http协议的svn服务器搭建(apache+subversion)

# 安装apache subversion

yum install httpd   
yum install  httpd-devel       //用于开发Apache模块   
yum install  subversion   
yum install  mod\_dav\_svn   //Subversion与Apache之间的接口（通过它，Apache就可以访问版本库，可以让客户端也使用HTTP的扩展协议 WebDAV/DeltaV进行访问）

yum install  mod\_auth\_mysql   //这一项用于用户身份验证，在网上有资料说貌似这个mod\_auth\_mysql没得配置访问权限的只有用户设置，建议不装，用htpasswd就可以，在这里我没有装，我还是打算用htpasswd，可能下回会试试。 （我是没装成功）  
验证 svn模块（mod\_dav\_svn）是否正确安装   
ls /etc/httpd/modules | grep svn

系统提示： mod\_authz\_svn.so   
mod\_dav\_svn.so   
验证安装：   
svn --version

＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝

# 创建SVN库和项目

mkdir /home/svn          //创建SVN库   
svnadmin create /home/svn/projcet         //创建项目

# 添加组及组员权限

groupadd subversion     //创建一个叫subversion的组为拥有仓库所在的目录   
usermod -G subversion -a apache   //将自己和Apache 用户加入组成员中   
查看：more /etc/group | grep subversion   
系统提示：subversion:x:500:apache

# 修改项目权限

chgrp -R subversion /home/svn/project   
chmod －R 776 /home/svn/project

这里最好根据自己的安全需求配置相 应的权限，我这是试验环境所以直接给了777

# ****配置httpd****

编辑 httpd.conf   
vim /etc/httpd/conf/httpd.conf   
查看有没有LoadModule dav\_svn\_module modules/mod\_dav\_svn.so这一行，如果没有就添加上为了使客户端能够通过Apache来访问SVN档案库，Apache需要加载 mod\_dav\_svn模块。如果Apache是按照与预设目 录安装的，mod\_dav\_svn 模块应该会安装在 Apache 安装位置 (默认路径是 /etc/httpd/) 的 modules 子目录内。同时在Apache的配置文件httpd.conf（默认路径为etc/httpd/conf/）中已经使用 LoadModule指令加载了该模块（如果没有，手动添加）注意这个指令必须出现在其它的Subversion 相关指令之前。还要加载mod\_authz\_svn.so模块   
例如：  
LoadModule dav\_svn\_modulemodules/mod\_dav\_svn.so   
LoadModule authz\_svn\_module modules/mod\_authz\_svn.so   
一般来说apache2都己正常加载了这两个模块   
启动httpd   
/etc/init.d/httpd start  或者 service httpd start   
打开浏览器查看是否能看到apache的页 面（注意如果开了防火墙记得开放80端口）

# 配置SVN

编辑 subversion.conf

vim /etc/httpd/conf.d/subversion.conf

按以下代码更改（栗红为修改的代码）：

LoadModule dav\_svn\_module modules/mod\_dav\_svn.so   
LoadModule authz\_svn\_module modules/mod\_authz\_svn.so   
    
<Location /svn>   
DAV svn   
SVNParentPath /home/svn   
#   
#   # Limit write permission to list of valid users.   
#   <LimitExcept GET PROPFIND OPTIONS REPORT>   
#      # Require SSL connection for passwordprotection.   
#      # SSLRequireSSL   
#   
AuthType Basic   
AuthName "Authorization Realm"   
AuthUserFile /home/svn/conf/passwd

AuthzSVNAccessFile /home/svn/conf/authz  
Require valid-user   
#   </LimitExcept>   
</Location>

# 建立用户

# htpasswd -c /home/svn/conf/passwd test         //第一次添加用户使用参数“-c”以后再添加就不用了   
**系统提示：**Newpassword:                       //输入新密码   
Re-type newpassword:         //确认密码    
Adding password for user test   
重启apache                                      
通过web页面访问，弹出用户登录验证，访问成功 （注意开放防火墙的3690端口）  
2011/2/15 15:22

重启服务器

service httpd retart

启动服务器  
# svnserve -d -r /home/svn

一般无法访问是库权限设置问题 chmod -R 776 /home/svn/project

# 备份与还原

svn备份一般采用三种方式：  
1)svnadmin dump   
2)svnadmin hotcopy   
3)svnsync.

注意，svn备份不宜采用普通的文件拷贝方式（除非你备份的时候将库暂停），如copy命令、rsync命令。   
笔者曾经用 rsync命令来做增量和全量备份，在季度备份检查审计中，发现备份出来的库大部分都不可用，因此最好是用svn本身提供的功能来进行备份。   
  
优缺点分析  
==============   
第一种svnadmin dump是官方推荐的备份方式，优点是比较灵活，可以全量备份也可以增量备份，并提供了版本恢复机制。   
缺点是：如果版本比较大，如版本数增长到数万、数十万，那么dump的过程将非常慢；备份耗时，恢复更耗时；不利于快速进行灾难恢复。   
个人建议在版本数比较小的情况下使用这种备份方式。   
第二种svnadmin hotcopy原设计目的估计不是用来备份的，只能进行全量拷贝，不能进行增量备份；   
优点是：备份过程较快，灾难恢复也很快；如果备份机上已经搭建了svn服务，甚至不需要恢复，只需要进行简单配置即可切换到备份库上工作。   
缺点是：比较耗费硬盘，需要有较大的硬盘支持（俺的备份机有1TB空间，呵呵）。   
第三种svnsync实际上是制作2个镜像库，当一个坏了的时候，可以迅速切换到另一个。不过，必须svn1.4版本以上才支持这个功能。   
优点是：当制作成2个镜像库的时候起到双机实时备份的作用；   
缺点是：当作为2个镜像库使用时，没办法做到“想完全抛弃今天的修改恢复到昨晚的样子”；而当作为普通备份机制每日备份时，操作又较前2种方法麻烦。

**增量备份**

svnadmin dump /home/svn/python > /tmp/svn\_bakup

增量备份：使用svnadmin dump的–incremental选项来实现  
svnadmin dump 版本库路径及名称 –revision 上次导出的版本号:到本次要导出到的版本号 –incremental > 导出的命名

Eg:

**svnadmin dump** /home/svn/python -r 7:8 --incremental > /tmp/svn\_backup

恢复：

mkdir new repos

svnadmin create newrepos

svnadmin load newrepos < svn\_backup

**热备份（全量）**

svnadmin hotcopy path/to/repository path/to/backup –clean-logs （后面的参数不知道怎么用）

还原

svnadmin hotcopy /home/backup.bak  /home/backup.bak

**svnsync实时备份**

创建备份空版本库

$ svnadmin create svn-mirror

配置来源、备份SVN，创建用户：syncuser

请确认syncuser用户对备份SVN有写入权限。对来源SVN有读取权限。且密码相同。

配置备份SVN的pre-revprop-change hook

复制代码

#!/bin/sh

USER="$3"

# 限制只有syncuser用户才能提交版本属性修改到此版本库

if [ "$USER" = "syncuser" ]; then exit 0; fi

echo "Only the syncuser user may change revision properties" >&2

exit 1

配置备份SVN的start-commit hook【可选，但建议配上】

复制代码

#!/bin/sh

USER="$2"

# 限制只用syncuser用户才能提交版本修改到版本库

if [ "$USER" = "syncuser" ]; then exit 0; fi

echo "Only the syncuser user may commit new revisions" >&2

exit 1

 初始化备份SVN

$ svnsync init http://192.168.3.10/svn-mirror \

http://192.168.2.5/Dev-rep \

--username syncuser --password syncpass

# 注意

# 提供给svnsync的URL必须是指向目标和源版本库的根目录，这个工具不支持对版本库子树的镜像处理。

在备份SVN测试数据同步

$ svnsync synchronize http://192.168.3.10/svn-mirror \

--username syncuser --password syncpass

在来源SVN配置post-commit hook，启用实时备份

$ svnsync synchronize --non-interactive http://192.168.3.10/svn-mirror \

--username syncuser --password syncpass

# 九、SVN服务关闭重启查看

关闭kill all svnserve kill -9

重启svnserve -d -r /home/svn

查看 ps -ef|grep svnserve

# 十、 SVN升级

1、指定源

打开文件：/etc/yum.repos.d/wandisco-svn.repo，填入以下代码

[WandiscoSVN]

name=Wandisco SVN Repo

baseurl=http://opensource.wandisco.com/centos/$releasever/svn-1.9/RPMS/$basearch/

enabled=1

gpgcheck=0

2、重新安装

卸载原文件

# yum remove subversion\*

3、安装SVN

# yum clean all

# yum install subversion

5、查看svn版本，升级成功

# svn –version