SVN - ENVIRONMENT SETUP

http://www.tutorialspoint.com/svn/svn_environment.htm

SVN Installation

Subversion is a popular open-source version control tool. It is open-source and available for free over the internet. It comes by default with most of the GNU/Linux distributions, so it might be already installed on your system. To check whether it is installed or not use following command.

```
[jerry@CentOS ~]$ svn --version
```

If Subversion client is not installed, then command will report error, otherwise it will display the version of the installed software.

```
[jerry@CentOS ~]$ svn --version
-bash: svn: command not found
```

If you are using RPM-based GNU/Linux, then use ***yum*** command for installation. After successful installation, execute the ***svn --version*** command.

```
[jerry@CentOS ~]$ su -
Password:
[root@CentOS ~]# yum install subversion

[jerry@CentOS ~]$ svn --version
svn, version 1.6.11 (r934486)
compiled Jun 23 2012, 00:44:03
```

And if you are using Debian-based GNU/Linux, then use ***apt*** command for installation.

```
[jerry@Ubuntu]$ sudo apt-get update
[sudo] password for jerry:

[jerry@Ubuntu]$ sudo apt-get install subversion

[jerry@Ubuntu]$ svn --version
svn, version 1.7.5 (r1336830)
compiled Jun 21 2013, 22:11:49
```

Apache Setup

We have seen how to install Subversion client on GNU/Linux. Let us see how to create a new repository and allow access to the users.

On server we have to install ***Apache httpd*** module and ***svnadmin*** tool.

```
[jerry@CentOS ~]$ su -
Password:
[root@CentOS ~]# yum install mod_dav_svn subversion
```

The ***mod_dav_svn*** package allows access to a repository using HTTP, via Apache httpd server and ***subversion*** package installs svnadmin tool.

The subversion reads its configuration from ***/etc/httpd/conf.d/subversion.conf*** file. After adding configuration, ***subversion.conf*** file looks as follows:

```
LoadModule dav_svn_module     modules/mod_dav_svn.so
LoadModule authz_svn_module   modules/mod_authz_svn.so

<Location /svn>
   DAV svn
   SVNParentPath /var/www/svn
   AuthType Basic
   AuthName "Authorization Realm"
   AuthUserFile /etc/svn-users
   Require valid-user
</Location>
```

Let us create Subversion users and grant them access to the repository. *htpasswd* command is used to create and update the plain-text files which are used to store *usernames* and *passwords* for basic authentication of HTTP users. '-c' options creates *password* file, if *password* file already exists, it is overwritten. That is why use '-c' option only the first time. '-m' option enables MD5 encryption for passwords.

User Setup

Let us create user ***tom***.

```
[root@CentOS ~]# htpasswd -cm /etc/svn-users tom
New password:
Re-type new password:
Adding password for user tom
```

Let us create user ***jerry***

```
[root@CentOS ~]# htpasswd -m /etc/svn-users jerry
New password:
Re-type new password:
Adding password for user jerry
[root@CentOS ~]#
```

Create Subversion parent directory to store all the work (see */etc/httpd/conf.d/subversion.conf*).

```
[root@CentOS ~]# mkdir /var/www/svn
[root@CentOS ~]# cd /var/www/svn/
```

Repository Setup

Create a project repository named *project_repo*. *svnadmin* command will create a new repository and a few other directories inside that to store the metadata.

```
[root@CentOS svn]# svnadmin create project_repo

[root@CentOS svn]# ls -l project_repo
total 24
drwxr-xr-x. 2 root root 4096 Aug  4 22:30 conf
drwxr-sr-x. 6 root root 4096 Aug  4 22:30 db
-r--r--r--. 1 root root    2 Aug  4 22:30 format
drwxr-xr-x. 2 root root 4096 Aug  4 22:30 hooks
drwxr-xr-x. 2 root root 4096 Aug  4 22:30 locks
-rw-r--r--. 1 root root  229 Aug  4 22:30 README.txt
```

Let us change the user and group ownership of the repository.

```
[root@CentOS svn]# chown -R apache.apache project_repo/
```

Check whether *SELinux* is enabled or not using the SELinux status tool.

```
[root@CentOS svn]# sestatus
SELinux status:                 enabled
SELinuxfs mount:                /selinux
Current mode:                   enforcing
Mode from config file:          enforcing
Policy version:                 24
Policy from config file:        targeted
```

For our server, SELinux is enabled, so we have to change the SELinux security context.

```
[root@CentOS svn]# chcon -R -t httpd_sys_content_t /var/www/svn/project_repo/
```

To allow commits over HTTP, execute the following command.

```
[root@CentOS svn]# chcon -R -t httpd_sys_rw_content_t /var/www/svn/project_repo/
```

Restart the Apache server and we are done with the configuration of Apache server.

```
[root@CentOS svn]# service httpd restart
Stopping httpd:                                              [FAILED]
Starting httpd: httpd: apr_sockaddr_info_get() failed for CentOS
httpd: Could not reliably determine the server's fully qualified domain name, using
127.0.0.1 for ServerName
                                                              [  OK  ]
[root@CentOS svn]# service httpd status
httpd (pid  1372) is running...
[root@CentOS svn]#
```

We have configured the Apache server successfully, now we will configure the repository. To provide repository access to only authentic users and to use the default authorization file; append the following lines to *project_repo/conf/svnserve.conf* file.

```
anon-access = none
authz-db = authz
```

Conventionally, every Subversion project has ***trunk, tags,*** and ***branches*** directories directly under the project's root directory.

The *trunk* is a directory where all the main development happens and is usually checked out by the developers to work on the project.

The *tags* directory is used to store named snapshots of the project. When creating a production release, the team will tag the code that goes into the release.

The *branches* directory is used when you want to pursue different lines of development.

Let us create *the trunk, tags, and branches* directory structure under the project repository.

```
[root@CentOS svn]# mkdir /tmp/svn-template
[root@CentOS svn]# mkdir /tmp/svn-template/trunk
[root@CentOS svn]# mkdir /tmp/svn-template/branches
[root@CentOS svn]# mkdir /tmp/svn-template/tags
```

Now import the directories from ***/tmp/svn-template*** to the repository.

```
[root@CentOS svn]# svn import -m 'Create trunk, branches, tags directory structure'
/tmp/svn-template/
Adding         /tmp/svn-template/trunk
Adding         /tmp/svn-template/branches
Adding         /tmp/svn-template/tags
Committed revision 1.
[root@CentOS svn]#
```

This is done! We have successfully created the repository and allowed access to ***Tom*** and ***Jerry***. From now, they can perform all the supported operations to the repository.