# Docker部署服务

## 安装docker

1. 安装：

yum install docker (什么？没有yum 自己看着办)

1. 关闭https连接：

vi /etc/docker/daemon.json 添加一下内容

{

"insecure-registries": [

"10.253.254.36"

]

}

1. Docker ps –a /docker images 不报错则运行正常。
2. 开启远程访问(打包镜像服务器需要开启，其他的不需要 )

Vi /usr/lib/systemd/system/docker.service

在 execStart = 后面添加 -H unix:///var/run/docker.sock -H tcp://0.0.0.0:2375

## 安装docker-compose

1. 下载docker-compose 到 /usr/local/bin目录中

###确保 /usr/local/bin/ 目录存在，不存在则创建 mkdir /usr/local/bin/

curl -L https://github.com/docker/compose/releases/download/1.23.0-rc3/docker-compose-`uname -s`-`uname -m` -o /usr/local/bin/docker-compose

1. ####提供可执行权限

chmod +x /usr/local/bin/docker-compose

1. ##查看版本

docker-compose version

# Docker常用命令

#查看所有镜像

docker imges

#查看所有容器

docker ps -a

#创建镜像（当前目录有Dockerfile 和所需的文件）

docker build -t yunce/[serviceName]:[tagName] .

docker build -t 10.253.254.36/yunce/[name]:1.0 -f Dockerfile\_[name] .

( -t 命名镜像 (ip(远程仓库)/文件夹/名称) -f 指定dockerfile 绝对路径)

#创建容器并运行

docker run -d -p [hip]:[ip] -v /home/app/logs:/logs -e "SPRING\_CLOUD\_NACOS\_DISCOVERY\_IP=[ip]" --name [serviceName] [IMG]

(-d 以守护进程运行，-p 映射端口 –v 挂载目录 –e 设置环境变量 –name 为容器命名 IMG (指定镜像))

#设置环境变量 ip地址

docker run -d -p 8088:8088 -e "SPRING\_CLOUD\_NACOS\_DISCOVERY\_IP=[ip]" -v /home/app/logs:/logs --name zuul 10.253.254.36/yunce/ycs-ms-service-zuul-v1:0.2 -it /bin/bash

#启动/重启/停止/强制停止/移除

docker start/restart/stop/kill/rm [containerName]

#连接容器 (进入容器中)

docker exec -it [containerName] /bin/bash

#移除镜像

docker rmi [imgName]

#修改镜像名

docker tag [oldImgName:oldTag] [IP]/[newImgName:newTag]

#查看镜像，容器信息

docker inspect [IMGNAME/CONTAINERNAME]

#pull镜像

docker pull [IP]/[IMGNAME:TAG]

#推送

docker push [IP]/[IMGNAME:TAG]

#删除所有镜像 容器 等

docker container prune # 删除所有退出状态的容器

docker volume prune # 删除未被使用的数据卷

docker image prune # 删除 dangling 或所有未被使用的镜像

删除容器：docker container rm $(docker container ls -a -q)

删除镜像：docker image rm $(docker image ls -a -q)

删除数据卷：docker volume rm $(docker volume ls -q)

删除 network：docker network rm $(docker network ls -q)

# Docker-compose 常用命令

docker-compose --help你会看到如下这么多命令 (什么？看不懂？我也看不懂)

build Build or rebuild services

bundle Generate a Docker bundle from the Compose file

config Validate and view the Compose file

create Create services

down Stop and remove containers, networks, images, and volumes

events Receive real time events from containers

exec Execute a command in a running container

help Get help on a command

images List images

kill Kill containers

logs View output from containers

pause Pause services

port Print the public port for a port binding

ps List containers

pull Pull service images

push Push service images

restart Restart services

rm Remove stopped containers

run Run a one-off command

scale Set number of containers for a service

start Start services

stop Stop services

top Display the running processes

unpause Unpause services

up Create and start containers

version Show the Docker-Compose version information

常用命令 **(nginx 示例服务名**)

docker-compose up -d nginx 构建建启动nignx容器

docker-compose exec nginx bash 登录到nginx容器中

docker-compose down 删除所有nginx容器,镜像

docker-compose ps 显示所有容器

docker-compose restart nginx 重新启动nginx容器

docker-compose run --no-deps --rm php-fpm php -v

在php-fpm中不启动关联容器，并容器执行php -v 执行完成后删除容器

docker-compose build nginx 构建镜像 。

docker-compose build --no-cache nginx 不带缓存的构建。

docker-compose logs nginx 查看nginx的日志

docker-compose logs -f nginx 查看nginx的实时日志

docker-compose config -q

验证（docker-compose.yml）文件配置，当配置正确时，不输出任何内容，当文件配置错误，输出错误信息。

docker-compose events --json nginx 以json的形式输出nginx的docker日志

docker-compose pause nginx 暂停nignx容器

docker-compose unpause nginx 恢复ningx容器

docker-compose rm nginx 删除容器（删除前必须关闭容器）

docker-compose stop nginx 停止nignx容器

docker-compose start nginx 启动nignx容器

# 编写dockerfile

示例：

FROM java:8 #基础镜像

MAINTAINER ZXX <1139835238@qq.com> #镜像信息，可随意写

ENV TZ=Asia/Shanghai #设置环境变量

#将当前目录中的 name.jar 复制到容器根目录中

ADD ycs-ms-service-zzpay-v1.jar /app.jar

#运行命令

#RUN chmod 777 -R /nacos

RUN ln -snf /usr/share/zoneinfo/$TZ /etc/localtime

#开放端口

#EXPOSE 8848 #监听端口 没啥用

#ENTRYPOINT ["sh","/nacos/bin/startup.sh","-m","standalone","&"]

#启动运行命令

ENTRYPOINT ["java","-jar","/app.jar","&"]

# Docker-Compose.yml 编写

version: '3' #docker-compose 版本号

services: # 服务列表

usercenter: # 服务名

image: 10.253.254.36/yuncetest/usercenter:2.0 #镜像来源

restart: always #始终重启

container\_name: "uc\_test" #容器名

volumes: # 挂载卷 可挂载多个 用 – 开头

- /home/app/logs:/logs

- /a:/b

network\_mode: "host" #容器网络模式 host ：宿主机模式

datacache:

image: 10.253.254.36/yuncetest/datacache:2.0

restart: always

container\_name: "dc\_test"

volumes:

- /home/app/logs:/logs

network\_mode: "host"

healthmaster:

image: 10.253.254.36/yuncetest/healthmaster:2.0

restart: always

container\_name: "hm\_test"

volumes:

- /home/app/logs:/logs

network\_mode: "host"