解决LeanCLoud定时唤醒失败的流控问题 - 优源博客 | YouYuan Blog

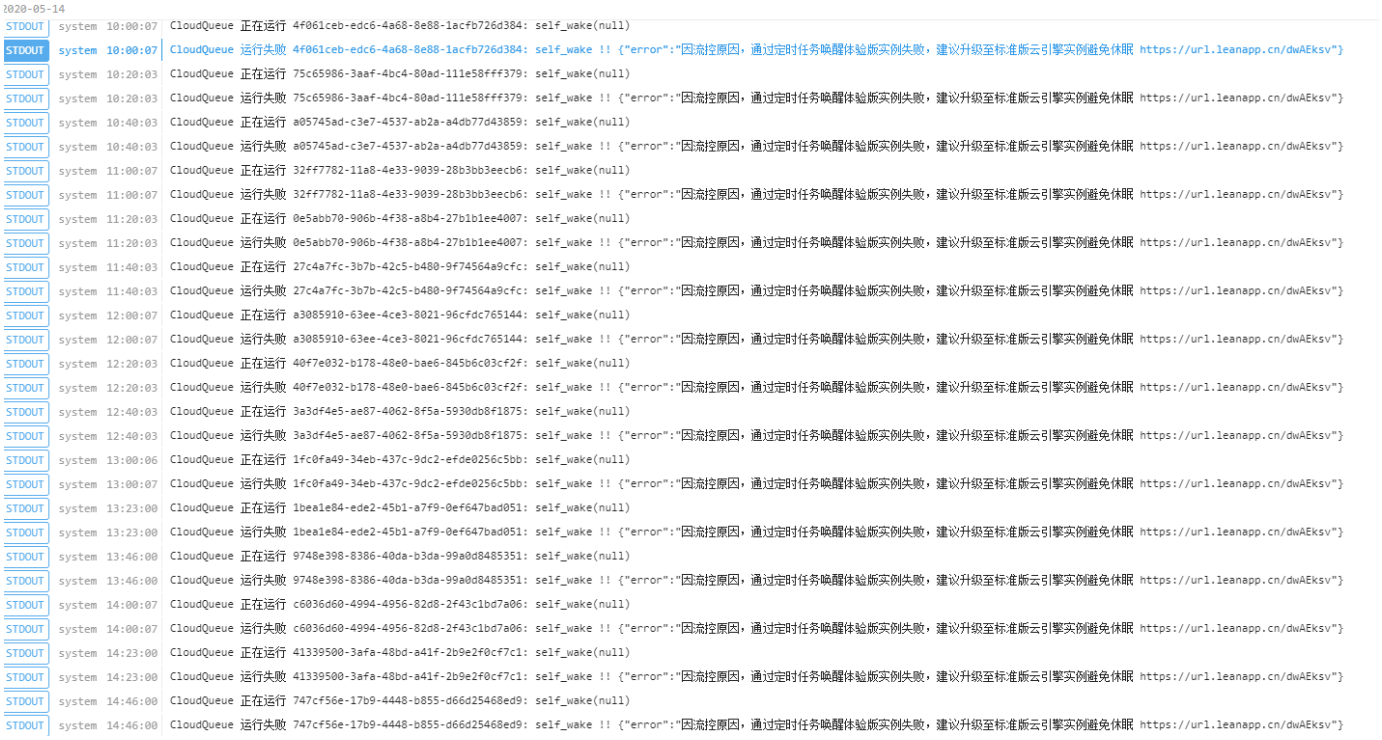
> 本文转载自小康博客-《优雅解决LeanCloud流控问题》

1.问题由来

> 最近好多人在使用LeanCloud时遇到了`"因流控原因，通过定时任务唤醒体验版实例失败，建议升级至标准版云引擎实例避免休眠"`提示。

看到官方所说：

既然是同一时刻，那么是不是意味着只要搓开时间就可以了呢?



原文作者便调整时间尝试了几天，第一天还好，但以后便又出现了流控导致的失败。

因此，调整时间避免的方案治标不治本。还需另寻他法。

2.目前解决思路

- 修改定时任务的唤醒时间

  这个方案在上边我已经介绍过了，治标不治本。这里我并不推荐。
- 在博客多加入一条请求。

  也就说每一次访问博客时，将 leancloud 唤醒。这种方法可以，这也是我最先想到的，但无疑，这会在一定程度上拖慢博客加载速度。
- 第二个方案的变种

  为什么说是变种。因为也是加一个请求，只不过不会在你博客加，那么加在哪里呢？请继续往后看。

3.优先解决方案

解决方案其实真的还蛮多的。因为方案很多，我也不可能每种方案都写一篇详细的小白教程，因此发现或者想到新方案后，我会将思路分享给大家。至于具体如何操作，请自己动手，详细过程不可能再会给出教程 (*特别繁琐除外*)，本文只会给出一些关键性的代码（脚本），以及代码（脚本）如何使用。

此篇文章详细介绍方案一的做法，其他方案为 2020 年 05 月 18 日后补方案

LeanCloud 的机器唤醒其实还有一种方式。详情请看休眠策略。

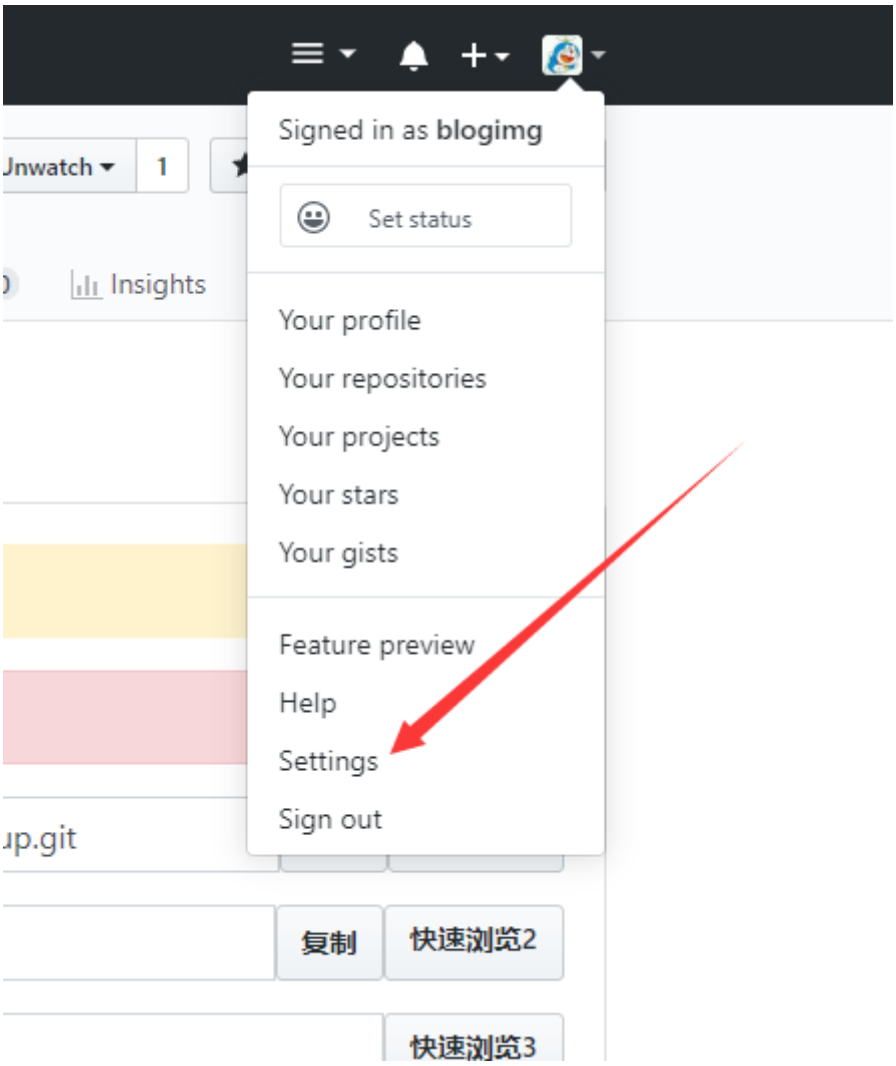
> - 如果应用最近一段时间（半小时）没有任何外部请求，则休眠。
> - 休眠后如果有新的外部请求实例则马上启动。访问者的体验是第一个请求响应时间是 5 ~ 30 秒（视实例启动时间而定），后续访问响应速度恢复正常。

> - 强制休眠：如果最近 24 小时内累计运行超过 18 小时，则强制休眠。此时新的请求会收到 503 的错误响应码，该错误可在 **云引擎 > 统计** 中查看。

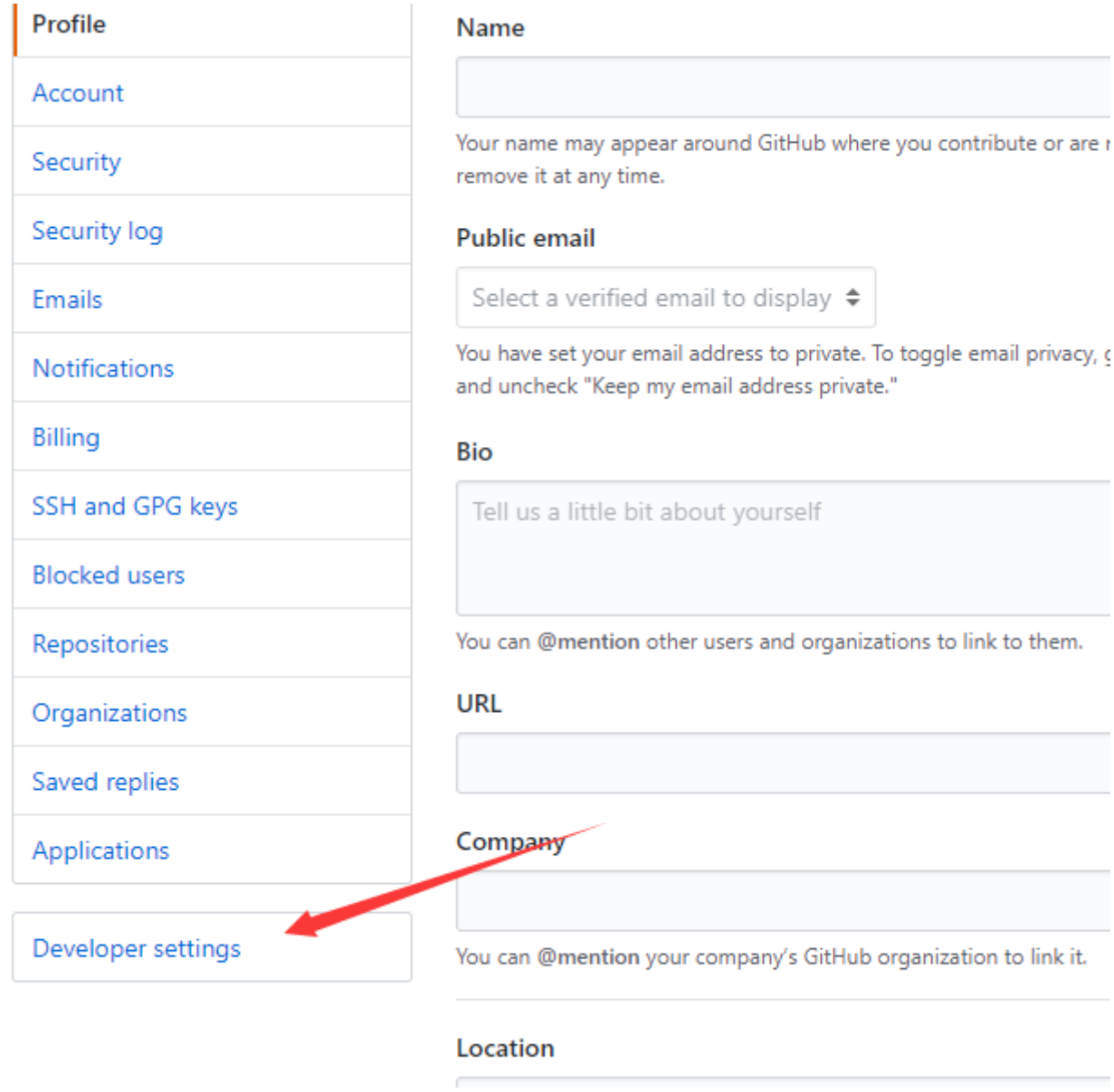
那么我们只要每三十分钟之内在外部访问一次不就可以解决了么？

于是我查看了一下 `valine-admin` 的唤醒源代码，自唤醒云函数也是这样实现的。于是便继续开始白嫖。

> **这里我使用的是 GitHub+Actions。是不是很熟悉，自动部署也是这个方案呢。**

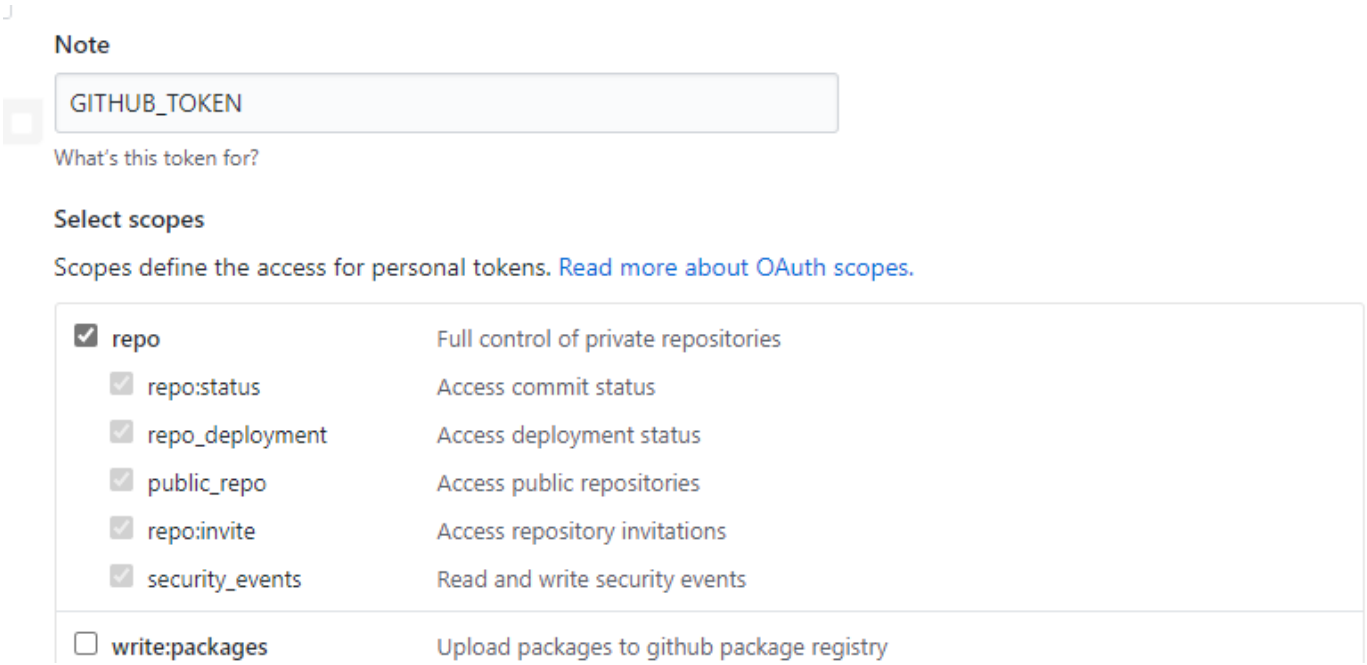


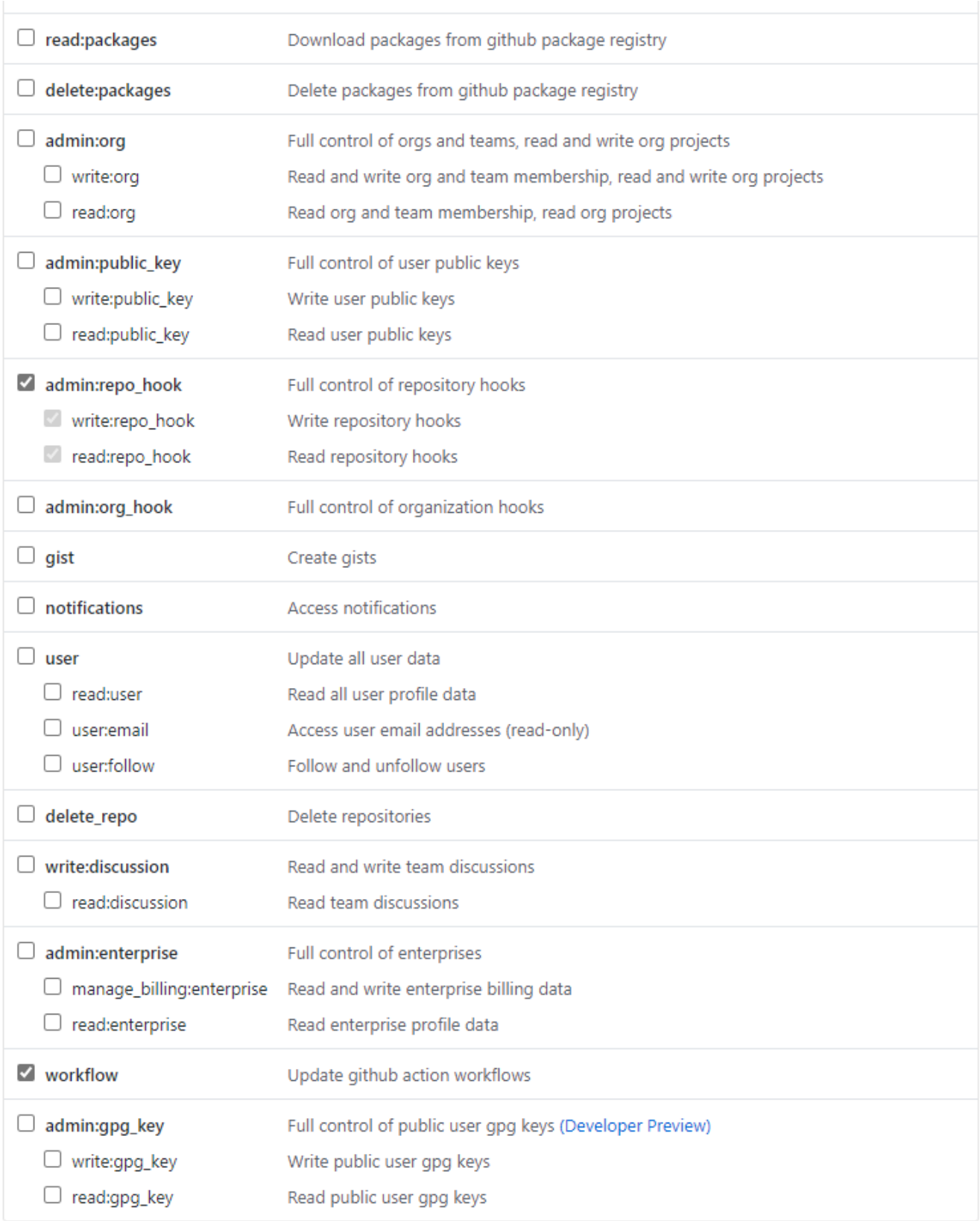
（1） 鼠标放在右上角，选择 setting



（2） 点击 `Developer settings`。

（3） 选择 `Personal access tokens`，添加一个新的 TOKEN。 这个 TOKEN 主要使用来启动 actions 和上传结果用的。





设置名字为 `GITHUB_TOKEN`，然后勾选 repo，admin:repo_hook，workflow 等选项，最后点击 Generate token 即可。

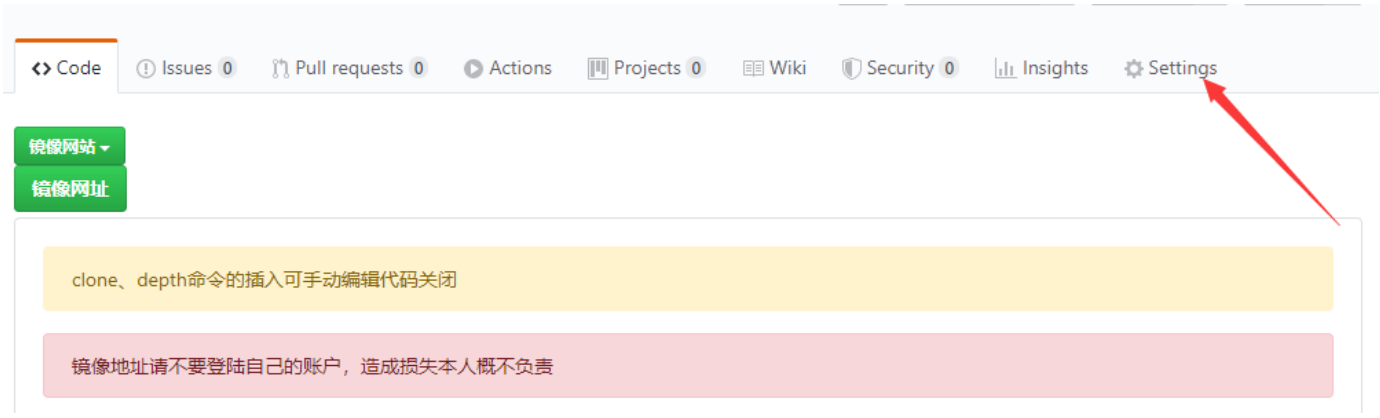
名字请务必使用`GITHUB_TOKEN`。

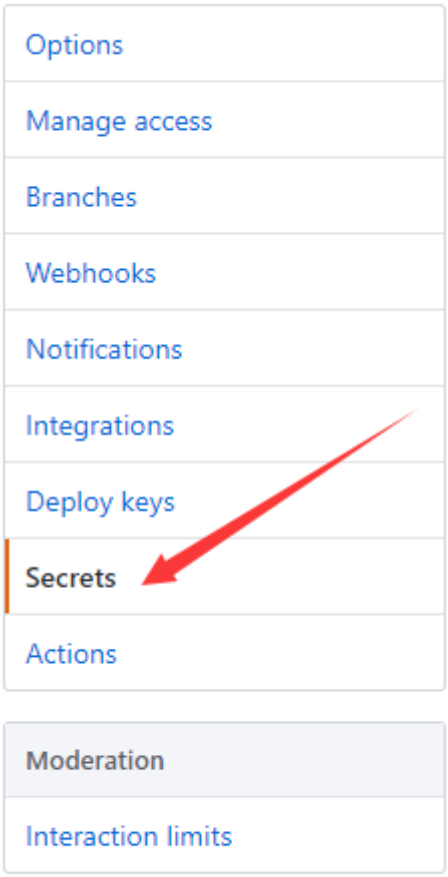
（4） 接下来 FORK 项目。

地址：https://github.com/blogimg/WakeLeanCloud

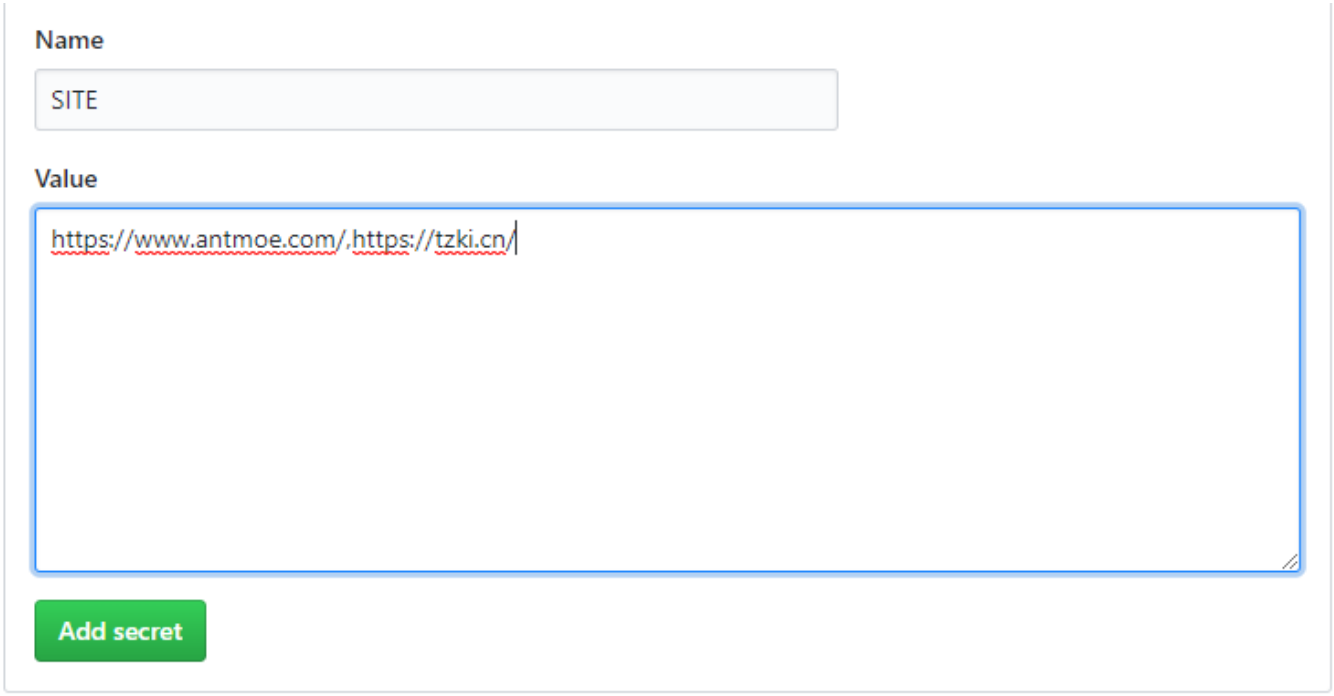
如果觉得好用可以点个赞哦！

（5） 成功 FORK 后，进入项目的设置。添加你的 leancloud 的后台地址（也就是评论管理的后台地址）





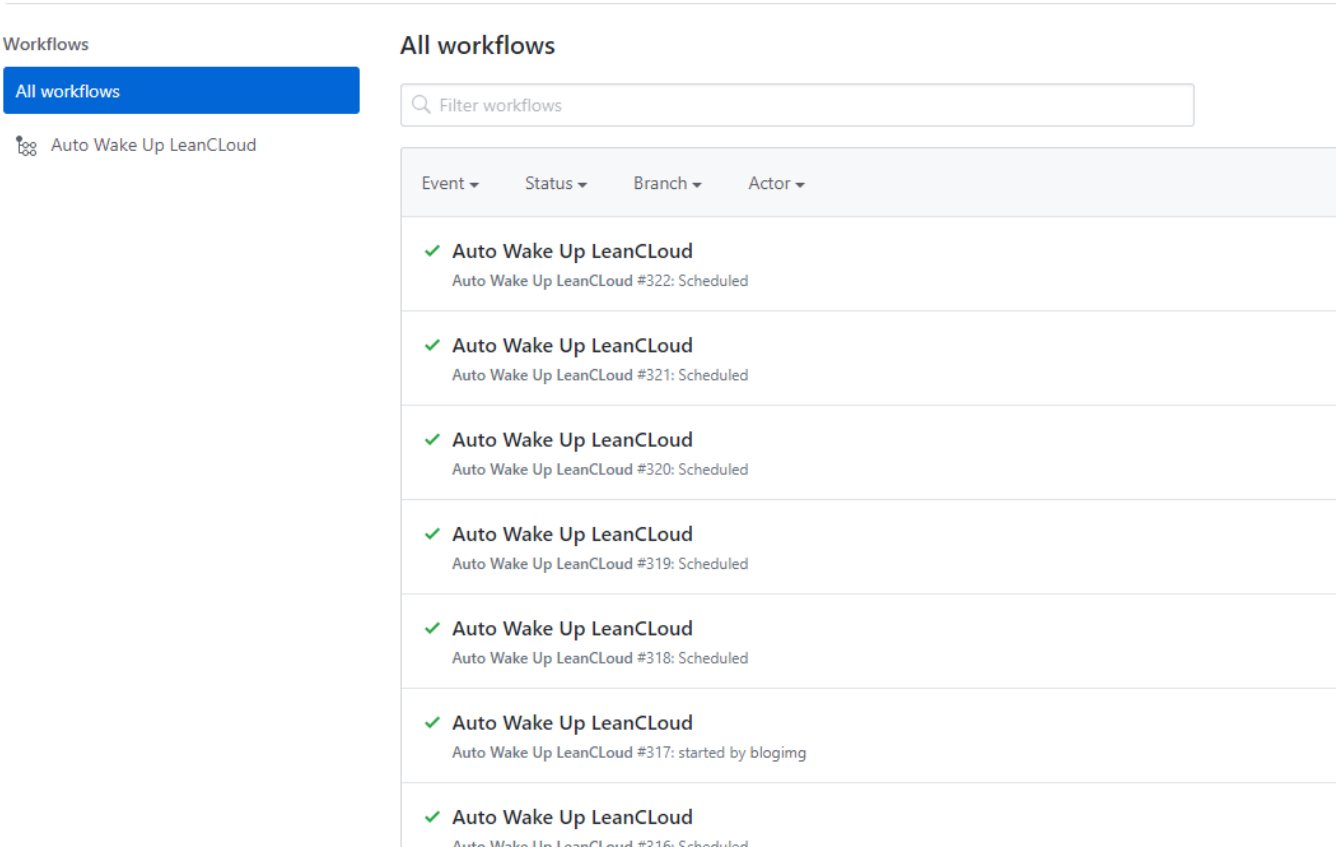
选择 Secrets，添加你的地址



其中 Name 的名字必须为 `SITE`，Value 可以是多个地址，用英文逗号分隔。不要中中文逗号，不要用中文逗号，不要用中文逗号

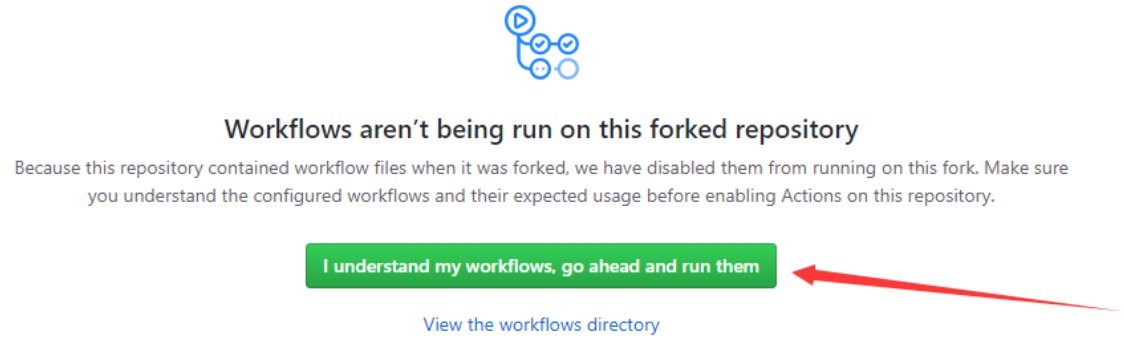
（6） 接下来对自己的项目点个 star 就能启动了，启动后请切换到当前项目的 actions（上方标签，不是左侧标签），看看是否运行成功。

- 成功
  那么你就可以关掉了，默认是每天 8:00-24:00 时每 20 分钟运行一次。(GitHub 时间稍有延迟，大概时 2-5 分钟。)



- 失败

如果你的 GitHub 从来没有用过 actions，那么需要先 “了解”。方法很简单，点击绿色的按钮即可。



其他问题：请认真看本教程并且思考为什么这么做，你就能想到你是哪里出了问题。

> 自己点自己的项目是手动执行一次 actions。是为了测试才设计这个功能的哦！
>
> 并不是不点星这个 actions 就不会运行。

（7） 最后，如果觉得好用，请给我点个 star 哦！

4.其他解决方案

这里为 2020-05-18 之后补充的其他方案。

方案二

利用国内的云函数，自己写一个脚本。然后定时监控即可。

或者宝塔、自己服务器的定时任务都是可以的。

方案三

`cloudflare` 的 `Workers` 可以在线定义脚本，通过链接即可触发脚本。

因此定义好自己的脚本后，通过监控即可触发来实现唤醒 LeanCloud

javascript：

```
addEventListener('fetch', (event) => {
    return event.respondWith(handleRequest(event.request));
})
```

```
const handleRequest = async (request) => {
    const render = (body) => {
        return new Response(`
<!doctype html>
<html>
<head>
<meta charset="utf-8" />
<meta name="viewport" content="width=device-width,initial-scale=1" />
<title>唤醒你的LeanCloud</title>
</head>
<body>${body}</body>
</html>`.trim(), {
            status: 200,
            headers: {
                'Content-Type': 'text/html; charset=utf-8'
            }
        });
    }
    var date = new Date();
    var hour = date.getHours();
    var minutes = date.getMinutes;
    // 事件控制，因此事件采用utc时间，因此需要手动-8
    if (hour >= 0 && hour <= 15) {
        // 列表里添加你的评论管理后台地址
        const Urls = ['https://www.antmoe.com/','https://www.tzki.cn/']
        var result = ''
        for(var i=0;i<Urls.length;i++){
            const response = await fetch(Urls[i]);
            const html = await response.status;
            result+=Urls[i]+ "状态："+html+'<br />'
        }
        return render(`
${result}<br />
`);
    }else{
        return render(`当前是休息时间哦！<br />`);
    }
}
```

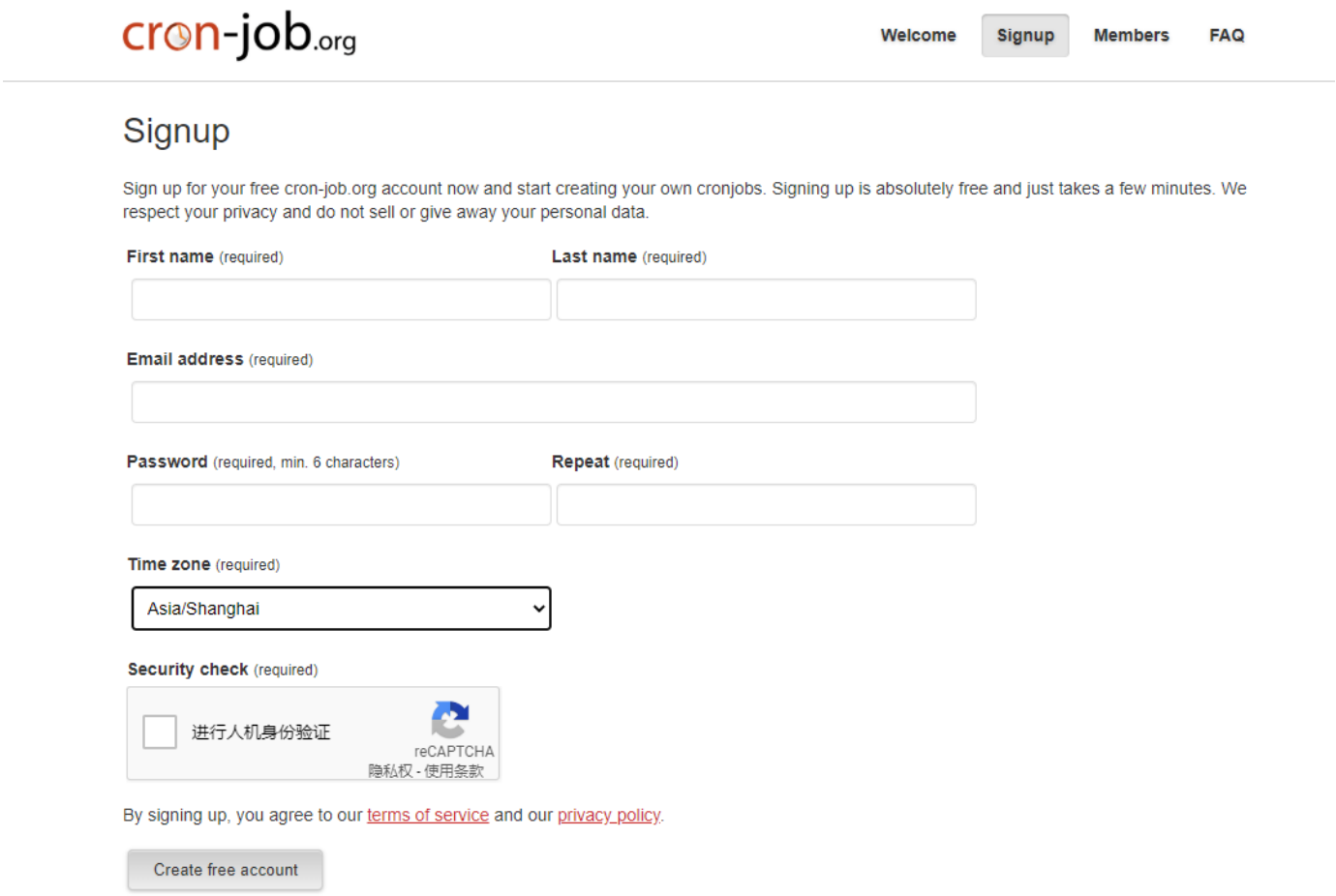
监控平台：https://uptimerobot.com/ 监控地址就是 Workers 的地址。监控频率看你自己。

另外网友 track13 评论到：其实只外部要唤醒一次就可以，之后都可以交给 leancloud 的定时任务。

方案四

通过 `cron-job` 平台进行监控。注册地址：https://cron-job.org/en/signup/

1. 注册



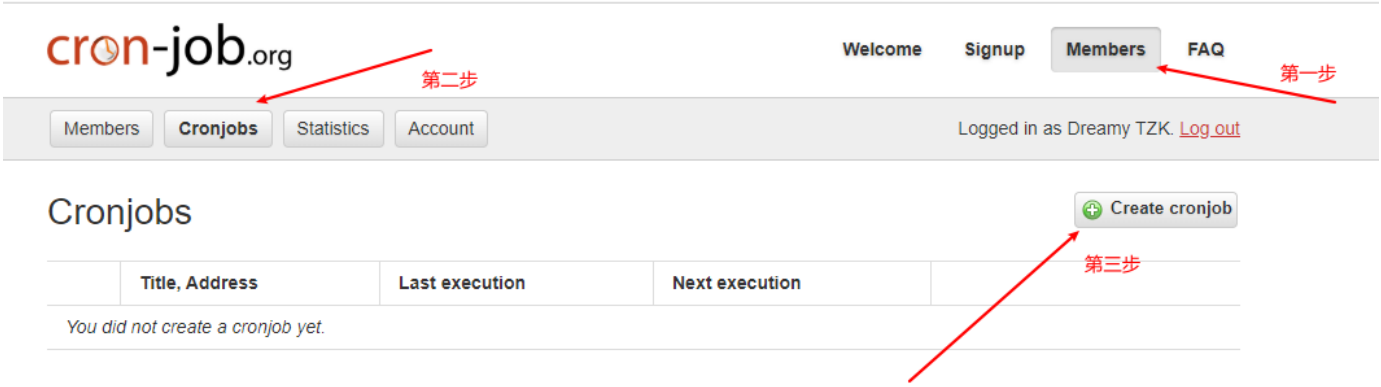
时区请选择 `Asia/Shanghai`，否则请手动下边的操作请手动换算时间。另外关于最下边的谷歌验证如果出不来，请采用特殊方式。这里不过多介绍。

1. 登录账号

首先去邮箱激活一下账号哦！*邮件可能在垃圾箱哦！*

1. 添加任务

登陆之后依次点击 `Members`,`cronjobs`,`Create cronjob`

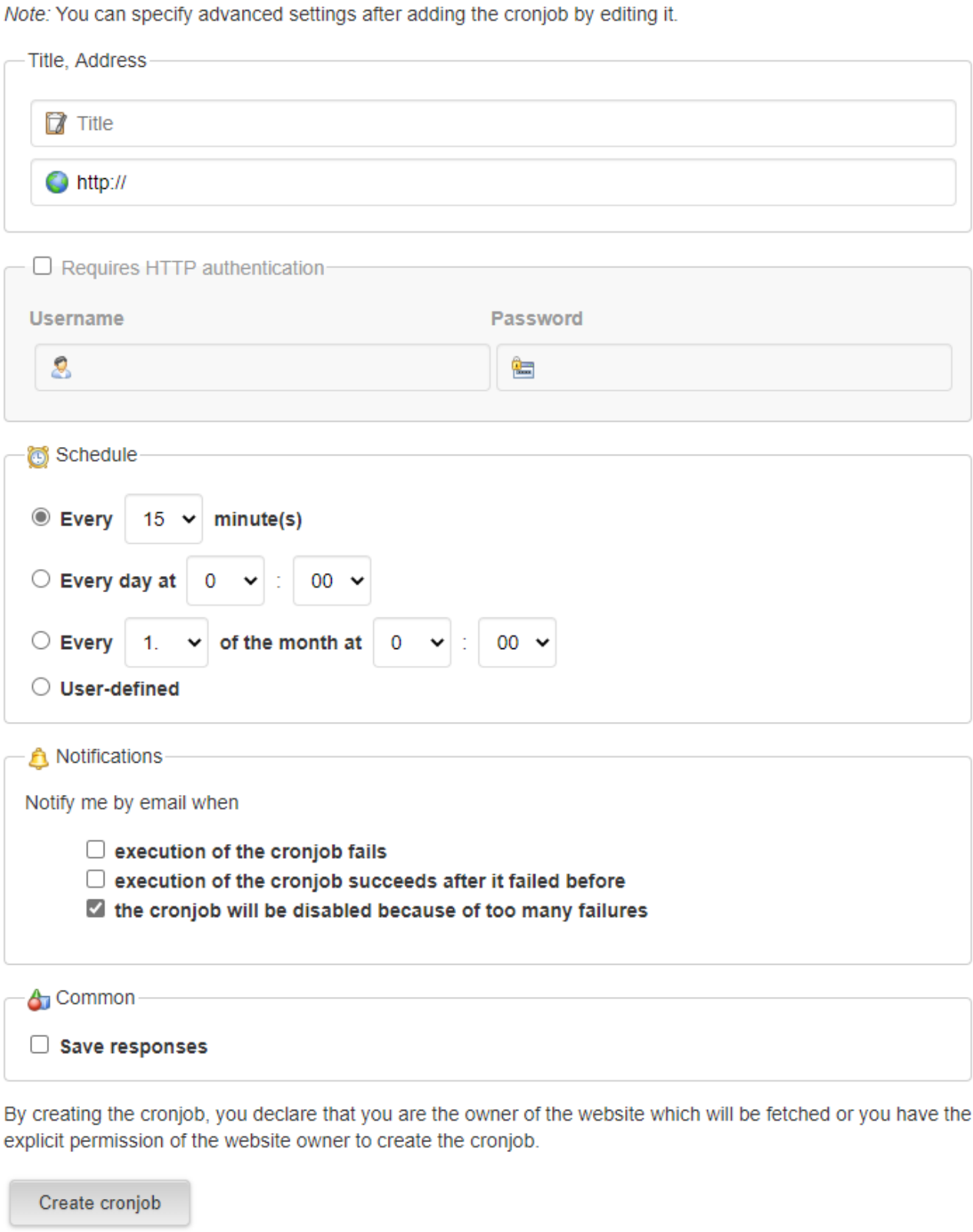


Cronjobs

Create cronjob

第三步

	Title, Address	Last execution	Next execution	
You did not create a cronjob yet.				

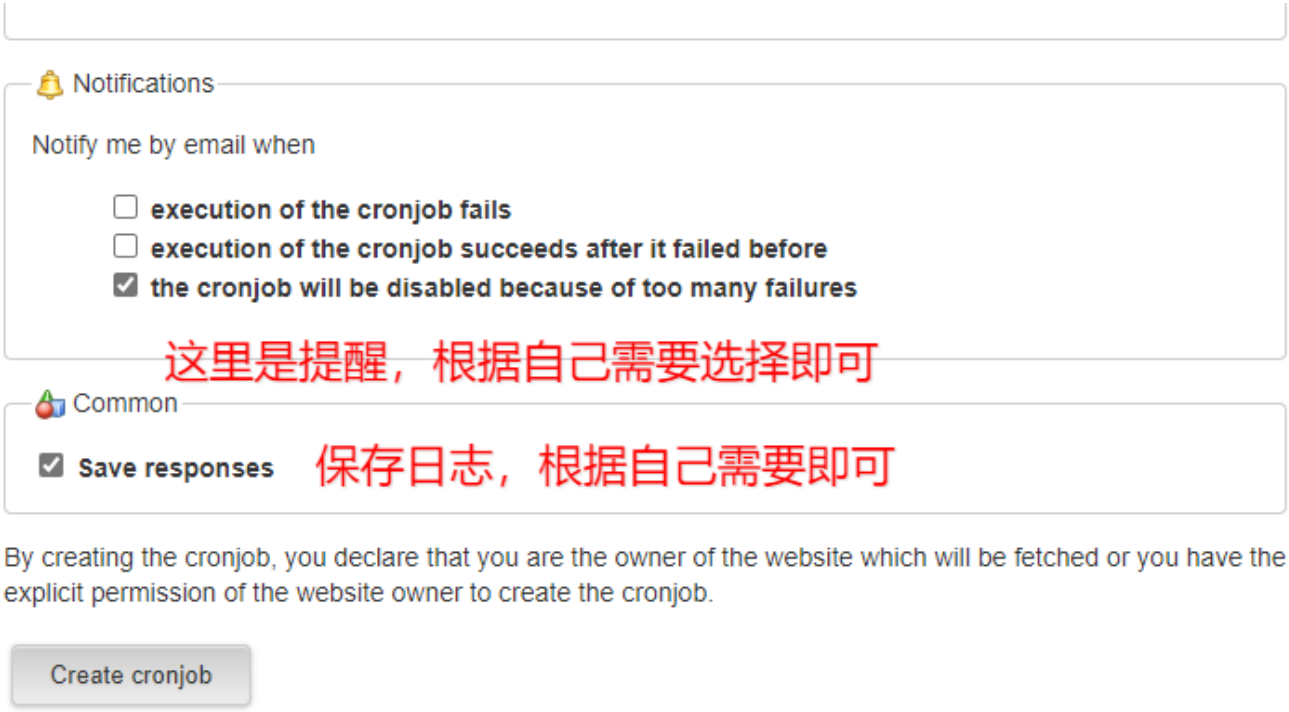


1. 各项配置的解释

字段	简单解释	补充说明
`Title`	任务名称	

字段	简单解释	补充说明
`Address`	监控地址	Leancloud 的 `Web 主机域名`，也就是环境变量的 `ADMIN_URL`
`Schedule`	任务周期	分别为每 X 分钟执行、每天 H:MM 执行、每月 DD 日 HH:MM 执行、自定义
`Notifications`	提醒通知	
`Save responses`	保存日志	

1. 配置示例



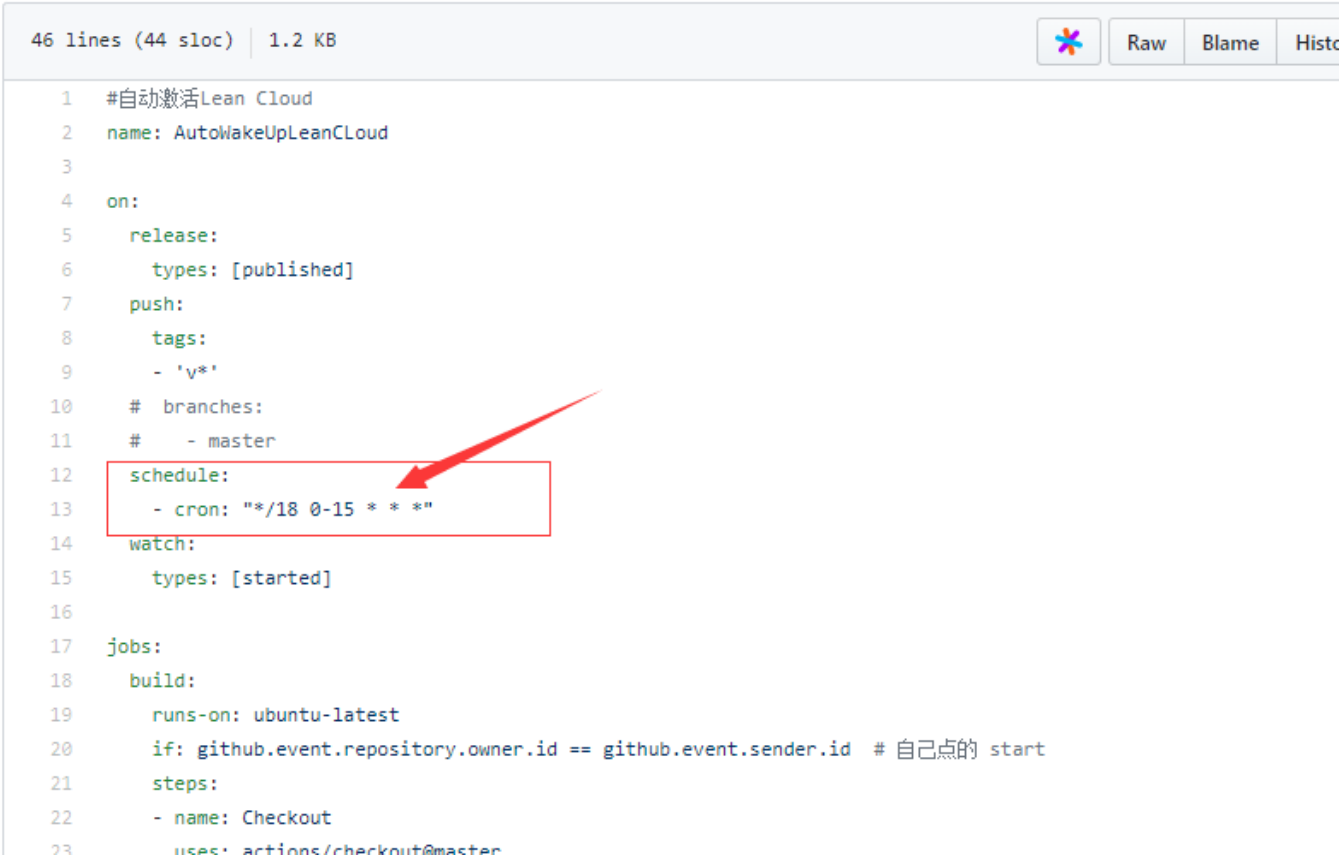
关于自定义时间，你勾哪个就会在哪个时间段执行。例如五个框里你全选了，那么会每分钟都会执行。因此请各位博主自己想好需要在哪个时间段唤醒，而不是无脑复制。

> 点下第一个，在按住 `shift` 点最后一个，会权限所有哦！另外 `ctrl` 可以多选

5.问题

1. 修改频率（时间）

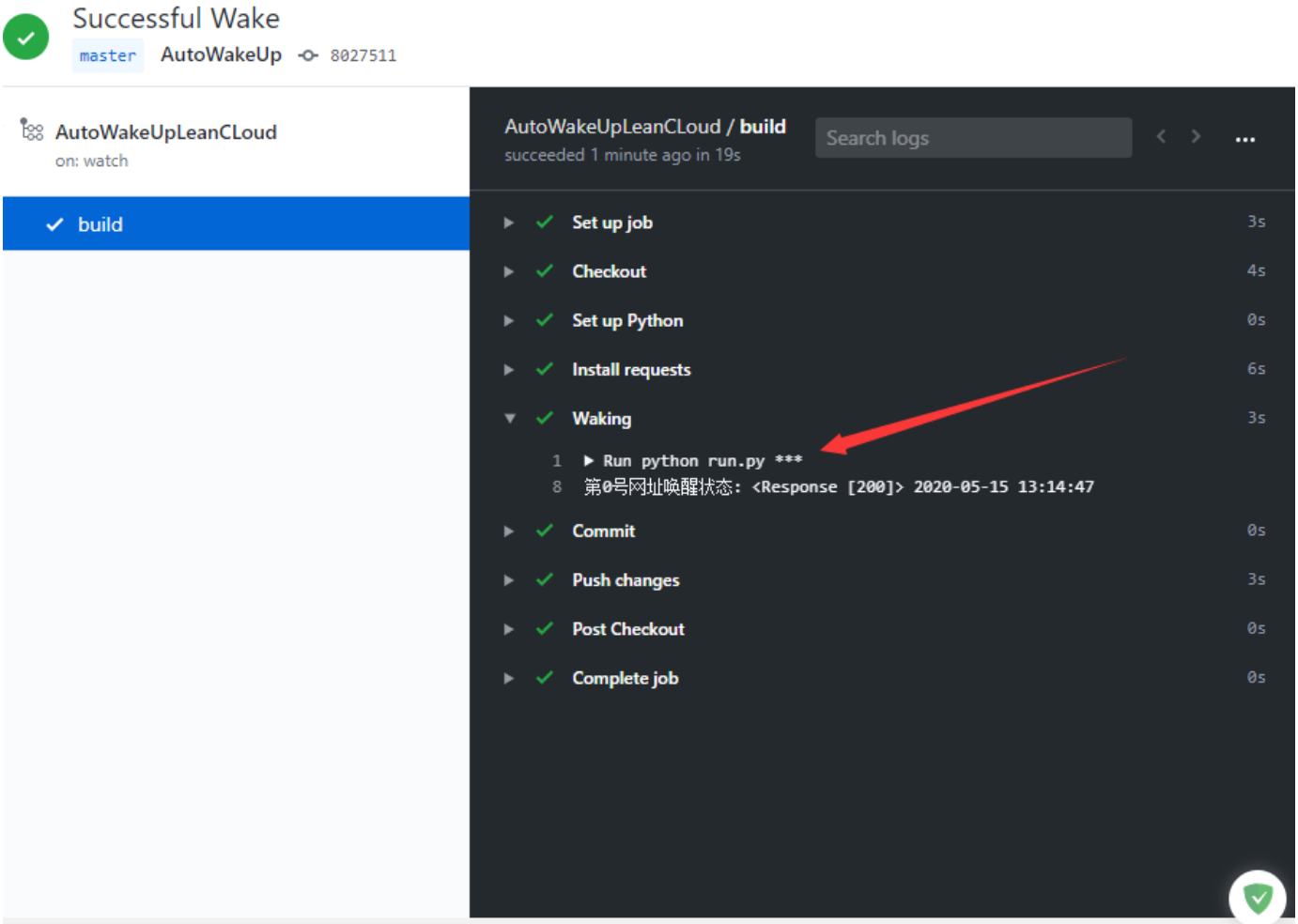
   修改`.github/workflows/autoWakeup.yml` 文件中的 `cron` 表达式即可。



1. 后台地址会不会暴露

   不会的

2. 没有效果



请确保你的第五步成功添加了网址，如果没有添加也会定时执行 actions 的动作而不会报错。可以在详情里查看是否监控的你的地址。正常情况下会如下图所以，多个网址会依次排列。如果没有填写网址则会默认访问作者的博客。

1. 每次都会 commit，太多了。

其实可以每天只运行一次，然后其他时间还是依靠定时函数来完成。例如我将 actions 的时间修改为每天早上 8:00 运行一次。而其他时间通过定时函数唤醒。这样**理论上**也是没问题的。

> Actions 的时间是按 UTC 时间计算的，因此设置时请手动将时间换算成 UTC 时间哦!

参考文献

*小康博客-《优雅解决LeanCloud流控问题》

相关文章:

- 使用Jekyll和github pages构建个人wiki知识库 [2021-01-21]
- 使用PicGo+GitHub图床 [2020-06-20]

- 解决LeanCLoud定时唤醒失败的流控问题 [2020-06-09]
- Jekyll不使用插件生成「书单」页面 [2019-10-27]
- 获取git博文仓库及Markdown源码 [2019-10-04]

「游客及非Github用户留言」：

「Github登录用户留言」：