# Git

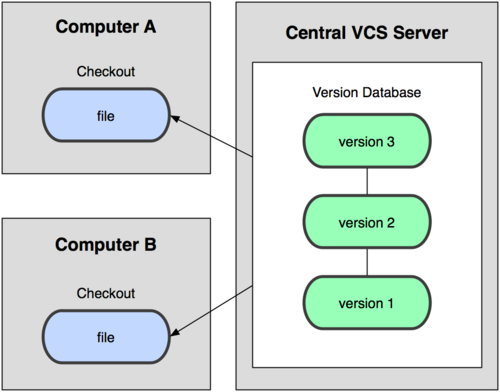
# 什么是Git

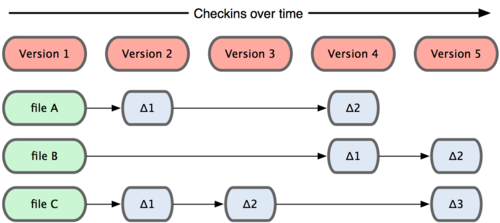
Git是一个版本控制工具

# Git和Svn区别在哪里?

## 从底层原理的角度

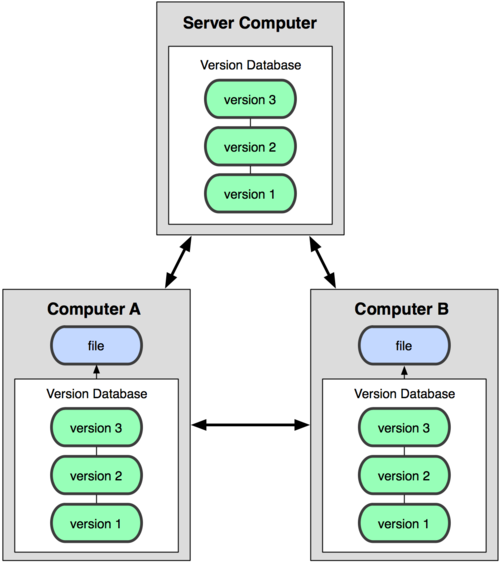
**SVN （CVCS）: SVN是集中化的版本控制系统，它有一个单一的集中管理的服务器，用于保存所有文件的修订版本。协同工作的人通过客户端.上传、更新文件或下载最新文件。**

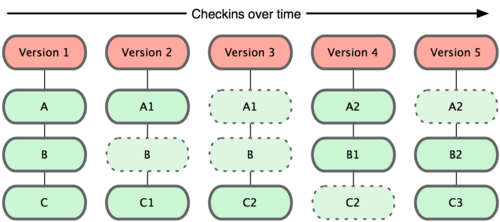




**Git：Git和Svn不同，Git是分布式版本控制系统，客户端并不只是拉取最新快照，而是将原始代码仓库镜像下来，这么一来，每个客户端都有一套完整的代码，如果git服务器发生故障，那么可以用客户端代码恢复。**

**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。**





## 从使用者的角度

1. **因为Git在客户端本地保存有一套完整的仓库，所以在无网络的情况下，使用者可以在本地提交代码文件。而SVN则只能在有网络的情况下提交代码。**
2. **git对程序源代码进行差异化的版本管理，代码库占极少的空间。易于代码的分支化管理。不支持中文，图形界面支持差，使用难度大，所以Git基本上只是开发人员使用。而SVN对中文支持好，操作简单，使用没有难度，美工人员，产品人员，测试人员，实施人员都可轻松上手。使用界面统一，功能完善，操作方便。**
3. **Git没有一个全局的版本号，而SVN有。**
4. **Git的内容完整性要优于SVN ：GIT的内容存储使用的是SHA-1哈希算法。这能确保代码内容的完整性，确保在遇到磁盘故障和网络问题时降低对版本库的破坏。**

# Git安装

## Github

### 安装客户端

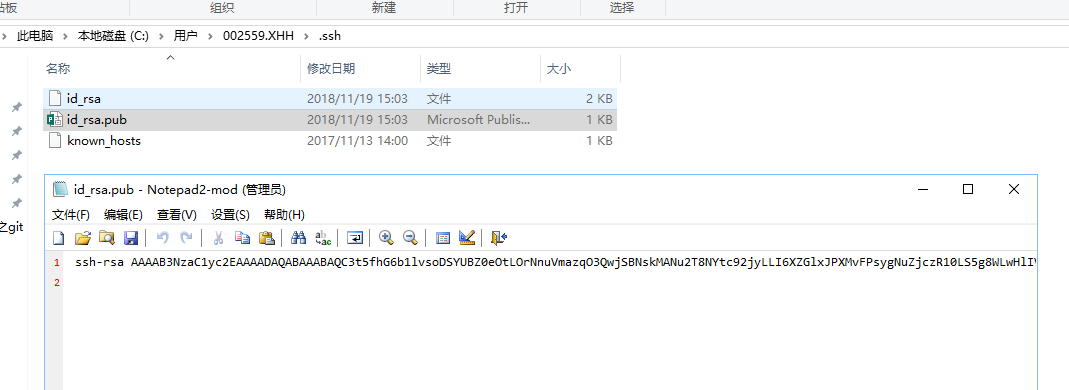
<https://gitforwindows.org/>

**首先安装好Github客户端。**

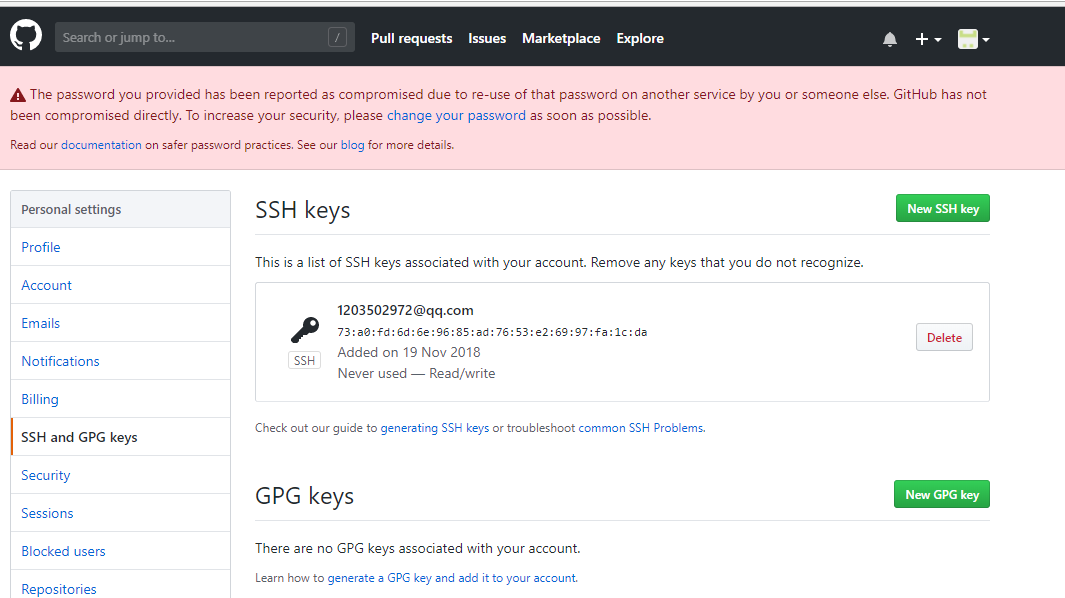
### 配置Git

#### 首先在本地创建ssh key

$ ssh-keygen -t rsa -C "1203502972@qq.com"

后面的1203502972@qq.com改为我们在github上注册的邮箱，之后会要求确认路径和输入密码，我们这使用默认的一路回车就行。成功的话会在~/下生成.ssh文件夹，进去，打开id\_rsa.pub，复制里面的key。

回到github上，进入 Account Settings（账户配置），左边选择SSH Keys，Add SSH Key,title随便填，粘贴在电脑上生成的key。



#### 测试连接

验证是否成功，在git bash下输入：

$ ssh -T git@github.com

如果是第一次的会提示是否continue，输入yes就会看到：You've successfully authenticated, but GitHub does not provide shell access 。这就表示已成功连上github。

#### 设置username和email

我们需要设置username和email，因为github每次commit都会记录他们。

$ git config --global user.name "your name"

$ git config --global user.email "your\_email@youremail.com"

#### 添加远程仓库地址

$ git remote add origin git@github.com:yourName/yourRepo.git

后面的yourName和yourRepo表示你再github的用户名和刚才新建的仓库，加完之后进入.git，打开config，这里会多出一个remote "origin"内容，这就是刚才添加的远程地址，也可以直接修改config来配置远程地址。

#### 检出仓库

git clone username@host:/path/to/repository