Docker使用文档

2021/04/16

目录

[简介 3](#_Toc69562994)

[Centos安装 4](#_Toc69562995)

[Docker操作 6](#_Toc69562996)

[拉取镜像 6](#_Toc69562997)

[镜像操作 6](#_Toc69562998)

[容器操作 7](#_Toc69562999)

[镜像导出与恢复/容器导出与恢复 9](#_Toc69563000)

[日志 9](#_Toc69563001)

[网络 10](#_Toc69563002)

[修改容器配置文件 11](#_Toc69563003)

[Dcokerfile编写 13](#_Toc69563004)

[阿里云容器镜像服务 16](#_Toc69563005)

[参考内容 17](#_Toc69563006)

\*注：1.master\_代表宿主机，docker\_代表容器

2.[]内的需要自定义

3.全局需要root。可以从#看出

# 简介

Docker是一个开源的应用容器引擎，基于Go语言并遵从Apache2.0协议开源。

Docker通过内核虚拟化技术(namespaces及cgroups)来提供容器的资源隔离与资源限制。通过操作系统层的虚拟化实现隔离。

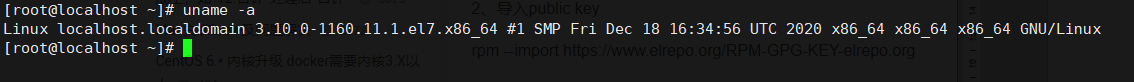
Docker容器在运行时，不需要类似虚拟机(VM)额外的操作系统开销，从而比kvm虚拟机更轻量。也可以把docker理解为一种简单的打包技术。

Docker架构包含镜像、容器、仓库。当启动一个新的容时，Docker会加载只读镜像，并在其之上添加一个读写层，即容器层。所以说，容器=镜像+可读层。

Docker如今分为Docker CE和Docker EE,即社区版和企业版。

# Centos安装

Docker运行需要Centos内核版本至少在3.8以上，建议3.10。运行在 CentOS 7 上，要求系统为64位、系统内核版本为3.10以上、Centos7的内核一般都是3.10的，而CentOS 6.X 的内核一般都是2.6，在2.6的内核下，Docker运行会比较卡，所以一般会选择升级到3.10版本。可以使用命令查看。

 # uname -a

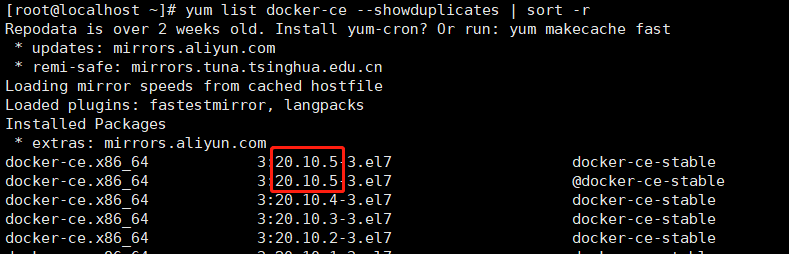
**以下操作不进入Docker。**

1.设置镜像源

# yum-config-manager --add-repo http://download.docker.com/linux/centos/docker-ce.repo（中央仓库）

# yum-config-manager --add-repo http://mirrors.aliyun.com/docker-ce/linux/centos/docker-ce.repo（阿里仓库）

2.查看docker-ce的版本(社区版)

# yum list docker-ce --showduplicates | sort -r

3.安装docker-ce

# yum install -y docker-ce-[版本号]

版本号位上图中红色框的内容。

4. 启动/停止

# systemctl start docker

# systemctl stop docker

开机自启自选

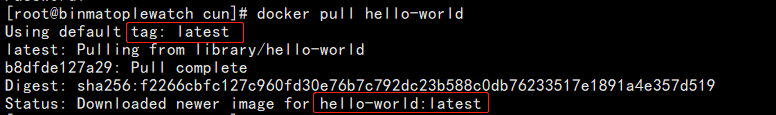
# systemctl enable docker

5. 查看版本

# docker version / docker info

# Docker操作

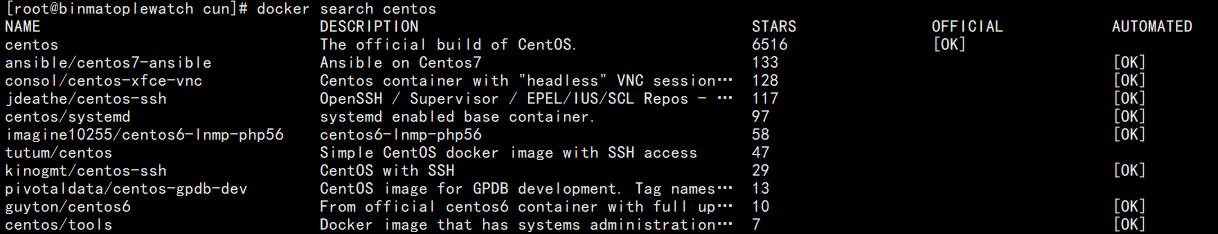
## 拉取镜像

 # docker pull [IMAGE\_NAME:tag]

如果不指定tag,表示拉取最新版本，即latest。见红框。这是Docker做的一个欢迎程序，只有13.3kb。

如何需要找寻合适自己的镜像，可以使用docker search 查看。

# docker search [IMAGE\_NAME]

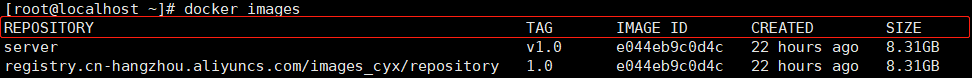


这里查找centos的镜像，根据描述选择需要的。

如果觉得这些都不适合需要，可以从已经创建的容器中更新镜像，并且提交这个镜像，具体操作见镜像和容器导出与恢复操作部分。或者可以使用Dockerfile指令来创建一个新的镜像，具体操作见Dockerfile编写部分。

镜像操作

1. 镜像查看

# docker images

主要是名称、标签(版本)、镜像ID、创建时间、镜像大小。

1. 镜像删除

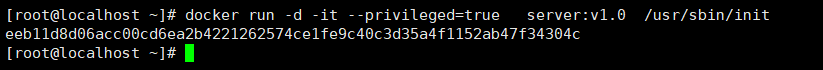
# docker image rm [IMAGE ID]

或者

# docker rmi [IMAGE ID]

容器操作

1.创建容器并启动

# docker run -d -it --privileged=true server:v1.0 /usr/sbin/init

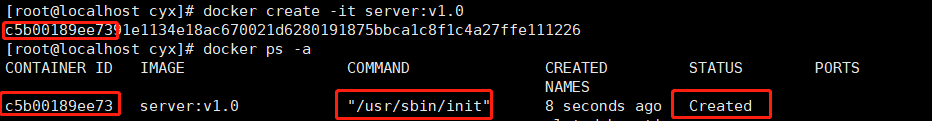
这是最简单的一个创建启动命令，server:v1.0代表的意义是上图中镜像的名称和版本号，中间加一个冒号。-d 后台运行，--privileged=true代表我们拥有这个容器的root权限，是指定容器为特权容器的参数，没有在容器内部会出错“无法访问D-Bus,被拒绝”。结果产生的是容器ID,一般只显示前12位。“/usr/sbin/init”是必须要的，具体意义不在此赘叙，自行百度。

这是一个指定多个选项的启动命令

# docker run -d -it --privileged=true --name="centos7" -p 10022:22 -p 10080:80 -h "docker" -v /maste\_dir:/docker\_dir centos:7.8.2003 /usr/sbin/init

简单解释，我们指定了容器的名称为centos7，hostname为“docker”，以及-v挂载的目录，两个端口(master:docker)。

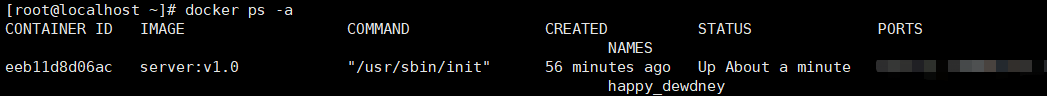
具体的参数请移步至此处：https://docs.docker.com/engine/reference/commandline/run/

 或者分为两步。先使用docker create [OPTIONS] [IMAGE ID] [COMMAND] [ARG...] ，用法与词法同docker run。结果如下图。自定义了创建命令。

然后使用 docker start [CONTAINER ID]就可以启动，状态变为UP，进入请看第4点。

1. 查看容器列表

# docker ps

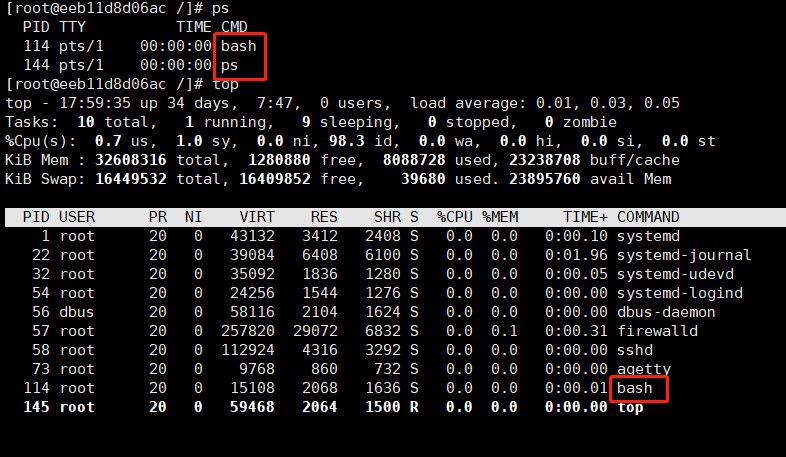
 命令结果会显示容器ID、镜像名称及版本、创建时的命令、创建经历时间、状态、端口信息，容器名称(此处显示错位)。“UP”代表正在运行中。

参数 -a:显示所有的容器，没有只显示正在运行的。

启动时没有指定 --name 会随机分配一个名字，反之就是指定的。

1. 进入容器

# docker exec -it [CONTAINER ID] /bin/bash

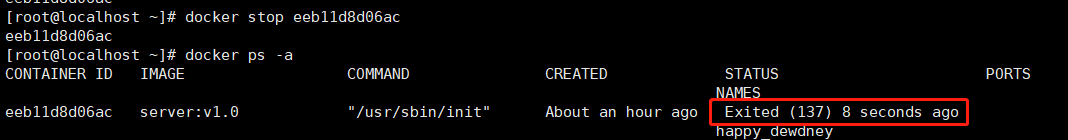
 首先必须有 /bin/bash,代表至少执行一个程序。可以使用**# ps**和 **# top**查看。只运行了最基础的程序，其他的没有。之后就可以做自己想做的事。

5.容器退出

# exit

6.开始/停止容器

# docker start [CONTAINER ID]

# docker stop [CONTAINER ID]

停止后状态会变为“Exited”，开启后会变为“UP”。

7.删除容器

# docker rm -f [CONTAINER ID]

8.查看容器状态

# docker stats [CONTAINER ID]

## 镜像导出与恢复/容器导出与恢复

# docker save -o image.tar [IMAGE ID] //导出镜像，格式tar

# docker load -i image.tar //恢复出来的是镜像

# docker export -o image.tar [CONTAINER ID] //导出容器

# docker import image.tar image\_name:v //恢复出来是镜像

黄色标注的名字可以自定义，青色标注的代表容器压缩包导出后的镜像名字及版本。导出、恢复命令比较费时，请耐心等待，它没有进度条显示。

## 日志

存储位置:主机目录:

/var/lib/docker/containers/[container\_id]/[container\_id-json.log]

在此文档编写过程中，容器ID为eeb11d8d06ac的容器日志并没有产生在上述的json文件中。我们可以在 /var/log/message即系统日志中查看容器的相关日志。

## 网络

具体请移步至此：https://docs.docker.com/network/

# 修改容器配置文件

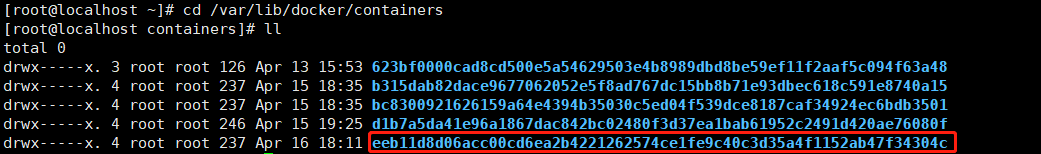
如果需要修改启动时的端口映射或者挂载的目录。可以通过容器的配置文件进行修改，其他属性可自行查阅修改。

1.首先，需要停止容器和docker

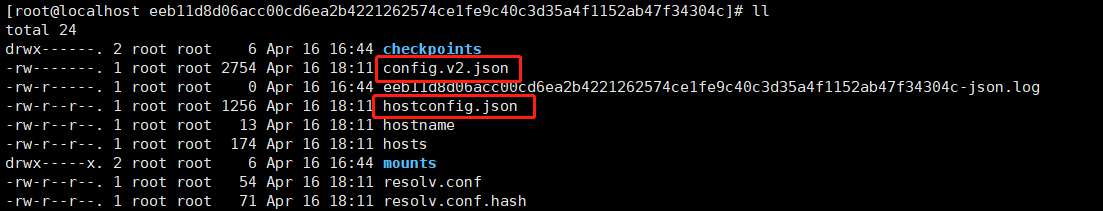
# docker stop [CONTAINER ID]

# systemctl stop docker

2.进入配置目录

 # cd /var/lib/docker/containers/[CONTAINER ID]\*

红框内的数字的前12位就是容器ID，这个目录会显示所有容器对应的配置目录，进入这个目录。

 只需要关注这两个json文件。

修改或添加或删除挂载目录：hostconfig.json下Binds属性添加，"Binds":["master\_dir1:docker\_dir1","master\_dir2:docker\_dir2"]。

config.v2.json下MountPoints属性添加

"MountPoints": {"docker\_dir": {"Source": "master\_dir","Destination": "docker\_dir","RW": true,"Name": "","Driver": "","Type": "bind","Propagation": "rprivate","Spec": {"Type": "bind","Source": "master\_dir","Target": "docker\_dir"},"SkipMountpointCreation": false}}

修改或添加或删除端口：hostconfig.json下的

"PortBindings":{"docker\_port/tcp":[{"HostIp":"","HostPort":"master\_port"}],"2345/tcp":[{"HostIp":"","HostPort":"12345"}]}

config.v2.json 下的 "ExposedPorts" "Ports" 字段

"ExposedPorts": {"docker \_port/tcp":{},"2345/tcp":{}},

"Ports":{"docker \_port/tcp":[{"HostIp": "", "HostPort":"master\_port"

}],"2345/tcp":[{"HostIp":"","HostPort":"12345"}]}

黄色标注的可自行修改

3.完成后启动容器和dcoker

# docker start [CONTAINER ID]

# systemctl start docker

目录挂载的方式可移步至此：https://blog.csdn.net/zedelei/article/details/90208183

# Dcokerfile编写

Dockerfile是一个用来构建镜像的文本文件，文本文件包含了构建镜像所需的指令和说明。

Dockerfile的作用是安装文件中指令定义的docker容器或者容器中的应用程序以及服务。

Dockerfile 分为四部分：基础镜像信息、维护者信息、镜像操作指令和容器启动时执行指令。

以下举例Dockerfile中常见的操作指令和作用：

#：添加注释信息

FROM：指定创建镜像的基础镜像；

MAINTAINER：Dockerfile作者信息，一般写的是联系方式；

RUN：运行Linux系统的命令使用；

CMD：指定容器启动执行的命令；启动容器中的服务；

LABEL：指定生成镜像的源数据标签；

EXPOSE：指定镜像容器监听端口号；发布服务使用；

ENV：使用环境变量；

ADD：对压缩文件进行解压缩；将数据移动到指定的目录；

COPY：复制宿主机数据到镜像内部使用；

WORKDIR：切换到镜像容器中的指定目录中；

VOLUME：挂载数据卷到镜像容器中；

USER：指定运行容器的用户；

ARG：指定镜像的版本号信息；

ONBUILD：创建镜像，作为其他镜像的基础镜像运行操作指令；

ENTRYPO

INT：指定运行容器启动过程执行命令，覆盖CMD参数；

参数的具体解释请移步至此：https://docs.docker.com/engine/reference/builder/

 用张图解释一下：

1.编写举例：

FROM centos:7.8.2003 // 指定基础镜像

# Install tools

RUN yum update -y && \ //安装需要的服务

yum install -y wget && \

yum install -y vim && \

yum clean all && \

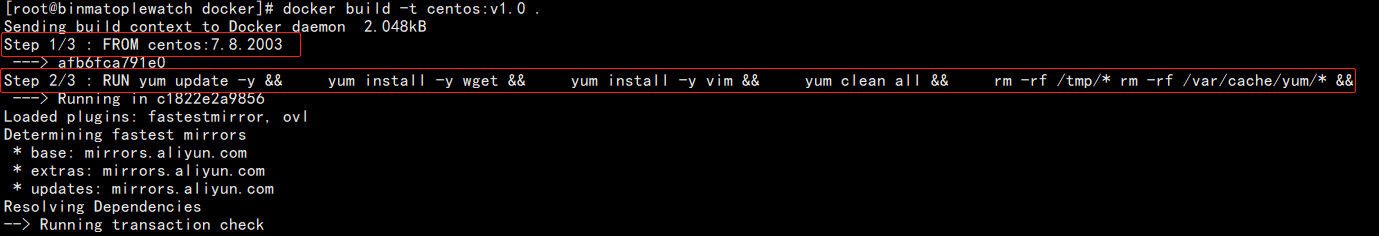
rm -rf /tmp/\* rm -rf /var/cache/yum/\* && \ //删除临时文件

ln -sf /usr/share/zoneinfo/Asia/Shanghai /etc/localtime //设置时间

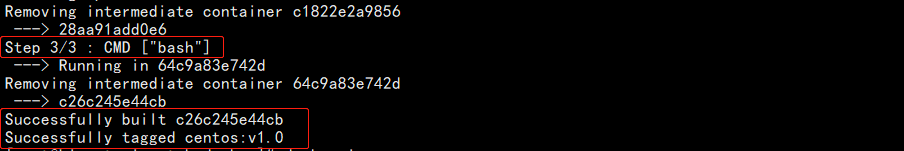
# Define default command.

CMD ["bash"] //执行命令

2.编写完成执行构建命令

# docker build -t name:tag . //后面有个点 “.”表示当前路径

中间省略的截图是整个安装过程，请耐心等待。



红框第一步就是拉取基础的镜像，第二部就是执行命令，第三步就是最后的bash，最后显示成功。命令中的“centos:v1.0”就是下图标注的内容。

之后可以参照前面的命令创建，启动容器。

关于docker build的相关参数请移步至此，此处只简单举例：

https://docs.docker.com/engine/reference/commandline/build/

# 阿里云容器镜像服务

免费使用个人版，存储镜像。

1. 登录
2. 进入产品与服务 -> 弹性计算 -> 容器镜像服务 -> 实例列表
3. 此处根据提示自行解决。
4. 仓库管理 -> 访问凭证，点击“设置固定密码”，或者每次都用临时密码，用于登录。
5. 推/拉镜像的具体操作在“镜像仓库/基本信息”中有具体描述，自行参考，此处不再赘述。

添加阿里云镜像加速

1.在阿里云的“容器镜像服务/镜像工具/镜像加速器”有个加速地址，按照下面的提示完成添加。

下面是举例：

# vim /etc/docker/daemon.json

添加自己的镜像地址,保存退出

{

"registry-mirrors": ["https://691af043.mirror.aliyuncs.com"]

}

# systemctl daemon-reload

# systemctl restart docker

# 参考内容

官方文档：https://docs.docker.com/