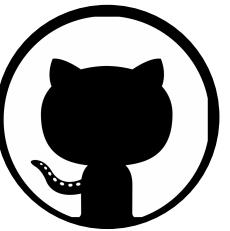


GitHub Actions

{ Kick Start }

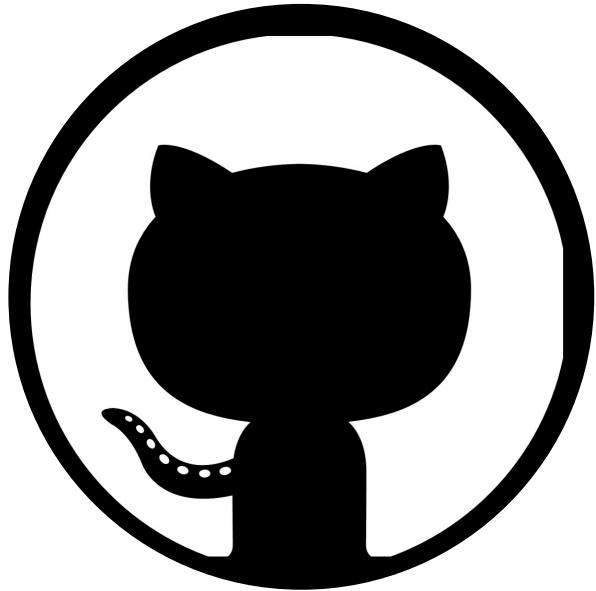
Michael Kaufmann



June 2018 **\$7.5 billion**



A different approach...



- ▶ Leverage the power of the community
- ▶ Platform
- ▶ Flexible
- ▶ Simple



The GitHub marketplace

Marketplace / Search results

Types

Actions

Search for apps and actions

Sort: Best Match

Actions

An entirely new way to automate your development workflow.

12769 results filtered by Actions

Categories

- API management
- Chat
- Code quality
- Code review
- Continuous integration
- Dependency management
- Deployment
- IDEs

First interaction
By actions ⚡
Greet new contributors when they create their first issue or open their first pull request
★ 140 stars

Close Stale Issues
By actions ⚡
Close issues and pull requests with no recent activity
★ 587 stars

Setup Go environment
By actions ⚡
Setup a Go environment and add it to the PATH
★ 694 stars

Download a Build Artifact
By actions ⚡
Download a build artifact that was previously uploaded in the workflow by the upload-artifact action
★ 539 stars

Where the world **builds** software

73M+

Developers

200M+

Repositories

1,000s

Open-Source Communities

2.6B+

Contributions / Year

4M+

Organizations

84%

Fortune 500 companies

GitHub Actions - More than CI/CD

Automate everything with workflows

35 events can trigger a workflow

GitHub Token and Workflow Permissions

Community-powered workflows

Any platform, any language, any cloud

Workflows

► A text file in your repository
.github/workflows)

► YAML Ain't Markup Language (**YAML**)

► Events trigger workflows (**on:**)

► One or multiple **jobs**

► Executed on a **runner**

► Contains **steps**

► A reusable step is **action**

AccelerateDevOps / .github / workflows / pr-validation.yml in main

```
<> Edit file ⌂ Preview changes Spaces 2 No wrap
1 name: PR-Validation
2
3 on: [pull_request]
4
5 jobs:
6   Build:
7     runs-on: ubuntu-latest
8
9   steps:
10  - name: 'Checkout Github Action'
11    uses: actions/checkout@master
12
13  - name: Set up .NET Core
14    uses: actions/setup-dotnet@v1
15    with:
16      dotnet-version: '5.0.x'
17
18  - name: Setup Node
19    uses: actions/setup-node@v2.5.1
20    with:
21      node-version: 10.16.3
22
23  - name: Install dependencies in client app
24    working-directory: src/Tailwind.Traders.Web/ClientApp
25    run: npm install
26
27  - name: Build and publish with dotnet
28    working-directory: src/Tailwind.Traders.Web
29    run: |
30      dotnet build --configuration Release|
```

Use Control + Space to trigger autocomplete in most situations.

YAML

YAML basics

Extension: .yml or .yaml

Data-serialization language
writable and readable by humans

A strict superset of **JSON**

Contains syntactically relevant
newlines and **indentation** instead
of braces

YAML basics

```
1 # This is a comment in yaml
2
3 # Scalar types:
4 key: value
5
6 # Data types:
7 integer: 42
8 float: 42.0
9 string: a text value
10 boolean: true
11 null value: null
12 datetime: 1999-12-31T23:59:43.12
13
14 # Keys and values can contain spaces and do not need quotation.
15 # You can quote both with single or double quotes:
16 'single quotes': 'have ''one quote'' as the escape pattern'
17 "double quotes": "have the \"backslash \\\" escape pattern"
18
19 # Literal blocks:
20 literal_block: |
21     Text blocks use 4 spaces as indentation. The entire
22     block is assigned to the key 'literal_block' and keeps
23     line breaks and empty lines.
24
25     The block continues until the next element.
26
```

```
27 # Collection types
28
29 # Maps
30 # Maps use 2 spaces of indentation:
31 nested_type:
32     key1: value1
33     key2: value2
34     another_nested_type:
35         key1: value1
36
37 #JSON syntax:
38 map: {key: value}
39
40 # Sequence
41 # Uses a dash before each item:
42 sequence:
43     - item1
44     - item2
45
46 #JSON syntax:
47 sequence: [item1, item2, item3]
```

Workflow syntax

Name and Triggers

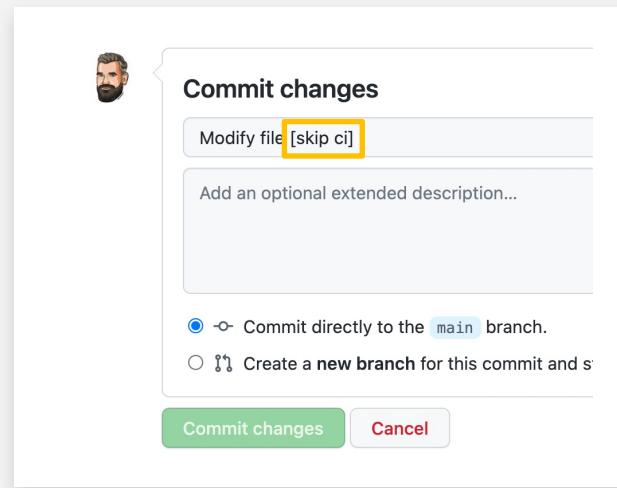


Name of the workflow



Trigger:

- › Webhook events
- › Scheduled events
- › Manual events



AccelerateDevOps / .github / workflows / starter.yml

in `main`

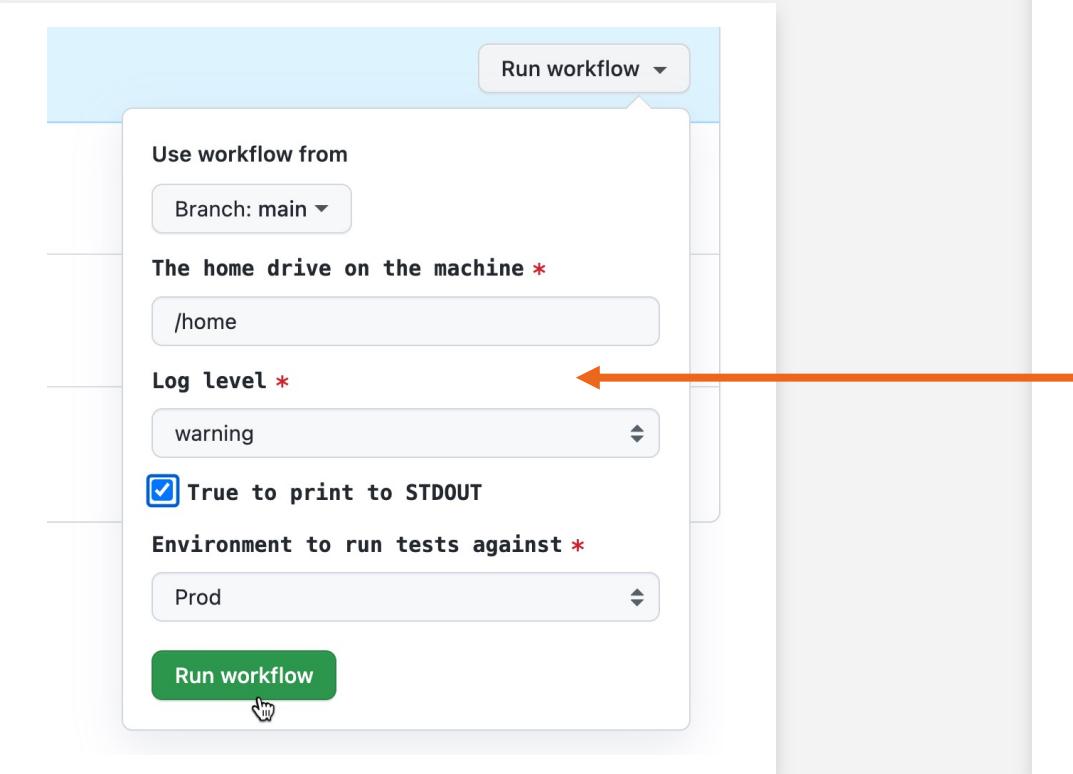
<> Edit file

Preview changes

```
1 name: Starter Workflow
2
3 on:
4   # Webhook events
5   push:
6     branches:
7       - main
8   issues:
9     types: [opened, edited, milestoned]
10
11 # Scheduled events
12
13
14   Runs every 15 minutes.
15   Actions schedules run at most every 5 minutes. Learn more
16   schedule:
17     - cron: '*/15 * ? * *'
18     - cron: '0 9-17 * * *'
19     - cron: '11 11 * * 5'
20
21 # Manual events
22 workflow_dispatch:
23   inputs:
24     homedrive:
25       description: 'The home drive on the machine'
26       required: true
27       default: '/home'
```

Name and Triggers

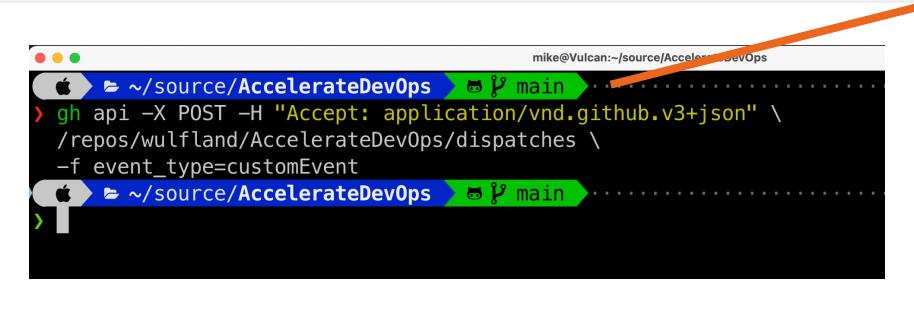
► Manual events



```
# Manual events
workflow_dispatch:
  inputs:
    homedrive:
      description: 'The home drive on the machine'
      required: true
      default: '/home'
    logLevel:
      description: 'Log level'
      required: true
      default: 'warning'
      type: choice
      options:
        - info
        - warning
        - debug
    print_tags:
      description: 'True to print to STDOUT'
      required: true
      type: boolean |
environment:
  description: 'Environment to run tests against'
  type: environment
  required: true
```

Name and Triggers

- ▶ Manual events: trigger using the API
(curl, octokit, GitHub CLI)



A screenshot of a macOS terminal window titled "main". The command entered is:

```
mike@Vulcan:~/source/AccelerateDevOps$ gh api -X POST -H "Accept: application/vnd.github.v3+json" \
/repos/wulfland/AccelerateDevOps/dispatches \
-f event_type=customEvent
```

```
# Trigger using the API
repository_dispatch:
  types: [customEvent]

# Call for example using GitHub CLI:
# $ gh api -X POST -H "Accept: application/vnd.github.v3+json" \
# /repos/wulfland/AccelerateDevOps/dispatches \
# -f event_type=customEvent
```

Demo

Strategy

- › For-loop - array
- › Nested for-loops:
multidimensional array
- › Runs for all combinations in all dimensions
- › Fail-fast (yes/no)
- › May parallel jobs

```
matrix_job:  
  name: matrix-job  
  runs-on: ${{ matrix.runner }}  
  if: github.event.inputs.run_matrix  
  
  strategy:  
    matrix:  
      runner: [ubuntu-18.04, ubuntu-20.04]  
      node: [10,12]  
  
    steps:  
      - run: echo "${{ matrix.runner }}"  
      - run: echo "${{ matrix.node }}"
```

✓ Starter Workflow Starter Workflow #17

Summary

Jobs

- ✓ job_1
- ✓ matrix-job (ubuntu-18.04, 10)
- ✓ matrix-job (ubuntu-18.04, 12)
- ✓ matrix-job (ubuntu-20.04, 10)
- ✓ matrix-job (ubuntu-20.04, 12)

matrix-job (ubuntu-20.04, 12)
succeeded now in 0s

- > ✓ Set up job
- > ✓ Run echo "ubuntu-20.04"
- > ✓ Run echo "12"
- > ✓ Complete job

Workflow jobs

- ▶ Map - run in parallel by default
-

- ▶ Can be chained using needs keyword
-

- ▶ Runs on a **runner** in one process
-

- ▶ Contains a sequence of steps

```
jobs:  
  job_1:  
    runs-on: ubuntu-latest  
  
    steps:  
      - run: echo "🎉 The job was triggered by a ${{ github.event_name }} event."  
      - run: echo "🏡 drive is `${{ github.event.inputs.homedrive }}`."  
      - run: echo "🌐 environment is `${{ github.event.inputs.environment }}`."  
      - run: echo "💻 log level is `${{ github.event.inputs.logLevel }}`."  
      - run: echo "🛠 Run the matrix? `${{ github.event.inputs.run_matrix }}`."  
  
  job_2:  
    runs-on: ubuntu-latest  
    needs: job_1  
    steps:  
      - run: echo "Status ${{ job.status }}"  
  
  job_3:  
    runs-on: ubuntu-latest  
    needs: job_1  
    steps:  
      - run: echo "Services ${{ job.services }}"  
  
  job_4:  
    runs-on: ubuntu-latest  
    needs: [job_2, job_3]  
    steps:  
      - run: echo "Status ${{ job.status }}"
```

Workflow steps

Sequence in a job

Run in the same process / same directory

Run in a shell

```
Get OS information
  1 ► Run import platform
  5 Linux-5.13.0-1021-azure-x86_64-with-glibc2.29

> Run actions/checkout@v3.0.0

Display documentation
  1 ► Run tree
  4 .
  5   ├── About.md
  6   ├── _config.yml
  7   ├── _posts
  8   |   ├── 2021-08-10-posting-source-code.md
  9   |   ├── 2021-08-12-writing-with-markdown.md
 10  |   └── 2021-08-13-posting-in-jekyll.md
 11  ├── about-markdown.md
 12  ├── get-started.md
 13  └── index.md

 15 1 directory, 8 files
```

| Parameter | Description |
|-------------------|--|
| bash | Bash shell. The default shell on all non-Windows platforms with a fallback to sh. When specified on Windows, the bash shell included with Git is used. |
| pwsh | PowerShell Core. Default on the Windows platform. |
| python | The python shell. Allows you to run python scripts |
| cmd | Windows only! The windows command prompt. |
| powershell | Windows only! The classical Windows PowerShell. |

```
job_1:
  runs-on: ubuntu-latest

  steps:
    - run: echo "🎉 The job was triggered by a ${{ github.event_name }} event."
    - run: echo "🏡 drive is '${{ github.event.inputs.homedrive }}'."
    - run: echo "🌐 environment is '${{ github.event.inputs.environment }}'."
    - run: echo "LogLevel is '${{ github.event.inputs.logLevel }}'."
    - run: echo "♾️ Run the matrix? '${{ github.event.inputs.run_matrix }}'."

    - name: Get OS information
      run: |
        import platform
        print(platform.platform())
      shell: python

    - uses: actions/checkout@v3.0.0
    - name: Display documentation
      run: tree
      working-directory: docs|
```

Actions

- ▶ A reusable step

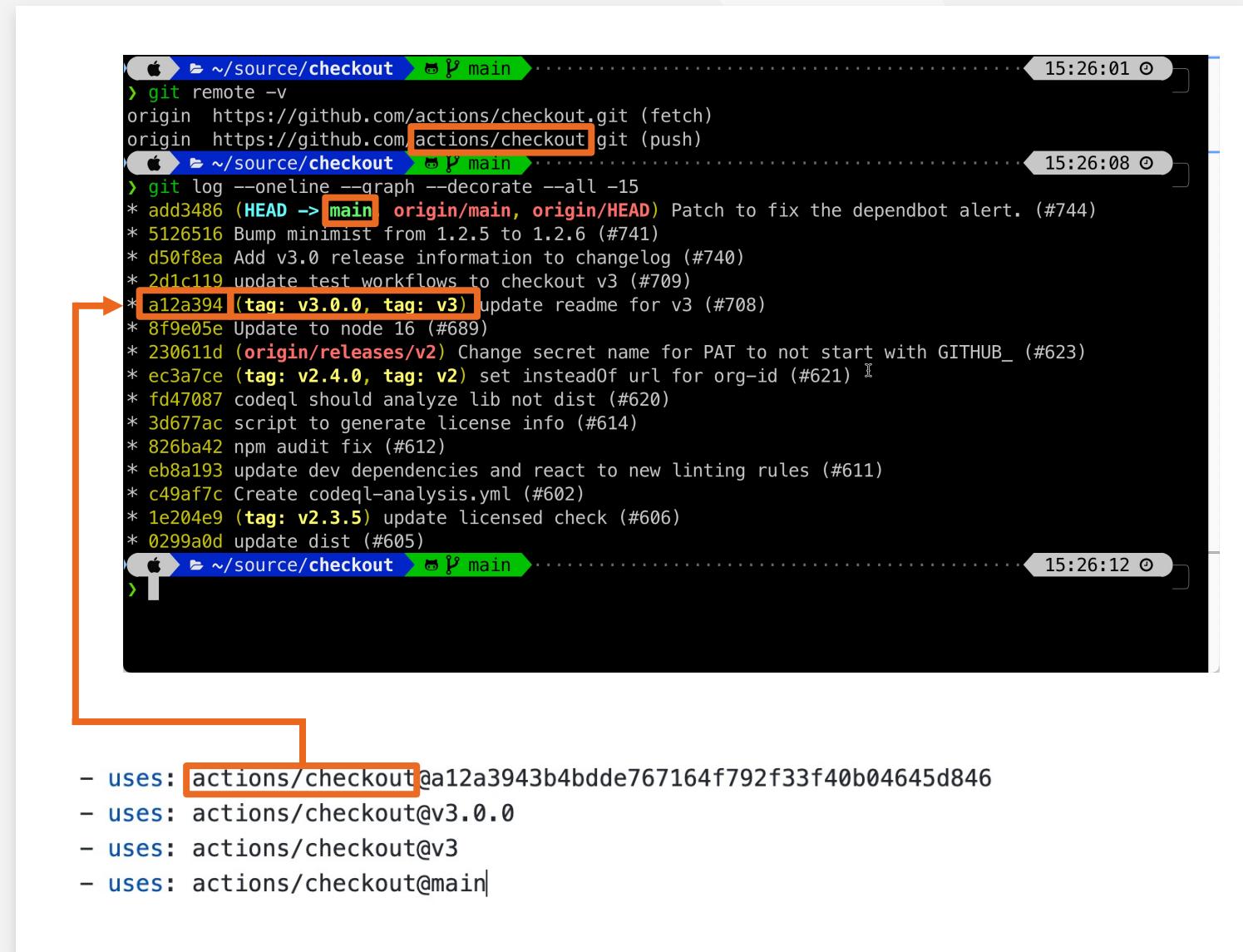
- ▶ Lives in a git repo

Syntax:

- › `{owner}/{repo}@{ref}`
- › `{owner}/{repo}/{path}@{ref}`
- › `./github/actions/my-action`

References:

- › SHA
- › Tag
- › Branch



```
git remote -v
origin https://github.com/actions/checkout.git (fetch)
origin https://github.com/actions/checkout.git (push)
git log --oneline --graph --decorate --all -15
* add3486 (HEAD -> main origin/main, origin/HEAD) Patch to fix the dependbot alert. (#744)
* 5126516 Bump minimist from 1.2.5 to 1.2.6 (#741)
* d50f8ea Add v3.0 release information to changelog (#740)
* 2d1c119 update test workflows to checkout v3 (#709)
* a12a394 (tag: v3.0.0, tag: v3) update readme for v3 (#708)
* 8f9e05e Update to node 16 (#689)
* 230611d (origin/releases/v2) Change secret name for PAT to not start with GITHUB_ (#623)
* ec3a7ce (tag: v2.4.0, tag: v2) set insteadOf url for org-id (#621)
* fd47087 codeql should analyze lib not dist (#620)
* 3d677ac script to generate license info (#614)
* 826ba42 npm audit fix (#612)
* eb8a193 update dev dependencies and react to new linting rules (#611)
* c49af7c Create codeql-analysis.yml (#602)
* 1e204e9 (tag: v2.3.5) update licensed check (#606)
* 0299a0d update dist (#605)
```

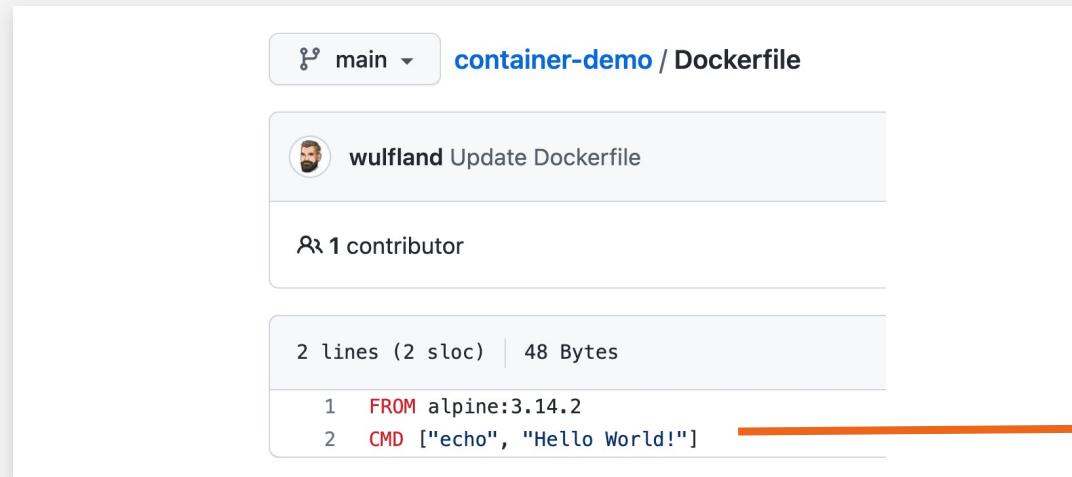
- `uses: actions/checkout@a12a3943b4bdde767164f792f33f40b04645d846`
- `uses: actions/checkout@v3.0.0`
- `uses: actions/checkout@v3`
- `uses: actions/checkout@main`

Actions

► User docker images as actions

```
- name: Run a docker containers as an action
  uses: docker://alpine:3.8

- uses: docker://ghcr.io/wulfland/container-demo:latest
```



A screenshot of a GitHub Actions workflow step titled 'Run ghcr.io/wulfland/container-demo:latest'. The step details the command run:

- ▶ Run docker://ghcr.io/wulfland/container-demo:latest
- /usr/bin/docker run --name ghcriowulflandcontainerdemolate
e GITHUB_RUN_ID -e GITHUB_RUN_NUMBER -e GITHUB_RETENTION_D
GITHUB_API_URL -e GITHUB_GRAPHQL_URL -e GITHUB_REF_NAME -e
GITHUB_PATH -e GITHUB_ENV -e GITHUB_STEP_SUMMARY -e RUNNER
ACTIONS_CACHE_URL -e GITHUB_ACTIONS=true -e CI=true -v "/v
"/home/runner/work/_temp/_github_workflow":"/github/workfl
ghcr.io/wulfland/container-demo:latest
- Hello World!

Demo

Contexts and expressions syntax

Contexts and expressions syntax

▶ `$(<expression>)`

▶ **context syntax:**

- › `context['key']` (if key starts with number or contains special characters)
- › `context.key`

▶ **Context:**

- › `matrix`
- › `github`
- › `env`
- › `runner`

```
steps:  
  - name: Dump runner context  
    run: echo '${{ toJSON(runner) }}'  
  - name: Dump GitHub context  
    run: echo '${{ toJSON(github) }}'
```

▼ **Dump runner context**

```
1 ► Run echo '{  
11 {  
12   "os": "Linux",  
13   "arch": "X64",  
14   "name": "GitHub Actions 2",  
15   "tool_cache": "/opt/hostedtoolcache",  
16   "temp": "/home/runner/work/_temp",  
17   "workspace": "/home/runner/work/AccelerateDevOps"  
18 }
```

Context
succeeded 1 minute ago in 3s

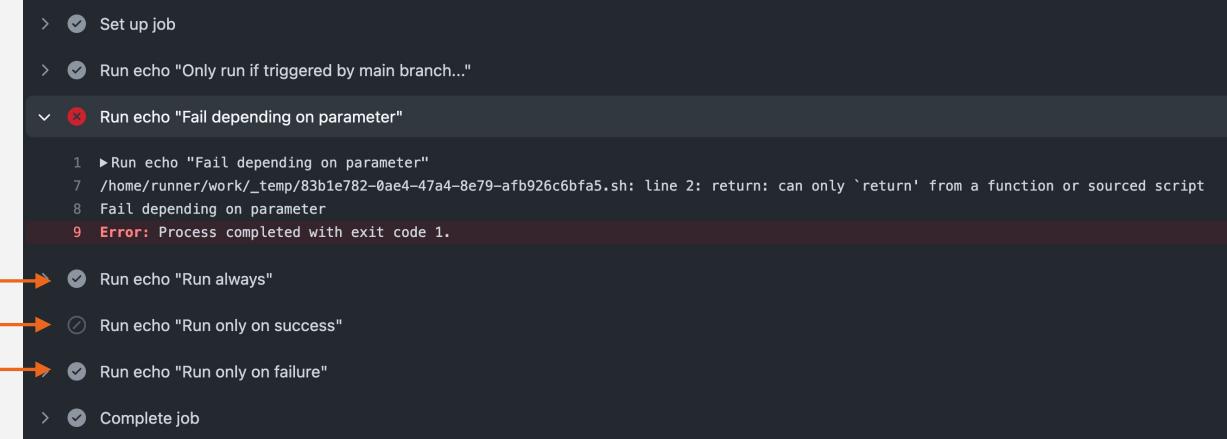
► **Set up job**

▼ **Dump GitHub context**

```
1 ► Run echo '{  
177 {  
178   "token": "***",  
179   "job": "context_job",  
180   "ref": "refs/heads/main",  
181   "sha": "c610cff739f85138a175c892651d204e71cedb43",  
182   "repository": "wulfland/AccelerateDevOps",  
183   "repository_owner": "wulfland",  
184   "repository_owner_id": "5276337",  
185   "repositoryUrl": "git://github.com/wulfland/AccelerateDevOps.git",  
186   "run_id": "2161816664",  
187   "run_number": "32",  
188   "retention_days": "90",  
189   "run_attempt": "1",  
190   "artifact_cache_size_limit": "10",  
191   "repository_id": "383720539",  
192   "actor_id": "5276337",  
193   "actor": "wulfland",  
194   "workflow": "Starter Workflow",  
195   "head_ref": "",  
196   "base_ref": "",  
197   "event_name": "workflow_dispatch",  
198   "event": {  
199     "inputs": {  
200       "environment": "github-pages",  
201       "homedrive": "/home",  
202       "logLevel": "warning",  
203       "run_matrix": "false"
```

Contexts and expressions syntax

```
expression_job:  
  runs-on: ubuntu-latest  
  name: Expressions  
  if: ${{ github.ref == 'refs/heads/main' && github.event.inputs.logLevel == 'debug' }}  
  
  steps:  
    - run: echo "Only run if triggered by main branch..."  
      if: contains(github.ref, 'main')  
  
    - run: |  
        echo "Fail depending on parameter"  
        return 1  
      if: github.event.inputs.run_matrix == 'true'  
  
    - run: echo "Run always"  
      if: always()  
    - run: echo "Run only on success"  
      if: success()  
    - run: echo "Run only on failure"  
      if: failure()
```



The screenshot shows the GitHub Actions logs for the 'expression_job'. The job starts with 'Set up job' and 'Run echo "Only run if triggered by main branch..."' (which succeeds). It then reaches a step that fails with 'Run echo "Fail depending on parameter"'. This failure causes the job to skip the 'Run always' step and move directly to the final success steps: 'Run echo "Run only on success"', 'Run echo "Run only on failure"', and finally 'Complete job'.

```
> ✓ Set up job  
> ✓ Run echo "Only run if triggered by main branch..."  
↳ ✘ Run echo "Fail depending on parameter"  
1 ► Run echo "Fail depending on parameter"  
7 /home/runner/work/_temp/83b1e782-0ae4-47a4-8e79-afb926c6bfa5.sh: line 2: return: can only `return` from a function or sourced script  
8 Fail depending on parameter  
9 Error: Process completed with exit code 1.  
  
► ✓ Run echo "Run always"  
► ✘ Run echo "Run only on success"  
► ✓ Run echo "Run only on failure"  
> ✓ Complete job
```

Contexts and expressions syntax

| Function | Description |
|--------------------------|---|
| <code>success()</code> | Returns true if none of the previous steps have failed or been cancelled. |
| <code>always()</code> | Returns true even if a previous step was cancelled and causes the step to always get executed anyway. |
| <code>cancelled()</code> | Returns only true if the workflow was canceled. |
| <code>failure()</code> | Returns true if a previous step of the job had failed. |

| Operator | Description |
|---------------------------|-------------------------------------|
| <code>()</code> | Logical group |
| <code>!</code> | Not |
| <code>< , <=</code> | Less than, less than or equal |
| <code>> , >=</code> | Greater than, greater than or equal |
| <code>==</code> | Equal |
| <code>!=</code> | Not equal |
| <code>&&</code> | And |
| <code> </code> | Or |

| Function | Description |
|--|--|
| <code>Contains (search, item)</code> | Returns true if search contains item. |
| <code>startsWith (search, item)</code> | Returns true if search start with item. |
| <code>endsWith (search, item)</code> | Returns true if search ends with item |
| <code>format(' {0} ', item)</code> | Replaces placeholders in a string. |
| <code>join(array, separator)</code> | All values in array are concatenated into a string. |
| <code>toJSON (value)</code> | Returns a pretty-print JSON representation of value. |
| <code>fromJSON (value)</code> | Returns a JSON object or JSON data type for value. |

Demo

Workflow commands

► Interact with the workflow from within your steps

► Write command to output (normally using echo)

► Examples:

- Set-output
- Error

```
steps:  
  - name: Set time  
    run: echo '::set-output name=my_time::$(date)'  
    id: set-time  
  
  - name: Output time  
    run: echo "It was ${{ steps.set-time.outputs.my_time }} in the previous step."  
  
  - name: Writing to the workflow log  
    run: |  
      echo "Writing to the workflow log"  
      echo "::group::Writing to the log"  
      echo "::notice file=starter.yml,line=169,col=19,endColumn=22::Writing a notice"  
      echo "::warning file=starter.yml,line=171,col=19,endColumn=22::Writing a warning"  
      echo "::endgroup::"
```

Workflow Commands
succeeded now in 1s

- > ✓ Set up job
- ✓ Set time
 - 1 ► Run echo '::set-output name=my_time::\$(date)'
- ✓ Output time
 - 1 ► Run echo "It was \$(date) in the previous step."
 - 6 It was Wed Apr 13 16:26:49 UTC 2022 in the previous step.
- ✓ Writing to the workflow log
 - 1 ► Run echo "Writing to the workflow log"
 - 10 Writing to the workflow log
 - 11 ▼Writing to the log
 - 12 Notice: Writing a notice
 - 13 Warning: Writing a warning
- > ✓ Complete job

Workflow commands

| Toolkit function | Equivalent workflow command | Toolkit function | Equivalent workflow command |
|----------------------------------|--|----------------------------------|--|
| <code>core.addPath</code> | Accessible using environment file GITHUB_PATH | <code>core.isDebugEnabled</code> | Accessible using environment variable RUNNER_DEBUG |
| <code>core.debug</code> | debug | <code>core.saveState</code> | save-state |
| <code>core.notice</code> | notice | <code>core.setCommandEcho</code> | echo |
| <code>core.error</code> | error | <code>core.setFailed</code> | Used as a shortcut for ::error and exit 1 |
| <code>core.endGroup</code> | endgroup | <code>core.setOutput</code> | set-output |
| <code>core.exportVariable</code> | Accessible using environment file GITHUB_ENV | <code>core.setSecret</code> | add-mask |
| <code>core.getInput</code> | Accessible using environment variable INPUT_{NAME} | <code>core.startGroup</code> | group |
| <code>core.getState</code> | Accessible using environment variable STATE_{NAME} | <code>core.warning</code> | warning |

Demo

Secrets

Secrets

- ▶ Defined on org, repo, or environment level

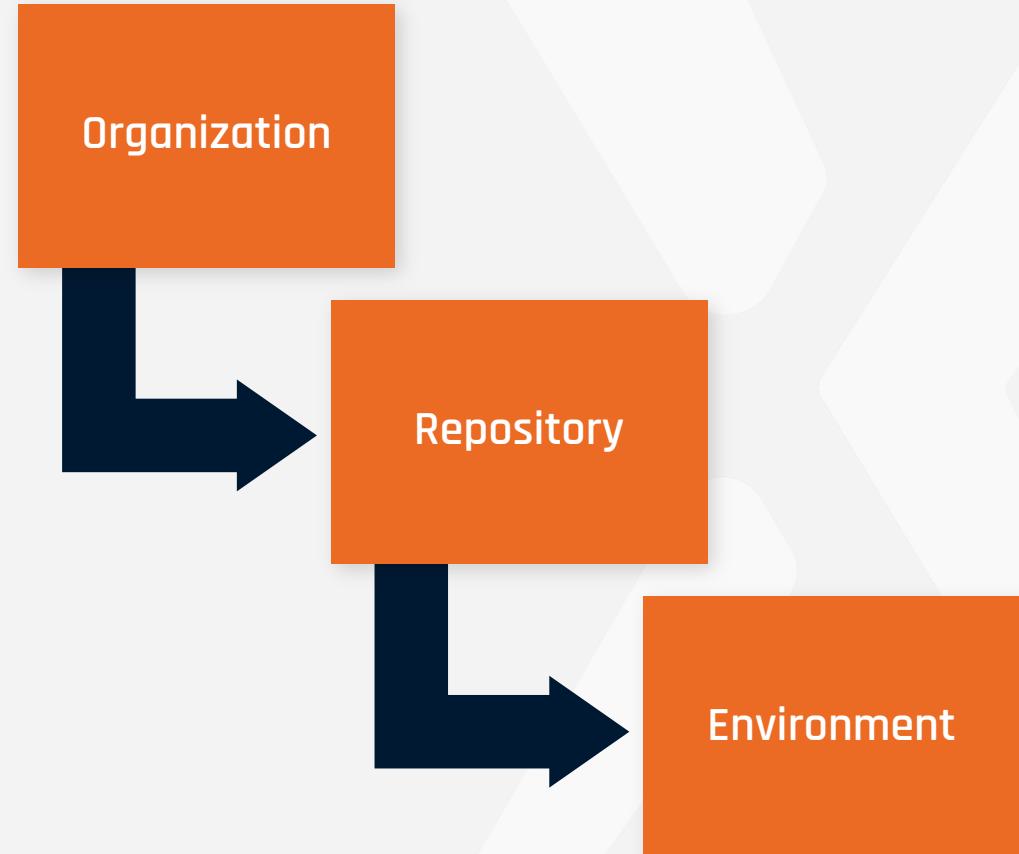
- ▶ Secret context

- › `${{ secrets.MY_SECRET }}`
- › Set as input (`with:`) or environment (`env:`) for actions

- ▶ Set in UI or CLI

- › `$ gh secret set MY_SECRET --body P4ssw0rd`
- › `$ gh secret set MY_SECRET --env Production`
- › `$ gh secret set MY_SECRET --org my-org`

- ▶ Masked in log



The GITHUB_TOKEN

▷ `${{ secrets.GITHUB_TOKEN }} or ${{ github.token }}`

▷ Authenticate to GitHub to perform automation

▷ Default permission read/write for all scopes (current default) or read repo

```
permissions:  
  contents: read  
  pull-requests: write
```

```
permissions: read-all
```

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

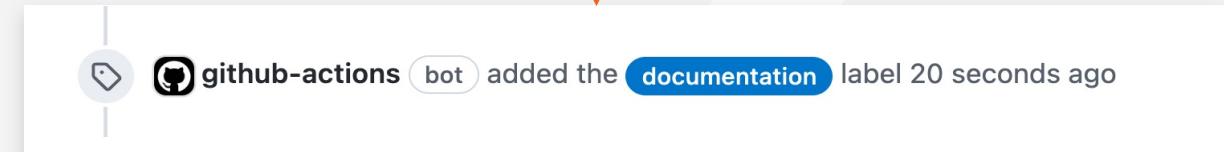
The GITHUB_TOKEN



Perform actions as github-actions:

```
permissions:  
  contents: read  
  issues: write
```

```
label_issues:  
  runs-on: ubuntu-latest  
  if: github.event_name == 'issues'  
  
steps:  
  - uses: andymckay/labeler@e6c4322d0397f3240f0e7e30a33b5c5df2d39e90  
    with:  
      add-labels: documentation  
      repo-token: ${{ secrets.GITHUB_TOKEN }}
```



Demo

Actions

GitHub Actions

► Actions are reusable

► 3 kind of Actions

- Container
- JavaScript / Typescript
- Composite Actions

The screenshot shows the GitHub Actions marketplace interface. At the top, there's a navigation bar with tabs: Overview (which is active), Repositories (47), Projects, Packages, People (19), and a Verified badge. Below the navigation is a section titled "Pinned" containing six action cards:

- starter-workflows** (Public) - Accelerating new GitHub Actions workflows. Created by TypeScript, 5.9k stars, 4.9k forks.
- toolkit** (Public) - The GitHub ToolKit for developing GitHub Actions. Created by TypeScript, 3.2k stars, 1.1k forks.
- setup-node** (Public) - Set up your GitHub Actions workflow with a specific version of node.js. Created by TypeScript, 2k stars, 726 forks.
- javascript-action** (Public template) - Create a JavaScript Action with tests, linting, workflow, publishing, and versioning. Created by JavaScript, 585 stars, 249 forks.
- typescript-action** (Public template) - Create a TypeScript Action with tests, linting, workflow, publishing, and versioning. Created by TypeScript, 1k stars, 275 forks.
- labeler** (Public) - An action for automatically labelling pull requests. Created by TypeScript, 899 stars, 308 forks.

Container Actions

Dockerfile or existing image

inputs

The screenshot shows a GitHub repository named "actions/container-action" which is a public template. The repository structure includes an "action.yml" file, a "Dockerfile", and an "entrypoint.sh" file.

The "action.yml" file contains the following configuration:

```
name: 'Wulfland Action'
description: 'Get started with Container Actions'
author: '@wulfland'
branding:
  icon: 'award'
  color: 'green'
inputs:
  myInput:
    description: 'Input to use'
    default: 'world'
runs:
  using: 'docker'
  image: 'Dockerfile'
  args:
    - ${{ inputs.myInput }}
```

The "Dockerfile" file contains the following Dockerfile code:

```
FROM alpine:3.10
COPY LICENSE README.md /
COPY entrypoint.sh /entrypoint.sh
ENTRYPOINT ["/entrypoint.sh"]
```

The "entrypoint.sh" file contains the following shell script:

```
#!/bin/sh -l
echo "hello $1"
```

Annotations highlight specific parts of the code:

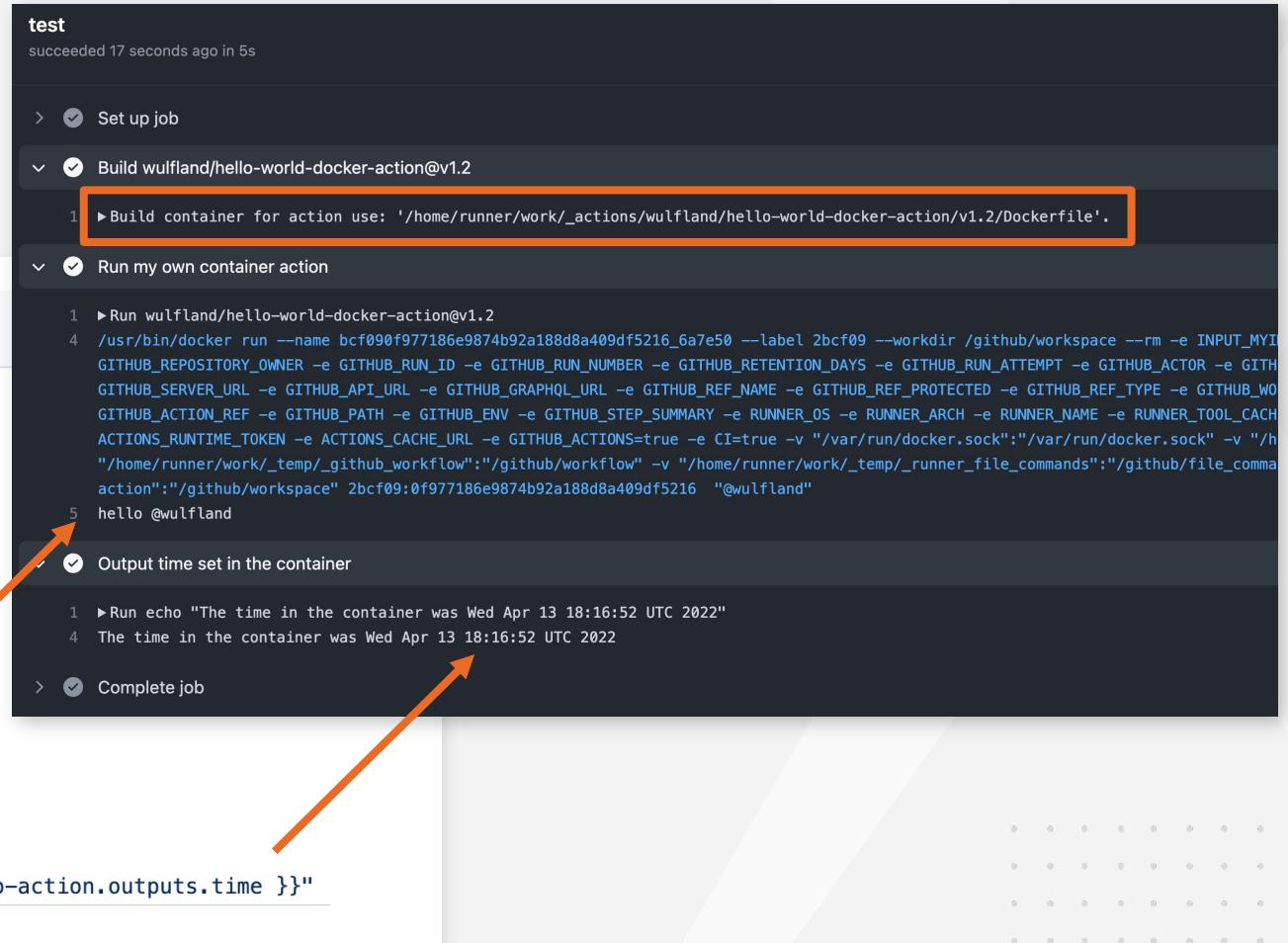
- A red box surrounds the "action.yml" file in the repository tree.
- A red box surrounds the "action.yml" file in the code editor.
- A red box surrounds the "myInput" section in the "action.yml" file.
- A red box surrounds the "ENTRYPOINT" line in the "Dockerfile".
- A red arrow points from the "myInput" section in "action.yml" to the "ENTRYPOINT" line in "Dockerfile".

Container Actions

- ▶ Dockerfile or existing image
- ▶ inputs

```
15 lines (13 sloc) | 398 Bytes

1 name: Test Action
2
3 on: [workflow_dispatch]
4
5 jobs:
6   test:
7     runs-on: ubuntu-latest
8     steps:
9       - name: Run my own container action
10         id: hello-action
11         uses: wulfland/hello-world-docker-action@v1.2
12         with:
13           myInput: '@wulfland'
14       - name: Output time set in the container
15         run: echo "The time in the container was ${{ steps.hello-action.outputs.time }}"
```



Demo

JavaScript Actions

The screenshot shows a GitHub repository interface with three main sections: Explorer, Code Editor, and GitHub Actions.

Explorer: Shows the directory structure of the repository. It includes files like `action.yml`, `index.js`, `main.ts`, `wait.ts`, and configuration files (`.eslintrc.json`, `.gitignore`, `.gitattributes`, etc.). The `dist` folder contains `index.js` and `index.js.map`.

Code Editor: Displays the `action.yml` file, which defines a GitHub Action named "My TS Action". It specifies inputs for milliseconds (required) and outputs for time. The `main` field points to `dist/index.js`.

Github Actions: Shows the `main.ts` file. It imports `@actions/core` and `./wait`. The `run` function waits for the specified milliseconds and then outputs the current date.

Annotations highlight several parts of the code:

- `index.js` in the `dist` folder.
- `main.ts` in the `src` folder.
- `action.yml` in the root.
- The `using: 'node16'` and `main: 'dist/index.js'` fields in `action.yml`.
- The `import` statements and the `run` function body in `main.ts`.

Demo

Compose Actions

► Just a `action.yml` file

► Inputs

► Outputs

► Runs

28 lines (25 sloc) | 689 Bytes

```
1 name: 'Hello World'
2 description: 'Greet someone'
3 inputs:
4   who-to-greet:
5     description: 'Who to greet'
6     required: true
7     default: 'World'
8 outputs:
9   random-number:
10    description: "Random number"
11    value: ${{ steps.random-number-generator.outputs.random-id }}
12 runs:
13   using: "composite"
14   steps:
15     - run: echo Hello ${{ inputs.who-to-greet }}.
16       shell: bash
17     - id: random-number-generator
18       run: echo "$ :set-output name=random-id:$RANDOM"
19       shell: bash
20
21     - run: echo "${{ github.action_path }}" >> $GITHUB_PATH
22       shell: bash
23
24     - run: echo "Goodbye $YOU"
25       shell: bash
26       env:
27         YOU: ${{ inputs.who-to-greet }}
```

Share your action in the marketplace

► Really easy to share

► Draft a release

► Unique name

► Check for README, Icon, Color

```
name: 'Wulfland Action'  
description: 'Get started with Container actions'  
author: '@wulfland'  
branding:  
  icon: 'award'  
  color: 'green'
```

The screenshot shows the 'Releases' tab selected in the GitHub Marketplace interface. A checkbox labeled 'Publish this Action to the GitHub Marketplace' is checked. Below it, the 'action.yml' file is displayed with the following configuration:

```
name: Wulfland Action  
description: Get started with Container actions  
icon: award  
color: green
```

A green success message indicates that everything looks good. Below this, the 'README' section shows a green message stating that a README exists.

At the bottom, there are dropdown menus for 'Primary Category' and 'Another Category — optional', both currently set to 'Choose an option'.

Demo

Running your workflows

GitHub hosted runners

- › **Linux**
- › **Windows**



Hardware:

- › [Standard DS2 v2](#) virtual machines in [Microsoft Azure](#)
- › 2-core CPU
- › 7 GB of RAM
- › 14 GB of SSD disk space

Passwordless sudo / UAC disabled

- › **MacOS**



Hardware:

- › 3-core CPU
- › 14 GB of RAM
- › 14 GB of SSD disk space

Passwordless sudo

Virtual Environments

| Environment | YAML Label | Included Software |
|---------------------|---------------------------------------|-------------------------------------|
| Ubuntu 20.04 | ubuntu-latest or ubuntu-20.04 | <u>ubuntu-20.04</u> |
| Ubuntu 18.04 | ubuntu-18.04 | <u>ubuntu-18.04</u> |
| macOS 11 | macos-latest or macos-11 | <u>macOS-11</u> |
| macOS 10.15 | macos-10.15 | <u>macOS-10.15</u> |
| Windows Server 2022 | windows-latest or windows-2022 | <u>windows-2022</u> |
| Windows Server 2019 | windows-2019 | <u>windows-2019</u> |
| Windows Server 2016 | windows-2016 | <u>windows-2016</u> |

<https://github.com/actions/virtual-environments>

GitHub hosted runners - pricing



Build minutes:

- › On Linux: \$0.008
- › On macOS: \$0.08
- › On Windows: \$0.016

| GitHub edition | Storage | Minutes | Max concurrent jobs |
|-------------------------------|---------|---------|---------------------|
| GitHub Free | 500 MB | 2,000 | 20 (5 for macOS) |
| GitHub Pro | 1 GB | 3,000 | 40 (5 for macOS) |
| GitHub Free for organizations | 500 MB | 2,000 | 20 (5 for macOS) |
| GitHub Team | 2 GB | 3,000 | 60 (5 for macOS) |
| GitHub Enterprise Cloud | 50 GB | 50,000 | 180 (50 for macOS) |

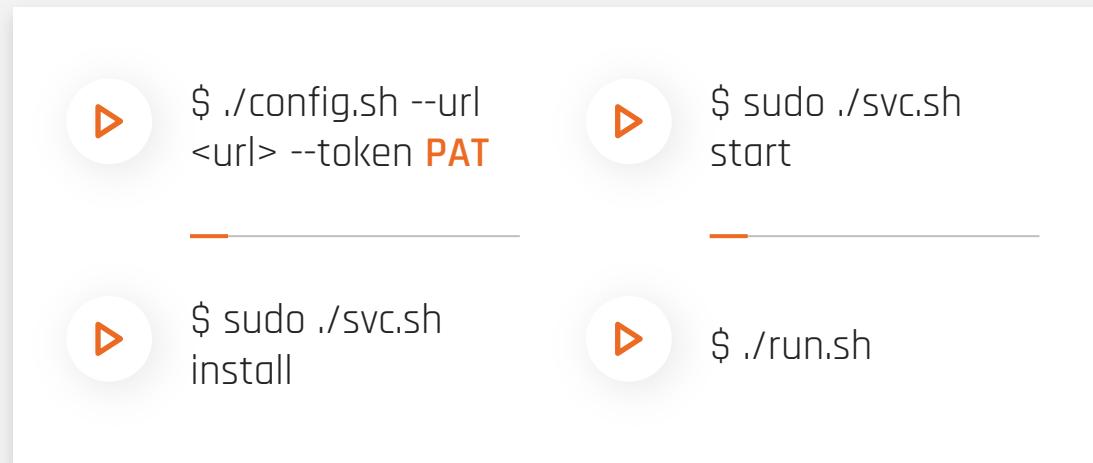
```
Set up job

1 Current runner version: '2.289.2'
2 ▼Operating System
3   Ubuntu
4     20.04.4
5   LTS
6   ▼Virtual Environment
7     Environment: ubuntu-20.04
8     Version: 20220405.4
9     Included Software: https://github.com/actions/virtual-environments/blob/ubuntu20/20220405.4/images/linux/Ubuntu2004-Readme.md
10    Image Release: https://github.com/actions/virtual-environments/releases/tag/ubuntu20%2F20220405.4
11  ▶Virtual Environment Provisioner
12  ▼GITHUB_TOKEN Permissions
13    Actions: write
14    Checks: write
15    Contents: write
16    Deployments: write
17    Discussions: write
18    Issues: write
19    Metadata: read
20    Packages: write
21    Pages: write
22    PullRequests: write
23    RepositoryProjects: write
24    SecurityEvents: write
25    Statuses: write
26    Secret source: Actions
27    Prepare workflow directory
28    Prepare all required actions
29    Getting action download info
30    Download action repository 'actions/checkout@{a12a3943b4bdde767164f792f33f40b04645d846}' (SHA:a12a3943b4bdde767164f792f33f40b04645d846)
31    Pull alpine:3.8
32    Pull ghcr.io/wulfland/container-demo:latest
33    Run echo "💡 The job was triggered by a workflow_dispatch event."
```

Demo

Self-hosted runners

- › **Free**
- › **Any platform** (x64: Linux, macOS, Window, ARM64 and ARM32 on Linux)
- › **HTTPS long polling** port 443 - 50 seconds
- › Can be used to **deploy to local resources**
- › Can be added at **Enterprise, Organization, and Repository level**



Runners / Create self-hosted runner

Adding a self-hosted runner requires that you download, configure, and execute the GitHub Actions Runner. By downloading and configuring the GitHub Actions Runner, you agree to the [GitHub Terms of Service](#) or [GitHub Corporate Terms of Service](#), as applicable.

Runner image

macOS Linux Windows

Architecture

x64

Download

```
# Create a folder
$ mkdir actions-runner && cd actions-runner

# Download the latest runner package
$ curl -o actions-runner-linux-x64-2.289.2.tar.gz -L
https://github.com/actions/runner/releases/download/v2.289.2/actions-runner-linux-x64-2.289.2.tar.gz

# Optional: Validate the hash
$ echo "7ba89bb75397896a76e98197633c087a9499d4c1db7603f21910e135b0d0a238" actions-runner-linux-x64-2.289.2.tar.gz | shasum -a 256 -c

# Extract the installer
$ tar xzf ./actions-runner-linux-x64-2.289.2.tar.gz
```

Self-hosted runners



Access: Runner Groups



A runner can only be in 1 group



Apply labels

- › \$./config.sh --labels self-hosted,x64,linux
- › runs-on: [self-hosted, linux, X64]

Runner Groups / New Runner Group

Group name

Repository access

Selected repositories ▾ 1 selected repository ⓘ

Allow public repositories

Runners can be used by public repositories. Allowing self-hosted runners on public repositories and allowing workflows on public forks introduces a significant security risk. [Learn more](#)

Workflow access

Control how these runners are used by restricting them to specific workflows. [Learn more](#)

Selected workflows ▾ 0 selected workflows ⓘ

Create group

Self-hosted runners



Access: Runner Groups



A runner can only be in 1 group



Apply labels

- › \$./config.sh --labels self-hosted,x64,linux,**matlab**
- › runs-on: [self-hosted, linux, X64, **matlab**]

Runner Groups / New Runner Group

Group name

Repository access

Selected repositories ▾ 1 selected repository ⓘ

Allow public repositories
Runners can be used by public repositories. Allowing self-hosted runners on public repositories and allowing workflows on public forks introduces a significant security risk. [Learn more](#)

Workflow access

Control how these runners are used by restricting them to specific workflows. [Learn more](#)

Selected workflows ▾ 0 selected workflows ⓘ

Create group

Self-hosted runners - gotchas

- ▶ Runners are not ephemeral per default - you have to clean up after a build yourself
 - › \$./config.sh --ephemeral

- ▶ Use web hooks to auto scale (<https://github.com/jonico/awesome-runners>)

- ▶ Do not allow public repositories!

- ▶ Limit Actions and use SHA or fork

- ▶ Create a company marketplace (<https://github.com/rajbos/actions-marketplace>)

The screenshot shows the 'General actions permissions' section of a GitHub repository settings page. It includes a 'Policies' section with a dropdown set to 'All repositories', three policy options, and a detailed view of the 'Allow accelerate-devops, and select non-accelerate-devops, actions and reusable workflows' option. This view shows two checked checkboxes: 'Allow actions created by GitHub' and 'Allow actions by Marketplace verified creators'. Below this is a 'Allow specified actions and reusable workflows' input field containing 'microsoft/*' and 'my-org/*', with a note about wildcards, tags, and SHAs being allowed.

Environments

Environments



Environments:

- › Reviewers / Approvers
- › Wait timer (until 30 days)
- › Branches (→ branch protection!)
- › Deployment branches
- › Secrets

```
Test:  
  runs-on: ubuntu-latest  
  environment: Test  
  needs: Build  
  steps:  
    - name: Test app  
      run: echo "Testing..."
```

```
Production:  
  runs-on: ubuntu-latest  
  environment:  
    name: prod  
    url: https://writeabout.net  
  needs: Staging  
  steps:  
    - name: Deploy app  
      run: echo "Deploying..."
```

[Environments / Configure Test](#)

Environment protection rules
Can be used to configure manual approvals and timeouts.

Required reviewers
Specify people or teams that may approve workflow runs when they access this environment.
[Add up to 5 more reviewers](#)
Search for people or teams...
wulfland

Wait timer
Set an amount of time to wait before allowing deployments to proceed.

[Save protection rules](#)

Deployment branches
Can be used to limit what branches can deploy to this environment using branch name patterns. [All branches ▾](#)

Environment secrets
Secrets are encrypted environment variables. They are accessible only by GitHub Actions in the context of this environment.
[+ Add Secret](#)

Environments



Approvals



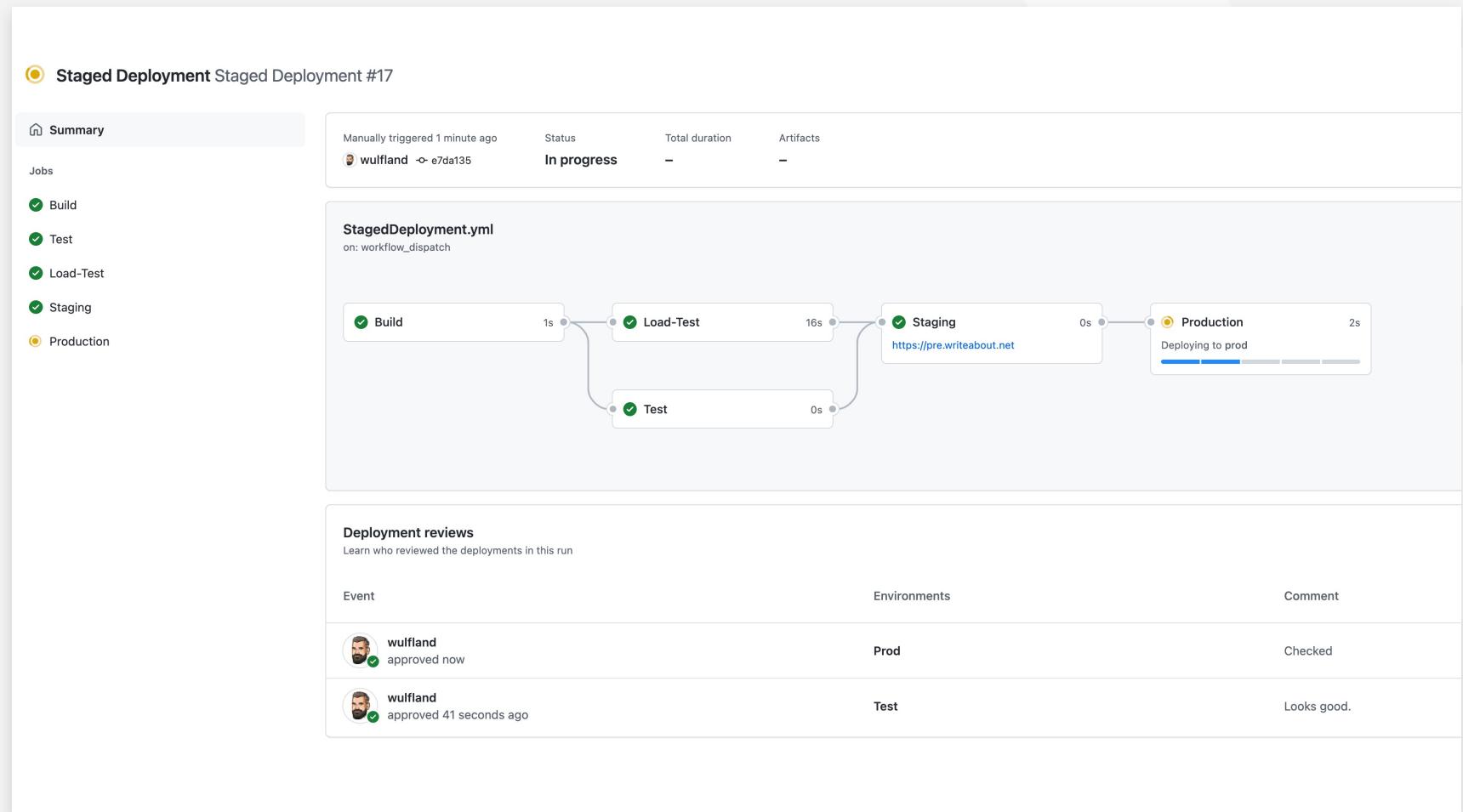
Secrets after approval



Set URL from output of other job/step



Progress



Demo

Workflow templates

Workflow templates

► Available in Actions / New workflow

► Get copied one time

► Starter workflows

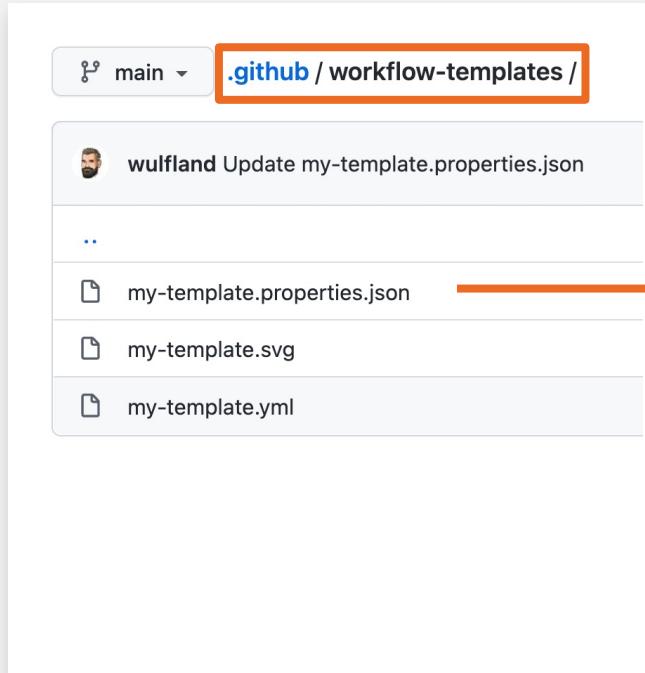
The screenshot shows the GitHub Actions Workflow Templates interface. At the top, there's a card for a user-created template named "My Workflow Template" by Accelerate DevOps. Below this, a section titled "Deployment" lists eight pre-built templates:

- Deploy Node.js to Azure Web App** By Microsoft Azure: Build a Node.js project and deploy it to an Azure Web App.
- Deploy to Amazon ECS** By Amazon Web Services: Deploy a container to an Amazon ECS service powered by AWS Fargate or Amazon EC2.
- Build and Deploy to GKE** By Google Cloud: Build a docker container, publish it to Google Container Registry, and deploy to GKE.
- Terraform** By HashiCorp: Set up Terraform CLI in your GitHub Actions workflow.
- Deploy to Alibaba Cloud ACK** By Alibaba Cloud: Deploy a container to Alibaba Cloud Container Service for Kubernetes (ACK).
- Deploy to IBM Cloud Kubernetes Service** By IBM: Build a docker container, publish it to IBM Cloud Container Registry, and deploy to IBM Cloud Kubernetes Service.
- Tencent Kubernetes Engine** By Tencent Cloud: This workflow will build a docker container, publish and deploy it to Tencent Kubernetes Engine (TKE).
- OpenShift** By Red Hat: Build a Docker-based project and deploy it to OpenShift.

Each template card includes "Configure" and "Deployment" buttons. A "View all" link is located at the top right of the deployment section.

Workflow templates

► <org>/.github/workflow-templates



```
13 lines (13 sloc) | 269 Bytes
```

```
1 {
2   "name": "My Workflow Template",
3   "description": "Description of template workflow",
4   "iconName": "my-template",
5   "categories": [
6     "javascript"
7   ],
8   "filePatterns": [
9     "package.json$",
10    "^Dockerfile",
11    ".*\.\.md$"
12  ]
13 }
```

```
15 lines (11 sloc) | 232 Bytes
```

```
1 name: My templated workflow
2
3 on:
4   push:
5     branches: [ $default-branch ]
6
7 jobs:
8   build:
9     runs-on: ubuntu-latest
10
11 steps:
12   - uses: actions/checkout@v2
13
14   - name: Run a one-line script
15     run: echo Hello World!
```

Reusable workflows

```
1 name: Reusable workflow
2
3 on:
4   workflow_call:
5     inputs:
6       who-to-greet:
7         description: 'The person to greet'
8         type: string
9         required: true
10        default: World
11
12      outputs:
13        current-time:
14          description: 'The time when greeting.'
15          value: ${{ jobs.reusable-job.outputs.current-time }}
16
17 jobs:
18   reusable-job:
19     runs-on: ubuntu-latest
20     outputs:
21       current-time: ${{ steps.time.outputs.current-time }}
22     steps:
23       - name: Greet someone
24         run: echo "Hello ${{ inputs.who-to-greet }}"
25       - name: Set time
26         id: time
27         run: echo "::set-output name=current-time::$(date)"
28
```

```
1 name: Reuse other workflow
2
3 on: [workflow_dispatch]
4
5 jobs:
6   call-workflow:
7     uses: ./github/workflows/reusable.yml
8     with:
9       who-to-greet: '@wulfland'
10
11 use-output:
12   runs-on: ubuntu-latest
13   needs: [call-workflow]
14   steps:
15     - run: echo "Time was ${{ needs.call-workflow.outputs.current-time }}"
16
```

Ref if not
in same
repo!

Concurrency

Workflow or job

Optional: cancel in-progress jobs

Use cases:

- › Wait job/workflow until deployment completed
- › Cancel deployment and deploy newer version instead

```
1  name: Concurrency
2
3  on: [workflow_dispatch]
4
5  jobs:
6    job1:
7      concurrency: group1
8      runs-on: ubuntu-latest
9      steps:
10     - run: sleep 60
11     - run: echo "Hello World! $(date)"
12
13  job2:
14    concurrency: group1
15    runs-on: ubuntu-latest
16    steps:
17     - run: sleep 60
18     - run: echo "Hello World! $(date)"
19
20  job3:
21    concurrency:
22      group: group2
23      cancel-in-progress: true
24
25    runs-on: ubuntu-latest
26    steps:
27     - run: sleep 60
28     - run: echo "Hello World! $(date)"
29
30  job4:
31    concurrency:
32      group: group2
33      cancel-in-progress: true
34
35    runs-on: ubuntu-latest
36    steps:
37     - run: sleep 60
38     - run: echo "Hello World! $(date)"
```

Caching



Use cache action



Use cache key for grouping



Use setup actions for package managers:

- › npm
- › gradle
- › pip
- › ...



Limits:

- › 7 days
- › 10 GB

```
runs-on: ubuntu-latest
```

```
steps:
```

- uses: actions/checkout@v3
- name: Cache Primes
id: cache-primes
uses: actions/cache@v3
with:
path: primes
key: \${{ runner.os }}-primes
- name: Generate Prime Numbers
if: steps.cache-primes.outputs.cache-hit != 'true'
run: |
sleep 60
echo "1 2 3..." > primes
- name: Use Prime Numbers
run: cat primes

Let's connect



@mike_kaufmann



@wulfland



<https://writeabout.net>

Let's connect!



**Michael
Kaufmann**

Founder/MD Xpirit Germany
Microsoft Regional Director, MVP
[Linkedin.com/in/mikaufmann](https://www.linkedin.com/in/mikaufmann)
mkaufmann@xpirit.com