**DAY1**

一，安装Nginx

1，首先，确认虚拟机的yum源可用；并设置防火墙为trusted，SELinux为关闭状态

2，确认lnmp\_soft.tar.gz ； nginx-1.12.2.tar.gz这2个压缩包在虚拟机里；

3，开始安装Nginx

#安装相应的依赖包

[root@proxy ~]# yum -y install gcc pcre-devel openssl-devel

#创建对应的用户

[root@proxy ~]# useradd -s /sbin/nologin nginx

#解压软件包

[root@proxy ~]# tar -xf lnmp\_soft.tar.gz

[root@proxy ~]# cd lnmp\_soft/

[root@proxy lnmp\_soft]# tar -xf nginx-1.10.3.tar.gz

#编译、安装

[root@proxy lnmp\_soft]# cd nginx-1.10.3/

[root@proxy nginx-1.10.3]# ./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

[root@proxy nginx-1.10.3]# make && make install

二，Nginx常见命令的用法

#启动服务

[root@proxy nginx-1.10.3]# /usr/local/nginx/sbin/nginx

#关闭服务

[root@proxy nginx-1.10.3]# /usr/local/nginx/sbin/nginx -s stop

#重新加载配置文件

[root@proxy nginx-1.10.3]# /usr/local/nginx/sbin/nginx -s reload

#查看软件信息

[root@proxy nginx-1.10.3]# /usr/local/nginx/sbin/nginx -V

#创建软链接，方便后期使用

[root@proxy nginx-1.10.3]# ln -s /usr/local/nginx/sbin/nginx /sbin/

三，用nginx测试命令是否可用

[root@proxy nginx-1.10.3]# nginx

四，用netstat命令查看nginx服务信息

[root@proxy nginx-1.10.3]# netstat -anutlp | grep nginx

tcp 0 0 0.0.0.0:80 0.0.0.0:\* LISTEN 6101/nginx:..

补充：netstat的各选项意思

-a 显示所有端口的信息

-n 以数字格式显示端口号

-t 显示TCP连接的端口

-u 显示UDP连接的端口

-l 显示服务正在监听的端口信息，如httpd启动后，会一直监听80端口

-p 显示监听端口的服务名称是什么（也就是程序名称）

五，客户端测试

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

案例一，升级Nginx

[root@proxy ~]# tar -xzvf nginx-1.12.2.tar.gz

[root@proxy ~]# cd nginx-1.12.2/

[root@proxy nginx-1.12.2]# ./configure \

> --prefix=/usr/local/nginx \

> --user=nginx \

> --group=nginx \

> --with-http\_ssl\_module

#注意，此处只能make，不可以make install！！！ 切记！切记！切记

[root@proxy nginx-1.12.2]# make

#备份老的配置文件

[root@proxy nginx-1.12.2]# mv /usr/local/nginx/sbin/nginx /usr/local/nginx/sbin/nginxold

#拷贝新版本

[root@proxy nginx-1.12.2]# cp objs/nginx /usr/local/nginx/sbin/

#查看升级后的版本

[root@client nginx-1.12.2]# nginx -V

nginx version: nginx/1.12.2

案例二，用户认证

目的：访问WEB页面需要进行用户认证；用户名为tom，密码为123456

#备份主配置文件，防止在修改出错时又找不到错误原因，可及时的恢复原配置文件

[root@proxy nginx-1.12.2]# cp /usr/local/nginx/conf/nginx.conf /usr/local/nginx/conf/nginx.conf.bak

#修改主配置文件 （添加auth\_basic开头的这两行）

[root@proxy nginx-1.12.2]# vim /usr/local/nginx/conf/nginx.conf

server {

listen 80;

server\_name localhost;

auth\_basic "shuru:"; #认证提示符（也就是在web页面上显示的）

auth\_basic\_user\_file "/usr/local/nginx/pass"; #认证密码文件

location / {

root html;

index index.html index.htm;

}

生成密码文件前，先确保系统已安装httpd-tools

[root@proxy nginx-1.12.2]# yum -y install httpd-tools

#创建密码文件，并设置tom，密码123456

[root@proxy nginx-1.12.2]# htpasswd -c /usr/local/nginx/pass tom

New password:

Re-type new password:

Adding password for user tom

#如果是追加用户，则不需要-c选项

[root@proxy nginx-1.12.2]# htpasswd /usr/local/nginx/pass jerry

New password:

Re-type new password:

Adding password for user jerry

#验证密码文件是否成功

[root@proxy nginx-1.12.2]# cat /usr/local/nginx/pass

tom:$apr1$R.D1Zjr9$d4zQMmFQVVLvcNTSLrNs11

jerry:$apr1$Rj3MaYdq$Xj/UHwaDopyCpHi.DcQmc/

#重启Nginx服务。（重启前确保Nginx是运行状态，否则会报错）

[root@proxy nginx-1.12.2]# nginx -s reload

#客户端验证

[root@client nginx-1.12.2]#firefox http://192.168.4.5

案例三，基于域名的虚拟主机

目标：

1，实现2个基于域名的虚拟主机，域名分别为www.a.com和www.b.com

2，对域名为www.a.com的站点进行用户认证，用户名称为tom，密码为123456

步骤一：

#修改主配置文件

#一个server就是一个虚拟主机

#root 为网页根目录

[root@proxy ~]# vim /usr/local/nginx/conf/nginx.conf

server {

listen 80;

server\_name www.a.com; #把原来的修改为www.a.com

auth\_basic "shuru:";

auth\_basic\_user\_file "/usr/local/nginx/pass";

location / {

root html;

index index.html index.htm;

}

error\_page 500 502 503 504 /50x.html;

location = /50x.html {

root html;

}

}

#新增一个server，可以把上面的复制、粘贴下来，并做相应的修改

server {

listen 80;

server\_name www.b.com; #把原来的修改为www.b.com

location / {

root www;

index index.html index.htm;

}

}

步骤二：

#创建网页根目录以及对应的首页文件

[root@proxy ~]# mkdir /usr/local/nginx/www

[root@proxy ~]# echo "wwww" > /usr/local/nginx/www/index.html

步骤三：

#重启nginx服务，客户端测试

1,先修改client客户端的/etc/hosts文件，进行域名解析

[root@client ~]# vim /etc/hosts

192.168.4.5 www.a.com www.b.com

2，client测试

[root@client ~]# firefox http://www.a.com

[root@client ~]# firefox http://www.b.com

案例四，拓展内容

一，基于端口的虚拟主机

步骤一：

#基于上一个实验，修改主配置文件

#在前面实验的基础上，新增2个新的server

[root@proxy ~]# vim /usr/local/nginx/conf/nginx.conf

server {

listen 8080;

server\_name web1.example.com;

location / {

root web1;

index index.html index.htm;

}

}

server {

listen 8000;

server\_name web1.example.com;

location / {

root web2;

index index.html index.htm;

}

}

步骤二：

#创建对应的网页根目录，以及首页文件

[root@proxy ~]# mkdir /usr/local/nginx/web1 /usr/local/nginx/web2

[root@proxy ~]# echo "web1" > /usr/local/nginx/web1/index.html

[root@proxy ~]# echo "web2" > /usr/local/nginx/web2/index.html

步骤三：

#重启nginx服务，并在客户端client测试

[root@proxy ~]# nginx -s reload

#在client上，/etc/hosts文件里添加一个域名解析

[root@client ~]# vim /etc/hosts

192.168.4.5 web1.example.com

[root@client ~]# firefox http://web1.example.com:8080

[root@client ~]# firefox http://web1.example.com:8000

二，基于IP的虚拟主机

步骤一：

#要配置基于IP的虚拟主机，必须要先给web服务器proxy的网卡设备添加2个IP别名（也就是一个网卡绑定多个IP）

#首先，执行ifconfig，查看网卡设备名称

[root@proxy ~]# ifconfig

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500

inet 192.168.4.5 netmask 255.255.255.0 broadcast 192.168.4.255

#我们可以看到网卡设备名称为 eth0

#下面，我们给该设备添加IP别名

#添加IP别名1

[root@proxy ~]# ifconfig eth0:1 192.168.4.6 broadcast 192.168.4.255 netmask 255.255.255.0 up

[root@proxy ~]# route add -host 192.168.4.6 dev eth0:1

[root@proxy ~]# ifconfig

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500

inet 192.168.4.5 netmask 255.255.255.0 broadcast 192.168.4.255

eth0:1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500

inet 192.168.4.6 netmask 255.255.255.0 broadcast 192.168.4.255

#添加IP别名2

[root@proxy ~]# ifconfig eth0:2 192.168.4.7 broadcast 192.168.4.255 netmask 255.255.255.0 up

[root@proxy ~]# route add -host 192.168.4.7 dev eth0:2

[root@proxy ~]# ifconfig

[root@proxy ~]# ifconfig

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500

inet 192.168.4.5 netmask 255.255.255.0 broadcast 192.168.4.255

eth0:1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500

inet 192.168.4.6 netmask 255.255.255.0 broadcast 192.168.4.255

eth0:2: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500

inet 192.168.4.7 netmask 255.255.255.0 broadcast 192.168.4.255

#到此，配置完成！！！

步骤二：

#修改主配置文件，在前面的基础之上新增2个server

[root@proxy ~]# vim /usr/local/nginx/conf/nginx.conf

server {

listen 192.168.4.6:80;

server\_name 192.168.4.6;

location / {

root web3;

index index.html index.htm;

}

}

server {

listen 192.168.4.7:80;

server\_name 192.168.4.7;

location / {

root web4;

index index.html index.htm;

}

}

步骤三：

#创建对应的网页根目录，以及首页文件

[root@proxy ~]# mkdir /usr/local/nginx/web3 /usr/local/nginx/web4

[root@proxy ~]# echo "web3" > /usr/local/nginx/web3/index.html

[root@proxy ~]# echo "web4" > /usr/local/nginx/web4/index.html

步骤四：

#重启nginx服务，并在客户端测试

[root@proxy ~]# nginx -s reload

[root@client ~]# firefox http://192.168.4.6

[root@client ~]# firefox http://192.168.4.7

思考题：

在配置基于IP的虚拟主机时，如果2个server\_name后面写的是同一个域名，客户端的/etc/hosts里也写了对应的域名解析。

那么，我们在测试时，输入的不是IP地址，而是域名时，会出现什么问题？原因是什么？？

案例五，SSL虚拟主机

沿用基于域名的虚拟主机，配置加密网站，目标为：

1，域名为www.a.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

步骤一：

#生成私钥与证书

[root@proxy ~]# cd /usr/local/nginx/conf/

#生成私钥匙

[root@proxy conf]# openssl genrsa > cert.key

Generating RSA private key, 2048 bit long modulus

.............................................................................+++

...............................................................................+++

e is 65537 (0x10001)

#生成证书

[root@proxy conf]# openssl req -new -x509 -key cert.key > cert.pem

You are about to be asked to enter information that will be incorporated

into your certificate request.

What you are about to enter is what is called a Distinguished Name or a DN.

There are quite a few fields but you can leave some blank

For some fields there will be a default value,

If you enter '.', the field will be left blank.

-----

Country Name (2 letter code) [XX]:CN #CN为国家

State or Province Name (full name) []:Guangdong #Guangdong 为省份

Locality Name (eg, city) [Default City]:GZ #GZ 为城市

Organization Name (eg, company) [Default Company Ltd]:tedu #tedu为公司名

Organizational Unit Name (eg, section) []:operation #operation为部门名

Common Name (eg, your name or your server's hostname) []:proxy #proxy为主机名

Email Address []:proxy@tedu.cn #proxy@tedu.cn 为邮箱

步骤二：

#修改主配置文件，设置加密网站的虚拟主机

#如果主配置文件有操作上面的虚拟主机，为了防止看着太乱，可以先把上面拓展的2个虚拟主机（端口、IP）的配置都注释掉

#vim批量操作的技巧：

#ctrl+v 一直按着选中要修改的行；再用shift+i 执行要做的操作（我们这边为加#）；然后再按esc键；就一下子全部加了#啦

#把配置最后一个被#注释掉的server开启，并做相应的修改

[root@proxy conf]# vim /usr/local/nginx/conf/nginx.conf

server {

listen 443 ssl;

server\_name www.c.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;

ssl\_prefer\_server\_ciphers on;

location / {

root html;

index index.html index.htm;

}

}

步骤三：

#重启nginx服务

[root@proxy conf]# nginx -s reload

#客户端修改/etc/hosts文件，添加www.c.com

[root@client ~]# vim /etc/hosts

192.168.4.5 www.a.com www.b.com web1.example.com www.c.com

#访问测试

#注意：需要添加信任证书后，才可以正常访问

[root@client ~]# firefox https://www.c.com