1. 注意根节点和home节点，home节点的路径~开头，~/代表当前用户文件夹
2. 使用cd /a/b是绝对路径，cd a相对路径
3. Scp 命令
4. Ssh 命令
5. 创建文件夹命令 mkdir
6. 修改文件 vi ，dd删除每行，dw删除光标后，:wq 写入并退出
7. Git 提交

进入你的项目目录

step1：拉取服务器代码，更新本地代码，避免覆盖他人代码

输入命令： git pull  （拉取master分支的代码）

                        git pull origin dev  拉取dev分支的代码

step2：查看当前项目中有哪些文件被修改过

输入命令：git status

具体状态如下：

1：Untracked: 未跟踪,一般为新增文件，此文件在文件夹中, 但并没有加入到git库, 不参与版本控制. 通过git add 状态变为Staged.

2：Modified: 文件已修改, 仅仅是修改, 并没有进行其他的操作.

3：deleted： 文件已删除，本地删除，服务器上还没有删除.

4：renamed：

step 3：将状态改变的代码提交至缓存

输入命令：git add .

使用上面的命令将所有的修改的文件提交到缓存区

git add + 文件

git add -u + 路径：将修改过的被跟踪代码提交缓存

git add -A + 路径: 将修改过的未被跟踪的代码提交至缓存

例如：

git add -u vpaas-frontend/src/components

将 vpaas-frontend/src/components 目录下被跟踪的已修改过的代码提交到缓存中

git add -A vpaas-frontend/src/components

将 vpaas-frontend/src/components 目录下未被跟踪的已修改过的代码提交到缓存中

step 4：将代码提交到本地仓库中

输入命令： git commit -m “修改项目代码”

step 5：将缓存区代码推送到远程仓库

输入命令：git push   （推送代码到master分支）

                        git push origin dev     推送代码到dev分支

1. 修改git远程分支名字

先将本地分支重命名

git branch -m oldBranch newBranch

删除远程分支（远端无此分支则跳过该步骤）

git push --delete origin oldBranch

将重命名后的分支推到远端

git push origin newBranch

把修改后的本地分支与远程分支关联

git branch --set-upstream-to origin/newBranch