第一章：掌握Docker安装及命令使用

安装前环境准备

```
#关闭防火墙与SELinux
[root@localhost ~]# systemctl stop firewalld
[root@localhost ~]# systemctl disable firewalld
[root@localhost ~]# setenforce 0
[root@localhost ~]# sed -i 's/SELINUX=enforcing/SELINUX=disabled/'
/etc/selinux/config

#配置本地yum源
[root@localhost ~]# mkdir /mnt/centos
[root@localhost ~]# mount /dev/cdrom /mnt/centos
[root@localhost ~]# rm -rf /etc/yum.repos.d/*
[root@localhost ~]# vim /etc/yum.repos.d/local.repo
[local]
name=local_centos
baseurl=file:///mnt/centos
enabled=1
gpgcheck=0
```

创建Docker存储库

官方安装文档：https://docs.docker.com/engine/installation/linux/centos/

```
#创建阿里源
[root@localhost ~]# curl -o /etc/yum.repos.d/CentOS-Base.repo
https://mirrors.aliyun.com/repo/Centos-7.repo

[root@localhost ~]# yum repolist

[root@localhost ~]# yum install -y yum-utils

#创建docker存储库（阿里）
[root@localhost ~]#  yum-config-manager --add-repo
http://mirrors.aliyun.com/docker-ce/linux/centos/docker-ce.repo

#安装docker软件包
root@localhost ~]# yum install docker-ce docker-ce-cli containerd.io -y

#查看docker版本信息
[root@localhost ~]# docker --version
```

配置镜像加速器

> 默认情况下镜像从docker hub下载，由于docker hub服务器在国外，由于网络原因镜像下载速度较慢，一般会配置镜像加速器进行下载

> 国内镜像加速器有阿里云、网易云、腾讯云、中科大等，本实验配置阿里云镜像加速器，速度较快

```
#创建阿里云镜像加速器
[root@localhost ~]# mkdir -p /etc/docker
[root@localhost ~]# tee /etc/docker/daemon.json <<-'EOF'
{
  "registry-mirrors": ["https://d3p1s1ji.mirror.aliyuncs.com"]
}
EOF

[root@localhost ~]# systemctl daemon-reload
[root@localhost ~]# systemctl restart docker
```

Docker服务命令

命令	作用
systemctl start docker	启动服务
sysmtectl status docker	查看服务状态
systemctl restart docker	重启服务
systemctl enable docker	设置服务随机自启
docker --version	查看docker版本信息
systemctl stop docker	停止服务

```
#启动服务
[root@localhost ~]# systemctl start docker

#查看服务运行状态
[root@localhost ~]# systemctl status docker
...
   Active: active (running)  #运行

#关闭服务
[root@localhost ~]# systemctl stop docker
[root@localhost ~]# systemctl stop docker
...
  Active: inactive (dead)   #死掉

#重启服务
[root@localhost ~]# systemctl restart docker

#设置服务随机自启
[root@localhost ~]# systemctl enable docker

#查看docker版本信息
[root@localhost ~]# docker version
```

Docker镜像命令

命令	作用
docker images	查看镜像
docker search 镜像名	搜索镜像
docker pull 镜像名:版本	拉取镜像
docker rmi 镜像名:版本	删除镜像

```
#查看docker可用命令
[root@localhost ~]# docker

#查看可用镜像
[root@localhost ~]# docker images
REPOSITORY    TAG         IMAGE ID    CREATED    SIZE
镜像名          版本         镜像ID      创建时间     大镜像小

#搜索镜像
[root@localhost ~]# docker search centos
NAME     DESCRIPTION                     STARS      OFFICIAL           AUTO
centos   The official buildofCentOS.  6639       [OK]
镜像仓库     镜像描述信息                    镜像数量    是否为官方发布镜像    是否为自动化构建的镜像

#下载镜像(如果不指定镜像版本则是最新版本，如需指定版本可从docker hub查看对应版本信息在进行下载)
root@localhost ~]# docker pull centos:7

#查看镜像
[root@localhost ~]# docker images
REPOSITORY    TAG         IMAGE ID        CREATED         SIZE
centos        7           8652b9f0cb4c    8 months ago    204MB

#查看所有镜像ID
[root@localhost ~]# docker images -q

#删除镜像（删除镜像可以根据镜像名称或者ID删除）
[root@localhost ~]# docker rmi centos

#如需删除所有镜像可以配合`docker images -q`获取命令的查询结果进行删除
[root@localhost ~]# docker rmi `docker images -q`
```

Docker容器命令

命令	作用
docker ps	查看正在运行容器
docker ps -a	查看所有容器
docker run 参数	创建容器
docker exec 容器ID/容器名	进入容器
docker stop 容器名/容器ID	停止容器
docker rm 容器名/容器ID	删除容器
docker start 容器名/容器ID	启动被停止的容器
docker kill 容器名/容器ID	强制停止正在运行的容器（一般不用，除非卡了）
docker inspect 容器名称	查看容器元数据信息
--restart=always	启动容器时设置容器随机自启
docker update --restart=always 容器名/容器ID	容器启动后设置容器随机自启

```
#创建容器：docker run
-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
--name="名称"：为容器指定一个名称
/bin/bash：在容器内执行/bin/bash命令
[root@localhost ~]# docker run -it --name=c1 centos:7 /bin/bash
[root@f52be4db44e7 /]#

#退出容器
[root@f52be4db44e7 /]# exit

#查看正在运行的容器信息（通过-it创建的容器退出后自动关闭）
[root@localhost ~]# docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES

#查看所有容器信息
-a：包括历史运行过的容器
[root@localhost ~]# docker ps -a
CONTAINER ID  IMAGE     COMMAND      CREATED       TATUS  PORTS     NAMES
容器ID        镜像名字    容器命令      创建时间       状态    端口      容器名字

#创建容器并放入后台运行（放入后台运行的容器退出后不会自动关闭）
-d：后台运行容器，并返回容器ID
[root@localhost ~]# docker run -id --name=centos2 centos:7
f34a8b2f79d15b2694d82bba23a9243c25bbbb8b8cde2e72f18b068fe53c97fc

#查看正在运行容器信息
[root@localhost ~]# docker ps
CONTAINER ID   IMAGE     COMMAND       CREATED          STATUS         PORTS
NAMES
f34a8b2f79d1   centos:7   "/bin/bash"   23 seconds ago   Up 21 seconds
centos2
```

```
#查看容器元数据信息
[root@localhost ~]# docker inspect centos2

#进入容器，在使用-d创建容器后，容器启动后会进入后台。此时想要进入容器，可以通过以下指令进入
docker attach
docker exec：推荐使用 docker exec 命令，因为此退出容器终端，不会导致容器的停止
[root@localhost ~]# docker exec -it centos2 /bin/bash
[root@f34a8b2f79d1 /]#

#停止容器(可以根据容器名称或者容器ID停止)
[root@localhost ~]# docker  stop  centos2

#启动被停止的容器
[root@localhost ~]# docker start centos1

#启动所有被停止的容器
[root@localhost ~]# docker start `docker ps -aq`

#删除容器(可以根据容器名称或者容器ID删除)
[root@localhost ~]# docker  rm  centos2

#删除所有容器(先获取所有容器ID在进行删除)
[root@localhost ~]# docker ps -aq
[root@localhost ~]# docker rm `docker ps -aq`
```

练习：通过Docker部署nginx的web应用

```
#下载nginx镜像
[root@localhost ~]# docker pull nginx:1.20.1

#创建容器并实现端口映射（默认容器无法被外网访问）
-p: 指定端口映射，格式为：宿主机端口:容器端口
[root@localhost ~]# docker run -id --name=ngx-v1 --restart=always -p 80:80
nginx:1.20.1

#浏览器访问：http://宿主机IP:端口
```