

1.Git与SVN的比较

对SVN了解不多，工作中主要还是用Git多一些，相对Git也熟悉些

最大的特点，Git是分布式的版本控制软件，SVN是集中式的版本控制软件，每个拥有仓库访问权限的人，都有完整的仓库副本，在不同同步进度的情况下，版本控制都是在本地完成，简单直接高效，而SVN作为集中式的版本管理软件，进度只保存在服务端，所有的修改和同步都需要和服务端做交互

轻量级以及灵活性，因为Git的版本管理都可以在本地完成，所以Git的操作都非常快速和轻量级，像分支管理，版本迭代相对于SVN都又快又灵活

权限管理，因为Git诞生于开源社区，所以Git并没有提供像SVN那样基于目录的权限管理机制，如果对权限管理有要求，但是又不想通过分库去解决，SVN比Git更合适

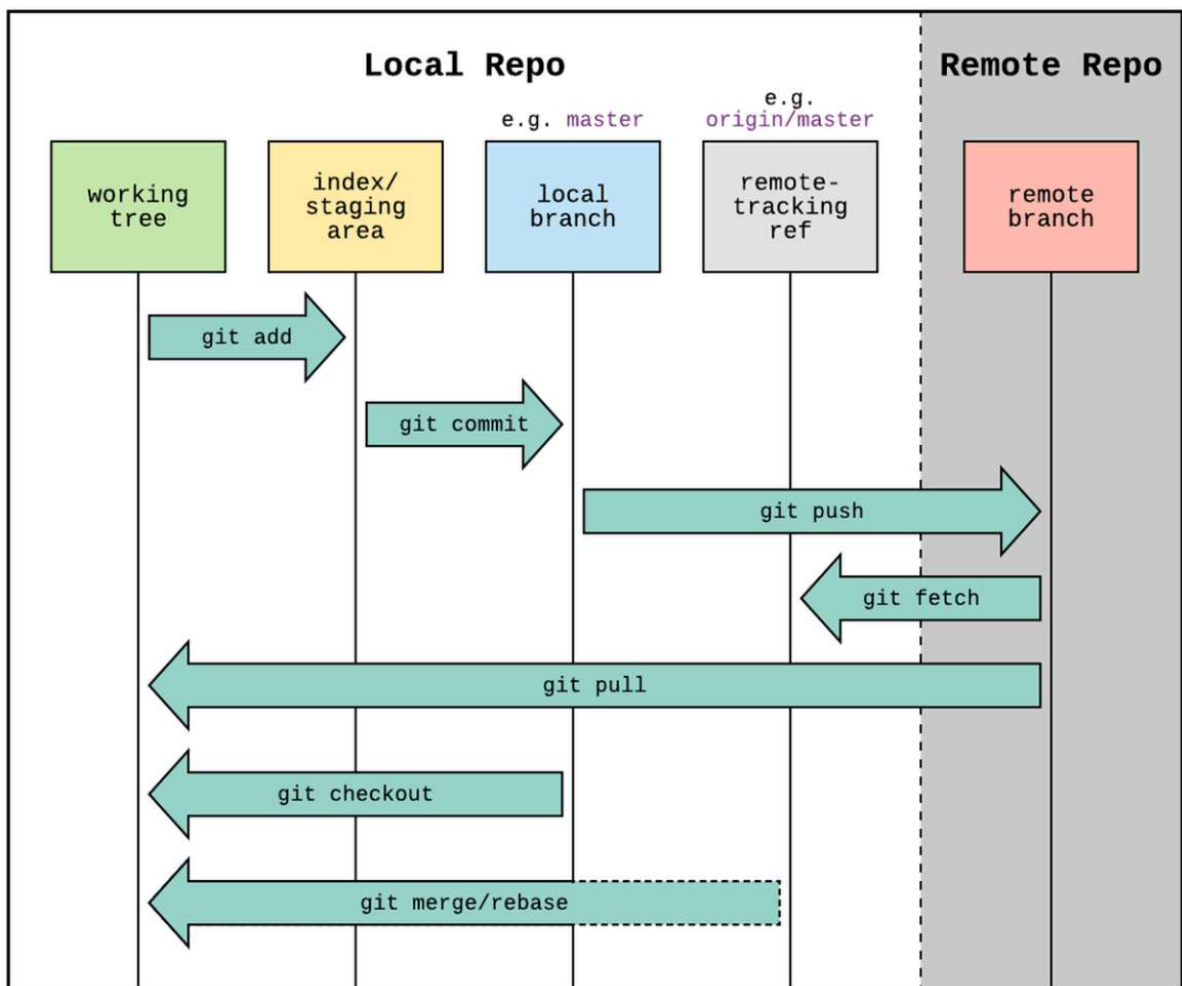
Git的安全性和系统鲁棒性优于SVN，因为Git是分布式的模式，所以并不强依赖于Git服务端的进度，服务端异常对项目安全影响有限，即使是服务的完全挂了，硬盘损坏，救不回来也没有太大关系，只需要重新部署一个服务端，大家重新同步下进度即可，但是SVN如果服务的异常，就只能等待服务端恢复

Git对于文本文件使用基于行的diff算法，对于二进制文件使用二进制比较算法；SVN是基于二进制比较

- **SVN (Subversion的简称)**
- **系统鲁棒 (Robustness) 性**是指系统在面对各种不利条件（如噪声、干扰、故障等）时，仍能保持其正常功能和性能的能力
- 不建议使用Git去管理二进制文件（处理文本文件可以识别并仅存储文件的差异部分，处理二进制文件时，由于无法识别其内部的差异，每次更改都会导致整个文件的新版本被存储；只能检测到文件的更改本身，而无法直接展示二进制文件内容的具体更改）

2.Git的常用命令有哪些？

核心概念



working tree工作区

在电脑上实际操作和修改的文件树，也就是你项目的当前状态。它是你进行代码编辑和修改的地方

index/staging area索引暂存区

一个中间区域，用于临时存放你的改动。它是一个存储在 `.git` 目录下的文件，记录了当前工作目录中所有文件的状态信息，包括文件的修改、新增和删除等操作

local branch本地分支

Git版本控制系统中用于管理不同版本或开发路径的实体

remote-tracking ref远程跟踪引用

本地仓库对远程仓库中的某个远程分支的状态的记录

remote repo远程仓库

托管代码的服务器

常用命令

常用的

```
git init 初始化一个仓库
git clone xxx 克隆一个远程仓库
git add 将文件加入git暂存区
git commit 将暂存区文件提交的本地分支
git status 查看当前工作区所有文件变更
git push 将本地分支推送到远程分支
```

```
git pull 拉取远程分支到本地
git log 查看git提交日志
```

分支操作

```
git branch 列出本地所有分支
git branch -r 列出所有远程分支
git checkout -b xxx 新建一个分支，并切换到新建分支
git branch -d xxx 删除xxx分支
git merge xxx 合并xxx分支到当前分支
```

进阶常用

```
git stash 清理工作区，暂存未提交的变更文件，注意这里是将工作区未commit的内容暂时保持的一个独立的空间，方便同步主分支进度，和git暂存区的概念不要搞混
git reset 重置提交
git rebase 变基
git cherry-pick 摘樱桃，把某一个commit记录合并进当前分支
git tag 标签操作，常用术语commit标注信息的扩展，用来管理commit非常方便
git rm 删除git追踪，不是删除文件
git fetch 只同步进度，不合并改变
git remote 远程分支操作
```

不常用

```
git diff 显示文件或目录的内容差异（不推荐看，正常开发流程都是使用GUI工具做diff，非必要不适用）
git config 配置
```

3.Git命令将一个代码的分支合并到master怎么做？

```
# Step 1: 切换到 master 分支
git checkout master

# Step 2: 同步远程 master 分支的最新进度
git pull origin master

# Step 3: 合并目标分支到 master
git merge feature-branch

# 处理可能出现的冲突，并提交解决后的更改
git add .
git commit -m "Resolved merge conflicts" # 如果有冲突需要解决

# Step 4: 推送合并后的 master 分支到远程仓库
git push origin master
```

4.Git rebase和Git merge有什么区别？

git merge xxx

合并xxx分支到当前分支

git rebase

变基，有代码分支洁癖的用的多，可以让分支变得很干净，不会像merge每次合并都会有一个merge的commit记录，但是rebase是把双刃剑，变基会导致commit链的状态变化，如果这个改变没有及时同步给其他人，变基将是一场灾难，所以不要盲目追随rebase带来的收益，也要评估使用风险，git的本质是方便代码协作，这个应该放在第一位，相比较来说，merge更普世，更方便大家追踪记录

5.Git pull和Git fetch有什么区别？

git pull

从远程获取最新版本并merge到本地，它会自动将提交合并到您的本地存储库中，而无需查看提交

git fetch

从远程获取最新版本到本地，但不会自动merge，用于从远程跟踪分支下载和查看其他人完成的最新提交，但不将这些提交合并到本地存储库中。它从远程存储库中获取更改并将其存储在本地存储库中

6.讲一下Git pull的原理

基于本地的FETCH HEAD记录，对比本地的FETCH HEAD与远程仓库分支的commit版本号，如果不同就会先git fetch远程仓库到本地，然后利用git merge将其与本地的分支合并

基本可以认为是git pull=gitfetch+git merge

7.pull的时候会遇到冲突的情况，如何解决冲突的？

找到冲突点，仔细检查冲突内容

修改冲突，删除多余信息，冲突提示符号等内容

git add冲突文件

git commit提交修改

8. Git怎么回退?



9. Git如何处理分支?

新建

git checkout -b (并切换到新分支)

git branch

删除

git branch -d xxx (强制删除git branch -D xxx)

合并

git merge

变基

git rebase

摘樱桃

git cherry-pick

10. Git分支合并的方式有哪些?

合并

git merge

变基

git rebase

摘樱桃

git cherry-pick