# System Administration HW7

B04705003 資工三 林子雋

## 1 Reference

### 1.1 Lab

#### 1.1.1 概念解釋

1. https://www.linuxbabe.com/it-knowledge/differences-between-forward-proxy-and-revers

#### 1.1.2 Apache

1. 第二十章、WWW 伺服器
2. [Apache] 2.4.6 Virtual Host 設定
3. Error: "ssl error rx record too long" in Firefox after installing the SSL certificate
4. SSL received a record that exceeded the maximum permissible length. (Error code: ssl error rx record too long) item HomeApacheRHEL / CentOS 7 安裝 Apache mod_ssl RHEL / CentOS 7 安裝 Apache mod_ssl
5. apache 配置中 ProxyPassReverse 指令的含义

## 2 Problem

### 2.1 Lab

#### 2.1.1 概念解釋

1. The main difference between proxy server and reverse proxy is that the first one is used by the client but the latter is used by the server such as a web server.
   Proxy server is also known as forward proxy. One of its application is that you can use proxy server to hide your real IP behind because you can use the proxy server's IP as your source IP when you visit websites.
   Reverse proxy is mainly used by the server to balance load and protect against web-based attack like DoS attack.

2. By Table 1, we can see that the advantage of nginx is that its model can handle more connections than Apache when they have same resources. However, the

disadvantage of nginx is its sophisticated architecture and developers need to be careful to create efficiency code. But for Apache, it is easier to develop due to its simple structure.
To see which one is better to act as a reverse proxy, apparently, **nginx is more suitable** since it has event-based model rather than thread-per-connection-based or process-per-connection-based model.

| | nginx | Apache |
|---|---|---|
| Model | event-driven model | process-or-thread-per-connection model |
| Development Complexity | more complex | simple |
| Role | as a proxy of a web server or a web server | mostly web server but also support reverse prox |

Table 1: Nginx and Apache comparison

#### 2.1.2 Apache

1. httpd use `mod_proxy` module. Moreover, all modules configuration files are loaded in `/etc/httpd/http.conf`'s `Include conf.modules.d/*.conf` line, where `conf.modules.d`'s `00-proxy.conf` states to load `mod_proxy.so` module.
2. Added directives are written down in the below [**BONUS PART**] section.
   (a) **ProxyPass**: By using this directive, clients' requests can be converted by apache docker container to http://web-app1/index.html
   (b) **ProxyPassReverse**: By using this directive, clients' responses containing http://web-app1/index.html can be successfully converted back to /NASA/.
3. Just simply run `systemctl restart httpd`

[**BONUS PART**]

**Procedure to build a docker image**
**Environment**: docker lab's ova virtual machine.

```
cd ~
# Build a image based on TA's preinstalled centos
mkdir apache-reverse-proxy
# Create docker file
vi Dockerfile
################
Dockerfile:
FROM centos
```

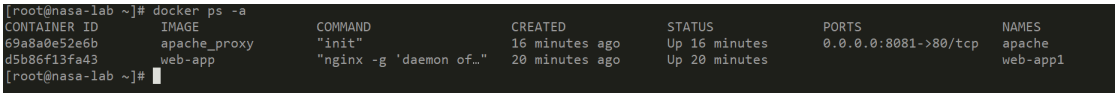


Figure 1: Virtualbox docker containers

```
RUN yum -y install httpd
################
# Run web-app1
docker run -d --network private --name web-app1 web-app
# Go into container to configure things
docker run -d --network private --name tmp \
   --privileged apache_proxy init
docker exec -it apache "/bin/bash"
# Configure httpd.conf file
vi /etc/httpd/conf/httpd.conf
###############
httpd.conf:
ProxyPass "/NASA/" "http://web-app1/index.html"
ProxyPassReverse "/NASA/" "http://web-app1/index.html"
###############
# Restart httpd service
systemctl restart httpd
systemctl enable httpd
# Commit changes
docker commit -m "add␣proxy,␣enable␣httpd" tmp apache_proxy
# Run new image
docker run -d --network private --name apache --privileged \
  --pubish 8081:80 apache_proxy init
```

**Show other machines cannot reach web-app1:**
**Enviroment:** See Figure 1 for Virtualbox's docker containers. See Figrue 2 for IP setting. See Figure 3 for `curl` result Because web-app1 doesn't publish its port to the virtual host, we can only reach virtual host's apache reverse proxy in my Windows host.

```
# In host machine (in my case, Windows 10)
curl 192.168.107.3:8081/NASA/
```

## 2.2 Directive

1. Add `DocumentRoot` directive.
2. Add `Require` directive.

```
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:c8:d1:02 brd ff:ff:ff:ff:ff:ff
    inet 192.168.107.3/24 brd 192.168.107.255 scope global dynamic enp0s8
       valid_lft 645sec preferred_lft 645sec
    inet6 fe80::147c:945e:d5cb:fc3a/64 scope link
       valid_lft forever preferred_lft forever
```

Figure 2: Virtual host's IP address

```
C:\Users\USER
λ curl 192.168.107.3:8081/NASA/
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>I am behind reverse proxy!!</title>
  </head>

  <body>
    <h1>I am behind reverse proxy!!</h1>
    <em>How AWESOME NASA is! Isn't it?</em>
  </body>

</html>

C:\Users\USER
λ
```

Figure 3: curl result

3. Add `Redirect` directive.

```
cd /etc/httpd/conf.d
vi virtual.conf
#################
<VirtualHost *:80>
        ServerName www.example.com
        DocumentRoot /var/www/html
        <Directory "/var/www/html/nasa">
                Require ip 192.168.1.105
        </Directory>
         Redirect / "https://www.example.com/"
</VirtualHost>
#################
systemctl restart httpd
```