Git的使用-1

千易雲(北京)教育科技有限公司

一. Git服务

Git简史

同生活中的许多伟大事件一样，Git 诞生于一个极富纷争大举创新的年代。Linux 内核开源项目有着为数众广的参与者。绝大多数的 Linux 内核维护工作都花在了提交补丁和保存归档的繁琐事务上（1991－2002年间）。到 2002 年，整个项目组开始启用分布式版本控制系统 BitKeeper 来管理和维护代码。

到了 2005 年，开发 BitKeeper 的商业公司同 Linux 内核开源社区的合作关系结束，他们收回了免费使用 BitKeeper 的权力。这就迫使 Linux 开源社区（特别是 Linux 的缔造者 Linus Torvalds ）不得不吸取教训，只有开发一套属于自己的版本控制系统才不至于重蹈覆辙。他们对新的系统制订了若干目标：

* 速度 * 简单的设计 * 对非线性开发模式的强力支持（允许上千个并行开发的分支） * 完全分布式 * 有能力高效管理类似 Linux 内核一样的超大规模项目（速度和数据量）

自诞生于 2005 年以来，Git 日臻成熟完善，在高度易用的同时，仍然保留着初期设定的目标。它的速度飞快，极其适合管理大项目，它还有着令人难以置信的非线性分支管理系统（见第三章），可以应付各种复杂的项目开发需求。

Git基础

Git 和其他版本控制系统的主要差别在于，Git 只关心文件数据的整体是否发生变化，而大多数其他系统则只关心文件内容的具体差异。这类系统 （CVS，Subversion，Perforce，Bazaar 等等）每次记录有哪些文件作了更新，以及都更新了哪些行的什么内容

Git 并不保存这些前后变化的差异数据。实际上，Git 更像是把变化的文件作快照后，记录在一个微型的文件系统中。每次提交更新时，它会纵览一遍所有文件的指纹信息并对文件作一快照，然后保存一个指向这次快照 的索引。为提高性能，若文件没有变化，Git 不会再次保存，而只对上次保存的快照作一链接

Git基础

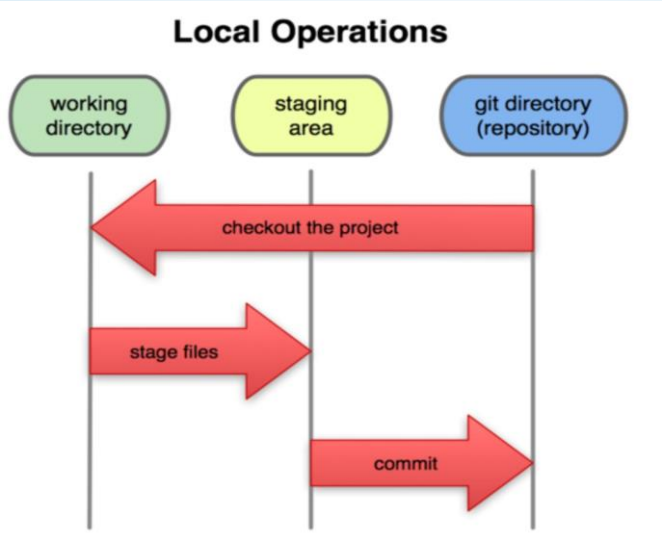
时刻保持数据完整性

在保存到 Git 之前，所有数据都要进行内容的校验和（checksum）计算，并将此结果作为数据的唯一标识和索引。换句话说，不可能在你修改了文件或目录之后，Git 一无所知。这项特性作为 Git 的设计哲学，建在整体架构的最底层。所以如果文件在传输时变得不完整，或者磁盘损坏导致文件数据缺失，Git 都能立即察觉。

Git 使用 SHA-1 算法计算数据的校验和，通过对文件的内容或目录的结构计算出一个 SHA-1 哈希值，作为指纹字符串。该字串由 40 个十六进制字符（0-9 及 a-f）组成，

Git文件的三种状态

对于任何一个文件，在 Git 内都只有三种状态：已提交（committed），已修改（modified）和已暂存（staged）。已提交表示该文件已经被安全地保存在本地数据库 中了；已修改表示修改了某个文件，但还没有提交保存；已暂存表示把已修改的文件放在下次提交时要保存的清单中。



由此我们看到 Git 管理项目时，文件流转的三个工作区域：Git 的工作目录，暂存区域，以及本地仓库。

Git文件的三种状态

每个项目都有一个 Git 目录（注：如果 git clone 出来的话，就是其中 .git 的目录；如果git clone --bare 的话，新建的目录本身就是 Git 目录。），它是 Git 用来保存元数据和对象数据库的地方。该目录非常重要，每次克隆镜像仓库的时候，实际拷贝的就是这个目录里面的数据。

从项目中取出某个版本的所有文件和目录，用以开始后续工作的叫做工作目录。这些文件实际上都是从 Git 目录中的压缩对象数据库中提取出来的，接下来就可以在工作目录中对这些文件进行编辑。

所谓的暂存区域只不过是个简单的文件，一般都放在 Git 目录中。有时候人们会把这个文件叫做索引文件，不过标准说法还是叫暂存区域。

Git工作流程

1. 在工作目录中修改某些文件。
2. 对修改后的文件进行快照，然后保存到暂存区域。
3. 提交更新，将保存在暂存区域的文件快照永久转储到 Git 目录中。

所以，我们可以从文件所处的位置来判断状态：如果是 Git 目录中保存着的特定版本文件，就属于已提交状态；如果作了修改并已放入暂存区域，就属于已暂存状态；如果自上次取出后，作了修改但还没有放到暂存区域，就 是已修改状态。到第二章的时候，我们会进一步了解其中细节，并学会如何根据文件状态实施后续操作，以及怎样跳过暂存直接提交。

Git安装Git

在 Linux 上安装

$ yum install git-core

$ apt-get install git-core

在 Windows 上安装

https://git-for-windows.github.io/

直接下载安装即可

Git如何用

首先你要点进去https://github.com/这个网站，注册一个github账号。注册好之后，要记住邮箱和密码。

打开之前的那个图标就是这样一个窗口，首先Git是分布式版本控制系统，所需要填写用户名和邮箱作为一个标识，分别输入这两个命令，用户名和邮箱需要换成自己的。

git config –global user.name “my-name”

git config –global user.email “test@sina.com"

打开你的项目所在目录**现在比如我在我的C:/wamp/www/aaa有一个项目**

cd c:/wamp/www/aaa

Git如何用

管理你的项目

git init

尝试些一些项目 这里我们写一个首页叫index.html

在git里面输入 git status

我们可以看到index.html这个文件是红色，这是说明我们这个文件已经做了修改，但是还没保存到本地仓库里面。

执行

git add index.html

Git如何用

这里我们发现 git status 之后index.html变成绿色

git commit -m “first”

使用git status发现上面说工作区很干净，没有文件要被提交。

Git如何用

首先你得明白这几个概念，工作区就是你现在编辑器所处的那个工程里面，在这个指的就是aaa这个文件夹。刚才你写了一个index.html写完之后就是往工作区增添了一个文件，然后你用git status查看状态发现，index.html是红色的。git status是什么意思呢，就是查看你工作区和暂存区有没有文件没被提交到本地仓库，如果有工作室未向暂存区保存的就显示红色，如果有暂存区没提交到本地仓库的就是绿色。那么什么是本地仓库呢，就是一开始说的那个不可见的文件夹，你一执行git init命令就会有一个本地仓库出来。

Git如何用

现在我们接着来看，如何从工作区把文件提交到暂存区，就是使用命令git add index.html就可以了。把文件从暂存区提交到本地仓库呢，就是使用命令git commit –m “the first time” 这个引号里面的内容是随意的，就是自己添加一个备注，比如自己改动了什么东西。理解了这句话，我们接着进行操作，我们我的index.html里面继续添加一段话。我再git status查看一下当前的状态，发现有未向暂存区提交的保存，接着我们采用 git commit –m “the second time”命令将暂存区的文件提交到本地仓库。之后再用git status来查看一下，发现已经没有文件要提交了。

Git版本回退

到目前为止，我们已经向工作区提交了两次修改。Git给我们提供了一个可以查看我们最近的提交历史。

git log

如果看的很乱，可以使用这个命令，

git log -–pretty=oneline

回退到上一个版本。
git reset –hard HEAD^

Git版本回退

git log –pretty=oneline 可以获得版本号

git reset –hard 版本号也可以回退这里版本号不用写全，写一部分能表示即可

Git删除文件

现在我们在aaa目录下新建一个文件叫test.txt。我们在里面随便写几个字之后，通过git add text.txt 已经git commit –m “the third time”就已经把test.txt已经写入本地库里了。

假设现在我们发现我们已经不需要test.txt这个文件了，所以我们把它给删了。这个时候你通过git status发现工作区和本地库里的文件不一样了。

现在我们有两种选择，一种是我们确实要删除。使用

git rm test.txt
Git commit –m “remve test.txt” 这个时候会发现我们本地库中的文件以及被删除了

Git删除文件

另外一种情况就是我删错了，我想还原回来怎么办呢，使用命令

git checkout – test.txt

就发现这个文件已经还原了。