

## SVN 搭建服务器的全过程

Apache 网站

<http://apr.apache.org/download.cgi>

需要的插件:

<http://mirrors.cnnic.cn/apache//apr/apr-1.5.2.tar.gz>

<http://mirrors.cnnic.cn/apache//apr/apr-util-1.5.4.tar.gz>

Subversion 的安装包:

<http://subversion.tigris.org/downloads/subversion-1.6.6.tar.gz>

<http://subversion.tigris.org/downloads/subversion-deps-1.6.6.tar.gz>

本次操作全程在 linux 的 CentOS 系统中完成。

安装过程中需要用到的 linux 命令

#检查是否安装了低版本的 SVN

[root@localhost /]#rpm -qa subversion

#如果存储旧版本, 卸载旧版本 SVN

[root@localhost modules]#yum remove subversion

#安装 SVN

[root@localhost modules]#yum install subversion

#解压命令

tar zxvf subversion-1.6.6.tar.gz

tar zxvf subversion-deps-1.6.6.tar.gz

将包下载完之后, 就可以解压.解压命令"tar zxvf subversion-1.6.6.tar.gz"下载的两个包解压之后就安装该文件, 找到解压之后的包运行./configure --prefix=安装路径。

开始安装 subversion 文件在 cd ./subversion-1.6.6 文件下编译./configure --prefix=/usr/local/svn --with-openssl=/usr/include/openssl --without-berkeley-db

后面以 svnserve 方式运行, 所以不加 apache 编译参数。以 fsfs 格式存储版本库, 不编译 berkeley-db

如果编译时报如下错误:

no acceptable C compiler found in \$PATH

说明没有 gcc 库, 使用如下命令安装 gcc 后再编译:

yum -y install gcc

安装

make

make install

如果 make install 出现下面错误:

/home/upload/subversion-1.6.6/subversion/svnversion/.libs/lt-svnversion: error while loading shared libraries: libiconv.so.2: cannot open shared object file: No such file or directory

make: \*\*\* [revision-install] Error 127

解决办法:

1、编辑/etc/ld.so.conf 文件

vi /etc/ld.so.conf

添加下面一行代码

/usr/local/lib

2、保存后运行 ldconfig:

/sbin/ldconfig

注: ld.so.conf 和 ldconfig 用于维护系统动态链接库。

3、然后再安装

make && make install

测试是否安装成功

/usr/local/svn/bin/svnserve --version

当前版本

svnserve, version 1.6.6 (r40053)

compiled Dec 25 2012, 13:14:38

Copyright (C) 2000-2009 CollabNet.

Subversion is open source software, see <http://subversion.tigris.org/>

This product includes software developed by CollabNet

(<http://www.Collab.Net/>).

The following repository back-end (FS) modules are available:

\* fs\_fs : Module for working with a plain file (FSFS) repository.

Cyrus SASL authentication is available.

SVN 安装完毕之后, 就开始安装相应的插件

<http://mirrors.cnnic.cn/apache//apr/apr-1.5.2.tar.gz>

<http://mirrors.cnnic.cn/apache//apr/apr-util-1.5.4.tar.gz>

这两个插件可以直接下载

wget <http://mirrors.cnnic.cn/apache//apr/apr-1.5.2.tar.gz>

wget <http://mirrors.cnnic.cn/apache//apr/apr-util-1.5.4.tar.gz>

可以用 wget 命令在 linux 下面进行下载, 下载完毕之后输入命令"ls"就可以看到下载完毕之后的文件, 就将两个文件进行解压, 解压方式如下

tar -zxvf 文件名

[tar -zxvf apr-1.5.2.tar.gz](#)

这两个文件解压完毕之后输入命令"ls"就能看到当前目录下的文件名为 **apr-1.5.2** 的文件夹, 然后进入该文件夹中, 输入命令"ls"就能在当前目录下文件为 **configure** 的文件夹, 然后输入命令"./configure--prefix=/usr/local/apr", 这是说明编译 configuer 这个文件到/usr/local/apr 文件夹当中。

将文件安装完毕之后, 就输入命令"make"命令进行编译。

Make 编译完毕之后再输入命令"make install"命令, 进行编译。。。。

Make install 编译完成之后, 就返回上一个目录, 进行解压安装第二个文件, 安装编译类似。

`tar -zxvf apr-util-1.5.4.tar.gz`

`apr-util-1.5.4.tar.gz` 文件包根据以上安装方法类似。

唯有不同的就是安装路径不一样,安装路径需要变更成`"/configure --prefix=/usr/local/apr-util --with-apr=/usr/local/apr"`

这两个文件解压完毕之后输入命令`"ls"`就能看到当前目录下的文件名为 `apr-util-1.5.4` 的文件夹,然后进入该文件夹中,输入命令`"ls"`就能在当前目录下文件为 `configure` 的文件夹,然后输入命令`"/configure --prefix=/usr/local/apr-util --with-apr=/usr/local/apr"`,这是说明编译 `configure` 这个文件到 `/usr/local/apr` 文件夹当中。

将文件安装完毕之后,就输入命令`"make"`命令进行编译。

`Make` 编译完毕之后再输入命令`"make install"`命令,进行编译。。。

`Make install` 编译完成之后,我们就需要更改一个内部文件为 `ld.so.conf` 的一个文件,输入命令`"vi /etc/ld.so.conf"`打开该文件之后,将刚刚安装的两个文件的路径放入到该文件里面。

`ld.so.conf` 文件

---

`include ld.so.conf.d/*.conf`

`/usr/local/apr/lib`

`/usr/local/apr-util/lib`

---

该文件编辑好之后保存,再进行加载,输入命令`"ldconfig -v"`。

到此为止,插件就安装好了。

`Cd /` 回到根目录

下一步就创建仓库的路径。

输入`"mkdir network"`命令创建这个为网络的 `network` 的文件,然后进入 `network` 文件夹中再创建一个子文件夹为 `svn` 的文件。

然后又进入 `svn` 文件夹当中创建 `project_1,project_2` 两个文件夹。

然后就创建 `svnadmin`,输入命令`"svnadmin create /network/svn/project_1"`

在这里需要注意的是:有可能 `svnadmin` 或者 `svnversion` 命令失败!这个该怎么解决呢!如果是 `svnadmin` 出现的问题,就在刚刚我们用 `vi` 打开的一个叫 `ld.so.conf` 的这么一个文件,里面把我们所安装的插件的路径存入到里面,先在这里介绍一下 `ld.so.conf` 文件的作用是我们电脑上面的环境变量配置一样的道理,所以只要你将你所安装的文件的路径放入到这个文件当中,你运行命令的时候就不需要手动输入路径运行,只需要输入命令即可。如果 `svnversion` 这个命令出现失败的话,也可以将我们安装的 `svnversion` 的路径放入到 `ld.so.conf` 文件中,这样就可以找到这个路径,还有最简单的方法就是:`echo $PATH` 查看环境变量下来的路径。将下面的路径到 `expor PATH=****`,\*\*\*\*就是你的 `svn` 的目录和刚刚打出来的路径都要写入进去,这种方法不提倡,因为如果你关闭了终端这个设置就失效了。

接着就输入命令`"svnadmin create /network/svn/project_2"`,好的我们现在就创建好了两个仓库, `project1/2` 两个仓库文件。

进入到这两个库中,输入命令`"cd /network/svn/project_1"`,然后输入`"ls"`,查看当前目录下的文件,这里面有很多文件,最重要的一个文件就是 `conf` 文件,然后进入到这个文件里面。

进入到 `conf` 这个文件之后呢,就找到 `svnserve.conf` 这个文件,需要配置这个文件。

注意: **Svnserve.conf** 文件配置, 将一下的内容进行修改, 修改的时候前面不能有空格, 去掉#

```
anon-access = read
auth-access = write
password-db = passwd
authz-db = authz
```

这里很重要, 不能配置出错, 以上的内容在 **svnserve.conf** 文件里面是有的, 只需要把前面的#去掉即可, 一定要小心, 前面不能有空格。

然后就配置第二个文件 **passwd** 文件, 这个文件就比较简单, 这个文件主要是 **svn** 用户登录的配置。

```
[users]
Zhangdl = Zhangdl
project1 = project1
project2 = project2
```

这里的是我自己创建的三个用户, 这个用户名是自己随意取的。

最后就是将 **authz** 这个文件进行配置, 用 **vi authz** 这个命令打开这个文件。

```
[groups]
admin = Zhangdl          ##这是 admin 管理员, 最高权限
project1 = project1      ##这是两个用户
project2 = project2

[/]
@admin = rw              ##admin 授权可读可写

[project1:/]
@admin = rw              ##对 project1 的权限, admin 可以对它读写
@project1 = rw           ##当然自己对自己也可以读写
```

**cp -r ./project\_1/conf ./** 这个句话的命令就是把 **conf** 的文件复制到当前文件下面。

然后就 **Liunx** 的防火墙, 修改防火墙的权限, 修改允许。因为默认是禁止

命令进入 **vi /etc/sysconfig/iptables**

文件打开之后, 在当前的文件下输入:

```
-A RH-Firewall-1-INPUT -p tcp -m state -m tcp --dport 3690 --state NEW -j ACCEPT
```

设置之后需要重启防火墙, **service iptables restart** 命令重新启动防火墙。

最后一步启动 **SVN** 服务器。

输入命令"**svnserve -d -r /network/svn/**"

输入服务器的 **IP** 地址就可以连接到 **SVN** 服务器了。

**svn://192.168.121.128/project\_1**

## Linux 中的环境变量配置方法

有两种配置方法：①临时有效。②永久有效

当然我们肯定要选择永久有效，如果你选择临时有效的話，你每次 SVN 服务器都要去手动配置一下 linux 的环境，这样显得很麻烦，在这里我把两种方法都写出来。

在终端下面输入即可.....

临时有效： **export PATH = /usr/local/svn/bin**

永久有效：

临时有效的方法很简单，只要在终端下面输入命令就可以搞定，但是永久有效肯定比较麻烦，首先我们要打开 **profile** 这个文件，输入命令“vi /etc/profile”

文件内容如下显示：

---

```
# /etc/profile
# System wide environment and startup programs, for login setup
# Functions and aliases go in /etc/bashrc
# are doing. It's much better to create a custom.sh shell script in
# /etc/profile.d/ to make custom changes to your environment, as this
# will prevent the need for merging in future updates.
pathmunge () {
    case ":{PATH}:" in
        *:"$1":*)
            ;;
        *)
            if [ "$2" = "after" ]; then
                PATH=$PATH:$1
            else
                PATH=$1:$PATH
            fi
    esac
}
```

**xport PATH = \$PATH:/usr/local/svn/bin**

```
for i in /etc/profile.d/*.sh ; do
    if [ -r "$i" ]; then
        if [ "${-#*i}" != "$-" ]; then
            . "$i"
        else
            . "$i" >/dev/null 2>&1
        fi
    fi
done
unset i
unset -f pathmunge
```

加红色那一段命令是我们自己加入了，只要加入到 for 循环前面就可以了。然后输入“source /etc/profile”如果没有出现报错就可以了。就说明配置成功！

---