公共访问

匿名的读取权限该怎么实现呢？也许除了内部私有的项目之外，你还需要托管一些开源项目。或者因为要用一些自动化的服务器来进行编译，或者有一些经常变化的服务器群组，而又不想整天生成新的 SSH 密钥 — 总之，你需要简单的匿名读取权限。

或许对小型的配置来说最简单的办法就是运行一个静态 web 服务，把它的根目录设定为 Git 仓库所在的位置，然后开启本章第一节提到的 post-update 挂钩。这里继续使用之前的例子。假设仓库处于 /opt/git 目录，主机上运行着 Apache 服务。重申一下，任何 web 服务程序都可以达到相同效果；作为范例，我们将用一些基本的 Apache 设定来展示大体需要的步骤。

首先，开启挂钩：

$ cd project.git

$ mv hooks/post-update.sample hooks/post-update

$ chmod a+x hooks/post-update

post-update 挂钩是做什么的呢？其内容大致如下：

$ cat .git/hooks/post-update

#!/bin/sh

#

# An example hook script to prepare a packed repository for use over

# dumb transports.

#

# To enable this hook, rename this file to "post-update".

#

exec git-update-server-info

意思是当通过 SSH 向服务器推送时，Git 将运行这个 git-update-server-info 命令来更新匿名 HTTP 访问获取数据时所需要的文件。

接下来，在 Apache 配置文件中添加一个 VirtualHost 条目，把文档根目录设为 Git 项目所在的根目录。这里我们假定 DNS 服务已经配置好，会把对 .gitserver 的请求发送到这台主机：

<VirtualHost \*:80>

ServerName git.gitserver

DocumentRoot /opt/git

<Directory /opt/git/>

Order allow, deny

allow from all

</Directory>

</VirtualHost>

另外，需要把 /opt/git 目录的 Unix 用户组设定为 www-data ，这样 web 服务才可以读取仓库内容，因为运行 CGI 脚本的 Apache 实例进程默认就是以该用户的身份起来的：

$ chgrp -R www-data /opt/git

重启 Apache 之后，就可以通过项目的 URL 来克隆该目录下的仓库了。

$ git clone http://git.gitserver/project.git

这一招可以让你在几分钟内为相当数量的用户架设好基于 HTTP 的读取权限。另一个提供非授权访问的简单方法是开启一个 Git 守护进程，不过这将要求该进程作为后台进程常驻 — 接下来的这一节就要讨论这方面的细节。