# 远程阀控git使用说明

为有效管理远程阀控代码，在本项目的ubuntu服务器上部署了git服务，此项目所有代码需上传git。

git服务地址：<http://118.31.102.18/>

windows安装git客户端：<https://my.oschina.net/longxuu/blog/141699>

gitlab教程：<https://zhangmengpl.gitbooks.io/gitlab-guide/content/introduce.html>

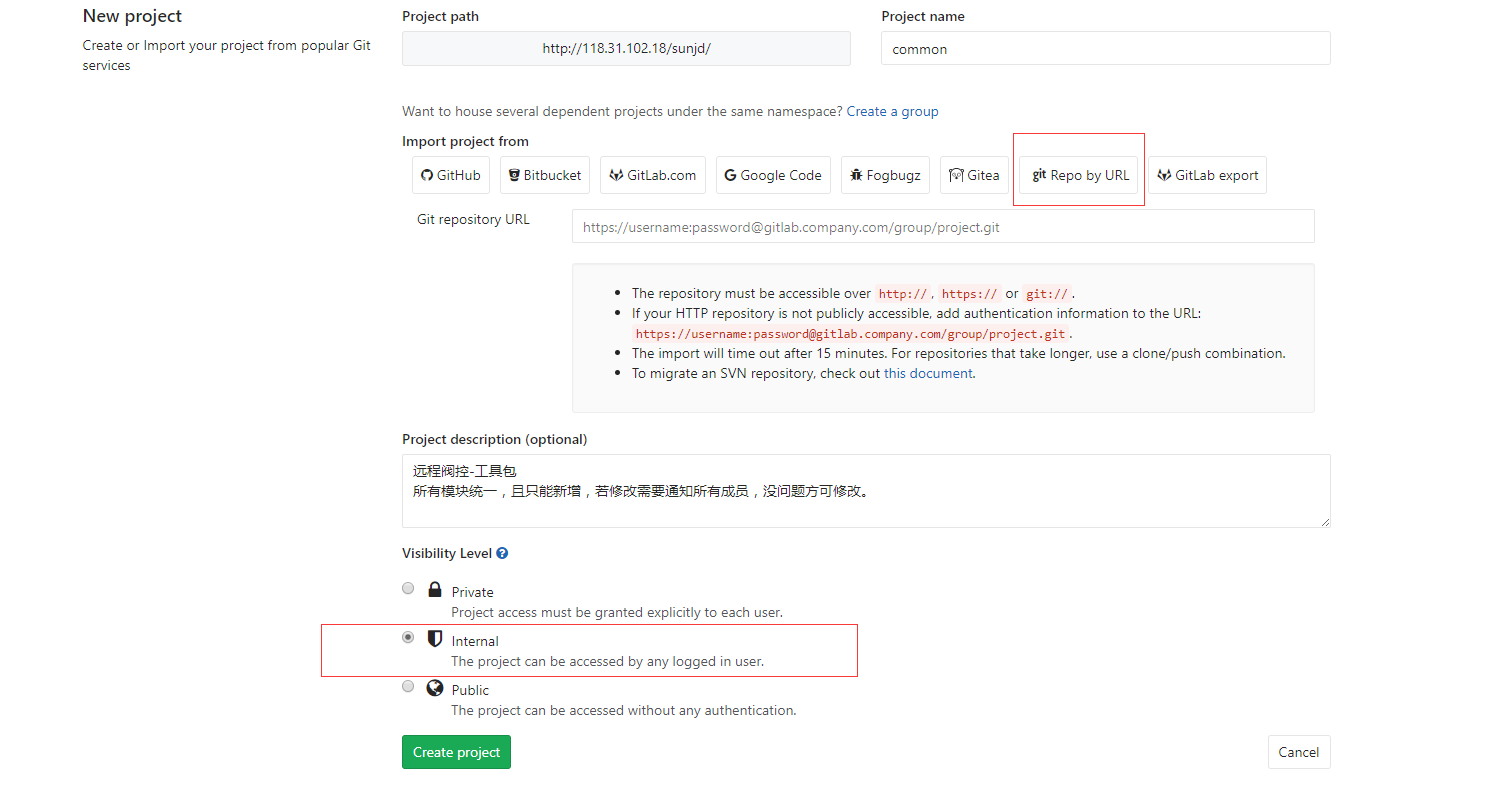
用户名规范：

用户名使用姓名全拼，或姓全拼+名字首字母（为了查看提交历史时，容易知道对应人员）

例如： 孙缙东 用户名为 sunjindong 或 sunjd

项目规范：创建项目时注意事项

1. 选择Repo by URL
2. 选择 Internal



Git工作流

1. 分支

master：

主分支

develop：

开发分支，用于某个版本的开发，例如

develop/ ver1

develop/ ver2

release：

发布分支，用于某一个发布版本，例如

release/V1.0

release/V1.1

feature：

功能开发分支，用于开发一个新的功能，例如

feature/login

feature/message

hotfix：

BUG/漏洞修复分支，例如

hotfix/xxx

1. 工作流程

新项目开发

1. 项目维护者在GitLab上建立项目，建立master，develop，release分支。
2. 团队的其它成员clone该项目到本地。
3. 各个开发人员以各自的开发任务建立相应的功能分支如feature/XXX。
4. 功能分支开发完成，push至服务端。
5. 开发者通过GitLab发起一个合并到develop 的Merge Request。
6. 团队的其它成员review code，讨论并修改。
7. 项目维护者合并功能到官方仓库中并关闭Merge Request。
8. 删除feature/XXX分支。

新功能开发

1. 开发者在本地仓库中从相应的develop分支新建一个专门的分支开发功能，如feature/XXX。
2. 开发者push分支修改到服务端。
3. 开发者通过GitLab发起一个合并到develop 的Merge Request。
4. 团队的其它成员review code，讨论并修改。
5. 项目维护者合并功能到官方仓库中并关闭Merge Request。
6. 删除feature/XXX分支。

版本发布

1. 项目维护者选择合适的develop，建立release分支如release/V1.0

已发布版本BUG修复

1. 根据已发布版本建立hotfix分支如hotfix/XXX。
2. 开发者push分支修改到服务端。
3. 开发者通过GitLab发起一个合并到develop 的Merge Request。
4. 团队的其它成员review code，讨论并修改。
5. 项目维护者合并功能到官方仓库中并关闭Merge Request。
6. 删除hotfix/XXX分支
7. 项目维护者建立新的release分支。
8. 说明

分支可以有多级路径用来区分不同的应用场景下的同一项目，例如在不同的城市部署同一个项目需要略有不同的配置文件，那么分支名称可以是release/NANJIN/V1，release/HANGZHOU/V1等等。但建议应当为不同的部署建立不同的项目，对共用代码以Submodules的方式进行引用。

团队合作时具体的人员分配具体的功能开发任务，然后建立相应的分支，不允许出现同一功能交叉开发的现象。合并请求必须进行review ，有条件的需要进行静态及动态代码分析，在完全实现业务逻辑的基础方可建立release分支。

同样的，出现BUG需要修复是也是指派到具体的人员建立hotfix分支进行修复，不能出现同一BUG交叉修改的现象。

对于已发布的版本出现BUG需要建立hotfix分支修复，按照流程最终建立新的release分支，而对于develop出现的BUG这可以直接在当前分支上修改。

master分支可以用来存放项目说明，操作手册及其他说明性质的文件。