

获取阿里云的镜像，
设置阿里云镜像加速器



针对安装了Docker for Mac的用户， 您可以参考以下配置步骤：

右键点击桌面顶栏的 docker 图标，选择 Preferences ， 在 Daemon 标签（Docker 17.03 之前版本为 Advanced 标签）下的 Registry mirrors 列表中将， 上面的加速器地址复制进去， 然后点击 Apply & Restart 就可以了， 上面的配置， 就是在说去阿里云上面拉取镜像， 而不是去国外的Docker hub 去拉。

通过查看命令 docker info 在最后面可以看到， 配置是否成功

除了阿里云， 还有网易云镜像配置可以使用， 阿里云比网易云更全面一些

docker run 镜像

运行一个镜像， 如果这个镜像在本地没有， 那么就会去仓库下载

docker run hello-world

有输出， 说明你的docker 安装成功

Run的时候是跑起来的是容器， 如果这个容器没有， 那么去找镜像， 如果这镜像是存在的， 那么使用这个镜像产生一个实例， 也就是产生容器， 如果连镜像都没有， 那么就会去仓库下载。 然后实例化， 跑起来

Docker 比虚拟机快， 原因是Docker不需要讲全部的系统软硬件的虚拟化， 有着比虚拟机更少的虚拟层， Docker容器上的程序直接使用的都是实际屋里机的硬件资源， 在CPU内存等 Docker更加的有效。

使用Docker引擎共享宿主机OS， 在虚拟机上， 虚拟机需要安装新的系统的时候需要将新安装的系统， 全部都虚拟化， 然后加载， 这个启动过程都很耗时间的， 是直接在宿主机上运行 OS， Docker是与宿主机共享OS， 快速， 妙级启动。

Docker —help

常用镜像命令：(镜像相当于 类、模板， 可以生产容器)

鲸鱼背上有集装箱

鲸鱼鲸鱼游在蓝色的大海里 — 相当于宿主机

鲸鱼 — — docker

集装箱 — — 容器实例 来自于镜像

docker images 命令

docker images 列出本地所有镜像，操作的时候可以对镜像名称或者 ID 镜像操作，Tag 版本号

-a 镜像是分层的，这里是将所有镜像，包括中间层的镜像，我们看到的是最上层的镜像，其实下面还有很多的

-q 显示当前镜像的全部镜像的ID

-digests 显示摘要信息，如果有就会显示，没有那就是没有，有什么显示什么

docker search images_name 在docker 仓库中查看你需要的image

docker search -s 30 images_name star 个数为30 的哪个镜像罗列

docker search -s -automated images_name 罗列 automated

docker pull images_name 从仓库下载image

这个命令等价于

docker pull images_name:latest 拉去最新版的镜像

docker rmi images_name 删除镜像或者镜像ID

这个命令等价于

docker rmi images_name:latest

-f 强制删除

docker rmi images_name_1, images_name_2 删除多个镜像

Docker rmi -f \$(docker images -qa) 全面批量删除 \$(docker images -qa) 去到所有的镜像

docker container 命令

新建并启动容器

docker run [OPTIONS] image [COMMAND][ARGS...]

-d 后台运行容器,并返回容器ID， 也即启动守护守护式容器

-i 以交互式模式运行容器，通常会与 -t 同时使用，可以合起来 -it

-t 为容器重新分配一个伪输入终端， 和 -i 同时使用

-P 随机端口映射

-p 指定断后映射

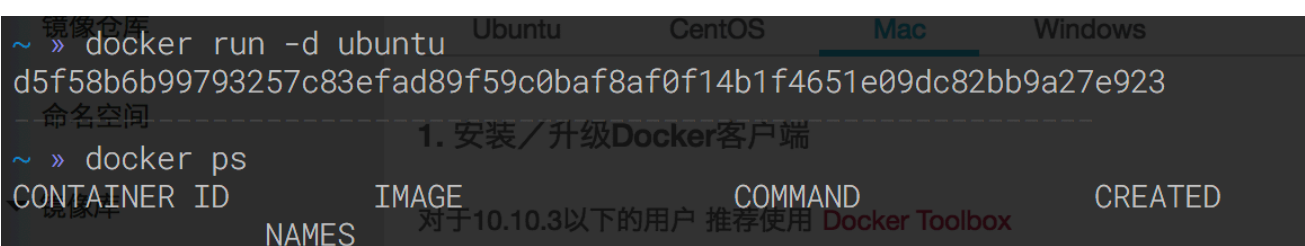
ip:hostPort:containerPort

ip:

-name 给新起来的容器一个名字，而不是ID 号



上图可以看出来，当运行我的ubuntu 的ID 号之后就进入到了ubuntu系统中， 也就是进入到一个容器中了



-d 没有交互， 后台运行。ubuntu 是肯定启动了的， ps命令并没有看见ubuntu 在运行， 这里涉及到docker的机制， 这里会将启动的ubuntu自动给杀掉， 原因在于docker 认为ubuntu启动起来是没事可做的， 所以最近的解决方案是， 将你的进程以前台进程的形式运行。

docker run -d ubuntu /bin/sh -c "while true; do echo hello zzyy;sleep 2;done"

这样执行的话， 后台就不会挂掉， 会一直执行

docker ps 查看当前所有正在运行的容器

-l 上一个已经运行被停止了了的容器

-a 所有的运行着的或者已经停止了了的容器

-n 3 上3次运行了的容器

-q 只显示容器ID， 可以 -ql 组合使用， 上一个容器的ID

--no-trunc 无截断输出

exit 退出容器， 退出的时候， 容器停止

ctrl + P+ Q 容器不停止退出，

Docker start 容器ID 启动容器

Docker restart 容器ID 重启容器

docker stop 容器ID 停止容器

docker kill 容器ID 马上直接强制停止容器(相当于直接拔掉插座)

docker rm \$(docker ps -a -q) 删除已经停止的容器

docker logs -f -t —tail 容器ID

-t 日志中加入时间

-f 跟随最新的日志打印

-tail数值 显示最后多少条

查看容器内的进程

docker top 容器ID

查看容器内部细节

docker inspect 容器ID

进入正在运行的容器， 并以命令行交互

docker exec -it 容器ID 在容器中打开终端， 并且可以启动新的进程

docker attach 容器ID 重新进入， 直接进入到启动命令的终端， 不会启动新的经常

docker run ubuntu echo hello docker 可以之间在容器外面对容器执行命令

或者这里使用 exec 也是可以这么做

docker exec -it ubuntu echo hello docker 可以之间在容器外面对容器执行命令

容器中的文件复制到主机中

docker cp 5474000bf487:/haha.txt /

容器ID:容器中文件 本机路径