[subversion (svn) 1](#_Toc1889661332)

[搭建svn与基本操作 1](#_Toc1791624119)

[Subversion服务器: 2](#_Toc1579194324)

[客户端: 3](#_Toc800336478)

[使用Subversion协同工作 4](#_Toc1624284197)

[使用dump指令备份版本库数据 6](#_Toc1687153994)

# subversion (svn)

## 搭建svn与基本操作

Svn 版本控制系统，允许你数据恢复到早期的版本名检查数据修改的历史；允许你和别人协作文档并跟踪所作的修改。

svn命令:

add 添加文件

Commit（ci） 提交更新

Checkout（co）下载代码

Cat 检查代码内容add 添加文件

Commit（ci） 提交更新

Checkout（co） 下载代码

Cat 检查代码内容

Del 删除文件

Diff 文件对比

Import 导入文件

Info 查看版本信息

List 查看文件列表

Log 查看版本历史

Update 客户端更新

Mkdir 创建目录

Del 删除文件

Diff 文件对比

Import 导入文件

Info 查看版本信息

List 查看文件列表

Log 查看版本历史

Update 客户端更新

Mkdir 创建目录

### Subversion服务器:

**安装包:** subversion

**服务名:** svnserve

**监听端口:** 3690

**配置文件:** /var/svn/project/conf/svnserve.conf (/var/svn/project 是svn仓库的地址.也就是说不同的仓库配置文件地址也不一样)

**用户和密码:** /var/svn/project/conf/passwd

1. 安装服务

# yum -y install subversion

2. 创建版本库

# mkdir /var/svn/

# svnadmin create /var/svn/project //创建仓库

# ls /var/svn/project/

conf/ db/ format hooks/ locks/ README.txt //以上是自动生成的文件

3. 本地导入初始化数据

# cd /usr/lib/systemd/system/

# svn import . file:///var/svn/project/ -m "Init Data" // 导入当前目录到svn仓库

4. 修改配置文件，创建账户与密码

# vim /var/svn/project/conf/svnserve.conf

[general]

### These options control access to the repository for unauthenticated

### and authenticated users. Valid values are "write", "read",

### and "none". The sample settings below are the defaults.

anon-access = none

//匿名无任何权限，此处可以设置为none read write

auth-access = write

//有效账户可写

### The password-db option controls the location of the password

### database file. Unless you specify a path starting with a /,

### the file's location is relative to the directory containing

### this configuration file.

### If SASL is enabled (see below), this file will NOT be used.

### Uncomment the line below to use the default password file.

password-db = passwd

//密码文件账户名称与密码的存放文件，该文件存放在conf目录下

### The authz-db option controls the location of the authorization

### rules for path-based access control. Unless you specify a path

### starting with a /, the file's location is relative to the the

### directory containing this file. If you don't specify an

### authz-db, no path-based access control is done.

### Uncomment the line below to use the default authorization file.

authz-db = authz

//ACL访问控制列表文件。基于路径的访问控制文件名。可以对文件或麋鹿设置全线

### This option specifies the authentication realm of the repository.

### If two repositories have the same authentication realm, they should

### have the same password database, and vice versa. The default realm

### is repository's uuid.

# realm = My First Repository

# vim /var/svn/project/conf/passwd

… …

[users]

harry = pass

//用户名和密码

tom =pass

//用户名和密码

# cat /var/svn/project/conf/authz

[/]  //定义ACL访问控制。子目录没有特殊定义，就继承根目录权限

harry = rw                        //用户对项目根路径可读可写

tom = rw

\* =                            //其他人只读，此处可写 ”rw空“

5. 启动服务

svnserve -d 共享所有目录

# svnserve -d -r /var/svn/project 只共享一个目录

# netstat -nutlp |grep svnserve

tcp 0 0 0.0.0.0:3690    0.0.0.0:\*    LISTEN 4043/svnserve

### 客户端:

安装包: subversion

1）将服务器上的代码下载到本地

# cd /tmp

# svn --username harry --password pass co svn://192.168.2.100/ code

//建立本地副本,从服务器192.168.2.100上co下载代码到本地code目录

//用户名harry,密码pass

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

ATTENTION! Your password for authentication realm:

<svn://127.0.0.1:3690> b72f45f0-bbe5-4a0c-ad4a-37f52704f0b1

can only be stored to disk unencrypted! You are advised to configure

your system so that Subversion can store passwords encrypted, if

possible. See the documentation for details.

You can avoid future appearances of this warning by setting the value

of the 'store-plaintext-passwords' option to either 'yes' or 'no' in

'/root/.subversion/servers'.

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

Store password unencrypted (yes/no)? yes            //提示是否保存密码

svn: E200002: /var/svn/project/conf/svnserve.conf:19: Option expected

客户端出现这个错误是因为服务端的配置文件没有顶格

**命令详解**

# cd /tmp/code

# ls

# vim user.slice                 //挑选任意文件修改其内容

# svn ci -m "modify user"        //将本地修改的数据同步到服务器

# svn update                    //将服务器上新的数据同步到本地svn

# svn info     svn://192.168.2.100    //查看版本仓库基本信息

# svn log     svn://192.168.2.100    //查看版本仓库的日志

# echo "test" > test.sh        //本地新建一个文件

# svn ci -m "new file"            //提交失败，该文件不被svn管理

# svn add test.sh                //将文件或目录加入版本控制

# svn ci -m "new file"            //再次提交，成功

# svn mkdir subdir                //创建子目录

# svn rm timers.target            //使用svn删除文件

# svn ci -m "xxx"                //提交一次代码

# vim umount.target            //任意修改本地的一个文件

# svn diff                     //查看所有文件的差异

# svn diff umount.target        //仅查看某一个文件的差异

# svn cat svn://192.168.2.100/reboot.target    //查看服务器文件的内容

# sed -i 'd' tmp.mount     //删除文件所有内容，但未提交

# svn revert tmp.mount   //还原tmp.mount文件

# rm -rf \*.target            //任意删除若干文件

# svn update     //还原

# sed -i '1a #test###' tuned.service //修改本地副本中的代码文件

# svn ci -m "xxx" //提交代码

# svn merge -r7:2    tuned.service    //将文件从版本7还原到版本2

## 使用Subversion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web1(192.168.2.100)和web2（192.168.2.200）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！

# cd /tmp

# svn --username harry --password pass \

> co svn://192.168.2.100/project mycode

# cd /tmp

# svn --username tom --password pass \

> co svn://192.168.2.100/project mycode

# cd mycode

# cd mycode

2) harry和tom修改不同的文件,允许提交

[root@web1 mycode]# sed -i "3a ###harry modify#####" tmp.mount

[root@web1 mycode]# svn ci -m "has modified"

[root@web2 mycode]# sed -i "3a ###tom modify#####" umount.target

[root@web2 mycode]# svn ci -m "has modified"

[root@web2 mycode]# svn update

[root@web1 mycode]# svn update

3）harry和tom修改相同文件的不同行,需要更新后才能提交

[root@srv5 ~]# cd harry

[root@web1 mycode]# sed -i "3a ###harry modify#####" user.slice

[root@web1 mycode]# svn ci -m "modified"

[root@web2 mycode]# sed -i "6a ###tom modify#####" user.slice

[root@web2 mycode]# svn ci -m "modified"        //提交失败

Sending svnserve

Transmitting file data .svn: Commit failed (details follow):

svn: File '/user.slice' is out of date（过期）

[root@web2 mycode]# svn update                  //提示失败后，先更新再提交即可

[root@web2 mycode]# svn ci -m "modified"        //提交成功

Sending user.slice

Transmitting file data .

4) harry和tom修改相同文件的相同行

[root@web1 mycode]# sed -i '1c [UNIT]' tuned.service

[root@web1 mycode]# svn ci -m "modified"

[root@web2 mycode]# sed -i '1c [unit]' tuned.service

[root@web2 mycode]# svn ci -m "modified"

Sending tuned.service

Transmitting file data .svn: Commit failed (details follow):

svn: File '/tuned.service' is out of date(过期)

[root@web2 mycode]# svn update                    //出现冲突，需要解决

Conflict discovered in 'tuned.service'.

Select: (p) postpone, (df) diff-full, (e) edit,

(mc) mine-conflict, (tc) theirs-conflict,

(s) show all options:p                    //选择先标记p，随后解决

[root@web2 mycode]# ls

tuned.service tuned.service.mine        tuned.service.r10    tuned.service.r9

[root@web2 mycode]# mv tuned.service.mine tuned.service

[root@web2 mycode]# rm -rf tuned.service.r10 tuned.service.r9

[root@web2 mycode]# svn ci -m "modified"    //解决冲突

## 使用dump指令备份版本库数据

# svnadmin dump /var/svn/project > project.bak //备份

\* Dumped revision 0.

\* Dumped revision 1.

\* Dumped revision 2.

\* Dumped revision 3.

\* Dumped revision 4.

\* Dumped revision 5.

\* Dumped revision 6.

\* Dumped revision 7.

\* Dumped revision 8.

\* Dumped revision 9.

\* Dumped revision 10.

\* Dumped revision 11.

# svnadmin load /var/svn/project2 < project.bak     //还原