

SVN服务实战应用指南

惨绿少年 Linux运维, 运维基本功

0评论

来源: 本站原创

43°C

字体:

小

中

大

本文转载自:

<https://www.abcdocker.com/abcdocker/2813>

一、什么是SVN (subversion) ?

SVN是近年来崛起的非常优秀的版本管理工具, 与CVS管理工具一样, SVN是一个固态的跨平台的开源的版本控制系统。SVN版本管理工具管理者随时间改变的各种数据。这些数据放置在一个中央资料档案库 (repository) 中, 这个档案库很像一个普通的文件服务器或者FTP服务器, 但是, 与其他服务器不同的是, SVN会备份并记录每个文件每一次的修改更新变动。这样我们就可以把任意一个时间点的档案恢复到想要的某一个旧的版本, 当然也可以直接浏览指定的更新历史记录。

为什么会有SVN这样一个项目?

官方解释: 为了接管CVS的用户基础, 确切的说, 我们写了一个新的版本控制系统, 它和CVS很相似, 但是它修正了以前CVS所没有解决的许多问题。

SVN是一个非常通用的软件系统, 它常被用来管理程序 源码, 但是他也可以管理任何类似的文件, 如文本、视频, 图片等等。

SVN相关站点:

1. Subversion官网:
2. <http://subversion.tigris.org/>
3. <http://subversion.apache.org/>
- 4.
5. svn客户端: <http://tortoisesvn.net/>
6. svn中文网站: <http://www.iusesvn.com/>
7. 中文常见问题解答: FAQ:<http://subversion.apache.org/faq.zh.html>
8. 官方手册: <http://svnbook.red-bean.com/>

二、SVN与git的区别

1.SVN集中式版本控制系统

SVN版本控制系统是集中式的数据管理, 存在一个中央版本库, 所有开发人员本地开发所使用的代码都是来自于这个版本库, 提交了代码也都必须提交到这个中央版本库。

SVN版本控制系统工作流程如下:

- 1.在中央库上创建或从主干复制一个分支。
- 2. 从中央库check out下这个分支的代码。
- 3. 增加自己的代码文件, 修改现存的代码或删除代码文件。
- 4. Commit代码。假设有有人在刚刚的分支上提交了代码, 你就会被提示代码过期。
- 你得先up你的代码后再提交。Up代码的时候如果出现冲突, 需要解决冲突后再进行提交

缺点:

- 当无法连接到中央版本库的环境下, 你无法提交代码, 将代码加入版本控制;

- 你无法查看代码的历史版本以及版本的变化过程。提交到版本控制系统中的代码我们都默认通过自测可运行的，如果某个模块的代码比较复杂，不能短时间内实现可测试的功能，那么你需要等很长的时间才能提交自己的代码，由于代码库集中管理，因为，需要对中央版本库的存储做备份。这点分布式的版本控制系统要好一些。Svn的备份要备份所有代码数据以及所有更改的版本记录。

2.git分布式版本控制

Git是由Linux开发的，所以很自然的git个Linux文件系统结合的比较紧密，以至于在Windows上你必须使用cygwin才能使其完美工作。

那git凭什么叫做分布式的版本控制系统呢？还是从模式讲起。

Git中没有了中央版本库的说法了，但是为了开发小组的代码共享，从某种程度上说本地的仓库和远程的仓库在身份上是等价的，没有主从之分。

如果你的项目是闭源项目，或者你习惯于以往的集中式的管理模式的的话，那么在git下你也可以像SVN那么工作，知识流程中可能增加一些步骤。

- 1.你本地创建一个git库，并将其add到远程git库中
- 2. 你在本地添加或者删除文件，然后commit。当然commit操作都是提交到本地的git库中了（其实是提交到git目录下的object目录中去了）
- 3.将本地git库的分支push到远程git库的分支，如果这个时候远程git库中已经有别人push过，那么远程git库将不允许你push，这个时候你需要先pull，然后如果有冲突，处理好冲突，commit到本地git库后，再push到远程git库。

3.运维人员掌握版本管理

对于版本管理系统，运维人员需要掌握的技术点：

- 1.安装，部署，维护，故障。
- 2.简单的使用，很多公司都是由开发来管理，包括建立账户和添加删除账号。
- 3.对于版本控制系统，运维人员相当于开发商，开发人员是业主，运维搭建的系统为开发人员服务。

4.Svn服务端运行方式

svn服务常见的运行访问方式有3种：

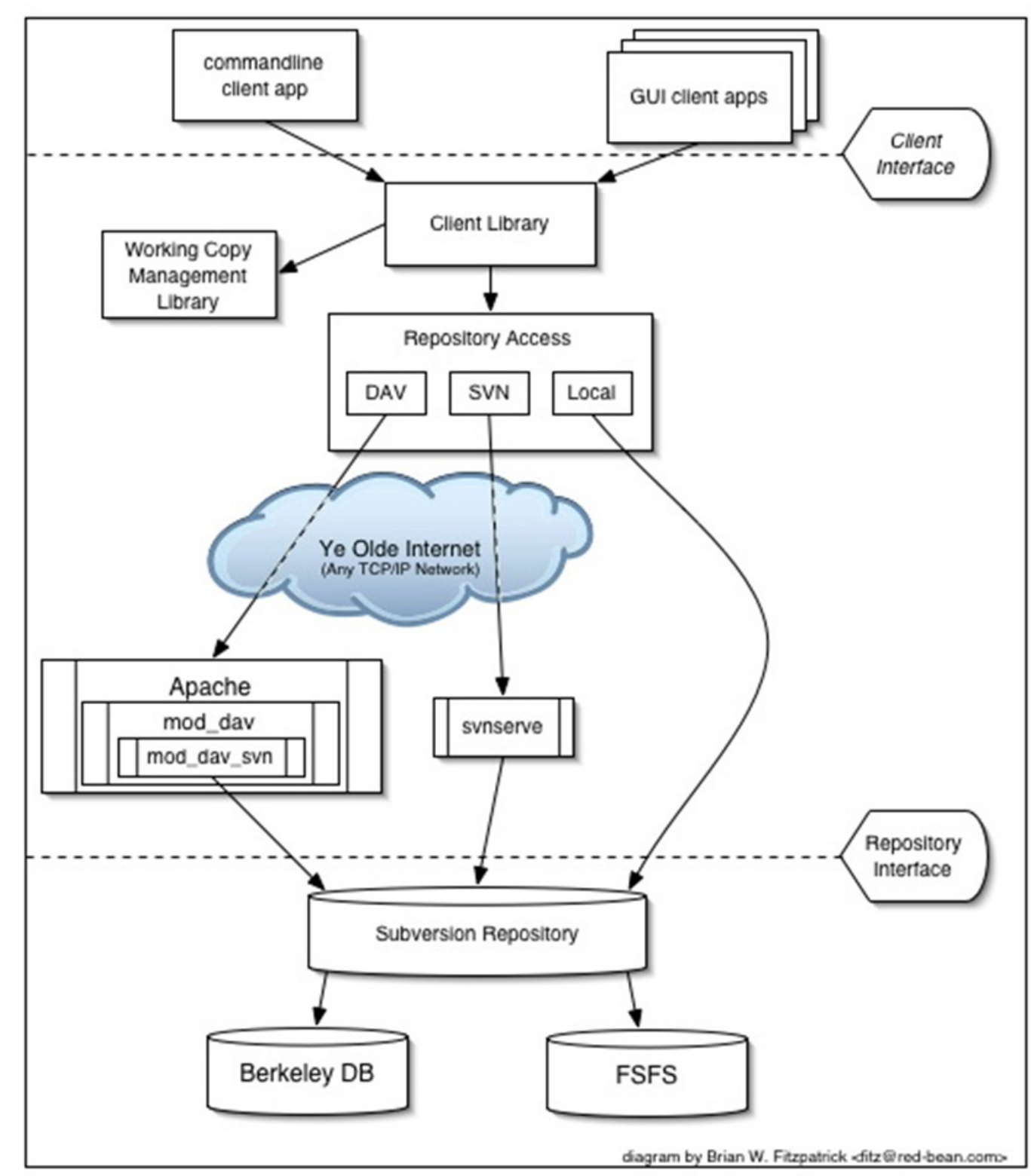
1. 1、独立服务器
2. 访问地址如：`svn://svn.etiantian.org/sadoc`；
- 3.
4. 2、借助apache等http服务
5. 访问地址如：`http://svn.etiantian.org/sadoc`；
6. a.单独安装apache+svn（不要用）。
7. b.CSVN(apache+svn)是一个单独的整合的软件，带web界面管理的SVN软件
- 8.
9. 3、本地直接访问（例如：`file:///root/svndata/sadoc`）

5.SVN客户端访问方式

SVN客户端可以通过多种方式访问服务器，例如：本地磁盘访问，或者各种各样不同的网络协议访问，但一个版本库地址永远都是URL，URL反映了访问方法。

访问方式	说明
file://	直接通过本地磁盘或者网络磁盘访问版本库
http://	通过WebDAV协议访问支持Subersion的Apache服务器
https://	与http://相似，但是使用SSL加密访问
svn://	通过TCP/IP自定义协议访问svnserver服务器
svn+ssh://	通过认证并加密的TCP/IP自定义协议访问svnserver服务器

三、SVN版本系统逻辑架构原理图



四、安装配置SVN服务

1. 环境准备

1. `[root@tomcat ~]# cat /etc/redhat-release`
2. `CentOS release 6.5(Final)`
3. `[root@tomcat ~]# uname -r`
4. `2.6.32-431.el6.x86_64`
5. `[root@tomcat ~]# /etc/init.d/iptables stop`

6. [root@tomcat ~]# wget -O/etc/yum.repos.d/CentOS-Base.repo http://mirrors.aliyun.com/repo/Centos-6.repo
7. [root@tomcat ~]# wget -O/etc/yum.repos.d/epel.repo http://mirrors.aliyun.com/repo/epel-6.repo
8. [root@tomcat ~]# yum clean all && yum makecache

2. 安装SVN

1. [root@tomcat ~]# rpm -qa subversion
2. subversion-1.6.11-15.el6_7.x86_64

如果没有上面的结果，则需要执行如下命令安装

1. [root@tomcat ~]# yum install subversion-y
2. [root@tomcat ~]# rpm -qa subversion
3. subversion-1.6.11-15.el6_7.x86_64
4. 补充：yum安装rpm包安装后本地不清除的方法
5. [root@tomcat ~]# grep keepcache/etc/yum.conf
6. keepcache=1

3. 配置并启动SVN

建立SVN版本库数据存储根目录（svndata）及用户，密码权限目录(svnpasswd)

1. [root@tomcat ~]# mkdir -p /home/svndata数据存储根目录
2. [root@tomcat ~]# mkdir -p /home/svnpasswd 用户密码权限目录

启动SVN服务指定访问的SVN根目录

1. [root@tomcat ~]# svnserve -d -r /home/svndata/

可以查看svnserve命令帮助，了解相关启动参数：

1. [root@tomcat ~]# svnserve --help
2. usage: svnserve [-d | -i | -t | -X][options]
- 3.
4. Valid options:
5. -d [--daemon]: daemon mode
6. -i [--inetd]: inetd mode
7. -t [--tunnel]: tunnel mode
8. -X [--listen-once]: listen-once mode (useful for debugging)
9. -r [--root] ARG : root of directory to serve
10. -R [--read-only]: force read only, overriding repository config file
11. --config-file ARG : read configuration from file ARG
12. --listen-port ARG : listen port
13. [mode: daemon, listen-once]
14. --listen-host ARG : listen hostname or IP address
15. [mode: daemon, listen-once]
16. -T [--threads]:use threads instead of fork [mode: daemon]
17. --foreground : run in foreground (useful for debugging)
18. [mode: daemon]
19. --log-file ARG : svnserve log file
20. --pid-file ARG : write server process ID to file ARG
21. [mode: daemon, listen-once]
22. --tunnel-user ARG : tunnel username (default is current uid's name)
23. [mode: tunnel]
24. -h [--help] : display this help
25. --version : show program version information

查看SVN进程

1. [root@tomcat ~]# ps -ef |grep svn
2. root 30081000:32?00:00:00 svnserve -d -r/root/svndata/
3. root 30102919000:32 pts/000:00:00 grep svn

检查SVN端口

1. [root@tomcat ~]# lsof -i :3690
2. COMMAND PID USER FD TYPE DEVICE SIZE/OFF NODE NAME
3. svnserve 3008 root 3uIPv4536170t0 TCP*:svn (LISTEN)
4. [root@tomcat ~]# netstat -lntup|grep svn
5. [root@tomcat ~]# lsof -i tcp:3690

3.建立项目版本库

创建一个新的SVN项目test其实类似test这样的项目可以创建多个，每个项目对应不同的代码，这里只是以创建一个项目为例。

1. [root@tomcat ~]# svnadmin create /home/svndata/test
2. [root@tomcat ~]# tree/home/svndata/test/
3. /root/svndata/sadoc/
4. |-- README.txt
5. |-- conf
6. |-- authz
7. |-- passwd
8. |-- svnserve.conf
9. |-- db
10. | |-- current
11. | |-- format
12. | |-- fs-type
13. | |-- fsfs.conf
14. | |-- min-unpacked-rev
15. | |-- rep-cache.db
16. | |-- revprops
17. | | `-- 0
18. || `-- 0
19. | |-- revs
20. | | `-- 0
21. || `-- 0
22. | |-- transactions
23. | |-- txn-current
24. | |-- txn-current-lock
25. | |-- txn-protorevs
26. | |-- uuid
27. | `-- write-lock
28. |-- format
29. |-- hooks
30. |-- post-commit.tmpl
31. |-- post-lock.tmpl
32. |-- post-revprop-change.tmpl
33. |-- post-unlock.tmpl
34. |-- pre-commit.tmpl
35. |-- pre-lock.tmpl
36. |-- pre-revprop-change.tmpl
37. |-- pre-unlock.tmpl
38. |-- start-commit.tmpl
39. `-- locks
40. |-- db-logs.lock
41. `-- db.lock

42. 10 directories, 28 files

4.调整SVN配置文件及权限

```

1. [root@tomcat ~]# cd /root/svndata/test/conf/
2. [root@tomcat conf]# ll
3. total 12
4. -rw-r--r-- 1 root root 1080Feb2316:11 authz
5. -rw-r--r-- 1 root root 309Feb2316:11 passwd
6. -rw-r--r-- 1 root root 2279Feb2316:11 svnserve.conf
7. [root@tomcat conf]# cp svnserve.conf svnserve.conf,ori 操作前备份
8. [root@tomcat conf]# vim svnserve.conf
9. anon-access = none
10. auth-access = write
11. password-db = /home/svnpasswd/passwd
12. authz-db = /home/svnpasswd/authz #为了方便统一管理我们把用户的权限和密码文件存放在一个目录下
13.
14. 配置解析:
15. anon-access=定义非授权用户(匿名用户)的访问权限, 有三种方式: none 、 read 、 write , 设置为 none 限制访问, read 为只读, write 为具有读写权限, 默认为 read。
16.
17. auth-access=定义授权用户的访问权限, 有三种方式: none 、 read 、 write , 设置为none 限制访问, read 为只读, write 为具有读写权限, 默认为 write 。
18.
19. password-db=定义保存用户名和密码的文件名称, 这里为 passwd , 和该文件位于同一目录。
20.
21. authz-db=定义保存授权信息的文件名称, 这里为 authz , 和该文件位于同一目录。

```

把密码文件模板拷贝到相关目录

```

1. [root@tomcat conf]# cp passwd authz /home/svnpasswd/
2. [root@tomcat conf]# ll /home/svnpasswd/
3. total 8
4. -rw-r--r-- 1 root root 1080Mar3100:51 authz
5. -rw-r--r-- 1 root root 309Mar3100:51 passwd

```

因为SVN默认都是明文密码, 为了安全起见加上权限

```

1. [root@tomcat conf]# cd /home/svnpasswd/
2. [root@tomcat svnpasswd]# chmod 700*
3. [root@tomcat svnpasswd]# ll
4. total 8
5. -rwx----- 1 root root 1080Mar3100:51 authz
6. -rwx----- 1 root root 309Mar3100:51 passwd

```

5.创建SVN用户及设置权限

```

1. [1.创建用户]
2. [root@tomcat svnpasswd]# cat passwd
3. ### This file is an example password file for svnserve.
4. ### Its format is similar to that of svnserve.conf. As shown in the
5. ### example below it contains one section labelled [users].
6. ### The name and password for each user follow, one account per line.
7.
8. [users]
9. cyh =123456
10. test =123..
11. itcast =123456

```

```

12.
13. [2.配置权限]
14. [root@tomcat svnpasswd]# cat authz
15. ### This file is an example authorization file for svnserve.
16. ### Its format is identical to that of mod_authz_svn authorization
17. ### files.
18. ### As shown below each section defines authorizations for the path and
19. ### (optional) repository specified by the section name.
20. ### The authorizations follow. An authorization line can refer to:
21. ### - a single user,
22. ### - a group of users defined in a special [groups] section,
23. ### - an alias defined in a special [aliases] section,
24. ### - all authenticated users, using the '$authenticated' token,
25. ### - only anonymous users, using the '$anonymous' token,
26. ### - anyone, using the '*' wildcard.
27. ###
28. ### A match can be inverted by prefixing the rule with '~'. Rules can
29. ### grant read ('r') access, read-write ('rw') access, or no access
30. ### ('').
31.
32. [aliases]
33. # joe = /C=XZ/ST=Dessert/L=Snake City/O=Snake Oil, Ltd./OU=Research Institute/CN=Joe Average
34.
35. [groups]
36. # harry_and_sally = harry,sally
37. # harry_sally_and_joe = harry,sally,&joe
38. [test:/]
39. cyh = rw
40. itcast = rw
41. test = rw

```

重启SVN命令

```
1. [root@tomcat svnpasswd]# svnserve -d -r/home/svndata/
```

客户端的操作此处就不介绍了，我们可以通过在服务器上使用svn
co svn://127.0.0.1/test 进行测试

一、什么是SVN (subversion) ?

SVN是近年来崛起的非常优秀的版本管理工具，与CVS管理工具一样，SVN是一个固态的跨平台的开源的版本控制系统。SVN版本管理工具管理者随时间改变的各种数据。这些数据放置在一个中央资料档案库 (repository) 中，这个档案库很像一个普通的文件服务器或者FTP服务器，但是，与其他服务器不同的是，SVN会备份并记录每个文件每一次的修改更新变动。这样我们就可以把任意一个时间点的档案恢复到想要的某一个旧的版本，当然也可以直接浏览指定的更新历史记录。

为什么会有SVN这样一个项目？

官方解释：为了接管CVS的用户基础，确切的说，我们写了一个新的版本控制系统，它和CVS很相似，但是它修正了以前CVS所没有解决的许多问题。

SVN是一个非常通用的软件系统，它常被用来管理程序 源码，但是他也可以管理任何类似的文件，如文本、视频，图片等等。

SVN相关站点：

1. [Subversion官网](#):
2. <http://subversion.tigris.org/>
3. <http://subversion.apache.org/>
- 4.
5. [svn客户端](#): <http://tortoisesvn.net/>
6. [svn中文网站](#): <http://www.iusesvn.com/>
7. [中文常见问题解答](#): [FAQ:http://subversion.apache.org/faq.zh.html](http://subversion.apache.org/faq.zh.html)
8. [官方手册](#): <http://svnbook.red-bean.com/>

二、SVN与git的区别

1.SVN集中式版本控制系统

SVN版本控制系统是集中式的数据管理，存在一个中央版本库，所有开发人员本地开发所使用的代码都是来自于这个版本库，提交了代码也都必须提交到这个中央版本库。

SVN版本控制系统工作流程如下：

- 1.在中央库上创建或从主干复制一个分支。
- 2. 从中央库check out下这个分支的代码。
- 3. 增加自己的代码文件，修改现存的代码或删除代码文件。
- 4. Commit代码。假设有人在刚刚的分支上提交了代码，你就会被提示代码过期。
- 你得先up你的代码后再提交。Up代码的时候如果出现冲突，需要解决冲突后再进行提交

缺点：

- 当无法连接到中央版本库的环境下，你无法提交代码，将代码加入版本控制；
- 你无法查看代码的历史版本以及版本的变化过程。提交到版本控制系统中的代码我们都默认通过自测可运行的，如果某个模块的代码比较复杂，不能短时间内实现可测试的功能，那么你需要等很长的时间才能提交自己的代码，由于代码库集中管理，因为，需要对中央版本库的存储做备份。这点分布式的版本控制系统要好一些。Svn的备份要备份所有代码数据以及所有更改的版本记录。

2.git分布式版本控制

Git是由Linux开发的，所以很自然的git个Linux文件系统结合的比较紧密，以至于在Windows上你必须使用cygwin才能使其完美工作。

那git凭什么叫做分布式的版本控制系统呢？还是从模式讲起。

Git中没有了中央版本库的说法了，但是为了开发小组的代码共享，从某种程度上说本地的仓库和远程的仓库在身份上是等价的，没有主从之分。

如果你的项目是闭源项目，或者你习惯于以往的集中式的管理模式的的话，那么在git下你也可以像SVN那么工作，知识流程中可能增加一些步骤。

- 1.你本地创建一个git库，并将其add到远程git库中
- 2. 你在本地添加或者删除文件，然后commit。当然commit操作都是提交到本地的git库中了（其实是提交到git目录下的object目录中去了）

- 3.将本地git库的分支push到远程git库的分支，如果这个时候远程git库中已经有别人push过，那么远程git库将不允许你push，这个时候你需要先pull，然后如果有冲突，处理好冲突，commit到本地git库后，再push到远程git库。

3.运维人员掌握版本管理

对于版本管理系统，运维人员需要掌握的技术点：

- 1.安装，部署，维护，故障。
- 2.简单的使用，很多公司都是由开发来管理，包括建立账户和添加删除账号。
- 3.对于版本控制系统，运维人员相当于开发商，开发人员是业主，运维搭建的系统为开发人员服务。

4.Svn服务端运行方式

svn服务常见的运行访问方式有3种：

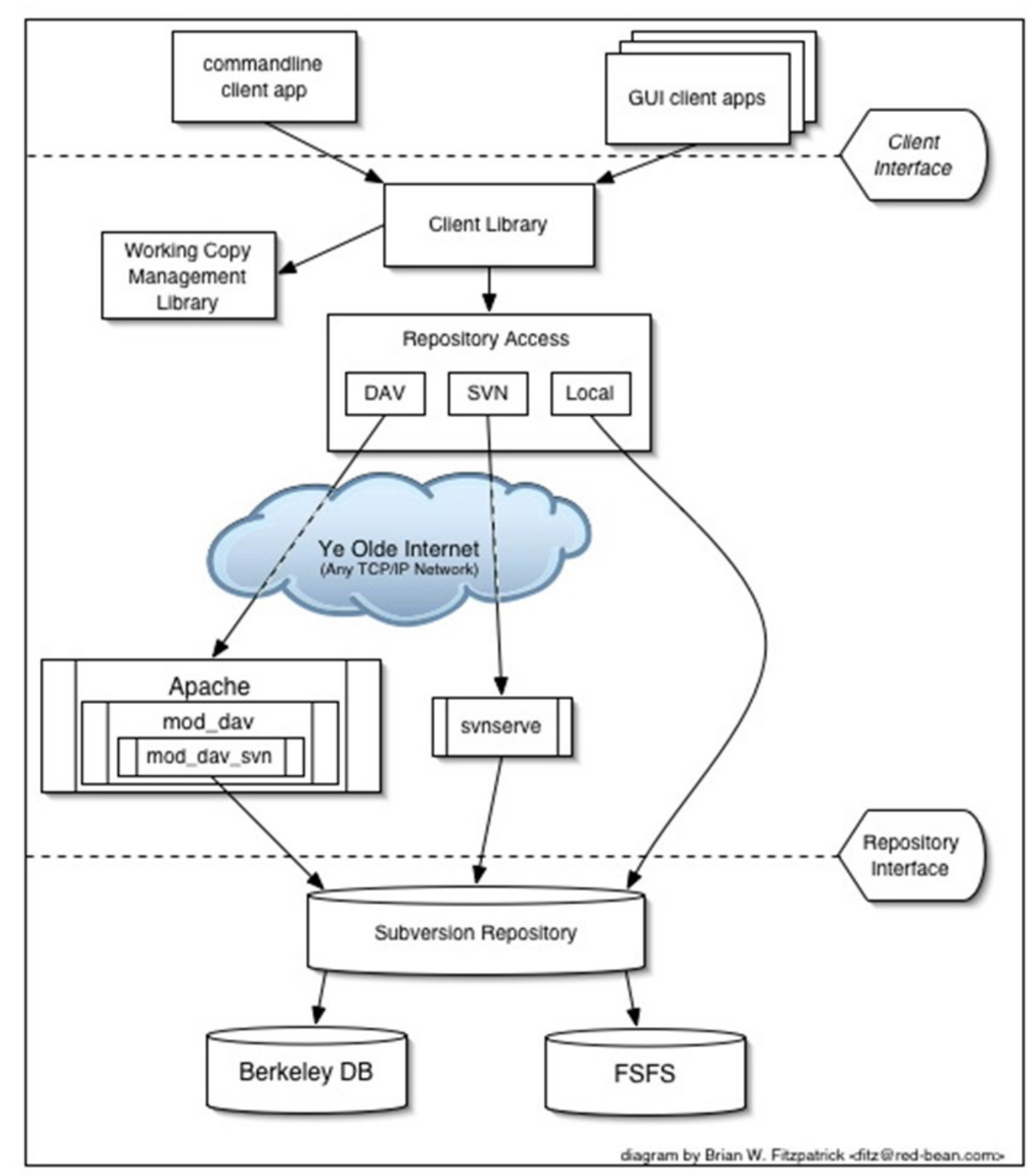
- 1、独立服务器
访问地址如：`svn://svn.etiantian.org/sadoc`；
- 2、借助apache等http服务
访问地址如：`http://svn.etiantian.org/sadoc`；
a.单独安装apache+svn（不要用）。
b.CSVN(apache+svn)是一个单独的整合的软件，带web界面管理的SVN软件
- 3、本地直接访问（例如：`file:///root/svndata/sadoc`）

5.SVN客户端访问方式

SVN客户端可以通过多种方式访问服务器，例如：本地磁盘访问，或者各种各样不同的网络协议访问，但一个版本库地址永远都是URL，URL反映了访问方法。

访问方式	说明
file://	直接通过本地磁盘或者网络磁盘访问版本库
http://	通过WebDAV协议访问支持Subersion的Apache服务器
https://	与http://相似，但是使用SSL加密访问
svn://	通过TCP/IP自定义协议访问svnserver服务器
svn+ssh://	通过认证并加密的TCP/IP自定义协议访问svnserver服务器

三、SVN版本系统逻辑架构原理图



四、安装配置SVN服务

1. 环境准备

1. `[root@tomcat ~]# cat /etc/redhat-release`
2. `CentOS release 6.5(Final)`
3. `[root@tomcat ~]# uname -r`
4. `2.6.32-431.el6.x86_64`
5. `[root@tomcat ~]# /etc/init.d/iptables stop`

6. [root@tomcat ~]# wget -O/etc/yum.repos.d/CentOS-Base.repo http://mirrors.aliyun.com/repo/Centos-6.repo
7. [root@tomcat ~]# wget -O/etc/yum.repos.d/epel.repo http://mirrors.aliyun.com/repo/epel-6.repo
8. [root@tomcat ~]# yum clean all && yum makecache

2. 安装SVN

1. [root@tomcat ~]# rpm -qa subversion
2. subversion-1.6.11-15.el6_7.x86_64

如果没有上面的结果，则需要执行如下命令安装

1. [root@tomcat ~]# yum install subversion-y
2. [root@tomcat ~]# rpm -qa subversion
3. subversion-1.6.11-15.el6_7.x86_64
4. 补充：yum安装rpm包安装后本地不清除的方法
5. [root@tomcat ~]# grep keepcache/etc/yum.conf
6. keepcache=1

3. 配置并启动SVN

建立SVN版本库数据存储根目录（svndata）及用户，密码权限目录(svnpasswd)

1. [root@tomcat ~]# mkdir -p /home/svndata数据存储根目录
2. [root@tomcat ~]# mkdir -p /home/svnpasswd 用户密码权限目录

启动SVN服务指定访问的SVN根目录

1. [root@tomcat ~]# svnserve -d -r /home/svndata/

可以查看svnserve命令帮助，了解相关启动参数：

1. [root@tomcat ~]# svnserve --help
2. usage: svnserve [-d | -i | -t | -X][options]
- 3.
4. Valid options:
5. -d [--daemon]: daemon mode
6. -i [--inetd]: inetd mode
7. -t [--tunnel]: tunnel mode
8. -X [--listen-once]: listen-once mode (useful for debugging)
9. -r [--root] ARG : root of directory to serve
10. -R [--read-only]: force read only, overriding repository config file
11. --config-file ARG : read configuration from file ARG
12. --listen-port ARG : listen port
13. [mode: daemon, listen-once]
14. --listen-host ARG : listen hostname or IP address
15. [mode: daemon, listen-once]
16. -T [--threads]:use threads instead of fork [mode: daemon]
17. --foreground : run in foreground (useful for debugging)
18. [mode: daemon]
19. --log-file ARG : svnserve log file
20. --pid-file ARG : write server process ID to file ARG
21. [mode: daemon, listen-once]
22. --tunnel-user ARG : tunnel username (default is current uid's name)
23. [mode: tunnel]
24. -h [--help] : display this help
25. --version : show program version information

查看SVN进程

1. [root@tomcat ~]# ps -ef |grep svn
2. root 30081000:32?00:00:00 svnserve -d -r/root/svndata/
3. root 30102919000:32 pts/000:00:00 grep svn

检查SVN端口

1. [root@tomcat ~]# lsof -i :3690
2. COMMAND PID USER FD TYPE DEVICE SIZE/OFF NODE NAME
3. svnserve 3008 root 3uIPv4536170t0 TCP*:svn (LISTEN)
4. [root@tomcat ~]# netstat -lntup|grep svn
5. [root@tomcat ~]# lsof -i tcp:3690

3.建立项目版本库

创建一个新的SVN项目test其实类似test这样的项目可以创建多个，每个项目对应不同的代码，这里只是以创建一个项目为例。

1. [root@tomcat ~]# svnadmin create /home/svndata/test
2. [root@tomcat ~]# tree/home/svndata/test/
3. /root/svndata/sadoc/
4. |-- README.txt
5. |-- conf
6. |-- authz
7. |-- passwd
8. |-- svnserve.conf
9. |-- db
10. | |-- current
11. | |-- format
12. | |-- fs-type
13. | |-- fsfs.conf
14. | |-- min-unpacked-rev
15. | |-- rep-cache.db
16. | |-- revprops
17. | | `-- 0
18. || `-- 0
19. | |-- revs
20. | | `-- 0
21. || `-- 0
22. | |-- transactions
23. | |-- txn-current
24. | |-- txn-current-lock
25. | |-- txn-protorevs
26. | |-- uuid
27. | `-- write-lock
28. |-- format
29. |-- hooks
30. |-- post-commit.tmpl
31. |-- post-lock.tmpl
32. |-- post-revprop-change.tmpl
33. |-- post-unlock.tmpl
34. |-- pre-commit.tmpl
35. |-- pre-lock.tmpl
36. |-- pre-revprop-change.tmpl
37. |-- pre-unlock.tmpl
38. |-- start-commit.tmpl
39. `-- locks
40. |-- db-logs.lock
41. `-- db.lock

42. 10 directories, 28 files

4.调整SVN配置文件及权限

```

1. [root@tomcat ~]# cd /root/svndata/test/conf/
2. [root@tomcat conf]# ll
3. total 12
4. -rw-r--r-- 1 root root 1080Feb2316:11 authz
5. -rw-r--r-- 1 root root 309Feb2316:11 passwd
6. -rw-r--r-- 1 root root 2279Feb2316:11 svnserve.conf
7. [root@tomcat conf]# cp svnserve.conf svnserve.conf,ori 操作前备份
8. [root@tomcat conf]# vim svnserve.conf
9. anon-access = none
10. auth-access = write
11. password-db =/home/svnpasswd/passwd
12. authz-db =/home/svnpasswd/authz #为了方便统一管理我们把用户的权限和密码文件存放在一个目录下
13.
14. 配置解析:
15. anon-access=定义非授权用户(匿名用户)的访问权限, 有三种方式: none 、 read 、 write , 设置为 none 限制访问, read 为只读, write 为具有读写权限, 默认为 read。
16.
17. auth-access=定义授权用户的访问权限, 有三种方式: none 、 read 、 write , 设置为none 限制访问, read 为只读, write 为具有读写权限, 默认为 write 。
18.
19. password-db=定义保存用户名和密码的文件名称, 这里为 passwd , 和该文件位于同一目录。
20.
21. authz-db=定义保存授权信息的文件名称, 这里为 authz , 和该文件位于同一目录。

```

把密码文件模板拷贝到相关目录

```

1. [root@tomcat conf]# cp passwd authz /home/svnpasswd/
2. [root@tomcat conf]# ll /home/svnpasswd/
3. total 8
4. -rw-r--r-- 1 root root 1080Mar3100:51 authz
5. -rw-r--r-- 1 root root 309Mar3100:51 passwd

```

因为SVN默认都是明文密码, 为了安全起见加上权限

```

1. [root@tomcat conf]# cd /home/svnpasswd/
2. [root@tomcat svnpasswd]# chmod 700*
3. [root@tomcat svnpasswd]# ll
4. total 8
5. -rwx----- 1 root root 1080Mar3100:51 authz
6. -rwx----- 1 root root 309Mar3100:51 passwd

```

5.创建SVN用户及设置权限

```

1. [1.创建用户]
2. [root@tomcat svnpasswd]# cat passwd
3. ### This file is an example password file for svnserve.
4. ### Its format is similar to that of svnserve.conf. As shown in the
5. ### example below it contains one section labelled [users].
6. ### The name and password for each user follow, one account per line.
7.
8. [users]
9. cyh =123456
10. test =123..
11. itcast =123456

```

```
12.
13. [2.配置权限]
14. [root@tomcat svnpasswd]# cat authz
15. ### This file is an example authorization file for svnserve.
16. ### Its format is identical to that of mod_authz_svn authorization
17. ### files.
18. ### As shown below each section defines authorizations for the path and
19. ### (optional) repository specified by the section name.
20. ### The authorizations follow. An authorization line can refer to:
21. ### - a single user,
22. ### - a group of users defined in a special [groups] section,
23. ### - an alias defined in a special [aliases] section,
24. ### - all authenticated users, using the '$authenticated' token,
25. ### - only anonymous users, using the '$anonymous' token,
26. ### - anyone, using the '*' wildcard.
27. ###
28. ### A match can be inverted by prefixing the rule with '~'. Rules can
29. ### grant read ('r') access, read-write ('rw') access, or no access
30. ### ('').
31.
32. [aliases]
33. # joe = /C=XZ/ST=Dessert/L=Snake City/O=Snake Oil, Ltd./OU=Research Institute/CN=Joe Average
34.
35. [groups]
36. # harry_and_sally = harry,sally
37. # harry_sally_and_joe = harry,sally,&joe
38. [test:/]
39. cyh = rw
40. itcast = rw
41. test = rw
```

重启SVN命令

```
1. [root@tomcat svnpasswd]# svnserve -d -r/home/svndata/
```

客户端的操作此处就不介绍了，我们可以通过在服务器上使用svn
co svn://127.0.0.1/test 进行测试

赞0

如无特殊说明，文章均为本站原创，转载请注明出处

- 转载请注明来源：SVN服务实战应用指南
- 本文永久链接地址：<https://www.nmtui.com/clsn/lx243.html>

该文章由 惨绿少年 发布



惨绿少年Linux www.nmtui.com