

Github Actions

吴温炎

GO →



精彩继续!

这里有更多前沿技术与工程实现

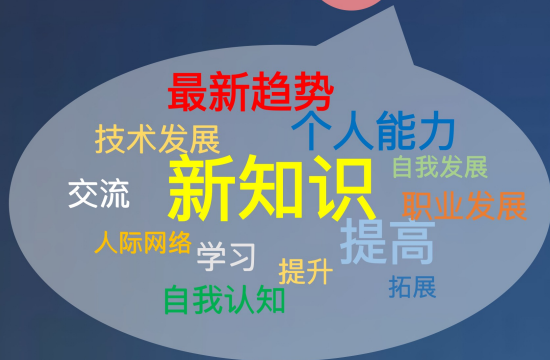
加入我们，你将可以收获：



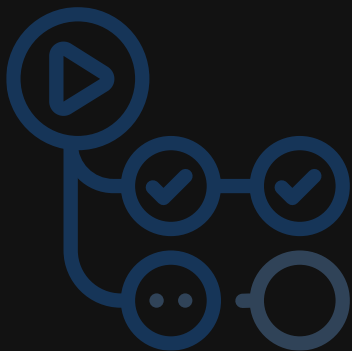
分享人



参与人



TOC



1. [基本介绍](#)
2. [上手指南](#)
3. [应用场景](#)
4. [高级特性](#)
5. [最佳实践](#)
6. [案例分享](#)
7. [参考资料](#)
8. [QA](#)



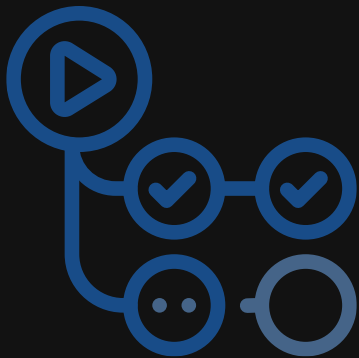
基本介绍

- Github Actions 是什么?
- Features
- Pricing
- 基础概念



基本介绍

Github Actions 是什么?



Automate your workflow from idea to production

从想法到生产，自动化您的工作流程

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.

GitHub Actions 现在可以使用世界一流的 CI/CD 轻松实现所有软件工作流程的自动化。直接从 GitHub 构建、测试和部署您的代码。按照您想要的方式进行代码审查、分支管理和问题分类。

基本介绍 Features (todo:图标完善, 丰富图文介绍)

- 🖥️ 操作系统 - 适用于 Linux、macOS、Windows、ARM 和容器；直接在虚拟机或容器内运行。
- 🎨 矩阵构建 - 通过矩阵工作流程可以同时跨多个操作系统和运行时版本进行测试，从而节省时间。
- 👤 任何语言 - 支持 Node.js、Python、Java、Ruby、PHP、Go、Rust、.NET 等。
- 📦 实时日志 - 查看您的工作流程实时运行的颜色和表情符号。
- 🔑 密钥存储 - 通过将包含 Git 流程的工作流程文件编码到您的存储库中，自动执行您的软件开发实践。
- 🐳 多容器测试 - compose 只需将一些内容添加到工作流程文件中即可在工作流程中测试您的 Web 服务及其数据库。
- 🏠 社区支持的工作流程 - GitHub Actions 连接您的所有工具，以自动化开发工作流程的每一步。

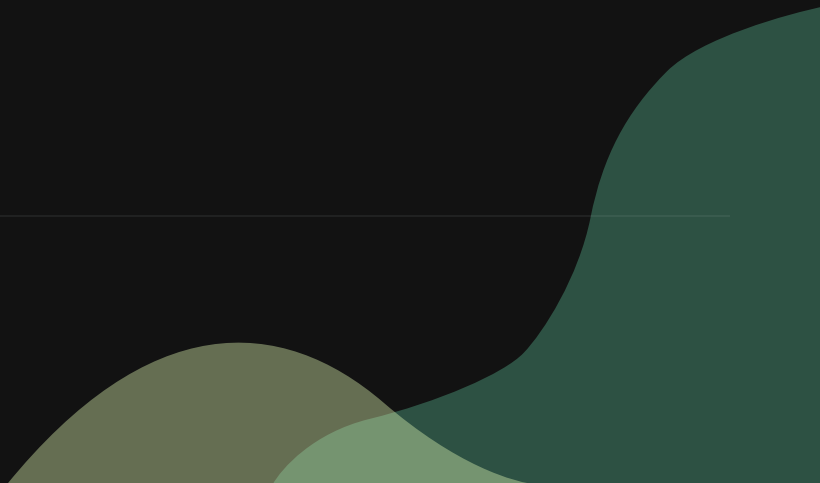
基本介绍 Pricing

- 免费版 - 2000 分钟/月
- Pro - 3000 分钟/月（其中 Pro 为 4 美元/月）
- Team - 3000 分钟/月（其中 Team 为 4 美元/用户/月）
- Enterprise - 50000 分钟/月（其中 Enterprise 为 21 美元/用户/月）

对于 开源 项目，无限制。

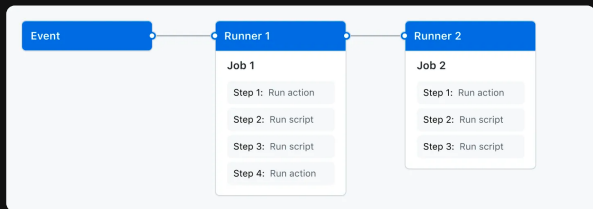
开源就可以薅羊毛.....

详见：[个人计划](#) | [组织计划](#)



基本介绍

基础概念



```
on: push
jobs:
  test:
    strategy:
      matrix:
        platform: [ubuntu-latest, macos-latest, windows-latest]
    runs-on: ${matrix.platform}
    steps:
      - uses: actions/checkout@v3
      - uses: actions/setup-node@v3
        with:
          node-version: 16
      - run: npm run test
```

工作流
workflows

自动化流程，可以有多个，存放在
.github/workflows 目录中。

事件 event

比如推送代码，创建 PR 等；支持的事件

作业 jobs

在同一运行器上执行的一组步骤。顺序执行，相互依赖。

操作 actions

作业中的一组任务，可以是自定义的，也可以是开源市场提供的。

运行程序
runners

运行工作流的服务器；支持 Ubuntu Linux、Microsoft Windows 和 macOS。

操作



变量

参考资料

- <https://docs.github.com/zh/actions/learn-github-actions/variables>



分支

123



上手指南

- 创建工作流(`workflows`)
- `workflows` 文件的结构和语法
- 触发器和事件
- 任务和步骤
- 使用环境变量和密钥



上手指南 创建 workflows

- 方式一：使用官方模板（适合新手）

The screenshot shows the GitHub repository page for 'github-actions-templates / hexo'. The 'Actions' tab is selected, and the 'New workflow' button is highlighted with a red circle. Below the 'Actions' tab, there is a section titled 'Choose a workflow' with a search bar and a list of suggested workflows. The 'Simple workflow' is highlighted, and the 'Configure' button is visible. Below this, there is a grid of workflow templates categorized under 'Deployment', including 'Deploy Node.js to Azure Web App', 'Deploy to Amazon ECS', 'Build and Deploy to OKE', 'Terraform', 'Deploy to Alibaba Cloud ACK', 'Deploy to IBM Cloud Kubernetes Service', 'Tencent Kubernetes Engine', and 'OpenShift'.

The screenshot shows the GitHub Actions workflow editor for a workflow named 'blank.yml' in the 'main' branch. The workflow is a basic CI workflow that runs on 'ubuntu-latest' and contains a single job named 'build'. The workflow is defined as follows:

```
1 # This is a basic workflow to help you get started with Actions
2
3 name: CI
4
5 # Controls when the workflow will run
6 on:
7   # Triggers the workflow on push or pull request events but only for the "main" branch
8   push:
9     branches: [ "main" ]
10  pull_request:
11    branches: [ "main" ]
12
13 # Allows you to run this workflow manually from the Actions tab
14 workflow_dispatch:
15
16 # A workflow run is made up of one or more jobs that can run sequentially or in parallel
17 jobs:
18   # This workflow contains a single job called "build"
19   build:
20     # The type of runner that the job will run on
21     runs-on: ubuntu-latest
22
23     # Steps represent a sequence of tasks that will be executed as part of the job
24     steps:
25       # Checks-out your repository under $GITHUB_WORKSPACE, so your job can access it
26       - uses: actions/checkout@v3
27
28       # Runs a single command using the runners shell
29       - name: Run a one-line script
30         run: echo Hello, world!
31
32       # Runs a set of commands using the runners shell
33       - name: Run a multi-line script
34         run: |
35           echo Add other actions to build,
36           echo test, and deploy your project.
```

- 方式二：手动创建（适合有经验的用户）

文件路径： `.github/workflows/[xxx].yaml` ，支持配置多个 `workflows` 文件。

```
name: CI

on:
  push:
    branches: [ "main" ]
  pull_request:
    branches: [ "main" ]

jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v3

      - name: Run a one-line script
        run: echo Hello, world!

      - name: Run a multi-line script
        run: |
          echo Add other actions to build,
          echo test, and deploy your project.
```

上手指南

workflows 文件的结构和语法

```
# 工作流的名称
name: CI

# 触发器
on:
  # 当往 main 分支推送代码, 或发起合并请求时触发
  push:
    branches: [ "main" ]
  pull_request:
    branches: [ "main" ]

# 任务列表 (此处只列了一个)
jobs:
  # 任务名称, 比如此处为 build 任务; 多个任务默认是并行, 可通过配置建立依赖关系
  build:
    # 任务运行的服务器环境
    runs-on: ubuntu-latest

    # 任务运行的步骤, 此处是串行
    steps:
      # 拉取代码, 此处使用的时第三方 (虽然是官方组织) 的 actions 组件
      - uses: actions/checkout@v3

      # 执行一个单行命令 (其中 name 为该命令的名称)
```

← CI

workflows

✓ Create blank.yml #1

Summary

Jobs

✓ build

job name

Run details

Usage

Workflow file

Triggered via push 4 hours ago

👤 f1c1125 pushed 🔗 5b8f613 `main`

Status

Success

Total duration

13s

Billable time

1m

Artifacts

—

blank.yml


on: push

✓ build

4s


build

succeeded 4 hours ago in 4s


>  Set up job

▼  Run actions/checkout@v3


```
1  ▶ Run actions/checkout@v3
14 Syncing repository: github-actions-templates/blank
15 ▶ Getting Git version info
19 Temporarily overriding HOME='/home/runner/work/_temp/4518519d-4856-4c1d-a53d-9330f8c03d7e' before making global git config changes
20 Adding repository directory to the temporary git global config as a safe directory
21 /usr/bin/git config --global --add safe.directory /home/runner/work/blank/blank
22 Deleting the contents of '/home/runner/work/blank/blank'
23 ▶ Initializing the repository
37 ▶ Disabling automatic garbage collection
39 ▶ Setting up auth
45 ▶ Fetching the repository
62 ▶ Determining the checkout info
63 ▶ Checking out the ref
67 /usr/bin/git log -1 --format='%H'
68 '5b8f6138f7722cc8c0ae4288bf8751ae4c72bd55ae'
```


▼  Run a one-line script

```
1  ▶ Run echo Hello, world!
4  Hello, world!
```

▼  Run a multi-line script

```
1  ▶ Run echo Add other actions to build,
5  Add other actions to build,
6  test, and deploy your project.
```

>  Post Run actions/checkout@v3

>  Complete job

上手指南

触发器和事件

工作流程触发器是导致工作流程运行的事件。 这些事件可以是：

- 工作流存储库中发生的事件，如：推送代码、创建 ISSUE、发起 PR
- 在 GitHub 之外发生并在 GitHub 上触发 `repository_dispatch` 事件的事件，如：通过 Webhook 发送自定义时间
- 预定时间，如：每天 4 点全量单测
- 手动，如：点击按钮

例如：当你推送代码到某个分支时，可以触发事件，以运行相关任务。

```
on:
  issues:
    types:
      - opened

jobs:
  label_issue:
    runs-on: ubuntu-latest
    steps:
      - env:
          GITHUB_TOKEN: ${ secrets.MY_TOKEN }
          ISSUE_URL: ${ github.event.issue.html_url }
        run: |
          gh issue edit $ISSUE_URL --add-label "triage"
```

常用 workflow 事件

- 推送代码，创建合并等

```
on:
  push:
    branches:
      - 'main'
      - 'releases/**' # 指定分支
  tags:
    - 'v1.*' # 指定 Tag
  paths:
    - '**.js' # 指定文件
    - '!*-alpha.js' # 排除文件
  pull_request:
    branches:
      - 'main'
      - '!releases/**-alpha' # 忽略分支
```



workflow events

- 定时事件

```
on:  
  schedule:  
    - cron: '30 5 * * 1,3'  
    - cron: '30 5 * * 2,4'
```

Cron 语法:

```
minute (0 - 59)  
hour (0 - 23)  
day of the month (1 - 31)  
month (1 - 12 or JAN-DEC)  
day of the week (0 - 6 or SUN-SAT)  
* * * * *
```



workflow 事件

- 更多事件: <https://docs.github.com/zh/actions/using-workflows/events-that-trigger-workflows>
- 基础概念: <https://docs.github.com/zh/actions/using-workflows/triggering-a-workflow>
- 手动运行 workflow: <https://docs.github.com/zh/actions/using-workflows/manually-running-a-workflow>
- workflow 语法: <https://docs.github.com/zh/actions/using-workflows/workflow-syntax-for-github-actions>



上手指南

任务和步骤



上手指南

使用环境变量和密钥



应用场景

- 自动化构建和测试
- 自动化部署
- 发布软件包和文档
- 定时任务和计划作业
- 集成第三方服务



应用场景

自动化构建和测试

222



应用场景 自动化部署

以为 Hexo 举例

```
name: Github Pages

on:
  push:
    branches:
      - main

jobs:
  pages:
    runs-on: ubuntu-latest
    permissions:
      contents: write
    steps:
      - uses: actions/checkout@v3

      - uses: actions/setup-node@v3
        with:
          node-version: 18
          cache: 'npm'

      - uses: actions/cache@v3
        with:
          path: node_modules
          key: ${{ runner.OS }}-npm-cache-${{ hashFiles('**/package.json') }}
```

- `on` - 触发条件: `push` 事件, 且分支为 `main`
- `jobs` - 作业
- `{{ secrets.GITHUB_TOKEN }}` - 自动令牌认证, GitHub 会自动创建一个唯一的GITHUB_TOKEN密钥以在您的工作流程中使用。需要配合 `permissions` 一起使用。

应用场景

发布软件包和文档



应用场景

定时任务和计划作业



应用场景

集成第三方服务



高级特性

- 并行和串行任务
- 依赖和条件
- 存储和共享数据
- 自定义动作
- 安全性和凭证管理



高级特性

并行和串行任务



高级特性

依赖和条件



高级特性

存储和共享数据



高级特性

自定义动作



高级特性

安全性和凭证管理

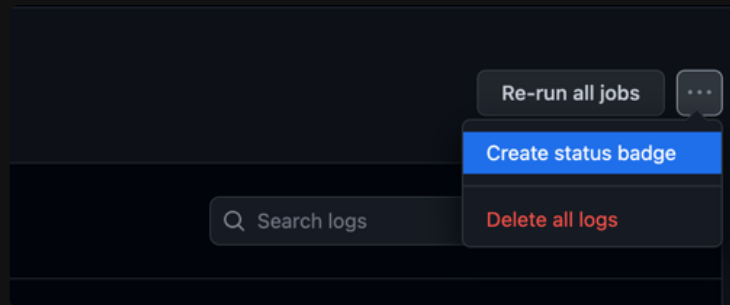
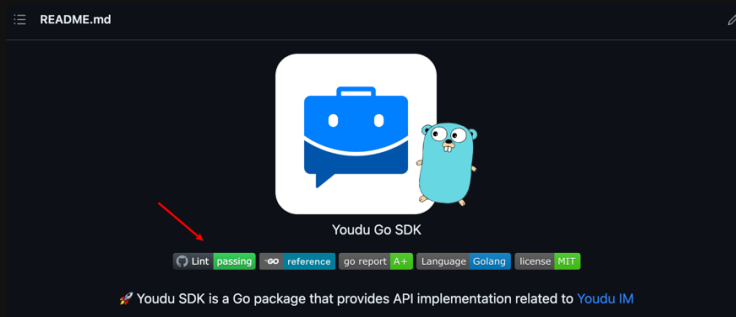


最佳实践

- 使用徽章来标记 CI 状态
- 使用缓存提高性能
- 错误处理和调试
- 版本控制和代码审查
- 效率和资源管理
- 社区资源和扩展



最佳实践 使用徽章来标记 CI 状态



```
[!Lint](https://github.com/addcnos/youdu/actions/workflows/lint.yml/badge.svg)](https://github.com/addcnos/youdu/actions/workflows/lint.
```



最佳实践

使用缓存提高性能



最佳实践

错误处理和调试



最佳实践

版本控制和代码审查



最佳实践

效率和资源管理



最佳实践

社区资源和扩展



案例分享