanlou.com

# 镜像操作命令
RUN echo "deb http://mirrors.cloud.aliyuncs.com/ubuntu/ trusty main universe" > /etc/apt/sources.list
RUN apt-get -yqq update && apt-get install -yqq supervisor && apt-get clean

# 容器启动命令
CMD ["supervisord"]
```

上面的Dockerfile创建了一个简单的镜像，使用 Supervisor 启动服务。

## 6. 完善 Dockerfile

在上述基本的架构下，我们根据需求可以增加新的内容到 Dockerfile 中。

### 6.1 安装依赖包

安装的依赖包分为两类，一类是系统需要的服务，比如 nginx 等，一类是 Wordpress 依赖的 php 组件。

增加下面的内容到 Dockerfile：

```
RUN apt-get -yqq install nginx supervisor wget php5-fpm php5-mysql
```

### 6.2 安装 Wordpress

首先创建安装目录：

```
RUN mkdir -p /var/www
```

然后下载 Wordpress 4.4.2 版本压缩包并解压：

➊ 动手实战学Docker (courses/498)  
ADD https://oss-cn-hangzhou-internal.aliyuncs.com/courses/498/wordpress-4.4.2.tar.gz  
/var/www/wordpress-4.4.2.tar.gz  
RUN cd /var/www && tar zxvf wordpress-4.4.2.tar.gz && rm -rf wordpress-4.4.2.tar.gz  
RUN chown -R www-data:www-data /var/www/wordpress

## 6.2 安装 SSH 服务

首先安装所需要的软件包：

```
RUN apt-get install -y openssh-server openssh-client
```

创建运行目录：

```
RUN mkdir /var/run/sshd
```

设置root密码及允许root通过ssh登陆：

```
RUN echo 'root:shianlou' | chpasswd  
RUN sed -i 's/PermitRootLogin without-password/PermitRootLogin yes/' /etc/ssh/sshd_config
```

## 6.3 安装Mysql

首先通过 `debconf-set-selections` 设置安装过程中需要的 root 密码为 shianlou，然后再执行 `apt-get` 安装 mysql：

```
RUN echo "mysql-server mysql-server/root_password password shianlou" | debconf-set-selections  
RUN echo "mysql-server mysql-server/root_password_again password shianlou" | debconf-set-selections  
RUN apt-get install -y mysql-server mysql-client
```

安装后需要创建安装Wordpress所需的 wordpress 数据库：

```
RUN service mysql start && mysql -uroot -pshianlou -e "create database wordpress;"
```

## 6.4 开放端口

开放80（Web服务）和22（SSH服务）端口：

```
EXPOSE 80 22
```

## 6.5 配置 Supervisor

动手实战学 Docker (/courses/498)

添加 Supervisor 配置文件来启动 php5-fpm, nginx, mysql 和 ssh, 创建文件 /home/shiyanlou/s  
hiyanlouwordpress/supervisord.conf, 添加以下内容:

```
[supervisord]
nodaemon=true

[program:php5-fpm]
command=/usr/sbin/php5-fpm -c /etc/php5/fpm
autorstart=true

[program:mysql]
command=/usr/bin/mysqld_safe

[program:nginx]
command=/usr/sbin/nginx
autorstart=true

[program:ssh]
command=/usr/sbin/sshd -D
```

Dockerfile 中增加向镜像内拷贝该文件的命令:

```
COPY supervisord.conf /etc/supervisor/conf.d/supervisord.conf
```

## 6.6 添加启动命令

启动 Supervisor:

```
CMD ["/usr/bin/supervisord"]
```

# 7. 配置文件

## 7.1 Nginx 配置文件

为了能让 Nginx 顺利支持 /var/www/wordpress 目录下的 Wordpress, 我们需要添加文件到 /etc/n  
ginx/sites-available/default。

在 Dockerfile 所在的 /home/shiyanlou/shiyanlouwordpress 目录下创建文件 nginx-config, 并  
输入以下内容:

## 动手实战学Docker (/courses/498)

```
listen *:80;
server_name localhost;

root /var/www/wordpress;
index index.php;

location / {
    try_files $uri $uri/ /index.php?$args;
}

location ~ \.php$ {
    try_files $uri =404;
    include fastcgi_params;
    fastcgi_pass unix:/var/run/php5-fpm.sock;
}
}
```

这是一个基本的Nginx配置，包含的核心配置信息：

1. 指定 Wordpress 的目录： /var/www/wordpress
2. 设置对 PHP 页面的支持，包括设置默认 index 页面为 index.php 等

## 7.2 Wordpress配置文件

Wordpress 配置文件为 /var/www/wordpress/wp-config.php，有两种方法修改这个文件：

1. 使用 sed 更改文件中需要配置的项目
2. 预先配置好该文件，在 docker build 过程中拷贝到镜像中替换原文件

我们这里介绍第一种方法：

```
RUN sed -i 's/database_name_here/wordpress/g' /var/www/wordpress/wp-config-sample.php
RUN sed -i 's/username_here/root/g' /var/www/wordpress/wp-config-sample.php
RUN sed -i 's/password_here/shiyanlou/g' /var/www/wordpress/wp-config-sample.php
RUN mv /var/www/wordpress/wp-config-sample.php /var/www/wordpress/wp-config.php
```

可以看到，其中配置了数据库连接的用户名和密码，还记得前面步骤中安装mysql时设置的内容吗？

修改完成后的 wp-config.php 文件节选：

```
define('DB_NAME', 'wordpress');
/** MySQL database username */
define('DB_USER', 'root');
/** MySQL database password */
define('DB_PASSWORD', 'shiyanlou');
```

## 7.3 Dockerfile 更新

动手实战学Docker (7/courses/498)

在Dockerfile中CMD前面的添加COPY命令，用来更新镜像中的配置文件：

```
COPY nginx-config /etc/nginx/sites-available/default

RUN sed -i 's/database_name_here/wordpress/g' /var/www/wordpress/wp-config-sample.php
RUN sed -i 's/username_here/root/g' /var/www/wordpress/wp-config-sample.php
RUN sed -i 's/password_here/shiyanlou/g' /var/www/wordpress/wp-config-sample.php
RUN mv /var/www/wordpress/wp-config-sample.php /var/www/wordpress/wp-config.php
```

## 8. 从 Dockerfile 创建镜像

将上述内容完成后放入到 /home/shiyanlou/shiyanlouwordpress/Dockerfile 文件中，最终得到的Dockerfile文件如下：

# 动手实战学 Docker (/courses/498)  
FROM ubuntu:latest

MAINTAINER shiyanlou@shiyanlou.com

```
RUN echo "deb http://mirrors.cloud.aliyuncs.com/ubuntu/ trusty main universe" > /etc/apt/sources.list
RUN apt-get -yqq update
RUN apt-get -yqq install nginx supervisor wget php5-fpm php5-mysql
RUN echo "daemon off;" >> /etc/nginx/nginx.conf

RUN mkdir -p /var/www
ADD http://labfile.oss-cn-hangzhou-internal.aliyuncs.com/courses/498/wordpress-4.4.2.tar.gz /var/www/wordpress-4.4.2.tar.gz
RUN cd /var/www && tar zxvf wordpress-4.4.2.tar.gz && rm -rf wordpress-4.4.2.tar.gz
RUN chown -R www-data:www-data /var/www/wordpress

RUN mkdir /var/run/ssh
RUN apt-get install -yqq openssh-server openssh-client
RUN echo 'root:shiyanlou' | chpasswd
RUN sed -i 's/PermitRootLogin without-password/PermitRootLogin yes/' /etc/ssh/sshd_config

RUN echo "mysql-server mysql-server/root_password password shiyanlou" | debconf-set-selections
RUN echo "mysql-server mysql-server/root_password_again password shiyanlou" | debconf-set-selections
RUN apt-get install -yqq mysql-server mysql-client

EXPOSE 80 22

COPY nginx-config /etc/nginx/sites-available/default
COPY supervisord.conf /etc/supervisor/conf.d/supervisord.conf
RUN service mysql start && mysql -uroot -pshiyanlou -e "create database wordpress;"
RUN sed -i 's/database_name_here/wordpress/g' /var/www/wordpress/wp-config-sample.php
RUN sed -i 's/username_here/root/g' /var/www/wordpress/wp-config-sample.php
RUN sed -i 's/password_here/shiyanlou/g' /var/www/wordpress/wp-config-sample.php
RUN mv /var/www/wordpress/wp-config-sample.php /var/www/wordpress/wp-config.php

CMD ["/usr/bin/supervisord"]
```

完成后查看目录文件：

```
shiyanlou@iZ23zslbpciZ: ~/shiyanlouwordpress
$ ls
Dockerfile  nginx-config  supervisord.conf
```



docker build 执行创建，-t 参数指定镜像名称：

```
docker build -t shiyanlouwordpress:0.2 /home/shiyanlou/shiyanlouwordpress/
```



```

shiyanolou@iZ23zslbpciZ: ~/shiyanolouwordpress
$ docker build -t shiyanolouwordpress:0.2 /home/shiyanolou/shiyanolouwordpress/
Sending build context to Docker daemon 5.632 kB
Step 1 : FROM ubuntu:latest
--> 14b59d36bae0
Step 2 : MAINTAINER shiyanolou@shiyanolou.com
--> Using cache
--> aa24f92a5748
Step 3 : RUN echo "deb http://mirrors.aliyuncs.com/ubuntu/ trusty main universe"
> /etc/apt/sources.list
--> Using cache
--> 129b1e62640e
Step 4 : RUN apt-get -yqq update
--> Using cache
--> 320cf65cb4cd
Step 5 : RUN apt-get -yqq install nginx supervisor wget php5-fpm php5-mysql
--> Using cache

```

docker images 查看创建的新镜像已经出现在了镜像列表中：

```

shiyanolou@iZ23zslbpciZ: ~/shiyanolouwordpress
$ docker images

```

REPOSITORY	TAG	IMAGE ID	CREATED
shiyanolouwordpress	0.2	4de22dcf9c72	17 minutes ago
ubuntu	latest	14b59d36bae0	2 weeks ago
redis	latest	fb46ec1d66e0	3 weeks ago
registry	latest	bca04f698ba8	6 weeks ago

SIZE  
460.8 MB  
188 MB  
151.3 MB  
422.9 MB

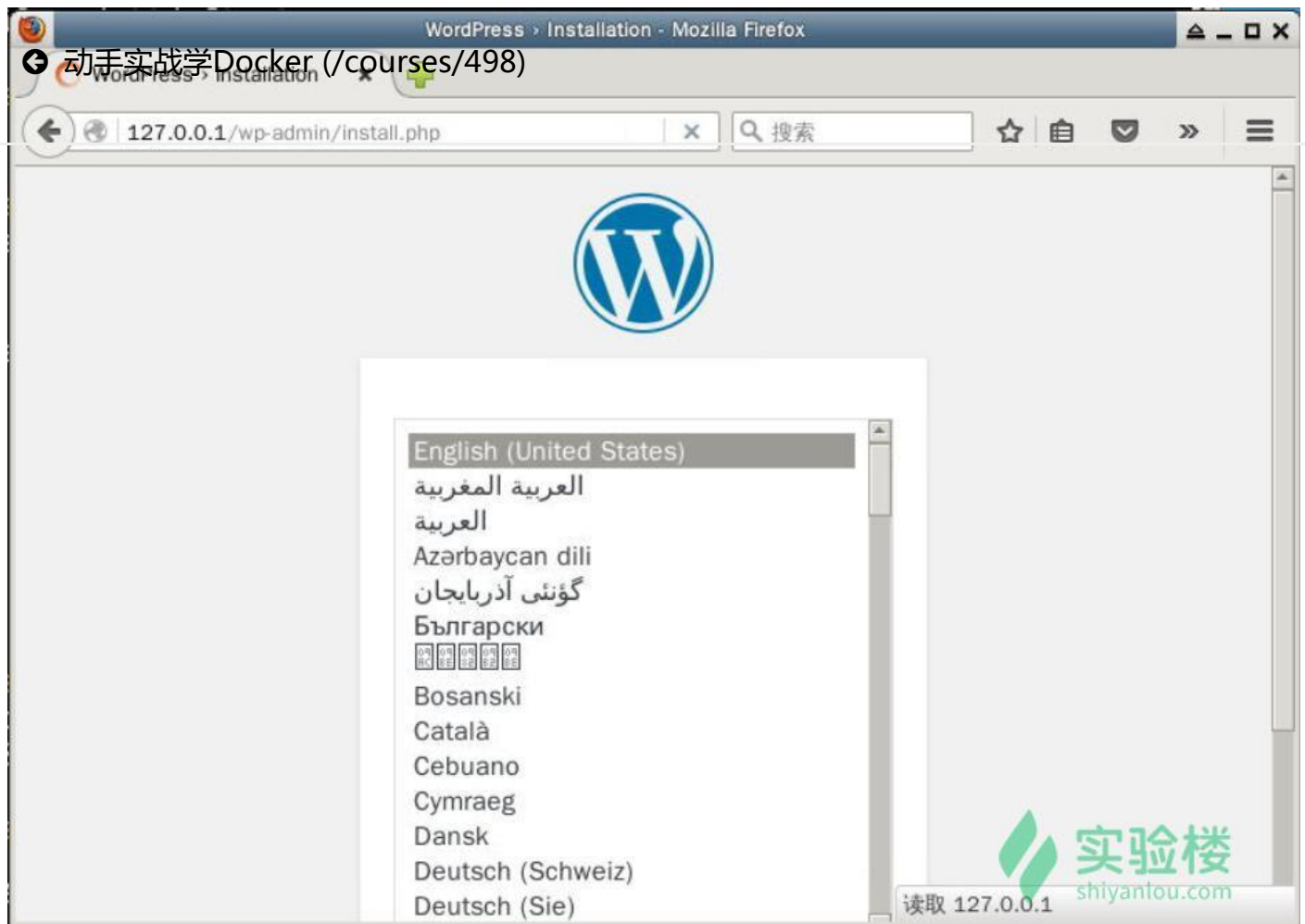
由该镜像创建新的容器 wordpress，并映射本地的 80 端口到容器的 80 端口：

```
$ docker run -d -p 80:80 --name wordpress shiyanolouwordpress:0.2
```

注意：一般出现问题都是配置的问题，注意检查配置是否有拼写错误。如果提示是 80 端口被占用，可能是我们本地的 nginx 占用了端口，使用 `sudo service nginx stop` 关闭即可。

使用 `docker container ls` 可以看到创建出来的容器。

最后打开桌面上的 firefox 浏览器，输入本地地址访问 `127.0.0.1/wp-admin/install.php`，看到我们的 Wordpress 网站安装配置界面，由于默认会连接 google 的文件，所以打开会比较慢：



## 9. 总结

本节实验中我们学习了以下内容：

1. Wordpress 的安装及配置
2. Dockerfile 的编写
3. 从 Dockerfile 构建镜像

请务必保证自己能够动手完成整个实验，只看文字很简单，真正操作的时候会遇到各种各样的问题，解决问题的过程才是收获的过程。

*\*本课程内容，由作者授权实验楼发布，未经允许，禁止转载、下载及非法传播。*

上一节：Docker运行MongoDB及Redis (/courses/498/labs/1709/document)

下一节：搭建自己的Docker Registry (/courses/498/labs/1711/document)