

Docker 实战案例

案例一：使用官方 nginx 镜像创建 Web 应用

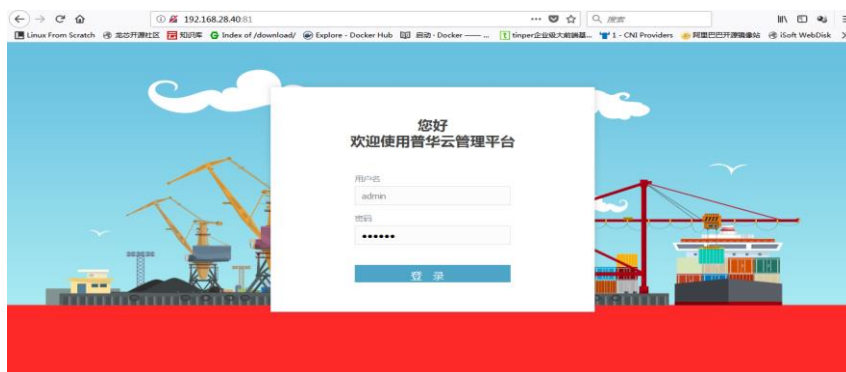
- (1) 新建工作目录: `mkdir docker_demo`
- (2) 准备官网基础镜像 `nginx`
使用命令: `docker pull nginx;`
- (3) 准备自己的 Web 程序, 并打包成 `.tar.gz` 或者 `gzip` 格式, 这里以 `rancher-web.tar.gz` 为例
- (4) 新建 Dockerfile 文件, `vim Dockerfile`, 内容如下所示:

```
#设置基础镜像
FROM nginx
#将自己的 web 程序添加到 nginx 目录中
ADD rancher-web.tar.gz /usr/share/nginx/html/
#端口映射
EXPOSE 80
```

- (5) 执行 `docker build` 生成镜像
使用命令: `docker build -t mynginx:v1 .`
- (6) 查看是否生成: `docker images`

```
[root@node1 docker_demo]# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
mynginx              v1                  7b34d87b4ebc      19 minutes ago    110MB
nginx                latest              27a188018e18      11 hours ago      109MB
```

- (7) 使用新创建的镜像创建容器
使用命令: `docker run -d -p 81:80 mynginx:v1`
- (8) 浏览器打开即可: <http://localhost:81>, 如下图所示。



案例二：使用 centos:v7.6 构建 Apache 镜像

(1) 新建 Dockerfile 文件，内容如下：

```
#基础镜像
FROM centos:v7.6
#启动 apache 服务
RUN yum -y install httpd httpd-devel httpd-server
RUN /usr/sbin/httpd
#复制服务启动脚本并设置权限
ADD run.sh /usr/sbin/run.sh
RUN chmod 755 /usr/sbin/run.sh
#开放 80 端口
EXPOSE 80
CMD ["/usr/sbin/run.sh"]
```

(2) 新建 run.sh 文件，内容如下

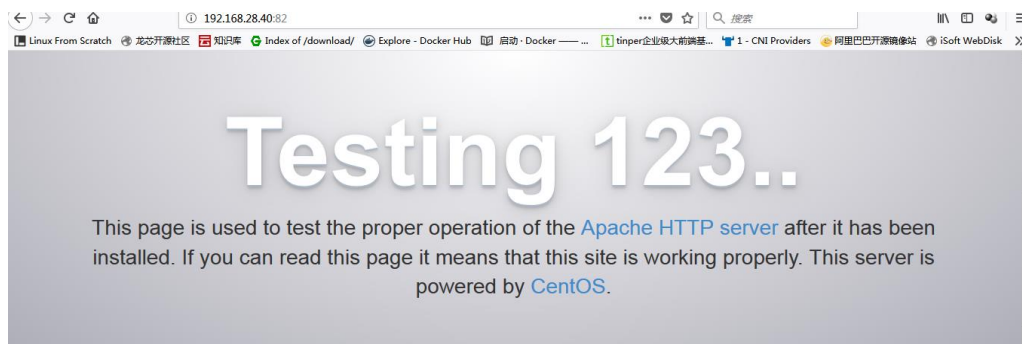
```
#!/bin/bash
/usr/sbin/httpd -D FOREGROUND
```

(3) 执行 docker build 生成镜像

使用命令：`Docker build -t myapache .`

(4) 使用 myapache 镜像创建容器：`docker run -d -p 82:80 myapache:v1`

(5) 浏览器打开即可，如图所示：



案例三：使用官网 Mariadb 构建数据库服务

(1) 拉取 Mariadb 镜像，拉取前可通过命令 `docker search mariadb` 进行查询。

```
docker search mariadb

NAME          DESCRIPTION          STARS   OFFICIAL   AUTOMATED
mariadb       MariaDB is a community-developed fork of MySQL... 2721    [OK]      [OK]
```

镜像拉取：`docker pull mariadb`

(2) 检查是否拉取成功：`docker images`

```
[root@node1 mariadb]# docker images
REPOSITORY    TAG        IMAGE ID      CREATED      SIZE
myapache      v1         c55161ade00b 4 days ago  318MB
mynginx       v1         7b34d87b4ebc 4 days ago  110MB
nginx         latest    27a188018e18 5 days ago  109MB
mariadb       latest    97df12fa9319 2 weeks ago 369MB
centos        v7.6      f1cb7c7d58b7 5 weeks ago 202MB
[root@node1 mariadb]#
```

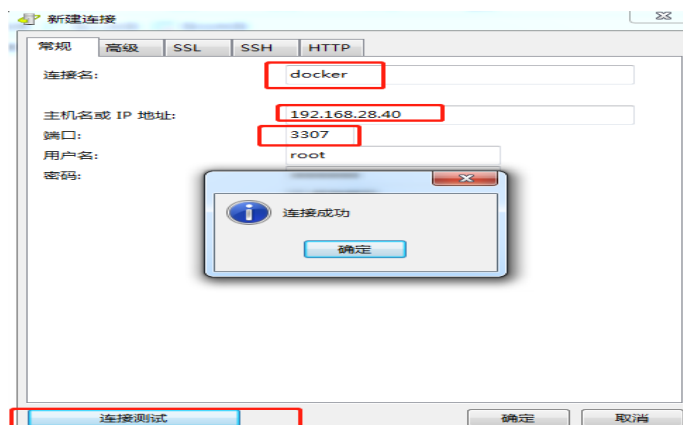
(3) 创建并启动一个 Mariadb 容器

使用命令：`docker run -d -e MYSQL_ROOT_PASSWORD=123456 -p 3307:3306 mariadb:latest`

- `- name`: 给新创建的容器命名，此处命名为默认
- `-e`: 配置信息，此处配置 mysql 的 root 用户的登陆密码
- `-p`: 端口映射，此处映射主机 3307 端口到容器的 3306 端口
- `-d`: 成功启动容器后输出容器的完整 ID
- 最后一个 mysql 指的是 mysql 镜像名字

(4) 测试连接，这里我在 windows 使用 navicat 远程连接，连接 MySQL 前需要防火墙开放端口或者关闭防火墙。开放端口：`$ sudo firewall-cmd --add-port=3307/tcp` 关闭防火墙：`$ sudo systemctl`

`stop firewalld`



案例四：使用 centos7 基础镜像构建 Apache PHP 服务

- (1) 拉取基础镜像 centos7, 拉取命令: `docker pull centos7`;
- (2) 编写 dockerfile 文件, 如下内容:

```
FROM centos:latest

RUN yum -y install httpd httpd-devel

RUN yum -y install php php-mysql php-gd php-imagick php-ldap php-odbc php-pear php-xml php-xmlrpc

EXPOSE 80

ADD info.php /var/www/html/info.php

RUN chmod 755 /var/www/html/info.php

ADD run.sh /root/run.sh

RUN chmod 755 /root/run.sh

CMD ["/root/run.sh"]
```

- (3) 编写测试 php 环境页面 `info.php`, 内容如下:

```
<?php
    phpinfo();
?>
```

- (4) 编写运行脚本 `run.sh`, 内容如下:

```
#!/bin/bash

/usr/sbin/httpd -D FOREGROUND
```

- (5) 构建镜像: `docker build -t myapachePHP:v1`
- (6) 使用镜像创建容器: `docker run -d -p 8181:80 myapachePHP:v1`, 运行结果

如图所示:



PHP Version 5.4.16	
System	Linux f749564bcc7 4.20.0-1.el7.elrepo.x86_64 #1 SMP Sun Dec 23 20:11:51 EST 2018 x86_64
Build Date	Oct 30 2018 19:31:42
Server API	Apache 2.0 Handler
Virtual Directory	disabled