



GitHub Actions

For **Enterprises** - ESDE Workshop

Duart Snel, David Greven

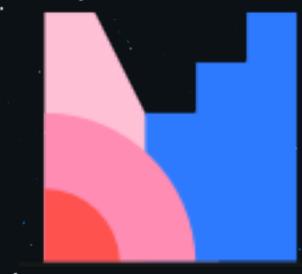
Structure

- What is GitHub?
- What are GitHub Actions?
- Terminology Introduction
- Workflow Basics
- Advanced Constructs
- Usage Examples
- Enterprise Features
- Are there Alternatives?
- Conclusion

A close-up photograph of a young man with dark hair and glasses, wearing a blue jacket over a red and white striped sweater. He is looking slightly to his left with a thoughtful expression, resting his chin on his hand. The background is blurred, showing what appears to be a workshop or studio environment.

What is GitHub?

GitHub Actions



Mentimeter

menti.com

<put menti code here>

What this workshop is NOT

GitHub Actions

Core features



Linux, macOS, Windows, ARM,
and containers



Matrix workflows that simultaneously test
across multiple operating systems and
versions of your runtime.



GitHub Actions supports Node.js, Python,
Java, Ruby, PHP, Go, Rust, .NET, and
more.

Terminology

Action

*A **program** that becomes a **reusable component** to be used in workflows. Actions can install software for the environment, set up authentication, or automate complex sets of tasks. You can find actions in the GitHub **Marketplace**, or create your own and **share them with your community**.*

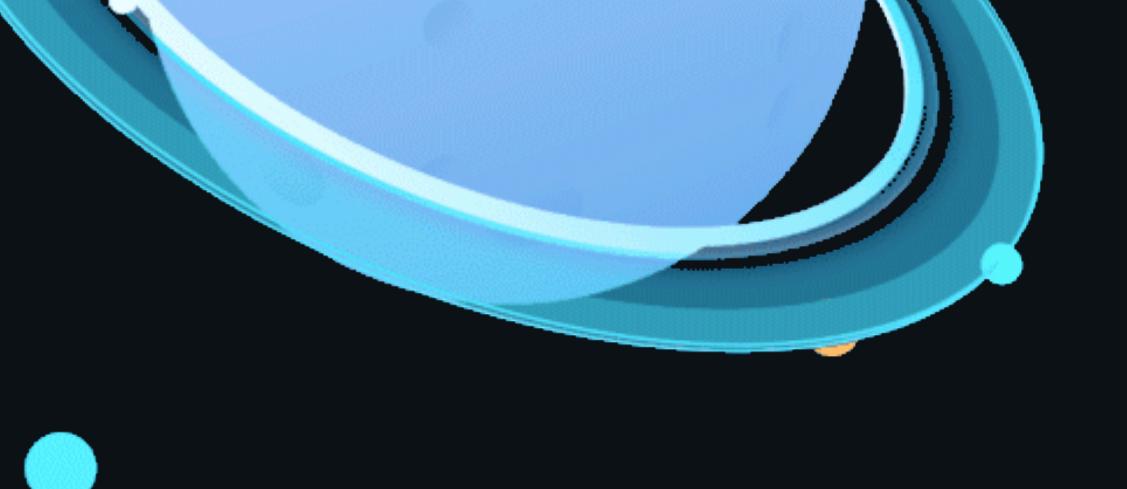
Workflow

*A **configurable**, automated **process** that you can use in your repository to build, test, package, release, or deploy your project. Workflows are made up of one or more “jobs” and can be **triggered** by GitHub events.*

Jobs

A list of the jobs that run as part of the workflow. Each job will run independently of the others, and will run on a different virtual environment. Jobs may have a name to make them easily identifiable in the UI. Jobs contain a set of steps that will be executed, in order.

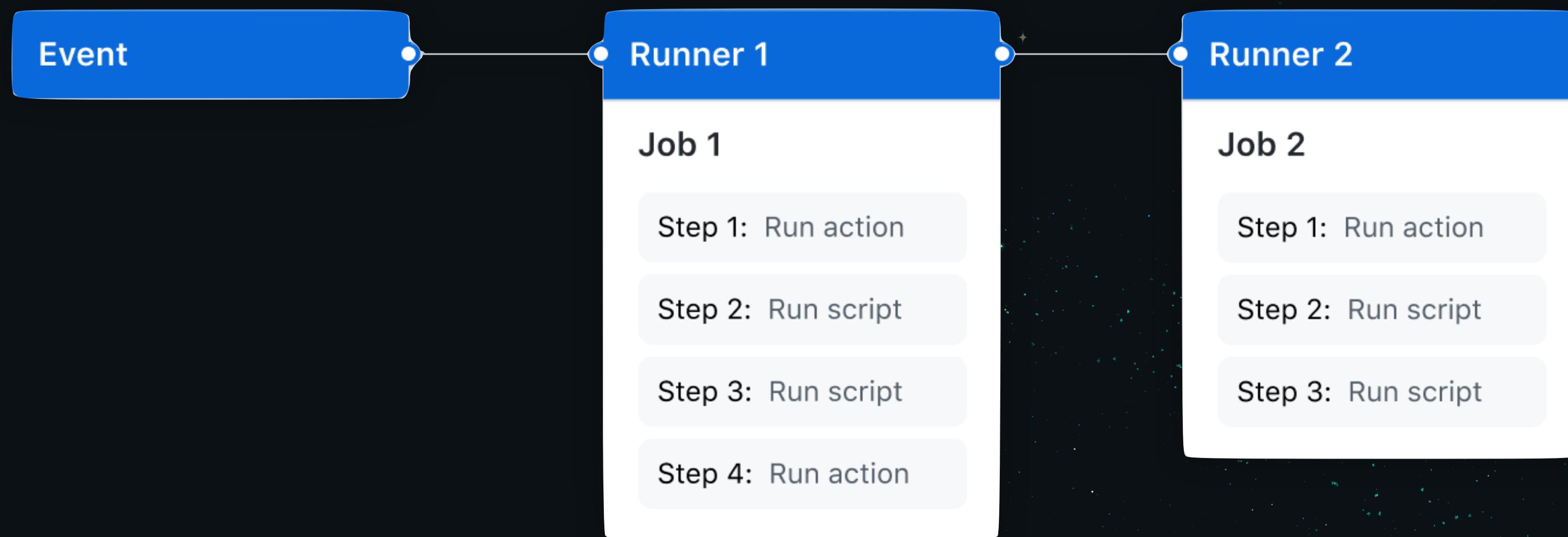
YAML



*YAML stands for “**Yet Another Markup Language**”. It’s a **human-readable** markup language commonly used for **configuration files**, especially by CI- and DevOps-focused software tools. GitHub Actions uses YAML as the basis of its configuration workflows.*

Workflow basics

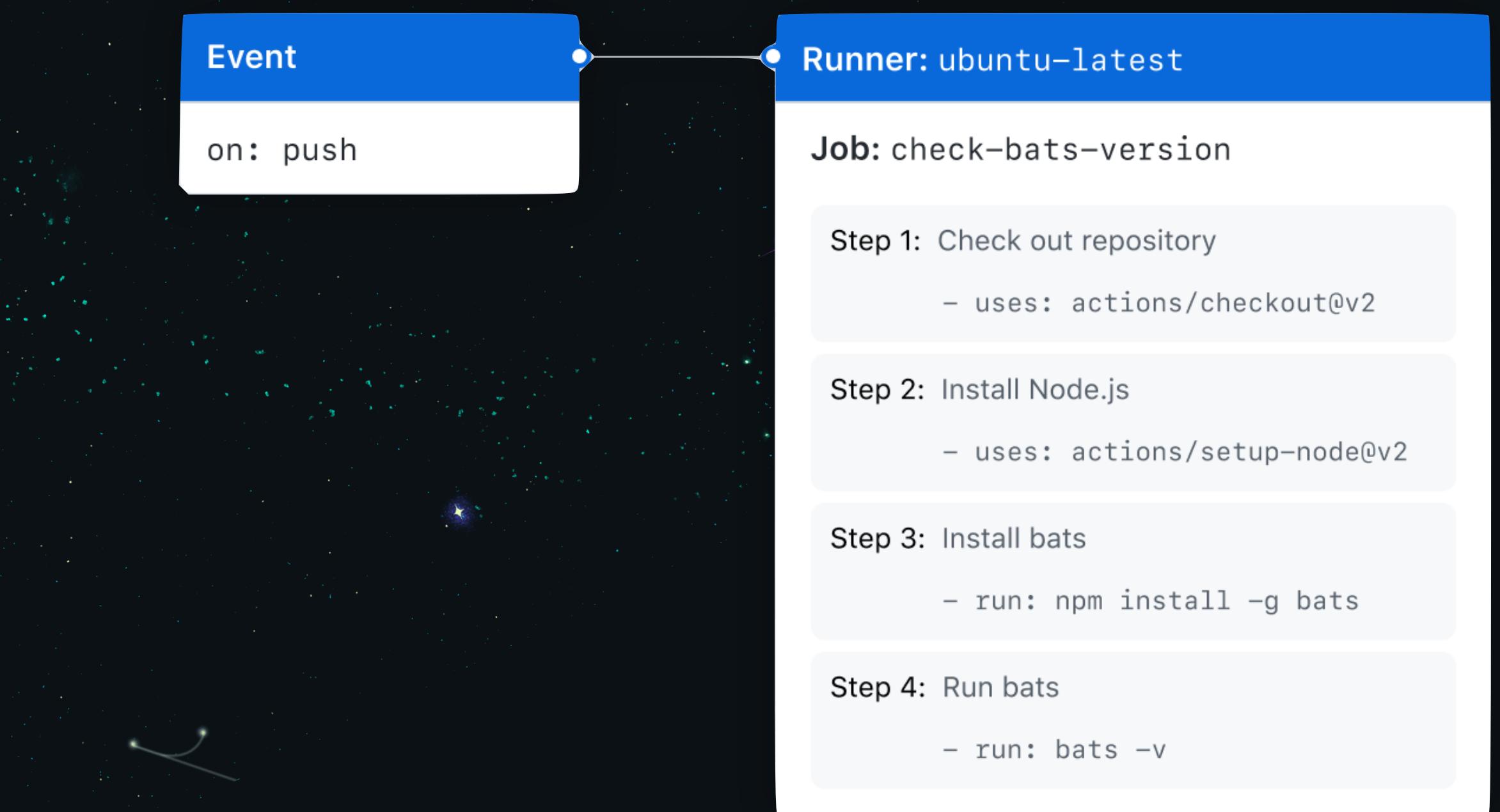
Visual representation of the structure



Workflow basics

Now the code

```
● ● ●  
name: learn-github-actions  
run-name: ${{ github.actor }} is learning GitHub Actions  
on: [push]  
jobs:  
  check-bats-version:  
    runs-on: ubuntu-latest  
    steps:  
      - uses: actions/checkout@v3  
      - uses: actions/setup-node@v3  
      with:  
        node-version: '14'  
      - run: npm install -g bats  
      - run: bats -v
```



Interactive - Hello World

Reusing workflows

Don't repeat yourself

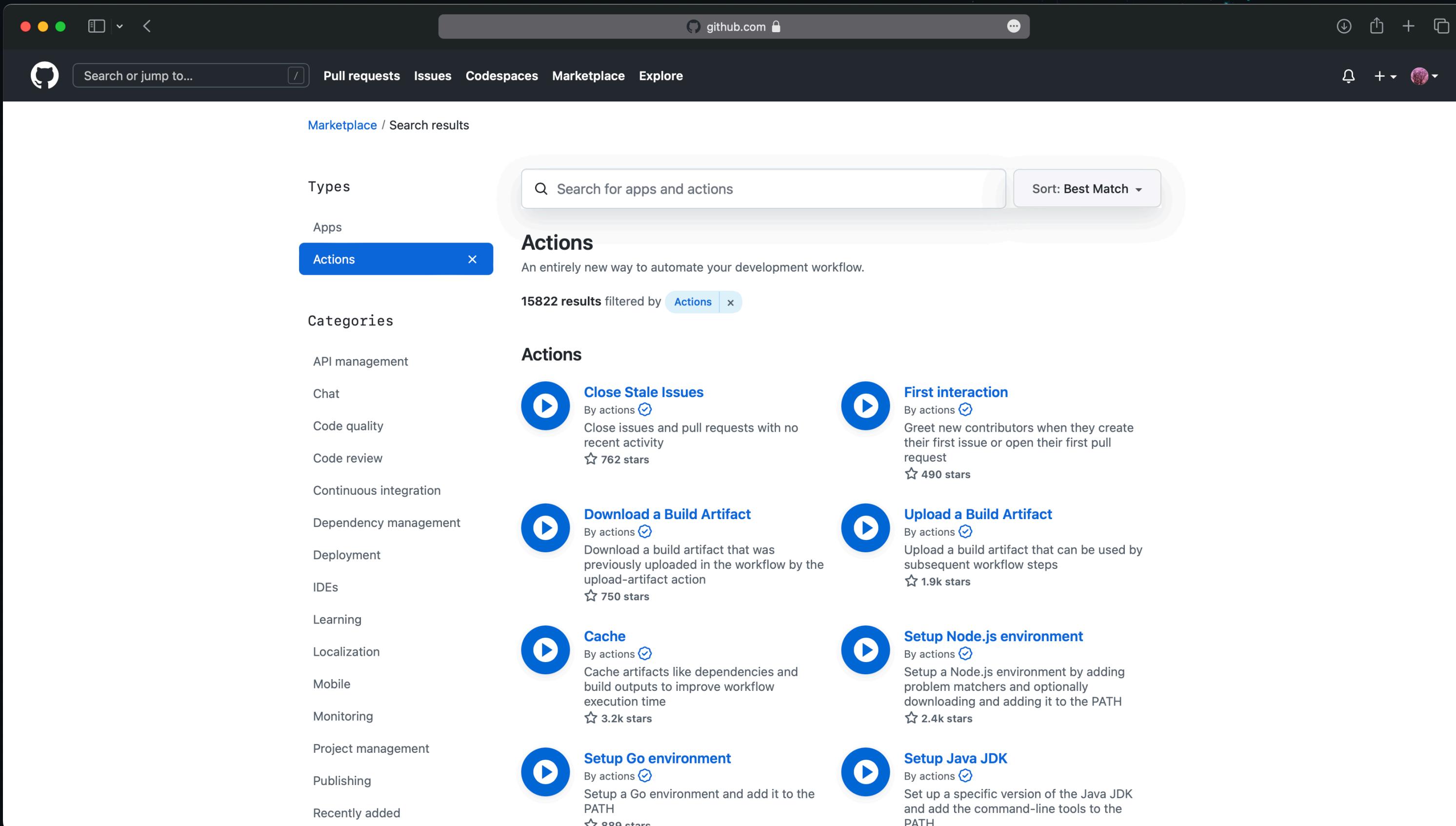
```
name: Reusable workflow example

on:
  workflow_call:
    inputs:
      config-path:
        required: true
        type: string
    secrets:
      token:
        required: true

jobs:
  triage:
    runs-on: ubuntu-latest
    steps:
    - uses: actions/llabeler@v4
      with:
        repo-token: ${{ secrets.token }}
        configuration-path: ${{ inputs.config-path }}
```

GitHub Marketplace

Community Actions



The screenshot shows a web browser window displaying the GitHub Marketplace. The URL in the address bar is `github.com`. The page title is "Marketplace / Search results". The left sidebar lists categories: Types (Apps, Actions), Categories (API management, Chat, Code quality, Code review, Continuous integration, Dependency management, Deployment, IDEs, Learning, Localization, Mobile, Monitoring, Project management, Publishing, Recently added). The main content area is titled "Actions" with the sub-subtitle "An entirely new way to automate your development workflow." It shows 15822 results filtered by "Actions". A search bar at the top right says "Sort: Best Match". Below the search bar, there are two columns of action cards:

Action	Description	Stars
Close Stale Issues By actions	Close issues and pull requests with no recent activity	762 stars
First interaction By actions	Greet new contributors when they create their first issue or open their first pull request	490 stars
Download a Build Artifact By actions	Download a build artifact that was previously uploaded in the workflow by the upload-artifact action	750 stars
Upload a Build Artifact By actions	Upload a build artifact that can be used by subsequent workflow steps	1.9k stars
Cache By actions	Cache artifacts like dependencies and build outputs to improve workflow execution time	3.2k stars
Setup Node.js environment By actions	Setup a Node.js environment by adding problem matchers and optionally downloading and adding it to the PATH	2.4k stars
Setup Go environment By actions	Setup a Go environment and add it to the PATH	889 stars
Setup Java JDK By actions	Set up a specific version of the Java JDK and add the command-line tools to the PATH	

Source: <https://github.com/marketplace?category=&query=&type=actions>

Inter Job Dependencies

Needs successful completion



```
jobs:  
  job1:  
  job2:  
    needs: job1  
  job3:  
    needs: [job1, job2]
```

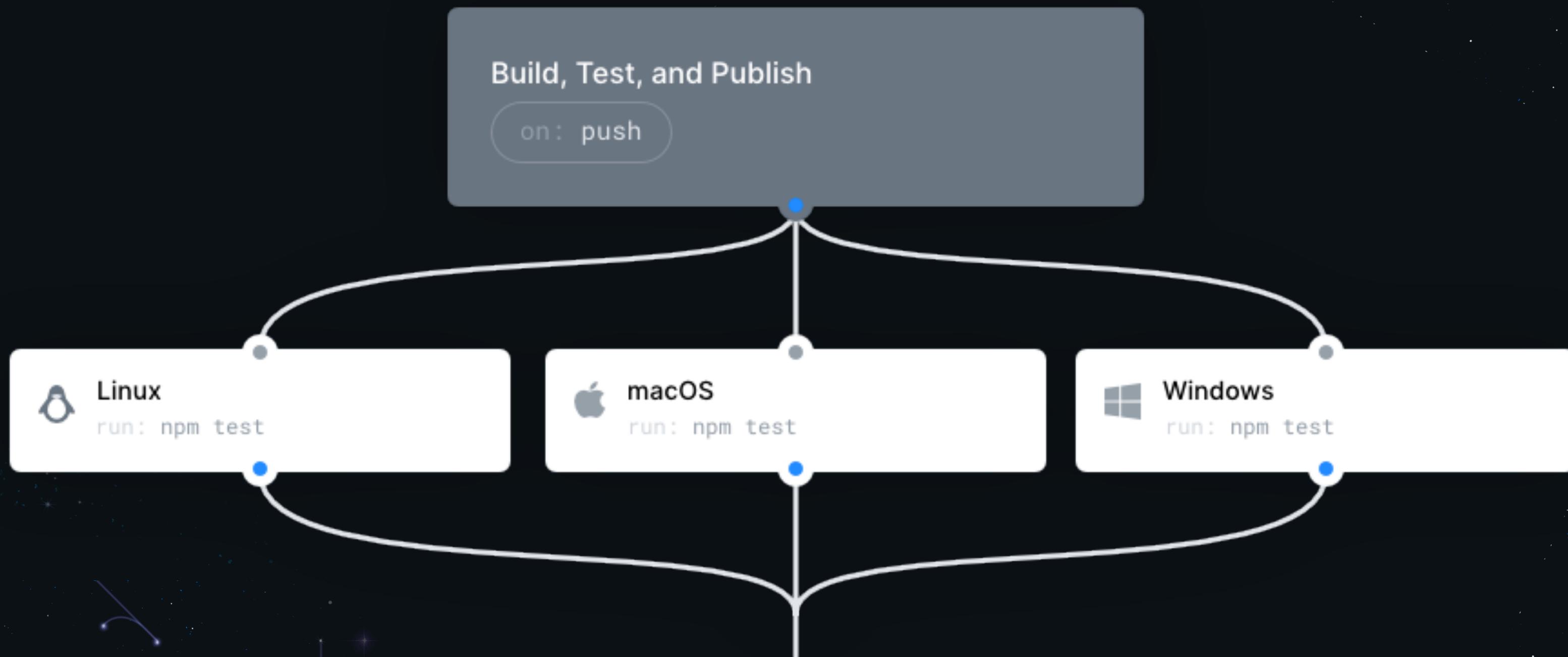
1 job1

2 job2

3 job3

Matrix Strategies

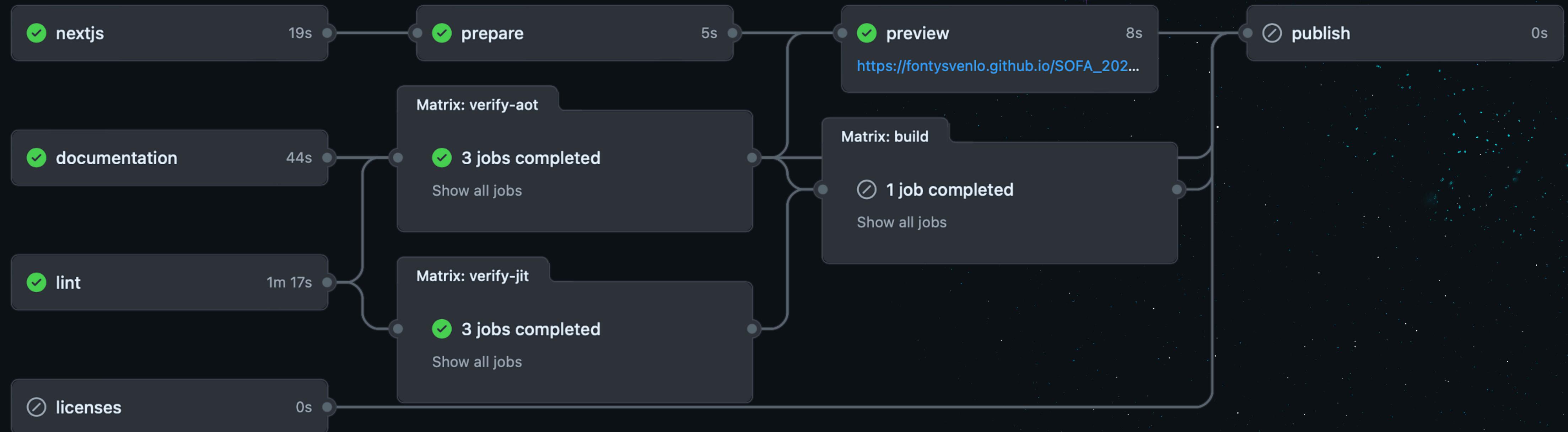
Multi Platform Actions



Are there any other features?

Robotica - SoFa

Composable pipeline



FOSS Plant Insights - PIOT

Open Source Intelligence using Flat Data



```
{  
    "roomTemp": 21.4,  
    "roomHumidity": 89,  
    "timestamp": 1668342997,  
    "light": {  
        "visible": 15.33333333333334,  
        "ultraviolet": 0.1,  
        "infrared": 48,  
        "isOn": true  
    },  
    "hasWater": false,  
    "isWatering": false,  
    "plants": [  
        {  
            "name": "Peper Anaheim",  
            "soilTemp": 85,  
            "soilMoisture": 1732  
        },  
        {  
            "name": "Peper Anaheim",  
            "soilTemp": 20.625,  
            "soilMoisture": 1655  
        }  
    ]  
}
```

Shamir's Secret Sharing Scheme

High performance impl. with hw. accelerated number generation

The screenshot shows a Mac OS X browser window with the Deno.land URL in the address bar. The main content is a module page for 'x/shamirs_secret_sharing@1.0.2'. The page includes a module summary, documentation links, source code links, repository information, and a list of versions. A detailed description of the module's purpose and implementation is provided in a box. The Deno logo is visible in the top left corner of the page content.

x/shamirs_secret_sharing@1.0.2

Performant secret sharing scheme implementation based on polynomial interpolation over finite fields

[View Documentation](#) [View Source](#)

Repository [grevend/shamirs-secret-sharing](#)

Current version released 2 minutes ago

Versions

1.0.2 <small>Latest</small>
1.0.1
1.0.0
0.0.0

Shamir's Secret-Sharing Scheme

A preponderance of the open-source Shamir's Secret-Sharing Scheme implementations operate on a finite field \mathbb{F}_q , where q is dependent on both the secret size as well as the number of shares, thus requiring both the search for a sufficiently large prime as well as the support of a large enough number representation or chunking of the secret. This implementation outlines an optimized low-level C and inline assembly-based alternative that uses the Galois field \mathbb{F}_{2^8} , referred to as GF(256), along with a Reed-Solomon inspired encoding approach and a hardware-accelerated entropy sourced secure number generator to offer the theoretical capacity to process secrets up to 8 Exabytes.

Why Deno?

- Develop Locally
- Deploy Globally
- Compare to Node.js

Products

- Deno CLI
- Deno Deploy
- Deploy Subhosting

Sources

- CLI Manual
- CLI Runtime API
- Deploy Docs

Community

- Artworks
- Translations
- Showcase

Company

- Blog
- Pricing
- News

All systems operational

Copyright © 2022 Deno Land Inc.
All rights reserved.

Source: https://deno.land/x/shamirs_secret_sharing@1.0.2

Interactive - Fix Repository

Kahoot!

kahoot.it

<put kahoot code here>

Governance & Compliance

Policy governed capabilities

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 can be used.

Save

Secrets & Deployment Resources

Security hardening workflows

The screenshot shows a list of secrets in a GitHub interface. There are two entries:

- EXAMPLE_API_KEY**: Available to private repositories. Last updated 21 hours ago. Actions: Update (blue), Remove (red).
- EXAMPLE_API_KEY2**: Available to all repositories. Last updated 17 days ago. Actions: Update (blue), Remove (red).

Permissions

GITHUB_TOKEN

```
permissions:  
  actions: read|write|none  
  checks: read|write|none  
  contents: read|write|none  
  deployments: read|write|none  
  id-token: read|write|none  
  issues: read|write|none  
  discussions: read|write|none  
  packages: read|write|none  
  pages: read|write|none  
  pull-requests: read|write|none  
  repository-projects: read|write|none  
  security-events: read|write|none  
  statuses: read|write|none
```

Secret Scanning

Partner patterns

Partner	Supported secret
Adafruit IO	Adafruit IO Key
Adobe	Adobe Device Token
Adobe	Adobe Service Token
Adobe	Adobe Short-Lived Access Token
Adobe	Adobe JSON Web Token
Alibaba Cloud	Alibaba Cloud Access Key ID and Access Key Secret pair
Amazon Web Services (AWS)	Amazon AWS Access Key ID and Secret Access Key pair
Atlassian	Atlassian API Token
Atlassian	Atlassian JSON Web Token



Managed service offering



Granular policies



Usage tracking



Log forwarding



Secret scanning



Autoscaling runners



Containerized builds



Environments



Marketplace



Permissions



Innersourcing



Vulnerability detection



- Managed service offering
- Granular policies
- Usage tracking
- Log forwarding
- Secret scanning
- Autoscaling runners
- Containerized builds
- Environments
- Marketplace
- Permissions
- Innersourcing
- Vulnerability detection





Managed service offering



Granular policies



Usage tracking



Log forwarding



Secret scanning



Autoscaling runners



Containerized builds



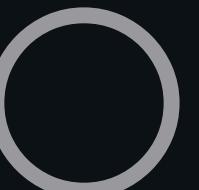
Environments



Marketplace



Permissions



Innersourcing



Vulnerability detection



Conclusion





Questions?

Feedback

