



## 刘坤的技术博客

- [BlueBox](#)
- [所有文章](#)

嗨，我是刘坤，一名来自中国的 IOS 开发者，现就职于杭州阿里，花名‘念纪’，沉淀技术，寻求创意

[GitHub](#) [RSS](#)

### 友情链接

• [Casatwy Talovum](#)  
• [gf&zjの盗梦空间](#)  
• [明弈](#)

## Gitlab的使用

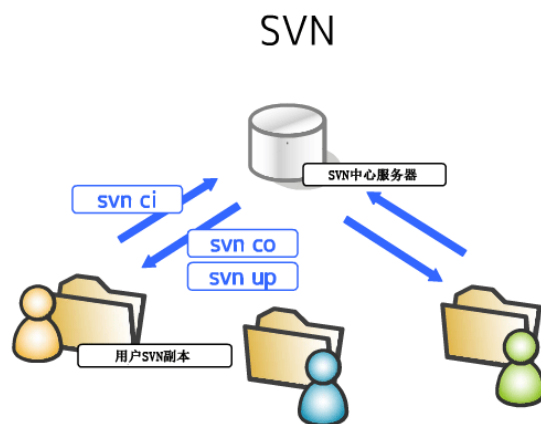
最近成功的在公司部署了gitlab，鉴于同学们还不会使用，这里写篇博客说明下。如果想安装gitlab的话，需要一些linux的基础知识，我在这里记录了我安装的参考  
《<http://www.cnbluebox.com/?p=378>》

### 1.什么是git

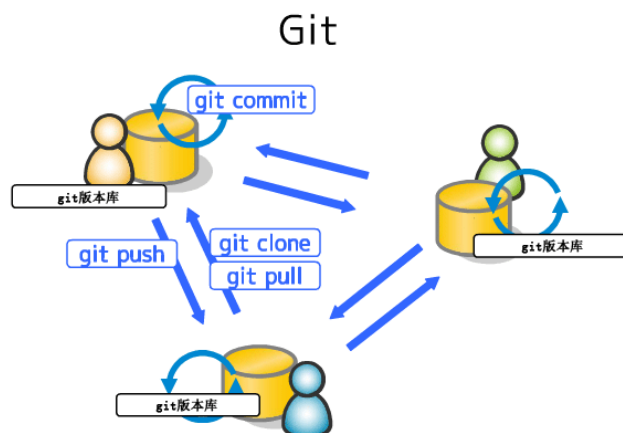
鉴于有同学还没用过git，就先介绍一下git吧。git就是一种版本控制工具。说到版本控制，大家可能就想到了svn。但是两者有着本质的区别。

svn是集中化的版本控制系统，只有一个单一的集中管理的服务器，保存所有文件的修订版本，而协同工作的人们都通过客户端连到这台服务器，取出最新的文件或者提交更新。

git是分布式的版本控制系统，每一个终端都是一个仓库，客户端并不只提取最新版本的文件快照，而是把原始的代码仓库完整地镜像下来。每一次的提取操作，实际上都是一次对代码仓库的完整备份。



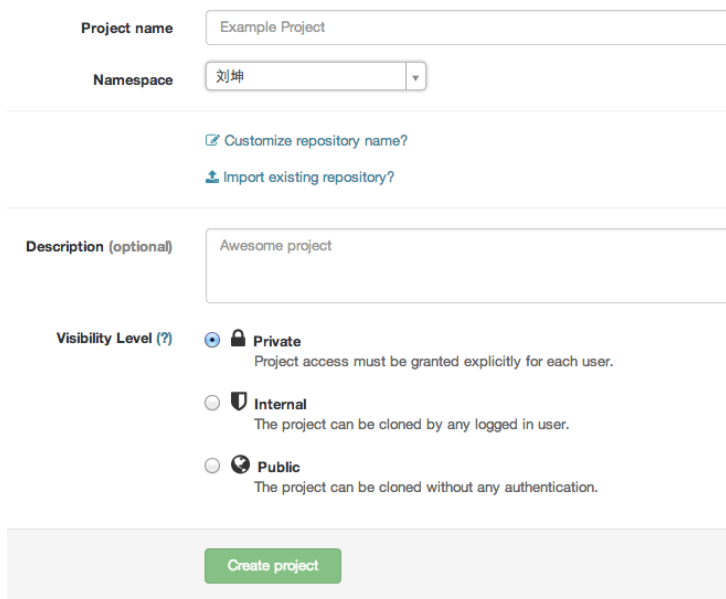
集中式版本控制：



分布式版本控制：

### 2.GitLab创建工程

点击导航条上的“+”就可以进入创建工程页面



这里面很简单,主要讲两个地方:

### 2.1 namespace

这个选择是用来决定这个工程所属的, 可以选User为你自己。或者选择组, 这个会影响到后面工程的url。例如我选择了组suning创建工程SuningTest、那么这个工程就会在这个组内可见, 那么访问路径就是<https://domain.com/suning/suningtest>

### 2.2 Visibility Level

权限等级分三种:

- Private 私有的, 只有你自己或者组内的成员能访问
- Internal 所有登录的用户
- Public 公开的, 所有人都可以访问

## 3.Git的使用

### 3.1 添加sshkey

git仓库之间的代码传输协议主要使用ssh协议。而一般搭建gitlab的时候使用的git用户是没有密码的, 因此直接ssh是不能登录的, 就需要使用ssh-keygen上传公钥, 使用非对称加密传输。下面讲述如何上传你的ssh公钥:

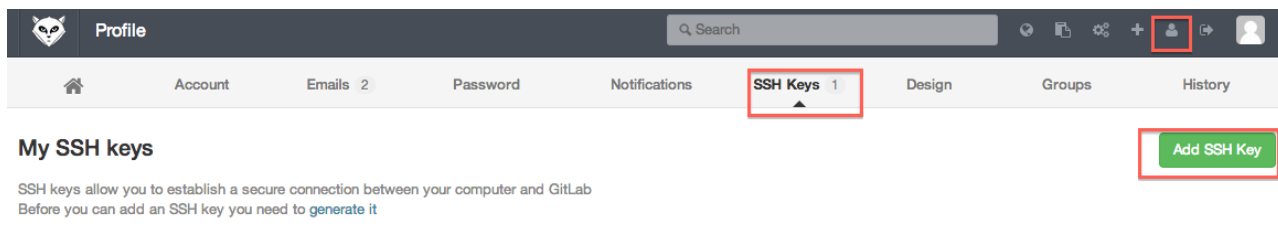
#### 3.1.1生成sshkey

在终端中敲下面的命令, 第一步会生成一对私钥和公钥, 分别存在 ~/.ssh/id\_rsa和~/.ssh/id\_rsa.pub中。第二步查看公钥字符串。

```
ssh-keygen -t rsa -C "$your_email"
cat ~/.ssh/id_rsa.pub
```

#### 3.1.2保存sshkey到gitlab

在面板上依次点击Profile Settings -> SSH Keys -> Add SSH Keys。然后把上一步中的id\_rsa.pub中的内容拷贝出来粘贴到输入框中, 保存。



完成上面两步之后就成功的添加了sshkey了, 然后就可以上传代码了。

### 3.2 初始上传代码

如果你已经使用过git了, 那么这一步对你来说可以跳过了。整体来说比较简单的。下面的\$project\_root代表工程根目录

- 进入工程目录 `cd $project_root`
- 初始化git仓库 `git init`
- 添加文件到仓库 `git add .`

- 提交代码到仓库 `git commit -m 'init commit'`
- 链接到git server `git remote add origin git@example.com:namespace/projectname.git`
- push代码到服务器 `git push origin master`

### 3.3 克隆代码到本地

在svn中，我们都叫checkout. 把代码checkout到本地。而git中我们叫克隆，克隆会把整个仓库都拉到本地。

如，我要把刚才的工程再clone到本地。

```
git clone git@example.com:namespace/projectname.git
```

### 3.4 设置gitignore

有一些文件或文件夹是我们不想要被版本控制的，比如.DS\_Store build\ xcuserdata thumbs.db, git提供了一种忽略的方案。

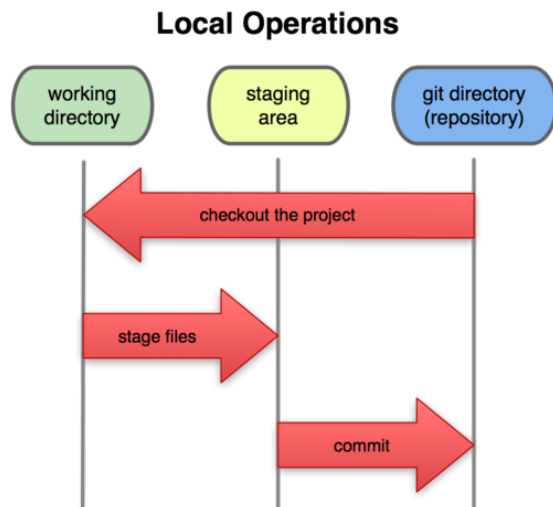
在项目根目录下创建.gitignore文件，然后把需要忽略的文件或文件夹名写进去。这样就可以忽略这些文件受版本控制啦。

svn也提供了这样忽略的方案，svn也可以设置全局忽略。svn的此配置放在~/.subversion/config中global-ignores的值。

通过设置ignore，我们可以实现git和svn双管理哦，就是在svn忽略.git文件夹，在gitignore中忽略.svn文件夹，有兴趣的同学可以试试

### 3.5 git文件的状态

git管理下的文件有3种状态，如下图：



### 3.6 git基础

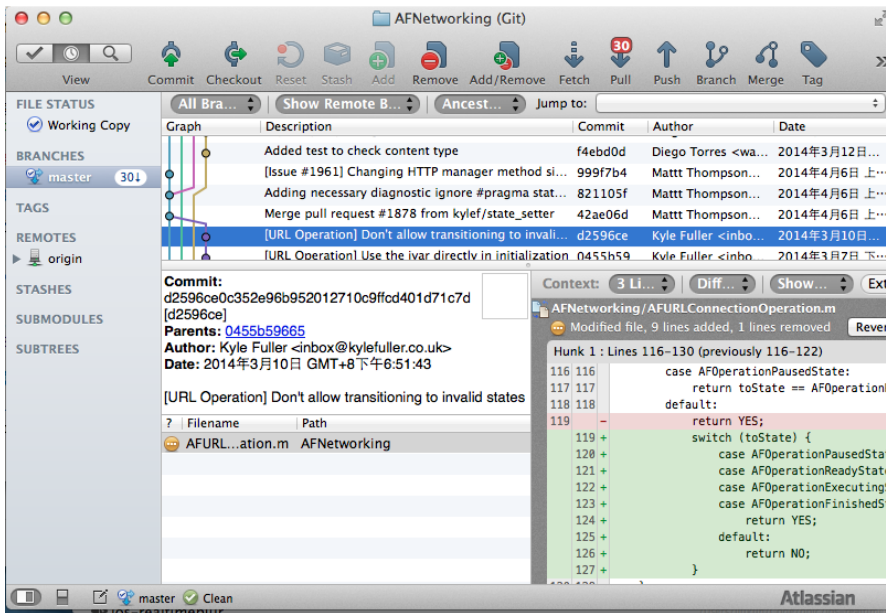
git博大精深，个人感觉，不管是从其强大的功能上，还是从其实现方案上来讲比svn强大很多。这里贴个学习链接吧，这一系列讲的非常全面非常详细：

[Git基础](#)

### 3.7 图形化git管理工具

如果说你不喜欢命令行的使用方式，你也可以使用图形界面工具SoureTree: <http://www.sourcetreeapp.com/> 界面简洁，使用方便,功能强大。

贴个预览：



## Comments

Copyright © 2016 刘坤 Design credit: [Shashank Mehta](#)