

docker问题 此docker问题为容器启动问题

改变storage driver类型，禁用容器的selinux

```
systemctl stop docker
```

清理镜像

```
rm -rf /var/lib/docker
```

修改存储类型

```
vi /etc/sysconfig/docker-storage
```

把空的DOCKER_STORAGE_OPTIONS参数改为overlay:

```
DOCKER_STORAGE_OPTIONS="--storage-driver overlay"
```

禁用selinux

```
vi /etc/sysconfig/docker
```

去掉option的-selinux-enabled

启动docker应该就可以了

```
systemctl start docker
```

docker搭建仓库(htpasswd 认证)

1, 拉取docker registry 镜像

```
docker pull registry
```

2, 创建证书存放目录

```
mkdir -p /home/registry/certs
```

3, 生成CA证书

Edit your /etc/ssl/openssl.cnf on the logstash host - add subjectAltName = IP:内网的IP地址 in [v3_ca] section.

一般情况下，证书只支持域名访问，要使其支持IP地址访问，需要修改配置文件openssl.cnf。

在redhat7系统中，openssl.cnf文件所在位置是/etc/pki/tls/openssl.cnf。在其中的[v3_ca]部分，添加subjectAltName选项：

```
[ v3_ca ]
subjectAltName = IP:外网的IP地址
```

生成证书

```
openssl req -newkey rsa:4096 -nodes -sha256 \
-keyout /home/registry/certs/domain.key -x509 \
-days 365 -out /home/registry/certs/domain.crt
```

注意Common Name最好写为registry的域名

修改权限，并将认证文件添加到(客户端) /etc/docker/certs.d/内网的IP地址:5000/

```
chcon -Rt svirt_sandbox_file_t /home/registry/certs
mkdir -p /etc/docker/certs.d/IP地址:5000/
cp /home/registry/certs/domain.crt /etc/docker/certs.d/内网的IP地址:5000/ca.crt
```

3, 使用registry镜像生成用户名和密码文件

```
mkdir /home/registry/auth
cd /home/registry/auth
echo "user:账号 passwd:密码" >htpasswd //htpasswd可以修改但是后面语句上都要修改
docker run --entrypoint htpasswd registry -Bbn 账号 密码 >
/home/registry/auth/htpasswd
chcon -Rt svirt_sandbox_file_t /home/registry/
```

4, 运行registry并指定参数。包括了用户密码文件和CA书位置。--restart=always 始终自动重启

```
docker run -d -p 5000:5000 --restart=always --name registry \
-v /home/registry/auth:/auth \
-e "REGISTRY_AUTH=htpasswd" \
-e "REGISTRY_AUTH_HTPASSWD_REALM=Registry Realm" \
-e REGISTRY_AUTH_HTPASSWD_PATH=/auth/htpasswd \
-v /home/registry/certs:/certs \
-e REGISTRY_HTTP_TLS_CERTIFICATE=/certs/domain.crt \
-e REGISTRY_HTTP_TLS_KEY=/certs/domain.key \
registry
```

5, 登陆和登出

###vim /etc/hosts

```
docker login 内网的IP地址:5000 -u uesr -p password
docker logout 内网的IP地址:5000
```

6, 添加用户

```
docker run --entrypoint htpasswd registry -Bbn Dapeng 123456 >>
/home/registry/auth/htpasswd
docker run --entrypoint htpasswd registry -Bbn user123 passwd123 >>
/home/registry/auth/htpasswd
```

无需执行 docker restart registry

上传镜像到仓库并从局域网内其它docker连接下载

vi /etc/docker/daemon.json

```
{
  "registry-mirrors": [""], #镜像加速地址
  "insecure-registries": ["内网的IP地址","等等非https的仓库"], # Docker如果需要从非https源管理镜像, 这里加上。
}
```

使用docker-compose文件搭建dnmp

步骤大概：： 制作Nginx mysql php的镜像

制作完成后使用docker compose 的compose.yml文件为3个镜像创建容器 并挂载卷到宿主机 并与宿主机相互连接 最终形成一个 docker + nginx + mysql + php-fpm的环境运行web程序

相对于制作镜像再写compose.yml文件创建docker容器运行环境， docker 分布式架构的优点是能通用 这是分布式架构的特点 如果没有特殊情况 使用互联网带来的compose.yml文件创建容器执行dnmp环境是最佳快速方案

分布式架构使用docker容器构建了环境后， 向其他机器部署环境直接使用镜像就好 也可以上传到之前搭建的docker仓库中

安装docker compose

```
sudo curl -L
"https://get.daocloud.io/docker/compose/releases/download/1.24.1/docker-
compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
```

如有需要，修改上面 1.24.1 为指定版本号即可

安装完后执行：

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

文件结构

下载对应的环境源码

```
wget http://nginx.org/download/nginx-1.16.1.tar.gz
wget https://libzip.org/download/libzip-1.2.0.tar.gz
wget https://www.php.net/distributions/php-7.3.9.tar.gz
```

```
root@test-docker1:/data/compose_lnmpp# tree -L 2
.
├── docker-compose.yml
├── mysql
│   ├── conf
│   └── data
├── nginx
│   ├── Dockerfile
│   ├── nginx-1.16.0.tar.gz
│   └── nginx.conf
├── php
│   ├── Dockerfile
│   ├── libzip-1.2.0.tar.gz
│   ├── php-7.3.9.tar.gz
│   └── php.ini
└── wwwroot
    ├── index.html
    └── test.php
```

compose部署文件

```
version: '3'
```

```

services:
  nginx:
    hostname: nginx
    build:
      context: ./nginx
      dockerfile: Dockerfile
    ports:
      - "80:80"
    links:
      - php:php-cgi
    volumes:
      - ./wwwroot:/usr/local/nginx/html

  php:
    hostname: php
    build: ./php
    links:
      - mysql:mysql-db
    volumes:
      - ./wwwroot:/usr/local/nginx/html

  mysql:
    hostname: mysql
    image: mysql:5.7
    ports:
      - "3306:3306"
    volumes:
      - ./mysql/conf:/etc/mysql/conf.d
      - ./mysql/data:/var/lib/mysql
    environment:
      MYSQL_ROOT_PASSWORD: 1qaz!QAZ
      MYSQL_DATABASE: ws1in
      MYSQL_USER: ws1in
      MYSQL_PASSWORD: 1qaz!QAZ

```

其中hostname为容器主机名 build为创建方式 port为映射端口 link为连接的其他容器 volumes为挂载容器的文件 image为创建容器的镜像 environment容器创建结束cmd的命令

PS:links php:php-cgi意思是链接到服务名为php的服务，可以使用host名 php-cgi访问该容器，在启动容器后进入容器ping

mysql/conf/my.cnf

```

[mysqld]
user=mysql
port=3306
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
pid-file=/var/run/mysqld/mysqld.pid
log_error=/var/log/mysql/error.log
character_set_server = utf8
max_connections=3600

```

nginx/Dockerfile

```

FROM centos:7
MAINTAINER wslin
RUN yum install -y gcc-c++ zlib-devel pcre-devel make
ADD nginx-1.16.1.tar.gz /tmp
RUN cd /tmp/nginx-1.16.1 && ./configure --prefix=/usr/local/nginx && make -j 2
&& make install
RUN rm -f /usr/local/nginx/conf/nginx.conf
COPY nginx.conf /usr/local/nginx/conf
RUN chmod -R 777 /usr/local/nginx/html
EXPOSE 80
CMD ["/usr/local/nginx/sbin/nginx", "-g", "daemon off;"]

```

nginx/nginx.conf

```

user root;
worker_processes auto;
error_log logs/error.log info;
pid logs/nginx.pid;
events {
    use epoll;
}
http {
    include mime.types;
    default_type application/octet-stream;

    log_format main '$remote_addr - $remote_user [$time_local] "$request" '
        '$status $body_bytes_sent "$http_referer" '
        '"$http_user_agent" "$http_x_forwarded_for"';

    access_log logs/access.log main;
    sendfile on;
    keepalive_timeout 65;

    server {
        listen 80;
        server_name localhost;
        root html;
        index index.html index.php;

        location ~ \.php$ {
            root html;
            fastcgi_pass php-cgi:9000;
            fastcgi_index index.php;
            fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
            include fastcgi_params;
        }
    }
}

```

php/Dockerfile

```

FROM centos:7
MAINTAINER wslin
RUN yum -y install epel-release gcc gcc-c++

```

```

RUN yum install -y libxml2 libxml2-devel openssl openssl-devel bzip2 bzip2-devel
libcurl libcurl-devel libjpeg libjpeg-devel libpng libpng-devel freetype
freetype-devel gmp gmp-devel libmcrypt libmcrypt-devel readline readline-devel
libxslt libxslt-devel zlib zlib-devel glibc glibc-devel glib2 glib2-devel
ncurses curl gdbm-devel db4-devel libxpm-devel libx11-devel gd-devel gmp-devel
expat-devel xmlrpc-c xmlrpc-c-devel libicu-devel libmcrypt-devel
libmemcached-devel libsqlite3x-devel oniguruma-devel make perl
ADD libzip-1.2.0.tar.gz /tmp
RUN cd /tmp/libzip-1.2.0 && \
    ./configure && \
    make && \
    make install
ADD php-7.3.9.tar.gz /tmp
RUN echo "/usr/local/lib64">>/etc/ld.so.conf && \
    echo "/usr/local/lib">>/etc/ld.so.conf && \
    echo "/usr/lib">>/etc/ld.so.conf && \
    echo "/usr/lib64">>/etc/ld.so.conf && \
    ldconfig -v
RUN cd /tmp/php-7.3.9 && \
    ./configure --prefix=/usr/local/php --with-config-file-
path=/usr/local/php/etc --with-curl --with-freetype-dir --enable-gd --with-
gettext --with-iconv-dir --with-kerberos --with-libdir=lib64 --with-libxml-dir -
-with-mysqli --with-openssl --with-pcre-regex --with-pdo-mysql --with-pdo-sqlite
--with-pear --with-png-dir --with-jpeg-dir --with-xmlrpc --with-xsl --with-zlib
--with-bz2 --with-mhash --enable-fpm --enable-bcmath --enable-libxml --enable-
inline-optimization --enable-mbregex --enable-mbstring --enable-opcache --
enable-pcntl --enable-shmop --enable-soap --enable-sockets --enable-sysvsem --
enable-sysvshm --enable-xml --enable-zip --enable-fpm && \
    cp /usr/local/lib/libzip/include/zipconf.h /usr/local/include/ && \
    make -j 4 && make install && \
    cp /usr/local/php/etc/php-fpm.conf.default /usr/local/php/etc/php-fpm.conf
&& \
    cp /usr/local/php/etc/php-fpm.d/www.conf.default /usr/local/php/etc/php-
fpm.d/www.conf && \
    sed -i "s/127.0.0.1/0.0.0.0/g" /usr/local/php/etc/php-fpm.d/www.conf && \
    cp ./sapi/fpm/init.d.php-fpm /etc/init.d/php-fpm && \
    chmod +x /etc/init.d/php-fpm
#COPY php.ini /usr/local/php/etc
EXPOSE 9000
CMD /etc/init.d/php-fpm start && tail -F /var/log/messages

```

PS: 1, CentOS7自带libzip版本较低需要下载编译安装

2, 需要把默认配置文件[www.conf.default](#)重命名为www.conf并且修改配置把127.0.0.1改成0.0.0.0否则会导致无法访问php页面, 因为启动了不同的容器无法通过127.0.0.1访问

3, 配置文件php.ini可以不拷贝

因为centos7selinux安全子系统问题 将所有相关的挂载目录权限开启

添加selinux规则 按照上面挂载的文件夹位置为./wwwroot 安装脚本当前位置的wwwroot以及mysql的conf和data文件夹

```
chcon -Rt svirt_sandbox_file_t 挂载的文件夹位置
```

运行docker compose

```
docker-compose up -d
```

docker教程

<https://www.cnblogs.com/zhujingzhi/category/1292035.html>