

TRAE 2.0 SOLO 出道，一键贯通从灵感火花到上线部署的全程协作

立即体验

✕

稀土掘金

NEW

首页

AI Coding

沸点

课程

直播

活动

AI刷题

APP

插件

探索稀土掘金

Q

创作者中心

会员

登录 | 注册

19

1

28

针不戳！GitHub Actions 入坑指南

宅小年 2021-05-09 14,569 阅读7分钟

<<< TRAE 2.0 SOLO 出道，一键贯通从灵感火花到上线部署的全程协作 >>>

前言

相信关注技术前沿的同学，多少也了解过 GitHub Actions

小年我碰巧最近遇上使用的需求，体验了一番之后，感觉两个字：真香！

GitHub Actions

Automate your workflow from idea to production

GitHub Actions makes it easy to automate all your software workflows, now with world-class CI/CD. Build, test, and deploy your code right from GitHub. Make code reviews, branch management, and issue triaging work the way you want.

Get started with Actions

Questions? Contact Sales →



@稀土掘金技术社区

什么是 GitHub Actions？

GitHub Actions 是 GitHub 推出的持续集成（Continuous Integration，简称 CI）服务，它提供了整套虚拟服务器环境，基于它可以进行构建、测试、打包、部署项目等等操作。

持续集成（CI \ CD）主要有三个：持续集成、持续交付、持续部署。

我们一般的软件开发流程是：

1. 开发人员本地代码 commit

2. 通过 git hook 触发自动化测试

3. 测试通过后，合并发布分支

4. 通过 git hook 触发自动部署服务

这里简单的描述了软件开发周期，当然实际上会更加复杂。

我们可以看到，CI \ CD 是由很多操作组成的，比如执行自动化测试、分支合并、服务部署等，而 GitHub 把这一系列的操作都称为 Actions。

当然 GitHub 创新点还不仅于此，不同的项目可能都会使用到相类似的 Action，GitHub 允许开发者把 action 写成独立的脚本文件，存放代码仓库，使得其他开发者可以引用。

GitHub 提供一个官方市场：[GitHub Action Market](#)，在这里可以搜索到你想要的任何 actions，直接引用别人造好的轮子。

Types

Q Search for apps and actions

Apps

Actions

Categories

Actions

An entirely new way to automate your development workflow.

8291 results filtered by Actions

宅小年

Java

38 文章

76k 阅读

96 粉丝

关注

私信

目录

收起

前言

什么是 GitHub Actions？

基本概念

快速入门

实例一：发送邮件

实例二：Github 同步 Gitee

Workflow 语法

name

on

jobs

相关推荐

想用github的pages服务，但github经常...
240阅读 · 0点赞

Github Action 安卓打包实践
2.3k阅读 · 17点赞

Dialog 弹窗也有“花活”？针不戳~
6.6k阅读 · 79点赞

针不戳！腾讯开源GitHub星标125K微服...
354阅读 · 0点赞

GitHub Actions 指南
1.1k阅读 · 9点赞

精选内容

MySQL事务原理：从ACID到隔离级别的...
SevenCoding · 28阅读 · 0点赞

Web不用跳白页，直接在当前页面下载...
SimonKing · 77阅读 · 0点赞

一文帮你掌握集合类库常见工具方法
桦说编程 · 30阅读 · 0点赞

SpringCloud + Elasticsearch + Redis +...
天天摸鱼的java... · 89阅读 · 1点赞

Apache Doris毫秒级分布式数据库引擎
IT果果日记 · 42阅读 · 1点赞

找对属于你的技术圈子

回复「进群」加入官方微信群



Captured by FireShot Pro: 12 8月 2025, 16:34:49
https://getfireshot.com

API management

Chat

Code quality

Code review

Continuous integration

Dependency management

Deployment

IDEs


Learning

Localization


Mobile

Monitoring


Project management




OWASP ZAP Baseline Scan
By zaproxy
Scans the web application with the OWASP ZAP Baseline Scan
154 stars




Sentry Release
By getsentry
GitHub Action for creating a release on Sentry
160 stars




OpenShift Client Installer
By redhat-actions
Install the OpenShift Client (oc) into an action runner
15 stars




Setup Ruby, JRuby and TruffleRuby
By ruby
Download a prebuilt Ruby and add it to the PATH in 5 seconds
348 stars




Codefresh pipeline runner
By codefresh-io
Github action that runs codefresh pipeline
45 stars



Amazon Chime "Chime Webhook" Action for GitHub Actions
By amazon-archives
Send message to Amazon Chime chat rooms
16 stars



Podman Login
By redhat-actions
Log into a container image registry
3 stars



Setup Jira
By atlassian
Set up Jira CLI
18 stars

@稀土掘金技术社区

基本概念

GitHub Actions 主要有以下几个概念

• Workflows

工作流，可以添加到存储库中的自动化过程。工作流由一个或多个作业组成，可以由事件调度或触发。

• Event

事件，触发工作流的特定动作。例如，向存储库提交 pr 或 pull 请求。

• Jobs

作业，在同一跑步器上执行的一组步骤。默认情况下，具有多个作业的工作流将并行运行这些作业。

• Steps

步骤，可以在作业中运行命令的单个任务。步骤可以是操作，也可以是 shell 命令。作业中的每个步骤都在同一个运行程序上执行，从而允许该作业中的操作彼此共享数据。

• Actions

操作是独立的命令，它们被组合成创建作业的步骤。操作是工作流中最小的可移植构建块。你可以创建自己的动作，或者使用 GitHub 社区创建的动作。

• Runners

运行器，安装了 GitHub Actions 运行器应用程序的服务器。。Github 托管的运行器基于 Ubuntu Linux、Microsoft Windows 和 macOS，工作流中的每个作业都在一个新的虚拟环境中运行。

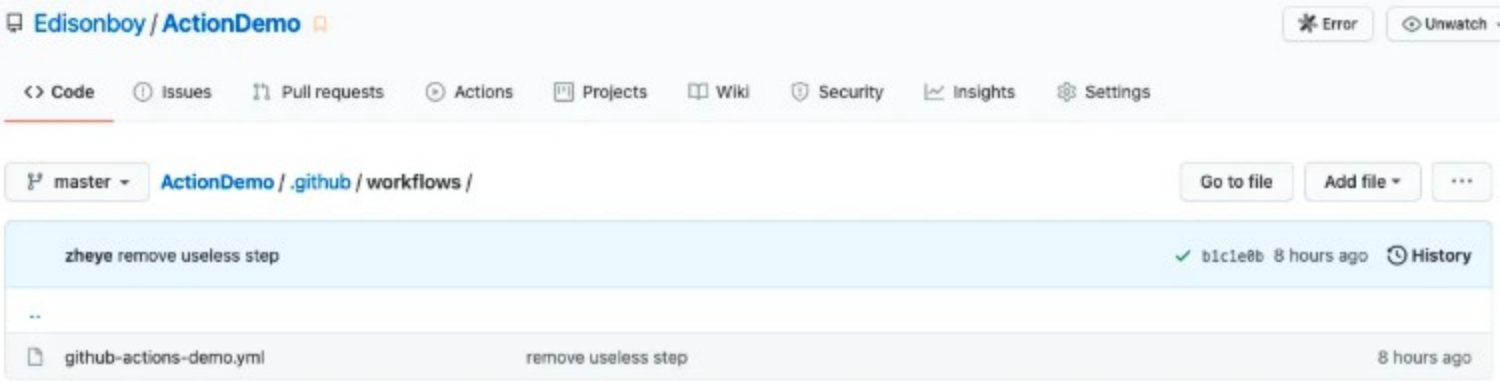
快速入门

闲话少说，咱们先通过简单的实例来体验一下 GitHub Action 的玩法。

实例一：发送邮件

例子比较简单，主要通过 GitHub Action 监听代码 push 事件，并发送邮件。（前提是邮箱需要开通 SMTP 服务）

第一步，在项目中 `./github/workflows/` 路径下添加 `.yaml` 或者 `.yml` 文件，名字可以随便取。在这里我取名为 `github-action-demo.yml`



@稀土掘金技术社区

下面是 `github-action-demo.yml` 的内容：

```
1 name: GitHub Action Demo
2 on: [push]
3 jobs:
4   build:
5     runs-on: ubuntu-latest
6     steps:
7       # copy git 仓库到虚拟机上
8       - name: 'Checkout codes'
9         uses: actions/checkout@v1
10      # 获取最新一条提交的git log
11      - name: Get Git Log
12        id: git_log
13        uses: Edisonboy/latest-git-log-action@main
14        with:
15          tag: origin/master
16      # 发送邮件
17      - name: Send email
18        uses: dawidd6/action-send-mail@v3
19        with:
20          server_address: smtp.qq.com
21          server_port: 465
22          username: ${secrets.MAIL_USERNAME}
23          password: ${secrets.MAIL_PASSWORD}
24          subject: Github Actions job result
25          to: ${secrets.MAIL_TOUSERNAME}
26          from: ${secrets.MAIL_USERNAME}
27          body: ${github.repository}} push log : ${steps.git_log.outputs.lo
```

这个 .yml 文件就是对 GitHub Action 的定义，这里大概讲解一下主要的步骤，有关具体的语法，下文面再详解

1. `on: [push]` 项目 push 的时候触发
 2. `jobs.build.runs-on` 运行在 `ubuntu` 的虚拟机上
 3. `jobs.build.steps` 定义了不同的步骤，如上注释

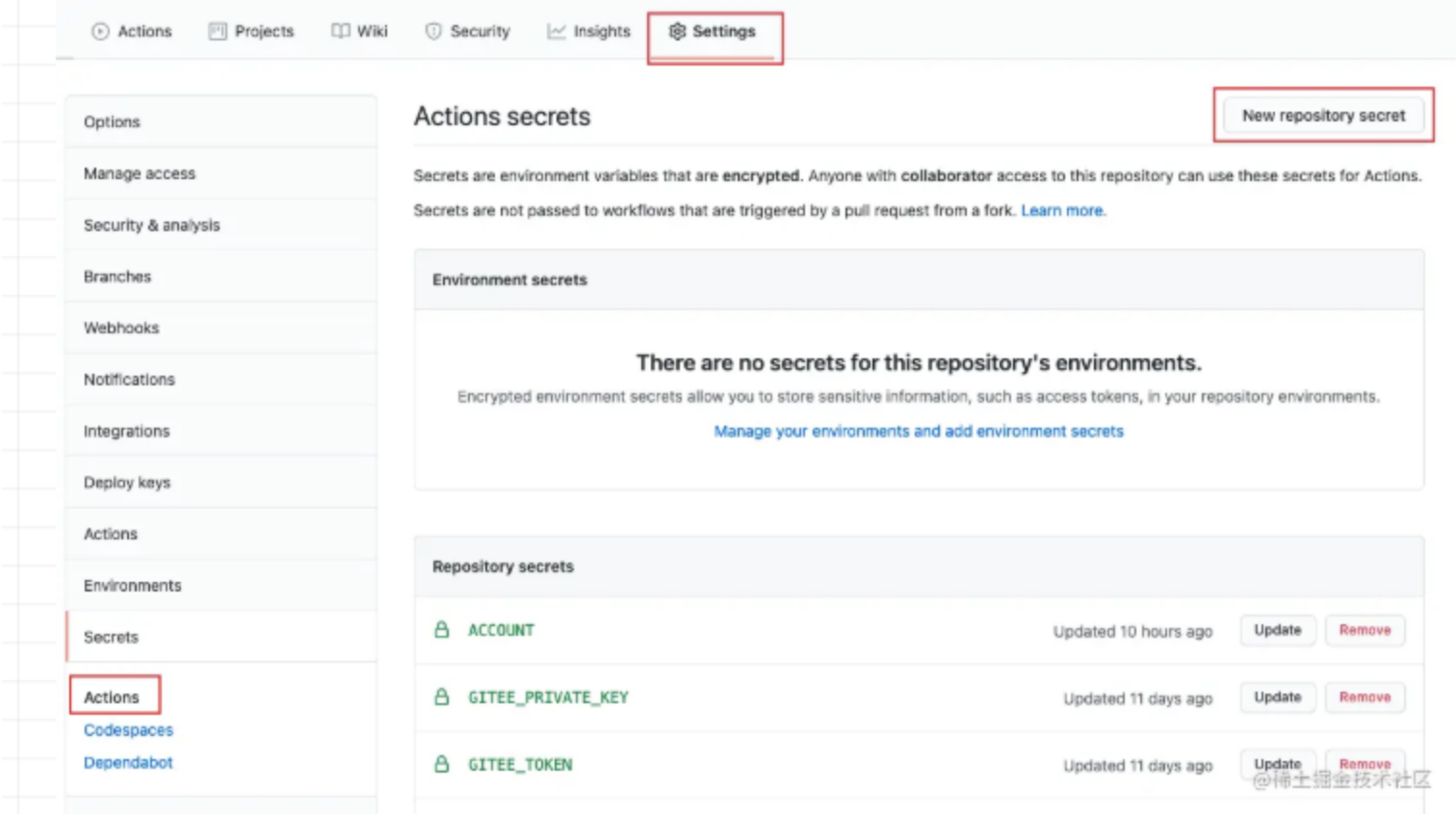
第二步，配置参数

聪明的同学可能还会问，那文件上的 `${{XXX}}` 变量又是从何而来，哪里定义的呢？

上面的实例只要使用了两种类型的参数变量：

- `secrets.XXX` ：GitHub 允许仓库所有者创建和管理需要保密性的参数。例如邮件的账号和密码都是属于敏感参数。

可以通过项目 Settings -> Secrets -> Actions 配置密码，在这里我们添加 `MAIL_USERNAME`、`MAIL_PASSWORD`、`MAIL_TOUSERNAME` 三个配置参数（注意：这里的密码是指 SMTP 服务的授权密码）



- 上下文：可以访问工作流程运行、运行器环境、作业及步骤相关信息的方式
 - `${{github.repository}}` ：当前仓库的的所有者和仓库名称。例如 `Edisonboy/ActionDemo`
 - `${{steps.git_log_outputs.log}}` ：获取step id 为 `git_log` 的输出集

第三步，因为我们定义 push 为触发条件，所以当我们只有push 代码后，我们定义的 GitHub Action 才会被执行。然后在 GitHub 上的 Action 能够实时看到当前的执行状态。

等运行完后，咱们就能收到 git push 的日志邮件了！！！！

实例地址: [Edisonboy/ActionDemo](#)

实例二：Github 同步 Gitee

参考：利用 GitHub Actions 实现 github 代码同步 gitee 仓库

Workflow 语法

结合上面的例子，咱们简单学习一下 workflow 的语法。

workflow 文件必须存储在仓库的 `.github/workflows` 的目录中，扩展名为 `.yaml` 或 `.yml`

name

workflow 的名称, GitHub 在仓库的操作页面上显示 workflow 的名称。

on

触发 workflow 的 GitHub 事件的名称。

```
1 # 单个事件
2 on: push
3
4 # 多个事件列表
5 on: [push, pull_request]
6
7 # 指定main分支的push
8 on:
9   push:
10     branches:
11       - main
```

还可以使用定时调度：

```
1 on:
2   schedule:
3     - cron: '*/30 5,17 * * *'
```

Jobs

作业，workflow 主要执行的核心任务。

jobs.<job_id>

每项作业必须关联一个 ID。例如上面实例的 ID 为 **build**

jobs.<job_id>.name

作业显示在 GitHub 上的名称。

jobs.<job_id>.needs

识别在此作业运行之前必须成功完成的任何作业。

```
yaml
1 jobs:
2   job1:
3   job2:
4     needs: job1
5   job3:
6     needs: [job1, job2]
```

在此示例中，job1 必须在 job2 开始之前成功完成，而 job3 要等待 job1 和 job2 完成。

jobs.<job_id>.runs-on

要运行作业的机器类型。 机器可以是 GitHub 托管的运行器或自托管的运行器。

可用的 GitHub 托管的运行器类型包括：

| 虚拟环境 | YAML 工作流程标签 |
|----------------------|-------------------------------|
| Windows Server 2019 | windows-latest 或 windows-2019 |
| Ubuntu 20.04 | ubuntu-latest 或 ubuntu-20.04 |
| Ubuntu 18.04 | ubuntu-18.04 |
| Ubuntu 16.04 | ubuntu-16.04 |
| macOS Big Sur 11.0 | macos-11.0 |
| macOS Catalina 10.15 | macos-latest 或 macos-10.15 |

注：MacOS 11.0 虚拟环境目前仅作为私人预览提供。任何已经在使用此运行器的用户或组织都可以继续使用它。但我们目前不接受任何其他用户或组织。 macos-latest YAML 工作流程标签仍使用 MacOS 10.15 虚拟环境。

@稀土掘金技术社区

jobs.<job_id>.steps

步骤，每个 Job 包含一个或多个步骤。步骤可以是运行命令、运行设置任务，或者运行仓库中的操作和 Docker 镜像发布等。

例如上面实例定义了 3 个步骤：

1. copy git 仓库到虚拟机上
2. 获取最新一条提交的git log
3. 发送邮件

每个步骤都可以定义以下几个字段：

```
yaml
1 jobs..steps[*].id : 步骤的唯一标识符
2 jobs..steps[*].name : 步骤显示在 GitHub 上的名称
3 jobs..steps[*].if : 自定义表达式，判断是否满足条件
4 jobs..steps[*].uses : 选择公共仓库中、或发布 Docker 容器映像作为一部分运行的操作。例如上面的
5 jobs..steps[*].run : 运行 shell 命令程序。
```


有关 workflow 语法更多的细节使用可以参考：[GitHub Actions 官方文档](#)

小结

在执行 workflow 的过程中，GitHub 会自动为我们提供相应的虚拟服务器资源，这时候脑洞大的同学就应该想到：有没有可能连接到这个服务器，白嫖一个服务器为所欲为呢。

看了一下 GitHub Actions 官方文档上提供的虚拟服务器资源，这个配置比一台低配个人服务器还香。

Windows 和 Linux 虚拟机的硬件规格：

- 2 核 CPU
- 7 GB RAM 内存
- 14 GB SSD 硬盘空间

MacOS 虚拟机的硬件规格：

- 3 核 CPU
- 14 GB RAM 内存
- 14 GB SSD 硬盘空间

有兴趣的同学可以参考：[SSH 连接到 GitHub Actions 虚拟服务器](#)

还有更硬核的操作！黑客可谓把白嫖发挥得淋漓尽致，尽然想到利用 GitHub 的服务器挖矿（详情：[黑客用GitHub服务器挖矿，三天跑了3万个任务，代码惊现中文](#)）

当然 GitHub Actions 也没有这么傻，肯定想到咱们白嫖的各种骚操作，所以对于使用增加了一定的限制

使用限制

当使用 GitHub 托管的运行器时，GitHub Actions 的使用有一些限制。这些限制可能会有变动。

注：对于自托管的运行器，适用不同的使用限制。更多信息请参阅[“关于自托管运行器”](#)。

- **作业执行时间** - 工作流程中的每个作业最多可以运行 6 个小时。如果作业达到此限制，该作业将会终止而无法完成。
- **工作流程运行时间** - 每个工作流程的运行时限为 72 小时。如果工作流程运行时间达到此限制，其运行将被取消。
- **API 请求** - 在一个仓库的所有操作中，一个小时内最多可执行 1000 个 API 请求。如果超出，额外的 API 调用将失败，这可能导致作业失败。
- **并发作业** - 您的帐户中可并发运行的作业数量，具体取决于您的 GitHub 计划，如下表所示。如果超出，任何额外的作业都会排队。

| GitHub 计划 | 同时运行的作业总数 | MacOS 作业同时运行的最大数量 |
|-----------|-----------|-------------------|
| 免费 | 20 | 5 |
| Pro | 40 | 5 |
| 团队 | 60 | 5 |
| 企业 | 180 | 50 |

- **作业矩阵** - 作业矩阵在每次工作流程运行时最多可生成 256 个作业。此限制也适用于自托管运行器。

@稀土掘金技术社区

总的来说，**GitHub Actions** 是一个非常实用有趣的功能，没有你做不到的，只有你想不到的。

参考资料

[GitHub Actions 官方文档](#)

[GitHub Actions 入门教程-阮一峰](#)

[GitHub Actions 进阶技巧](#)

普通的改变，将改变普通

我是宅小年，一个在互联网低调前行的小青年

关注公众号「宅小年」，个人博客 [edisonz.cn](#)，阅读更多分享文章



微信搜一搜

宅小年

@稀土掘金技术社区

标签： [GitHub](#)

评论 1



[登录 / 注册](#) 即可发布评论!

最热 | [最新](#)



稀土君 最酷的 @稀土

你的作品数据较昨天有所变化，赶紧来创作者中心看一下；现在体验专栏新功能还有大额京东卡拿，点击查看：[sourl.co](#)；app用户请升级最新版本应用后在我的页面查看哟！

4年前 [点赞](#) [评论](#)

...

为你推荐

完整解析使用 Github Action 构建和发布 Flutter 应用

恋猫de小郭 | 3年前 | 5.3k | 56 | 13 | 前端 | Android | Flutter

github-actions 入门实战

月下吴刚在掘金 | 1年前 | 138 | 1 | 评论 | 自动化运维 | GitHub

带你学习通过GitHub Actions如何快速构建和部署你自己的项目，打造一条属于自己的流水线

初夏的阳光 | 1年前 | 12k | 36 | 25 | GitHub | Docker | Java

前端工程化：Github Action 实现 CI/CD

Moment | 4月前 | 570 | 23 | 1 | 前端 | 后端 | 前端工...

Github上手指南（十一） | GitHubActions使用大全

白哥学前端 | 2年前 | 1.2k | 21 | 评论 | 前端 | GitHub

使用VitePress和GitHub Actions高效搭建文档网站

长林啊 | 1年前 | 806 | 10 | 4 | Vue.js | VitePress | GitHub

Github Action 快速上手指南

蛮三刀酱 | 4年前 | 1.7k | 2 | 1 | GitHub

GitHub Actions实现定时任务，免费运行

你不会困 | 1年前 | 767 | 13 | 8 | Node.js

Github Actions自动化工作流——省力省心

小子王 | 3年前 | 3.9k | 13 | 1 | GitHub | 前端

Github Action 是什么？能干什么？怎么做到的？如何开发一个action

thinkasany | 1年前 | 984 | 4 | 评论 | CI/CD

10 个你该了解的 GitHub Actions 进阶技巧

shaowenchen | 4年前 | 1.7k | 6 | 3 | GitHub

10分钟教程-使用GitHub Actions进行WordPress持续部署

未定义变量 | 4年前 | 1.2k | 8 | 评论 | GitHub

Github Action 基础与自动发布npm实例

赵_叶紫 | 1年前 | 387 | 1 | 评论 | GitHub

Github Actions 工作流的创建管理

ZFE | 4年前 | 2.7k | 14 | 评论 | 掘金翻译... | GitHub

Github Actions 自动化部署 Vue3 博客项目

小猪皮皮呆 | 3年前 | 2.5k | 23 | 4 | CI/CD