



链滴

# Git 使用分享

作者: [younghu](#)

原文链接: <https://ld246.com/article/1456378048088>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

# Git使用分享

## Git的由来

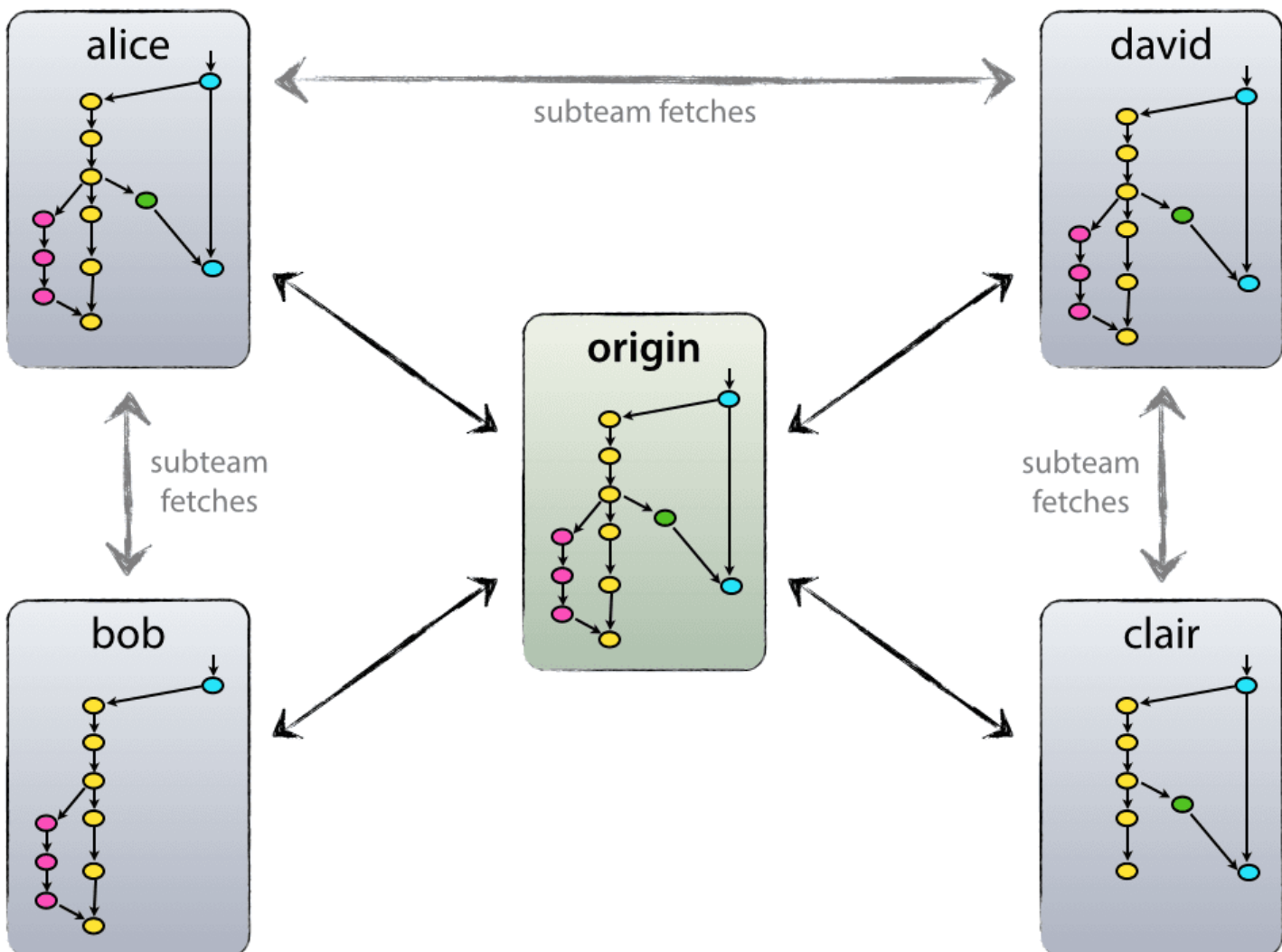
Git是由Linux之父Linus在2005年开发的一款跨平台的版本管理工具

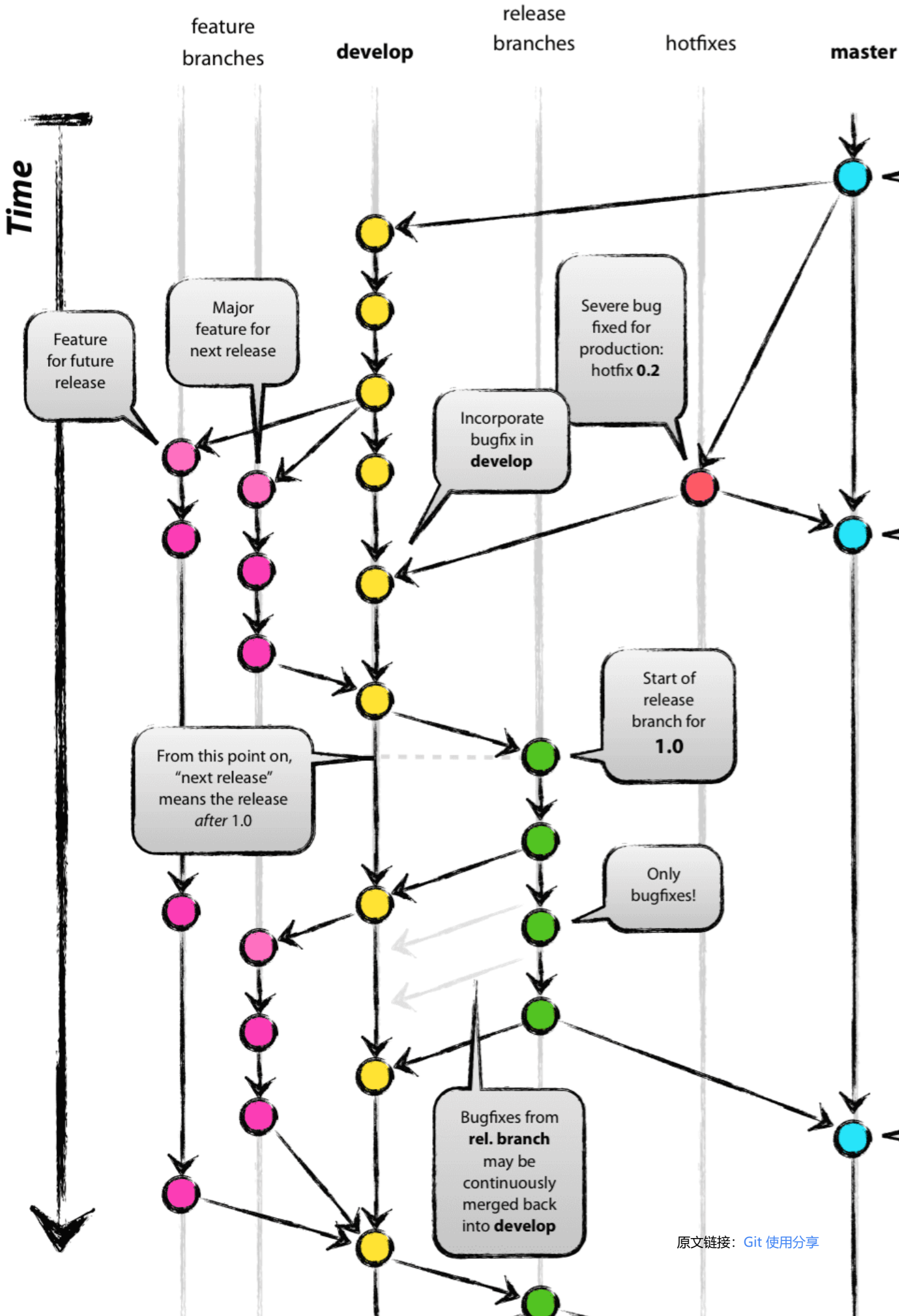
## Git和SVN的区别

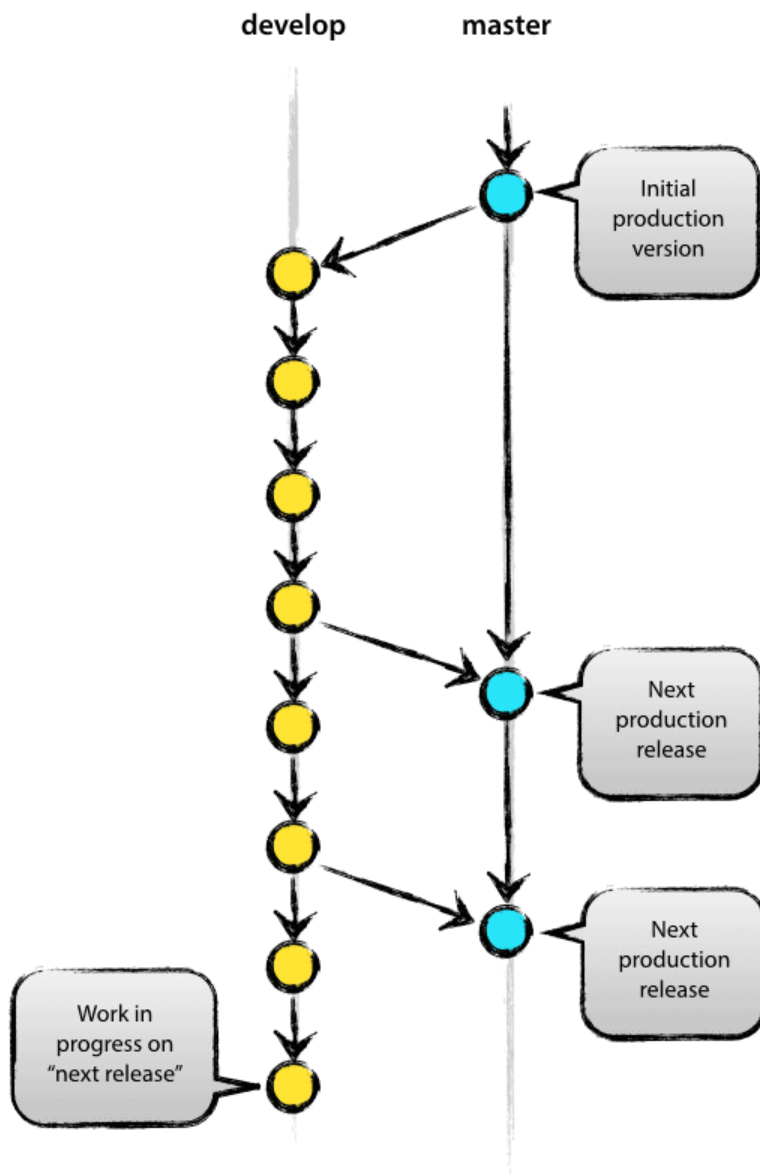
- Git是分布式的，所有提交到服务器上的版本在本地仓库都有备份，而SVN没有，SVN服务器一旦掉，版本会丢失
- 正因为Git有本地仓库，你的每一次提交都不需要直接提交到服务器，只需提交到本地，下班一起push到服务器
- 程序员能完全掌控版本服务器的文件，而不再需要找管理员帮忙。(大文件的删除)
- 现在越来越多的人都从SVN转Git了，github.com就是一个很好的证明
- gitignore可以针对项目忽略文件和文件夹，SVN的ignore很复杂的样子

## git的强大之处

协作流程







## Git的安装

Git官网和百度上都有所以操作系统的图文安装步骤，这里不再赘述

## Git的配置

git 项目的所有配置都在.git/config,git的全局配置在C:\Users\your-name\.gitconfig。

```
$ git --version //检查git版本
$ git config --global user.name zhoubotong //设置全局git commiter的名字，如果去掉--global
数，只会针对本项目有效
$ git config --global user.email zhoubotong@baofoo.com //设置全局git commiter的邮箱
```

现有的远程仓库一般都有两种协议，HTTP/HTTPS和GIT。

- http/https 基于用户名和密码的认证，每次版本操作需要输入用户名密码
- git 基于RSA加密的密钥对认证， 仅需第一次配置

# Git基本操作

## 创建一个本地版本库

```
$ git init gittest  
$ ls -af
```

## 进行第一次提交

```
$ //add HelloWorld.java  
$ git add . //将修改加入暂存区  
$ git commit -m "init" //提交到本地版本库  
$ git commit --amend //修改上次的提交说明
```

## 查看提交的日志

```
$ git log --pretty=oneline //查看日志简短说明 git log查看详细日志
```

## 备份你的版本库

```
$ cd ..  
$ git clone gittest demo //工作区复制一份出来
```

## 修改文件看个阶段的变化

```
$ //change file  
$ git status | git status -s // _M  
$ git add . //修改添加进暂存区  
$ git status | git status -s // M_  
$ git reset HEAD HelloWorld.java //从本地版本库拉出HelloWorld.java文件  
$ git checkout HelloWorld.java //从暂存区拉出HelloWorld.java文件  
$ git status | git status -s // __
```

## 火眼金睛看不同

```
$ git diff version2 version1 //对比版本库中两个版本的差异， version1为基准  
$ git diff //工作区和暂存区差异  
$ git diff --cached //暂存区和HEAD差异  
$ git diff HEAD //工作区和HEAD差异
```

## 为pull铺平道路

在本地有修改的情况下，去pull远程版本库是会失败的。要先stash（缓存修改到别处，清空工作区）下，再pull，再pop

```
$ git stash  
$ git pull  
$ git stash pop
```

## 后悔药

有的时候版本提交了，但是发现需要撤销提交的版本，例如提交了敏感信息等。这个时候我们可以使用 `git reset` 或者 `git revert` 命令来实现。这两个命令的差别在于 `git reset` 是删除你指定的版本之后的所有本（清除提交记录）而 `git revert` 是将某一次提交的内容删除，版本记录没有变化，revert后需要再次交。

```
$ git reset HEAD~{N} //N为整数，表示回退到当前提交的前N个提交点。
```

```
$ git revert {version} //version为版本号
```

## Git branch 操作

### 查看当前分支

```
$ git branch
```

### 创建新分支

```
$ git branch dev
```

### 切换分支

```
$ git checkout dev
```

### 分支合并到master

```
$ git checkout master
```

```
$ git merge dev
```

## Git 远程操作

### 查看远程库

```
$ git remote -v
```

### 直接从远程仓库clone

```
$ git clone url-address
```

### 本地有创建仓库，需要push到远端

```
$ git remote add origin url-address
```

```
$ git push -u origin master
```

### checkout远程的其他分支

```
$ git checkout -b dev origin/dev
```

## 本地分支和远程分支建立关系

```
$ git branch --set-upstream dev origin/dev
```

## 改变远端仓库地址

```
$ git remote set-url origin url-address
```