**环境安装**

安装epel

yum -y install <http://dl.fedoraproject.org/pub/epel/6/x86_64/Packages/e/epel-release-6-8.noarch.rpm>

安装Docker

yum install docker-io

启动Docker

service docker start

搜索centos镜像

docker search centos

下载镜像

docker pull centos

查看本地所有的镜像：

docker images

当下载完成后，你应该会看到：

REPOSITORY TAG IMAGE ID CREATED SIZE

docker.io/centos latest 196e0ce0c9fb 6 weeks ago 196.6 MB

如果看到以上输出，说明你可以使用“docker.io/centos”这个镜像了，或将其称为仓库（Repository），该镜像有一个名为“latest”的标签（Tag）。此外还有一个名为“196e0ce0c9fb”的镜像ID（可能你所看到的镜像ID与此处的不一致，那是正常现象，因为这个数字是随机生成的）。此外，我们可以看到该镜像只有196.6MB，非常小巧，而不像虚拟机的镜像文件那样庞大。

现在镜像已经有了，我们下面就需要使用该镜像来启动容器。

只需使用以下命令即可启动容器：

docker run -i -t -v /root/software/:/mnt/software/ 196e0ce0c9fb /bin/bash

这条命令比较长，我们稍微分解以下，其实包含以下三个部分：

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>

假设我们的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。

需要说明的是，不一定要使用“镜像ID”，也可以使用“仓库名：标签名”，例如：docker.io/centos: latest

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”，表示什么也不做，只需进入命令行即可。

启动之前请将所需软件包预先放置在/root/software/目录下，

|  |
| --- |
| wget <http://download.oracle.com/otn-pub/java/jdk/8u151-b12/e758a0de34e24606bca991d704f6dcbf/jdk-8u151-linux-x64.tar.gz>  wget <http://mirrors.hust.edu.cn/apache/tomcat/tomcat-8/v8.5.23/bin/apache-tomcat-8.5.23.tar.gz> |

**安装相关软件**

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

安装JDK

首先，解压JDK程序包：

tar -zxf /mnt/software/jdk-8u151-linux-x64.tar.gz -C .

安装 Tomcat

首先，解压 Tomcat 程序包：

tar -zxf /mnt/software/apache-tomcat-8.5.23.tar.gz -C .

设置环境变量

首先，编辑.bashrc文件

vi ~/.bashrc

然后，在该文件末尾添加如下配置：

export JAVA\_HOME=/opt/jdk1.8.0\_151

export PATH=$PATH:$JAVA\_HOME

最后，需要使用source命令，让环境变量生效：

source ~/.bashrc

编写运行脚本

我们需要编写一个运行脚本，当启动容器时，运行该脚本，启动 Tomcat，具体过程如下：

首先，创建运行脚本：

vi /root/run.sh

然后，编辑脚本内容如下：

#!/bin/bash

source ~/.bashrc

sh /opt/apache-tomcat-8.5.23/bin/catalina.sh run

注意：这里必须先加载环境变量，然后使用 Tomcat 的运行脚本来启动 Tomcat 服务。

最后，为运行脚本添加执行权限：

chmod u+x /root/run.sh

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，您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：

|  |
| --- |
| [root@iZ2zecs3yv99wgfq7ez1mjZ software]# docker ps -a  CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES  b927efd148bd 196e0ce0c9fb "/bin/bash" About an hour ago Exited (0) 4 seconds ago mad\_keller  4d6e3bda4e43 196e0ce0c9fb "/bin/bash" 2 hours ago Exited (0) 2 hours ago nostalgic\_snyder |

记住以上CONTAINER ID（容器 ID），随后我们将通过该容器，创建一个可运行 Java Web 的镜像。

创建 Java Web 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 b927efd148bd javaweb:0.1

该容器的 ID 是“57c312bbaad1”，所创建的镜像名是“javaweb:0.1”，随后可使用镜像来启动 Java Web 容器。

启动 Java Web 容器

有必要首先使用docker images命令，查看当前所有的镜像：

|  |
| --- |
| [root@iZ2zecs3yv99wgfq7ez1mjZ software]# docker images  REPOSITORY TAG IMAGE ID CREATED SIZE  javaweb 0.1 e11c69241e21 49 seconds ago 594.4 MB  docker.io/centos latest 196e0ce0c9fb 6 weeks ago 196.6 MB |

可见，此时已经看到了最新创建的镜像“javaweb:0.1”，其镜像 ID 是“e11c69241e21”。正如上面所描述的那样，我们可以通过“镜像名”或“镜像 ID”来启动容器，与上次启动容器不同的是，我们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 --name javaweb javaweb:0.1 /root/run.sh

稍作解释：

　　-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。  
　　--name：表示容器名称，用一个有意义的名称命名即可。  
关于 Docker 网桥的内容，需要补充说明一下。实际上 Docker 在宿主机与容器之间，搭建了一座网络通信的桥梁，我们可通过宿主机 IP 地址与端口号来映射容器内部的 IP 地址与端口号，

　　当运行以上命令后，会立即输出一长串“容器 ID”，我们可通过docker ps命令来查看当前正在运行的容器。

|  |
| --- |
| [root@iZ2zecs3yv99wgfq7ez1mjZ software]# docker ps  CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES  5e4fafd6b8c0 javaweb:0.1 "/root/run.sh" 3 minutes ago Up 3 minutes 0.0.0.0:58080->8080/tcp javaweb |

在浏览器中，输入以下地址，即可访问 Tomcat 首页：  
http://[ip]:58080/  
注意：这里使用的是宿主机的 IP 地址，与对外暴露的端口号 58080，它映射容器内部的端口号 8080。  
由于我测试的虚拟机采用nat映射连接，如果采用这种方法测试则需要重新将58080端口映射到宿主主机才能够访问。

以上操作较多，为避免混淆，这里做下解释。

1. 首先从网上下载centos的image
2. 然后利用这个空的image我们创建了一个容器，并在里面安装了JDK和Tomcat，为了做成一个共通性的容器，可以理解为模板。
3. 把刚才做的模板容器commit成一个新的image叫做javaweb，tag为0.1，方便以后生成其他容器使用
4. 利用javaweb创建一个新的容器进行使用，此容器里面已经有了tomcat和JDK，并且启动时自动运行了run.sh脚本。

如何进入停止的Docker容器

|  |
| --- |
| [r**oot@iZ2zecs3yv99wgfq7ez1mjZ software]#** docker ps -a  **CONTAINER ID IMAGE COMMAND CREATED STATUS**  PORTS NAMES  b927efd148bd 196e0ce0c9fb "/bin/bash" About an hour ago Exited (2) About an hour ago mad\_keller  4d6e3bda4e43 196e0ce0c9fb "/bin/bash" About an hour ago Exited (0) About an hour ago nostalgic\_snyder  [root@iZ2zecs3yv99wgfq7ez1mjZ software]# docker start b927efd148bd  b927efd148bd  [root@iZ2zecs3yv99wgfq7ez1mjZ software]# docker ps -a  CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES  b927efd148bd 196e0ce0c9fb "/bin/bash" About an hour ago Up 6 seconds mad\_keller  4d6e3bda4e43 196e0ce0c9fb "/bin/bash" About an hour ago Exited (0) About an hour ago nostalgic\_snyder  [root@iZ2zecs3yv99wgfq7ez1mjZ software]# docker attach b927efd148bd  [root@b927efd148bd /]# pwd  / |