1. 列出所有容器ID

   `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），这样才能够删除其中的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

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

11. 从宿主机到容器复制

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

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;
    ```

    `docker exec -it f94d2c317477 /bin/bash`