**CentOS7安装Docker全程并启动**

原创 2017年03月14日 17:53:21

* 12044

一、安装docker  
    Docker CE 支持 CentOS 7.3 64-bit.  
    升级Docker

1. yum -y update

    设置 Docker CE 资源库:

1. sudo yum install -y yum-utils
2. sudo yum-config-manager \
3. --add-repo \
4. https://download.docker.com/linux/centos/docker-ce.repo
5. sudo yum makecache fast

  安装docker

1. sudo yum -y install docker-ce

  启动docker

1. sudo systemctl start docker

二、设置mirror      
    新版的 Docker 使用 /etc/docker/daemon.json 来配置 Daemon  
    在该配置文件中加入（没有该文件的话，请先创建一个）：  
    **{  
      "registry-mirrors": ["https://docker.mirrors.ustc.edu.cn"]  
    }**  
三、开放管理端口映射  
    管理端口在 /lib/systemd/system/docker.service 文件中  
    将其中第11行的 ExecStart=/usr/bin/dockerd 替换为：  
    **ExecStart=/usr/bin/dockerd -H tcp://0.0.0.0:2375 -H unix:///var/run/docker.sock -H tcp://0.0.0.0:7654**  
    （此处默认2375为主管理端口，unix:///var/run/docker.sock用于本地管理，7654是备用的端口）  
    将管理地址写入 /etc/profile

1. echo 'export DOCKER\_HOST=tcp://0.0.0.0:2375' **>>** /etc/profile

  使profile生效

1. source /etc/profile

四、启动docker

1. systemctl daemon-reload && service docker restart

五、测试docker

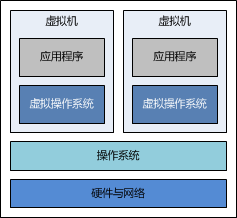
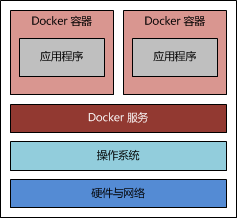
1. sudo docker run hello-world

若成功：

Hello from Docker!  
This message shows that your installation appears to be working correctly.

# Docker部署Tomcat

**1 Docker与虚拟机**

**2 搭建过程**

**2.1 准备宿主系统**

准备一个 CentOS 7操作系统，具体要求如下：

* 必须是 64 位操作系统
* 建议内核在 3.8 以上

通过以下命令查看您的 CentOS 内核：

|  |  |
| --- | --- |
| 1 | # uname -r |

**2.2 安装Docker**

|  |  |
| --- | --- |
| 1 | # yum install docker |

可使用以下命令，查看 Docker 是否安装成功：

|  |  |
| --- | --- |
| 1 | # docker version |

若输出了 Docker 的版本号，则说明安装成功了，可通过以下命令启动 Docker 服务：

|  |  |
| --- | --- |
| 1 | # systemctl start  docker.service |

一旦 Docker 服务启动完毕，就可以开始使用 Docker 了。

**2.3 下载镜像**

以 CentOS 为例，下载一个 CentOS 的镜像：

|  |  |
| --- | --- |
| 1 | # docker pull centos:7.2.1511 |

下载完成后，使用命令查看本地镜像列表：

|  |  |
| --- | --- |
| 1  2  3 | # docker images  REPOSITORY        TAG        IMAGE ID      CREATED      VIRTUAL SIZE  docker.io/centos  7.2.1511   83ee614b834e  9 weeks ago  194.6 MB |

**2.4 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：

|  |  |
| --- | --- |
| 1 | # docker run -i -t -v /root/software/:/mnt/software/ 83ee /bin/bash |

命令包含以下三个部分：

|  |  |
| --- | --- |
| 1 | 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。

|  |  |
| --- | --- |
| 1  2  3  4  5 | # 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。

**2.5 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。

**2.5.1 安装JDK**

首先，解压 JDK 程序包：

|  |  |
| --- | --- |
| 1 | # tar -zxf /mnt/software/jdk1.7.0\_79.tar.gz -C . |

然后，移动 JDK 目录：

|  |  |
| --- | --- |
| 1 | # mv jdk1.7.0\_79/ /opt/jdk/ |

**2.5.2 安装Tomcat**

首先，解压Tomcat程序包：

|  |  |
| --- | --- |
| 1 | # tar -zxf /mnt/software/apache-tomcat-7.0.67.tar.gz -C . |

然后，移动Tomcat目录：

|  |  |
| --- | --- |
| 1 | # mv apache-tomcat-7.0.67/ /opt/tomcat/ |

**2.5.3 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：

|  |  |
| --- | --- |
| 1  2  3 | # touch /root/run.sh    # vi /root/run.sh |

然后，编辑脚本内容如下：

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | #!/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：

|  |  |
| --- | --- |
| 1 | # chmod u+x /root/run.sh |

**2.6 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：

|  |  |
| --- | --- |
| 1 | # 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：

|  |  |
| --- | --- |
| 1  2  3 | # 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。

**2.7 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”：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | # 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。

**2.8 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：

|  |  |
| --- | --- |
| 1  2  3  4  5 | <html>  <body>  <h2>Hello World!</h2>  </body>  </html> |

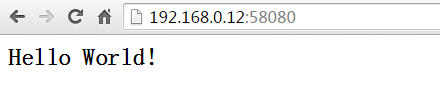
正如上面所描述的那样，可以通过“镜像名”或“镜像 ID”来启动容器，与上次启动容器不同的是，现在不再进入容器的命令行，而是直接启动容器内部的 Tomcat 服务。此时，需要使用以下命令：

|  |  |
| --- | --- |
| 1 | # docker run -d -p 58080:8080 -v /root/webapps/:/opt/tomcat/webapps/ --name mytomcat\_1 mytomcat:1.0 /root/run.sh |

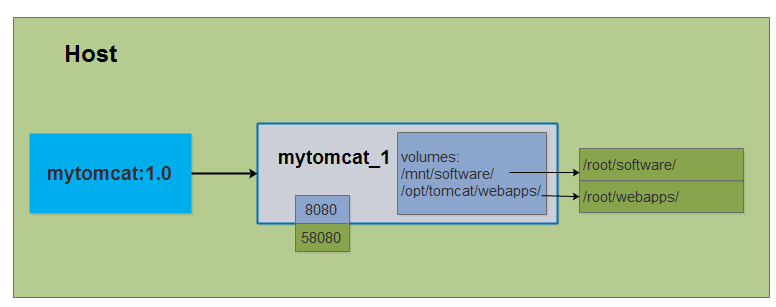
其中，相关参数包括：

* -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：



**2.9 最终示意图：**



**2.10 停止Tomcat容器**

|  |  |
| --- | --- |
| 1  2  3  4  5 | # 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 |

**2.11 移除容器**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | # 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 |

**2.12 移除镜像**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | # 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 |

# 其他指令

1.进入容器: docker exec -it <docker\_name> /bin/bash