

Hacking Git - a Blocktree

陈嘉杰

2018 年 10 月

Git 是个啥子

大家知道 Git 是什么吗?

NAME

git - the stupid content tracker

SYNOPSIS

```
git [--version] [--help] [-C <path>] [-c <name>=<value>]
      [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
      [-p|--paginate|-P|--no-pager] [--no-replace-objects] [--bare]
      [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
      [--super-prefix=<path>]
      <command> [<args>]
```

Git 是个啥子

大家知道 Git 是什么吗?

NAME

git - the stupid content tracker

SYNOPSIS

```
git [--version] [--help] [-C <path>] [-c <name>=<value>]
    [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
    [-p|--paginate|-P|--no-pager] [--no-replace-objects] [--bare]
    [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
    [--super-prefix=<path>]
    <command> [<args>]
```

the stupid content tracker

Git 是个啥子

大家知道 Git 是什么吗?

NAME

git - the stupid content tracker

SYNOPSIS

```
git [--version] [--help] [-C <path>] [-c <name>=<value>]
    [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
    [-p|--paginate|-P|--no-pager] [--no-replace-objects] [--bare]
    [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
    [--super-prefix=<path>]
    <command> [<args>]
```

the stupid content tracker

a block tree

先开始动手

咱们边做边讲，先请大家安装一下 Git：

```
$ apt install -y git
```

先开始动手

咱们边做边讲，先请大家安装一下 Git：

```
$ apt install -y git
```

然后设置一下自己的名字和邮箱：

```
$ git config --global user.name "Jiajie Chen"
```

```
$ git config --global user.email "jiegec@qq.com"
```

先开始动手

咱们边做边讲，先请大家安装一下 Git：

```
$ apt install -y git
```

然后设置一下自己的名字和邮箱：

```
$ git config --global user.name "Jiajie Chen"
```

```
$ git config --global user.email "jiegec@qq.com"
```

然后新建一个文件夹，在里面建立一个空的 Git 仓库：

```
$ mkdir learn_git
```

```
$ cd learn_git
```

```
$ git init
```

Git 能做什么

Git 是一个“版本控制系统”，这意味着：

- ▶ 保存你文件修改的历史记录，你可以看到你以前在什么时候都做了什么更改
- ▶ 可以多人同时工作在同一个项目里，合理地把大家写的代码合并
- ▶ 学好了 Git，有利于软工等课程的学习
- ▶ 学会了 Git，你就可以参与到 GitHub 的广阔开发者社区中
- ▶ 学会了 Git，你才能有朝一日成为 Linux Kernel 开发者

不要怕，让我们一点一点来。最后自然也有给大家的一个挑战环节。

保存文件历史记录

假如要记录和妹子的出游记录（基于 NanoApe 真实事件改编）

```
$ echo "2018-10-04 Shenyang" >> log.txt
```

```
$ git status
```

```
On branch master
```

```
No commits yet
```

```
Untracked files:
```

```
(use "git add <file>..." to include in what will be committed)
```

```
    log.txt
```

```
nothing added to commit but untracked files present (use "git add" to track)
```

git add

Git 默认情况下不会把你的文件考虑进来。你必须告诉 Git 让它替你管理这个文件的历史记录。我们采用的命令是：

```
$ git add log.txt
```

当然了，它还有别的写法，我们经常会用

```
$ git add .
```

来把当前目录和所有子目录里文件都加进来。执行以后，是这个效果：

```
$ git status
```

```
On branch master
```

```
No commits yet
```

```
Changes to be committed:
```

```
(use "git rm --cached <file>..." to unstage)
```

```
new file:   log.txt
```

年轻人的第一个提交！

Git 有暂存区 (stage) 的概念，允许你在最终决定一个版本前先做一些修改。git add 把当前文件系统的更改添加到暂存区，而 git commit 则把暂存区的更改保存在历史记录中。

首先我们可以看看暂存区有什么内容：

```
$ git diff --cached
```

```
diff --git a/log.txt b/log.txt
new file mode 100644
index 0000000..c52d43f
--- /dev/null
+++ b/log.txt
@@ -0,0 +1 @@
+2018-10-04 Shenyang
```

这个格式叫做 Unified Diff 。以后大家还会经常看到。

年轻人的第一个提交！（续）

然后 NanoApe 检查了一下，时间和地点都没错，保存！

```
$git commit -m "Add shenyang tour"
```

```
[master (root-commit) ffd4912] Add shenyang tour  
1 file changed, 1 insertion(+)  
create mode 100644 log.txt
```

我们再看一下当前的状态：

```
$ git status
```

```
On branch master  
nothing to commit, working tree clean
```

妙啊，我的文件都已经保存进去了。那我想看我之前都做了什么，怎么办？

git log

有一个命令可以查看最近的提交: `git log`

`$ git log`

```
commit ffd4912fdd74c222787623d9263cb35771996611 (HEAD -> master)
Author: Jiajie Chen <jiegec@qq.com>
Date:   Thu Oct 25 09:54:58 2018 +0800

    Add shenyang tour
```

可以看到我在这个时候添加了一个提交, 作者是我, 时间是这个时间, 然后这个提交的描述是这个。

git log (续)

怎么看这个提交具体做了什么呢？

```
$ git log -p
```

```
commit ffd4912fdd74c222787623d9263cb35771996611 (HEAD -> master)
```

```
Author: Jiajie Chen <jiegec@qq.com>
```

```
Date: Thu Oct 25 09:54:58 2018 +0800
```

```
    Add shenyang tour
```

```
diff --git a/log.txt b/log.txt
```

```
new file mode 100644
```

```
index 0000000..c52d43f
```

```
--- /dev/null
```

```
+++ b/log.txt
```

```
@@ -0,0 +1 @@
```

```
+2018-10-04 Shenyang
```

一个提交太单调，我们加多一个

```
$ echo "2018-10-24 128 Days" >> log.txt
```

```
$ git commit -a -m "Add 1024 day"
```

```
$ git log -p
```

注：git commit -a -m "abc" 相当于 git add . && git commit -m "abc"

commit 7560697aa94262d1fec23a10300e77dd48a2c60e (HEAD -> master)

Author: Jiajie Chen <jiegec@qq.com>

Date: Thu Oct 25 10:07:03 2018 +0800

Add 1024 day

diff --git a/log.txt b/log.txt

index c52d43f..21b376d 100644

--- a/log.txt

+++ b/log.txt

@@ -1,2 @@

2018-10-04 Shenyang

+2018-10-24 128 Days

commit ffd4912fdd74c222787623d9263cb35771996611

Author: Jiajie Chen <jiegec@qq.com>

Date: Thu Oct 25 09:54:58 2018 +0800

Add shenyang tour

diff --git a/log.txt b/log.txt

new file mode 100644

index 0000000..c52d43f

--- /dev/null

+++ b/log.txt

@@ -0,0 +1 @@

啊，忘了一件事情

NanoApe: 唔，忘记提妹子的 ID 了，这不行。我得改一下

```
$ cat log.txt
```

```
2018-10-04 Shenyang
```

```
2018-10-24 128 Days with muvseea
```

```
$ git status
```

```
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   log.txt

no changes added to commit (use "git add" and/or "git commit -a")
```

啊，忘了一件事情（续）

我们可以看到刚刚的更改：

```
$ git diff
```

```
diff --git a/log.txt b/log.txt
index 21b376d..76656d2 100644
--- a/log.txt
+++ b/log.txt
@@ -1,2 +1,2 @@
   2018-10-04 Shenyang
-2018-10-24 128 Days
+2018-10-24 128 Days with muvseea
```

啊，忘了一件事情（续续）

我看刚才的提交不爽，可不可以更改我上一次的提交？可以：

```
$ git commit -a -amend -m "Add 128 days with muvseea"
```

```
$ git log
```

```
commit 182ffe291d1b3c041247b7ee1eb2a27ee61e9023 (HEAD -> master)
```

```
Author: Jiajie Chen <jiegec@qq.com>
```

```
Date: Thu Oct 25 10:07:03 2018 +0800
```

```
Add 128 days with muvseea
```

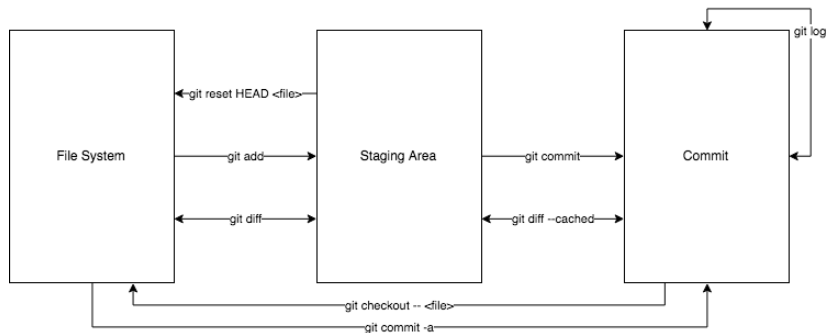
```
commit ffd4912fdd74c222787623d9263cb35771996611
```

```
Author: Jiajie Chen <jiegec@qq.com>
```

```
Date: Thu Oct 25 09:54:58 2018 +0800
```

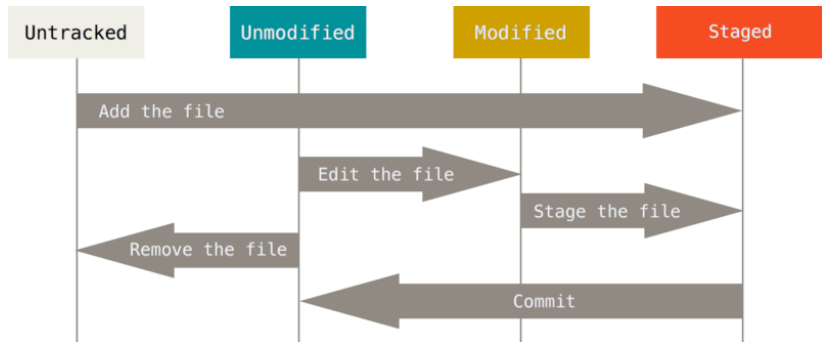
```
Add shenyang tour
```

谈谈暂存区



"git status" is your friend.

Pro Git 上的类似表述



发布虐狗版本!

```
$ git tag nue-gou-1.0
```

```
$ git log
```

```
commit 052a5268e8915a0c5156b22185b9e599a0fe101c (HEAD -> master, tag: nue-gou-1.0)
Author: Jiajie Chen <jiegec@qq.com>
Date: Thu Oct 25 10:07:03 2018 +0800

    Add 128 days with muvseea

commit ffd4912fdd74c222787623d9263cb35771996611
Author: Jiajie Chen <jiegec@qq.com>
Date: Thu Oct 25 09:54:58 2018 +0800

    Add shenyang tour
```

这样我们就创建了一个名为 `nue-gou-1.0` 的 tag。查看所有的 tag：

```
$ git tag
```

```
macbookair ➤ ... > Data > temp > learn_git ➤ git tag
nue-gou-1.0
```

虐狗回顾

假如 NanoApe 虐狗了 99 年的时候，想看一下当年的自己？（好浪漫啊）

```
$ git checkout -b nue-gou-1.0
```

```
Note: checking out 'nue-gou-1.0'.
```

```
You are in 'detached HEAD' state. You can look around, make experimental changes and commit them, and you can discard any commits you make in this state without impacting any branches by performing another checkout.
```

```
If you want to create a new branch to retain commits you create, you may do so (now or later) by using -b with the checkout command again. Example:
```

```
git checkout -b <new-branch-name>
```

```
HEAD is now at 052a526 Add 128 days with muvseaa
```

这样就可以回到当时的这个版本。实际上就是对某个 commit 进行了命名。

Git 协作

经过上面这部分，你应该已经学会如何用 Git 管理自己的代码了。但 Git 的强大很大程度在于，可以让很多人一起在一个项目上工作。

我们以清华 Git 为例。

`https://git.tsinghua.edu.cn`

欢迎使用清华大学代码托管服务

欢迎使用清华大学代码托管服务

Sign in

Sign in with

清华统一认证服务登录

☐ Remember me

使用清华 ID 登录即可。

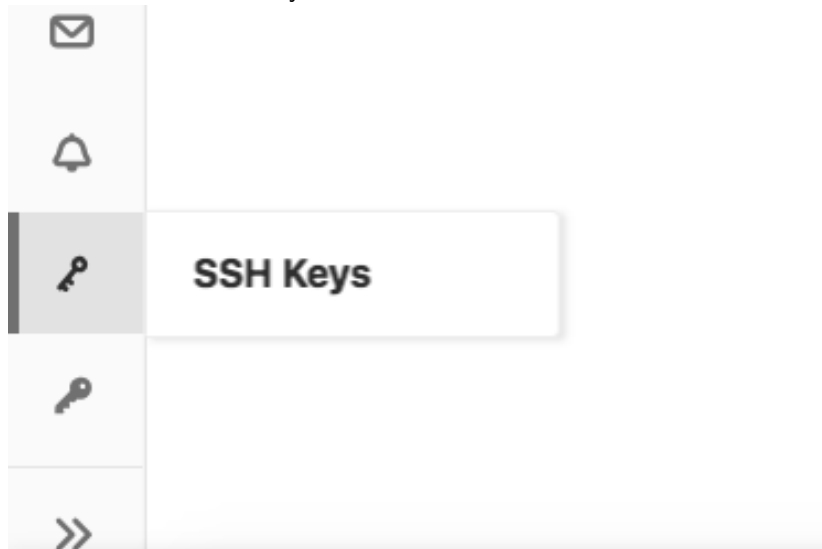
GitLab 配置

进去后，点击右上角的图标，点击 Settings



GitLab 配置 (续)

在左下角找到 SSH Keys



GitLab 配置 (续续)



如果大家还没有生成过，回到终端，生成一个自己的 SSH Key：

```
$ ssh-keygen
```

按照要求生成 SSH Key 。注意保存！拥有这个 Key 就可以对你拥有的 GitLab 仓库进行修改。

然后查看 `~/.ssh/id_rsa.pub` 文件的内容，复制到浏览器中，输入一个名称，最后添加到 GitLab 中。效果如图：

Your SSH keys (1)

 **jiegec@qq.com** 80:47:4d:fe:a8:d8:65:3f:fb:6d:5f:4a:50:cc:b0:8d created 6 months ago 
last used: 1 week ago

测试一下是否成功配置了：

```
$ ssh -T git@git.tsinghua.edu.cn
```

应该会输出 Welcome To GitLab 的信息。

创建仓库

找到 New Project , 进入创建仓库界面

`https://git.tsinghua.edu.cn/projects/new`

Blank project	Create from template	Import project		
<div><div>Project path</div><div>https://git.tsinghua.edu.cn/ chenjj17 ▼</div></div>			<div><div>Project name</div><div>learn_git</div></div>	
Want to house several dependent projects under the same namespace? Create a group				
<div><div>Project description (optional)</div><div>Description format</div></div>				
<div><div>Visibility Level ?</div><div><div><input type="radio"/> Private</div><div>Project access must be granted explicitly to each user.</div></div><div><div><input checked="" type="radio"/> Internal</div><div>The project can be accessed by any logged in user.</div></div><div><div><input type="radio"/> Public</div><div>Public visibility has been restricted by the administrator.</div></div></div>				
<div>Create project</div>				<div>Cancel</div>

点击 Create Project 。

把刚才的仓库推到 GitLab 上

```
$ git remote add origin  
git@git.tsinghua.edu.cn:chenjj17/learn_git.git  
$ git push origin master
```

```
Enumerating objects: 6, done.  
Counting objects: 100% (6/6), done.  
Delta compression using up to 4 threads  
Compressing objects: 100% (3/3), done.  
Writing objects: 100% (6/6), 501 bytes | 250.00 KiB/s, done.  
Total 6 (delta 0), reused 0 (delta 0)  
To git.tsinghua.edu.cn:chenjj17/learn_git.git  
* [new branch]      master -> master
```