

Git介绍

1. Git 简介

1.1 版本控制

1.2 分布式 vs 集中式

2. Git 相关概念

2.1 三种状态：已提交、已修改、已暂存

2.2 三个阶段：工作区、暂存区、版本库

3. Git 常用命令

3.1 help

3.2 创建新仓库

3.3 添加新文件

3.4 提交版本

3.5 推送改动

3.6 更新本地代码至最新

4. DevCloud 配置代码仓库

4.1 安装并配置 Git 客户端

4.2 创建代码仓库

4.3 克隆/推送代码

参考学习链接：

本次课程要求「所有Lab项目」使用 Git 作为版本控制系统。

1. Git 简介

Git 是一个免费的开源分布式版本控制系统，旨在快速高效地处理从小型到大型项目的所有内容。

1.1 版本控制

版本控制是一种记录一个或若干文件内容变化，以便将来查阅特定版本修订情况的系统。

1.2 分布式 vs 集中式

- 集中式版本控制系统，版本库集中存放在中央服务器，而工作时使用自己的电脑，所以要先从中央服务器取得最新的版本，然后工作，完成后再把自己的更新推送给中央服务器。必须联网才能工作。

- 分布式版本控制系统则没有中央服务器，每个人的电脑上都是一个完整的版本库，工作时就不需要非得联网才行。如果需要协作，把修改推送给对方即可，较集中式而言安全度更高。

2. Git 相关概念

2.1 三种状态：已提交、已修改、已暂存

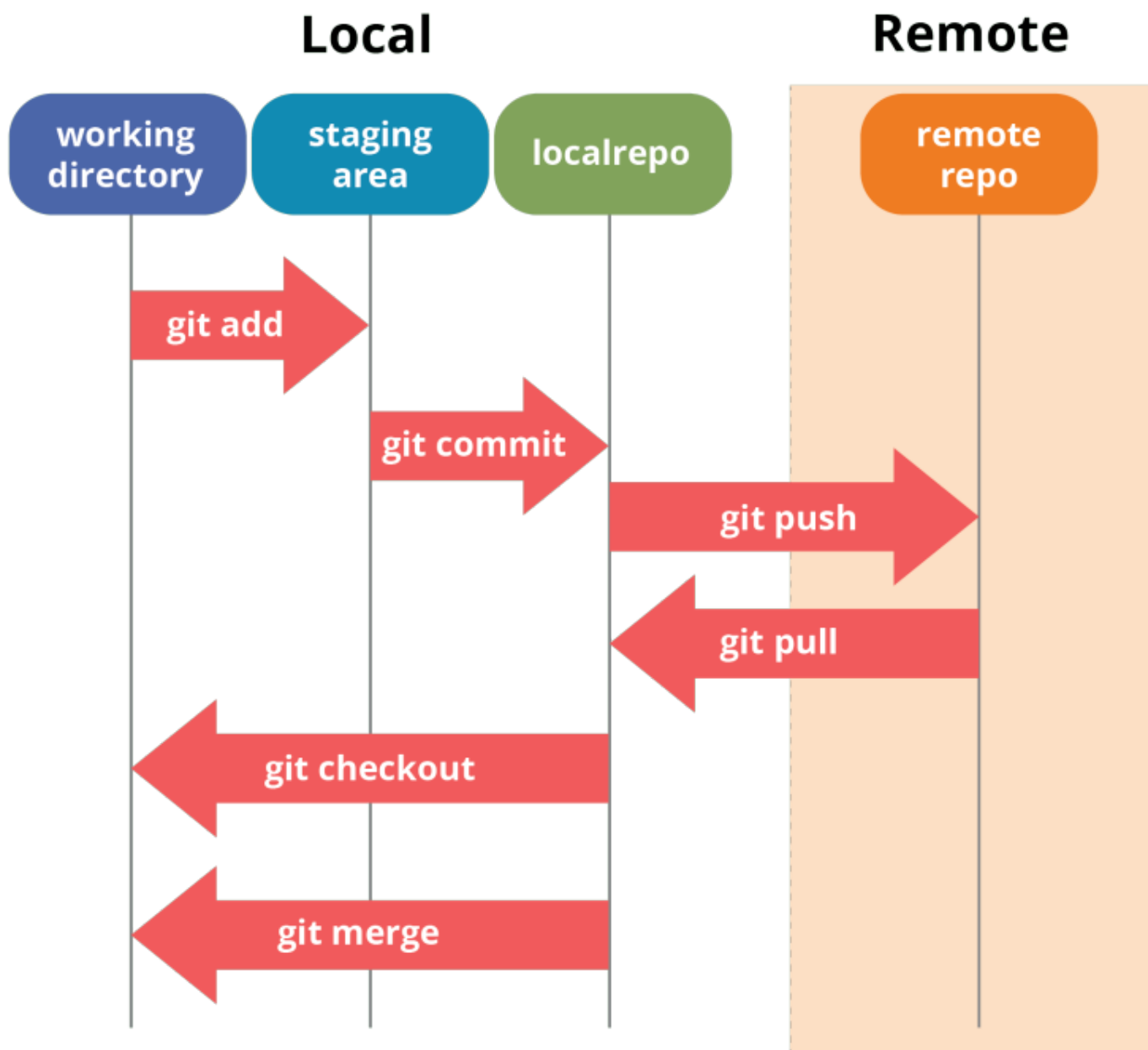
Git 有三种状态，你的文件可能处于其中之一：**已提交**、**已修改** 和 **已暂存**。

- **已提交 (Committed)** 表示数据已经安全地保存在本地数据库中。
- **已修改 (Modified)** 表示修改了文件，但还没保存到数据库中。
- **已暂存 (Staged)** 表示对一个已修改文件的当前版本做了标记，使之包含在下次提交的快照中。

2.2 三个阶段：工作区、暂存区、版本库

Git 的三种状态会让我们的 Git 项目拥有三个阶段：**工作区**、**暂存区** 以及 **版本库**。

- **工作区 (Working Directory)** 是对项目的某个版本独立提取出来的内容。这些从 Git 仓库的压缩数据库中提取出来的文件，放在磁盘上供你使用或修改。
- **暂存区 (Staging Area)** 是一个文件，保存了下次将要提交的文件列表信息，一般在 Git 仓库目录中。按照 Git 的术语叫做“索引”，不过一般说法还是叫“暂存区”。
- **版本库 (Repository)** 是 Git 用来保存项目的元数据和对象数据库的地方。这是 Git 中最重要的部分，从其它计算机克隆仓库时，复制的就是这里的数据。



3. Git 常用命令

3.1 help

如果有不熟悉的命令可以使用 `git help` 进行查询。注释示例为查询“`git config`”的用法。

```
1 # 第一种方法
2 git help <verb>    # git help config
3 # 第二种方法
4 git <verb> --help  # git config --help
```

3.2 创建新仓库

方法一：【本地新建】

在文件夹的根目录下，执行 `git init`，以创建新的 git 仓库。该文件夹即为 git 仓库。

```
1 git init
```

方法二：【本地新建】

指定文件夹名称，执行 `git init dir_name`，以创建新的 git 仓库。文件夹 dir_name 即为 git 仓库。

```
1 git init dir_name
```

方法三：【克隆远程仓库】

当你执行 'git clone' 命令的时候，默认配置下远程 Git 仓库中的每一个文件的每一个版本都将被拉取下来。

```
1 git clone <url>
```

Git 支持多种数据传输协议，可以使用 `https://` 协议，也可以使用 `git://` 协议或者使用 `SSH` 传输协议。

3.3 添加新文件

我们有一个仓库，但什么也没有，可以使用 `git add` 命令添加文件，也就是将文件提交至暂存区。

注意：file_name.txt 必须是存在的文件。

```
1 git add file_name.txt # 新添名为 "file_name.txt" 的文件
2 git add . # 新添当前目录下的所有文件
```

3.4 提交版本

现在我们已经添加了这些文件，使用命令 `git commit` 将文件从暂存区提交到本地仓库。

其中，-m 之后添加的字符串是对本次提交的注释说明。

```
1 git commit -m "Adding files"
```

如果不添加该注释的话，采用 `git commit` 会启动文本编辑器以便输入本次提交的说明。若文本编辑器里什么都不写，直接退出的话，会导致提交无效，即必须写注释才允许提交。

3.5 推送改动

使用 `git push` 将文件从本地仓库提交至远程仓库：

```
1 git push origin master # 可以把 master 换成你想要推送的任何分支
```

如果你还没有克隆现有仓库，并欲将你的仓库连接到某个远程服务器，你可以使用如下命令添加：

```
1 git remote add origin <server>
```

'git remote' 先将本地仓库与远端仓库建立一个链接，add 后面的单词其实就是你为远程仓库所起的名字，一般都是叫 origin，也可以叫其他的名字，<server> 则是远程仓库的真实地址。

3.6 更新本地代码至最新

```
1 git pull
```

当你的开发进行到一半，但是代码还不想进行提交，却需要同步远程代码时：若本地代码和远程代码没有冲突，可直接通过 `git pull` 获取最新代码；若产生冲突，直接 `git pull` 会拒绝覆盖当前的修改。此时需要先保存本地的代码，后 `git pull`，最后 pop 出本地代码：

```
1 git stash      # 保存当前代码
2 git pull
3 git stash pop # 将代码追加到最新的提交之后
```

还有更多的一些操作，如 Git 强大的分支管理，相信以大家的学习能力，可以在项目实践的过程中，边用边学，有问题的可以随时请教助教～

4. DevCloud 配置代码仓库

4.1 安装并配置 Git 客户端

- 通过[Git官网](#)下载安装包，并使用默认配置安装Git客户端到本地电脑。
- 运行Git Bash，Git Bash中输入以下命令行配置用户名和邮箱。

```
1 git config --global user.name "你的名字"
2 git config --global user.email "你的邮箱"
```

- 输入以下命令行，生成一对SSH密钥。生成的密钥通常保存在“~/.ssh/id_rsa.pub”中。

```
1 ssh-keygen -t rsa -C "<你的邮箱>"
```

- 输入以下命令可显示密钥内容。

```
1 cat ~/.ssh/id_rsa.pub
```

4.2 创建代码仓库

1. 进入已创建的项目，单击页面上方导航“服务 > 代码托管”。
2. 根据需要选择[普通新建](#)、[按模板新建](#)或[导入外部仓库](#)，按照页面提示输入仓库名称等基本信息，单击“确定”。

普通新建

代码仓库名称:

Test

归属项目:

2021软件工程_第X小组

描述:

请简要描述代码仓库

您最多还可以输入 500 个字符

选择gitignore:

请根据编程语言选择.gitignore

权限设置:

☒ 允许项目内人员访问仓库
 ☒ 允许生成README文件

☒ 自动创建代码检查任务 (免费)

确定

取消

3. 代码仓库创建成功后，页面自动跳转到仓库详情页，可查看仓库内的文件。

代码托管

CodeHub 基础版

0

租户仓库数

0.00M

租户存储空间

请输入关键字，按enter键搜索

Q

普通新建

设置我的SSH密钥

设置我的HTTPS密码

租户级IP白名单

仓库名称	归属项目	仓库URL	合并请求	仓库容量 (GIT LFS)	最近更新时间	操作
Test	2021软件工程_第X小组	SSH HTTPS	0	0.05M 0.00M	刚刚	

10

总条数: 1

< 1 >

4.3 克隆/推送代码

1. 在刚才的【普通新建】右侧，点击【设置我的SSH密钥】。
2. 单击“添加SSH密钥”，输入标题，以及Git配置时生成的密钥，单击“确定”。

添加SSH密钥

在密钥栏贴上您的公钥，公钥如何生成参考下面的帮助文档

* 标题:

macos

* 密钥:

ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQACpF6AkPsYShNb9E
sUpiQqSW1CKiZTLFektsRXkz53c4tSckWRRkQfD+2lahdhM9jc/
oXWoaQj1YsB/xg8ojJJG55EfXstimVmfmk+2488JnmKpL8P65z
UNRpeQruypq9BAnaodtjaPOF2eqQkqR54mBnlegLW5IJ2Dadj4

☒ 我已阅读并同意 [《隐私政策声明》](#) 和 [《软件开发服务使用声明》](#)

确定

取消

3. 返回代码仓库，单击“克隆/下载”，复制仓库地址。运行Git Bash，输入以下命令行，克隆云端仓库到本地。

```
1 git clone <仓库的SSH下载链接>
```

Test

代码健康度 未检查

+ 设置构建

克隆/下载

Fork: 0

CloudIDE

创建者: CandiceYu9 | 创建时间: 2021/03/02 20:27:51 | 最近更新时间: 2021/03/02 20:27:51

文件

分支

标签

合并请求

评审记录

成员列表

关联工作项

仓库统计

master

Q

Test

:

README.md

用SSH克隆

git@codehub.devcloud.cn-north-4.huaweicloud.com:2021rjg

复制

下载

zip tar.gz tar.bz2 tar

Git新手常见问题:

- Git客户端安装与配置
- 如何设置SSH密钥/HTTPS密码
- 客户端克隆/推送如何推送代码
- Git异常提示时, 如何处理

不再显示

4. 在本地完成代码的编辑后，在Git Bash中输入以下命令行，保存代码并推送到代码仓库。

```
1 git add .  
2 git commit -m "<你对本次提交备注的信息>"
```



```
3 git push origin master
```

5. 返回代码仓库，即可查看更新后的文件。

Test 代码健康度 未检查 + 设置构建 克隆/下载 Fork: 0 CloudIDE

创建者: CandiceYu9 | 创建时间: 2021/03/02 20:27:51 | 最近更新时间: 2021/03/02 20:42:21

[文件](#) [分支](#) [标签](#) [合并请求](#) [评审记录](#) [成员列表](#) [关联工作项](#) [仓库统计](#) [提交网络](#) [设置](#)

master

Test

☐ README.md

文件	最近一次提交	更新时间	创建者
<input type="checkbox"/> README.md	test	刚刚	CandiceYu8

参考学习链接：

1. Git官方文档：<https://git-scm.com/doc>
2. 廖雪峰的Git教程：<https://www.liaoxuefeng.com/wiki/896043488029600>