

Mac安装docker:

下载Docker.dmg, 地址: <https://download.docker.com/mac/stable/Docker.dmg>, 然后进行安装。

Ubuntu安装 docker-ce:

- 1) `sudo apt-get remove docker docker-engine docker.io`
- 2) `sudo apt-get update`
- 3) `sudo apt-get install \`
`apt-transport-https \`
`ca-certificates \`
`curl \`
`software-properties-common`
- 4) `curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -`
- 5) `sudo add-apt-repository \`
`"deb [arch=amd64] https://download.docker.com/linux/ubuntu xenial stable"`
- 6) `sudo apt-get update`
- 7) `sudo apt-get install docker-ce`

centOS安装docker-ce:

- 1) `sudo yum update`
- 2) `sudo yum remove docker docker-common docker-selinux docker-engine`
- 3) `sudo yum-config-manager --add-repo https://download.docker.com/linux/centos/`
`docker-ce.repo`
- 4) `sudo yum install docker-ce`

Mac保存docker的image的位置：

/Users/{YourUserName}/Library/Containers/com.docker.docker/Data/
com.docker.driver.amd64-linux/Docker.qcow2

sshfs设置：

sshfs是基于fuse构建的ssh文件系统客户端程序，通过它远程主机的配置无需作任何改变，就可以透过SSH协议来挂载远程文件系统了，非常方便及安全。

在 <https://osxfuse.github.io> 上下载FUSE for macOS 和 SSHFS，并安装。

命令：sshfs {远程用户名}@{远程计算机地址}:{远程文件目录} {本地文件目录}

如：sshfs username@192.168.31.86:/home/username/sshfiles ~/remote

常见命令：

- 运行容器

docker run hello-world

- 显示本地镜像

docker images

- 获取镜像

docker pull ubuntu:12.04

- 创建一个容器，并让其运行bash

docker run -t -i ubuntu:12.04 /bin/bash

说明：-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。

- 利用DocFile来创建镜像

mkdir sinatracd sinatra/touch Dockerfilevim Dockerfiledocker build -t="ouruser/sinatra:v2".

- 运行容器

docker start [stop | restart] xxx

- 导出容器

```
docker ps -a | docker save -o ubuntu.tar.gz ubuntu:14.04
```

(保存的是所有这个镜像的版本记录，包括一些历史数据.)

```
docker export pid > ./ubuntu.tar.gz
```

(导出的是容器当前所用的镜像内容)

- 导入容器

```
docker load < /tmp/mynewimage.tar
```

(docker save 的逆操作)

```
cat ubuntu.tar | docker import - test/ubuntu:v1.0
```

(export 的逆操作)

- 删除容器

```
docker stop xxx | docker rm xxx
```

说明：docker 的文件是放在 /var/lib/docker/containers/ 中的。

- 保存镜像

```
docker commit xxx abc
```

说明：这里保存成一个名叫 abc 的镜像，镜像名称中只允许出现小写字母和数字。

- 删除镜像

```
docker rmi abc
```

- 获取镜像历史

```
docker history abc | docker images --tree
```

(查看所有镜像的历史过程)

说明：只能对本地存在的 docker 镜像执行这个命令。

- 杀死一个或多个指定容器进程

```
docker kill container
```

说明：-s “KILL” 自定义发送至容器的信号。

- 查看一个正在运行容器进程

```
docker top container
```

说明：也支持 ps 命令参数。

- 暂停某一容器的所有进程

`docker pause container`

- 恢复某一容器的所有进程

`docker unpause container`

- 标记本地镜像，将其归入某一仓库

`docker tag image name`

说明：-f 覆盖已有标记。

- 获取容器运行时的输出日志

`docker logs container`

- `docker attach`

`docker run -itd ubuntu:14.04 /bin/bash`
`docker attach 44fc0f0582d9`

说明：`docker attach`可以attach到一个已经运行的容器的stdin，然后进行命令执行的动作。但是需要注意的是，如果从这个stdin中exit，会导致容器的停止。

- `docker exec`

`docker exec -i name /bin/sh`

只用-i时，由于没有分配伪终端，看起来像pipe执行一样。但是执行结果、命令返回值都可以正确获取。

`docker exec -it name /bin/sh`

使用-it时，则和我们平常操作console界面类似。而且也不会像attach方式因为退出，导致整个容器退出。

`docker exec -t name /bin/sh`

如果只使用-t参数，则可以看到一个console窗口，但是执行命令会发现由于没有获得stdin的输出，无法看到命令执行情况。

`docker exec -d name /bin/sh a.sh`

在后台执行一个进程。可以看出，如果一个命令需要长时间进程，使用-d参数会很快返回。程序在后台运行。

- 分层级显示一个镜像或容器的信息

`docker inspect 465c60612e5c`