



链滴

# svn 安装、配置及 apache 支持

作者: [jusong](#)

原文链接: <https://ld246.com/article/1567749512709>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

**svn**作为版本管理系统，安装还是很简单的，主要的麻烦在于配置项目的访问权限，比如创建用户、用户分组、为用户分配访问权限等。

另外**svn**可以结合**apache**实现通过**http**协议进行访问，通过**http**协议进行访问时，账号授权方式和过**svn**直接访问有共用的地方也有不同的地方。

---

## 一、svn安装及配置

### 安装svn

这个比较简单，不同发行版通过不同包管理工具直接进行安装即可，如果发行版提供的版本不合适也去官网下载源码编译安装即可，我用的**freebsd**，安装方式如下：

```
>>> sudo pkg install subversion
>>> rehash      # csh需要执行这个
```

### 创建svn仓库

```
>>> mkdir /home/svn                # 创建svn基础目录
>>> svnadmin create /home/svn/project  # 在基础目录下创建仓库
```

上面命令会创建一个**svn**仓库，名字是**project**，里面的目录结构如下：

```
>>> tree -L 1 /home/svn/project/
/home/svn/project/
|-- README.txt
|-- conf
|-- db
|-- format
|-- hooks
-- locks
```

其中重点是**conf**目录，里面包含了**project**这个项目的用户账号、密码、访问权限及其他基础配置，户添加和权限控制就在这里修改了。

### 启动服务

```
>>> svnserve -d -r /home/svn
```

上面的命令会以当前用户运行服务，这就要求当前用户对**/home/svn**目录有相应的读写权限。

这样就启动了**svn**服务

### 配置svn仓库

首先看一下仓库配置文件目录结构

```
>>> tree /home/svn/project/conf/
/home/svn/project/conf/
```

```
|-- authz                # 授权配置文件
|-- hooks-env.templ
|-- passwd               # 账号、密码配置文件
`-- svnserve.conf        # 基础配置文件
```

## 1、基础配置

配置文件 **svnserve.conf**:

```
[general]
anon-access = none        # 禁止匿名用户访问
auth-access = write       # 已授权用户可读写
password-db = passwd      # 指定账号密码配置文件位置
authz-db = authz          # 指定授权配置文件位置
realm = My First Repository # 认证时的提示
```

## 2、创建用户

配置文件 **passwd**:

```
[users]
user1 = 123456
user2 = 123456
user3 = 123456
```

上面就是增加三个用户 **user1**、**user2**、**user3** 用户，密码是明文：**123456**

## 3、用户授权

配置文件 **authz**:

```
[groups]                # 分组配置
g_admin = user1,user2    # 建立一个分组 g_admin, 成员: user1, user2

[/]                      # 通用（所有项目）的根目录的授权配置
@g_admin = rw            # g_admin 分组的所有成员有读写权限
user3 = rw               # user3 有读权限

[/code/java]             # 通用（所有项目）的 /code/java 授权配置
user3 = r                # user3 有读写权限

[project:/code/java]     # project 项目的根目录的授权配置
@g_admin = rw            # g_admin 分组的所有成员有读写权限
user3 = rw               # user3 有读写权限
```

**svn** 的项目路径是:

svn://[host]/[project]

比如 **project** 项目的根路径:

svn://127.0.0.1/project

对于授权配置需要注意以下几点：

- 授权配置`authz`文件中的路径都是基于项目的根路径的
- 指明了项目名称的授权路径，其权限的优先级要高于通用的授权路径的配置

---

## 二、svn 增加 apache 支持

svn 增加 `apache` 支持后，便可以通过`http`协议的方式访问，可以直接在浏览器中浏览项目

### 安装 `apache`

如已经安装`apache`可跳过，可以使用包管理器进行安装，我使用的`freebsd`，安装方式如下

```
>>> sudo pkg install apache24
```

安装后可以在`/etc/rc.conf`中配置开启启动`apache`服务

```
>>> cat /etc/rc.conf
apache24_enable="YES"
```

### 安装 `mod_dav_svn`

该模块用于在`apache`中支持访问`svn`

```
>>> sudo pkg install mod_dav_svn
```

### 配置 `svn` 支持

首先需要打开`dav_module`模块，如下

```
>>> cat /usr/local/etc/apache24/httpd.conf | grep dav
LoadModule dav_module libexec/apache24/mod_dav.so
```

然后配置`svn`的路径，如下

```
>>> cat /usr/local/etc/apache24/modules.d/220_subversion.conf | grep -v # | grep -v ^$
LoadModule dav_svn_module      libexec/apache24/mod_dav_svn.so
LoadModule authz_svn_module    libexec/apache24/mod_authz_svn.so
<IfModule dav_svn_module>
<Location /svn/>
  <ifModule mime_module>
    RemoveEncoding .gz .tgz .Z
    RemoveType     .gz .tgz .Z
  </ifModule>
  DAV svn
  SVNParentPath /home/svn/          # 设置 svn 基础目录
  SVNListParentPath on               # 可以看到项目列表
  SVNAutoversioning off
  SVNReposName "Project XYZ repo"
  AuthzSVNAnonymous off            # 禁止匿名访问
```

```
SVNIndexXSLT "/style/svnindex.xsl"
AuthType Basic # 基础认证类型
AuthName "Authorization Realm"
AuthUserFile /home/svn/project/conf/htpasswd # 指定账号配置文件的位置
AuthzSVNAccessFile /home/svn/project/conf/authz # 指定授权文件位置
Require valid-user # 只允许授权的用户访问
LimitXMLRequestBody 0
</Location>
</IfModule>
```

上面配置中，重点关注`AuthUserFile`，这里存放用户账号信息，注意这个账号信息和上面说的`svn`账配置不一致，`svn`原生配置中使用的密码是明文，这里的账号密码使用的是密文，而生成密文需要借助`htpasswd`工具，这个是`apache`自带的，使用方式如下：

```
>>> htpasswd -nb user1 123456
user1:$apr1$heAgGt3/$V28L3wRUGLXGJGecUdYzr/
```

上面生成的是用户名为`user1`，密码为`123456`的账号，将生成的账号密码信息放入`/home/svn/project/conf/htpasswd`文件，注意还需要在`/home/svn/project/conf/authz`中配置账号`user1`的授权信息然后重启`apache`服务重启即可在浏览器中正常访问了，访问路径如下：

`http://[host]/svn/[project]`

如：

`http://127.0.0.1/svn/project`

然后输入账号和密码即可。

### 三、使用svn子域名

可以借助 `nginx` 的反向代理功能将访问域名改成`svn`子域名，去掉路径中的`svn`，比如

```
http://www.example.com/svn/project
替换成如下：
http://svn.example.com/project
```

这样需要`apache`让出`80`端口，改用其他端口，由`nginx`监听`80`端口，配置如下：

### apache 配置

Listen 82

### nginx 配置

```
>>> cat /usr/local/etc/nginx/sites/svn.conf
upstream svn_backend {
    server localhost:82 max_fails=2 fail_timeout=30s;
}

server {
```

```
listen 80;
server_name svn.example.com;

charset utf-8;
access_log /var/log/nginx/access.log main;

location / {
    proxy_set_header Host $host:$server_port;
    proxy_set_header X-Real-IP $remote_addr;
    client_max_body_size 10m;
    proxy_pass http://svn_backend/svn/;
}
```

重启[apache](#)和[nginx](#)后就可以使用[svn.example.com](#)访问了。

**注意：**如果修改后没有变化，可能是浏览器缓存导致的，所以要禁用缓存，否则就要查看[apache](#)和[ngx](#)的日志查看具体原因了