# Svn源码包安装

1软件列表

apr-1.5.2.tar.gz

apr-util-1.5.4.tar.gz

zlib-1.2.11.tar.gz

sqlite-autoconf-3170000.tar.gz

httpd-2.4.25.tar.gz

openssl-1.0.1u.tar.gz

scons-2.5.1-1.noarch.rpm

serf-1.3.9.tar.bz2

subversion-1.9.5.tar.gz

2软件依赖及软件安装

1. 编译安装apr

[root@localhost apr-1.5.2]# tar -zxvf apr-1.5.2.tar.gz

[root@localhost apr-1.5.2]# cd apr-1.5.2/

[root@localhost apr-1.5.2]# ./configure --prefix=/usr/local/apr

[root@localhost apr-1.5.2]# make && make install

验证是够安装成功

[root@localhost apr-1.5.2]# /usr/local/apr/bin/apr-1-config --version

1.5.2

1. 编译安装apr-util

[root@localhost apr-util-1.5.4]# tar -zxvf apr-util-1.5.4.tar.gz

[root@localhost apr-util-1.5.4]# cd apr-util-1.5.4/

[root@localhost apr-util-1.5.4]# ./configure --prefix=/usr/local/apr-util --with-apr=/usr/local/apr

[root@localhost apr-util-1.5.4]# make && make install

验证是否安装成功

[root@localhost apr-util-1.5.4]# /usr/local/apr-util/bin/apu-1-config --version

1.5.4

1. 编译安装zlib

[root@localhost svnsoft]# tar -zxvf zlib-1.2.11.tar.gz

[root@localhost svnsoft]# cd zlib-1.2.11/

安装到默认路径即可（/usr/local/lib）

[root@localhost zlib-1.2.11]# ./configure

[root@localhost zlib-1.2.11]# make && make install

1. 编译安装sqlite3

[root@localhost svnsoft]# tar -zxvf sqlite-autoconf-3170000.tar.gz

[root@localhost svnsoft]# cd sqlite-autoconf-3170000/

安装到默认路径即可（/usr/local/lib）

[root@localhost sqlite-autoconf-3170000]# ./configure

[root@localhost sqlite-autoconf-3170000]# make && make install

1. 编译安装openssl

[root@localhost svnsoft]# tar -zxvf openssl-1.0.1u.tar.gz

[root@localhost svnsoft]# cd openssl-1.0.1u/

[root@localhost openssl-1.0.1u]# ./config --prefix=/usr/local/openssl -fPIC no-gost

[root@localhost openssl-1.0.1u]#make depend

[root@localhost openssl-1.0.1u]#make install

验证安装是否成功

[root@localhost openssl-1.0.1u]# /usr/local/openssl/bin/openssl version

OpenSSL 1.0.1u 22 Sep 2016

为了方便使用，将openssl命令路径加入到PATH环境变量

vim /etc/profie.d/openssl.sh

加入如下内容

export OPENSSL\_HOME=/usr/local/openssl

export PATH=$OPENSSL\_HOME/bin:$PATH

使变量生效

source /etc/profile

直接在命令行中输入openssl version命令，验证环境变量是否生效

[root@localhost profile.d]# openssl version

OpenSSL 1.0.1e-fips 11 Feb 2013

1. 编译安装serf
2. 安装scons

[root@localhost svnsoft]# rpm -ivh scons-2.5.1-1.noarch.rpm

1. 安装serf

[root@localhost svnsoft]# tar -xvf serf-1.3.9.tar.bz2

[root@localhost serf-1.3.9]# cd serf-1.3.9/

[root@localhost serf-1.3.9]# scons PREFIX=/usr/local/serf APR=/usr/local/apr/bin/apr-1-config APU=/usr/local/apr-util/bin/apu-1-config OPENSSL=/usr/local/openssl/

[root@localhost serf-1.3.9]# scons install

安装完成后将serf的lib库路径追加到动态库配置文件中

[root@localhost serf-1.3.9]# echo "/usr/local/serf/lib" >> /etc/ld.so.conf

[root@localhost serf-1.3.9]# ldconfig -v

1. 编译安装httpd2.4

httpd还依赖pcre（正则表达式），为了能够正常编译安装httpd，请确保已经安装了pcre，此处为了方便，使用yum进行安装，确保已经安装了pcre和pcre-devel。

如果没有安装pcre或者pcre-devel，在编译安装httpd过程中，可能会出现如下错误提示

configure: error: pcre-config for libpcre not found. PCRE is required and available from

[root@localhost openssl]# yum -y install pcre

[root@localhost openssl]# yum -y install pcre-devel

注：yum安装前检查yum源的配置，以及光盘的挂载情况

[root@localhost svnsoft]# tar zxvf httpd-2.4.25.tar.gz

[root@localhost svnsoft]# cd httpd-2.4.25/

[root@localhost httpd-2.4.25]# ./configure --prefix=/usr/local/apache2 --with-apr=/usr/local/apr/bin/apr-1-config --with-apr-util=/usr/local/apr-util/bin/apu-1-config --enable-so --enable-dav --enable-maintainer-mode --enable-rewrite --enable-ssl --with-ssl=/usr/local/openssl

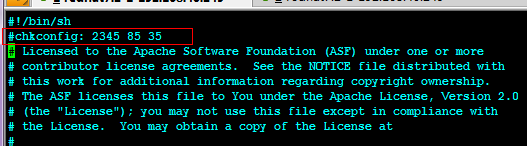
[root@localhost httpd-2.4.25]# make && make install

本机并没有默认安装的httpd,此处我们将编译安装的httpd设置为开启自动启动，方法如下

[root@localhost httpd-2.4.25]# cp /usr/local/apache2/bin/apachectl /etc/init.d/httpd

在#/bin/shh的下面加入如下行，保存退出

#chkconfig:2345 85 35



设置httpd开启自动启动

[root@localhost httpd-2.4.25]# chkconfig httpd on

检查确认，2345级别为on

[root@localhost httpd-2.4.25]# chkconfig --list httpd

注意：该输出结果只显示 SysV 服务，并不包含原生 systemd 服务。SysV 配置数据可能被原生 systemd 配置覆盖。

如果您想列出 systemd 服务,请执行 'systemctl list-unit-files'。

欲查看对特定 target 启用的服务请执行

'systemctl list-dependencies [target]'。

httpd 0:关 1:关 2:开 3:开 4:开 5:开 6:关

为了方便使用，将对应命令路径加入环境变量，添加环境变量文件

[root@localhost httpd-2.4.25]# cd /etc/profile.d/

[root@localhost apache2]# vim httpd.sh

##文件内容如下

export HTTPD\_HOME=/usr/local/apache2

export PATH=$HTTPD\_HOME/bin:$PATH

使变量生效

[root@localhost profile.d]# source /etc/profile

验证

[root@localhost profile.d]# httpd -version

Server version: Apache/2.4.25 (Unix)

Server built: Nov 19 2017 19:18:35

1. 编译安装subversion

[root@localhost svnsoft]# tar -zxvf subversion-1.9.5.tar.gz

[root@localhost svnsoft]# cd subversion-1.9.5/

[root@localhost subversion-1.9.5]# ./configure --prefix=/usr/local/svn --with-apxs=/usr/local/apache2/bin/apxs --with-apr=/usr/local/apr/bin/apr-1-config --with-apr-util=/usr/local/apr-util/bin/apu-1-config --with-zlib --enable-maintainer-mode --with-serf=/usr/local/serf

[root@localhost subversion-1.9.5]# make && make install

将svn加入系统环境变量

[root@localhost subversion-1.9.5]# cd /etc/profile.d/

[root@localhost profile.d]# vim svn.sh

##内容如下

export SVN\_HOME=/usr/local/svn

export PATH=$SVN\_HOME/bin:$PATH

使环境变量生效

[root@localhost profile.d]# source /etc/profile

使用svnserve –version命令查看

[root@localhost openssl]# svnserve --version

svnserve，版本 1.7.14 (r1542130)

编译于 Aug 12 2015，15:46:21

版权所有 (C) 2013 Apache 软件基金会。

此软件包含了许多人的贡献，请查看文件 NOTICE 以获得更多信息。

Subversion 是开放源代码软件，请参阅 http://subversion.apache.org/ 站点。

下列版本库后端(FS) 模块可用:

\* fs\_base : 模块只能操作BDB版本库。

\* fs\_fs : 模块与文本文件(FSFS)版本库一起工作。

Cyrus SASL 认证可用。

**代码库创建**

SVN软件安装完成后还需要建立SVN库  
[root@localhost modules]#mkdir -p /opt/svn/             #创建SVN根目录  
[root@localhost modules]# svnadmin create --pre-1.6-compatible /opt/svn/repo #创建repo测试库  
执行上面的命令后，自动建立repo测试库，查看/opt/svn/repo 文件夹发现包含了conf, db,format,hooks, locks, README.txt等文件，说明一个SVN库已经建立。

注：

svnadmin create --pre-1.6-compatible /opt/svn/repo

注意:如果这里不写--pre-1.6-compatible,更新svn的时候会报如下错误，--pre-1.6-compatible表示使用与1.6版本的svn兼容的模式。

post commit FS processing had error:

Couldn't open rep-cache database

**配置代码库**

进入上面生成的文件夹conf下，进行配置  
[root@localhost modules]#cd /opt/svn/repo/conf

**用户密码passwd配置**

[root@localhost password]#cd /opt/svn/repo/conf  
[root@admin conf]#vim passwd  
修改passwd为以下内容：

[users]  
# harry = harryssecret  
# sally = sallyssecret

hello = 123

aaa = 123

www = 123

用户名=密码

这样我们就建立了hello用户， 123密码

以上语句都必须顶格写, 左侧不能留空格, 否则会出错.

**权限控制authz配置**

[root@admin conf]#vim authz  
目的是设置哪些用户可以访问哪些目录，向authz文件追加以下内容：

[groups]            #组

admin = hello,www   #创建一个admin组，将用户加入到组

[/]                 #根目录权限设置（就是“svn”这个文件夹）

aaa = rw            #aaa对svn下的所有版本库有读写权限

[repo:/]            #repo:/,表示对repo版本库下的所有资源设置权限

@admin = rw         #admin组的用户对repo版本库有读写权限

[repo2:/occi], ,表示对版本库repo2中的occi项目设置权限

[repo2:/occi/aaa], ,表示对版本库2中的occi项目的aaa目录设置权限　　权限主体可以是用户组、用户或\*，用户组在前面加@，\*表示全部用户。权限可以是w、r、wr和空，空表示没有任何权限。

以上语句都必须顶格写, 左侧不能留空格, 否则会出错.

**服务svnserve.conf配置**

[root@admin conf]#vim svnserve.conf

追加以下内容：

[general]  
#匿名访问的权限，可以是read,write,none,默认为read  
anon-access = none  
#使授权用户有写权限  
auth-access = write  
#密码数据库的路径  
password-db = passwd  
#访问控制文件  
authz-db = authz  
#认证命名空间，subversion会在认证提示里显示，并且作为凭证缓存的关键字

realm = /opt/svn/repo

以上语句都必须顶格写, 左侧不能留空格, 否则会出错.

**配置防火墙端口(或直接关闭防护墙)**

~~[root@localhost conf]#vim /etc/sysconfig/iptables  
添加以下内容：  
-A INPUT -m state --state NEW -m tcp -p tcp --dport 3690 -j ACCEPT  
保存后重启防火墙  
[root@localhost conf]#service iptables restart~~

[root@localhost repo]# systemctl stop firewalld.service

**启动SVN**

svnserve -d -r /opt/svn/         #指定SVN根目录

**查看SVN进程**

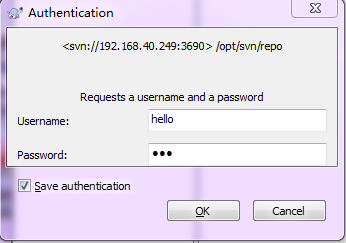
ps -ef|grep svnserve

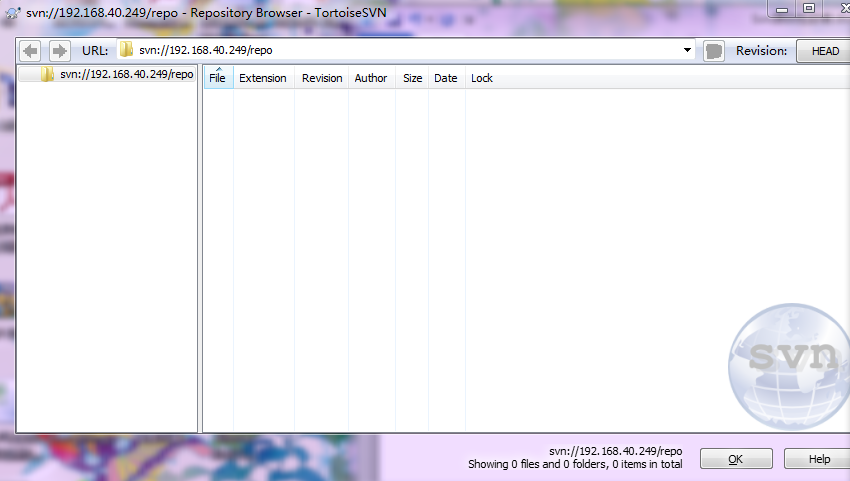
windows客户端验证

地址：svn://192.168.40.249/repo

用户名：hello

密码：123





整合svn与httpd

初步整合httpd与svn，以达到使用http地址访问svn的目的，这也是配置https的基础。

进行httpd相关设置

将httpd访问svn依赖的模块从svn目录拷贝到httpd的模块目录，以便httpd使用

cp /usr/local/svn/libexec/mod\_authz\_svn.so /usr/local/apache2/modules/

cp /usr/local/svn/libexec/mod\_dav\_svn.so /usr/local/apache2/modules/

编辑httpd的配置文件，加载对应的模块，并且设置对应location，以便客户端能够访问。

在httpd.conf配置文件中的最后加入如下内容（两个"#for svn"之间的内容都要加入httpd.conf,部分内容可根据你的实际情况进行修改）

#for svn

#加载整合svn所需的模块文件，刚才已经将这些动态库拷贝到了modules目录中

LoadModule dav\_svn\_module modules/mod\_dav\_svn.so

LoadModule authz\_svn\_module modules/mod\_authz\_svn.so

#设置对应location，以便访问对应svn仓库

#此处将location路径设置为/，也就是说，通过浏览器访问svn时，IP地址后面的"/"就对应了svn的根仓库目录。

#SVNParentPath设置为svn仓库的总根目录，此处设置为/data,而Location设置为"/",所以，在IP地址中输入 即表示访问的是文件系统中的/data目录中的svn仓库根目录。

#AuthzSVNAccessFile设置svn权限配置文件位置

#AuthUserFile 设置svn用户名密码配置文件位置

<Location />

DAV svn

SVNParentPath /opt/svn

SVNListParentPath On

SVNAutoversioning On

SVNReposName "svn"

AuthzSVNAccessFile /opt/svn/authz.conf

AuthType Basic

AuthName "svn repo auth"

AuthUserFile /opt/svn/passwd.conf

Require valid-user

</Location>

#for svn

注：ServerName配置

#

ServerName localhost:80

apache线程的用户为daemon用户，所以，为了httpd能够正常访问此仓库，将仓库目录为daemon用户授权

[root@redhat72 svn]# cd /opt/

[root@redhat72 opt]# chown -R daemon svn

chmod -R 755 svn

配置认证需要的相关文件

使用htpasswd命令，生成用户名和密码，密码默认经过MD5加密，非明文密码

htpasswd -c /opt/svn/passwd.conf dsw

使用上述命令将会创建passwd.conf文件，并且添加一个用户名为dsw的用户，同时提示输入用户的密码。

以后再次添加新的svn用户时，不要使用-c选项，示例如下

htpasswd /opt/svn/passwd.conf abc

查看/opt/svn/passwd.conf文件的内容，发现已经存在两个svn用户，对应密码已经经过加密。

passwd.conf文件已经配置完毕，现在开始配置权限控制文件，authz.conf

vim /opt/svn/authz.conf

内容如下

[aliases]

# joe = /C=XZ/ST=Dessert/L=Snake City/O=Snake Oil, Ltd./OU=Research Institute/CN=Joe Average

[groups]

admin = dsw,abc

# harry\_and\_sally = harry,sally

# harry\_sally\_and\_joe = harry,sally,&joe

# [/foo/bar]

# harry = rw

# &joe = r

# \* =

# [repository:/baz/fuz]

# @harry\_and\_sally = rw

# \* = r

[/]

aaa = rw

[repo:/]

@admin = rw

注：可从生成的模板 authz复制生成

/opt/svn/repo/conf/authz

初步测试

完成上述步骤后，可以进行初步的测试

但是为了保险起见，重启svn服务

killall svnserve

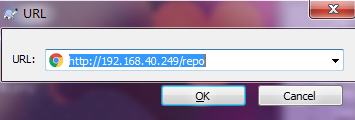
svnserve -d -r /opt/svn

然后启动httpd

service httpd start

查看httpd的日志：/usr/local/apache2/logs

Windows端访问



配置HTTPS

生成rsa证书私钥

openssl genrsa 2048 > /opt/svn/server.key

根据私钥生成证书请求文件，执行如下命令，根据提示填写证书申请信息

openssl req -new -key /opt/svn/server.key > /opt/svn/server.csr

[root@redhat72 logs]# openssl req -new -key /opt/svn/server.key > /opt/svn/server.csr

You are about to be asked to enter information that will be incorporated

into your certificate request.

What you are about to enter is what is called a Distinguished Name or a DN.

There are quite a few fields but you can leave some blank

For some fields there will be a default value,

If you enter '.', the field will be left blank.

-----

Country Name (2 letter code) [AU]:CN

State or Province Name (full name) [Some-State]:

Locality Name (eg, city) []:

Organization Name (eg, company) [Internet Widgits Pty Ltd]:

Organizational Unit Name (eg, section) []:

Common Name (e.g. server FQDN or YOUR name) []:

Email Address []:

Please enter the following 'extra' attributes

to be sent with your certificate request

A challenge password []:

An optional company name []:

[root@redhat72 logs]#

根据私钥和证书请求文件生成自签证书。

[root@redhat72 logs]# openssl req -x509 -days 36500 -key /opt/svn/server.key -in /opt/svn/server.csr > /opt/svn/server.crt

将私钥与证书拷贝至httpd的配置文件目录。

[root@redhat72 logs]# cp /opt/svn/server.key /usr/local/apache2/conf/

[root@redhat72 logs]# cp /opt/svn/server.crt /usr/local/apache2/conf/

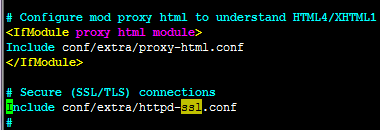
[root@redhat72 logs]#

设置httpd支持ssl

编辑httpd配置文件，将Include conf/extra/httpd-ssl.conf前面的注释去掉

[root@redhat72 conf]# cd /usr/local/apache2/conf

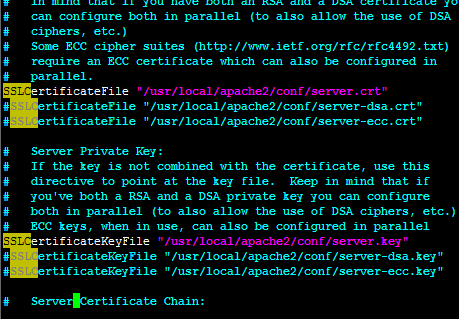
[root@redhat72 conf]# vim httpd.conf



确定httpd-ssl.conf配置文件中的证书与私钥指定正确，如下

[root@redhat72 extra]# cd /usr/local/apache2/conf/extra

[root@redhat72 extra]# vim httpd-ssl.conf



完成上述步骤后，修改svn对应的location

**[root@redhat72 extra]# vim /usr/local/apache2/conf/httpd.conf**

在对应的location中加入如下配置



完成上述配置后重启httpd

service httpd restart

使用https访问对应仓库地址即可