docker学习笔记
1.Linux 系统内核要求
#： uname -r
3.10.0-1160.45.1.el7.x86_64

2.系统版本
cat /etc/os-release

开始安装 （注意各linux的区别，这里用CentOs 7 为模板）
帮助文档 https://docs.docker.com/engine/install/centos/
Install Docker Engine on CentOS | Docker Documentation
1.清除旧版本
```
yum remove docker \
                  docker-client \
                  docker-client-latest \
                  docker-common \
                  docker-latest \
                  docker-latest-logrotate \
                  docker-logrotate \
                  docker-engine
```

2. 安装依赖库多种安装方式（我 选择yum安装）
```
yum install -y yum-utils
```

3. 设置镜像仓库
```
yum-config-manager \
                  --add-repo \
                  https://download.docker.com/linux/centos/docker-ce.repo （这是国外的默认仓库，估计很慢可以换国内源）
                  http://mirrors.aliyun.com/docker-ce/linux/centos/docker-ce.repo （阿里云镜像 ）
```

4. 配置部分（暂时不需要）

5.安装doccer相关依赖
A.最好先**yum缓存命令**
yum makecache fast

B.安装docker-ce 是社区版 docker-ee是企业版 默认最新版
yum install docker-ce docker-ce-cli containerd.io

C.启动服务
systemctl start docker

D.查看版本

docker version

E. 测试 hello-world 镜像
docker run hello-world

F.查看 hello-world 镜像
docker images

其他命令
1. **卸载 Docker 引擎，删除 3 个依赖**

yum remove docker-ce docker-ce-cli containerd.io
2.清除资源
sudo rm -rf /var/lib/docker
$ sudo rm -rf /var/lib/containerd

配置镜像加速器

针对 Docker 客户端版本大于 1.10.0 的用户
您可以通过修改 daemon 配置文件 /etc/docker/daemon.json 来使用加速器 4 不部

sudo mkdir -p /etc/docker

sudo tee /etc/docker/daemon.json <<-'EOF'
{
  "registry-mirrors": ["https://fk5oluct.mirror.aliyuncs.com"]
}
EOF

sudo systemctl daemon-reload

sudo systemctl restart docker

回顾 Hello-world 流程

本地 images 查找，远程仓库

docker 的常用命令

docker version # 查看版本信息
Docker info # 更加详细的信息 包括镜像和容器的数量
docker 命令 —help # 帮助命令
官网帮助文档 https://docs.docker.com/reference/

镜像命令

查看镜像
docker images

[root@iZuf63uv0sb0bltnjq53ltZ /]# docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
hello-world latest feb5d9fea6a5 5 months ago 13.3kB
镜像的仓库源 镜像标签 镜像🆔 创建时间 大小
-a, --all 列出所有的镜像
-q, --quiet 只显示ID

搜索镜像
docker search mysql
--filter=SATRT=3000 #只要3K星的

下载镜像 docker pull 镜像名[：tag]
docker pull

[root@iZuf63uv0sb0bltnjq53ltZ /]# docker pull mysql.
Using default tag: latest.
latest: Pulling from library/mysql # 分层下载 docker image的核心， 联合文件系统
72a69066d2fe: Pull complete
93619dbc5b36: Pull complete
99da31dd6142: Pull complete
626033c43d70: Pull complete
37d5d7efb64e: Pull complete
ac563158d721: Pull complete
d2ba16033dad: Pull complete
688ba7d5c01a: Pull complete
00e060b6d11d: Pull complete
1c04857f594f: Pull complete
4d7cfa90e6ea: Pull complete
e0431212d27d: Pull complete
Digest:
sha256:e9027fe4d91c0153429607251656806cc784e914937271037f7738bd5b8e7709 #签名
Status: Downloaded newer image for mysql:latest
docker.io/library/mysql:latest # 真实地址
等价于 docker pull mysql

指定版本下载
docker pull mysql:5.7

删除镜像. 可以名称 或者镜像ID
docker rmi -f 镜像ID 镜像ID 镜像ID
递归删除,全部删除
docker rmi -f $(docker images -aq)

容器命令

A.创建一个centos容器

docker pull centos

B 运行容器

docker run [可选参数] image
一般运行方式
启动并进入容器
docker run -it centos /bin/bash
[root@iZuf63uv0sb0bltnjq53ltZ /]# docker run -it centos /bin/bash
[root@ed908713fb2e /]#   已经进入容器

退出容器　　后台运行容器
exit　　ctrl + p + q

参数说明
--name='Name' 容器名称
-d　　后台方式运行
-I -t　　使用交互方式运行，进入容器查看
-p　　指定端口　-p 8080：8080 和宿主机映射
-P　　随机指定端口

C.查看正在运行的容器

docker ps -aq
-a 显示曾经运行的容器
-n=? 显示最近创建的容器
-q 显示容器🆔

D.删除容器

Docker rm　　删除指定容器id，不能删除运行中的容器 ，强制删除 rm -f
docker rm -f $(docker ps -aq)　递归删除所有容器
Docker ps -a -q | xargs docker rm. 删除所有容器,linux删除法

E.启动和停止容器的操作

docker start 容器ID.　　启动
docker restart 容器ID　重启
docker stop 容器ID　　停止
docker kill 容器ID　　杀死

F.常用命令

1.后台启动．注意 问题

docker run -d centos
常见的坑：docker容器使用后台运行，必须要有个前台进程，docker没有引用就会停止
docker run -d centos /bin/sh -c “while true; do echo hobby;sleep 1; done” # 加了 shell命令
Nginx ，容器启动会

2.查看日志

docker logs -tf —tail 10 容器ID
-tf. # 显示日志
—tail number # 显示日志条数

3.查看容器中的进程信息

Docker top 容器ID

4.查看镜像中元数据 显示容器

docker inspect 容器🆔

5.进入当前正在运行的容器

方式一 .进入容器后打开一个新的终端（常用）
docker exec -it 容器id．/bin/bash
方式二. 正在执行当前的代码的终端，不会开启新进程
docker attach 容器id.

6.容器内外文件交互

1容器内部往外部拷贝文件
docker cp 容器🆔：/home /home

7.查看容器性能

docker stats

练习 配置nginx
1.search
2.pull
3 .run -nginx 并暴露端口3344映射到nginx80端口
docker run -d --name nginx01 -p:3344:80 nginx
4.测试
curl localhost3344
5.设置配置文件
进入容器 docker exec -it 容器🆔 /bin/bash
root@24f8bc55a22e:/# whereis nginx
nginx: /usr/sbin/nginx /usr/lib/nginx /etc/nginx /usr/share/nginx

配置 tomcat
Docker run -it —rm tomcat．用完就删除，测试用

docker run -d -p3355：8080 容器🆔　正常方式
curl localhost3355
docker exec -it 容器🆔 /bin/bash

部署 es+ kibana

增加图形化管理工具
docker run -d -p 8888:9000 --restart=always -v /var/run/docker.sock:/var/run/docker.sock --privileged=true portainer/portainer

8，镜像的制作
方式一：
docker commit -a =“aticle_hobby” -m= “info_add” 容器🆔 name:xxx

容器数据卷

1.使用数据卷
方式一： -v 宿主机目录：容器目录

查看 docker inspect 容器🆔 在 Mounts 中查看挂在情况

2.Mysql：5.7 数据持久化

docker run -d -p 3320:3306 --name mysql3306
-v /home/mysql/conf:/etc/mysql/conf.d # 数据库配置文件
-v /home/mysql/data:/var/lib/mysql # 数据库文件
-e MYSQL_ROOT_PASSWORD=123456 容器🆔 # 创建 mysql 加密码

3 具名和匿名挂载
Docker volume［age］
Age : create 创建卷
　　inspect 检查卷
　　ls 列出卷
　　prune 删除不在使用的卷
　　rm 删除卷

A: docker run -d -p 3344:80 --name nginx_hobby -v honny-nginx:/etc/nginx(:权限) 605c77e624dd 具名挂载 - V 后面 卷🆔
ro Readonly 只读. 只能外部改变
rw. Readwite 读写 默认
查看
B:
Docker volume ls
Docker volume inspect 卷🆔

ex：

```
{
      "CreatedAt": "2022-03-19T16:13:35+08:00",
      "Driver": "local",
      "Labels": null,
      "Mountpoint": "/var/lib/docker/volumes/honny-nginx/_data",      卷路径
      "Name": "honny-nginx",       卷名称
      "Options": null,
      "Scope": "local"
  }
```

DockerFile

原来构建镜像的文件

1.方式二

A.创建 dockerfile文件 dockerfile1 是脚本

```
from centos
VOLUME [“volume01” , “volume02”]
CMD echo  '_______end_____'
CMD /bin/bash
```

Docker File指令

基础知识：

B.执行 build构建成一个文件

```
docker build -f dockerfile1 -t  hobby/centos:1.0 ./
```

C.运行镜像

查看绑定的路径，在路劲内创建文件

退出容器

docker inspect 容器🆔 查看

```
"Mounts": [
        {
            "Type": "volume",
            "Name":
"e4be0db356808f75957b2419c6137d9015ffa24b1182085b0b6b4453f5a880c
9",
            "Source": "/var/lib/docker/volumes/
e4be0db356808f75957b2419c6137d9015ffa24b1182085b0b6b4453f5a880c9
/_data",   宿主机内位置
            "Destination": "[volume01,volume02]", 容器内名称
            "Driver": "local",
            "Mode": "",
            "RW": true,
```

```
            "Propagation": ""
        }
    ],
```

4. 发布镜像（DockerHbu，阿里云）

容器内通讯

多mysql数据共享

1.启动第一个容器
docker run -it --name test01 镜像🆔
2.使用相同镜像启动第二个容器并继承第一个容器
docker run -it --name test02 --volumes-from test01(父容器) 镜像🆔
⚠️
⚠️ 容器数据卷中的文件是互相同步，但是容器内数据有独立的备份，是多分
⚠️
3.在任意一个挂载的卷内部创建文件，会同步文件
Mysql 可以直接继承
☝️第一步启动一个mysql容器
docker run -d -p 3320:3306 --name mysql001
-v /home/mysql/conf # 数据库配置文件
-v /home/mysql/data # 数据库文件
-e MYSQL_ROOT_PASSWORD=123456 容器🆔 # 创建mysql加密码
☝️ 启动第二个mysql容器，并继承容器一

docker run -d -p 3320:3306 --name mysql002
—volumes-from mysql01. # 继承容器001
-e MYSQL_ROOT_PASSWORD=123456 容器🆔 # 创建mysql加密码

dockerfile详细

Dockerfile 构建过程

```
FROM          #基础镜像
MAINTAIMER    # 维护者信息 姓名+邮箱
RUN           # 运行的时候需要的命令
ADD           #  步骤添加内容
WORKDIR       #设置工作目录
VOLIUME       # 设置挂在目录
EXPOSE        #  对外端口
CMD           # 容器的运行的命令，只有最后一个会生效，可被替代
ENTRYPOINT    #指定容器启动运行时的命令，可以追加命令
ONBUUILD      # 当构建一个被继承的父容器，这个时候就会触发ONBUILD命令
COPY          # 类似ADD，将文件拷贝到镜像中
ENV           # 设置环境变量
```