**方法一：**

1.初始化一个空的裸仓库

$ cd /home/repo

$ mkdir teamwork.[**Git**](http://lib.csdn.net/base/git)

$ cd teamwork.[**git**](http://lib.csdn.net/base/git)

$ git init - -bare      注：这是在服务器上运行的，是为了初始化一个根级的git仓库

2.空仓库是不能clone的，所以需要某一个有写权限的人初始 化一个版本

usr@~/work$ mkdir teamwork-ori

usr@~/work$ cd teamwork-ori/

usr@~/work/teamwork-ori$ git init 注：这是在用户端的PC上执行的，为的是初始化一个本地的版本库

usr@~/work/teamwork-ori$ echo “/\*add something\*/” > hello

usr@~/work/teamwork-ori$ git add .

usr@~/work/teamwork-ori$ git commit -am “initial version”

usr@~/work/teamwork-ori$ git remote add origin git@<server>:teamwork.git

usr@~/work/teamwork-ori$ git push origin master

到此为止teamwork已经有了一个版本了，team的其他成员只要先clone一下 teamwork仓库，就可以任意玩了。

注：origin是默认远程版本库。

**方法二：**

1.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

 Initialized empty Git repository in /opt/projects/my\_project.git/

现在 my\_project.git 目录中已经有了一份 Git 目录数据的副本。

2.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。

**注意：方法1是无法在裸仓库上进行git操作的，如果进行了git命令操作，会提示：fatal: This operation must be run in a work tree。而方法二是可以进行git命令操作的。**