在服务器上部署 Git

**开始架设 Git 服务器前，需要先把现有仓库导出为裸仓库 — 即一个不包含当前工作目录的仓库。做法直截了当，克隆时用 --bare 选项即可。裸仓库的目录名一般以 .git 结尾，**像这样：

$ git clone --bare my\_project my\_project.git

Cloning into bare repository 'my\_project.git'...

done.

该命令的输出或许会让人有些不解。其实 clone 操作基本上相当于 git init 加 git fetch，所以这里出现的其实是 git init 的输出，先由它建立一个空目录，而之后传输数据对象的操作并无任何输出，只是悄悄在幕后执行。现在 my\_project.git 目录中已经有了一份 Git 目录数据的副本。

整体上的效果大致相当于：

$ cp -Rf my\_project/.git my\_project.git

但在配置文件中有若干小改动，不过对用户来讲，使用方式都一样，不会有什么影响。它仅取出 Git 仓库的必要原始数据，存放在该目录中，而不会另外创建工作目录。

把裸仓库移到服务器上

有了裸仓库的副本后，剩下的就是把它放到服务器上并设定相关协议。假设一个域名为 git.example.com 的服务器已经架设好，并可以通过 SSH 访问，我们打算把所有 Git 仓库储存在 /opt/git 目录下。只要把裸仓库复制过去：

$ scp -r my\_project.git user@git.example.com:/opt/git

现在，所有对该服务器有 SSH 访问权限，并可读取 /opt/git 目录的用户都可以用下面的命令克隆该项目：

$ git clone user@git.example.com:/opt/git/my\_project.git

如果某个 SSH 用户对 /opt/git/my\_project.git 目录有写权限，那他就有推送权限。如果到该项目目录中运行git init 命令，并加上 --shared 选项，那么 Git 会自动修改该仓库目录的组权限为可写（译注：实际上 --shared 可以指定其他行为，只是默认为将组权限改为可写并执行 g+sx，所以最后会得到 rws。）。

$ ssh user@git.example.com

$ cd /opt/git/my\_project.git

$ git init --bare --shared

由此可见，根据现有的 Git 仓库创建一个裸仓库，然后把它放上你和同事都有 SSH 访问权的服务器是多么容易。现在已经可以开始在同一项目上密切合作了。

值得注意的是，这的的确确是架设一个少数人具有连接权的 Git 服务的全部 — 只要在服务器上加入可以用 SSH 登录的帐号，然后把裸仓库放在大家都有读写权限的地方。一切都准备停当，无需更多。

下面的几节中，你会了解如何扩展到更复杂的设定。这些内容包含如何避免为每一个用户建立一个账户，给仓库添加公共读取权限，架设网页界面，使用 Gitosis 工具等等。然而，只是和几个人在一个不公开的项目上合作的话，仅仅是一个 SSH 服务器和裸仓库就足够了，记住这点就可以了。

小型安装

如果设备较少或者你只想在小型开发团队里尝试 Git ，那么一切都很简单。架设 Git 服务最复杂的地方在于账户管理。如果需要仓库对特定的用户可读，而给另一部分用户读写权限，那么访问和许可的安排就比较困难。

SSH 连接

如果已经有了一个所有开发成员都可以用 SSH 访问的服务器，架设第一个服务器将变得异常简单，几乎什么都不用做（正如上节中介绍的那样）。如果需要对仓库进行更复杂的访问控制，只要使用服务器操作系统的本地文件访问许可机制就行了。

如果需要团队里的每个人都对仓库有写权限，又不能给每个人在服务器上建立账户，那么提供 SSH 连接就是唯一的选择了。我们假设用来共享仓库的服务器已经安装了 SSH 服务，而且你通过它访问服务器。

有好几个办法可以让团队的每个人都有访问权。第一个办法是给每个人建立一个账户，直截了当但略过繁琐。反复运行 adduser 并给所有人设定临时密码可不是好玩的。

第二个办法是在主机上建立一个 git 账户，让每个需要写权限的人发送一个 SSH 公钥，然后将其加入 git 账户的~/.ssh/authorized\_keys 文件。这样一来，所有人都将通过 git 账户访问主机。这丝毫不会影响提交的数据 — 访问主机用的身份不会影响提交对象的提交者信息。

另一个办法是让 SSH 服务器通过某个 LDAP 服务，或者其他已经设定好的集中授权机制，来进行授权。只要每个人都能获得主机的 shell 访问权，任何可用的 SSH 授权机制都能达到相同效果。