Docker（CentOS）

安装Docker

1.更新yum

```
[root@localhost ~]#yum -y update
```

2.安装Docker、查看版本

```
[root@localhost ~]#yum install docker -y
[root@localhost ~]#docker -v
```

3.设置国内仓库地址

```
vi /etc/docker/daemon.json
{
    "registry-mirrors": ["http://hub-mirror.c.163.com"]
}
#重启服务!
systemctl restart docker.service
```

国内加速地址:

Docker中国区官方镜像

网易

ustc

中国科技大学

阿里云容器 服务

4.启动Docker

```
[root@localhost ~]#systemctl start docker.service
[root@localhost ~]#systemctl status docker.service
```

看到active (running)就表示已经在运行了

以下是docker的生命周期管理常用命令

```
[root@localhost ~]#systemctl stop docker.service
[root@localhost ~]#systemctl start docker.service
[root@localhost ~]#systemctl restart docker.service
[root@localhost ~]#systemctl status docker.service
```

卸载Docker

1.搜索已经安装的docker 安装包

```
[root@localhost ~]#yum list installed|grep docker 或者 rpm -qa|grep docker
docker.x86_64 2:1.12.6-16.el7.centos @extras
docker-client.x86_64 2:1.12.6-16.el7.centos @extras
docker-common.x86_64 2:1.12.6-16.el7.centos @extra
```

2.分别删除安装包

```
[root@localhost ~]#yum -y remove docker.x86_64
[root@localhost ~]#yum -y remove docker-client.x86_64
[root@localhost ~]#yum -y remove docker-common.x86_64
```

3.删除docker 镜像

```
[root@localhost ~]#rm -rf /var/lib/docker
```

4.再次check docker是否已经卸载成功,如果没有搜索到表示已经卸载成功

```
[root@localhost ~]# rm -rf /var/lib/docker
[root@localhost ~]#
```

镜像管理

常见的镜像管理：

1. **search** 查看仓库里有些什么镜像　docker search 关键字（ho2j，mysql，tomcat，nginx）

   注：　镜像名称前面会默认加上 docker.io/
2. **pull** 拉取镜像
3. **images** 查看本地有些什么镜像
4. **rmi** 删除本地镜像
5. **tag** 修改本地镜像名称
6. **push** 把镜像提交到仓库
7. **run** 激活镜像

   ```
   [root@localhost ~]# docker run -dit --privileged -p21:21 -p80:80 -p8080:8080
   -p30000-30010:30000-30010 --name how2jtmall how2j/tmall:latest
   /usr/sbin/init
   ```

   - 　-dit：　-d -i -t 的缩写。 -d ，表示 detach，即在后台运行。 -i 表示提供交互接口，这样才可以通过 docker 和 跑起来的操作系统交互。 -t 表示提供一个 tty (伪终端)，与 -i 配合就可以通过 ssh 工具连接到 这个容器里面去了
   - --privileged：　启动容器的时候，把权限带进去。 这样才可以在容器里进行完整的操作
   - -p：-p21:21 第一个21，表示在CentOS 上开放21端口。 第二个21 表示在容器里开放21端口。 这样当访问CentOS 的21端口的时候，就会间接地访问到容器里了。-p80:80 和 21一个道理
   - --name：--name how2jtmall 给容器取了个名字，叫做 how2jtmall，方便后续管理
   - how2j/tmall:latest how2j/tmall就是镜像的名称，　latest是版本号，即最新版本
   - /usr/sbin/init: 表示启动后运行的程序，即通过这个命令做初始化

Docker安装操作Mongodb

```
#拉取镜像下载Mongo
docker pull mongo
#查看镜像
docker images
#制作容器
docker run --name mongo -p 27017:27017 -d mongo
#查看容器ID
docker ps -a
#进入容器内部
docker exec -it 容器ID /bin/bash
#找到客户端
whereis mongo
find / -name mongo
#启动客户端加入数据
./mongo
```

Docker删除容器与镜像e

1.列出所有容器IDbe

```
docker ps -aq
```

2.查看所有运行或者不运行容器

```
docker ps -a
```

3.停止所有的container（容器），这样才能删除其中的images:

```
docker stop $(docker ps -a -q) 或者 docker stop $(docker ps -aq)
```

4.如果想要删除所有container（容器）的话再加一个指令：

```
docker rm $(docker ps -a -q) 或者 docker rm $(docker ps -aq)
```

5.查看当前有些什么images（镜像）

```
docker images
```

6.删除images（镜像），通过image的id来指定删除谁

```
docker rmi <image id>
```

7.想要删除untagged images，也就是那些id为的image的话可以用

```
docker rmi $(docker images | grep "^<none>" | awk "{print $3}")
```

8.要删除全部image（镜像）的话

```
docker rmi $(docker images -q)
```

9.强制删除全部image的话

```
docker rmi -f $(docker images -q)
```

10.从容器到宿主机复制

```
docker cp tomcat:/webapps/js/test.js /home/admin
docker cp 容器名:      容器路径             宿主机路径
```

11.从宿主机到容器复制

```
docker cp /home/admin/test.js tomcat:/webapps/js
docker cp 宿主路径中文件          容器名       容器路径
```

12.删除所有停止的容器

```
docker container prune
```

13.删除所有不使用的镜像

```
docker image prune --force --all 或者 docker image prune -f -a
```

14.停止、启动、杀死、重启一个容器

```
docker stop Name或者ID
docker start Name或者ID
docker kill Name或者ID
docker restart Name或者ID
```

15.docker进入容器，查看配置文件

```
docker exec:在运行的容器中执行命令
        -d:分离模式，在后台运行
        -i:即使没有附加也保持STDIN(标准输入)打开，以交互式运行容器，通常与-t同时使用;
        -t:为容器重新分配一个为输入终端，通常与-i同时使用;
docker exec -it 容器ID /bin/bash  进入容器内部
[root@ 容器ID ~]# 已经进入容器内部
```

16.假如出现root@f94d2c317477:/usr/share/elasticsearch/config# vi elasticsearch.yml
bash: vi: command not found

```
apt-get update && apt-get install vim -y
```

17.修改配置、退出容器

```
1、如果要正常退出不关闭容器，请按Ctrl+P+Q进行退出容器
2、如果使用exit退出，那么在退出之后会关闭容器，可以使用下面的流程进行恢复
使用docker restart命令重启容器
使用docker attach命令进入容器
```