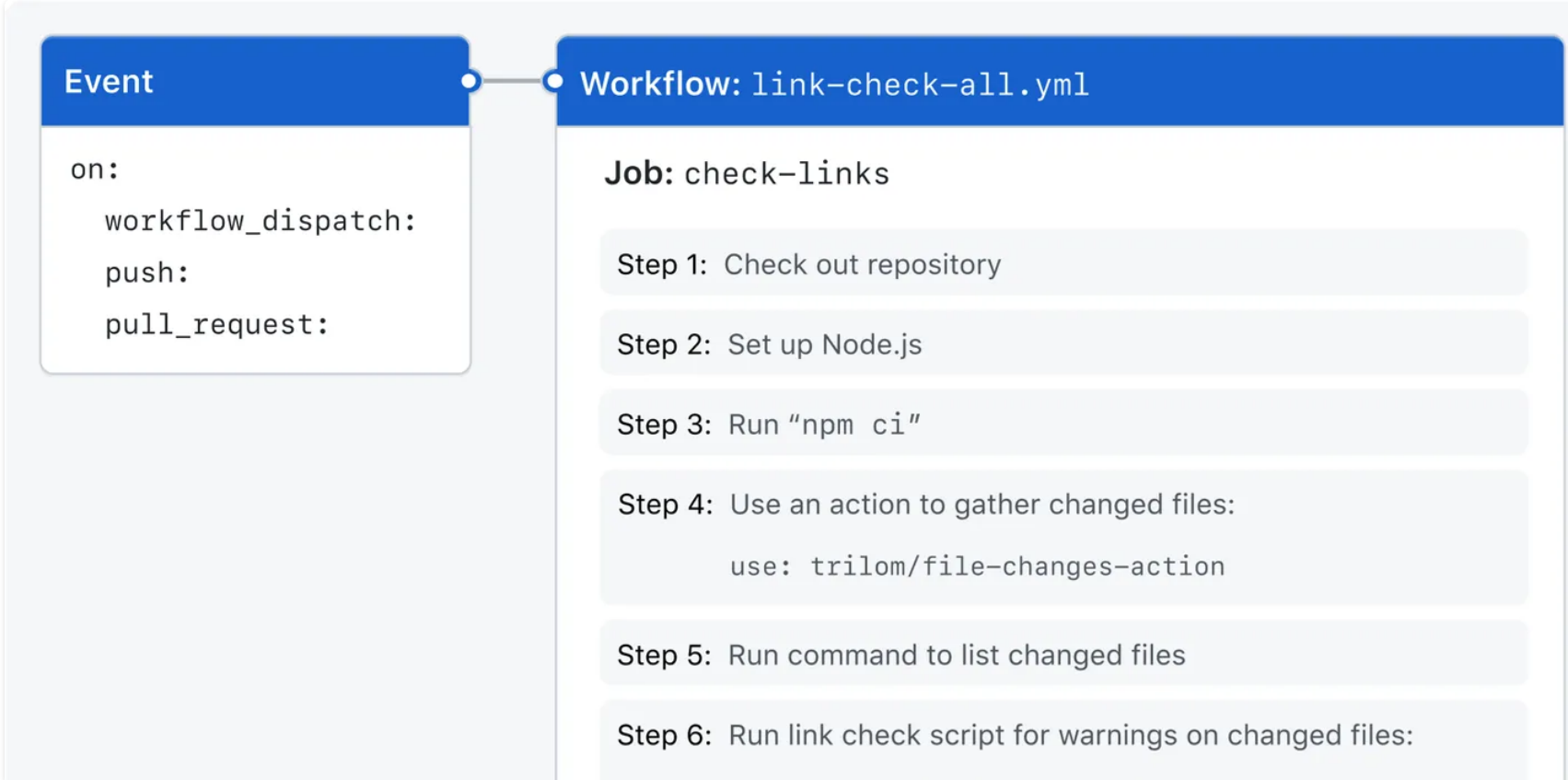

GitHub Actions & Runners
Presentation slides for developers
Press Space for next page →

Table of contents

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.

```yaml
---
name: ci
on:
  push:
    branches:
      - main
jobs:
  deploy:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v3
      - uses: actions/setup-python@v2
        with:
          python-version: 3.x
      - run: pip install mkdocs-material
      - run: pip install mkdocs-git-revision-date-plugin
      - run: mkdocs gh-deploy --force
```

Example Workflow Run In a Container

```yaml
name: ansible-and-yaml-linting

on:
  pull_request:
  workflow_dispatch:

jobs:
  liniting:
    runs-on: ubuntu-latest
    container:
      image: ghcr.io/idsys-unibe-ch/idsys.ansible:v1.0.0
      env:
        # These tell ansible-lint to use github compatible annotation format:
        GITHUB_ACTIONS: "true"
        GITHUB_WORKFLOW: "{{ github.workflow.name }}"

    steps:
      - name: Checkout repository ${{ github.repository.name }}
        uses: actions/checkout@v3
        with:
          fetch-depth: 0
        env:
          GITHUB_TOKEN: ${{ secrets.CI_BOT_PAT }}

       ...

```

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.

```yaml
on:
  workflow_dispatch:
    inputs:
      job:
        type: choice
        description: What job to run
        options:
        - ksldbtest1 ( kslp → ksldbtest1 )
                    ...
        - parisdbiam ( parisprod → parisdbiam )
jobs:
  clone_DB:
    runs-on: [self-hosted, ubuntu]
    env:
      SELECTION: ${{ github.event.inputs.job }}
    steps:
    - uses: actions/checkout@v3
    - name: Set up Python 3.10
      uses: actions/setup-python@v4
      with:
        python-version: '3.10'
```

Interactive Workflow With User Input (continued)

```
      - name: Install dependencies
        run: 'python -m pip install --upgrade pip

          if [ -f requirements.txt ]; then pip install -r requirements.txt; fi

          '
      - id: systems
        run: echo "systems=$(echo '${{ github.event.inputs.job }}' | cut -d ' ' -f1 )" >> $GITHUB_OUTPUT
      - id: originDB
        run: echo "originDB=$(echo '${{ github.event.inputs.job }}' | cut -d ' ' -f3 )" >> $GITHUB_OUTPUT
      - id: destinationDB
        run: echo "destinationDB=$(echo '${{ github.event.inputs.job }}' | cut -d ' ' -f5 )" >> $GITHUB_OUTPUT
      - id: OriginDBPWName
        run: echo "OriginDBPWName=${{ format('POSTGRES_DB_{0}_KEY', steps.originDB.outputs.originDB) }}"
          >> $GITHUB_OUTPUT
      - id: DestinationDBPWName
        run: echo "DestinationDBPWName=${{ format('POSTGRES_DB_{0}_KEY', steps.destinationDB.outputs.destinationDB)
          >> $GITHUB_OUTPUT
      - name: execute py script
        run: python main.py -p ${{ steps.systems.outputs.systems }} -opw "${{ secrets[steps.OriginDBPWName.outputs.O
          }}" -dpw "${{ secrets[steps.DestinationDBPWName.outputs.DestinationDBPWName]
          }}" -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 ressources (border firewall)

Self-hosted Runners

Docs on self-hosted Runners

Pros:

- Managed by ID => anything possible

Cons:

- Manged by ID =>

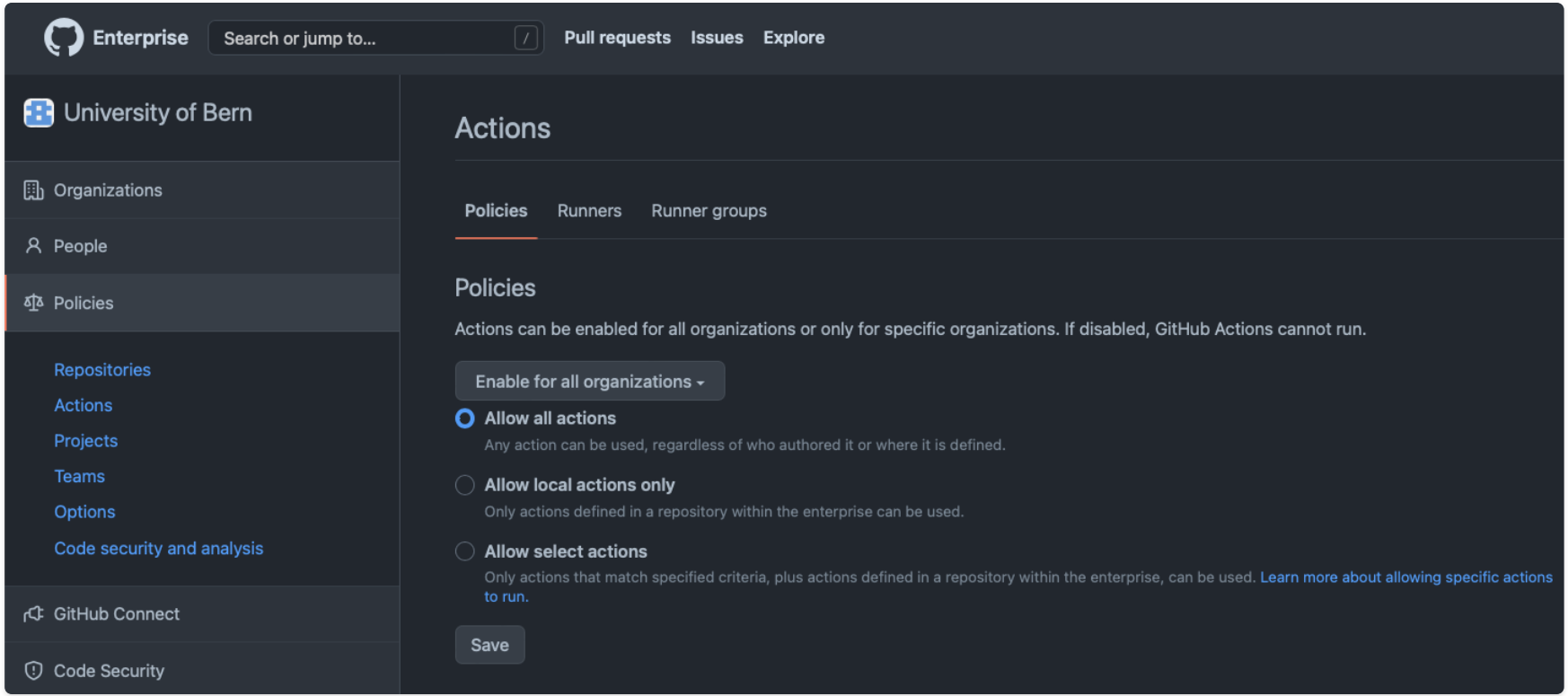
Current Runner State

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."

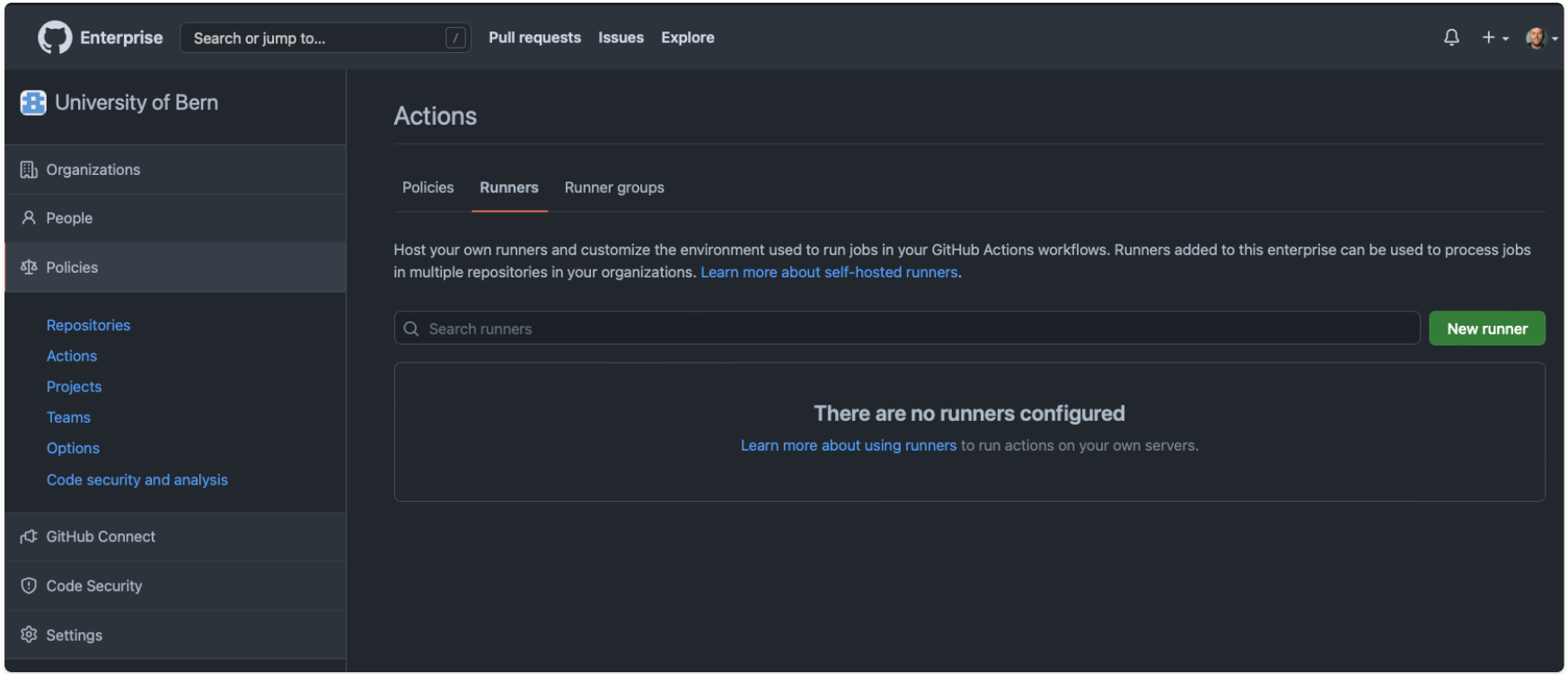
Demo:

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

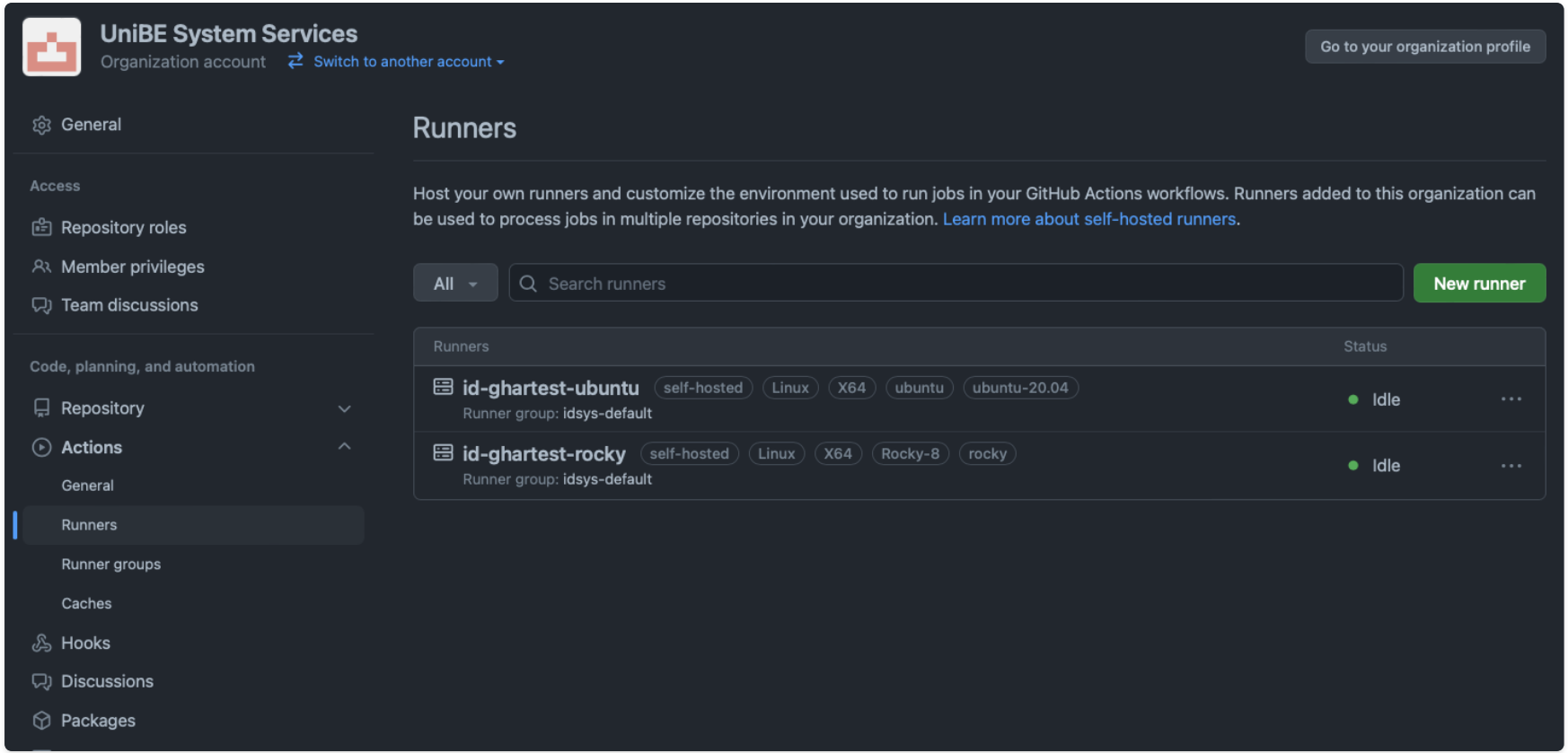
GHES: Action Policies



GHES: Self-hosted Enterprise Runners



GHES: Self-hosted Runners ID-SYS



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