

Trac 开发社区网站的搭建

以下以 debian 发行版为例，说明 Trac 开发网站的搭建过程。服务器的跟目录在 /var/www，另外以 web-devel 为示例，阐述搭建新项目的步骤

| | |
|-------------------------------------|----|
| 安装相应软件包..... | 1 |
| 搭建系统与配置..... | 2 |
| 配置 trac.wsgi..... | 2 |
| 配置 gitweb..... | 2 |
| 配置 Apache..... | 3 |
| Trac 插件与配置..... | 4 |
| GitPlugin..... | 5 |
| TracGitHack..... | 5 |
| Account Manager Plugin..... | 6 |
| Spam Filtering..... | 7 |
| TracDownloader..... | 7 |
| TracDiscussion..... | 7 |
| 搭建 git 服务器..... | 8 |
| gitosis..... | 8 |
| 创建新项目..... | 10 |
| 创建 git 仓库..... | 10 |
| trac-admin..... | 11 |
| 配置/创建对应项目的数据库（以 PostgreSQL 为例）..... | 11 |
| 创建 Trac 中的项目..... | 11 |
| trac.ini..... | 12 |
| web 自定义配置..... | 12 |
| 资源（例子）..... | 12 |

安装相应软件包

- Python 2.5
 - setuptools
 - docutils > 0.3.9
 - pygments
 - pytz
- Apache > 2.2
 - [Apache mod_python > 3.1.3]
 - Apache mod_wsgi > 2.3
- PostgreSQL

- psychopg2 (python 绑定)
- git-core > 1.5.2
- gitweb
- subversion
 - SWIG (python-subversion)
 - libapache2-svn
- Trac
 - Genshi
 - 相关插件

搭建系统与配置

服务器的根目录为/var/www/

```
mkdir /var/www/code
cd /var/www/code
mkdir -p files git trac/projects svn
cd ..
chown -R www-data:www-data code
```

配置 trac.wsgi

```
#保存下述内容为/var/www/code/trac/trac.wsgi
import sys
sys.stdout = sys.stderr

#python-eggs 不能在 TRAC_ENV_PARENT_DIR 内，否则会提示python-eggs
#下没有 VERSION 文件(当作项目搜寻了)
import os
os.environ['PYTHON_EGG_CACHE'] = '/var/www/code/trac/.python-eggs'

import trac.web.main
application = trac.web.main.dispatch_request
```

配置 cgkit

1. 文件安装至/var/www/code/cgkit
2. 新建目录/var/cache/cgkit，修改所有者为 www-data:www-data

3. 修改 cgitr :

```
projectroot=/var/www/code/git
projects_list=/var/www/code/gitosis_u/gitosis/projects.list
git_base_url_list=git://172.16.2.41 http://172.16.2.41/http_git ssh://git@
172.16.2.41
```

配置 gitweb

收集 gitweb 相关资源，保存到/var/www/code/gitweb 文件夹下：

- /usr/lib/cgi-bin/: gitweb.cgi
- /usr/share/gitweb/:
 - git-favicon.png
 - git-logo.png
 - gitweb.css
- /etc/gitweb.conf

修改 gitweb.conf：

```
$projectroot = "/var/www/code/git";
$projects_list = "/path/to/git_home/gitosis/projects.list";
@git_base_url_list = ('git://172.16.2.41');
$stylesheet = "files/gitweb.css";
$logo = "files/git-logo.png";
$favicon = "files/git-favicon.png";
```

配置 Apache

配置 apache

```
#use mod_wsgi daemon mode to avoid to many database connections(PostgreSQL,
Mysql?)
#WSGIDaemonProcess code_sites user=trac group=trac processes=3 threads=25
WSGIDaemonProcess code_sites processes=3 threads=25
WSGIScriptAlias /code /var/www/code/trac/trac.wsgi

<Directory /var/www/code>
    WSGIProcessGroup code_sites
    WSGIApplicationGroup %{GLOBAL}
    SetEnv trac.env_parent_dir /var/www/code/trac/projects
    Order deny,allow
    Allow from all
</Directory>
```

```
Alias /files/cgit.css /var/www/code/cgit/cgit.css
Alias /files/cgit.png /var/www/code/cgit/cgit.png

ScriptAlias /cgit /var/www/code/cgit/cgit.cgi
<Directory "/var/www/htdocs/cgit/">
    SetEnv CGIT_CONFIG /var/www/code/cgit/cgitrc
    AllowOverride None
    Options FollowSymLinks -SymLinksIfOwnerMatch Indexes +MultiViews
    Order allow,deny
    Allow from all
</Directory>

Alias /files/gitweb.css /var/www/code/gitweb/gitweb.css
Alias /files/git-logo.png /var/www/code/gitweb/git-logo.png
Alias /files/git-favicon.png /var/www/code/gitweb/git-favicon.png

ScriptAlias /git /var/www/code/gitweb/gitweb.cgi
<Directory "/var/www/code/gitweb">
    SetEnv GITWEB_CONFIG /var/www/code/gitweb/gitweb.conf
    Options FollowSymLinks -SymLinksIfOwnerMatch Indexes +MultiViews
    Order allow,deny
    Allow from all
    AllowOverride None
</Directory>

Alias /files /var/www/code/files
<Location /files>
    SetHandler None
</Location>

<Location /svn>
    DAV svn
    SVNParentPath /var/www/code/svn
    AuthType Basic
    AuthName "Subversion Repositories"
    AuthUserFile /var/www/code/trac/trac.htpasswd
    Require valid-user
</Location>
```

Trac 插件与配置

资源：

- 插件列表：<http://trac.edgewall.org/wiki/PluginList> (<http://trac-hacks.org/>)
- 插件构架：<http://trac.edgewall.org/wiki/TracDev/ComponentArchitecture>
- Genshi 模板：<http://genshi.edgewall.org/wiki/GenshiTutorial>
- Babel 国际化：<http://babel.edgewall.org/>
- Bitten (Continuous Integration [trac 插件]) <http://bitten.edgewall.org/>
- 界面自定义：<http://trac.edgewall.org/wiki/TracInterfaceCustomization>

全局配置：

```
[inherit]
file = .../path/to/trac_global.ini
```

指定默认编码

```
[trac]
default_charset = utf-8
```

配置 LOGO

```
[header_logo]
src = site/my_logo.gif
alt = My Project
width = 300
height = 100
```

限制附件大小，单位：字节。默认值：256K

```
[attachment]
max_size= 262144
render_unsafe_content = false
```

配置日志，在项目的 log 文件夹下

```
[logging]
log_file = trac.log
log_level = INFO
log_type = file
```

设置通知相关配置（邮件服务器）

```
[notification]
smtp_enabled = true
smtp_from =
smtp_from_name =
smtp_server =
smtp_user =
smtp_password =
```

GitPlugin

网址：<http://trac-hacks.org/wiki/GitPlugin>

Bug：<http://trac-hacks.org/ticket/3654>（版本管理仓库为空时出现断言错误）

安装：下载后解包

```
python setup.py build
python setup.py install
```

配置：

```
[components]
tracext.git.* = enabled

[git]
## let Trac cache meta-data via CachedRepository wrapper; default: false
cached_repository = true #仓库不为空时
## disable automatic garbage collection for in-memory commit-tree cache; default:
false
persistent_cache = true

## length revision sha-sums should be tried to be abbreviated to (must be >= 4 and
<= 40); default: 7
shortrev_len = 6

## executable file name (optionally with path) of git binary; default: 'git'
git_bin = /usr/bin/git
```

TracGitHack

```
[components]
gitweb.* = enabled

#每个项目单独设置
[gitweb]
url =
```

Account Manager Plugin

网址：<http://trac-hacks.org/wiki/AccountManagerPlugin>

配置：注意，以下配置是把用户名和密码存储在数据库中，而 Trac 中，数据库是每个项目单独使用。于是不同项目之间不能共享用户名和密码。

```
[components]
acct_mgr.db.SessionStore = enabled
acct_mgr.pwhash.htdigesthashmethod = enabled
acct_mgr.pwhash.htpasswdhashmethod = enabled

acct_mgr.admin.AccountManagerAdminPage = enabled
```

```
acct_mgr.web_ui.AccountModule = enabled

trac.web.auth.LoginModule = disabled
acct_mgr.web_ui.LoginModule = enabled

acct_mgr.web_ui.RegistrationModule = enabled

[account-manager]
password_store = SessionStore
hash_method = HtDigestHashMethod
```

以下配置适合多个项目间共享用户名与密码，注意用户的其他配置信息仍就存储在数据库中，因此不能在不同的项目之间共享。

```
[components]
acct_mgr.htfile.HtPasswdStore = enabled
acct_mgr.admin.AccountManagerAdminPage = enabled
acct_mgr.web_ui.AccountModule = enabled
trac.web.auth.LoginModule = disabled
acct_mgr.web_ui.LoginModule = enabled
acct_mgr.web_ui.RegistrationModule = enabled

[account-manager]
password_store = HtPasswdStore

#the file where user accounts are stored
#the webserver will need write permissions to this file
#and its parent folder
password_file = /var/www/code/trac/trac.htpasswd
```

Spam Filtering

网址：<http://trac.edgewall.org/wiki/SpamFilter>

依赖：dnspython (easy_install dnspython)

备注：匹配 wiki 页 BadContent 中的正则表达式，则内容被过滤。一个示例的 wiki 页 BadContent：<http://trac.edgewall.org/wiki/BadContent>

配置：

```
[components]
tracspamfilter.* = enabled

[spam-filter]
#每个 IP 每小时帖子数
```

```
max_posts_by_ip = 5
```

TracDownloader

网址：<http://trac-hacks.org/wiki/TracDownloaderPlugin>

Bug：<http://trac-hacks.org/ticket/2929> (PostgreSQL 后端执行 trac-admin path/to/project upgrade --no-backup 失败)

说明：目前不可用。依赖旧的 ClearSilver 模板系统

安装：

```
easy_install --find-links http://www.pythonware.com/products/pil/ Imaging
easy_install PyCaptcha #行不通，手工 http 下载全部文件
easy_install http://trac-hacks.org/svn/tracdownloaderplugin/0.11/
```

配置：

```
[components]
tracdownloader.admin.downloaderadminpage = enabled
tracdownloader.web_ui.downloadermodule = enabled
```

TracDiscussion

网址：<http://trac-hacks.org/wiki/DiscussionPlugin>

配置：

```
[components]
# Enable this component for 0.11 branch and 0.10 branch from revision r3518.
tracd DiscussionApi = enabled

tracd DiscussionWebAdmin = enabled
tracd DiscussionCore = enabled
tracd DiscussionInit = enabled
tracd DiscussionSearch = enabled
tracd DiscussionTimeline = enabled
tracd DiscussionWiki = enabled
tracd DiscussionNotifyEmail = enabled

[discussion]
title = Discussion
default_display = tree
forum_sort = id
forum_sort_direction = asc
topic_sort = id
topic_sort_direction = asc
```


搭建 git 服务器

gitorious

参见：

- <http://scie.nti.st/2007/11/14/hosting-git-repositories-the-easy-and-secure-way>
- <http://eagain.net/gitweb/?p=gitorious.git;a=blob;f=README.rst;hb=master>

安装：

```
git clone git://eagain.net/gitorious.git
python setup.py install
```

配置：

1 . 增加用户 git

```
adduser \
--system \
--shell /bin/sh \
--gecos 'git version control' \
--group \
--disabled-password \
--home /home/git \
git
```

2 . 本地创建管理员公钥，并拷贝到服务器上

```
ssh-keygen -t rsa
scp $HOME/.ssh/id_rsa.pub root@172.16.2.41:/home/git/
```

3 . 服务器上初始化 gitorious

```
sudo -H -u git gitorious-init < /home/git/id_rsa.pub
#修正 post-update 权限：使其可执行
chmod 755 /home/git/repositories/gitorious-admin.git/hooks/post-update
```

4 . 本地远程修改 gitorious 服务器的配置

```
git clone git@172.16.2.41:gitorious-admin.git
cd gitorious-admin
```

修改配置文件 gitorious.conf

```
[gitorious]
repositories = /var/www/code/git
#全局禁止访问 git 下的文件
daemon = no
```

```
gitweb = no
#DEBUG, INFO, WARNING, ERROR, CRITICAL
loglevel = DEBUG

[group gitosis-admin]
#值为空格分隔的用户密钥文件的主文件名，带后缀 pub 的公钥文件放到
keydir 文件夹下
#group 没有 writeable 与 readonly 时仅仅为一方便用户分组管理
writable = gitosis-admin
#readonly =
members = cee1

#[group web-devel]
#writable = web-devel
#readonly =
#members = cee1

#[repo web-devel]
#gitweb = yes
#daemon = yes
#owner = cee1
#description =

#[gitweb]
## Where to make gitweb link to as it's "home location".
## NOT YET IMPLEMENTED.
# homelink = http://example.com/
```

完成后提交修改：

```
git commit -a -m "..."  
git push
```

5 . 服务器仓库移动到/var/www/code/git 下

```
#服务器相应修改确保权限没有问题  
chown -R git /var/www/code/git  
chgrp -R git /var/www/code/git  
cd /home/git/  
mv repositories/gitosis-admin.git /var/www/code/git/  
adduser www-data git  
ln -sf /var/www/code/git/gitosis-admin.git/gitosis.conf .gitosis.conf
```

6 . 运行 git-daemon (可以添加到/etc/rc.local 来自动运行)

```
sudo -u git git-daemon --base-path=/var/www/code/git/
```

创建新项目

创建 git 仓库

1. 管理员配置 gitosis.conf , 分配权限

```
[group web-devel]
writable = web-devel
#readonly =
members = ceel somebody

[repo web-devel]
gitweb = yes
daemon = yes
owner = ceel
description =
```

把 somebody 的 ssh 公钥(.pub) , 放到 keydir 目录下 , 之后 commit & push

2. 创建 git 仓库

```
mkdir web-devel
cd web-devel
git init
git remote add origin git@172.16.2.41:web-devel.git

#这步不能省 : do some work, git add and commit files
#不然下面的 push 操作会失败 , 而且空的仓库会触发 Trac 的 GitPlugin 的 Bug

git push origin master:refs/heads/master
```

3. 或从另外一个 SVN 仓库导入

```
git-svn clone svn://172.16.2.41/web-devel
cd web-devel
git remote add origin git@172.16.2.41:web-devel.git
#可以通过 git remote rm origin , 来移除 origin 发布源
git push origin --all
```

trac-admin

配置/创建对应项目的数据库（以 *PostgreSQL* 为例）

每开一个新项目，需要为新项目分配一个独有的数据库。

```
su postgres
createuser -U postgres -E -P trac_postgres_u

#以 postgres 用户，创建数据库 web-devel，所有者为 trac_postgres_u
createdb -U postgres -O trac_postgres_u -E UTF8 web-devel

#可以通过 psql 命令来删除数据库，如果你需要这么作的话
#DROP DATABASE "database_name";
```

之后配置 trac 时字符串为：

```
sqlite:/path/to/database.db
postgres://user:pass@server/database
mysql://user:pass@server/database
```

在本例子中为：

```
postgres://trac_postgres_u:loongson@localhost/web-devel
```

创建 *Trac* 中的项目

切换到/var/www/code/trac，运行（可以指定

--inherit=**full**/path/to/global/）

```
cd /var/www/code/trac
#使用 gitwebhack，不要填写仓库的文件系统路径
trac-admin /var/www/code/trac/web-devel initenv

#可能需要升级，若为 sqlite 数据库，可以去掉--no-backup
#trac-admin /var/www/code/trac/web-devel upgrade --no-backup

#更改所有者，目前为止的操作在 root 用户下进行
chown -R www-data:www-data web-devel
```

trac.ini

之后若有需要，配置/var/www/code/trac/web-devel/conf/trac.ini：

- 插件的配置
- 权限的配置

- <http://trac.edgewall.org/wiki/TracPermissions>
- <http://trac.edgewall.org/wiki/TracFineGrainedPermissions>

web 自定义配置

- 注册用户，授予相关用户 webAdmin 管理员权限：

```
trac-admin /path/to/projenv permission add bob TRAC_ADMIN
```

- 通过 webAdmin 配置

资源（例子）

- Trac 的访问：<http://172.16.2.41/code>
- gitweb 的访问：<http://172.16.2.41/git>
- SVN 的访问在 http://172.16.2.41/svn/project_name 下
- git 的访问在 [git://172.16.2.41/project_name](http://172.16.2.41/project_name) 下
- git 的提交在 ssh://git@172.16.2.41:project_name 下