

Phabricator Code Review 操作手册

🕒 Updated 58 Days Ago

👤 All Users

本文作者为系统组大佬 [@Jayce](#)，授权转载，只调整部分样式。

更新日志

2018-08-30:

- 更新代码审查申请备注信息的说明及对应截图;
- 新增arc diff信息提交后的截图;

Tags

None

Subscribers

[shenjun](#), [Jayce](#)

概述

Phabricator支持两条不同的代码审查 workflow，一条叫做“reivew”（审查），是在推送前进行的；另一条叫做“audit”（审计），是在推送后进行的。本文主要介绍review。

Phabricator的代码审查功能目前没有图形化的客户端，只能通过命令行进行操作，使用的工具为 [Arcanist](#)。

环境的准备

注：本步骤为**被审查者**需要进行的操作，其他人可以跳过。

Windows用户请参考 [Arcanist环境的准备\(for Windows\)](#)

macOS用户请参考 [Arcanist环境的准备\(for macOS\)](#)

环境的配置

配置工程

注：本步骤为**仓库管理员**要进行的操作，其他人只需要拉取最新代码即可。

在工程根目录下创建 `.arccconfig` 文件，并写入以下内容（注意修改 `phabricator.uri` 的值为对应地址）：

```
{
  "phabricator.uri" :
  "https://phabricator.example.com/",
  "history.immutable" : true
}
```

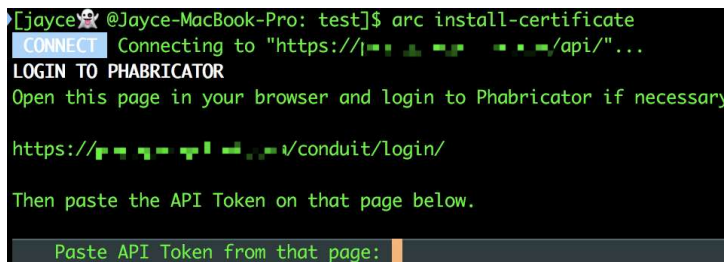
然后将其提交并推送至远程仓库。

安装Arcanist证书

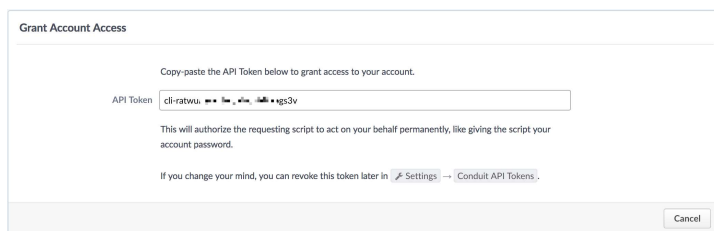
注：本步骤为被审查者需要进行的操作，其他人可以跳过。

使用命令行工具执行

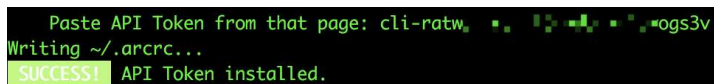
```
cd PROJECT_PATH/ #切换到你的项目目录下，注意替换
PROJECT_PATH
arc install-certificate #安装Arcanist证书
```



使用浏览器访问提示的地址获取API Token（需要登录）



复制API Token，粘贴至 Paste API Token from that page: 处



证书安装完成。

提交代码审查申请

注：本步骤为代码审查的关键步骤之一，请大家仔细阅读。

代码审查发生在提交合并之后，推送之前。之前的git工作流保持不变，只在完成代码提交、分支合并后，使用Arcanist提交

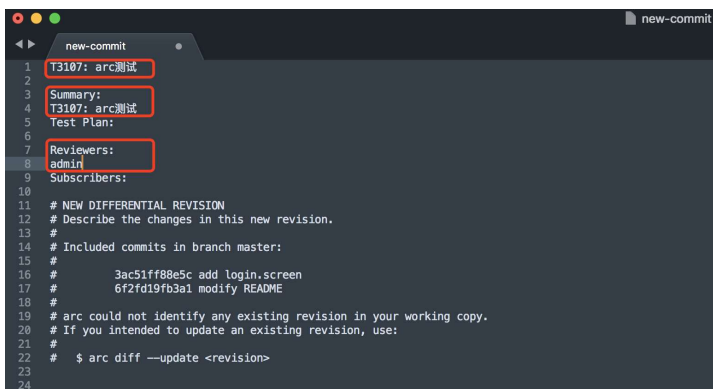
代码审查申请。此时**被审查者**无法再使用 `git push` 推送至远程仓库。强行推送时，会看到如下的报错：

```
jlayce@Jayce-MacBook-Pro: test$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 281 bytes | 281.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0)
remote: +-----+
remote: |          * * * PUSH REJECTED BY EVIL DRAGON BUREAUCRATS * * *          |
remote: +-----+
remote:
remote:      \
remote:       ^   ^
remote:    /---\ /--\
remote:   /     V   //
remote:  @_@'  \___//         *-.-*-
remote: 0/0/0/  \__//        |
remote: 0/0/0/0/|  \\_____|
remote: 0/0/0/0/_ _ \| C ___||
remote: 0/0/0/0/0/'-_- -_- ) ; __|
remote:           }-}             *_-.l----~
remote: \_____(_oo)            *.  }                {
remote: C (---)                  --V--
remote: //_\\ \ DENIED! ///-----<
remote: //_\\              ///-----{^ -----~
remote:
remote:
remote: This push was rejected by Herald push rule H4.
remote: Change: commit/
remote: Rule: CodeReview4Test
remote: Reason: This repo enable Code Review for you, please refer to: https://bit.ly/2PixABv
Transcript: https://[REDACTED]/herald/transcript/134577
```

完成代码提交、分支合并后（注意保持 Git 当前工作区为 Clean 状态），使用命令行工具，切换到工程目录下执行：

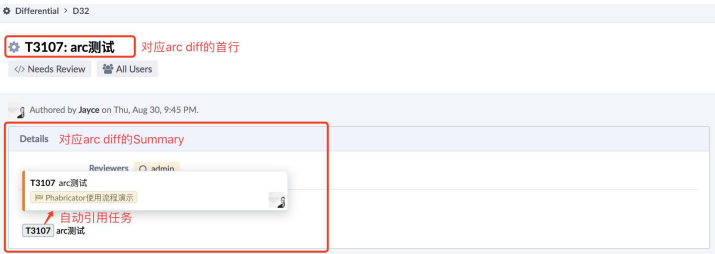
```
cd PROJECT_PATH/ #注意替换成真实的工程目录
arc diff #执行此命令时注意保持 Git 当前工作区为
Clean 状态
```

此时会自动打开之前定义的文本编辑器



注:

- 1.首行为代码审查申请的标题，请简明扼要的写明本次申请所涉及的内容;
- 2.Summary项可以详细介绍本次申请所涉及的内容，使用**"Tn"**会自动引用对应的任务;
- 3.Reviewers为必填项，务必准确无误地填入**审查者**的Phabricator的用户名;支持多人审查，用户名用";"隔开;此处的admin仅做演示，实际使用时，请修改为自己的审查人。



填写完成后，保存，并关闭文本编辑器（Arcanist会等待文本编辑器程序的退出，因而此处必须**完全退出**程序）。

```
[jayce@Jayce-MacBook-Pro: test]$ arc diff
Linting...
No lint engine configured for this project.
Running unit tests...
No unit test engine is configured for this project.
SKIP STAGING No staging area is configured for this repository.
Created a new Differential revision:
Revision URI: https://[redacted]/D32

Included changes:
M README
A webRoot/login.screen
```

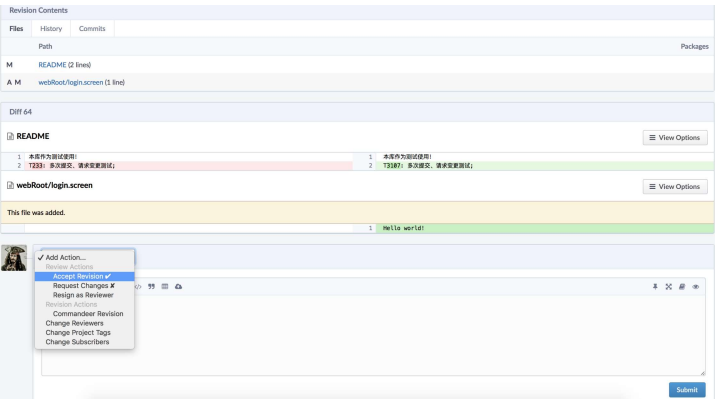
此时已完成代码审查的提交的，Phabricator会自动发送邮件通知**审查者**进行代码审查，也可以手动复制Revision URI发送给审查者。

注意：此时需记下代码审查的ID，此处为“**D32**”，后面会用到。

代码审查

注：本步骤为**审查者**需要进行的操作，其他人可以跳过。

审查者收到代码审查申请后，直接访问对应的链接打开代码审查界面。



在这里可以看到修订的文件列表、文件改动的具体内容等，审查者完成代码审查后，可以选择“Accept Revision”（接受该修订版）、“Request Changes”（请求变更）、“Change Reviewers”（改变审查者）等。此外，还可以添加备注信息，用于记录信息或者提醒被审查者做代码修改。

这里选择接收该修订版，被审查者将收到审查通过的通知邮件。

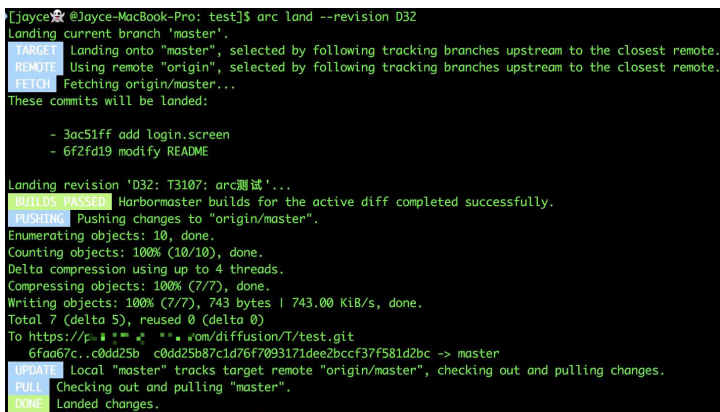
推送代码

注：本步骤为被审查者需要进行的操作，其他人可以跳过。

1. 审查通过

被审查者收到审查通过的通知邮件后，使用命令行工具执行：

```
cd PROJECT_PATH/ #切换到工程目录下，注意替换成真实的工程目录
arc land --revision D32 #提交代码审查申请时自动分配的ID
```



```
[j@jays ~]$ arc land --revision D32
Landing current branch 'master'.
TARGET: Landing onto "master", selected by following tracking branches upstream to the closest remote.
REMOTE: Using remote "origin", selected by following tracking branches upstream to the closest remote.
FETCH: Fetching origin/master...
These commits will be landed:
- 3ac51ff add login.screen
- 6f2fd19 modify README

Landing revision 'D32: T3107: arc测试'...
BUILDING: Harbormaster builds for the active diff completed successfully.
PUSHING: Pushing changes to "origin/master".
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 4 threads.
Compressing objects: 100% (7/7), done.
Writing objects: 100% (7/7), 743 bytes | 743.00 KiB/s, done.
Total 7 (delta 5), reused 0 (delta 0)
To https://p...:
 6faa67c..c0dd25b c0dd25b87c1d76f7093171dee2bccf37f581d2bc -> master
UPDATE: Local "master" tracks target remote "origin/master", checking out and pulling changes.
PULL: Checking out and pulling "master".
DONE: Landed changes.
```

代码推送成功。

2. 审查未通过

如果收到的是请求变更的邮件，请按照审查者的说明重新修改代码，并再次执行 `arc diff`，同初次提交代码审查申请时相同，此时会自动打开之前定义的文本编辑器（与之前的文本内容略有不同），填写完成后，保存，退出编辑器程序，以更新代码审查申请。

该过程可能需要重复多次，直到审查者接受了变更，然后执行 `arc land --revision DXX` 进行代码推送。

Last Author [shenjun](#)

Last Edited Jun 3 2019, 11:06 AM

Apr 30 2019, 10:51 AM

shenjun added subscribers: **Jayce**, **shenjun**.


May 3 2019, 7:10 PM





May 3 2019, 10:00 PM





May 4 2019, 10:54 AM

May 4 2019, 11:08 AM

Jun 3 2019, 11:06 AM

B I T 

 </> “ ”  

Submit

PREVIEW