

GitHub Actions & Runners

Presentation slides for developers

Press Space for next page →

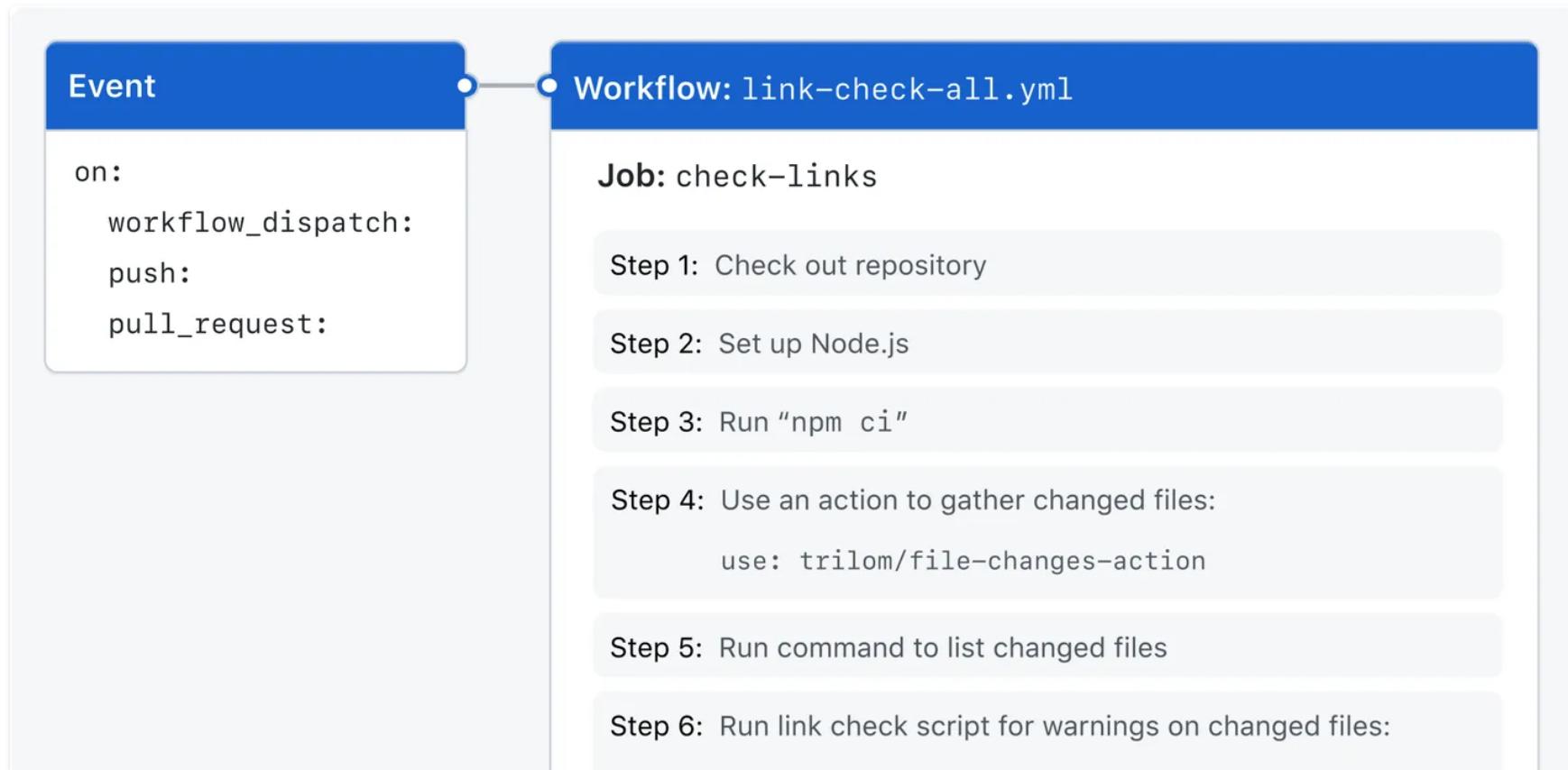


Table of contents

1. GitHub Actions
2. Action Examples Used Inhouse
 1. Example Workflow Run In a Container
 2. Interactive Workflow With User Input
 3. Interactive Workflow With User Input (continued)
3. GitHub Runners
4. Current Runner State
 1. GHES: Action Policies
 2. GHES: Self-hosted Enterprise Runners
 3. GHES: Self-hosted Runners ID-SYS
5. Runners Conclusion:
6. Book Recommendations

GitHub Actions

<https://docs.github.com/en/actions/examples/using-scripts-to-test-your-code-on-a-runner>



Basic workflow to render mkdocs-material documentation

Github-hosted runner, turns markdown to HTML and commits to gh-pages branch.

```
1  ---
2  name: ci
3  on:
4    push:
5      branches:
6        - main
7  jobs:
8    deploy:
9      runs-on: ubuntu-latest
10     steps:
11       - uses: actions/checkout@v3
12       - uses: actions/setup-python@v2
13         with:
14           python-version: 3.x
15       - run: pip install mkdocs-material
16       - run: pip install mkdocs-git-revision-date-plugin
17       - run: mkdocs gh-deploy --force
```

Example Workflow Run In a Container

```
1  name: ansible-and-yaml-linting
2
3  on:
4    pull_request:
5    workflow_dispatch:
6
7  jobs:
8    liniting:
9      runs-on: ubuntu-latest
10   container:
11     image: ghcr.io/idsys-unibe-ch/idsys.ansible:v1.0.0
12     env:
13       # These tell ansible-lint to use github compatible annotation format:
14       GITHUB_ACTIONS: "true"
15       GITHUB_WORKFLOW: "{{ github.workflow.name }}"
16
17   steps:
18     - name: Checkout repository ${{ github.repository.name }}
19       uses: actions/checkout@v3
20       with:
21         fetch-depth: 0
22       env:
23         GITHUB_TOKEN: ${{ secrets.CI_BOT_PAT }}
24
25     ...
26
```

Interactive Workflow With User Input

This is a part of a PoC workflow that will allow ASM to make dumps from one env in to another, i.e. parisprod -> paristest.

```
1  on:
2    workflow_dispatch:
3      inputs:
4        job:
5          type: choice
6          description: What job to run
7          options:
8            - kslldbtest1 ( kslp → kslldbtest1 )
9            ...
10           - parisdbiam ( parisprod → parisdbiam )
11
12  jobs:
13    clone_DB:
14      runs-on: [self-hosted, ubuntu]
15      env:
16        SELECTION: ${{ github.event.inputs.job }}
17      steps:
18        - uses: actions/checkout@v3
19        - name: Set up Python 3.10
20          uses: actions/setup-python@v4
21          with:
22            python-version: '3.10'
```

Interactive Workflow With User Input (continued)

```
1  - name: Install dependencies
2    run: 'python -m pip install --upgrade pip
3
4      if [ -f requirements.txt ]; then pip install -r requirements.txt; fi
5
6      '
7
8    - id: systems
9      run: echo "systems=$(echo '${{ github.event.inputs.job }}' | cut -d ' ' -f1 )" >> $GITHUB_OUTPUT
10   - id: originDB
11     run: echo "originDB=$(echo '${{ github.event.inputs.job }}' | cut -d ' ' -f3 )" >> $GITHUB_OUTPUT
12   - id: destinationDB
13     run: echo "destinationDB=$(echo '${{ github.event.inputs.job }}' | cut -d ' ' -f5 )" >> $GITHUB_OUTPUT
14   - id: OriginDBPWName
15     run: echo "OriginDBPWName=${{ format('POSTGRES_DB_{0}_KEY', steps.originDB.outputs.originDB) }}"
16         >> $GITHUB_OUTPUT
17   - id: DestinationDBPWName
18     run: echo "DestinationDBPWName=${{ format('POSTGRES_DB_{0}_KEY', steps.destinationDB.outputs.destinationDB)
19         >> $GITHUB_OUTPUT
20   - name: execute py script
21     run: python main.py -p ${{ steps.systems.outputs.systems }} -opw "${{ secrets[steps.OriginDBPWName.outputs.O
22         }}" -dpw "${{ secrets[steps.DestinationDBPWName.outputs.DestinationDBPWName]
23         }}" -f
```

Github-hosted Runners

Docs on self-hosted Runners

Pros:

- Managed by GitHub
- Each runner is a new VM with
- Many OS supported (Ubuntu Linux, Windows, macOS)

Cons:

- dynamic IPs, shared with everybody
- no access to UniBE internal resources (border firewall)

Self-hosted Runners

Docs on self-hosted Runners

Pros:

- Managed by ID => anything possible

Cons:

- Manged by ID =>

Current Runner State

With GitHub Actions, developers can write and combine individual tasks called actions to create custom workflows. To enable GitHub Actions for your GitHub Enterprise Server instance, you must host at least one machine to execute jobs. This machine is called a self-hosted runner. Self-hosted runners can be physical, virtual, in a container, on-premises, or in a cloud. Your runner machine connects to GitHub Enterprise Server using the GitHub Actions self-hosted runner application. Self-hosted runners can run Linux, Windows, or macOS. For more information, see "About self-hosted runners."

Demo:

- <https://github.unibe.ch/michael-rolli/testy>
- <https://github.unibe.ch/idsys-unibe-ch/testy/settings>

GHES: Action Policies

The screenshot shows the GitHub Enterprise interface for managing Action Policies. The left sidebar is for the 'University of Bern' organization, with the 'Policies' section selected. The main content area is titled 'Actions' and shows the 'Policies' tab is active. A dropdown menu 'Enable for all organizations' is open, showing three options: 'Allow all actions' (selected), 'Allow local actions only', and 'Allow select actions'. Below the dropdown, there is a note about enabling GitHub Actions and a 'Save' button.

Enterprise Search or jump to... / Pull requests Issues Explore

University of Bern

Organizations People Policies

Repositories Actions Projects Teams Options Code security and analysis

Github Connect Code Security

Actions

Policies Runners Runner groups

Policies

Actions can be enabled for all organizations or only for specific organizations. If disabled, GitHub Actions cannot run.

Enable for all organizations ▾

Allow all actions
Any action can be used, regardless of who authored it or where it is defined.

Allow local actions only
Only actions defined in a repository within the enterprise can be used.

Allow select actions
Only actions that match specified criteria, plus actions defined in a repository within the enterprise, can be used. [Learn more about allowing specific actions to run.](#)

Save

GHES: Self-hosted Enterprise Runners

The screenshot shows the GitHub Enterprise Settings interface. The left sidebar has a dark theme with the following items:

- University of Bern (highlighted)
- Organizations
- People
- Policies (highlighted)
- Repositories
- Actions
- Projects
- Teams
- Options
- Code security and analysis
- Github Connect
- Code Security
- Settings

The main content area has a light background. At the top, there are navigation links: Enterprise, Search or jump to..., Pull requests, Issues, Explore, and a user icon. Below these are three tabs: Policies, Runners (which is selected), and Runner groups.

A descriptive text block states: "Host your own runners and customize the environment used to run jobs in your GitHub Actions workflows. Runners added to this enterprise can be used to process jobs in multiple repositories in your organizations. [Learn more about self-hosted runners.](#)"

Below this is a search bar labeled "Search runners" and a green "New runner" button. A large callout box in the center says "There are no runners configured" and provides a link: "Learn more about using runners to run actions on your own servers."

GHES: Self-hosted Runners ID-SYS

 **UniBE System Services**
Organization account [Switch to another account ▾](#) [Go to your organization profile](#)

General

Access

[Repository roles](#)

[Member privileges](#)

[Team discussions](#)

Code, planning, and automation

[Repository](#) ▼

[Actions](#) ^

General

Runners

Runner groups

Caches

Hooks

Discussions

Packages

Runners

Host your own runners and customize the environment used to run jobs in your GitHub Actions workflows. Runners added to this organization can be used to process jobs in multiple repositories in your organization. [Learn more about self-hosted runners.](#)

All ▼ [New runner](#)

Runners	Status	...
 id-ghertest-ubuntu self-hosted Linux X64 ubuntu ubuntu-20.04 Runner group: idsys-default	Idle	...
 id-ghertest-rocky self-hosted Linux X64 Rocky-8 rocky Runner group: idsys-default	Idle	...

Runners Conclusion:

Three possiblities:

- ID runs their own self-hosted runner fleet (who from whom at what cost => Vision, RE, Budget ...?)
- Use large runners, currently beta
- Wait and buy GitHub-hosted runners in Azure => when, at what cost?

Book Recommendations

