# 常用指令(centos7实验通过)

|  |  |  |
| --- | --- | --- |
| 功能 | 命令 | 备注 |
| 安装 | yum -y install docker-io |  |
| 版本 | docker version |  |
| 信息 | docker info |  |
|  |  |  |
|  |  |  |
|  |  |  |
| 容器启停状态 | docker ps -a |  |
| 容器内存监控 | docker stats |  |
|  |  |  |
|  |  |  |
| docker服务 |  |  |
| 启动/停止 | service docker start/stop |  |
| 开机启动 | chkconfig docker on |  |
| 状态 | service docker status |  |
| 镜像 |  |  |
| 下载镜像 | docker pull \*\*\* | docker pull centos:latest   #从docker.io中下载centos镜像到本地 /var/lib/docker/graph |
| 导出（镜像） | docker save <imageNme> > <path> | docker save centos > /home/centos.tar |
| 启动（image） | docker run -itd centos /bin/bash | docker run -itd centos /bin/bash  -d 后台运行容器，并返回容器ID  -i 以交互模式运行容器，通常与 -t 同时使用  -t 为容器重新分配一个伪输入终端，通常与 -i 同时使用；  -h "mars" 指定容器的hostname  -e username="ritchie" 设置环境变量  --name="nginx-lb" 为容器指定一个名称  --dns 8.8.8.8 指定容器使用的DNS服务器，默认和宿主一致  --dns-search example.com 指定容器DNS搜索域名，默认和宿主一致 |
| 启动（每次开机自动启动容器） | docker run -itd --restart=always centos /bin/bash |  |
| 查看已下载的镜像 | docker images |  |
| 删除镜像 | docker rmi image\_id |  |
| 加载镜像 | docker load < /home/centos.tar |  |
| 搜索可用镜像 | docker search <name> | docker search mysql |
|  |  |  |
| 容器 |  |  |
|  |  |  |
| 启动（容器） | docker start container | #开启一个容器（注意container\_id和image\_id是完全不一样） |
| 查看容器状态 | docker ps -a |  |
| 删除容器 | docker rm \*\* | docker rm $(docker ps -a -q) #删除所有容器  docker rm containerId #删除某个容器 |
| 进入容器 | docker exec -it <容器ID> /bin/bash | docker exec -it cb2bb5ef655e /bin/bash |
| 导出（容器） | docker export <容器ID> > /home/centos1.tar | docker export d3ea2299b8b3 > /home/centos1.tar |
|  |  |  |
|  |  |  |
| 其它 |  |  |
| 挂载宿主机地址 | docker run -it --privileged=true -v /home:/home/outer:rw centos /bin/bash | --privileged=true 增加权限，否则docker没有权限  -v /home:/home/outer:rw centos 挂载目录，冒号前是宿主机目录，冒号后是docker中的目录 |
| 启动docker时启动mysql | docker run -e MYSQL\_ROOT\_PASSWORD=123qwe --name mysql -d mysql |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Build方式构建镜像

* **Docker的构建脚本名称必须为：Dockerfile。**
* **构建脚本中ADD使用的文件必须和Dockerfile在同一目录**。

docker build -t="repository/eurekaserver:latest" /home/liuzg/docker/builds/centos-jdk8

黄色字中不能包括大写字符。

* Dockerfile

|  |
| --- |
| # 设置继承镜像  FROM centos:latest  # 下面是一些创建者的基本信息  MAINTAINER liuzg (liuzg@163.com)  # 复制文件到镜像中  ADD jdk-8u101-linux-x64.rpm /software/  # 执行rpm命令安装JDK  ADD eurekaServer.jar /software/  # 执行rpm命令安装JDK  RUN rpm -ivh /software/jdk-8u101-linux-x64.rpm  # 设置JDK的环境变量  ENV JAVA\_HOME /usr/java/jdk1.8.0\_101  ENV PATH $PATH:$JAVA\_HOME/bin:$JAVA\_HOME/jre/lib  ENV CLASSPATH .:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar:$CLASSPATH  # 设置字符集  RUN export LANG=zh\_CN.UTF-8  CMD ["java -jar /software/eurekaServer.jar"] |

# 常用docker软件

## Svn(mamohr/subversion-edge)

|  |
| --- |
| //下载软件  docker pull mamohr/subversion-edge  //启动镜像  docker run -d -p 3343:3343 -p 4434:4434 -p 18080:18080 --privileged=true -v /data/svn:/opt/csvn/data --name svn-server mamohr/subversion-edge |

* 访问地址：

http://192.168.0.100:3343/

初始帐户信息：admin/admin

## 搭建subversion

|  |
| --- |
| * #启动docker   docker run -itd -p 3690:3690 --privileged=true -v /home/lzg/svn:/opt/svn/repositories:rw --name svnserver centos /bin/bash  #svn的库目录要挂载到宿主机目录下，以防止docker容器删除时丢失。   * #安装   yum install subversion  yum install net-tools   * #创建库   svnadmin create /opt/svn/repositories/myProject   * #配置   vi /opt/svn/repositories/myProject/conf/svnserve.conf  #anon-access = none  #auth-access = write  #password-db = passwd  #authz-db = authz  vi /opt/svn/repositories/myProject/conf/passwd  #配置用户=密码  vi /opt/svn/repositories/myProject/conf/authz  #配置权限   * #启动   svnserve -d -r /opt/svn/repositories   * #停止   killall svnserve   * #访问   svn co svn://localhost:3690 |

# 搭建docker私服

* 下载私服镜像

docker pull registry

* 启动私服窗口

docker run -d -p 5000:5000 registry

* 挂载私服的镜像存储目录

docker run -d -p 5000:5000 --privileged=true -v /home/liuzg/docker/registry:/tmp/registry registry

注：默认情况下，会将仓库存放于容器内的/tmp/registry目录下，这样如果容器被删除，则存放于容器中的镜像也会丢失，所以我们一般情况下会指定本地一个目录挂载到容器内的/tmp/registry下