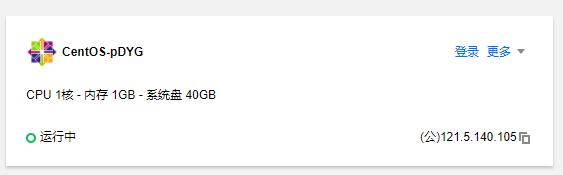
**《基于centos7做docker容器化部署》**

**朱春旭**

**2021.03.16**

# 所需软硬件环境准备

## 云服务器



## Xshell 6

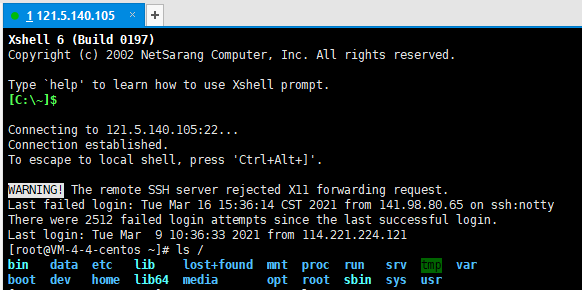


## Xftp 6



# 一．准备docker环境

## 进入操作系统



## 查看内核版本

uname -a

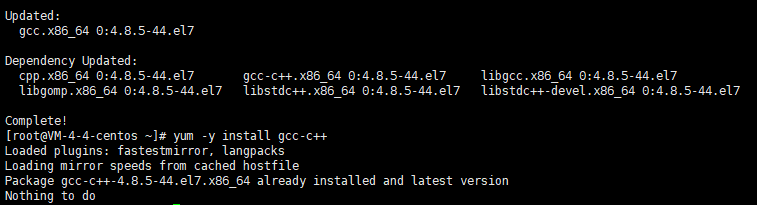


Docker部署要求64位的系统且内核版本至少为3.10

## 安装gcc环境

yum -y install gcc

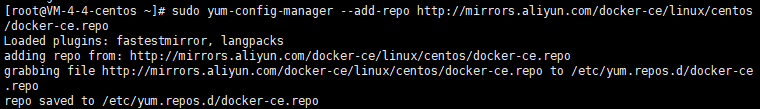
yum -y install gcc-c++



## 配置阿里云yum镜像

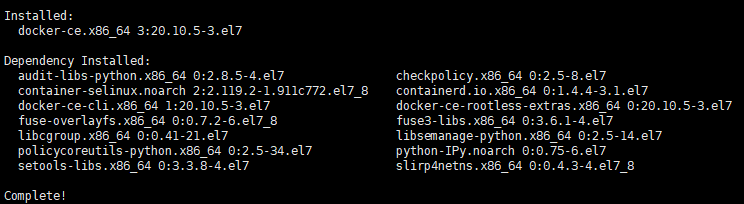
yum-config-manager --add-repo [http://mirrors.aliyun.com/docker-ce/linux/centos/docker-ce.repo](http://mirrors.aliyun.com/docker-ce/linux/centos/docker-ce.repo" \t "https://www.cnblogs.com/mazhilin/p/_blank)

如果报没有yum-config-manager命令需要先执行yum -y install yum-utils



## 安装docker

yum install docker-ce



## 设置开机自启动

systemctl enable docker



## 启动docker服务

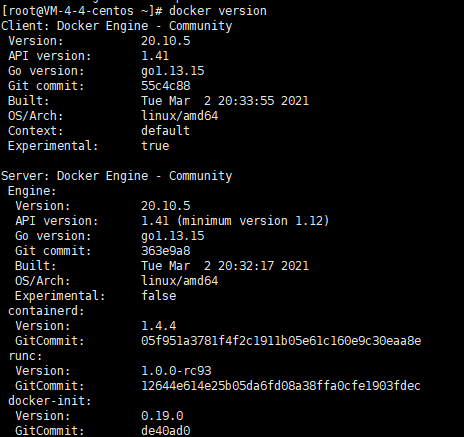
systemctl start docker



基本命令：  
systemctl enable docker # 允许开机启动  
systemctl start docker #运行Docker守护进程  
systemctl stop docker #停止Docker守护进程  
systemctl restart docker #重启Docker守护进程  
systemctl status docker #查看Docker的运行状态

## 查看docker版本

docker version

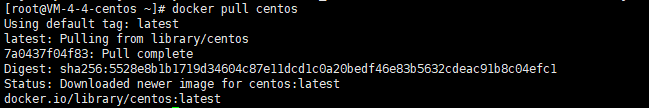


# 二．Dockerfile构建镜像

以构建仅包含nginx的镜像为例

## 1.下载centos基础镜像

docker pull centos



## 创建并进入工作目录

mkdir /opt/nginx

cd /opt/nginx/



## 从官网下载linux版的nginx



## 编写nginx.conf

该文件包含了自己业务所需的配置



## 制作Dockerfile

为了方便，这里直接从windows系统编写好

----------------------------------------------------------------------------------------------------------------

FROM centos

MAINTAINER zcx

ADD nginx-1.19.8.tar.gz /opt

WORKDIR /opt/nginx-1.19.8

RUN yum -y install gcc pcre-devel openssl-devel make \

&& groupadd www-data && useradd -s /sbin/nologin -g www-data www-data \

&& ./configure \

--prefix=/usr/local/nginx \

--user=www-data \

--group=www-data \

--with-pcre \

--with-http\_v2\_module \

--with-http\_ssl\_module \

--with-http\_realip\_module \

--with-http\_addition\_module \

--with-http\_sub\_module \

--with-http\_dav\_module \

--with-http\_flv\_module \

--with-http\_mp4\_module \

--with-http\_gunzip\_module \

--with-http\_gzip\_static\_module \

--with-http\_random\_index\_module \

--with-http\_secure\_link\_module \

--with-http\_stub\_status\_module \

--with-http\_auth\_request\_module \

--with-mail \

--with-mail\_ssl\_module \

--with-file-aio \

--with-http\_v2\_module \

--with-threads \

--with-stream \

--with-stream\_ssl\_module && make && make install

ENV PATH /usr/local/nginx/sbin:$PATH

COPY nginx.conf /usr/local/nginx/conf/nginx.conf

EXPOSE 80

ENTRYPOINT ["nginx"]

CMD ["-g","daemon off;"]

----------------------------------------------------------------------------------------------------------------



## 使用ftp上传到工作目录



## 生成镜像

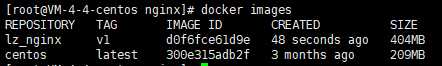
docker build -t lz\_nginx:v1 .

过程无报错即可构建成功并自动命名



## 查看镜像列表

docker images



## 启动容器

docker run -p 80:80 --name nginx-80 -d lz\_nginx:v1



## 查看容器列表

docker ps -a



其他常用命令

停止容器：docker stop容器id

删除容器：docker rm 容器id

删除镜像：docker rmi 镜像id

# 三．阿里云镜像服务

## 1.开通服务并创建仓库



命名空间 ：取公司名称

仓库名称 ：取项目名称

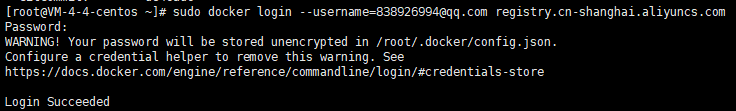
镜像版本 ：不同版本的项目实例

关系：命名空间下有多个仓库，每个仓库下有多个镜像版本

## 2.登录阿里云镜像服务

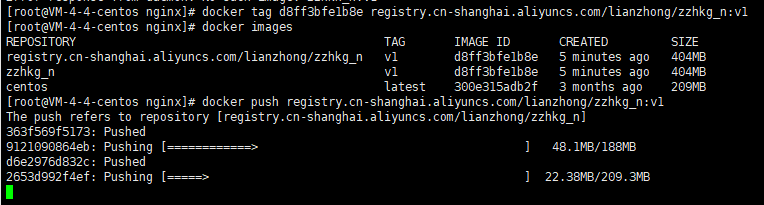


推送和拉取操作都需要先进行登录，参考阿里云镜像仓库下的操作指南



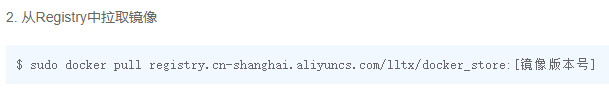
## 推送镜像到仓库

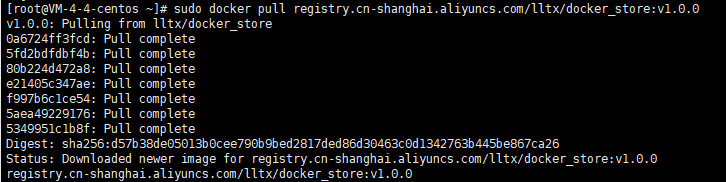






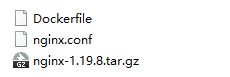
## 从仓库拉取镜像





# 四．常用镜像示例

## 1.nginx镜像



**Dockerfile**

FROM centos

MAINTAINER zcx

ADD nginx-1.19.8.tar.gz /opt

WORKDIR /opt/nginx-1.19.8

RUN yum -y install gcc pcre-devel openssl-devel make \

&& groupadd www-data && useradd -s /sbin/nologin -g www-data www-data \

&& ./configure \

--prefix=/usr/local/nginx \

--user=www-data \

--group=www-data \

--with-pcre \

--with-http\_v2\_module \

--with-http\_ssl\_module \

--with-http\_realip\_module \

--with-http\_addition\_module \

--with-http\_sub\_module \

--with-http\_dav\_module \

--with-http\_flv\_module \

--with-http\_mp4\_module \

--with-http\_gunzip\_module \

--with-http\_gzip\_static\_module \

--with-http\_random\_index\_module \

--with-http\_secure\_link\_module \

--with-http\_stub\_status\_module \

--with-http\_auth\_request\_module \

--with-mail \

--with-mail\_ssl\_module \

--with-file-aio \

--with-http\_v2\_module \

--with-threads \

--with-stream \

--with-stream\_ssl\_module && make && make install

ENV PATH /usr/local/nginx/sbin:$PATH

COPY nginx.conf /usr/local/nginx/conf/nginx.conf

EXPOSE 8080

ENTRYPOINT ["nginx"]

CMD ["-g","daemon off;"]

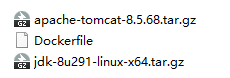
**nginx.conf**

一般的nginx配置即可，按自己的需求修改

**nginx-1.19.8.tar.gz**

从官网上下载一个安装包，后缀必须是tar.gz

## tomcat镜像



**Dockerfile**

FROM centos

MAINTAINER zcx

ADD apache-tomcat-7.0.90.tar.gz /root

ADD jdk-8u291-linux-x64.tar.gz /root

ENV JAVA\_HOME /root/jdk1.8.0\_291

ENV PATH $JAVA\_HOME/bin:$PATH

#define entry point which will be run first when the container starts up

ENTRYPOINT /root/apache-tomcat-7.0.90/bin/startup.sh && tail -F /root/apache-tomcat-7.0.90/logs/catalina.out

**apache-tomcat-7.0.90.tar.gz**

从官网下载

**jdk-8u291-linux-x64.tar.gz**

从官网下载

## 3.mysql镜像



docker pull mysql:5.7.26

**my.cnf**

[mysqld]

default-character-set = utf8

character\_set\_server = utf8

lower\_case\_table\_names = 1

[mysql]

default-character-set = utf8

# 五．常用手动操作

## 将外部文件复制到容器根目录

docker cp /opt/lz-mysql/a.sql 容器id:/

常用于将web项目复制到tomcat的webapps目录下，或是复制my.cnf到mysql目录下。如果不是要复制到根目录，而是指定目录，就把最后的/改成指定路径。

## 启动mysql容器并挂载目录

sudo docker run --restart=always --name lz\_mysql -p 3306:3306 -v /opt/lz-mysql/conf/my.cnf:/etc/mysql/my.cnf -v /opt/lz-mysql/logs:/logs -v /opt/lz-mysql/data:/var/lib/mysql -e MYSQL\\_ROOT\\_PASSWORD=zhlz\_2021 -d mysql:5.7.26

--restart=always : docker重启时自动启动容器

--name : 容器命名

-p : 主机端口和容器内部端口映射

-v : 目录挂载

-d : 来源镜像

目录挂载是将mysql容器内的目录映射到主机的指定目录上，实际生成的数据也都在主机的目录下。如果不做目录挂载，mysql容器一旦被删除，之前的数据就全部没了（实质就是将容器与数据分离，容器的删除不会影响数据，当容器再次被部署时仍然可以通过目录挂载链接上以前的数据）；

## 启动tomcat容器并挂载目录

docker run --restart=always -p 8080:8080 --name lz\_tomcat -v /opt/lz-tomcat/webapps:/root/apache-tomcat-7.0.90/webapps --link lz\_mysql:db -d lz\_tomcat:7.0.9

--link : 连通两个容器

解释：

1. 在这个过程中，挂载目录的目的是防止容器被删除，项目也被删除，如果把项目挂载到主机目录上，容器被删除但项目不会被删除，当容器再次部署时仍然可以通过挂载目录的方式链接上以前的项目。
2. 为什么要连通两个容器？要知道tomcat和mysql是部署在两个容器里的，例如tomcat下部署的web项目肯定是要在jdbc配置文件中连接mysql的，那么ip地址写什么？localhost:3306这种写法是不行的。一种做法是查看mysql容器的ip地址，写在里面，但是使用--link方式会更好。--link lz\_mysql:db表示让tomcat容器连通名为“lz\_mysql”的容器并命名一个别名为db，则jdbc里写成db:3306就可以了。

## 进入容器内部

docker exec -it 容器id bash

如果是基于centos做的镜像（例如nginx，tomcat），则进入容器内部后就像操作linux系统一样操作，nginx或tomcat的安装目录就在linux系统里。

如果是mysql镜像，进入后即可用mysql -uroot -p连接mysql，就像windows系统的dos窗口那样操作。

## 镜像打包

docker save -o xxx.tar 镜像id

## 从tar包加载镜像

docker load xxx.tar

docker save 和 load一般配合使用，适用于要部署的服务器对网络做了限制，不能使用pull命令拉取官方镜像，可以先在能够使用pull命令的机器上拉取目标镜像，再用save打包成文件，通过ftp上传到目标服务器上使用load加载。