## **编写 DockerFile**

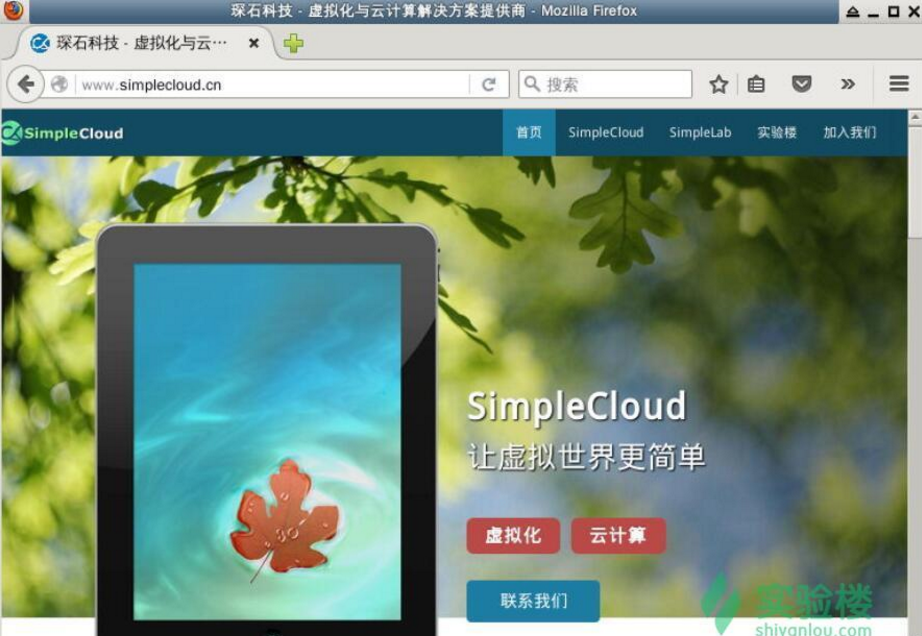
## **本节内容简介**

在前面的实验中我们多次用到的 Dockerfile，在本实验里我们将通过完成一个实例来学习Dockerfile的编写。

本节中，我们需要依次完成下面几项任务：

1. Dockerfile 基本框架
2. Dockerfile 编写常用命令
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Web应用，该web应用为apache托管的一个静态页面网站，换句话说，我们写一个Dockerfile，用来创建一个实验楼公司的网站应用，就是[http://www.simplecloud.cn](http://www.simplecloud.cn/" \t "http://www.cnblogs.com/liuyansheng/p/_blank)这个站点。这个站点是纯静态的页面，我们也可以直接下载得到。



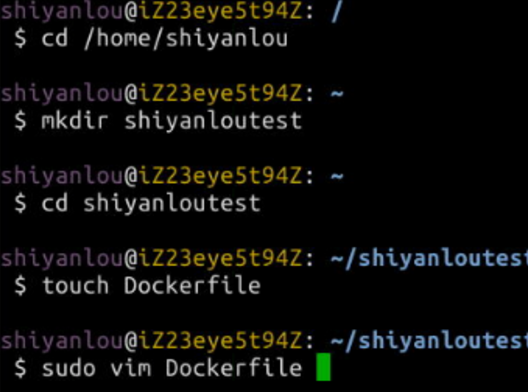
**一、实验准备**

1、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/shiyanloumkdir shiyanloutest

cd shiyanloutesttouch Dockerfile



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

**二、Dockerfile 基本框架**

Dockerfile一般包含下面几个部分：

1. 基础镜像：以哪个镜像作为基础进行制作，用法是FROM 基础镜像名称
2. 维护者信息：需要写下该Dockerfile编写人的姓名或邮箱，用法是MANITAINER 名字/邮箱
3. 镜像操作命令：对基础镜像要进行的改造命令，比如安装新的软件，进行哪些特殊配置等，常见的是RUN 命令
4. 容器启动命令：当基于该镜像的容器启动时需要执行哪些命令，常见的是CMD 命令或ENTRYPOINT

在本节实验中，我们依次先把这四项信息填入文档。Dockerfile中的#标志后面为注释，可以不用写，另外实验楼的环境不支持中文输入，比较可惜。

依次输入下面的基本框架内容：

[IMG_258](http://www.cnblogs.com/liuyansheng/p/javascript:void(0);)

# 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 apt-get -yqq update && apt-get install -yqq apache2 && apt-get clean

# 容器启动命令

CMD ["/usr/sbin/apache2ctl", "-D", "FOREGROUND"]

[IMG_259](http://www.cnblogs.com/liuyansheng/p/javascript:void(0);)

上面的Dockerfile非常简单，创建了一个apache的镜像。包含了最基本的四项信息。

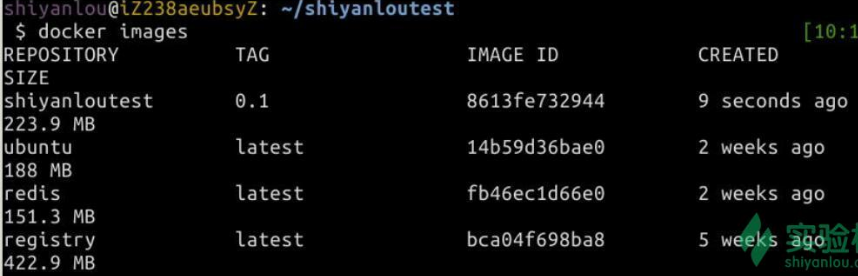
其中FROM指定基础镜像，如果镜像名称中没有制定TAG，默认为latest。RUN命令默认使用/bin/sh Shell执行，默认为root权限。如果命令过长需要换行，需要在行末尾加\。CMD命令也是默认在/bin/sh中执行，并且默认只能有一条，如果是多条CMD命令则只有最后一条执行。用户也可以在docker run命令创建容器时指定新的CMD命令来覆盖Dockerfile里的CMD。

这个Dockerfile已经可以使用docker build创建新镜像了，先构建一个版本shiyanloutest:0.1：

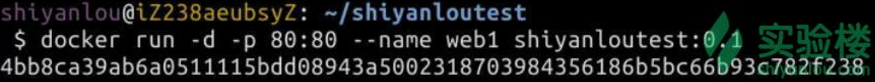
cd /home/shiyanlou/shiyanloutest

docker build -t shiyanloutest:0.1 .

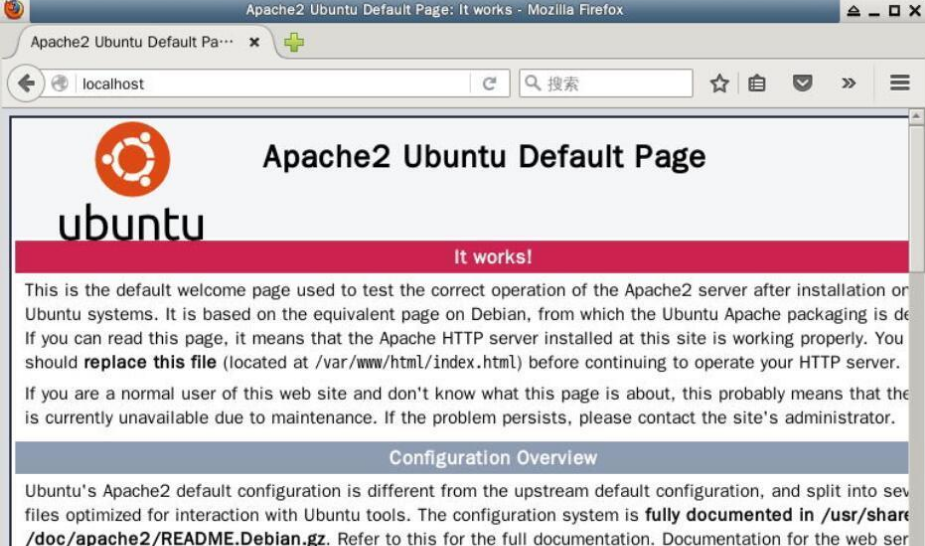
构建需要安装apache2，会花几分钟，最后查看新创建的镜像：



使用该镜像创建容器web1，将容器中的端口80映射到本地80端口：



使用实验环境桌面上的firefox浏览器打开localhost进行测试，查看是否apache已运行：



**三、Dockerfile 编写常用命令**

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。后续的扩展操作都需要放置在Dockerfile的镜像操作部分。其中部分命令在本实验中并不会用到，但需要有所了解。

1、指定容器运行的用户

该用户将作为后续的RUN命令执行的用户。这个命令本实验不需要，但在一些需要指定用户来运行的应用部署时非常关键，比如提供hadoop服务的容器通常会使用hadoop用户来启动服务。

命令使用方式，例如使用shiyanlou用户来执行后续命令：

USER shiyanlou

2、指定后续命令的执行目录

由于我们需要运行的是一个静态网站，将启动后的工作目录切换到/var/www/html目录：

WORKDIR /var/www/html

3、对外连接端口号

由于内部服务会启动Web服务，我们需要把对应的80端口暴露出来，可以提供给容器间互联使用，可以使用EXPOSE命令。

在镜像操作部分增加下面一句：

EXPOSE 80

4、设置容器主机名

ENV命令能够对容器内的环境变量进行设置，我们使用该命令设置由该镜像创建的容器的主机名为shiyanloutest，向Dockerfile中增加下面一句：

ENV HOSTNAME shiyanloutest

5、向镜像中增加文件

向镜像中添加文件有两种命令：COPY 和ADD。

COPY simplecloudsite /var/www/html

ADD 命令支持添加本地的tar压缩包到容器中指定目录，压缩包会被自动解压为目录，也可以自动下载URL并拷贝到镜像，例如：

ADD html.tar /var/www

ADD http://www.shiyanlou.com/html.tar /var/www

根据实验需求，我们把需要的一个网站放到镜像里，需要把一个tar包添加到apache的/var/www目录下，因此选择使用 ADD命令：

ADD html.tar /var/www

**四、CMD 与 ENTRYPOINT**

ENTRYPOINT 容器启动后执行的命令，让容器执行表现的像一个可执行程序一样，与CMD的区别是不可以被docker run覆盖，会把docker run后面的参数当作传递给ENTRYPOINT指令的参数。Dockerfile中只能指定一个ENTRYPOINT，如果指定了很多，只有最后一个有效。docker run命令的-entrypoint参数可以把指定的参数继续传递给ENTRYPOINT。

在本实验中两种方式都可以选择。

**五、挂载数据卷**

将apache访问的日志数据存储到宿主机可以访问的数据卷中：

VOLUME ["/var/log/apche2"]

**六、设置容器内的环境变量**

使用ENV设置一些apache启动的环境变量：

ENV APACHE\_RUN\_USER www-data

ENV APACHE\_RUN\_GROUP www-data

ENV APACHE\_LOG\_DIR /var/log/apche2

ENV APACHE\_PID\_FILE /var/run/apache2.pid

ENV APACHE\_RUN\_DIR /var/run/apache2

ENV APACHE\_LOCK\_DIR /var/lock/apche2

**七、使用**Supervisord

CMD如果只有一个命令，那如果我们需要运行多个服务怎么办呢？最好的办法是分别在不同的容器中运行，通过link进行连接，比如先前实验中用到的web，app，db容器。如果一定要在一个容器中运行多个服务可以考虑用Supervisord来进行进程管理，方式就是将多个启动命令放入到一个启动脚本中。

首先安装Supervisord，添加下面内容到Dockerfile:

RUN apt-get install -yqq supervisor

RUN mkdir -p /var/log/supervisor

拷贝配置文件到指定的目录：

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

其中supervisord.conf文件需要放在/home/shiyanlou/shiyanloutest下，文件内容如下：

[supervisord]

nodaemon=true

[program:apache2]

command=/bin/bash -c "source /etc/apache2/envvars && exec /usr/sbin/apache2ctl -D FOREGROUND"

如果有多个服务需要启动可以在文件后继续添加[program:xxx]，比如如果有ssh服务，可以增加[program:ssh]。

修改CMD命令，启动Supervisord：

CMD ["/usr/bin/supervisord"]

**八、从 Dockerfile 创建镜像**

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

[IMG_263](http://www.cnblogs.com/liuyansheng/p/javascript:void(0);)

# Version 0.2

# 基础镜像

FROM ubuntu:latest

# 维护者信息

MAINTAINER shiyanlou@shiyanlou.com

# 镜像操作命令

RUN apt-get -yqq update && apt-get install -yqq apache2 && apt-get clean

RUN apt-get install -yqq supervisor

RUN mkdir -p /var/log/supervisor

VOLUME ["/var/log/apche2"]

ADD html.tar /var/www

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

WORKDIR /var/www/html

ENV HOSTNAME shiyanloutest

ENV APACHE\_RUN\_USER www-data

ENV APACHE\_RUN\_GROUP www-data

ENV APACHE\_LOG\_DIR /var/log/apche2

ENV APACHE\_PID\_FILE /var/run/apache2.pid

ENV APACHE\_RUN\_DIR /var/run/apache2

ENV APACHE\_LOCK\_DIR /var/lock/apche2

EXPOSE 80

# 容器启动命令

CMD ["/usr/bin/supervisord"]

[IMG_264](http://www.cnblogs.com/liuyansheng/p/javascript:void(0);)

同时在/home/shiyanlou/shiyanloutest目录下，添加supervisord.conf文件：

[supervisord]

nodaemon=true

[program:apache2]

command=/bin/bash -c "source /etc/apache2/envvars && exec /usr/sbin/apache2ctl -D FOREGROUND"

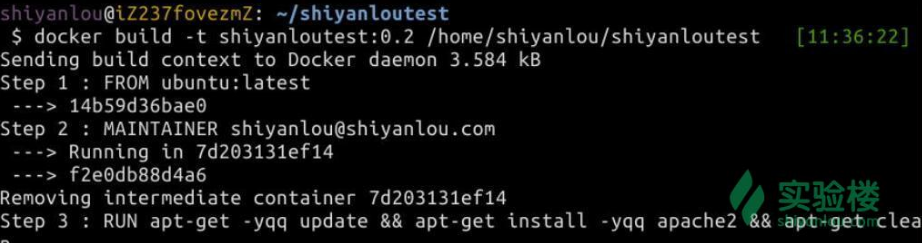
并下载静态页面文件压缩包：

cd /home/shiyanlou/shiyanloutestwget http://labfile.oss.aliyuncs.com/courses/498/html.tar

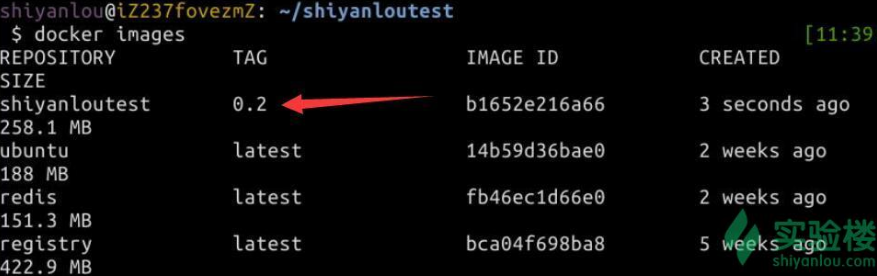
将http://simplecloud.cn网站的页面tar包下载到/home/shiyanlou/shiyanloutest目录：

docker build 执行创建，-t参数指定镜像名称：

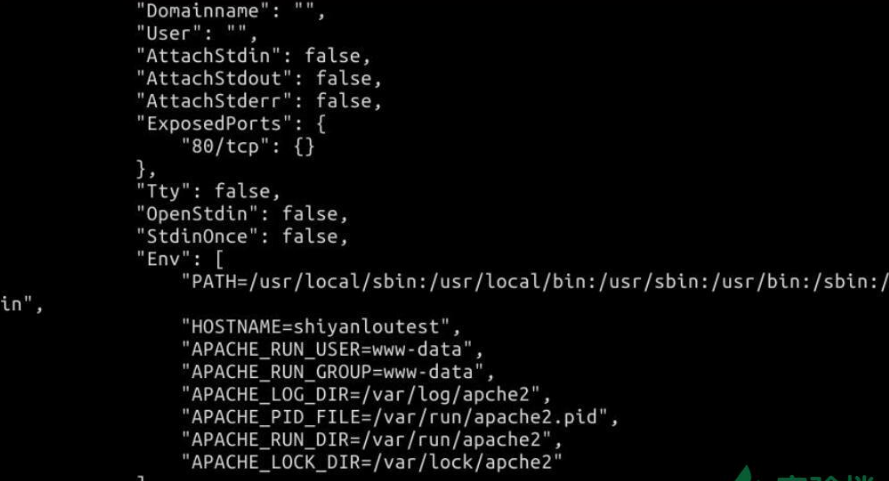
docker build -t shiyanloutest:0.2 /home/shiyanlou/shiyanloutest/



docker images 查看创建的新镜像已经出现在了镜像列表中：



docker inspect shiyanloutest:0.2 查看该镜像的详细信息：



由该镜像创建新的容器web2，并映射本地的80端口到容器的80端口：

docker run -d -p 80:80 --name web2 shiyanloutest:0.2

最后打开桌面上的firefox浏览器，输入本地地址访问127.0.0.1，看到我们克隆的琛石科技的网站：



## **总结**

1. Dockerfile 基本框架
2. Dockerfile 编写常用命令
3. 从 Dockerfile 构建镜像