

使用Nginx实现服务器中多容器共存

背景

有一台腾讯的Linux云主机，在服务器上部署了一个docker（称为ServiceDocker，名称为sign，下同），ServiceDocker内部使用了80、443、3306端口，分别映射到宿主机（云主机）的相同端口（即80、443、3306）。

ServiceDocker中安装了XAMPP，在此基础上部署了扫码签到小程序的服务器。ServiceDocker的80和443端口分别提供http和https服务，3306为MySQL数据库的端口。

ServiceDocker绑定了域名sign.famend.cn。

目标

云主机中的一个ServiceDocker占用了80、443端口，如果想在主机中再部署一台服务器对外服务，就不能使用80和443端口了。

我们的目标是在主机中可以部署多个ServiceDocker，每个ServiceDocker绑定各自的域名，对外提供服务，保证80和443端口可用。

思路

修改ServiceDocker的端口映射，把ServiceDocker端口80、443分别映射到主机的89、449端口，这样主机的80、443端口释放出来了。

释放出来的80和443端口供Nginx使用。安装带Nginx的docker(称为NginxDocker，名称为mynginx，下同)。NginxDocker内部使用端口80、443，分别映射到主机的80和443端口。

NginxDocker用来做反向代理。当有访问请求时，读取Nginx配置后，不同的URL定向到各自对应的Docker。例如：访问<http://sign.famend.cn:80>，则自动映射到<http://sign.famend.cn:89>。

å®·žçŽ°æ-¥éª

1.修改ServiceDocker的端口映射，释放80和443端口。

Docker没有提供命令修改端口，从网上我找到了2种方法。

方法1：先停止容器，再将容器打包成镜像，接着运行新的镜像。在运行新的镜像时指定新的端口。使用的命令如下：

```
#先停止容器
```

```
docker stop containerA
```

```
#将容器commit成为一个镜像
```

```
docker commit containerA newImageB
```

```
#运行容器
```

```
docker run -p 8080:8080 -p 8081:8081 -v /home/data:/home/data/ -dt newImageB
```

方法2：先停止容器，接着停止容器服务，然后修改容器配置文件，最后启动容器服务、启动容器。步骤如下：

① 停止ServiceDocker (ServiceDocker的名称为sign) , 停止docker服务。

```
sudo docker stop sign  
sudo service docker stop
```

- ② 使用 `docker ps -a` 命令找到要修改容器的 CONTAINER ID。
- ③ 运行 `docker inspect 【CONTAINER ID】 | grep Id` 命令。
- ④ 执行 `cd /var/lib/docker/containers` 命令进入找到与 Id 相同的目录。
在执行cd 命令时如果提示permission denied , 可以先执行sudo -s。
进入id对应的目录后, 打开文件hostconfig.json。
找到80端口的映射, 如下:

```
"80/tcp": [  
{  
  "HostIp": "0.0.0.0",  
  "HostPort": "80"  
}]
```

把" HostPort" : "80" 修改成 "HostPort" : "89" , 如下

```
"80/tcp": [  
{  
  "HostIp": "0.0.0.0",  
  "HostPort": "89"  
}]
```

修改之前, ServiceDocker内部的80端口映射为主机的80端口; 修改之后, ServiceDocker内部的80端口映射为主机的89端口。

稍加说明, 有文章 (参考资料中的2、3) 提到还需要修改config.v2.json , 亲测不需要 (如果Docker内部的端口没有改变, 那么就不需要修改此文件) 。这个文件在启动ServiceDocker时会自动修改。

⑤ 启动docker服务, 然后启动ServiceDocker (名称为sign) 。

```
sudo service docker start  
sudo docker start sign
```

方法(1)和方法(2)都可以, 我选用了方法(2)。
在执行完方法(2)之后, 打开浏览器验证, 提示 "网站无法访问" 。

初步估计docker中的服务器没有启动，运行命令：

```
sudo /opt/lampp/lampp stop
sudo /opt/lampp/lampp start
```

运行stop的时候，发现apache 没有启动，可能是修改了80端口的缘故。

在执行start之后，打开浏览器验证<http://sign.famend.cn:89>和<https://sign.famend.cn:449>，成功访问。

修改Docker端口的方法并不复杂，如果docker以后的版本能够提供相应命令，相信会方便很多。

顺便，运行 `service crontab status`

查看ServiceDocker中的定时任务是否启动，如果没启动，运行 `service cron start` 启动定时任务。

BTW: `crontab -l` 查看定时任务，`crontab -e` 编辑定时任务。

下一步，配置NginxDocker反向代理，让<http://sign.famend.cn:80>和<https://sign.famend.cn:443>也能成功访问。

2.配置NginxDocker反向代理。

① 下载nginx，并运行它。

```
docker container run \
-d \
-p 80:80 \
-p 443:443 \
--rm \
--name mynginx \
nginx
```

② 配置nginx的配置文件。

```
mkdir nginx-files
docker container cp mynginx:/etc/nginx .
mv nginx conf
vi conf/nginx.conf
```

在nginx.conf中，添加如下反向代理信息。

```
server{
listen      443 ssl;
server_name sign.famend.cn;
```

```

ssl_certificate
/etc/nginx/ssl/sign.famend.cn/1_sign.famend.cn_bundle.crt;
ssl_certificate_key /etc/nginx/ssl/sign.famend.cn/2_sign.famend.cn.key;
#ssl_certificate /etc/nginx/ssl/famend.cn/1_famend.cn_bundle.crt;
#ssl_certificate_key /etc/nginx/ssl/famend.cn/2_famend.cn.key;
location / {
proxy_set_header HOST $host;
proxy_set_header X-Real-IP $remote_addr;
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
proxy_set_header X-Forwarded-Proto $scheme;
proxy_pass http://sign.famend.cn:89/;
}
}
server{
listen 80;
server_name famend.cn sign.famend.cn;
location / {
proxy_set_header HOST $host;
proxy_set_header X-Real-IP $remote_addr:89;
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
proxy_set_header X-Forwarded-Proto $scheme;
proxy_pass http://sign.famend.cn:89/;
}
}

```

对于443端口，需要使用ssl_certificate和ssl_certificate_key。在ServiceDocker中证书使用的LetsEncrypt的SSL证书，每90天更新一次。

SSL证书的来源：一种方法，可以让它们共享LetsEncrypt的SSL证书。除此之外，在腾讯注册的域名，腾讯免费提供SSL证书，有效期为1年。

为了简便，我直接使用了腾讯的SSL证书，当然，一年之内，必须在过期之前更新证书。

③ 停止mynginx，然后重启它。

```

docker container run \
--name mynginx \
--volume "$PWD/conf":/etc/nginx \
-p 80:80 \
-p 443:443 \
-d \
nginx

```

这次，去掉 `-rm` 参数，以便停止运行的时候保留容器。
至此，配置完成。

éªœè•

浏览器中分别打开

sign.famend.cn:80
sign.famend.cn:89
sign.famend.cn:449
sign.famend.cn:443

都能正常访问。当然，打开449和443时候，可以发现两个URL使用的证书不一样。449是LetsEncrypt提供的，有效期90天；443来自腾讯（颁发机构TrustAsia），有效期1年。

当然，ServiceDocker中的另外一个网站famend.cn，也可以访问：

famend.cn:80
famend.cn:89

成功。

参考资料

Docker更改容器端口映射
修改docker容器端口映射的方法
通过配置文件修改docker容器端口映射
Nginx 容器教程
crontab 定时任务