

# NSD OPERATION DAY07

1. [案例1：Subversion基本操作](#)
2. [案例2：使用Subversion协同工作](#)
3. [案例3：制作nginx的RPM包](#)

## 1 案例1：Subversion基本操作

### 1.1 问题

本案例要求先快速搭建好一台Subversion服务器，并测试该版本控制软件：

- 创建版本库
- 导入初始化数据
- 检出数据至用户本地副本
- 对本地副本进行增删改查等操作

### 1.2 方案

使用YUM安装subversion软件，使用svn客户端工具连接svnserver服务器并测试版本控制软件。

使用svn命令测试svnserver服务时可以使用的命令列表如表-1所示。

表 - 1 svn命令列表

命令	作用
add	添加文件
commit	提交更新
checkout	检出代码
cat	查看代码文件内容
del	删除文件
diff	文件对比
import	导入代码
info	查看版本信息
list	查看文件列表
log	查看版本历史
update	更新
mkdir	创建目录

### 1.3 步骤

实现此案例需要按照如下步骤进行。

#### 步骤一：安装Subversion服务器

1 ) YUM安装subversion软件

[Top](#)

01 [root@svr5 ~]# yum -y install subversion

```
02. [root@svr5 ~] # rpm -q subversion
```

## 2)创建版本库

```
01. [root@svr5 ~] # mkdir /var/svn/  
02. [root@svr5 ~] # svnadmin create /var/svn/project  
03. [root@svr5 ~] # ls /var/svn/project/  
04. conf/ db/ format hooks/ locks/ README.txt
```

## 3 ) 修改配置文件 , 创建账户与密码

```
01. [root@svr5 ~] # vim /var/svn/project/conf/svnserve.conf  
02. [general]  
03. ### These options control access to the repository for unauthenticated  
04. ### and authenticated users. Valid values are "write", "read",  
05. ### and "none". The sample settings below are the defaults.  
06. anon- access = none      //匿名无任何权限  
07. auth- access = write     //有效账户可写  
08. ### The password- db option controls the location of the password  
09. ### database file. Unless you specify a path starting with a /,  
10. ### the file's location is relative to the directory containing  
11. ### this configuration file.  
12. ### If SASL is enabled ( see below ) , this file will NOT be used.  
13. ### Uncomment the line below to use the default password file.  
14. password- db = passwd    //密码文件  
15. ### The authz- db option controls the location of the authorization  
16. ### rules for path- based access control. Unless you specify a path  
17. ### starting with a / , the file's location is relative to the the  
18. ### directory containing this file. If you don't specify an  
19. ### authz- db , no path- based access control is done.  
20. ### Uncomment the line below to use the default authorization file.  
21. authz- db = authz        //ACL访问控制列表文件  
22. ### This option specifies the authentication realm of the repository .  
23. ### If two repositories have the same authentication realm , they should  
24. ### have the same password database , and vice versa. The default realm  
25. ### is repository 's uuid.  
26. # realm = My First Repository  
27.  
28. [root@svr5 ~] # vim /var/svn/project/conf/passwd
```

[Top](#)

```

29.  ... ..
30.  [ users]
31.  harry = pass          //用户名和密码
32.  tom= pass            //用户名和密码
33.
34.  [ root@svr5 ~] # cat /var/svn/project/conf/authz
35.  [ aliases]
36.  # joe = /C=XZ/ST=Dessert/L=Snake City/O=Snake Oil, Ltd./OU=Research Institute/CN=Joe /
37.
38.  [ groups]
39.  harry_and_tom = harry,tom      //定义组账户，组成员为harry和tom
40.
41.  [/]                          //定义ACL访问控制
42.  harry = rw                  //用户对项目根路径可读可写
43.  tom = rw
44.  * = r                      //其他人只读

```

#### 4) 启动服务

```

01.  [ root@svr5 ~] # systemctl start svnserve
02.  [ root@svr5 ~] # netstat -n | grep svnserve
03.  tcp      0      0 0.0.0.0:3690 0.0.0.0:* LISTEN    4043/svnserve

```

### 步骤二：客户端测试

#### 1) 本地导入初始化数据

```

01.  [ root@svr5 ~] # cd /usr/lib/systemd/system/
02.  [ root@svr5 ~] # svn import . file:///var/svn/project/ -m "Init Data"
03.  [ root@svr5 ~] # cd /var/tmp
04.  [ root@svr5 ~] # svn --username harry --password pass \
05.  co svn://127.0.0.1/var/svn/project harry      //harry账户检查数据，建立本地副本
06.
07.  -----
08.  ATTENTION! Your password for authentication realm:
09.
10.  <svn://127.0.0.1:3690> b72f45f0-bbe5-4a0c-ad4a-37f52704f0b1
11.
12.  can only be stored to disk unencrypted! You are advised to configure

```

[Top](#)

```

13.   your system so that Subversion can store passwords encrypted, if
14.   possible. See the documentation for details.
15.
16.   You can avoid future appearances of this warning by setting the value
17.   of the 'store-plaintext-passwords' option to either 'yes' or 'no' in
18.   '/root/.subversion/servers'.
19.   -----
20.   Store password unencrypted (yes/no) ? yes          //提示是否保存密码
21.
22.   [ root@srv5 ~ ] # cd /var/tmp/harry
23.   [ root@srv5 harry ] # ls
24.   [ root@srv5 harry ] # vim test.sh                //新建脚本文件
25.   #!/bin/bash
26.   case $1 in
27.   start)
28.       echo start;;
29.   stop)
30.       echo stop;;
31.   *)
32.       echo Error
33.   esac
34.   [ root@srv5 harry ] # chmod +x test.sh
35.   [ root@srv5 harry ] # svn add test.sh             //将文件或目录加入版本控制
36.   [ root@srv5 harry ] # svn mkdir subdir           //创建子目录
37.   [ root@srv5 harry ] # svn status                 //检查状态，结果为两个添加append
38.   A    test.sh
39.   A    subdir
40.   [ root@srv5 harry ] # svn del cups               //删除版本库中的文件
41.   [ root@srv5 harry ] # svn move test.sh test      //脚本重命名
42.   A    test
43.   D    test.sh
44.   [ root@srv5 harry ] # svn commit -m "add a file and subdir,remove cups file"
45.                                           //将本地副本的修改提交版本库
46.   Deleting    cups
47.   Adding      subdir
48.   Adding      test
49.   Transmitting file data .
50.   Committed revision 2.
51.   [ root@srv5 harry ] # sed -i '1a###test###' sshd.service //修改本地副本中的代码文件
52.   [ root@srv5 harry ] # sed -i '2a###test###' sshd.service
53.   [ root@srv5 harry ] # svn diff sshd.service        //查看单个文件本地副本与版本库的差异

```

[Top](#)

```
54. [root@srv5 harry] # svn diff //查看所有本地副本与版本库的差异
55. [root@srv5 harry] # svn log svn: //127.0.0.1/var/svn/project //查看修改历史
56. [root@srv5 harry] # svn update //更新本地副本文件，从版本库下载更新数据
57. [root@srv5 harry] # cd /var/tmp
58. [root@srv5 tmp] # svn - r2 co svn: //127.0.0.1/var/svn/project code2 //下载历史版本
59. [root@srv5 tmp] # cd harry
60. [root@srv5 harry] # svn merge - r10:5 sshd.service
```

## 2 案例2：使用Subversion协同工作

### 2.1 问题

沿用练习一，通过svn工具，对subversion版本库进行多人协同工作测试，要求如下：

- 使用subversion管理公司的shell脚本
- 脚本包括/etc/rc.d/init.d/目录下的启动脚本
- 以及任何用户自己编写的脚本
- 创建脚本版本库
- 该版本库支持多个账户同时协作编辑脚本
- 测试演示多人协作编辑的具体操作
- 手动解决版本冲突问题
- 备份版本库数据

### 2.2 方案

使用svn客户端工具连接subversion服务器并测试多人协同工作以及如何手动解决冲突问题，账户名称分别为harry和tom，最后使用svnadmin dump指令对版本库进行备份工作。

### 2.3 步骤

实现此案例需要按照如下步骤进行。

#### 步骤一：多人协同工作

1) 远程连接两个终端，每个人下载代码本地副本，注意user1和user2代表了两个不同的主机，看清楚操作是在哪一台计算机上执行！

```
01. [user1@srv5 ~] # svn --username harry --password pass co svn: //127.0.0.1/project harry
02. [user2@srv5 ~] # svn --username tom --password pass co svn: //127.0.0.1/project tom
03. [user1@srv5 ~] # ls harry
04. [user2@srv5 ~] # ls tom
```

2) harry和tom修改不同的文件

[Top](#)

```
01. [user1@srv5 ~] # cd harry
```

```

02. [ user1@srv5 harry ] # sed -i "3a####harry \s modify #####" vmttoolsd.service
03. [ user1@srv5 harry ] # svn commit -m "vmttoolsd has modified"
04. [ user2@srv5 ~ ] # cd tom
05. [ user2@srv5 tom ] # sed -i "3a####tom \s modify #####" sshd.servie
06. [ user2@srv5 tom ] # svn commit -m "sshd has modified"
07. [ user1@srv5 harry ] # svn update
08. [ user2@srv5 tom ] # svn update

```

### 3 ) harry和tom修改相同文件的不同行

```

01. [ user1@srv5 ~ ] # cd harry
02. [ user1@srv5 harry ] # sed -i "3a####harry \s modify #####" user.slice
03. [ user1@srv5 harry ] # svn commit -m "user.slice has modified"
04. [ user2@srv5 ~ ] # cd tom
05. [ user2@srv5 tom ] # sed -i "6a####tom \s modify #####" user.slice
06. [ user2@srv5 tom ] # svn commit -m "user.slice has modified"
07. Sending      svnservice
08. Transmitting file data .svn: Commit failed (details follow):
09. svn: File '/user.slice' is out of date
10. [ user2@srv5 tom ] # svn update           //提示失败后，先更新再提交即可
11. [ user2@srv5 tom ] # svn commit -m "user.slice has modified"
12. Sending      user.slice
13. Transmitting file data .

```

### 4) harry和tom修改相同文件的相同行

```

01. [ user1@srv5 ~ ] # cd harry
02. [ user1@srv5 harry ] # sed -i "8c####harry \s modify #####" zram.service
03. [ user1@srv5 harry ] # svn commit -m "zram.service has modified"
04. [ user2@srv5 ~ ] # cd tom
05. [ user2@srv5 tom ] # sed -i "8c####tom \s modify #####" zram.service
06. [ user2@srv5 tom ] # svn commit -m "zram.service has modified"
07. svn commit -m "zram.service has modified"
08. Sending      zram.service
09. Transmitting file data .svn: Commit failed (details follow):
10. svn: File '/zram.service' is out of date
11. [ user2@srv5 tom ] # svn update           //出现冲突，需要解决
12. Conflict discovered in 'zram.service'.
13. Select: (p) postpone, (df) diff-full, (e) edit,

```

[Top](#)

14. (mc) mine conflict, (tc) theirs conflict,
15. (s) show all options: p //选择先标记，随后解决
16. [ user2@srv5 tom] # ls
17. zram.service zram.service.mine zram.service.r10 zram.service.r9
18. [ user2@srv5 tom] # vm zram.service.mine zram.service
19. [ user2@srv5 tom] # rm zram.service.mine zram.service.r10 zram.service.r9
20. [ user2@srv5 tom] # svn commit -m "zram.service has modified" //解决冲突

## 步骤二：使用dump指令备份版本库数据

01. [ root@srv5 ~] # svnadmin dump /var/svn/project > project.bak
02. \* Dumped revision 0.
03. \* Dumped revision 1.
04. \* Dumped revision 2.
05. \* Dumped revision 3.
06. \* Dumped revision 4.
07. \* Dumped revision 5.
08. \* Dumped revision 6.
09. \* Dumped revision 7.
10. \* Dumped revision 8.
11. \* Dumped revision 9.
12. \* Dumped revision 10.
13. \* Dumped revision 11.

## 3 案例3：制作nginx的RPM包

### 3.1 问题

本案例使用nginx-1.8.0版本的源码软件，生产对应的RPM包软件，具体要求如下：

- 软件名称为nginx
- 软件版本为1.8.0
- RPM软件包可以查询描述信息
- RPM软件包可以安装及卸载

### 3.2 方案

安装rpm-build软件包，编写SPEC配置文件，创建新的RPM软件包。

配置文件中的描述信息如表-2：

表 - 2 SPEC描述信息

[Top](#)

选项	值
Name	Nginx
Version	1.8.0
Release	1
Summary	Nginx is a web server software.
License	GPL
URL	www.nginx.org
Source0	nginx-1.8.0.tar.gz
BuildRequires	gcc pcre-devel zlib-devel
%description	nginx [engine x] is an HTTP and reverse proxy server... ..
mkdir	创建目录

### 3.3 步骤

实现此案例需要按照如下步骤进行。

#### 步骤一：安装rpm-build软件

##### 1 ) 安装rpm-build软件包

```
01. [ root@svr5 ~] # yum -y install rpm-build
```

##### 2 ) 生成rpmbuild目录结构

```
01. [ root@svr5 ~] # rpmbuild -ba nginx.spec
02. [ root@svr5 ~] # ls /root/rpmbuild
03. BUILD BUILDROOT RPMS SOURCES SPECS SRPMS
```

##### 3 ) 准备工作，将源码软件复制到SOURCES目录

```
01. [ root@svr5 ~] # cp nginx-1.8.0.tar.gz /root/rpmbuild/SOURCES/
```

##### 4 ) 创建并修改SPEC配置文件

```
01. [ root@svr5 ~] # vim SPECS/nginx.spec
02. Name: nginx
03. Version: 1.8.0
04. Release: 1%{ ?dist}
05. Summary: test
06.
```

[Top](#)



```

07.   License: GPL
08.   URL:    www.test.com
09.   Source0: nginx- 1.8.0.tar.gz
10.
11.   #BuildRequires:
12.   #Requires:
13.
14.   %description
15.   test too
16.
17.   %prep
18.   %setup - q
19.
20.
21.   %build
22.   ./configure
23.   make %{ ?_smp_mflags}
24.
25.
26.   %install
27.   make install DESTDIR=%{ buildroot}
28.   mkdir -p %{ buildroot} /etc/init.d/
29.   install /root/rpmbuild/SPECS/nginx.sh %{ buildroot} /etc/init.d/
30.   ##注意，这里是将一个脚本拷贝到安装目录，如果没有则不需要该指令
31.
32.   %files
33.   %doc
34.   /etc/init.d/nginx.sh      #将前面拷贝的脚本，打包到RPM中
35.   /usr/local/nginx/*
36.
37.   %changelog

```

## 步骤二：使用配置文件创建RPM包

### 1 ) 安装依赖软件包

```
01. [ root@svr5 ~] # yum -y install gcc pcre-devel zlib-devel
```

### 2 ) rpmbuild创建RPM软件包

[Top](#)

```

01. [ root@svr5 ~] # rpmbuild - ba SPECS/nginx.spec
02. [ root@svr5 ~] # ls RPMS/x86_64/nginx- 1.8.0- 1.x86_64.rpm
03. [ root@svr5 ~] # rpm - qpi RPMS/x86_64/nginx- 1.8.0- 1.x86_64.rpm
04. Name      : nginx      Relocations: ( not relocatable)
05. Version   : 1.8.0      Vendor: ( none)
06. Release   : 1          Build Date: Mon 02 May 2016 02: 30: 53 AMPDT
07. Install Date: ( not installed)      Build Host: localhost
08. Group      : Applications/Internet    Source RPM: nginx- 1.8.0- 1.src.rpm
09. Size       : 721243      License: GPL
10. Signature  : ( none)
11. URL        : www.nginx.org
12. Summary    : Nginx is a web server software.
13. Description :
14. nginx [ engine x] is an HTTP and reverse proxy server , a mail proxy server , and a generic T
15. [ root@svr5 ~] # rpm - qpl RPMS/x86_64/nginx- 1.8.0- 1.x86_64.rpm
16. /usr
17. /usr/local
18. /usr/local/nginx
19. /usr/local/nginx/conf
20. /usr/local/nginx/conf /fastcgi.conf
21. /usr/local/nginx/conf /fastcgi.conf.default
22. /usr/local/nginx/conf /fastcgi_params
23. /usr/local/nginx/conf /fastcgi_params.default
24. /usr/local/nginx/conf /koi-utf
25. /usr/local/nginx/conf /koi-win
26. /usr/local/nginx/conf /mime.types
27. /usr/local/nginx/conf /mime.types.default
28. /usr/local/nginx/conf /nginx.conf
29. /usr/local/nginx/conf /nginx.conf.default
30. /usr/local/nginx/conf /scgi_params
31. /usr/local/nginx/conf /scgi_params.default
32. /usr/local/nginx/conf /uwsgi_params
33. /usr/local/nginx/conf /uwsgi_params.default
34. /usr/local/nginx/conf /win-utf
35. /usr/local/nginx/html
36. /usr/local/nginx/html/50x.html
37. /usr/local/nginx/html/index.html
38. /usr/local/nginx/logs
39. /usr/local/nginx/sbin
40. /usr/local/nginx/sbin/nginx

```

[Top](#)

### 步骤三：安装、卸载软件

01. [ root@svr5 ~] # rpm - ivh RPMS/x86\_64/nginx-1.8.0-1.x86\_64.rpm
02. [ root@svr5 ~] # rpm - qa | grep nginx
03. [ root@svr5 ~] # /usr/local/nginx/sbin/nginx
04. [ root@svr5 ~] # curl http://127.0.0.1/
05. [ root@svr5 ~] # /usr/local/nginx/sbin/nginx - s stop
06. [ root@svr5 ~] # rpm - e nginx