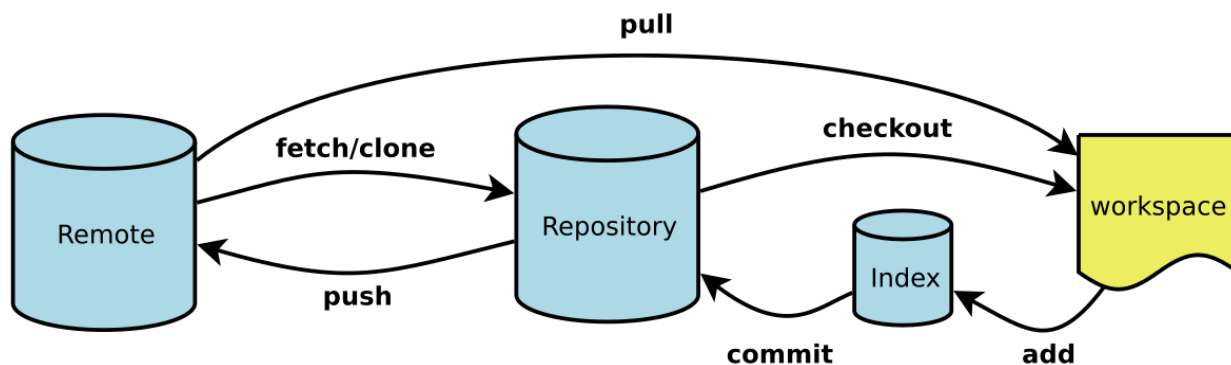


git

1.git简介



- workspace 工作区
- index 暂存区
- repository 本地版本库
- remote 远程仓库

2.常用git命令

所有的git命令必须在工作区内能使用，工作区就是你的项目目录

- git --version 查看版本
- git config -l 查看配置信息
- 修改name和email
 - git config --global user.name 'ddd'
 - git config --global user.email 'ddd@jdjdjd.com'
 - git config --global push.default matching
- 克隆版本库
 - git clone <版本库地址> <本地目录> 将远端服务上的项目克隆到本地
- 创建版本库
 - git init 初始化一个新的项目，必须切换到版本库所在目录
- 将文件添加到暂存区
 - git add 文件名 将指定文件添加到暂存区
 - git add . 将所有文件添加到暂存区

- git reset HEAD 撤销暂存区的修改
- 查看修改
 - git diff
- 提交代码到版本库
 - git commit <文件名> -m "说明信息"
 - git commit -a -m 把工作区中修改添加到暂存区并提交到本地库
- 将本地代码推送到远端托管服务器
 - git push <远程主机名> <本地分支名>:<远程分支名> 如果远程分支不存在，则创建
 - git push origin master 默认推送到主分支
 - git push origin : 分支名 删除远程分支
 - git push --all origin 将本地分支推送到服务器
 - git push --force origin 如果远程主机的版本比本地版本更新，推送时Git会报错，要求先在本地做 `git pull` 合并差异，然后再推送到远程主机。这时，如果你一定要推送，可以使用 `--force` 选项。
- 从远端服务器拉取内容
 - git pull <远程主机名> <远程分支名>:<本地分支名>
 - --allow-unrelated-histories 解决本地和远程库不同的问题
- 删除或恢复文件
 - git rm 从版本库中删除文件
 - git checkout -- 放弃单个文件修改
 - git checkout . 放弃所有文件修改
- 查看版本库状态
 - git status
- 查看提交记录
 - git log
 - git log --pretty=oneline -版本回退
 - git reset HEAD~n
 - HEAD~ 上一个版本
 - HEAD~~上两个版本
 - HEAD~n 上n版本
 - git reset --hard 版本号
 - git reflog 查看每个版本的版本号

3.git使用流程

- 首先到托管服务器上创建一个空版本库，例如在github、coding、oschina等
- 然后克隆到本地(clone)创建一个新项目
- 或者可以通过初始化项目创建一个新项目 `git init`
 - `git remote add origin` 远程仓库地址
- 添加代码文件 (`git add`)
- 提交代码到本地库 `git commit -m`
- 将代码推送到远端服务器 `git push`
- 从远端服务器拉取代码`git pull`

4 冲突管理

- 如果有多个人同时修改同一个文件的相同行，在推送时会有冲突
- 发生冲突后，首先将服务器端代码拉到本地（pull），手动合并冲突
- 然后添加修改后的代码文件，重新提交
- 将代码推送到服务器

5 分支管理

- master 主分支，记录发布版本
- online 线上分支 正在运行的版本
- develop 开发分支 用于测试
- local 本地分支
- 分支管理常用命令
 - `git branch` 查看分支
 - `git branch 新分支 [旧分支]` 基于旧分支创建新分支
 - `git branch -d 分支名` 删除分支
 - `git checkout 分支` 切换到指定分支
 - `git merge 分支名` 将指定分支合并到当前分支

问题

- 忽略提交，在工作区中编写一个.gitignore文件，在文件中记录不需要提交的目录和文件
- 如果克隆的时候出现了：unable to access '<https://git.coding.net/landmark/php1702.git/>': error setting certificate 解决方式：
 - Git config --global http.sslVerify false
- 免密提交
 - 1 生成公钥对

```
cd      #回到家目录
ssh-keygen -t rsa -C "自己的邮箱"
id_rsa 私钥  id_rsa.pub 公钥
```

- 2 将公钥添加到码云服务器上

```
复制id_rsa.pub的内容
码云 ----> 个人设置-->ssh公钥，将复制的内容黏贴到“公钥内容”中
选中永久有效
添加
```

- 3.测试连接

```
ssh -T git@git.coding.net

python@ubuntu:~/.ssh$ ssh -T git@git.coding.net
Warning: Permanently added the RSA host key for IP address
'123.59.83.79' to the list of known hosts.
Coding 提示: Hello landmark, You've connected to Coding.net
via SSH. This is a personal key.
landmark, 你好, 你已经通过 SSH 协议认证 Coding.net 服务, 这是一个
个人公钥 # 表示通过
```

- 4 使用项目的ssh地址，可以clone或init/remote add