## 1、启动容器

容器是在镜像的基础上来运行的，一旦容器启动了，我们就可以登录到容器中，安装自己所需的软件或应用程序。

使用以下命令即可启动容器：

# docker run -i -t -v /root/software/:/mnt/software/ 83ee /bin/bash

命令包含以下三个部分：

docker run <相关参数> <镜像 ID> <初始命令>

其中，相关参数包括：

* -i：表示以“交互模式”运行容器
* -t：表示容器启动后会进入其命令行
* -v：表示需要将本地哪个目录挂载到容器中，格式：-v <宿主机目录>:<容器目录>

本例中，所有安装程序都放在了宿主机的/root/software/目录下，现在需要将其挂载到容器的/mnt/software/目录下。

# pwd

/root/software

# ls

apache-tomcat-7.0.67.tar.gz  jdk1.7.0\_79.tar.gz

初始命令表示一旦容器启动，需要运行的命令，此时使用“/bin/bash”，表示启动后直接进入bash shell。

## 2、 安装软件

为了搭建 Java Web 运行环境，需要安装 JDK 与 Tomcat，下面的过程均在容器内部进行。本例中，选择/opt/目录作为安装目录，首先需要通过cd /opt/命令进入该目录。

### 2.1 安装JDK

首先，解压 JDK 程序包：

# tar -zxf /mnt/software/jdk1.7.0\_79.tar.gz -C .

然后，移动 JDK 目录：

# mv jdk1.7.0\_79/ /opt/jdk/

### 2.2 安装Tomcat

首先，解压Tomcat程序包：

# tar -zxf /mnt/software/apache-tomcat-7.0.67.tar.gz -C .

然后，移动Tomcat目录：

# mv apache-tomcat-7.0.67/ /opt/tomcat/

### 2.3 编写运行脚本

编写一个运行脚本，当启动容器时，运行该脚本，启动 Tomcat。

首先，创建运行脚本：

# touch /root/run.sh

# vi /root/run.sh

然后，编辑脚本内容如下：

#!/bin/bash

export JAVA\_HOME=/opt/jdk/

export PATH=$JAVA\_HOME/bin:$PATH

sh /opt/tomcat/bin/catalina.sh run

最后，为运行脚本添加执行权限：

# chmod u+x /root/run.sh

## 3、 退出容器

当以上步骤全部完成后，可使用exit命令，退出容器。

随后，可使用如下命令查看正在运行的容器：

# docker ps

此时，应该看不到任何正在运行的程序，因为刚才已经使用exit命令退出的容器，此时容器处于停止状态，可使用如下命令查看所有容器：

# docker ps -a

CONTAINER ID  IMAGE  COMMAND     CREATED         STATUS

02bebc3f546a  83ee   "/bin/bash" 12 minutes ago  Exited (0) 7 seconds ago

记住以上CONTAINER ID（容器 ID），随后将通过该容器，创建一个可运行 Tomcat 镜像。

## 4、 创建Tomcat镜像

使用以下命令，根据某个“容器 ID”来创建一个新的“镜像”：

# docker commit 02be mytomcat:1.0

65c88ec597e04812ec3b06b7749578bebcae3aa3d735b565ed25db6818d9d7f3

# docker images

REPOSITORY        TAG       IMAGE ID      CREATED             VIRTUAL SIZE

mytomcat          1.0       65c88ec597e0  About a minute ago  514.4 MB

docker.io/centos  7.2.1511  83ee614b834e  9 weeks ago         194.6 MB

该容器的ID是02be，所创建的镜像名是“mytomcat:1.0”，随后可使用镜像来启动Tomcat容器。

## 5、 启动Tomcat容器

首先，新建/root/webapps/ROOT目录，并在该目录下创建一个index.html文件，文件内容如下：

<html>

<body>

<h2>Hello World!</h2>

</body>

</html>

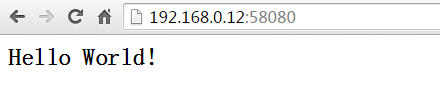
正如上面所描述的那样，可以通过“镜像名”或“镜像 ID”来启动容器，与上次启动容器不同的是，现在不再进入容器的命令行，而是直接启动容器内部的 Tomcat 服务。此时，需要使用以下命令：

# docker run -d -p 58080:8080 -v /root/webapps/:/opt/tomcat/webapps/ --name mytomcat\_1 mytomcat:1.0 /root/run.sh

其中，相关参数包括：

* -d：表示以“守护模式”执行/root/run.sh脚本，此时 Tomcat 控制台不会出现在输出终端上。
* -p：表示宿主机与容器的端口映射，此时将容器内部的 8080 端口映射为宿主机的 58080 端口，这样就向外界暴露了 58080 端口，可通过 Docker 网桥来访问容器内部的 8080 端口了。
* -v：表示需要将本地哪个目录挂载到容器中，格式：-v <宿主机目录>:<容器目录>
* --name：表示容器名称，用一个有意义的名称命名即可。

在浏览器中，输入宿主IP和端口号，即可访问 Tomcat：



## 6、 停止Tomcat容器

# docker ps -a

CONTAINER ID  IMAGE         COMMAND         CREATED         STATUS

f23598b6544d  mytomcat:1.0  "/root/run.sh"  6 minutes ago   Up 6 minutes

# docker stop f235

## 7、 移除容器

# docker ps -a

CONTAINER ID  IMAGE         COMMAND         CREATED        STATUS

f23598b6544d  mytomcat:1.0  "/root/run.sh"  8 minutes ago  Exited (137)

# docker rm f235

f235

# docker ps -a

CONTAINER ID  IMAGE         COMMAND         CREATED        STATUS

## 8、 移除镜像

# docker images

REPOSITORY          TAG        IMAGE ID       CREATED         VIRTUAL SIZE

mytomcat            1.0        65c88ec597e0   31 minutes ago  514.4 MB

docker.io/centos    7.2.1511   83ee614b834e   9 weeks ago     194.6 MB

# docker rmi 65c8

Untagged: mytomcat:1.0

Deleted: 65c88ec597e04812ec3b06b7749578bebcae3aa3d735b565ed25db6818d9d7f3

# docker images

REPOSITORY          TAG        IMAGE ID       CREATED         VIRTUAL SIZE

docker.io/centos    7.2.1511   83ee614b834e   9 weeks ago     194.6 MB