operation day 01

================================================

**Nginx（"engine x"）**

**是俄罗斯人编写的十分轻量级的HTTP服务器**

**是一个高性能的HTTP和反向代理服务器，同时也是一个IMAP/POP3/SMTP 代理服务器**

**官方网站：**[**http://nginx.org/**](http://nginx.org/)

**下面使用proxy主机配置nginx：**

**准备工作**

ip add show //查看ip

安装常用工具

yum -y install bash-completion //安装支持tab键的包,装完

之后,要退出登录重新进系统才有效

yum list | grep bash //查询与bash名称有关的软件包

yum -y install net-tools //安装网络相关工具,比如ifconfig

yum -y install psmisc //安装支持killall命令软件包

nmcli connection modify ens33 connection.autoconnect yes //将

ens33网卡设置为开机自启(此处具体网卡名要根据实际填写)

yum -y install vim //安装vim编辑器

**Nginx环境搭建：**

1. 找到lnmp\_soft.tar.gz，拷贝到虚拟机的root家目录，并释放到原地

tar -xf lnmp\_soft.tar.gz

2，释放lnmp\_soft目录中的nginx-1.17.6.tar.gz

cd lnmp\_soft

tar -xf nginx-1.17.6.tar.gz

cd nginx-1.17.6

yum -y install gcc //安装编译工具

yum -y install pcre-devel //可以让nginx支持正则

yum -y install openssl-devel //支持搭建基于ssl(安全加密)技术的网站

3，编译安装

cd nginx-1.17.6 //进入nginx目录

./configure --help | grep http\_ssl //配置时，使用help与grep

查询过滤模块部分名，即可找到模块完整名

./configure --prefix=/usr/local/nginx --user=nginx --with-http\_ssl\_module

//--with-http\_ssl\_module是安全模块, --prefix是安装路径, --user是

指定的用户,nginx运行时拥有该用户的权限

make //编译

make install //安装

**ls /usr/local/nginx //查看nginx安装目录**

**主要目录功能:**

**conf //存放配置文件**

**html //存放网站页面**

**logs //存放日志**

**sbin //主程序目录**

useradd nginx -s /sbin/nologin //创建账户

/usr/local/nginx/sbin/nginx //开启服务

netstat -ntulp | grep :80 //检查80端口

/usr/local/nginx/sbin/nginx -V //查看版本已经添加的功能模块

/usr/local/nginx/sbin/nginx -s stop //关闭服务

/usr/local/nginx/sbin/nginx -s reload //重加载配置文件

**测试网站页面：**

cd /usr/local/nginx

echo "proxy nginx ~~~~" > html/a.html //创建自定义页面

<http://192.168.2.5/a.html> //访问测试

echo "proxy nginx index ~~~~" > html/index.html //修改默认页

[http://192.168.2.5](http://192.168.2.5/a.html) //访问测试

yum -y install unzip //安装解压缩工具

cd /root/lnmp\_soft

unzip www\_template.zip //解压缩网页模板文件

cp -r www\_template/\* /usr/local/nginx/html //拷贝网站模板的文件

到nginx的页面目录，会询问是否覆盖index.html，回答y然后使用

浏览器访问192.168.2.5可以看到该网站

**为nginx增加网站认证功能**

**通常情况下网站搭建好之后，只要知道ip或者域名，那么任何用户都可以访问该网站，如果仅仅想让某些用户访问就可以使用该功能**

1. 首先打开nginx主配置文件,在第42行，43行添加

注意:以下操作都在/usr/local/nginx目录进行

vim conf/nginx.conf

auth\_basic "password:"; //提示信息，用户登录网站时看到的

auth\_basic\_user\_file "/usr/local/nginx/pass"; //存放用户名密码的文件路径

yum -y install httpd-tools //安装网站工具包,支持htpasswd命令

htpasswd -c /usr/local/nginx/pass tom //创建网站的用户与密码文件，第1

个用户名为tom（可以自定义），之后输入2次密码

sbin/nginx -s reload //如果nginx服务已经开启,就重加载配,如果未开启,就开启

即可

htpasswd /usr/local/nginx/pass abc //追加新账户，无需c选项

最后使用火狐浏览器打开192.168.2.5发现已经需要用户名和密码认证

如果要反复测试该功能，需要清空浏览器的历史记录

-----------------------------------------------------------------

**还原nginx配置：**

**在conf目录中存放了nginx.conf.default，这是配置文件的备份，在重新做实验或者配置文件被破坏需要还原是可以用该文件拷贝覆盖nginx.conf**

cd conf/

[root@proxy conf]#

[root@proxy conf]# cp nginx.conf.default nginx.conf

cp：是否覆盖"nginx.conf"？ y

-----------------------------------------------------------

**通常使用一台服务器开启一个nginx服务就可以开启一个网站，但是如果公司需要很多不同域名的网站，而每个网站的业务量不大时，不必购买多台服务器，使用一台服务器利用虚拟主机技术既可以实现。**

配置nginx实现虚拟主机

回顾虚拟主机

httpd中配置:

<virtualhost \*:80>

servername [www.a.com](http://www.a.com)

documentroot /var/www/html

</virtualhost>

<virtualhost \*:80>

servername [www.b.com](http://www.b.com)

documentroot /var/www/html2

</virtualhost>

nginx 虚拟主机

http {

server {

listen :80;

server\_name www.a.com;

root html;

index index.html index.htm;

}

server {

listen :80;

server\_name www.b.com;

root html2;

index index.html index.htm;

}

}

----------------------------------------------------

**创建nginx的虚拟主机**

**1，修改主配置文件，在34~39行开始添加以下内容**

34 server {

35 listen 80; //监听端口

36 server\_name www.b.com; //域名

37 root b; //网页文件存放地

38 index index.html index.htm; //默认页文件名

39 }

另外要把原本的虚拟主机的localhost 修改为 [www.a.com](http://www.a.com)

sbin/nginx -s reload //重加载配置

**2，准备测试页面**

cd /usr/local/nginx

mkdir b //创建b网站的页面存放目录

echo " nginx web A~~~" > html/index.html

echo " nginx web B~~~" > b/index.html

vim /etc/hosts 添加域名与ip的映射关系

192.168.2.5 www.a.com www.b.com www.c.com //注意，如果使用client主

机测试的话，这里的ip要写成192.168.4.5

[root@proxy nginx]# curl [www.a.com](http://www.a.com) //测试a网站

nginx web A~~

[root@proxy nginx]# curl [www.b.com](http://www.b.com) //测试b网站

nginx web B~~

如果在windows环境测试域名访问网站，需要修改hosts文件，添加一样的内

容（hosts文件的权限需要设置为完全控制）

C:\Windows\System32\drivers\etc\hosts

设置hosts的权限要先右键---属性---安全---编辑---users---完全控制勾选--点确定，

然后右键该文件---打开方式---选择文本方式打开

在最后一行下面写入与之前linux中hosts文件里一样的内容

192.168.2.5 [www.a.com](http://www.a.com) [www.b.com](http://www.b.com) [www.c.com](http://www.c.com)

最后使用火狐浏览器访问 [www.a.com](http://www.a.com) 或者 [www.b.com](http://www.b.com)

------------------------------------------------------------------

**搭建基于ssl技术的安全网站**

**1，了解加密算法**

**加密算法一般分为对称算法、非对称算法、信息摘要。**

**对称算法有：AES、DES，主要应用在单机数据加密。**

**非对称算法有：RSA、DSA，主要应用在网络数据加密。**

**信息摘要：MD5、sha256，主要应用在数据完整性校验。**

对称算法 一个密码可以用来加密,同时也用来解密

非对称算法 私钥(钥匙) 公钥(锁) ,公钥用来加密,私钥用来解密

**2，修改主配置文件第103行左右，找安全网站的虚拟主机的配置，将所有注释去掉，可以使用:103,120s/#//**

103 server {

104 listen 443 ssl;

105 server\_name [www.c.com](http://www.c.com); //这里修改域名

106

107 ssl\_certificate cert.pem; //证书文件（包含公钥）

108 ssl\_certificate\_key cert.key; //私钥文件

109

110 ssl\_session\_cache shared:SSL:1m;

111 ssl\_session\_timeout 5m;

112

113 ssl\_ciphers HIGH:!aNULL:!MD5;

114 ssl\_prefer\_server\_ciphers on;

115

116 location / {

117 root c; //这里修改页面存储目录

118 index index.html index.htm;

119 }

120 }

**3，创建私钥与证书**

cd /usr/local/nginx

openssl genrsa > conf/cert.key 创建私钥

openssl req -new -x509 -key conf/cert.key > conf/cert.pem 根据刚刚创建的私钥，再创建证书(包含了公钥),生成过程会询问诸如你在哪个国家之类的问题，可以随意回答，但要走完全过程

Country Name (2 letter code) [XX]:dc 国家名称

State or Province Name (full name) []:dc 省份名称

Locality Name (eg, city) [Default City]:dc 城市名称

Organization Name (eg, company) [Default Company Ltd]:dc 公司名称

Organizational Unit Name (eg, section) []:dc 部门名称

Common Name (eg, your name or your server's hostname) []:dc 服务器名称

Email Address []:dc@dc.com 邮件地址

最后测试:

sbin/nginx -s reload //重新加载配置

[root@proxy nginx]# mkdir c

[root@proxy nginx]# echo "nginx web C~~" > c/index.html

[root@proxy nginx]# curl -k https://www.c.com //访问c网站测试,k选项是忽略风险提示,自己随意创建的加密网站,通常浏览器可能会认为不合法有危险

或使用火狐浏览器访问https://www.c.com/ 看到提示---高级---接收风险并继续

另外，如果实验还是无法成功，可以先用killall nginx杀掉程序，从新开启再测试