NSD Operation DAY02

1 案例1：搭建Nginx服务器

1.1 问题

在IP地址为192.168.4.5的主机上安装部署Nginx服务，并可以将Nginx服务器，要求编译时启用如下功能：

- SSL加密功能
- 设置Nginx账户及组名称均为nginx

可选项：Nginx服务器升级到更高版本。

然后客户端访问页面验证Nginx Web服务器：

- 使用火狐浏览器访问
- 使用curl访问

1.2 方案

使用2台RHEL6虚拟机，其中一台作为Nginx服务器（192.168.4.5）、另外一台作为测试用的Linux客户机（192.168.4.100），如图-1所示。

Top

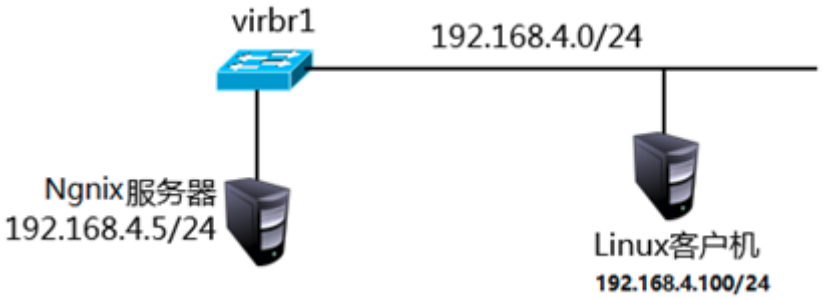

图-1

安装nginx-1.8.0版本时，需要使用如下参数：

- with-http_ssl_module：提供SSL加密功能
- user：指定账户
- group：指定组

1.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：构建Nginx服务器

1）使用源码包安装nginx软件包

```
[root@svr5 ~]# yum –y install gcc pcre-devel openssl-devel      //安装常见依赖包
[root@svr5 ~]# useradd –s /sbin/nologin nginx
[root@svr5 ~]# tar -xf nginx-1.8.0.tar.gz
[root@svr5 ~]# cd nginx-1.8.0
[root@svr5 nginx-1.8.0]# ./configure \
>--prefix=/usr/local/nginx \            //指定安装路径
>--user=nginx \                         //指定用户
>--group=nginx \                        //指定组
>--with-http_ssl_module                 //开启SSL加密功能
```

Top

```
10.      .. ..
11.    nginx path prefix: "/usr/local/nginx"
12.      nginx binary file: "/usr/local/nginx/sbin/nginx"
13.      nginx configuration prefix: "/usr/local/nginx/conf"
14.      nginx configuration file: "/usr/local/nginx/conf/nginx.conf"
15.      nginx pid file: "/usr/local/nginx/logs/nginx.pid"
16.      nginx error log file: "/usr/local/nginx/logs/error.log"
17.      nginx http access log file: "/usr/local/nginx/logs/access.log"
18.      nginx http client request body temporary files: "client_body_temp"
19.      nginx http proxy temporary files: "proxy_temp"
20.      nginx http fastcgi temporary files: "fastcgi_temp"
21.      nginx http uwsgi temporary files: "uwsgi_temp"
22.      nginx http scgi temporary files: "scgi_temp"
23.    [root@svr5 nginx-1.7.10]# make && make install    //编译并安装
```

2）nginx命令的用法

```
01.    [root@svr5 ~]# /usr/local/nginx/sbin/nginx                //启动服务
02.    [root@svr5 ~]# /usr/local/nginx/sbin/nginx -s stop        //关闭服务
03.    [root@svr5 ~]# /usr/local/nginx/sbin/nginx -s reload      //重新加载配置文件
04.    [root@svr5 ~]# /usr/local/nginx/sbin/nginx –V             //查看软件信息
```

nginx服务默认通过TCP 80端口监听客户端请求：

Top

```
01.    [root@svr5 ~]# netstat -anptu | grep nginx
02.    tcp        0      0 0.0.0.0:80          0.0.0.0:*           LISTEN      10441/nginx
```

3）为Nginx Web服务器建立测试首页文件

Nginx Web服务默认首页文档存储目录为/usr/local/nginx/html/，在此目录下建立一个名为index.html的文件：

```
[root@svr5 ~]# cat  /usr/local/nginx/html/index.html
<html>
<head>
<title>Welcome to nginx! </title>
</head>
<body bgcolor="white" text="black">
<center><h1>Welcome to nginx! </h1></center>
</body>
</html>
```

步骤二：升级Nginx服务器

1）编译新版本nginx软件

```
[root@svr5 ~]# tar -zxvf  nginx-1.9.0.tar.gz
[root@svr5 ~]# cd nginx-1.9.0
[root@svr5 nginx-1.9.0]# ./configure  \
>--prefix=/usr/local/nginx  \
>--user=nginx  \
>--group=nginx  \
>--with-http_ssl_module
[root@svr5 nginx-1.9.0]# make
```

Top

2) 备份老的nginx主程序，并使用编译好的新版本nginx替换老版本

```
[root@svr5 nginx-1.9.0]# mv /usr/local/nginx/sbin/nginx \
>/usr/local/nginx/sbin/nginxold
[root@svr5 nginx-1.9.0]# cp objs/nginx /usr/local/nginx/sbin/    //拷贝新版本
[root@svr5 nginx-1.9.0]# make upgrade                            //升级
/usr/local/nginx/sbin/nginx -t
nginx: the configuration file /usr/local/nginx/conf/nginx.conf syntax is ok
nginx: configuration file /usr/local/nginx/conf/nginx.conf test is successful
kill -USR2 `cat /usr/local/nginx/logs/nginx.pid`
sleep 1
test -f /usr/local/nginx/logs/nginx.pid.oldbin
kill -QUIT `cat /usr/local/nginx/logs/nginx.pid.oldbin`
[root@svr5 ~]# /usr/local/nginx/sbin/nginx –v                    //查看版本
```

步骤三：客户端访问测试

1）分别使用浏览器和命令行工具curl测试服务器页面

```
[root@client ~]# firefox http://192.168.4.5
[root@client ~]# curl http://192.168.4.5
```

2 案例2：用户认证

Top

2.1 问题

沿用练习一，通过调整Nginx服务端配置，实现以下目标：

1. 访问Web页面需要进行用户认证
2. 用户名为：tom，密码为：123456

2.2 方案

通过Nginx实现Web页面的认证，需要修改Nginx配置文件，在配置文件中添加auth语句实现用户认证。最后使用htpasswd命令创建用户及密码即可。

2.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：修改Nginx配置文件

1）修改/usr/local/nginx/conf/nginx.conf

```
[root@pc205 ~]# vim /usr/local/nginx/conf/nginx.conf
.. ..
server {
        listen       80;
        server_name  localhost;
        auth_basic "Input Password: ";        //认证提示符
        auth_basic_user_file "/usr/local/nginx/pass";        //认证密码文件
        location / {
            root   html;
            index  index.html index.htm;
        }
    }
```

Top

2）生成密码文件，创建用户及密码

使用htpasswd命令创建账户文件，需要确保系统中已经安装了httpd-tools。

```
[root@svr5 ~]# yum -y install httpd-tools
[root@svr5 ~]# htpasswd -cm /usr/local/nginx/pass tom        //创建密码文件
New password:
Re-type new password:
Adding password for user tom
[root@svr5 ~]# htpasswd -m /usr/local/nginx/pass jerry
//追加用户，不使用-c选项
New password:
Re-type new password:
Adding password for user jerry
```

3）重启Nginx服务

```
[root@svr5 ~]# /usr/local/nginx/sbin/nginx –s reload
//请先确保nginx是启动状态才可以执行命令成功，否则报错
```

步骤二：客户端测试

1）登录192.168.4.100客户端主机进行测试

```
[root@client ~]# firefox http://192.168.4.5              //输入密码后可以访问
```

Top

3 案例3：基于域名的虚拟主机

3.1 问题

沿用练习二，配置基于域名的虚拟主机，实现以下目标：

1. 实现两个基于域名的虚拟主机，域名分别为www.aa.com和www.bb.com
2. 对域名为www.aa.com的站点进行用户认证，用户名称为tom，密码为123456

3.2 方案

修改Nginx配置文件，添加server容器实现虚拟主机功能；对于需要进行用户认证的虚拟主机添加auth认证语句。

3.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：修改配置文件

1）修改Nginx服务配置，添加相关虚拟主机配置如下

```
[root@svr5 ~]# vim /usr/local/nginx/conf/nginx.conf
.. ..
server {
        listen       80;                                  //端口
        server_name  www.aa.com;                          //域名
auth_basic "Input Password: ";                            //认证提示符
        auth_basic_user_file "/usr/local/nginx/pass";     //认证密码文件
location / {
            root   html;                                  //指定网站根路径
            index  index.html index.htm;
        }

}
… …
```

Top

```
15.
16.         server {
17.             listen  80;                                  //端口
18.             server_name  www.bb.com;                              //域名
19.     location / {
20.     root   www;                             //指定网站根路径
21.     index  index.html index.htm;
22.     }
23.     }
```

2）创建账户及密码

```
01.     [root@svr5 ~]# htpasswd –cm  /usr/local/nginx/pass  tom       //创建账户密码文件
02.     New password:
03.     Re-type new password:
04.     Adding password for user tom
```

3）创建网站根目录及对应首页文件

```
01.     [root@svr5 ~]# mkdir  /usr/local/nginx/www
02.     [root@svr5 ~]# echo "www" > /usr/local/nginx/www/index.html
```

4）重启nginx服务

Top

```
[root@svr5 ~]# /usr/local/nginx/sbin/nginx -s reload
```

步骤二：客户端测试

1）修改客户端主机192.168.4.100的/etc/hosts文件，进行域名解析

```
[root@client ~]# vim /etc/hosts
192.168.4.5    www.aa.com www.bb.com
```

2）登录192.168.4.100客户端主机进行测试

注意：SSH –X远程连接调用虚拟机的firefox时，请先关闭真实机的firefox。

```
[root@client ~]# firefox http://www.aa.com        //输入密码后可以访问
[root@client ~]# firefox http://www.bb.com        //直接访问
```

4 案例4：SSL虚拟主机

4.1 问题

沿用练习二，配置基于加密网站的虚拟主机，实现以下目标：

1. 域名为www.cc.com
2. 该站点通过https访问
3. 通过私钥、证书对该站点所有数据加密

Top

4.2 方案

源码安装Nginx时必须使用--with-http_ssl_module参数，启用加密模块，对于需要进行SSL加密处理的站点添加ssl相关指令（设置网站需要的私钥和证书）。

4.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：配置SSL虚拟主机

1）生成私钥与证书

```
[root@svr5 ~]# cd /usr/local/nginx/conf
[root@svr5 ~]# openssl genrsa -out cert.key                    //生成私钥
[root@svr5 ~]# openssl req -new -x509 -key cert.key -out cert.pem     //生成证书
```

2）修改Nginx配置文件，设置加密网站的虚拟主机

```
[root@svr5 ~]# vim /usr/local/nginx/conf/nginx.conf
…… ……
server {
        listen       443 ssl;
        server_name  www.cc.com;
        ssl_certificate      cert.pem;
        ssl_certificate_key  cert.key;

        ssl_session_cache    shared:SSL:1m;
        ssl_session_timeout  5m;

        ssl_ciphers  HIGH:!aNULL:!MD5;
```

Top

```
13.             ssl_prefer_server_ciphers  on;
14.
15.             location / {
16.                 root   html;
17.                 index  index.html index.htm;
18.             }
19.         }
```

步骤二：客户端验证

1）修改客户端主机192.168.4.100的/etc/hosts文件，进行域名解析

```
01.     [root@client ~]# vim /etc/hosts
02.     192.168.4.5    www.cc.com  www.aa.com   www.bb.com
```

2）登录192.168.4.100客户端主机进行测试

```
01.     [root@client ~]# firefox https://www.cc.com          //信任证书后可以访问
```

5 案例4：Nginx反向代理

5.1 问题

使用Nginx实现Web反向代理功能，实现如下功能：

Top

- 后端Web服务器两台，可以使用httpd实现
- Nginx采用轮询的方式调用后端Web服务器
- 两台Web服务器的权重要求设置为不同的值

- 最大失败次数为1，失败超时时间为30秒

5.2 方案

使用4台RHEL7虚拟机，其中一台作为Nginx代理服务器，该服务器需要配置两块网卡，IP地址分别为192.168.4.5和192.168.2.5，两台Web服务器IP地址分别为192.168.2.100和192.168.2.200。客户端测试主机IP地址为192.168.4.100。如图-2所示。

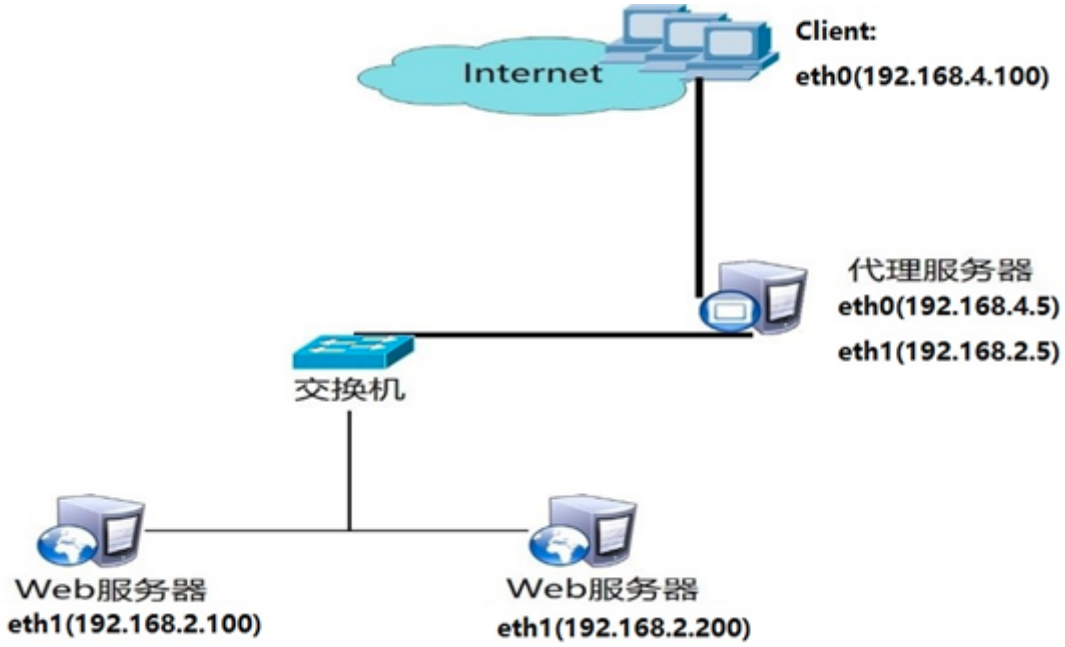

图-2

5.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：部署实施后端Web服务器

1）部署后端Web1服务器

后端Web服务器可以简单使用yum方式安装httpd实现Web服务，为了可以看出后端服务器的不同，可以将两台后端服务器的首页文档内容设置为不同的内容。

Top

```
[root@web1 ~]# yum -y install httpd
[root@web1 ~]# echo "192.168.2.100" > /var/www/html/index.html
[root@web1 ~]# systemctl restart httpd
```

2）部署后端Web2服务器

```
[root@web2 ~]# yum -y install httpd
[root@web2 ~]# echo "192.168.2.200" > /var/www/html/index.html
[root@web2 ~]# systemctl restart httpd
```

步骤二：配置Nginx服务器，添加服务器池，实现反向代理功能

1）修改/usr/local/nginx/conf/nginx.conf配置文件

```
[root@svr5 ~]# vim /usr/local/nginx/conf/nginx.conf
.. ..
http {
.. ..
upstream webserver {
        server 192.168.2.100 weight=1 max_fails=2 fail_timeout=10;
        server 192.168.2.200 weight=2 max_fails=2 fail_timeout=10;
    }
.. ..
server {
    listen       80;
```

Top

```
12.             server_name  www.tarena.com;
13.                 location / {
14.                 proxy_pass http://webserver;
15.             }
16.     }
```

2）重启nginx服务

```
01.     [root@svr5 ~]# /usr/local/nginx/sbin/nginx –s reload
```

3）使用浏览器访问代理服务器测试轮询效果

```
01.     [root@client ~]# curl http://192.168.4.5          //使用该命令多次访问查看效果
```

步骤二：配置upstream服务器集群池属性

1）设置失败次数，超时时间，权重

```
01.     [root@svr5 ~]# vim /usr/local/nginx/conf/nginx.conf
02.     ……
03.     http {
04.     ……
05.     upstream webserver {
06.                 server 192.168.2.100 weight=1 max_fails=2 fail_timeout=10;
07.                 server 192.168.2.200 weight=2 max_fails=2 fail_timeout=10;
```

Top

```
        }
    .. ..
    server {
        listen       80;
        server_name  www.tarena.com;
            location / {
            proxy_pass http://webserver;
        }
    }
```

2）重启nginx服务

```
[root@svr5 ~]# /usr/local/nginx/sbin/nginx -s reload
```

3）使用浏览器访问代理服务器测试轮询效果

```
[root@client ~]# curl http://192.168.4.5          //使用该命令多次访问查看效果
```

4）设置相同客户端访问相同Web服务器

```
[root@svr5 ~]# vim /usr/local/nginx/conf/nginx.conf
.. ..
http {
.. ..
```

Top

```
upstream webserver {
                ip_hash;
                server 192.168.2.100 weight=1 max_fails=2 fail_timeout=10;
                server 192.168.2.200 weight=2 max_fails=2 fail_timeout=10;
        }
.. ..
server {
        listen       80;
        server_name  www.tarena.com;
            location / {
            proxy_pass http://webserver;
        }
}
```

5）重启nginx服务

```
[root@svr5 ~]# /usr/local/nginx/sbin/nginx –s reload
```

6）使用浏览器访问代理服务器测试轮询效果

```
[root@client ~]# curl http://192.168.4.5          //使用该命令多次访问查看效果
```

Top