

NSD ENGINEER DAY06

1. [案例1：独立Web站点的快速部署](#)
2. [案例2：虚拟Web主机的部署](#)
3. [案例3：配置网页内容访问](#)
4. [案例4：使用自定Web根目录](#)
5. [案例5：部署并测试WSGI站点](#)

1 案例1：独立Web站点的快速部署

1.1 问题

本例要求为 `http://server0.example.com` 配置Web站点，要求如下：

1. 从<http://classroom/pub/materials/station.html>下载一个主页文件，将其重命名为 `index.html`
2. 将此文件拷贝到站点的 `DocumentRoot` 目录下，不要对文件 `index.html` 的内容作任何修改
3. 使用 `elinks` 或 `firefox` 浏览上述Web站点

1.2 方案

Web网站服务端：软件包`httpd`、系统服务`httpd`

Web网站浏览器：软件包`elinks`或`firefox`

传输协议及端口：TCP 80

Web网站服务端配置文件：

- `/etc/httpd/conf/httpd.conf`
- `/etc/httpd/conf.d/*.conf`

默认首页文件：`index.html`

`httpd`网站文档的默认根目录：`/var/www/html`

URL (Uniform Resource Locator , 统一资源定位器) 网址的基本组成：

01 `http://服务器地址[:端口号]/目录/文件名`

对于需要验证的FTP资源，还需要指定用户名密码信息：

01 `ftp://用户名:密码@服务器地址[:端口号]/目录/文件名`

1.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：构建及部署网站服务器

[Top](#)

1) 安装软件包httpd

```
01. [root@server0 ~]# yum -y install httpd
02. ...
```

2) 部署网页

```
01. [root@server0 ~]# cd /var/www/html/ //进入网页目录
02. [root@server0 html]# wget http://classroom/pub/materials/station.html -O index.html
03. ...
04. 2016-11-26 19:33:49 (1.36 MB/s) - 'index.html' saved [14/14]
05.
06. [root@server0 html]# cat index.html //检查网页文件
07. Default Site.
```

3) 启动系统服务httpd，并设置开机自启

```
01. [root@server0 html]# systemctl restart httpd
02. [root@server0 html]# systemctl enable httpd
03. ln -s '/usr/lib/systemd/system/httpd.service' '/etc/systemd/system/multi-user.target.w'
```

步骤二：访问网站服务器

1) 使用elinks浏览器查看

Elinks浏览器可以在命令行模式显示出网页文本，经常用来测试网站的可用性。

```
01. [root@desktop0 ~]# yum -y install elinks //安装elinks
02. ...
03. [root@desktop0 ~]# elinks -dump http://server0.example.com/ //访问指定网址
04. Default Site.
```

2) 使用firefox浏览器查看

Firefox浏览器支持更多网页特性，是访问复杂网页、网址的优秀工具。

在桌面终端直接运行“`firefox http://server0.example.com/`”，或者通过菜单快捷方式打开Firefox浏览器再输入对应网址，都可以看到目标网页（如图-1所示）。 Top

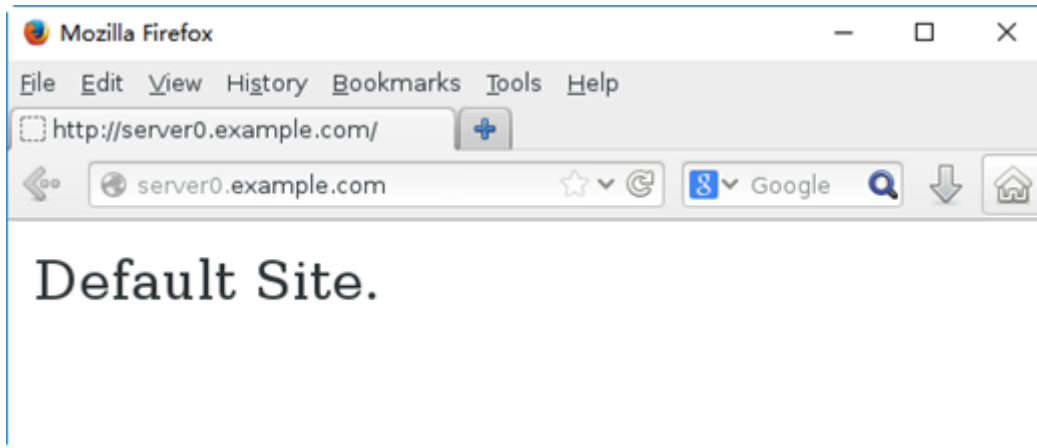


图-1

2 案例2：虚拟Web主机的部署

2.1 问题

本例要求为server0扩展Web站点，新建虚拟主机 `http://www0.example.com`，具体要求如下：

1. 设置 DocumentRoot 为 `/var/www/virtual`
2. 从 `http://classroom/pub/materials/www.html` 下载主页文件，并重命名为 `index.html`
3. 不要对文件 `index.html` 的内容作任何修改，将其放到此虚拟主机的 DocumentRoot 目录下
4. 确保 fleyd 用户能在 `/var/www/virtual` 目录建文件
5. 确保站点 `http://server0.example.com` 仍然可用

2.2 方案

单一网站平台（比如172.25.0.11）：

- 多个域名 ---> 相同的网页内容
- 配置文件：`/etc/httpd/conf/httpd.conf`
- 网页目录定义：DocumentRoot `/var/www/html`

虚拟主机平台（比如172.25.0.11）：

- 在同一套httpd平台上跑很多个网站
- 多个域名 ---> 不同的网页内容
- 网页目录由<VirtualHost ...>区段配置定义

多个虚拟主机站点的典型设置（`/etc/httpd/conf.d/*.conf`）：

01. `<VirtualHost *:80>`
02. ServerName 网站1的FQDN
03. DocumentRoot 网站1的网页根目录
04. `</VirtualHost>`
05. `<VirtualHost *:80>`
06. ServerName 网站2的FQDN
07. DocumentRoot 网站2的网页根目录

[Top](#)

- 08. </VirtualHost>
- 09. ...

2.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：部署网页文档

1) 建立网页目录

- 01. [root@server0 ~] # mkdir /var/www/virtual
- 02. [root@server0 ~] # useradd fleyd
- 03. [root@server0 ~] # setfacl -m u:fleyd:rwX /var/www/virtual/

2) 部署网页文件

- 01. [root@server0 ~] # cd /var/www/virtual/
- 02. [root@server0 virtual] # wget http://classroom/pub/materials/www.html -O index.html
- 03. ...
- 04. 100%[=====>] 14 --.- K/s in 0s
- 05.
- 06. 2016-11-26 20:01:14 (826 KB/s) - 'index.html' saved [14/14]
- 07. [root@server0 virtual] # cat index.html //检查网页文件
- 08. Virtual Site.

步骤二：配置虚拟主机 http://www0.example.com/

1) 为新站点创建独立的配置文件

- 01. [root@server0 virtual] # vim /etc/httpd/conf.d/01-~~www0~~.conf
- 02. <VirtualHost *:80>
- 03. ServerName www0.example.com
- 04. DocumentRoot /var/www/virtual
- 05. </VirtualHost>
- 06. [root@server0 virtual] # httpd -t //确保语法检查OK
- 07. Syntax OK

[Top](#)

2) 重启系统服务httpd

```
01. [root@server0 virtual] # systemctl restart httpd
```

步骤三：访问虚拟主机 <http://www0.example.com/>

访问此虚拟站点，可以看到预期的网页内容：

```
01. [root@desktop0 ~] # elinks - dump http://www0.example.com/
02.      Virtual Site.
```

步骤四：完善原始站点 <http://server0.example.com/>

需要注意的是，原始的独立站点可能出现异常，访问时并不是原始的网页：

```
01. [root@desktop0 ~] # elinks - dump http://server0.example.com/
02.      Virtual Site.
```

原因是一旦启用虚拟站点机制以后：

- 外部的 DocumentRoot、ServerName 会被忽略
- 第1个虚拟站点被视为默认站点，若客户机请求的URL不属于任何已知站点，则由第1个站点响应

若要解决此异常，需要将原始站点转换为第一个虚拟主机，启用顺序的设置可以通过文件名开头的数字来实现。

1) 为原始站点建立虚拟主机配置

```
01. [root@server0 ~] # vim /etc/httpd/conf.d/00-default.conf
02. <VirtualHost *:80>
03.     ServerName server0.example.com
04.     DocumentRoot /var/www/html
05. </VirtualHost>
```

2) 重启系统服务httpd

```
01. [root@server0 virtual] # systemctl restart httpd
```

[Top](#)

3) 访问两个虚拟站点，确保各自的网页内容正确

01. [root@desktop0 ~] # elinks - dump http://server0.example.com/
02. Default Site.
03. [root@desktop0 ~] # elinks - dump http://www0.example.com/
04. Virtual Site.

3 案例3：配置网页内容访问

3.1 问题

本例要求在 Web 网站 http://server0.example.com 的 DocumentRoot 目录下创建一个名为 private 的子目录，要求如下：

1. 从 http://classroom/pub/materials/private.html 下载一个文件副本到这个目录，重命名为 index.html
2. 不要对文件 index.html 的内容作任何修改
3. 从 server0 上，任何人都可以浏览 private 的内容，但是从其他系统不能访问这个目录的内容

3.2 方案

配置Web内容的访问控制需要添加Directory区段，主要形式可参考

01. <Directory "父目录路径">
02. Require all denied //上层目录拒绝任何访问
03. </Directory>
04. <Directory "子目录1路径">
05. Require all granted //子目录1允许任何访问
06. </Directory>
07. <Directory "子目录2路径">
08. Require ip IP或网段地址 ... //子目录2允许少数客户机
09. </Directory>

3.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：部署网页子目录及文档

1) 建立子目录

01. [root@server0 ~] # mkdir /var/www/html/private

[Top](#)

2) 部署网页

```

01. [root@server0 ~]# cd /var/www/html/private
02. [root@server0 private]# wget http://classroom/pub/materials/private.html -O index.html
03. ...
04. 2016-11-26 20:30:28 (1.90 MB/s) - 'index.html' saved [14/14]
05.
06. [root@server0 private]# cat index.html //检查网页文件
07. Private Site.

```

步骤二：为指定的网页子目录限制访问

在httpd服务的标准配置中，根目录 / 默认拒绝任何访问，但网页目录/var/www/默认允许任何访问。因此，只需要为个别子目录增加访问控制即可。

1) 调整虚拟站点server0.example.com的配置文件

```

01. [root@server0 ~]# vim /etc/httpd/conf.d/00-default.conf
02. ...
03. <Directory "/var/www/html/private">
04.     Require ip 127.0.0.1 ::1 172.25.0.11
05. </Directory>

```

2) 重启系统服务httpd

```

01. [root@server0 ~]# systemctl restart httpd

```

步骤三：测试目录访问限制

1) 从desktop0上访问http://server0.example.com/private/被拒绝

```

01. [root@desktop0 ~]# elinks -dump http://server0.example.com/private/
02. Forbidden
03.
04. You don't have permission to access /private/ on this server.

```

2) 从desktop0上访问http://server0.example.com/仍然是正常的

```

01. [root@desktop0 ~]# elinks -dump http://server0.example.com/
02. Default Site.

```

[Top](#)

3) 从server0本机上访问http://server0.example.com/private/也不受限制

```
01. [root@server0 ~]# elinks -dump http://server0.example.com/private/
02. Private Site.
```

4 案例4：使用自定Web根目录

4.1 问题

本例要求调整 Web 站点 http://server0.example.com 的网页目录，要求如下：

1. 新建目录 /webroot，作为此站点新的网页目录
2. 从 http://classroom/pub/materials/station.html 下载一个文件副本到这个目录，重命名为 index.html
3. 不要对文件 index.html 的内容作任何修改
4. 确保站点 http://server0.example.com 仍然可访问

4.2 方案

在SELinux强制启用模式下，增加新的合规网页目录的方法：

1) 参照标准目录，重设新目录的属性

```
01. chcon [-R] --reference=模板目录 新目录
```

或者

2) 将新目录增加到预设的标准Web目录范围

```
01. semanage fcontext -a -t httpd_sys_content_t '新目录(/.*)?'
```

4.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：部署网页目录及文档

1) 建立网页目录

```
01. [root@server0 ~]# mkdir /webroot
```

[Top](#)

2) 部署网页文件


```

01. [root@server0 ~] # cd /webroot/
02. [root@server0 webroot] # wget http://classroom/pub/materials/station.html -O index.html
03. ...
04. 2016-11-26 20:01:14 (826 KB/s) - 'index.html' saved [14/14]
05. [root@server0 webroot] # cat index.html //检查网页文件
06. Default Site.

```

步骤二：调整虚拟站点http://server0.example.com/的配置

1) 修改配置文件

```

01. [root@server0 ~] # vim /etc/httpd/conf.d/00-default.conf
02. <VirtualHost *:80>
03.     ServerName server0.example.com
04.     DocumentRoot /webroot
05. </VirtualHost>
06. ...

```

2) 重启系统服务httpd

```

01. [root@server0 ~] # systemctl restart httpd

```

步骤三：确保虚拟站点http://server0.example.com/仍然可以访问

1) 未调整网页目录SELinux上下文文件的情况

为虚拟站点http://server0.example.com/更换了新的网页目录以后，从浏览器访问将会失败，只能看到红帽测试页。

```

01. [root@desktop0 ~] # elinks -dump http://server0.example.com/
02. Red Hat Enterprise Linux Test Page
03.
04. This page is used to test the proper operation of the Apache HTTP server
05. after it has been installed. If you can read this page, it means that the
06. Apache HTTP server installed at this site is working properly.
07. ...

```

[Top](#)

针对此问题，可以参考目录/var/www的属性为网页目录/webroot设置SELinux安全上下文。

```

01. [ root@server0 ~] # chcon -R --reference=/var/www/webroot/
02. [ root@server0 ~] # ls -Z /webroot/index.html //确认结果
03. -rw-r--r--. root root system_u:object_r:httpd_sys_content_t:s0 /webroot/index.html

```

2) 未配置目录内容访问的情况

尽管已经调整过/webroot的SELinux安全上下文，但是从浏览器访问此虚拟站点时仍然会被拒绝，还是只能看到红帽测试页。

还需要修改对应的配置文件，添加内容访问控制：

```

01. [ root@server0 ~] # vim /etc/httpd/conf.d/00-default.conf
02. <VirtualHost *:80>
03.     ServerName server0.example.com
04.     DocumentRoot /webroot
05. </VirtualHost>
06. <Directory "/webroot">
07.     Require all granted
08. </Directory>
09. <Directory "/webroot/private">
10.     Require ip 127.0.0.1 ::1 172.25.0.11
11. </Directory>
12.
13. [ root@server0 ~] # systemctl restart httpd //重启httpd服务

```

若要保持原有private子目录，建议也拷贝过来：

```

01. [ root@server0 ~] # cp -rf /var/www/html/private/ /webroot/

```

3) 最终访问测试

从浏览器能成功访问调整后的虚拟站点http://server0.example.com/。

```

01. [ root@desktop0 ~] # elinks -dump http://server0.example.com/
02.      Default Site.

```

5 案例5：部署并测试WSGI站点

[Top](#)

5.1 问题

本例要求为站点 webapp0.example.com 配置提供动态Web内容，要求如下：

1. 此虚拟主机侦听在端口8909
2. 测试网页从以下地址下载，不要作任何更改http://classroom/pub/materials/webinfo.wsgi
3. 从浏览器访问 http://webapp0.example.com:8909 可接收到动态生成的 Web 页面
4. 此站点必须能被 example.com 域内的所有系统访问

5.2 方案

为httpd增加对Python网页程序的支持，可以安装mod_wsgi模块。关于此模块的配置说明，建议参考软件包提供的readme文档。

在SELinux处于Enforcing模式时，若要开放非80、81等常规Web端口，需要调整SELinux保护策略。

5.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：部署动态网页文档

1) 创建网页目录

```
01. [ root@server0 ~] # mkdir /var/www/webapp0
```

2) 部署webinfo.wsgi网页程序

```
01. [ root@server0 ~] # cd /var/www/webapp0
02. [ root@server0 webapp0] # wget http://classroom/pub/materials/webinfo.wsgi
03. ...
04. 2016-11-27 01:52:26 (16.0 MB/s) - 'webinfo.wsgi' saved [397/397]
05.
06. [ root@server0 webapp0] # cat webinfo.wsgi //检查下载文件
07. #!/usr/bin/env python
08. import time
09. ...
```

步骤二：配置新的虚拟主机http://webapp0.example.com : 8909/

1) 安装mod_wsgi模块软件包

```
01. [ root@server0 ~] # yum -y install mod_wsgi
02. ...
```

[Top](#)

2) 为新虚拟主机建立配置

```

01. [root@server0 ~]# vim /etc/httpd/conf.d/02-webapp0.conf
02. Listen 8909
03. <VirtualHost *:8909>
04.     DocumentRoot /var/www/webapp0
05.     ServerName webapp0.example.com
06.     WSGIScriptAlias / /var/www/webapp0/webinfo.wsgi
07. </VirtualHost>

```

3) 调整SELinux策略，允许Web服务使用8909端口

列出当前许可的Web端口：

```

01. [root@server0 ~]# semanage port -l | grep ^http_port
02. http_port_t      tcp      80, 81, 443, 488, 8008, 8009, 8443, 9000

```

添加新的Web端口：

```

01. [root@server0 ~]# semanage port -a -t http_port_t -p tcp 8909
02. [root@server0 ~]#

```

确认配置结果：

```

01. [root@server0 ~]# semanage port -l | grep ^http_port
02. http_port_t      tcp      8909, 80, 81, 443, 488, 8008, 8009, 8443, 9000

```

4) 重启系统服务httpd

```

01. [root@server0 ~]# systemctl restart httpd
02. [root@server0 ~]# netstat -antpu | grep httpd //确认已监听8909端口
03. tcp6      0      0 :::443          :::*             LISTEN          2477/httpd
04. tcp6      0      0 :::8909         :::*             LISTEN          2477/httpd
05. tcp6      0      0 :::80           :::*             LISTEN          2477/httpd

```

步骤三：测试动态网页效果

[Top](#)

使用elinks或firefox访问此动态站点http://webapp0.example.com:8909/。

多刷新访问几次，每次看到的是动态网页内容，内容并不固定。

01. [root@desktop0 ~] # elinks - dump http://webapp0.example.com: 8909/
02. UNIX EPOCH time is now: 1480184916.52 //第1次访问
03. [root@desktop0 ~] # elinks - dump http://webapp0.example.com: 8909/
04. UNIX EPOCH time is now: 1480184919.21 //第2次访问
05. [root@desktop0 ~] # elinks - dump http://webapp0.example.com: 8909/
06. UNIX EPOCH time is now: 1480184951.99 //第3次访问

[Top](#)