

Table of Contents

1. [企业服务负载均衡架构](#)
2. [实验步骤](#)

企业服务负载均衡架构

用户在浏览器上多次访问nginx服务器，nginx将流量轮询传递给两台web服务器，实现了负载均衡，页面登录产生的session信息交给memcache镜像处理，用户数据通过mysql镜像存储在宿主机磁盘中，实现数据的持久化。

实验步骤

1. 首先进入一次nginx容器，然后将其配置文件复制到宿主机的 **/root/data/nginx** 中

```
docker run -it -v /root/data/nginx:/data --rm nginx \ncp /etc/nginx/conf.
```

2. 修改default.conf

```
upstream 192.168.1.110{
    server web1;
    server web2;
}
server {
    listen      80;
    server_name localhost;

    #access_log  /var/log/nginx/host.access.log  main;

    location / {
        proxy_pass http://192.168.1.110;
        # root    /usr/share/nginx/html;
        # index   index.html index.htm;
    }

    #error_page  404              /404.html;

    # redirect server error pages to the static page /50x.html
    #
    error_page   500 502 503 504  /50x.html;
    location = /50x.html {
```

```

        root    /usr/share/nginx/html;
    }

    # proxy the PHP scripts to Apache listening on 127.0.0.1:80
    #
    #location ~ /\.php$ {
    #    proxy_pass    http://127.0.0.1;
    #}

    # pass the PHP scripts to FastCGI server listening on 127.0.0.1:9000
    #
    location ~ /\.php$ {
        proxy_pass    http://192.168.1.110;
        #    root            html;
        #    fastcgi_pass    127.0.0.1:9000;
        #    fastcgi_index   index.php;
        #    fastcgi_param   SCRIPT_FILENAME    /scripts$fastcgi_script_name;
        #    include          fastcgi_params;
    }

    # deny access to .htaccess files, if Apache's document root
    # concurs with nginx's one
    #
    #location ~ /\.ht {
    #    deny    all;
    #}
}

```

3. 构建web服务器

```

FROM php:5.6-apache
RUN mv "$PHP_INI_DIR/php.ini-production" "$PHP_INI_DIR/php.ini" \
    && apt-get update && apt-get install -y libmemcached-dev zlib1g-dev \
    && pecl install memcached-2.2.0 \
    && docker-php-ext-install mysql && docker-php-ext-enable mysql \
    && docker-php-ext-install mysqli && docker-php-ext-enable mysqli \
    && docker-php-ext-enable memcached

```

问题：实验文档中需要创建 **/usr/local/etc/php/conf.d**，在实际操作过程中提示文件夹已存在，所以我去除了这条命令。

1. 将php配置文件复制到 **/root/data/phpini** 中

```
docker run --rm -v /root/data/phpini:/data myphp cp -r /usr/local/etc/php/
```

2. 修改 **php.ini** 文件

```
session.save_handler = memcache  
session.save_path = "tcp://session:11211"
```

3. 启动web容器

```
docker run -d -p 81:80 \\\n-v /root/data/html:/var/www/html \\\n-v /root/dat
```

img img img

4. 正式开始搭建网站

◦ session缓存容器

```
docker run -d --restart=always --name session memcached
```

◦ mariadb数据库容器

```
docker run --name db --restart=always \\\n-v /root/data/mysql:/var/lib/
```

◦ 启动web服务容器

```
docker run -d --link db:db --link session:session \\\n-v /root/data/htr
```

```
docker run -d --link db:db --link session:session \\\n-v /root/data/htr
```

◦ 启动nginx容器

```
docker run -d --restart=always --name lb -p 80:80 -v /root/data/nginx:
```

◦ 创建数据库

```
root in ~/data/phpini took 3s >>> docker run -d --link db:db -p 81:80 \
>>> -v /root/data/html \
>>> -v /root/data/phpini:/usr/local/etc/php --restart=always --name web1 myphp
e49b62c56a53817e64fd608c742c882b3489cf2675f2b687ade2531bd1670004
root in ~/data/phpini >>> docker ps -a
CONTAINER ID   IMAGE                                COMMAND                                  CREATED        STATUS
US            PORTS                                NAMES
e49b62c56a53   myphp                                "docker-php-entrypoi..." About a minute ago Up A
bout a minute  0.0.0.0:81->80/tcp, :::81->80/tcp web1
cd35fd030337   mariadb:5.5.64                      "docker-entrypoint.s..." 9 minutes ago  Up 9
minutes       3306/tcp                             db
root in ~/data/phpini >>> █
```

- 将typecho文件复制入 **root/data/html** 下，更改html文件夹读写权限

```
chmod 777 /root/data/html
```

- 然后设置管理员帐号密码等，安装完成

