Docker入门基础之应用实战

当我们掌握了Docker镜像和容器的基本用法后，我们现在能做些什么事情呢？现在我们就来看看使用Docker容器如何安装常见的软件，然后运行一个动态网站。

下面我们来学习：

- 1、安装Nginx
- 2、安装PHP
- 3、使用MySQL服务
- 4、运行wordpress博客

安装Nginx

运行一个Alpine的容器，选择Alpine作为系统基础镜像是因为Alpine轻巧的体积，基础镜像只有5.53MB，相比ubuntu镜像的88.9MB要小十几倍。

```
root@ubuntu:~# docker run -it -p 8080:80 alpine sh
```

安装nginx

```
apk add nginx
```

修改nginx配置

```
vi /etc/nginx/conf.d/default.conf
```

default.conf内容如下：

```
server {
        listen 80 default_server;

        root /home/www;
        index index.php index.html;
}
```

创建Hello World

```
mkdir /home/www && echo "Hello World" > /home/www/index.html
```

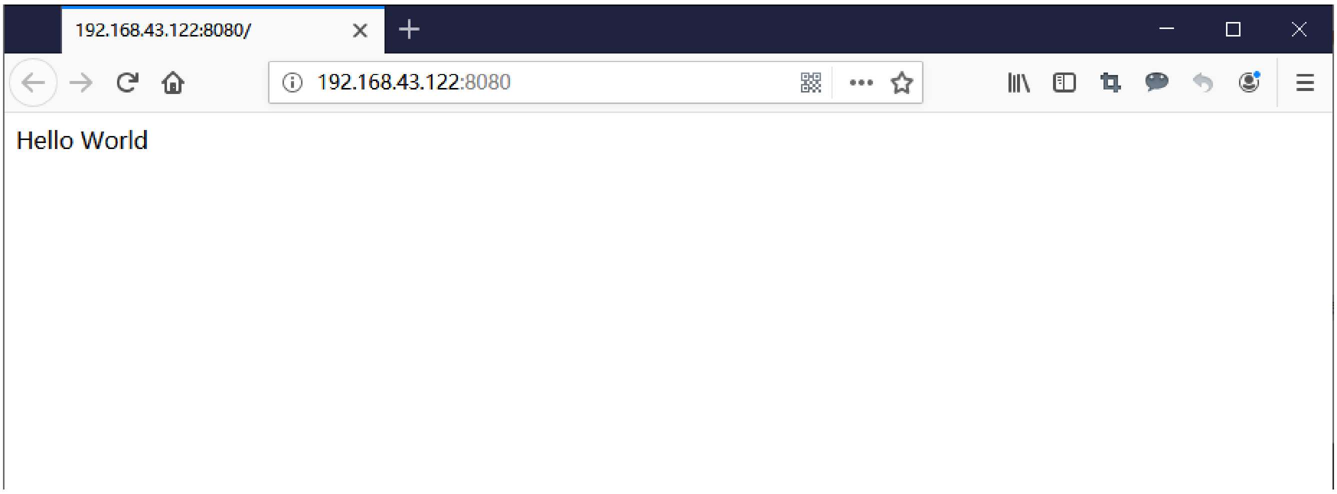
创建/run/nginx目录

```
mkdir /run/nginx
```

启动nginx

```
nginx
```

在浏览器中访问http://192.168.43.122:8080



Hello World

nginx安装成功，WEB服务访问正常！

安装PHP

现在我们来安装PHP，方法还是一样，使用 `apk add` 命令来安装php7，php-fpm以及相关扩展。

```
apk add --no-cache php7 php7-fpm php7-ftp php7-pdo php7-mysqli php7-simplexml php7-xmlwriter php7-zlib php7-imagick php7-memcached php7-sockets php7-mcrypt php7-zip php7-pgsql php7-pdo_odbc php7-odbc php7-curl php7-iconv php7-xml php7-json php7-gd php7-session php7-opcache php7-pdo_sqlite php7-mbstring php7-common php7-pdo_mysql
```

以上顺带安装了很多php扩展，可根据实际需求增减。

现在启动php-fpm7

```
php-fpm7
```

修改nginx配置，添加php支持

```
vi /etc/nginx/conf.d/default.conf
```

内容如下

```
server {
        listen 80 default_server;

        root /home/www;
        index index.php index.html;

        location ~ \.php$ {
            try_files $uri =404;
```

```
                fastcgi_split_path_info ^(.+\.php)(/.+)$;
                fastcgi_pass 127.0.0.1:9000;
                fastcgi_index index.php;
                include fastcgi_params;
                fastcgi_param SCRIPT_FILENAME /home/www$fastcgi_script_name;
                fastcgi_param PATH_INFO $fastcgi_path_info;
        }
}
```

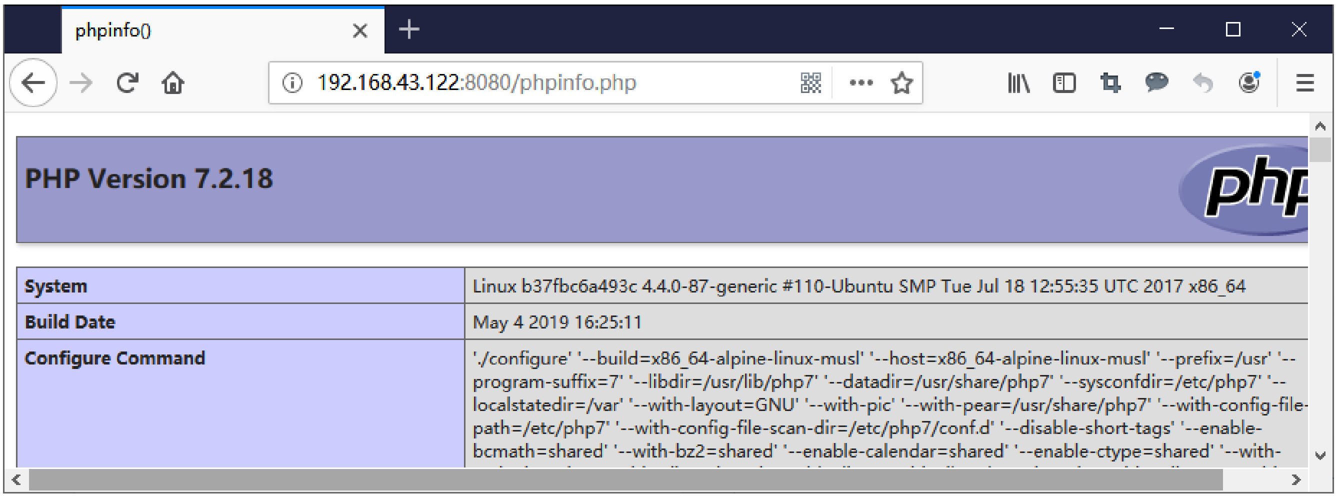
在网站目录中创建phpinfo.php

```
echo "<?php phpinfo(); ?>" > /home/www/phpinfo.php
```

重启nginx

```
nginx -s reload
```

在浏览器中访问http://192.168.43.122:8080/phpinfo.php



PHP Version 7.2.18

System	Linux b37fbc6a493c 4.4.0-87-generic #110-Ubuntu SMP Tue Jul 18 12:55:35 UTC 2017 x86_64
Build Date	May 4 2019 16:25:11
Configure Command	'./configure' '--build=x86_64-alpine-linux-musl' '--host=x86_64-alpine-linux-musl' '--prefix=/usr' '--program-suffix=7' '--libdir=/usr/lib/php7' '--datadir=/usr/share/php7' '--sysconfdir=/etc/php7' '--localstatedir=/var' '--with-layout=GNU' '--with-pic' '--with-pear=/usr/share/php7' '--with-config-file-path=/etc/php7' '--with-config-file-scan-dir=/etc/php7/conf.d' '--disable-short-tags' '--enable-bcmath=shared' '--with-bz2=shared' '--enable-calendar=shared' '--enable-ctype=shared' '--with-

PHP安装成功，WEB服务访问正常！

使用MySQL服务

我们不在Alpine里面安装MySQL，为了使用效率，这里我们直接使用官方镜像即可，新开一个ssh终端连接宿主机，在home目录下新建database目录

```
mkdir /home/database
```

使用官方mysql5.7镜像运行一个容器，root密码设置为123456，映射3306端口，并将宿主机/home/database目录挂载到容器/var/lib/mysql目录

```
docker run -dit -e MYSQL_ROOT_PASSWORD=123456 -p 3306:3306 -v
/home/database:/var/lib/mysql mysql:5.7
```

`docker ps`查看容器

上图中看到，我们现在运行了2个容器，1个是安装了nginx和php的Alpine容器，1个是mysql5.7的容器

进入mysql容器

```
docker exec -it c8c85af64822 sh
```

登录并创建wordpress数据库

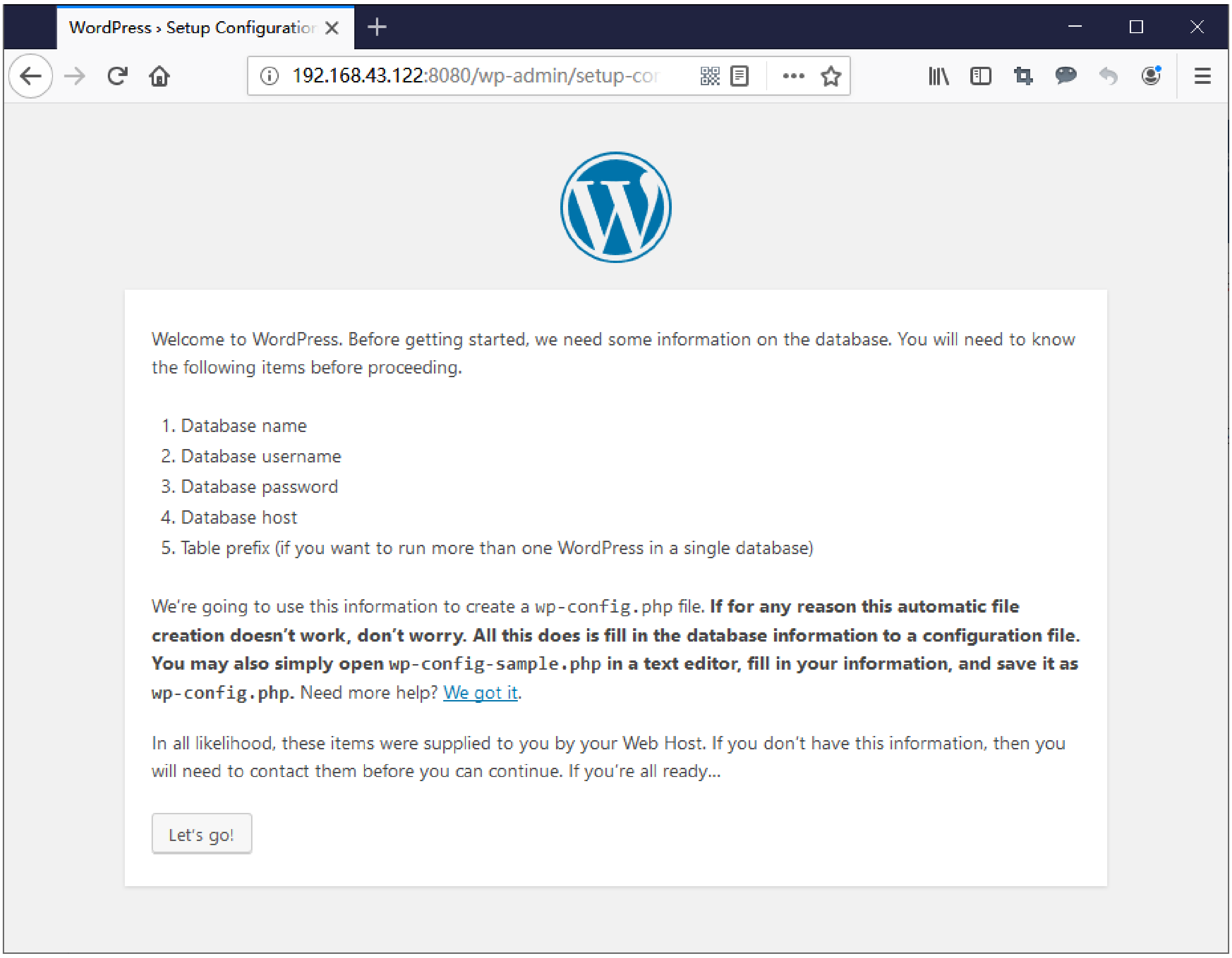
```
#用root用户登录
mysql -u root -p
#创建wordpress数据库
mysql> CREATE DATABASE wordpress;
```

运行wordpress博客

php网站环境和mysql数据库都准备好了，现在我们回到Alpine的ssh终端上，在容器内下载一个wordpress网站程序。

```
#进入网站目录
cd /home/www/
#下载wordpress
wget https://wordpress.org/latest.tar.gz
#解压
tar zxvf latest.tar.gz
#将wordpress目录下的文件复制到www目录下
cp -ri /home/www/wordpress/* /home/www/
#设置777权限
chmod -R 777 /home/www/
```

在浏览器中访问http://192.168.43.122:8080，可以看到wordpres页面


WordPress › Setup Configuration
192.168.43.122:8080/wp-admin/setup-co

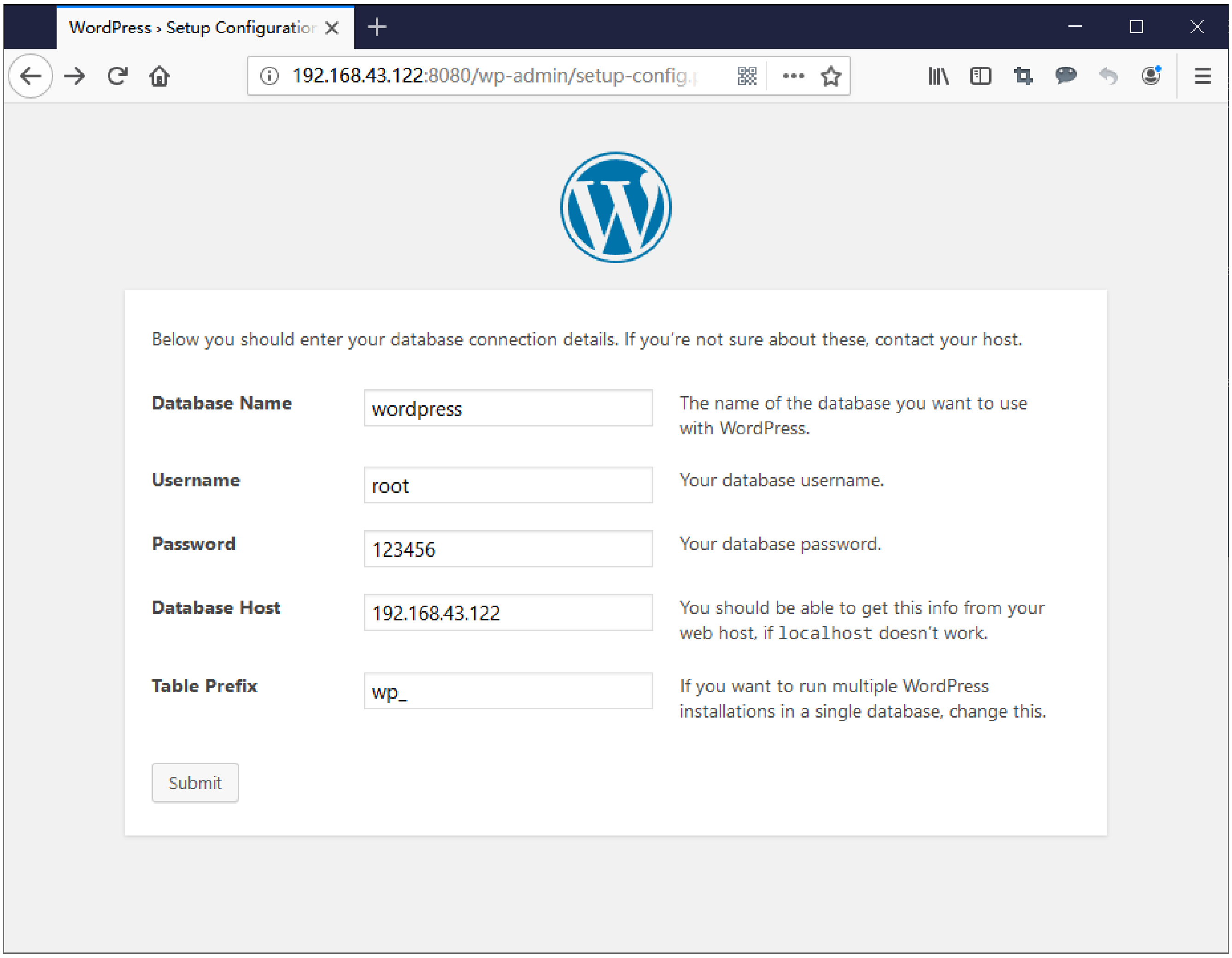
Welcome to WordPress. Before getting started, we need some information on the database. You will need to know the following items before proceeding.

1. Database name
2. Database username
3. Database password
4. Database host
5. Table prefix (if you want to run more than one WordPress in a single database)

We're going to use this information to create a `wp-config.php` file. **If for any reason this automatic file creation doesn't work, don't worry. All this does is fill in the database information to a configuration file. You may also simply open `wp-config-sample.php` in a text editor, fill in your information, and save it as `wp-config.php`.** Need more help? We got it.

In all likelihood, these items were supplied to you by your Web Host. If you don't have this information, then you will need to contact them before you can continue. If you're all ready…

填写数据库信息

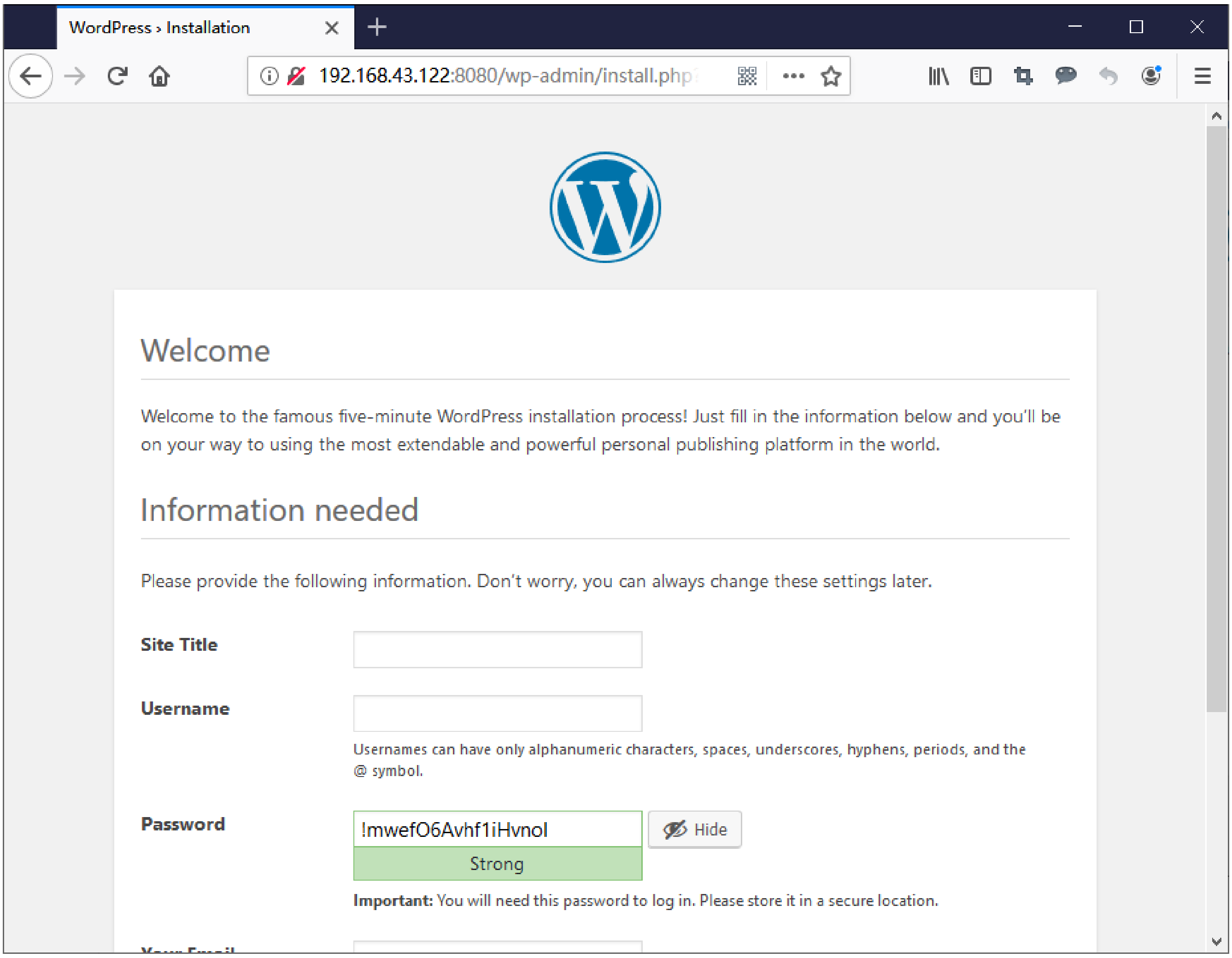


WordPress › Setup Configuratior
192.168.43.122:8080/wp-admin/setup-config.p

Below you should enter your database connection details. If you're not sure about these, contact your host.

Database Name	wordpress	The name of the database you want to use with WordPress.
Username	root	Your database username.
Password	123456	Your database password.
Database Host	192.168.43.122	You should be able to get this info from your web host, if `localhost` doesn't work.
Table Prefix	wp_	If you want to run multiple WordPress installations in a single database, change this.

安装完成，wordpress正常使用!



Welcome

Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

Information needed

Please provide the following information. Don't worry, you can always change these settings later.

Site Title

Username

Usernames can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.

Password !mwefO6Avhf1iHvnol

Strong

Important: You will need this password to log in. Please store it in a secure location.