**[Top](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201801/OPERATION/DAY01/CASE/01/index.html" \l "page_top_case)**

# NSD Operation DAY01

1. [案例1：搭建Nginx服务器](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201801/OPERATION/DAY01/CASE/01/index.html" \l "case1)
2. [案例2：用户认证](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201801/OPERATION/DAY01/CASE/01/index.html" \l "case2)
3. [案例3：基于域名的虚拟主机](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201801/OPERATION/DAY01/CASE/01/index.html" \l "case3)
4. [案例4：SSL虚拟主机](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201801/OPERATION/DAY01/CASE/01/index.html" \l "case4)

## **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：

* 支持SSL加密功能
* 设置Nginx账户及组名称均为nginx
* 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**

提前准备运维课程所需的所有虚拟机，为后续所有实验做准备，克隆4台RHEL7虚拟机，实验环境所需要的主机及对应的IP设置列表如表-1所示，正确配置IP地址、主机名称，并且为每台主机配置YUM源。不需要配置网关与DNS。

表－1 主机列表



第一天课程需要使用2台RHEL7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。

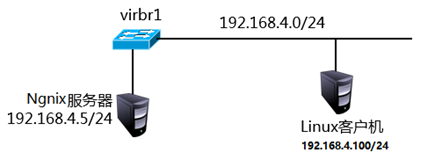


图-1

安装nginx-1.10.3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

1. **[**root@proxy **~]**# yum **-**y install gcc pcre**-**devel openssl**-**devel        //安装依赖包
2. **[**root@proxy **~]**# useradd **-**s **/**sbin**/**nologin nginx
3. **[**root@proxy **~]**# tar **-**xf nginx**-1.10.3.**tar**.**gz
4. **[**root@proxy **~]**# cd nginx**-1.10.3**
5. **[**root@proxy nginx**-1.10.3]**# **./**configure **\**
6. **>** **--**prefix**=**/usr/local**/**nginx **\**                //指定安装路径
7. **>** **--**user**=**nginx **\**                            //指定用户
8. **>** **--**group**=**nginx **\**                            //指定组
9. **>** **--with-**http\_ssl\_module                        //开启SSL加密功能
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@proxy nginx**-1.10.3]**# make **&&** make install    //编译并安装

2）nginx命令的用法

1. **[**root@proxy **~]**# /usr/local**/**nginx**/**sbin**/**nginx                    //启动服务
2. **[**root@proxy **~]**# /usr/local**/**nginx**/**sbin**/**nginx **-**s stop            //关闭服务
3. **[**root@proxy **~]**# /usr/local**/**nginx**/**sbin**/**nginx **-**s reload        //重新加载配置文件
4. **[**root@proxy **~]**# /usr/local**/**nginx**/**sbin**/**nginx –V                //查看软件信息
5. **[**root@proxy **~]**# ln **-**s **/**usr**/**local**/**nginx**/**sbin**/**nginx **/**sbin**/**        //方便后期使用

netstat命令可以查看系统中启动的端口信息，该命令常用选项如下：

-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）

nginx服务默认通过TCP 80端口监听客户端请求：

1. root@proxy **~]**# netstat **-**anptu **|** grep nginx
2. tcp        **0**        **0** **0.0.0.0:80**        **0.0.0.0:\***        LISTEN        **10441/**nginx

3）设置防火墙与SELinux

1. **[**root@proxy **~]**# firewall**-**cmd **--**set**-default-**zone**=**trusted
2. **[**root@proxy **~]**# setenforce **0**

4）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/，在此目录下默认有一个名为index.html的文件，使用客户端访问测试页面：

1. **[**root@client **~]**# curl http**:**//192.168.4.5
2. **<**html**>**
3. **<**head**>**
4. **<**title**>**Welcome to nginx**!</**title**>**
5. **</**head**>**
6. **<**body bgcolor**=**"white" text**=**"black"**>**
7. **<**center**><**h1**>**Welcome to nginx**!<**/h1></center**>**
8. **</**body**>**
9. **</**html**>**

**步骤二：升级Nginx服务器**

1）编译新版本nginx软件

1. **[**root@proxy **~]**# tar **-**zxvf nginx**-1.12.2.**tar**.**gz
2. **[**root@proxy **~]**# cd nginx**-1.12.2**
3. **[**root@proxy nginx**-1.12.2]**# **./**configure **\**
4. **>** **--**prefix**=**/usr/local**/**nginx **\**
5. **>** **--**user**=**nginx **\**
6. **>** **--**group**=**nginx **\**
7. **>** **--with-**http\_ssl\_module
8. **[**root@proxy nginx**-1.12.2]**# make

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

1. **[**root@proxy nginx**-1.12.2]**# mv **/**usr**/**local**/**nginx**/**sbin**/**nginx **\**
2. **>**/usr/local**/**nginx**/**sbin**/**nginxold
3. **[**root@proxy nginx**-1.12.2]**# cp objs**/**nginx **/**usr**/**local**/**nginx**/**sbin**/**     //拷贝新版本
4. **[**root@proxy nginx**-1.12.2]**# make upgrade                            //升级
5. /usr/local**/**nginx**/**sbin**/**nginx **-**t
6. nginx**:** the configuration file **/**usr**/**local**/**nginx**/**conf**/**nginx**.**conf syntax is ok
7. nginx**:** configuration file **/**usr**/**local**/**nginx**/**conf**/**nginx**.**conf test is successful
8. kill **-**USR2 `cat **/**usr**/**local**/**nginx**/**logs**/**nginx**.**pid`
9. sleep **1**
10. test **-**f **/**usr**/**local**/**nginx**/**logs**/**nginx**.**pid**.**oldbin
11. kill **-**QUIT `cat **/**usr**/**local**/**nginx**/**logs**/**nginx**.**pid**.**oldbin`
12. **[**root@proxy **~]**# /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

1. **[**root@client **~]**# firefox http**:**//192.168.4.5
2. **[**root@client **~]**# curl http**:**//192.168.4.5

## **2 案例2：用户认证**

### **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**

模板配置文件框架如下：

1. **[**root@proxy **~]**# vim **/**usr**/**local**/**nginx**/**conf**/**nginx**.**conf
2. 全局配置（用户名**,**日志**,**进程**)**
3. http**{**
4. server**{**
5. listen **80;**
6. server\_name localhost**;**
7. root html**;**
8. **}**
9. server**{**
10. listen **80;**
11. server\_name www**.**xyz**.**com**;**
12. root www**;**
13. **}**
14. **}**

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

1. **[**root@proxy **~]**# vim **/**usr**/**local**/**nginx**/**conf**/**nginx**.**conf
2. **..** **..**
3. server **{**
4. listen **80;**
5. server\_name localhost**;**
6. auth\_basic "Input Password:"**;**                        //认证提示符
7. auth\_basic\_user\_file "/usr/local/nginx/pass"**;**        //认证密码文件
8. location **/** **{**
9. root html**;**
10. index index**.**html index**.**htm**;**
11. **}**
12. **}**

**\*\*两个网页不能放在格式里注意一定在向以下的方式写；和一定要注意括号位置，如server里面有server会出现报错；**

**server{}**

**Server{}**

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。

1. **[**root@proxy **~]**# yum **-**y install httpd**-**tools
2. **[**root@proxy **~]**# htpasswd **-**c **/**usr**/**local**/**nginx**/**pass tom        //创建密码文件
3. New password**:**
4. Re**-**type **new** password**:**
5. Adding password **for** user tom
6. **[**root@proxy **~]**# htpasswd **/**usr**/**local**/**nginx**/**pass jerry     //追加用户，不使用-c选项
7. New password**:**
8. Re**-**type **new** password**:**
9. Adding password **for** user jerry
10. **[**root@proxy **~]**# cat **/**usr**/**local**/**nginx**/**pass

3）重启Nginx服务

1. **[**root@proxy **~]**# /usr/local**/**nginx**/**sbin**/**nginx **-**s reload    //重新加载配置文件
2. #请先确保nginx是启动状态才可以执行命令成功，否则报错**,**报错信息如下：
3. #**[**error**]** open**()** "/usr/local/nginx/logs/nginx.pid" failed **(2:** No such file or directory**)**

**步骤二：客户端测试**

1）登录192.168.4.100客户端主机进行测试

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

## **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.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

### **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。

虚拟主机一般可用分为：基于域名、基于IP和基于端口的虚拟主机。

### **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

1. **[**root@proxy **~]**# vim **/**usr**/**local**/**nginx**/**conf**/**nginx**.**conf
2. **..** **..**
3. server **{**
4. listen **80;**                                     //端口
5. server\_name www**.**a**.**com**;**                             //域名
6. auth\_basic "Input Password:"**;**                        //认证提示符
7. auth\_basic\_user\_file "/usr/local/nginx/pass"**;**        //认证密码文件
8. location **/** **{**
9. root html**;**                                    //指定网站根路径
10. index index**.**html index**.**htm**;**
11. **}**
13. **}**
14. … …
15. server **{**
16. listen **80;**                                        //端口
17. server\_name www**.**b**.**com**;**                             //域名
18. location **/** **{**
19. root www**;**                                 //指定网站根路径
20. index index**.**html index**.**htm**;**
21. **}**
22. **}**

2）创建网站根目录及对应首页文件

1. **[**root@proxy **~]**# mkdir **/**usr**/**local**/**nginx**/**www
2. **[**root@proxy **~]**# echo "www" **>** /usr/local**/**nginx**/**www**/**index**.**html

3）重启nginx服务

1. **[**root@proxy **~]**# /usr/local**/**nginx**/**sbin**/**nginx **-**s reload
2. #请先确保nginx是启动状态才可以执行命令成功，否则报错**,**报错信息如下：
3. #**[**error**]** open**()** "/usr/local/nginx/logs/nginx.pid" failed **(2:** No such file or directory**)**

**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

1. **[**root@client **~]**# vim **/**etc**/**hosts
2. **192.168.4.5**    www**.**a**.**com www**.**b**.**com

2）登录192.168.4.100客户端主机进行测试

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

1. **[**root@client **~]**# firefox http**:**//www.a.com            //输入密码后可以访问
2. **[**root@client **~]**# firefox http**:**//www.b.com            //直接访问

扩展其他虚拟主机：

1.基于端口的虚拟主机（参考模板）

1. server **{**
2. listen **8080;**             //端口
3. server\_name web1**.**example**.**com**;**         //域名
4. **......**
5. **}**
6. server **{**
7. listen **8000;**
8. server\_name web1**.**example**.**com**;**
9. **.......**
10. **}**

2.基于IP的虚拟主机（参考模板）

1. server **{**
2. listen **192.168.0.1:80;**         //端口
3. server\_name web1**.**example**.**com**;**         //域名
4. **...** **...**
5. **}**
6. server **{**
7. listen **192.168.0.2:80;**
8. server\_name web1**.**example**.**com**;**
9. **...** **...**
10. **}**

## **4 案例4：SSL虚拟主机**

### **4.1 问题**

沿用练习三，配置基于加密网站的虚拟主机，实现以下目标：

1. 域名为www.c.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

### **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）。

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、数据秒传等。

### **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

1. **[**root@proxy **~]**# cd **/**usr**/**local**/**nginx**/**conf
2. **[**root@proxy **~]**# openssl genrsa **>** cert**.**key                            //生成私钥
3. **[**root@proxy **~]**# openssl req **-new** **-**x509 **-**key cert**.**key **>** 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

1. **[**root@proxy **~]**# vim **/**usr**/**local**/**nginx**/**conf**/**nginx**.**conf
2. … …
3. server **{**
4. listen **443** ssl**;**
5. server\_name          www**.**c**.**com**;**
6. ssl\_certificate cert**.**pem**;** #这里是证书文件
7. ssl\_certificate\_key cert**.**key**;** #这里是私钥文件
8. ssl\_session\_cache shared**:**SSL**:**1m**;**
9. ssl\_session\_timeout 5m**;**
10. ssl\_ciphers HIGH**:!**aNULL**:!**MD5**;**
11. ssl\_prefer\_server\_ciphers on**;**
12. location **/** **{**
13. root html**;**
14. index index**.**html index**.**htm**;**
15. **}**
16. **}**

3）重启nginx服务

1. **[**root@proxy **~]**# /usr/local**/**nginx**/**sbin**/**nginx **-**s reload
2. #请先确保nginx是启动状态才可以执行命令成功，否则报错**,**报错信息如下：
3. #**[**error**]** open**()** "/usr/local/nginx/logs/nginx.pid" failed **(2:** No such file or directory**)**

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

1. **[**root@client **~]**# vim **/**etc**/**hosts
2. **192.168.4.5**    www**.**c**.**com www**.**a**.**com www**.**b**.**com

2）登录192.168.4.100客户端主机进行测试

1. **[**root@client **~]**# firefox https**:**//www.c.com            //信任证书后可以访问