



Browser

GitHub account (free tier is fine)

Basic knowledge of GitHub

Labs document

Direct link: <https://github.com/skilldocs/gha3/blob/main/gha3-labs.pdf>

Download: <https://raw.githubusercontent.com/skilldocs/gha3/main/gha3-labs.pdf>





GitHub Actions in 3 Hours

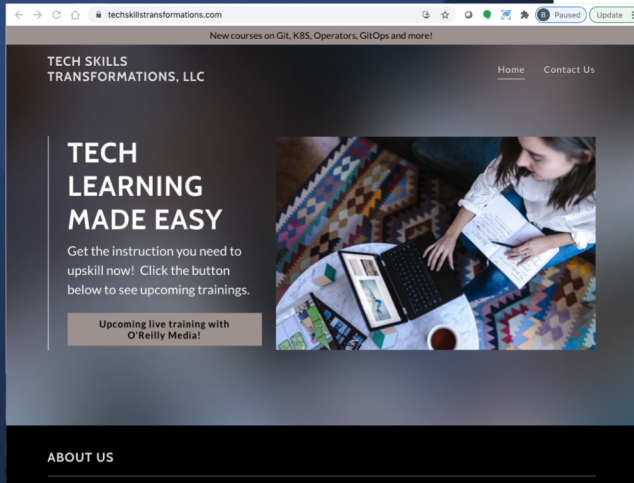
How to easily automate and integrate with GitHub

Tech Skills Transformations & Brent Laster



About me

- Founder, Tech Skills Transformations LLC
- R&D DevOps Director
- Global trainer – training (Git, Jenkins, Gradle, CI/CD, pipelines, Kubernetes, Helm, ArgoCD, operators)
- Author -
 - Professional Git
 - Jenkins 2 – Up and Running book
 - Learning GitHub Actions
 - Various reports on O'Reilly Learning
- <https://www.linkedin.com/in/brentlaster>
- @BrentCLaster
- GitHub: brentlaster





Professional Git Book

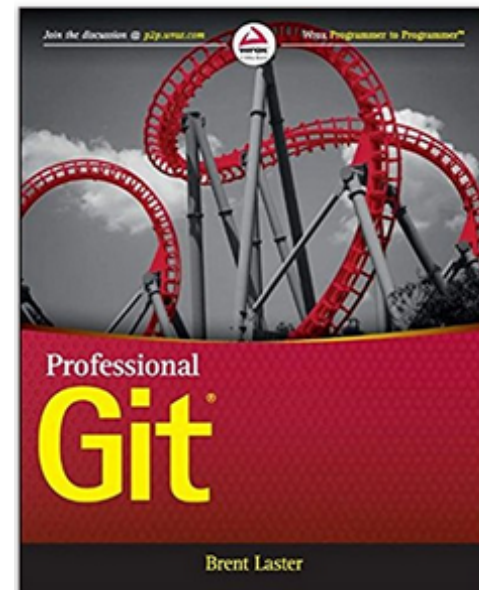
- Extensive Git reference, explanations,
- and examples
- First part for non-technical
- Beginner and advanced reference
- Hands-on labs

Professional Git 1st Edition

by [Brent Laster](#) (Author)

★★★★★ 7 customer reviews

[Look inside](#) ↓



© 2023 Brent C. Laster &

Tech Skills Transformations LLC



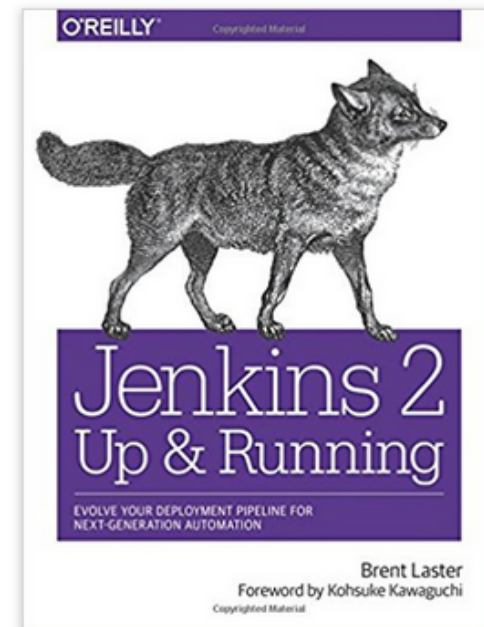
Jenkins 2 Book

- Jenkins 2 – Up and Running
- “It’s an ideal book for those who are new to CI/CD, as well as those who have been using Jenkins for many years. This book will help you discover and rediscover Jenkins.” *By Kohsuke Kawaguchi, Creator of Jenkins*

★★★★☆ 5 customer reviews

#1 New Release in Java Programming

[Look inside](#) ↴



© 2023 Brent C. Laster &

Tech Skills Transformations LLC




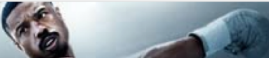
GitHub Actions book

7

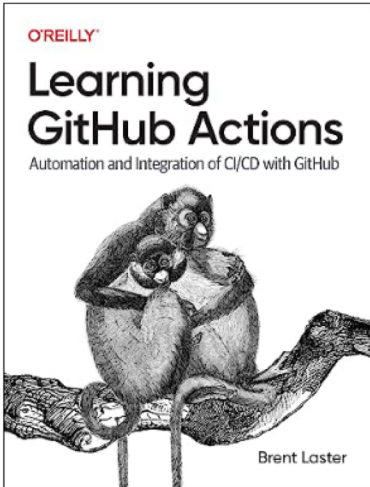
[All](#) [Get the app](#) [Prime Day](#) [Back to School](#) [Add People](#) [Buy Again](#) [Gift Cards](#) [Recommendations](#) [IT Supplies](#) [Business Savings](#) [Amazon Basics](#) [EN](#) [Hello, Account](#)

[Guide buyers in your org](#) [Group: Tech Skills Transf](#)

[Books](#) [Advanced Search](#) [New Releases](#) [Best Sellers & More](#) [Children's Books](#) [Textbooks](#) [Textbook Rentals](#) [Best Books of the Month](#) [Your Company Bookshelf](#)


  **CREED III** [Watch now](#)

[Books](#) > [Computers & Technology](#) > [Programming](#)



Roll over image to zoom in

Follow the Author



[Brent Laster](#) [Follow](#)

Learning GitHub Actions: Automation and Integration of CI/CD with GitHub 1st Edition

by [Brent Laster](#) (Author)

[See all formats and editions](#)

Paperback
\$65.99 ✓prime
1 New from \$65.99

Pre-order Price Guarantee. [Terms](#)

Automate your software development processes with GitHub Actions, the continuous integration and continuous delivery platform that integrates seamlessly with GitHub. With this practical book, open source author, trainer, and DevOps director Brent Laster explains everything you need to know about using and getting value from GitHub Actions. You'll learn what actions and workflows are and how they can be used, created, and incorporated into your processes to simplify, standardize, and automate your work in GitHub.

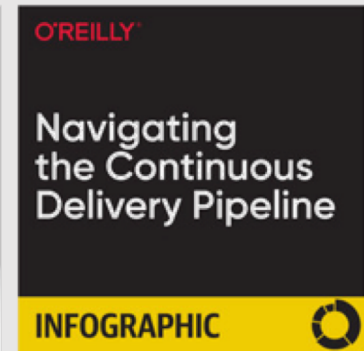
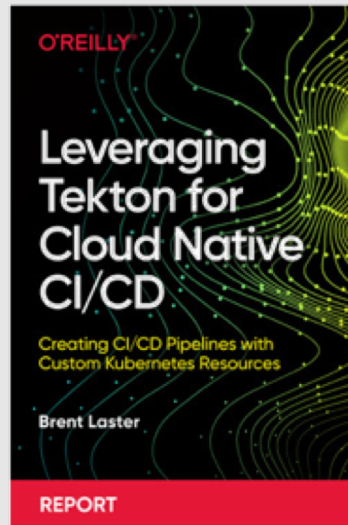
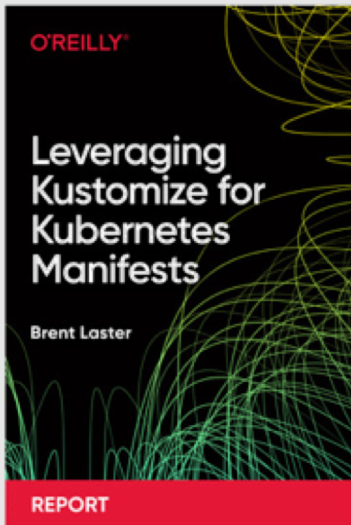
This book explains the platform, components, use cases, implementation, and integration points of actions, so you can leverage them to provide the functionality and features needed in today's complex pipelines and software development processes. You'll learn how to design and implement automated workflows that respond to common events like pushes, pull requests, and review updates. You'll understand how to use the components of the GitHub Actions platform to gain maximum automation and benefit.

With this book, you will:

- Learn what GitHub Actions are, the various use cases for them, and how to incorporate them into your processes
- Understand GitHub Actions' structure, syntax, and semantics
- Automate processes and implement functionality
- Create your own custom actions with Docker, JavaScript, or shell approaches
- Troubleshoot and debug workflows that use actions
- Combine actions with GitHub APIs and other integration options
- Identify ways to securely implement workflows with GitHub Actions
- Understand how GitHub Actions compares to other options

[^ Read less](#)







April 26 & 27, 2021

Git Fundamentals

Join Brent Laster to gain the knowledge and skills you need to leverage Git to greatly simplify and speed up managing all of the changes in your source code. Once you ...



June 3, 10, 17 & 24, 2021

Git in 4 Weeks

Join author, trainer, and DevOps director Brent Laster to learn how Git works—and discover all the things you can do with it. Over four sessions, Brent walks you through everything you ...

LIVE ONLINE TRAINING

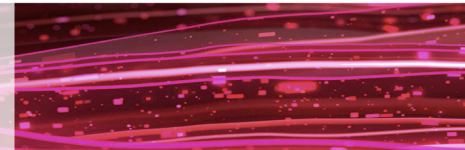
Git Troubleshooting

How to solve practically any problem that comes your way

Topic: **Software Development**



BRENT LASTER



LIVE ONLINE TRAINING

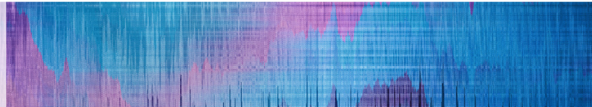
Next Level Git - Master your content

Use powerful tools in Git to simplify merges, rewrite history, and perform automatic updates

Topic: **Software Development**



BRENT LASTER



LIVE ONLINE TRAINING

Next Level Git - Master your workflow

Use Git to find problems, simplify working with multiple branches and repositories, and customize behavior with hooks

Topic: **Software Development**



BRENT LASTER



LIVE ONLINE TRAINING

Getting started with continuous delivery (CD)

Move beyond CI to build, manage, and deploy a working pipeline

Topic: **System Administration**



BRENT LASTER



LIVE ONLINE TRAINING

Building a deployment pipeline with Jenkins 2

Manage continuous integration and continuous delivery to release software

Topic: **System Administration**



BRENT LASTER



June 28, 2021

Troubleshooting Kubernetes

In this 3-hour course, global trainer, author, and DevOps director Brent Laster will show you how to respond to the most common problem situations you may encounter with Kubernetes. You'll learn ...

LIVE ONLINE TRAINING

Helm Fundamentals

Deploying, upgrading, and rolling back applications in Kubernetes

Topic: **System Administration**



BRENT LASTER



LIVE ONLINE TRAINING

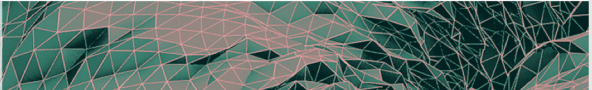
Containers A-Z

An overview of containers, Docker, Kubernetes, Istio, Helm, Kubernetes Operators, and GitOps

Topic: **System Administration**



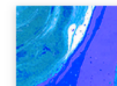
BRENT LASTER



May 24, 2021

Continuous Delivery in Kubernetes with ArgoCD

Join expert Brent Laster to explore GitOps and learn how to use Argo CD to implement GitOps in your Kubernetes deployments. APAC time friendly - You're a Kubernetes admin who wants ...



May 17, 2021

Building a Kubernetes Operator

Join expert Brent Laster to learn how the Operator pattern helps address these kinds of situations by allowing you to create custom controllers that extend the functionality of the Kubernetes API ...



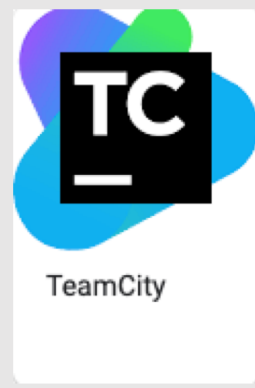
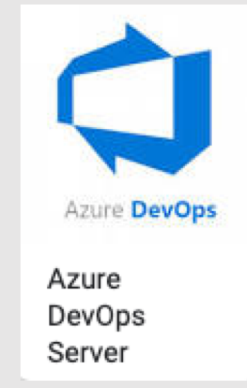
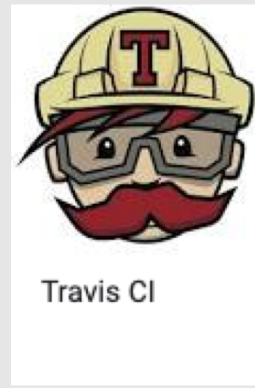
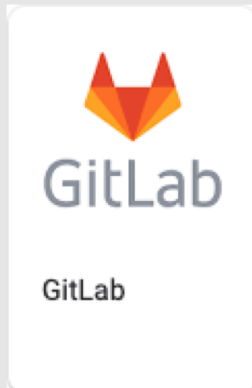


- What are GitHub Actions?
- How do they work?
- The GitHub interface to actions
- About Actions Runners and Virtual Environments
- Using public actions
- Workflow runs
- Custom actions
- Manually running workflows
- Monitoring and Troubleshooting
- Creating secrets for actions
- Chaining workflows together
- Using GitHub API calls with actions
- Reusable workflows





- Way to create custom, automated workflows in GitHub
- Executed based on repository operations
- Building blocks - can be combined & shared
- Used for usual SDLC tasks
- Can be used for CI/CD as alternatives to other apps





Other use cases (starter workflows)

https://github.com/brentlaster/greetings-actions/actions/new

Deployment

[View all](#)

Deploy a Java .jar app to an Azure Web App
By Microsoft Azure
Build a Java project and deploy it to an Azure Web App.
[Configure](#) [Deployment](#)

Deploy to Amazon ECS
By Amazon Web Services
Deploy a container to an Amazon ECS service powered by AWS Fargate or Amazon EC2.
[Configure](#) [Deployment](#)

Build and Deploy to GKE
By Google Cloud
Build a docker container, publish it to Google Container Registry, and deploy to GKE.
[Configure](#) [Deployment](#)

Terraform
By HashiCorp
Set up Terraform CLI in your GitHub Actions workflow.
[Configure](#) [Deployment](#)

Deploy to Alibaba Cloud ACK
By Alibaba Cloud
Deploy a container to Alibaba Cloud Container Service for Kubernetes (ACK).
[Configure](#) [Deployment](#)

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.
[Configure](#) [Deployment](#)

Tencent Kubernetes Engine
By Tencent Cloud
This workflow will build a docker container, publish and deploy it to Tencent Kubernetes Engine (TKE).
[Configure](#) [Deployment](#)

OpenShift
By Red Hat
Build a Docker-based project and deploy it to OpenShift.
[Configure](#) [Deployment](#)

Security

[View all](#)

Dependency Review
By GitHub Actions
Scans Pull Requests on each push for the introduction and/or resolution of vulnerable dependencies to the repository
[Configure](#) [Dependency review](#)

CodeQL Analysis
By GitHub
Security analysis from GitHub for C, C++, C#, Go, Java, JavaScript, TypeScript, Python, and Ruby developers.
[Configure](#) [Code scanning](#)

APIsec Scan
By APIsec
APIsec provides the industry's only automated and continuous API testing platform that uncovers security vulnerabilities and logic flaws in APIs.
[Configure](#) [Code scanning](#)

CxSAST
By Checkmarx
Scan your code with Checkmarx CxSAST and see your results in the GitHub security tab.
[Configure](#) [Code scanning](#)

Continuous integration

[View all](#)

Java with Maven
By GitHub Actions
Build and test a Java project with Apache Maven.
[Configure](#) [Java](#)

Scala
By GitHub Actions
Build and test a Scala project with SBT.
[Configure](#) [Scala](#)

Erlang
By GitHub Actions
Build and test an Erlang project with rebar.
[Configure](#) [Erlang](#)

.NET Desktop
By GitHub Actions
Build, test, sign and publish a desktop application built on .NET.
[Configure](#) [C#](#)

Automation

Labeler
By GitHub Actions
[Configure](#)

Manual workflow
By GitHub Actions
[Configure](#)

Greetings
By GitHub Actions
[Configure](#)

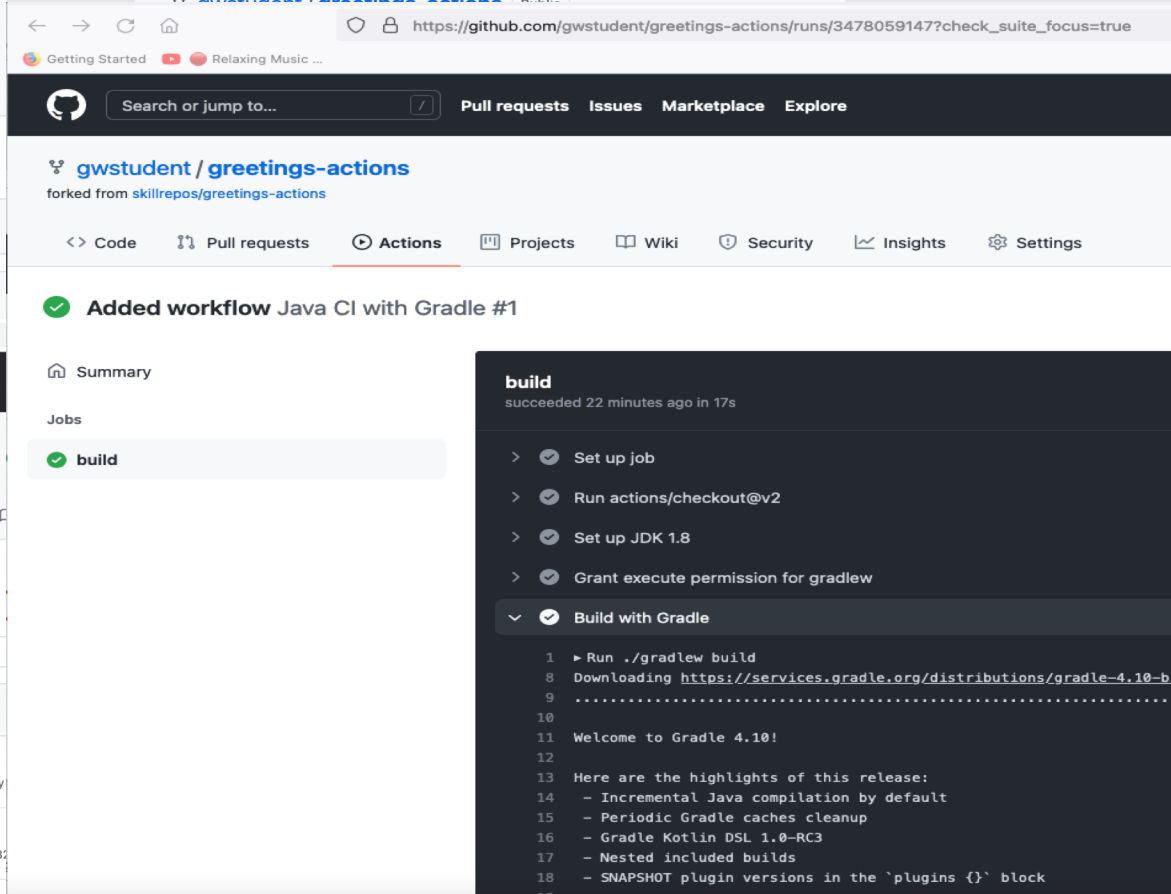
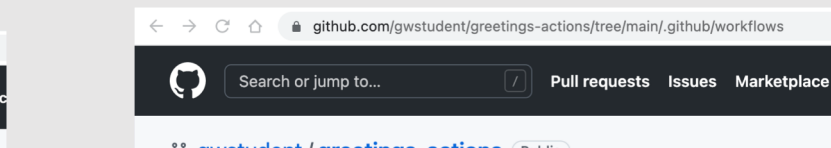
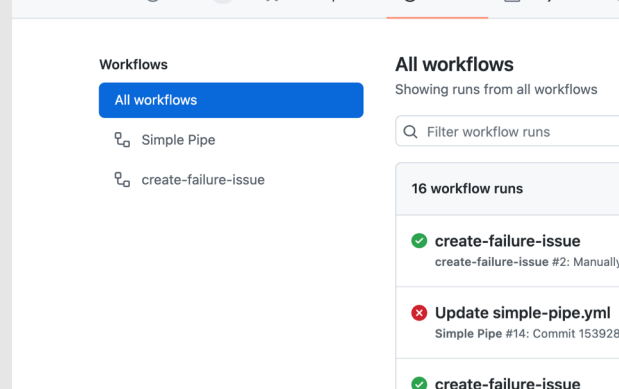
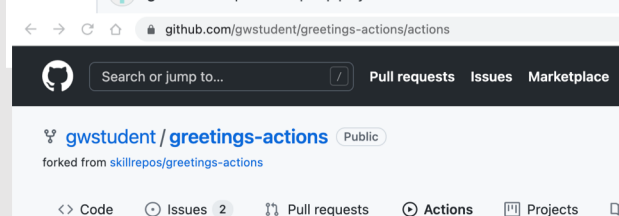
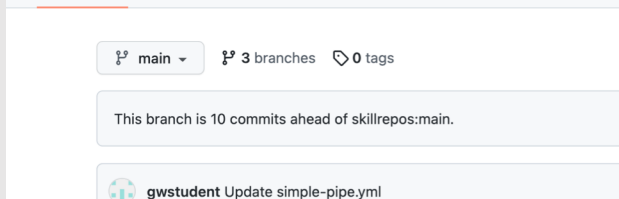
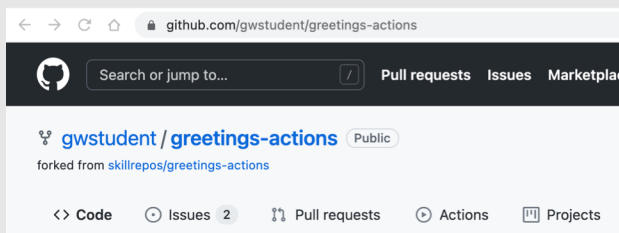
Stale
By GitHub Actions
[Configure](#)





What's interesting about it?

- Direct integration with GitHub





What else is interesting about it?

- Lots of events can trigger an action workflow - automate practically anything!

on: pull_request

on: repository_dispatch

on: fork

on: watch

on: public

on: page_build

on: project

on: delete

on: scheduled

on: check_run

on: pull_push

on: pull_request_review_comment

<https://docs.github.com/en/actions/learn-github-actions/events-that-trigger-workflows>





Types of triggers for events

- *single event* - **on:** `push`
- *list of events* - **on:** `[push, pull_request]`
- *event types with qualifiers, such as branches, tags, or files*

on:

push:

branches:

- `main`
- `'rel/v*'`

tags:

- `v1.*`
- `beta`

paths:

- `'**.*ts'`

- *scheduled events*

on:

scheduled

- **cron:** `'30 5,15 * * *'`

- *manual events* - **on:** `workflow-dispatch`
on: `repository-dispatch`
- *workflow reuse events* - **on:** `workflow-call`
- *activity types*

Webhook event payload	Activity types
issues	<ul style="list-style-type: none">- opened- edited- deleted- transferred- pinned- unpinned- closed- reopened- assigned- unassigned- labeled- unlabeled- locked- unlocked- milestoned- demilestoned

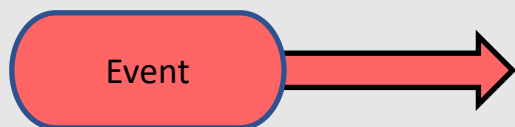




How actions work (events)

16

- Events occur
 - Example: pull request triggers build for validation
 - Example: push for commit
- Repository Dispatch Events
 - Endpoint that can be used to trigger webhook event
 - Can be used for activity that is not in GitHub
 - Can trigger a GitHub Actions workflow or GitHub App webhook
- ref
<https://docs.github.com/en/rest/reference/repos#create-a-repository-dispatch-event>



```
3
4  name: Simple Pipe
5
6  on:
7    push:
8      branches: [ main ]
9    pull_request:
10     branches: [ main ]
11 workflow_dispatch:
12   inputs:
13     myValues:
14       description: 'Input Values'
15
16
17 jobs:
18   build:
```

Scheduled events

schedule

Manual events

workflow_dispatch

repository_dispatch

Webhook events

check_run

check_suite

create

delete

deployment

deployment_status

fork

gollum

issue_comment

issues

label

milestone

page_build

project

project_card

project_column

public

pull_request

pull_request_review

pull_request_review_comment

pull_request_target

push

registry_package

release

status

watch

workflow_run

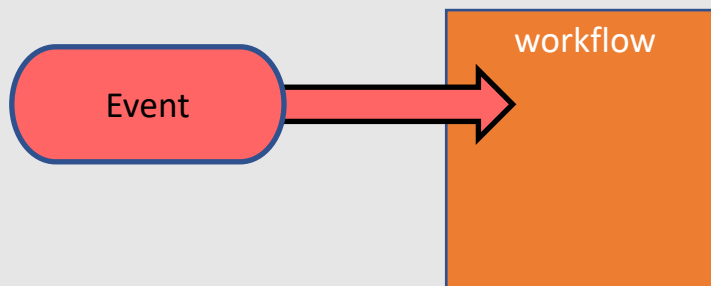




How actions work (workflows)

17

- Events trigger workflows
 - Procedure to run automatically
 - Added to your repository
 - Composed of one+ jobs
 - Can be triggered/scheduled by event
 - Useful for doing CI/CD actions such as building, testing, packaging, etc.



The screenshot shows the GitHub Actions interface. The top navigation bar includes links for Code, Pull requests, Actions (selected), Projects, Wiki, Security, Insights, and Settings. On the left sidebar, there's a "Workflows" section with a "New workflow" button and a list of workflows, including "Simple Pipe". The main content area is titled "All workflows" and shows "Showing runs from all workflows". Below this is a search bar "Filter workflow runs". A table lists workflow runs with columns for Event, Status, Branch, and Actor. One run is shown: "Rename extra/simple-pipe.yml to .github/w..." on the "main" branch, triggered by "Simple Pipe #1: Commit 2393804 pushed by gwstudent", and is currently "In progress" (23 seconds ago).

Event	Status	Branch	Actor
Rename extra/simple-pipe.yml to .github/w...	In progress	main	gwstudent





How actions work (jobs, steps, actions, shell cmds)

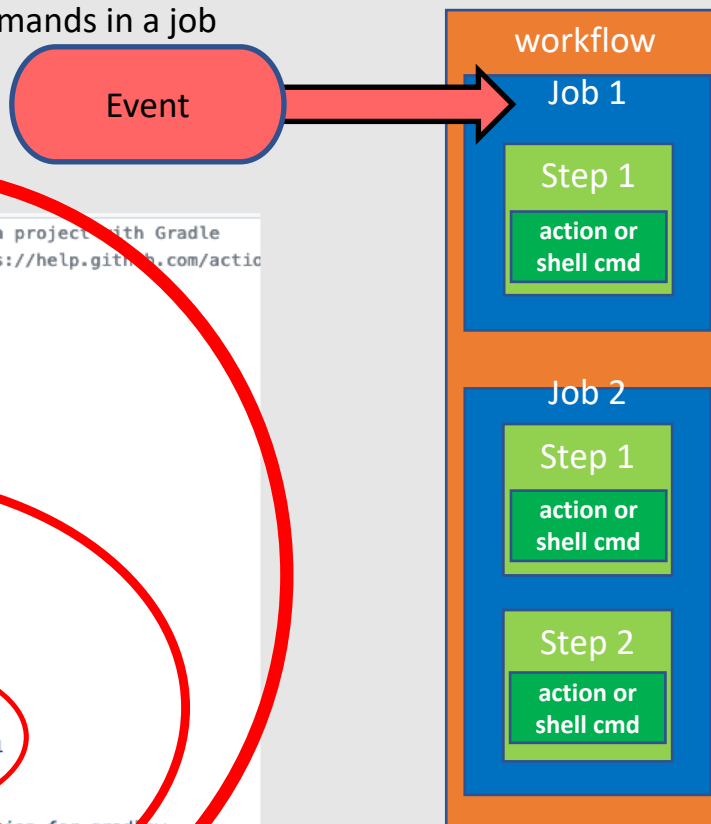
Workflows contain jobs

- Set of steps
- Workflow with multiple jobs runs them in parallel

Jobs contain steps

- A task that can execute commands in a job
- Can be either
 - » action
 - » shell command

```
1 # This workflow will build a Java project with Gradle
2 For more information see: https://help.github.com/actions
3
4 name: Java CI with Gradle
5
6 on:
7   push:
8     branches: [ main ]
9   pull_request:
10    branches: [ main ]
11
12 jobs:
13   build:
14
15     runs-on: ubuntu-latest
16
17     steps:
18     - uses: actions/checkout@v2
19     - name: Set up JDK 1.8
20       uses: actions/setup-java@v1
21       with:
22         java-version: 1.8
23     - name: Grant execute permission for gradlew
24       run: chmod +x gradlew
25     - name: Build with Gradle
26       run: ./gradlew build
```



action

- Independent commands
- Based off repository in GitHub
- Used in steps to form a job
- Smallest unit in a workflow
- Can be created or pulled in from the community
- Only usable if included as a step

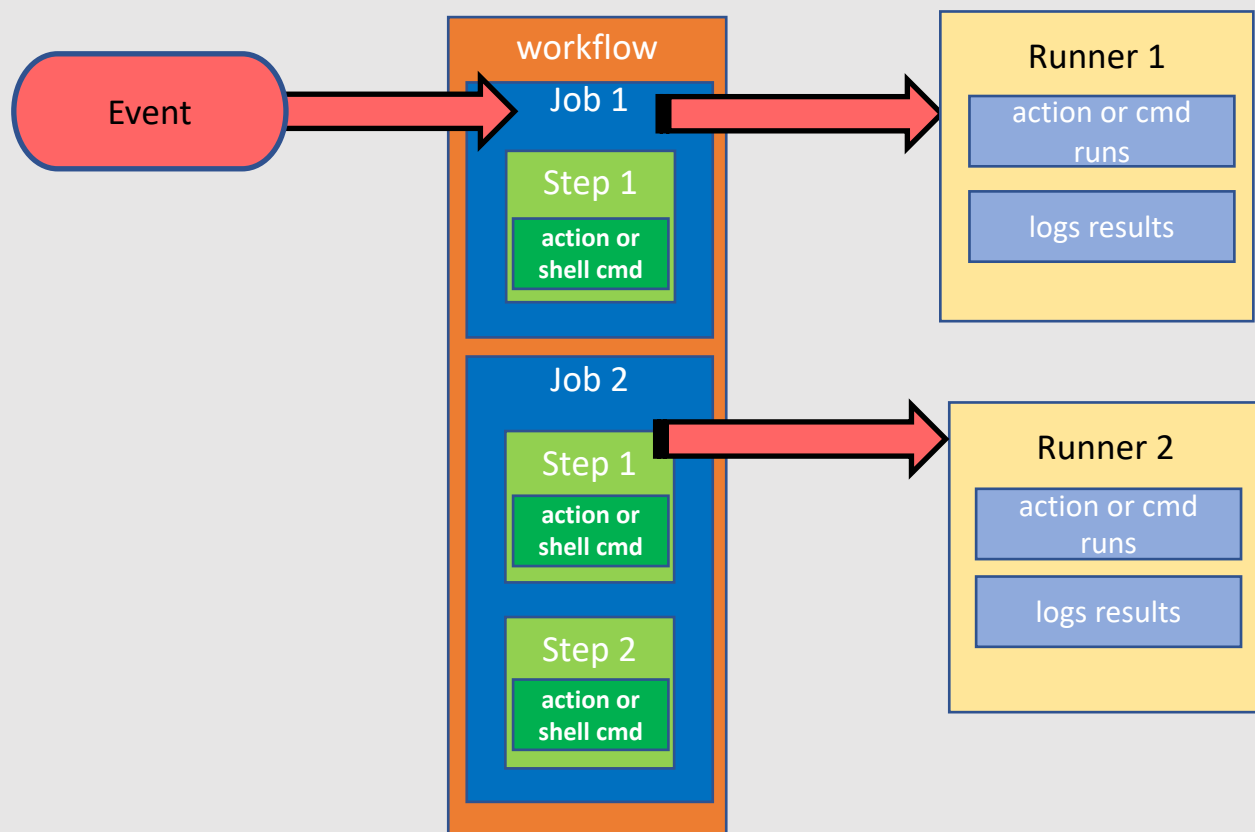


How actions work (runner)

19

Runner

- A server with GitHub Actions runner app on it
- Can use runner provided by/hosted via GitHub or use your own
- Runner listens for available jobs
- Steps in a job execute on the same runner

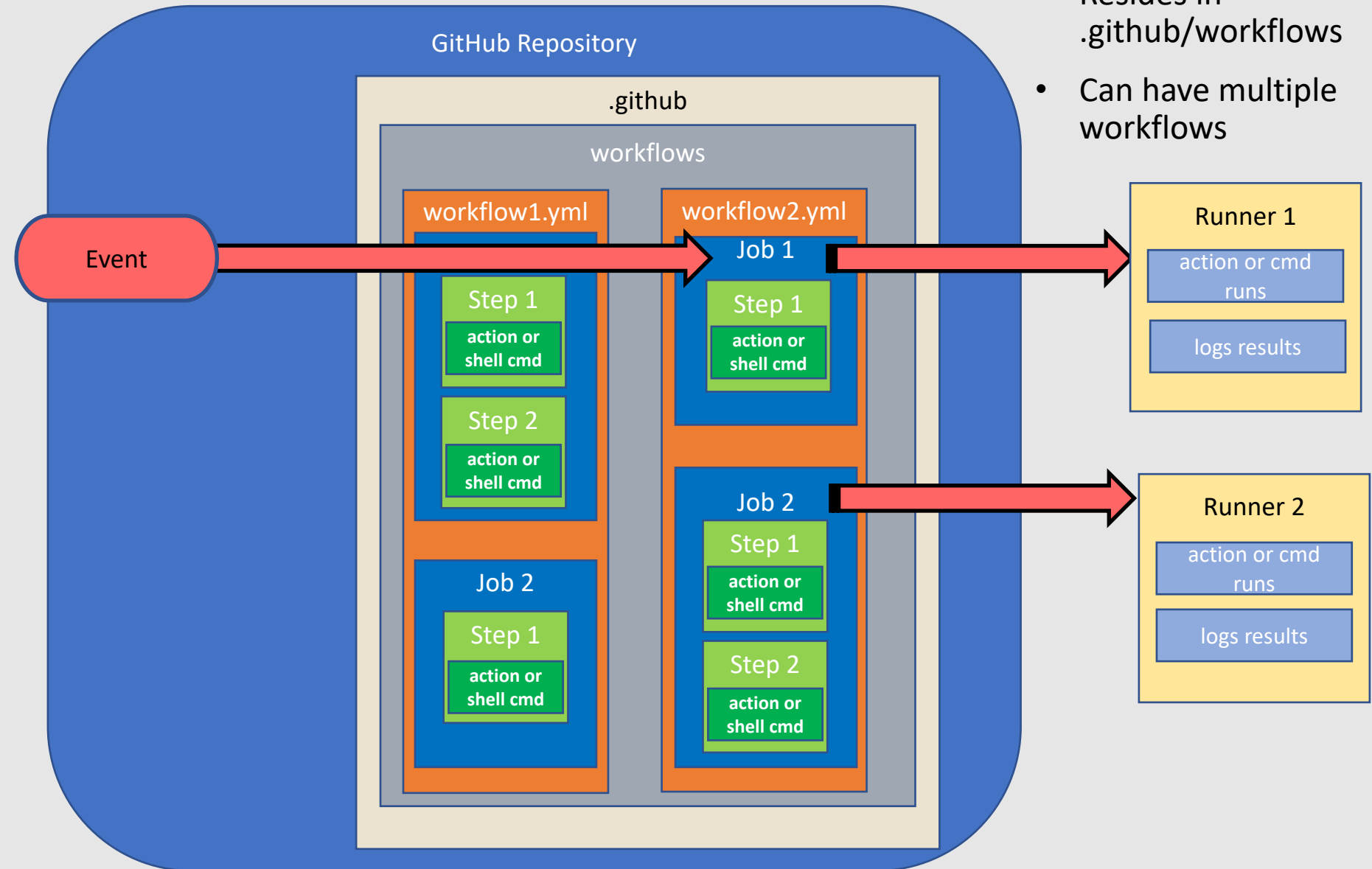




How actions work

20

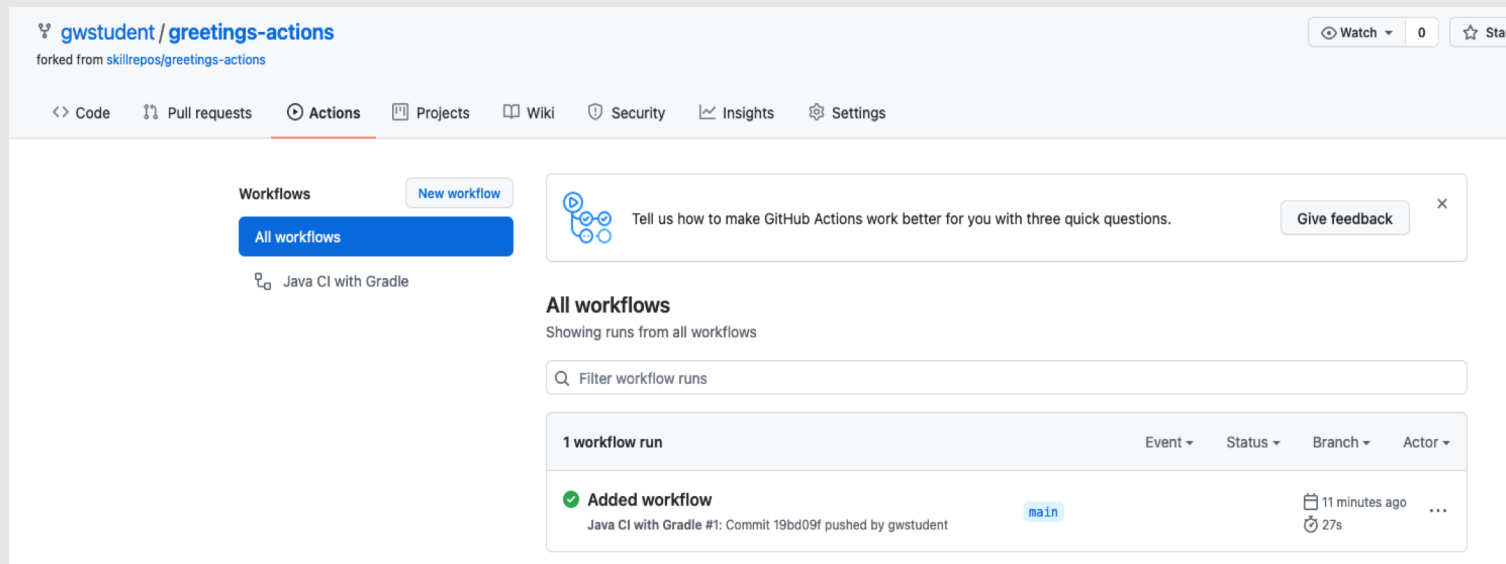
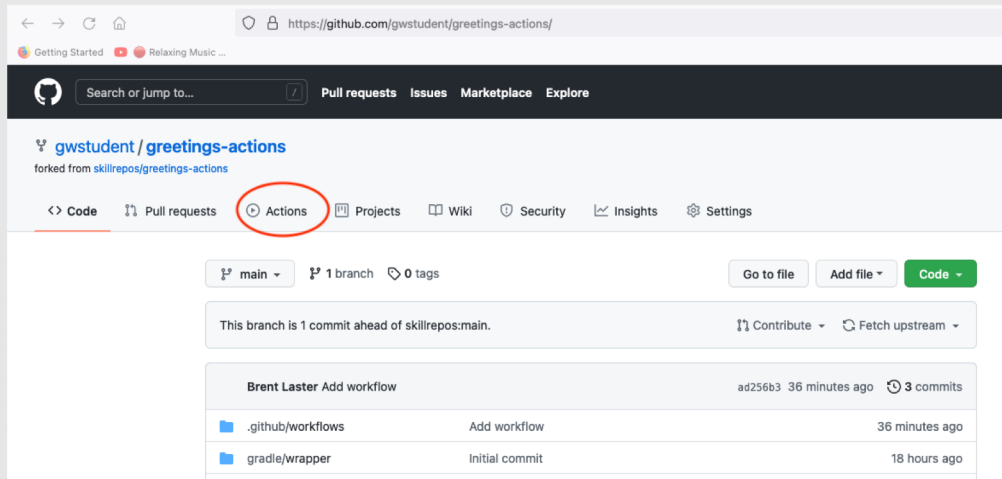
- Resides in `.github/workflows`
- Can have multiple workflows





Integration with GitHub Interface

21





Viewing Logs

22

All workflows
Showing runs from all workflows

Filter workflow runs

1 workflow run

✓ **Added workflow**
Java CI with Gradle #1: Commit 19bd09f pushed by

https://github.com/gwstudent/greetings-actions

gwstudent / greetings-actions
forked from skillrepos/greetings-actions

Code Pull requests **Actions** Projects Wiki Settings

✓ **Added workflow Java CI with Gradle #1**

Summary

Triggered via push 18 minutes ago
gwstudent pushed -> 19bd09f

Jobs

✓ **build**

gradle.yml
on: push

✓ **build**

https://github.com/gwstudent/greetings-actions/runs/3478059147?check_suite_focus=true

Getting Started Relaxing Music ...

Search or jump to... Pull requests Issues Marketplace Explore

gwstudent / greetings-actions
forked from skillrepos/greetings-actions

Code Pull requests **Actions** Projects Wiki Security Insights Settings

✓ **Added workflow Java CI with Gradle #1**

Summary

Jobs

✓ **build**

build
succeeded 22 minutes ago in 17s

- ✓ Set up job
- ✓ Run actions/checkout@v2
- ✓ Set up JDK 1.8
- ✓ Grant execute permission for gradlew
- ✓ **Build with Gradle**

```
1 ▶ Run ./gradlew build
8 Downloading https://services.gradle.org/distributions/gradle-4.10-bi
9 .....
10
11 Welcome to Gradle 4.10!
12
13 Here are the highlights of this release:
14 - Incremental Java compilation by default
15 - Periodic Gradle caches cleanup
16 - Gradle Kotlin DSL 1.0-RC3
17 - Nested included builds
18 - SNAPSHOT plugin versions in the `plugins {}` block
19
```





Editing Workflows

23

The screenshot shows a GitHub repository named `gwstudent/greetings-actions`, which is a fork of `skillrepos/greetings-actions`. The repository is public and has 0 watchers. The navigation bar includes links for Code, Issues (2), Pull requests, Actions, Projects, Wiki, Security, and Insights. The current view is the `simple-pipe.yml` workflow file in the `main` branch. The file content is as follows:

```
1 # This workflow will build a Java project with Gradle initially
2 # For more information see: https://help.github.com/actions/language-and-framework-guides/building-and-testing-java-with-gradle
3
4 name: Simple Pipe
5
6 on:
7   push:
8     branches: [ main ]
9   pull_request:
10     branches: [ main ]
11   workflow_dispatch:
12     inputs:
13       myValues:
14         description: 'Input Values'
15
16 jobs:
17   build:
18     runs-on: ubuntu-latest
19
20     steps:
21     - uses: actions/checkout@v2
22     - name: Set up JDK 1.8
23       uses: actions/setup-java@v1
24       with:
25         java-version: 1.8
26     - name: Grant execute permission for gradlew
27       run: chmod +x gradlew
28     - name: Build with Gradle
29       run: ./gradlew build
30     - name: Upload Artifact
31       uses: actions/upload-artifact@v2
32       with:
33         name: build-artifact
34         path: build/
```

The `simple-pipe.yml` file path is highlighted with a red circle. A `Commit changes` dialog box is open, showing the commit message `Create simple-pipe.yml` and the extended description `Add an optional extended description..`. The dialog box has two radio buttons: `Commit directly to the main branch` (selected) and `Create a new branch for this commit and start a pull request`. The `Commit changes` button is highlighted with a red circle. The `Commit changes...` button in the top right corner of the repository view is also highlighted with a red circle.





Lab 1 – Creating a simple example

Purpose: In this lab, we'll get a quick start learning about GitHub Actions by creating a simple project that uses them. We'll also see what a first run of a workflow with actions looks like.





Self-hosted Runners

25

- Useful when you need more configurability and control
- Allows you to choose and customize configuration, system resources, available software, etc.
- Can be physical systems, VMs, containers, on-prem or cloud
- Runner system connects to GitHub via GitHub Actions self-hosted runner
- Automatically kicked off of GitHub if no connection to GitHub Actions after 30 days
- Recommended to use only with private repos
 - Forks of repos could run dangerous code on self-hosted runner machine (via PR that executes code in workflow)

Supported architectures and operating systems for self-hosted runners

Linux

- Red Hat Enterprise Linux 7 or later
- CentOS 7 or later
- Oracle Linux 7
- Fedora 29 or later
- Debian 9 or later
- Ubuntu 16.04 or later
- Linux Mint 18 or later
- openSUSE 15 or later
- SUSE Enterprise Linux (SLES) 12 SP2 or later

Windows

- Windows 7 64-bit
- Windows 8.1 64-bit
- Windows 10 64-bit
- Windows Server 2012 R2 64-bit
- Windows Server 2016 64-bit
- Windows Server 2019 64-bit

macOS

- macOS 10.13 (High Sierra) or later

Architectures

The following processor architectures are supported for the self-hosted runner application.

- x64 - Linux, macOS, Windows.
- ARM64 - Linux only.
- ARM32 - Linux only.





GitHub-hosted vs Self-hosted Runners

	GitHub-hosted	Self-hosted
Definition	Hosted by GitHub	Any system/configuration you want to use
Prereqs	Running GitHub actions runner application	GitHub actions self-hosted runner application
Platforms	Windows, Linux, MacOS	Any that can be made to work
Advantages	Quicker and simpler	Highly configurable
Management/Ownership	GitHub	As defined
Clean instance	For every job execution	Optional
Cost	Free minutes on GitHub plan with cost for overages	Free to use with actions, but owner is responsible for any other cost
Automatic updates	For OS, installed packages, tools, and hosted runner application	Only for self-hosted runner application
Implementation	Virtual	Virtual or physical





Creating self-hosted runners

27

- Repo->Settings->Actions->Runners->New self-hosted runner

The screenshot shows the GitHub interface for the repository 'gwstudent/greetings'. The browser address bar displays 'https://github.com/gwstudent/greetings/settings/actions/runners'. The repository is public and forked from 'brentlaster/greetings'. The 'Settings' tab is selected in the navigation bar. On the left sidebar, the 'Runners' section is highlighted under the 'Actions' category. The main content area is titled 'Runners' and includes a 'New self-hosted runner' button. Below this, a message states 'There are no runners configured' with a link to 'Learn more about using runners'. The sidebar on the left lists various settings categories: General, Access (Collaborators, Moderation options), Code and automation (Branches, Tags), and Actions (General, Runners).





Configuring self hosted runners

28

Download

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

# Download the latest runner package
$ curl -o actions-runner-osx-x64-2.301.1.tar.gz -L https://github.com/actions/runner/releases/download/v2.301.1/actions-runner-osx-x64-2.301.1.tar.gz

# Optional: Validate the hash
$ echo "3e0b037ea67e9626e99e6d3ff803ce0d8cc913938ddd1948b3a410ac6a75b878  actions-runner-osx-x64-2.301.1.tar.gz" | shasum -a 256 -c

# Extract the installer
$ tar xzf ./actions-runner-osx-x64-2.301.1.tar.gz
```

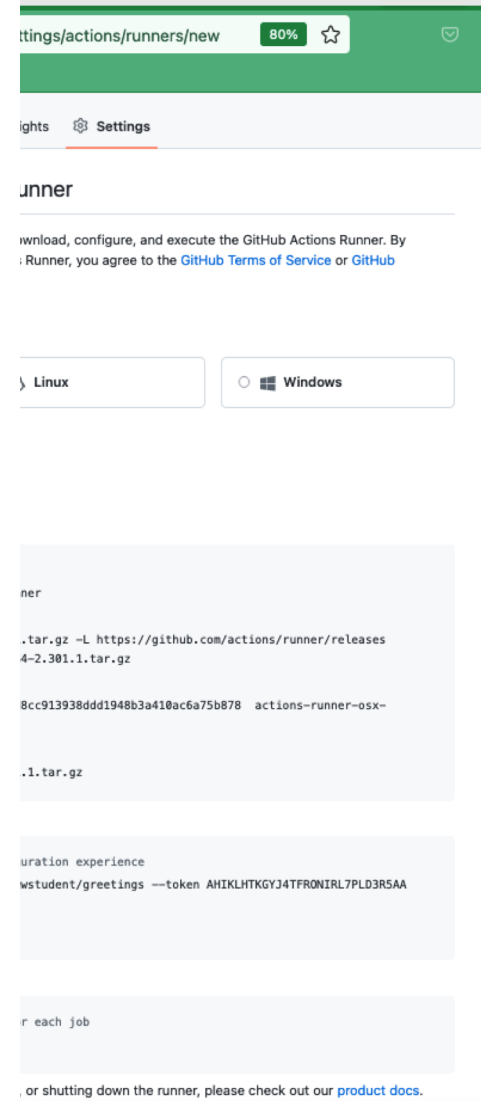
Configure

```
# Create the runner and start the configuration experience
$ ./config.sh --url https://github.com/gwstudent/greetings --token
AHIKLHUGDUGKANX5BKBZDH3D3R5UY

# Last step, run it!
$ ./run.sh
```

Using your self-hosted runner

```
# Use this YAML in your workflow file for each job
runs-on: self-hosted
```





- Virtual environments for GitHub actions hosted runners
- New VM for each job run
- VM images of Microsoft-hosted agents used for Azure Pipelines
- Updated periodically by GitHub

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

g Music ...

☰ README.md

Available Environments

Environment	YAML Label	Included Software	Latest Release & Rollout Progress
Ubuntu 20.04	ubuntu-latest or ubuntu-20.04	ubuntu-20.04	ubuntu20 20210816.1 (99.99%)
Ubuntu 18.04	ubuntu-18.04	ubuntu-18.04	ubuntu18 20210816.1 (99.99%)
macOS 11	macos-11	macOS-11	macos-11 20210814.QYCS9D (2.62%)
macOS 10.15	macos-latest or macos-10.15	macOS-10.15	macos-10.15 20210814.QY09X7 (3.74%)
Windows Server 2022 ^[beta]	windows-2022	windows-2022	windows-2022 20210819.7 (100.00%)
Windows Server 2019	windows-latest or windows-2019	windows-2019	windows-2019 20210815.1 (100.00%)
Windows Server 2016	windows-2016	windows-2016	windows-2016 20210815.1 (100.00%)







GitHub Actions Storage - Artifacts

- Actions allow you to persist data after job has completed - as artifacts
- Artifact - file or collection of files produced during workflow run
 - build and test output
 - log files
 - binary files
- Artifacts are uploaded during workflow run
 - can be shared between jobs in same workflow
 - visible in the UI
- Default retention period is 90 days

```
32     - name: Upload Artifact
33       uses: actions/upload-artifact@v2
34       with:
35         name: greetings-jar
36         path: build/libs
37
```

Artifacts		
Produced during runtime		
Name	Size	
 greetings-jar	1006 Bytes	





What are artifacts?

- An item that is either a deliverable (something used by the final product) or included in a deliverable
- Examples:
 - executable file created by compiling source that links in several other libraries
 - compressed file such as a war or zip file that contains another set of files with it
- Artifacts get versioned, stored, and retrieved as needed for use in
 - assembly
 - testing
 - validation



Runner.Worker.dll

Microsoft dynamic link library - 334 KB

Information

Created	Wednesday, June 22, 2022 at 12:09 PM
Modified	Wednesday, June 22, 2022 at 12:09 PM



greetings-ci-0.6.0.jar

Java JAR file - 1 KB

Information

Created	Thursday, August 4, 2022 at 8:52 PM
Modified	Friday, August 5, 2022 at 12:45 AM



tkn_0.24.0_Darwin_all.tar.gz

gzip compressed archive - 46.4 MB

Information

Created	Sunday, July 3, 2022 at 4:43 PM
Modified	Sunday, July 3, 2022 at 4:43 PM

[Show Less](#)





Managing Artifacts with GitHub Actions

- Actions provides functionality for managing artifacts
- As defined by Actions, artifacts = files or collections of files created as result of a workflow run and persisted in GitHub
- Provides the ability to persist artifacts created during a workflow run
- Jobs in the same workflow can access and use persisted artifacts - like projects in a pipeline
- Actions also provides ability to cache content to speed up future runs
- By default, GitHub keeps artifacts (and build logs) around for 90 days
- Artifacts must be uploaded to be persisted

The screenshot shows the GitHub Actions interface for a workflow run titled "feat: upload artifact Java CI with Gradle #18". The workflow is successful, and an artifact named "greetings-jar" is listed under the "Artifacts" section. The artifact is 1006 Bytes in size. The "greetings-jar" artifact is circled in red.

Name	Size
greetings-jar	1006 Bytes



Searching for a Public Action

Marketplace Documentation

github.com/marketplace/actions/upload-a-build-artifact

upl

Marketplace / Actions / Upload a Build Artifact

Upload a Build Artifact

Upload a build artifact that can be used by subsequent workflow steps

INSTALLATION

Copy and paste the following snippet into your .yaml file.

```
- name: Upload a Build Artifact
  uses: actions/upload-artifact@v3.1.3
```

[Learn more about this action in actions/upload-artifact](#)

- Easier upload
 - Specify a wildcard pattern
 - Specify an individual file

Use latest version

Verified creator

GitHub has verified that this action was created by actions.

[Learn more about verified Actions.](#)

Stars

Star 2.4k

Contributors

```
- name: Upload a Build Artifact
  uses: actions/upload-artifact@v
  with:
    # Artifact name
```



Actions use Workflows

The image is a collage of three screenshots from the GitHub Actions ecosystem.

Top Left Screenshot: Shows the 'Checkout V2' action page on the GitHub Marketplace. The page includes the GitHub Action logo, the name 'Checkout', version 'v2.3.4', and a 'Use latest version' button. The description states: 'This action checks out your repository under \$GITHUB_WORKSPACE, so your workflow can access it.' It also mentions 'What's new' with an improved performance and fetches only a single commit by default.

Top Right Screenshot: Shows the 'actions/checkout' repository page. The 'Actions' tab in the navigation bar is circled in red. The page displays statistics like 41 Watchers, 1.8k Stars, and 613 Forks. A blue banner encourages using the GitHub Action with a project.

Bottom Screenshot: Shows the 'actions/checkout/actions' page, displaying a list of workflow runs. The 'All workflows' section is active, showing 584 workflow runs. The table lists runs with columns for Event, Status, Branch, and Actor.

Event	Status	Branch	Actor
Build and Test #532: Commit afe4af0 pushed by thboop	Success	main	...
Licensed #62: Commit afe4af0 pushed by thboop	Success	main	...



Upload a Build Artifact Action Usage

35

- Note: "my-artifact" is a generic reference - not the actual filename

github.com/marketplace/actions/upload-a-b...

Usage

See [action.yml](#)

Upload an Individual File

```
steps:
- uses: actions/checkout@v2

- run: mkdir -p path/to/artifact

- run: echo hello > path/to/artifact/world.txt

- uses: actions/upload-artifact@v3
  with:
    name: my-artifact
    path: path/to/artifact/world.txt
```

Upload an Entire Directory

```
- uses: actions/upload-artifact@v3
  with:
    name: my-artifact
    path: path/to/artifact/ # or path/to/artifact
```

Upload using a Wildcard Pattern

```
- uses: actions/upload-artifact@v3
  with:
    name: my-artifact
    path: path/**/[abc]rtifac?/*
```

github.com/actions/upload-artifact/blob/main...

main upload-artifact / action.yml

thboop Update to node16 (#293) Latest commit 2244c82 on Feb 7 History

5 contributors

28 lines (26 sloc) | 941 Bytes

Raw Blame

```
1 name: 'Upload a Build Artifact'
2 description: 'Upload a build artifact that can be used by subsequent workflow steps'
3 author: 'GitHub'
4 inputs:
5   name:
6     description: 'Artifact name'
7     default: 'artifact'
8   path:
9     description: 'A file, directory or wildcard pattern that describes what to upload'
10    required: true
11    if-no-files-found:
12      description: >
13        The desired behavior if no files are found using the provided path.
14
15    Available Options:
16      warn: Output a warning but do not fail the action
17      error: Fail the action with an error message
18      ignore: Do not output any warnings or errors, the action does not fail
19    default: 'warn'
20  retention-days:
21    description: >
22      Duration after which artifact will expire in days. 0 means using default retention.
23
24    Minimum 1 day.
25    Maximum 90 days unless changed from the repository settings page.
26 runs:
27   using: 'node16'
28   main: 'dist/index.js'
```





Downloading artifact

36

- In workflow run, click on artifact name
- Download and expand

fix: update package.json Java CI with Gradle #23 Re-run all jobs ...

Summary

Jobs

- build

Triggered via push 4 days ago


gwestudent pushed → 20d8f33 [artifact](#)

Status: **Success** Total duration: 27s Artifacts: 1

pipeline.yml
on: push



build 17s

Artifacts
Produced during runtime

Name	Size
 greetings-jar	1006 Bytes



build summary ...


Gradle Builds

Root Project	Requested Tasks	Gradle Version	Build Outcome	Build Scan™
greetings-ci	build	4.10		 Build Scan™ NOT PUBLISHED

< > Searching "This Mac"

Search: **This Mac** "Downloads" Save +

Name	Kind	Date Last Opened
 greetings-ci-0.6.0.jar	Java JAR file	--
 greetings-jar.zip	ZIP archive	--



greetings-ci-0.6.0.jar





Lab 2 – Learning more about Actions

Purpose: In this lab, we'll see how to get more information about Actions and how to update our workflow to use others.





- Can be created by you
- Can result from customizing community actions
- Can be put on Marketplace and shared
 - repository must be public
- Can integrate with any public API
- Can run directly on a machine or in Docker container
- Can define inputs, outputs, and environment variables
- Require metadata file
 - defines inputs, outputs, and entrypoint
 - must be named either action.yaml or action.yml
- Should be tagged with a label and then pushed to GitHub
- To use:
 - In file in .github/workflows/main.yml
 - In steps
 - "uses: <github path to action>@<label>

```
$ <create GitHub repo>
$ <clone repo>
$ <create files>

$ git add action.yml <other files>
$ git commit -m "action files"
$ git tag -a -m "first release of action" v1
$ git push --follow-tags
```

.github/workflows/example.yml

```
on: [push]

jobs:
  example_job:
    runs on: ubuntu-latest
    name: An example job
    steps:
      - name: Example step
        id: example_step
        uses: <repo name>/<action name>@<tag>
      ...
```



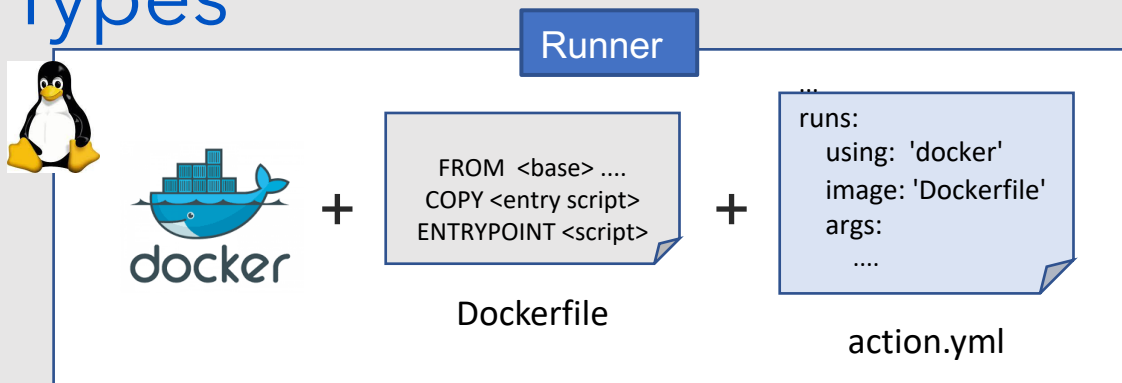


Custom Action Types

39

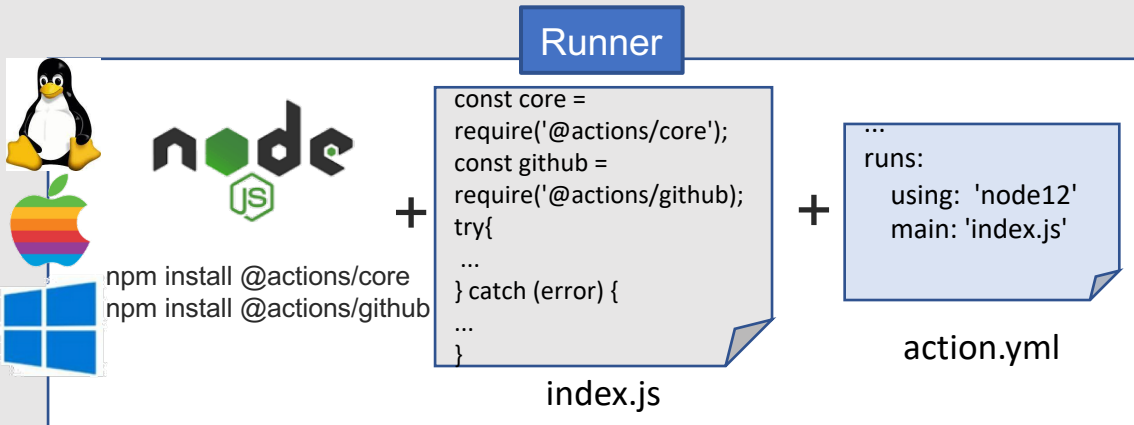
- Docker

- Packages env with actions code
- Env can include specific OS versions, config, tools, etc.
- Well suited for specific env needs
- Runs only on Linux runners with Docker installed
- Slower due to cost of building and retrieving container
- GitHub builds image from Dockerfile and runs commands in a new container based on image



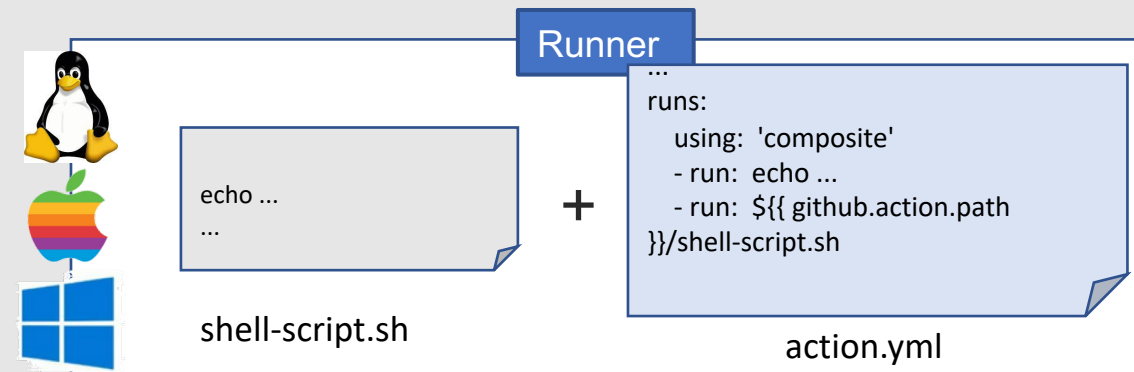
- Javascript

- Run directly on runner for any of macOS, Linux, or Windows
- Separates action from env
- Fast than Docker
- Needs to be pure JavaScript (no other binaries)
- Can use binaries already on runner



- Composite

- Combines multiple workflow steps in one action





Action Yaml File

40

- Key metadata file for a GitHub Action
- Can be named *action.yml* or *action.yaml*
- Has a well-defined format
- Defines
 - inputs
 - outputs
 - configuration for the action
 - configuration information includes
 - basic identification information
 - optional branding information
 - details about the kind of environment action is intended to run in
 - any special settings for that environment

The screenshot shows a GitHub repository interface for a file named `cache / action.yml`. At the top, it indicates the file is on the `main` branch. Below this, a merge pull request #554 is shown, submitted by Phantsure. The file is 27 lines long (27 sloc) and 946 bytes. The content of the file is as follows:

```
1 name: 'Cache'
2 description: 'Cache artifacts like dependencies and build outputs to improve workflow execut.
3 author: 'GitHub'
4 inputs:
5   path:
6     description: 'A list of files, directories, and wildcard patterns to cache and restore'
7     required: true
8   key:
9     description: 'An explicit key for restoring and saving the cache'
10    required: true
11  restore-keys:
12    description: 'An ordered list of keys to use for restoring stale cache if no cache hit o
13    required: false
14  upload-chunk-size:
15    description: 'The chunk size used to split up large files during upload, in bytes'
16    required: false
17 outputs:
18   cache-hit:
19     description: 'A boolean value to indicate an exact match was found for the primary key'
20 runs:
21   using: 'node16'
22   main: 'dist/restore/index.js'
23   post: 'dist/save/index.js'
24   post-if: 'success()'
25 branding:
26   icon: 'archive'
27   color: 'gray-dark'
```





What is a Pull Request?

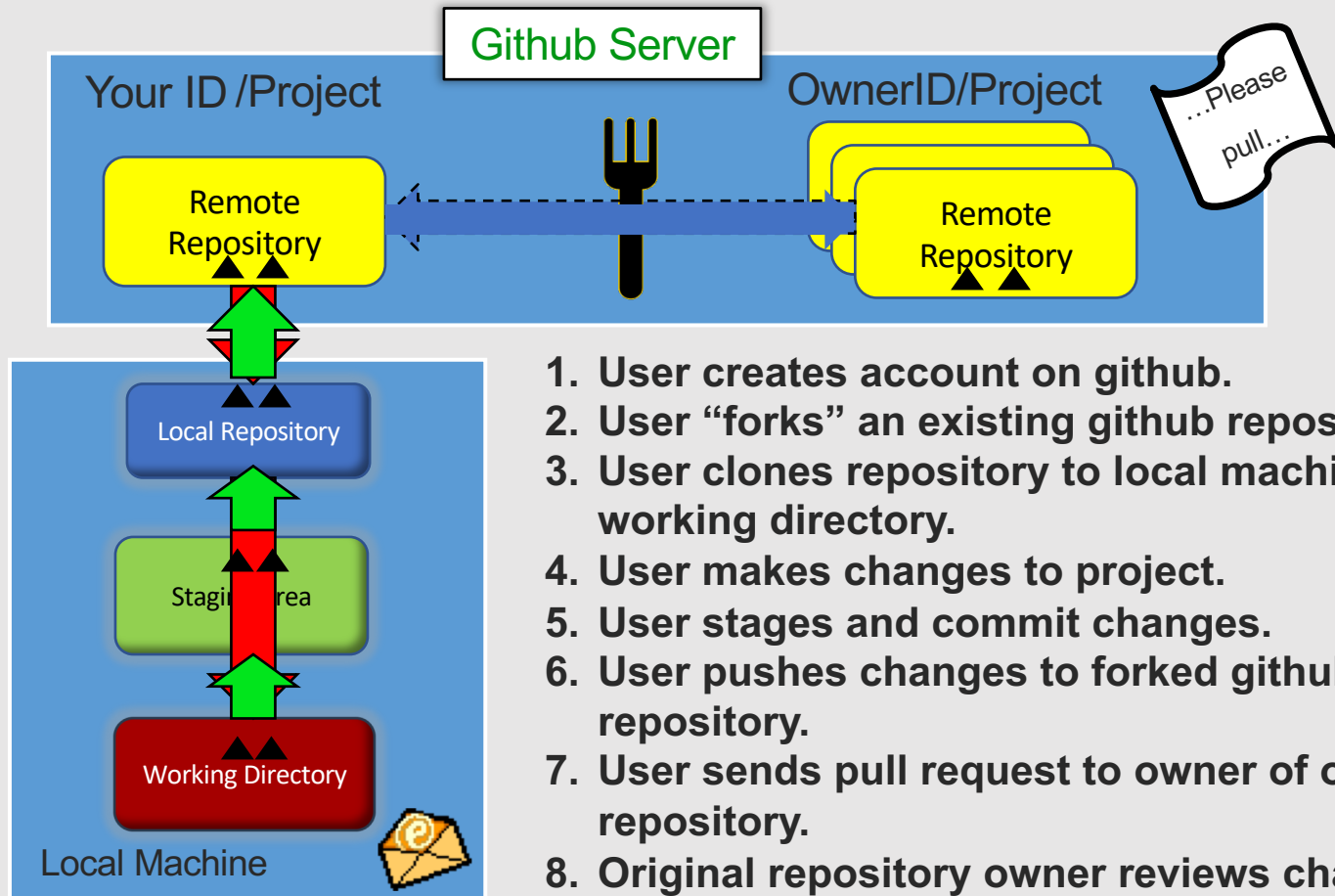
- GitHub model for contributing and gating changes into a repository.
- Proposed change is made in one place (source) and targeted for another place (destination)
- Source and destination can be:
 - Branch-to-branch in the same project
 - Project-to-project
- Source and destination are specified when you open (create) a new PR
- PR's can be automatically built for verification





Pull Requests

42





Creating a Pull Request after a push

43

- Can see changes and what's being compared
- Can modify source and target (items being compared)
- Lets you know if automatic merging is possible

Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#).

base: main ← compare: patch-1 ✓ Able to merge. These branches can be automatically merged.

Choose a base ref

Find a branch

Branches Tags

✓ main default

about pull requests

Create pull request

1 file changed 1 contributor

Verified ed7d430 <>

Split Unified

Showing 1 changed file with 1 addition and 1 deletion.

github/workflows/pipeline.yml

@@ -29,6 +29,6 @@ jobs:

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).

base repository: gwstudent/greetings-ci ← base: main ← head repository: gwstudent/greetings-ci ← compare: patch-1

✓ Able to merge. These branches can be automatically merged.





Continuous Integration with Pull Requests

44

- If can be merged automatically, GitHub Action workflows will be kicked off
- If GitHub Action workflows are successful (and all checks successful), “Merge” button enabled

Update pipeline.yml #2

Open gwstudent wants to merge 1 commit into main from patch-1

Conversation 0 Commits 1 Checks 0 Files changed 1

gwstudent commented now

No description provided.

Update pipeline.yml Verified ed7d438

gwstudent requested a review from brentlaster now

Add more commits by pushing to the patch-1 branch on gwstudent/greetings-ci.

Review requested
Review has been requested on this pull request. It is not required to merge. [Learn more.](#)

1 pending reviewer

Some checks haven't completed yet
1 in progress check

Java CI with Gradle / build (pull_request) In progress — This check has started... [Details](#)

This branch has no conflicts with the base branch
Merging can be performed automatically.

Reviewers: brentlaster
Still in progress

Assignees: No one—assign

Labels: None yet

Projects: None yet

Milestone: No milestone

Notifications: You're receiving the thread.

Changes approved
1 approving review [Learn more.](#)

1 approval

Require approval from specific reviewers before merging
Ensure specific people or teams approve pull requests before they're merged into your main branch. [Add rule](#) [×](#)

All checks have passed
1 successful check [Show all checks](#)

This branch has no conflicts with the base branch
Merging can be performed automatically.

Merge pull request You can also open this in GitHub Desktop or view [command line instructions.](#)

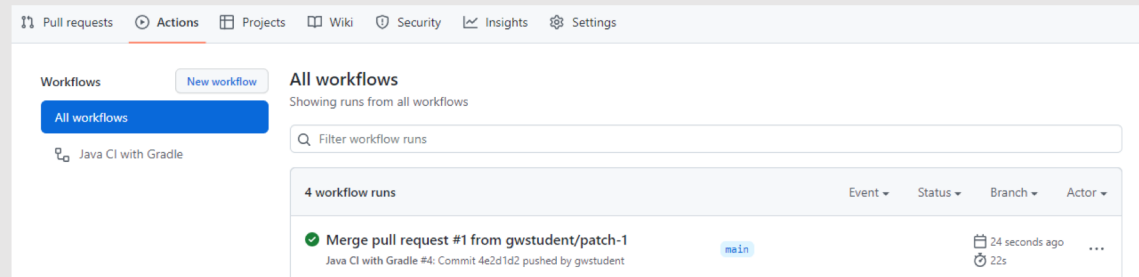
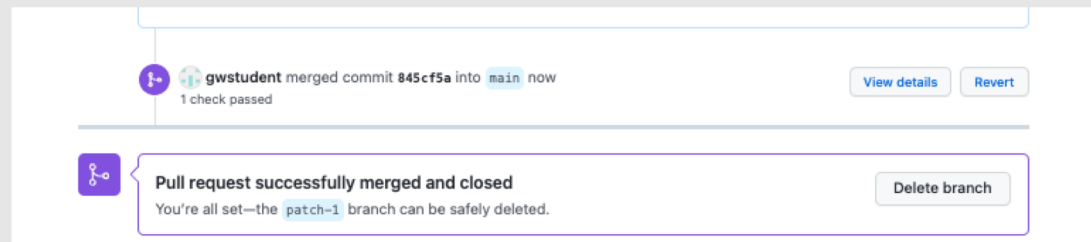
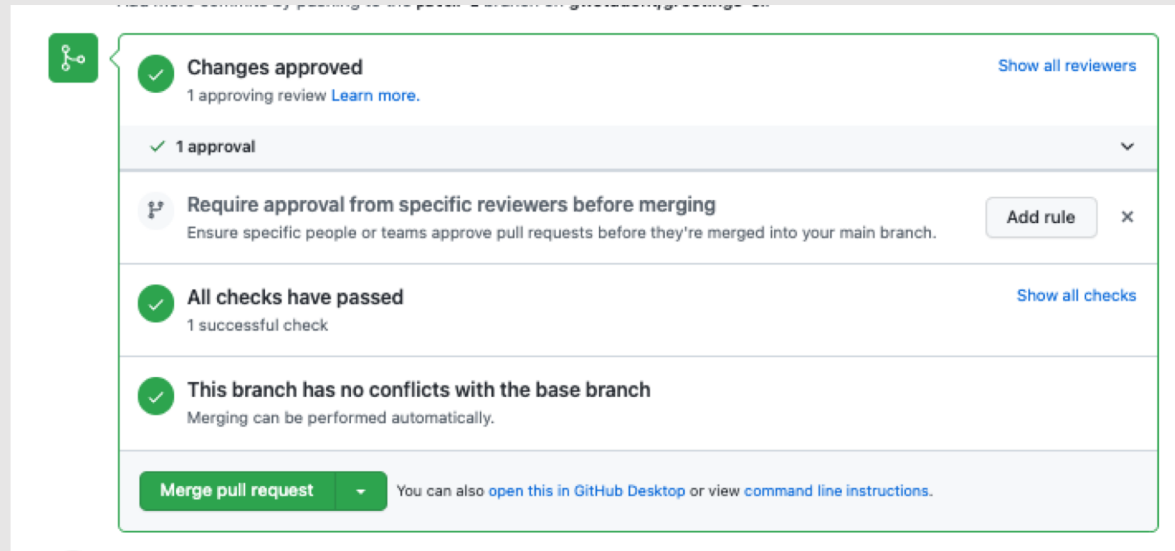




Completing the PR

45

- After checks have passed, you can click on “Merge” button and confirm
- After a moment, results will show in GitHub
- New workflow run will be kicked off





Lab 3 – Adding your own action

Purpose: In this lab, we'll see how to create how to add and use a custom GitHub Action.

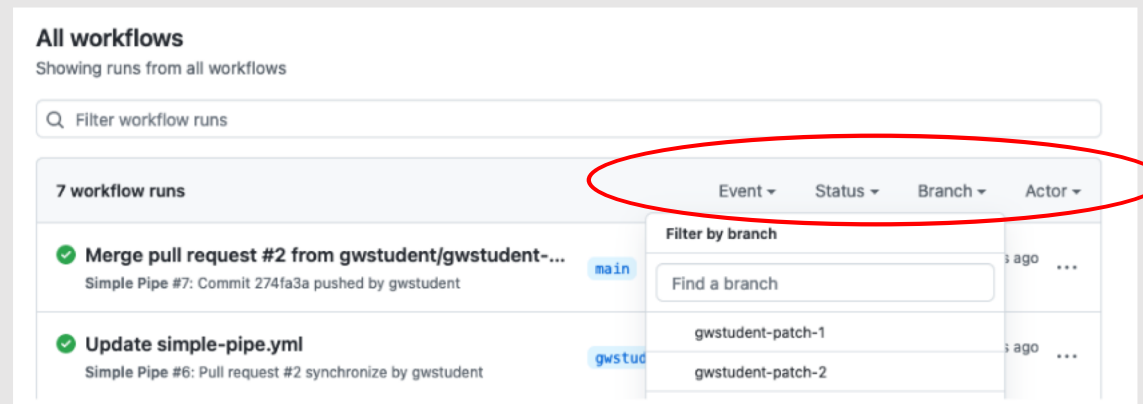
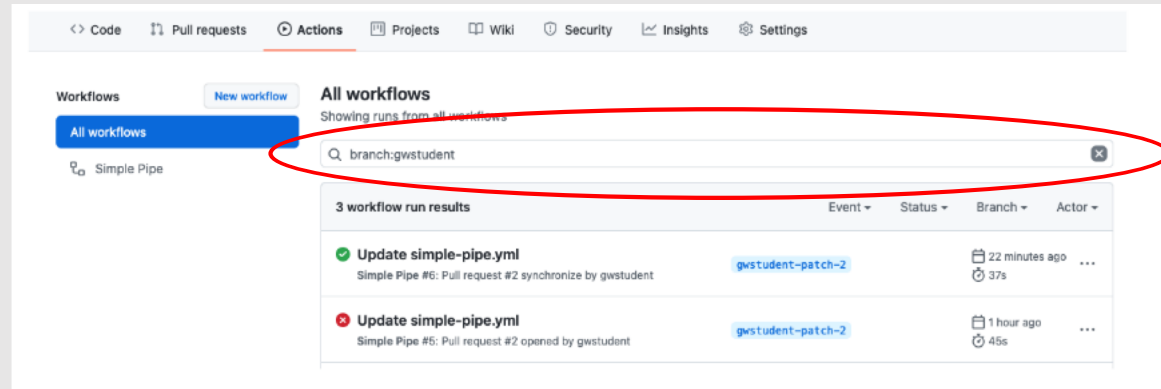




Filtering workflows

49

- Enter text in search bar
- Click on "x" at end to clear
- Click on category in "workflow run results" bar
- Select from the list





Creating a status badge

- Adds status information displayed in README file
- Click on "... " at end of selected line in workflow

The screenshot shows the GitHub interface for a repository named "greetings-actions". At the top, the "Workflows" section is visible, showing a workflow named "Simple Pipe" (simple-pipe.yml) with 7 workflow runs. A dropdown menu is open for the workflow, showing options: "Create status badge" (highlighted with a red circle), "Disable workflow", and "22 minutes ago". Below this, the "README.md" file is shown. The README content includes the title "greetings-actions" and a description "Simple hello world type of program for use with learning GitHub Actions". A status badge is displayed: "Java CI w/ Gradle and test-run" with a green "passing" label. At the bottom, the "Edit new file" view for README.md is shown, with a red arrow pointing to the end of the third line of code: `!![Java CI w/ Gradle and test-run](https://github.com/gwstudent/greetings-actions/actions/workflows/gradle2.yml/badge.svg)](https://github.com/gwstudent/greetings-actions)`.





Drilling into logs

51

- Click on workflow in list
- Click on step
- Step log is shown
- Gear icon can be used for options - such as timestamps
- Can also view raw logs

count-args
failed on Sep 6 in 3s

Search logs

✓ Show timestamps
Show full screen (Shift+F)
Download log archive
View raw logs

Run echo argument count is 0

```
39 Mon, 06 Sep 2021 18:33:28 GMT ##[debug]', steps.report-count.outputs.arg
40 Mon, 06 Sep 2021 18:33:28 GMT ##[debug]Evaluating format:
41 Mon, 06 Sep 2021 18:33:28 GMT ##[debug]..Evaluating String:
42 Mon, 06 Sep 2021 18:33:28 GMT ##[debug]..=> 'echo argument count is {0}'
43 Mon, 06 Sep 2021 18:33:28 GMT ##[debug]exit 1
44 Mon, 06 Sep 2021 18:33:28 GMT ##[debug]
45 Mon, 06 Sep 2021 18:33:28 GMT ##[debug]..Evaluating Index:
46 Mon, 06 Sep 2021 18:33:28 GMT ##[debug]...Evaluating Index:
47 Mon, 06 Sep 2021 18:33:28 GMT ##[debug].....Evaluating Index:
48 Mon, 06 Sep 2021 18:33:28 GMT ##[debug].....Evaluating steps:
49 Mon, 06 Sep 2021 18:33:28 GMT ##[debug].....=> Object
50 Mon, 06 Sep 2021 18:33:28 GMT ##[debug].....Evaluating String:
51 Mon, 06 Sep 2021 18:33:28 GMT ##[debug].....=> 'report-count'
52 Mon, 06 Sep 2021 18:33:28 GMT ##[debug].....=> Object
53 Mon, 06 Sep 2021 18:33:28 GMT ##[debug].....Evaluating String:
54 Mon, 06 Sep 2021 18:33:28 GMT ##[debug].....=> 'outputs'
55 Mon, 06 Sep 2021 18:33:28 GMT ##[debug]....=> Object
56 Mon, 06 Sep 2021 18:33:28 GMT ##[debug]...Evaluating String:
57 Mon, 06 Sep 2021 18:33:28 GMT ##[debug]....=> 'arg-count'
58 Mon, 06 Sep 2021 18:33:28 GMT ##[debug]..=> '0'
59 Mon, 06 Sep 2021 18:33:28 GMT ##[debug]> 'echo argument count is 0'
60 Mon, 06 Sep 2021 18:33:28 GMT ##[debug]exit 1
61 Mon, 06 Sep 2021 18:33:28 GMT ##[debug]
62 Mon, 06 Sep 2021 18:33:28 GMT ##[debug]Result: 'echo argument count is 0'
63 Mon, 06 Sep 2021 18:33:28 GMT ##[debug]exit 1
64 Mon, 06 Sep 2021 18:33:28 GMT ##[debug]
65 Mon, 06 Sep 2021 18:33:28 GMT ##[debug]Loading env
66 Mon, 06 Sep 2021 18:33:28 GMT ▶ Run echo argument count is 0
72 Mon, 06 Sep 2021 18:33:28 GMT ##[debug]/usr/bin/bash --noprofile --norc -e -o pipefail /home/runner
/work/_temp/87ab4e52-5e69-4ef4-8da4-906581805221.sh
argument count is 0
73 Mon, 06 Sep 2021 18:33:28 GMT
74 Mon, 06 Sep 2021 18:33:28 GMT Error: Process completed with exit code 1.
75 Mon, 06 Sep 2021 18:33:28 GMT ##[debug]Finishing: Run echo argument count is 0
```

> Complete job





Running your workflow manually

- Add a "workflow dispatch event" trigger
- Merge changes
- Afterwards will have "Run workflow" button in workflow list

The screenshot shows the GitHub Actions interface for a workflow named "Simple Pipe" (simple-pipe.yml). The top navigation bar includes links for Code, Pull requests, Actions (selected), Projects, Wiki, Security, Insights, and Settings. On the left, the "Workflows" section is active, showing "All workflows" and a button for "Simple Pipe". The main content area displays "9 workflow runs" with a search bar and filters for Event, Status, Branch, and Actor. A message states: "This workflow has a workflow_dispatch event trigger." To the right of this message is a "Run workflow" button, which is circled in red. Below the message, a list of workflow runs is shown, including "Update simple-pipe.yml", "Create README.md", and "Merge pull request #2 from gwstudent/gwstudent-p...". A dropdown menu is open from the "Run workflow" button, showing options for "Use workflow from" (Branch: main) and "Input Values" (abc def ghi), with a green "Run workflow" button at the bottom.





Lab 4 – Exploring logs

Purpose: In this lab, we'll take a closer look at the different options for getting information from logs.





Workflow commands

- Used to communicate with the runner machine to
 - Set environment variables
 - Set output values used by other actions
 - Add debug messages to logs
- Typically use "echo" command with certain format

```
echo "::workflow-command parameter1={data},parameter2={data}::{command value}"
```

- Also can be used to execute some commands in Actions Toolkit





Managing state and output

- Previously, you had workflow commands to set state ("save-state") and ("set-output")

```
- name: Save state
  run: echo "::save-state name={name}::{value}"
- name: Set output
  run: echo "::set-output name={name}::{value}"
```

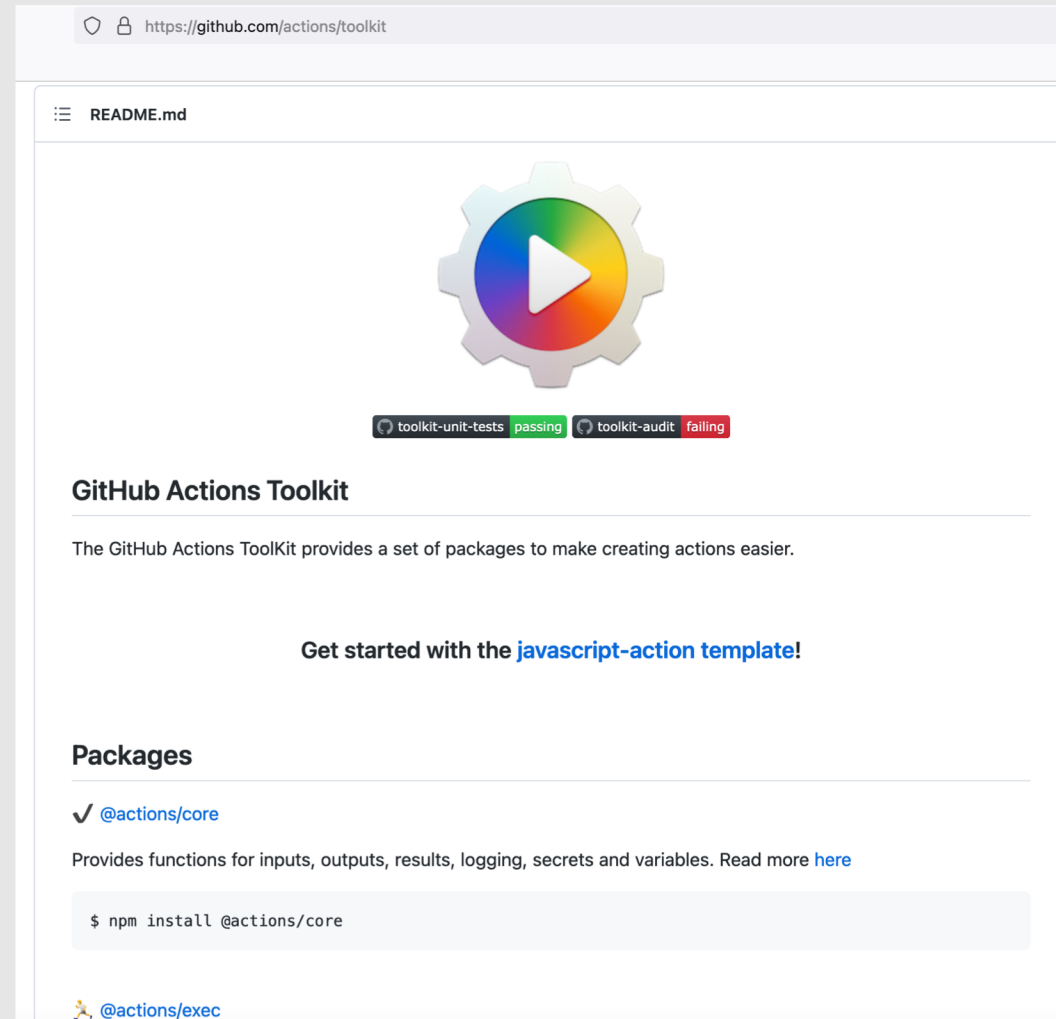
- Deprecated due to concerns over untrusted log data
- Actions now provides new environment files to manage state and output

```
- name: Save state
  run: echo "{name}={value}" >> $GITHUB_STATE
- name: Set output
  run: echo "{name}={value}" >> $GITHUB_OUTPUT
```





- Provides packages for more easily creating/working with actions
 - Functions in packages can be run in code as in
- ```
core.setOutput('SELECTED_COLOR', 'red');
```
- or (in many cases)
- run as workflow commands
- ```
- name: Set selected color run:  
  echo '::set-output  
  name=SELECTED_COLOR::green'
```





Actions Toolkit packages

57

- Collections of related functions
- Can be imported into code for actions

The screenshot displays the GitHub repository for `@actions/toolkit/packages/core`. The repository is public and has 84 watches, 2.8k stars, and 960 forks. The main branch is selected, and the file list shows the following files and their commit history:

File	Commit Message	Time
__tests__	Full release of actions/core 1.6.0 with oidc behavior (#919)	16 days ago
src	Full release of actions/core 1.6.0 with oidc behavior (#919)	16 days ago
LICENSE.md	Add License.md to all npm packages (#548)	14 months ago
README.md	Full release of actions/core 1.6.0 with oidc behavior (#919)	16 days ago
RELEASES.md	Full release of actions/core 1.6.0 with oidc behavior (#919)	16 days ago
package-lock.json	Full release of actions/core 1.6.0 with oidc behavior (#919)	16 days ago
package.json	Full release of actions/core 1.6.0 with oidc behavior (#919)	16 days ago
tsconfig.json	Add Bryan's core code	2 years ago

The `README.md` file is open, showing the following content:

```
@actions/core
```

Core functions for setting results, logging, registering secrets and exporting variables across actions

Usage

Import the package

```
// javascript
const core = require('@actions/core');

// typescript
import * as core from '@actions/core';
```





Monitoring and Troubleshooting

- Use "default" information
 - Visualization graph of a workflow
 - Workflow run history
 - Workflow logs
 - Print debug messages in logs
 - can use core.debug or "echo "::debug::<message>"
 - Display of debug messages can be via:
 - » Option on job rerun
 - » setting up repo secret OR variable
 - » name: ACTIONS_STEP_DEBUG
 - » value: true

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

steps:
- name: Print warning message
  run: |
    echo "::warning::This version is for debugging only."
- name: Dump context for runner
  env:
    RUNNER_CONTEXT: ${ toJson(runner) }
  run:
    echo "::debug::Runner context is above."
```

```
info
succeeded 32 seconds ago in 1s
Search logs

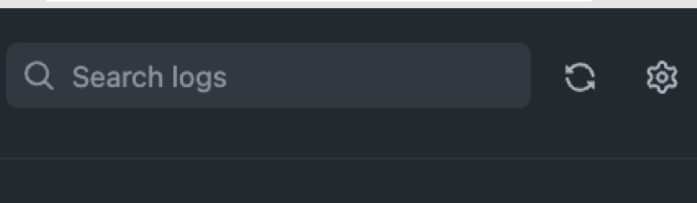
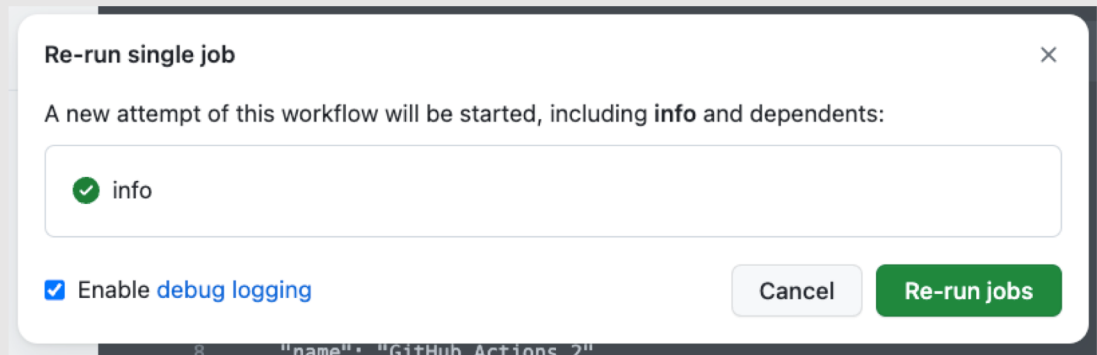
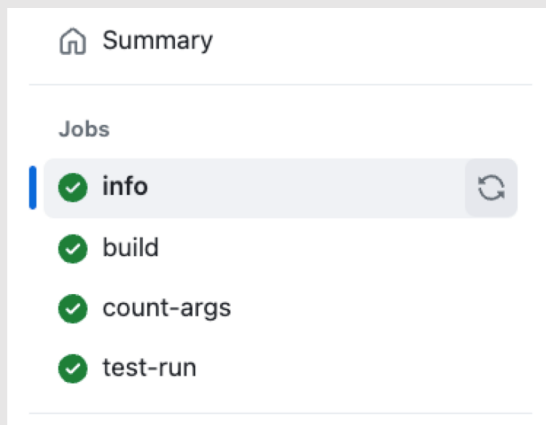
▼ [✓] Dump context for runner
15 ##[debug] "tool_cache": "/opt/hostedtoolcache",
16 ##[debug] "temp": "/home/runner/work/_temp",
17 ##[debug] "workspace": "/home/runner/work/greetings-actions"
18 ##[debug]]'
19 ##[debug]Evaluating condition for step: 'Dump context for runner'
20 ##[debug]Evaluating: success()
21 ##[debug]Evaluating success:
22 ##[debug]=> true
23 ##[debug]Result: true
24 ##[debug]Starting: Dump context for runner
25 ##[debug]Loading inputs
26 ##[debug]Loading env
27 ▶ Run echo "::debug::Runner context is above."
38 ##[debug]/usr/bin/bash -e /home/runner/work/_temp/4282ca63-108c-4656-8c6a-4a05c1ff90b3.sh
39 ##[debug]Runner context is above.
40 ##[debug]Finishing: Dump context for runner
```





Rerunning a job

- When job has completed, within 30 days, can rerun
- Curved arrows in jobs list or logs windows can be used to rerun
- Will rerun dependent jobs as well
- Option to rerun with debug info





Creating repo variables

The screenshot shows the GitHub repository settings page for 'Actions secrets and variables'. The 'Settings' tab is selected (1). In the left sidebar, 'Secrets and variables' is selected (2). The 'Variables' tab is selected (3). The 'New repository variable' button is highlighted (4).

Actions secrets and variables

Secrets and variables allow you to manage reusable configuration data. Secrets are **encrypted** and are used for sensitive data. [Learn more about encrypted secrets](#). Variables are shown as plain text and are used for **non-sensitive** data. [Learn more about variables](#).

Anyone with collaborator access to this repository can use these secrets and variables for actions. They are not passed to workflows that are triggered by a pull request from a fork.

Secrets Variables 3

New repository variable 4

Environment variables Manage environments

There are no variables for this repository's environments.

Repository variables

There are no variables for this repository.





Defining repo variables

⚙️ General

Access

👤 Collaborators

💬 Moderation options

Code and automation

🔑 Branches

🏷️ Tags

🎬 Actions

🔗 Webhooks

📁 Environments

💻 Codespaces

📄 Pages

Security

🔒 Code security and analysis

🔑 Deploy keys

* Secrets and variables

Actions

Codespaces

Actions variables / New variable

Note: Variable values are exposed as plain text. If you need to encrypt and mask sensitive information, [create a secret](#) instead.

Name *

ACTIONS_STEP_DEBUG

- Alphanumeric characters ([a-z], [A-Z], [0-9]) or underscores (_) only.
- Spaces are not allowed.
- Cannot start with a number.
- Cannot start with GITHUB_ prefix.

Value *

true

Add variable





Lab 5 – Looking at debug info

Purpose: In this lab, we'll look at some ways to get more debugging info from our workflows.



- Provides specific accesses
- Replaces passwords
- Can be done through <https://github.com/settings/tokens>
- Or can be setup via Settings->Profile->Developer Settings->Personal Access Tokens
- Generate new token and save it

The screenshot shows the 'New personal access token' page on GitHub. On the left, there is a sidebar with 'GitHub Apps', 'OAuth Apps', and 'Personal access tokens' (which is highlighted). The main content area has the title 'New personal access token'. Below the title, there is a note explaining that personal access tokens function like ordinary OAuth access tokens. A text input field labeled 'workflows' is shown with the question 'What's this token for?'. Below that, the 'Expiration' section shows a dropdown set to '30 days' and a note that the token will expire on Tue, Oct 5 2021. The 'Select scopes' section explains that scopes define the access for personal tokens and provides a link to 'Read more about OAuth scopes'. A table of scopes is displayed with the following items:

<input checked="" type="checkbox"/> repo	Full control of private repositories
<input checked="" type="checkbox"/> repo:status	Access commit status
<input checked="" type="checkbox"/> repo_deployment	Access deployment status
<input checked="" type="checkbox"/> public_repo	Access public repositories
<input checked="" type="checkbox"/> repo:invite	Access repository invitations
<input checked="" type="checkbox"/> security_events	Read and write security events
<input checked="" type="checkbox"/> workflow	Update GitHub Action workflows
<input type="checkbox"/> write:packages	Upload packages to GitHub Package Registry

The screenshot shows the 'tokens' page on GitHub. At the top right, there is a button labeled 'Generate new token'. Below the title, there is a note explaining that the generated token can be used to access the GitHub API. A large blue box contains the text: 'your personal access token now. You won't be able to see it again!'. Below this, the generated token is displayed: 'fQtQwNXWU9u2mXYJNVa3Ksu4Iz0iG'. The token is highlighted in a green box, and a red circle is drawn around the copy icon (a document with a plus sign) next to the token.



Creating a repo secret

64

<> Code Pull requests Actions Projects Wiki Security Insights **1** Settings

General

Access

Collaborators

Moderation options

Code and automation

Branches

Tags

Actions

Webhooks

Environments

Codespaces

Pages

Security

Code security and analysis

Deploy keys

2 Secrets and variables

Actions

Actions secrets and variables

Secrets and variables allow you to manage reusable configuration data. Secrets are **encrypted** and are used for sensitive data. [Learn more about encrypted secrets.](#) Variables are shown as plain text and are used for **non-sensitive** data. [Learn more about variables.](#)

Anyone with collaborator access to this repository can use these secrets and variables for actions. They are not passed to workflows that are triggered by a pull request from a fork.

Secrets Variables

3 New repository secret

Environment secrets Manage environments

There are no secrets for this repository's environments.

Repository secrets

There are no secrets for this repository.





- Workflows can invoke GitHub or other functionality via techniques such as curl

```
# This is a basic workflow to help you get started with Actions

name: create-failure-issue

# Controls when the workflow will run
on:
  # Allows you to run this workflow manually from the Actions tab
  workflow_dispatch:
    inputs:
      title:
        description: 'Issue title'
        required: true
      body:
        description: 'Issue body'
        required: true

# A workflow run is made up of one or more jobs that can run sequentially or in parallel
jobs:

  create_issue_on_failure:
    runs-on: ubuntu-latest

    permissions:
      issues: write
    steps:
      - name: Create issue using REST API
        run: |
          curl --request POST \
            --url https://api.github.com/repos/${{ github.repository }}/issues \
            --header 'authorization: Bearer ${{ secrets.GITHUB_TOKEN }}' \
            --header 'content-type: application/json' \
            --data '{
              "title": "${{ github.event.inputs.title }}",
              "body": "${{ github.event.inputs.body }}"
            }' \
            --fail
```



- Can be done via GitHub API calls
- Requires API Token saved as a secret
- One workflow then invokes the other via API call

create-issue-on-failure:

```
runs-on: ubuntu-latest
needs: [test-run, count-args]
if: always() && failure()
steps:
  - name: invoke workflow to create issue
    run: >
      curl -X POST
      -H "authorization: Bearer ${ secrets.WORKFLOW_USE }"
      -H "Accept: application/vnd.github.v3+json"
      "https://api.github.com/repos/${ github.repository }}/actions/workflows/create-failure-issue.yml/dispatches"
      -d '{"ref":"main",
        "inputs":
          {"title":"Automated workflow failure issue for commit ${ github.sha }}",
          "body":"This issue was automatically created by the GitHub Action workflow ** ${ github.workflow } **"}
      '
```



- Workflow able to be called from another workflow
- Avoids duplication/increases maintainability & reuse
- Workflow that uses another workflow is "caller"
- Reuse calls entire workflow, as if it were in caller
- Reusable workflow can be
 - [{owner}/{repo}/.github/workflows/{filename}@{ref}](#) for other repos
 - » {ref} can be any valid identifier (SHA, tag, branch)
 - [./github/workflows/{filename}](#) if in same repo
 - » {filename} is actual yaml file name
- Reusable workflow is defined via workflow_call trigger
- Reusable workflow is invoked via job.<job id>.uses





Reusable workflow - callable

68

```
1 # This is a reusable workflow for creating an issue
2
3 name: create-failure-issue
4
5 # Controls when the workflow will run
6
7 on:
8   # Allows you to run this workflow from another workflow
9
10  workflow_call:
11    inputs:
12      title:
13        required: true
14        type: string
15      body:
16        required: true
17        type: string
18
19  # Allows you to call this manually
20  workflow_dispatch:
21    inputs:
22      title:
23        description: 'Issue title'
24        required: true
25      body:
26        description: 'Issue body'
27        required: true
28
29
30  jobs:
31
32    create_issue_on_failure:
33      runs-on: ubuntu-latest
34
35      permissions:
36        issues: write
37
38      steps:
39        - name: Create issue using REST API
40          run: |
41            curl --request POST \
42              --url https://api.github.com/repos/${{ github.repository }}/issues \
43              --header 'authorization: Bearer ${{ secrets.GITHUB_TOKEN }}' \
44              --header 'content-type: application/json' \
45              --data '{
46                "title": "Failure: ${{ inputs.title }}",
47                "body": "Details: ${{ inputs.body }}"
48              }' \
49              --fail
```





Reusable workflow example - caller

69

```
1  create-issue-on-failure:
2
3    needs: [test-run, count-args]
4    if: always() && failure()
5    uses: ../github/workflows/create-failure-issue.yml
6    with:
7      title: "Automated workflow failure issue for commit ${ github.sha }"
8      body: "This issue was automatically created by the GitHub Action workflow ** ${ github.workflow } **"
9
```





Expressions and Conditions

70

- Expression
 - Any combination of literal values, references to a context (github, env, job, runner, etc.), or functions.
 - Can combine using operators.
 - Can be used to set environment variables
 - Commonly used with "conditional" "if" keyword
 - » Used to determine if step should run
 - » If conditional is true, step runs
- Requires special syntax for evaluation
 - `${{ <expression> }}`
 - Without syntax, would just be treated as string
 - Can omit special syntax with GitHub and "if"

```
63     test-run:
64
65     runs-on: ubuntu-latest
66     needs: build
67
68     env:
69         TESTING_ACTIVE: ${{ true }}
70
```

```
80     create-issue-on-failure:
81
82     runs-on: ubuntu-latest
83     needs: test-run
84     if: always() && failure()
85     steps:
86         - name: invoke workflow to create issue
```





- Set of functions that can be used as expressions in *if* conditionals
- Used as part of steps
- format is *if: \${{ expression() }}*
- can be simplified to *if: expression()*
- examples:
 - if: success()* - true if none of previous steps have failed or been canceled
 - if: always()* - step always executed and return true - even if canceled
 - if: cancelled()* - true if workflow is cancelled
 - if: failure()* - true if any previous step failed or any job has failed that this job is dependent upon





Lab 6 – Using reusable workflows

Purpose: In this lab, we'll learn some alternative ways to driver workflows.





That's all - thanks!

73

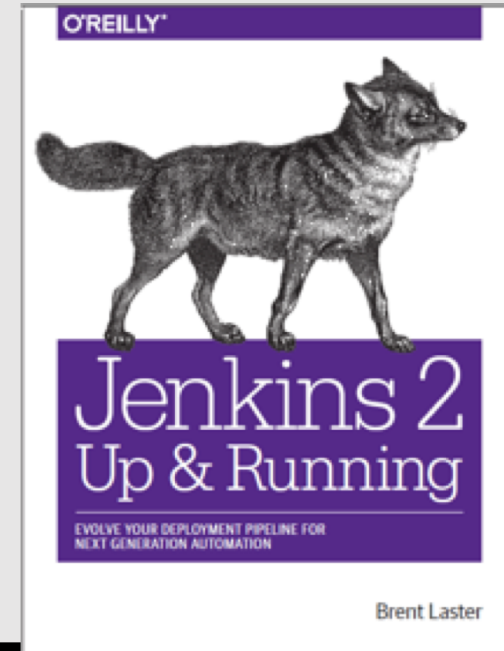
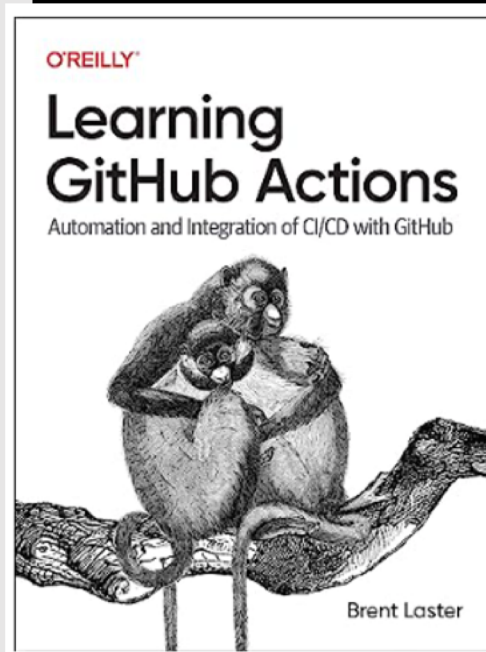
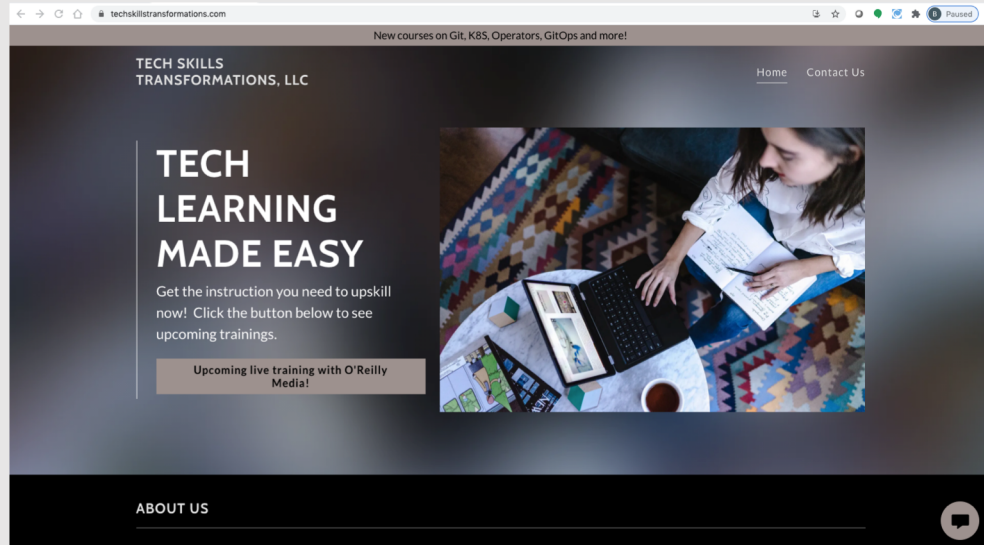
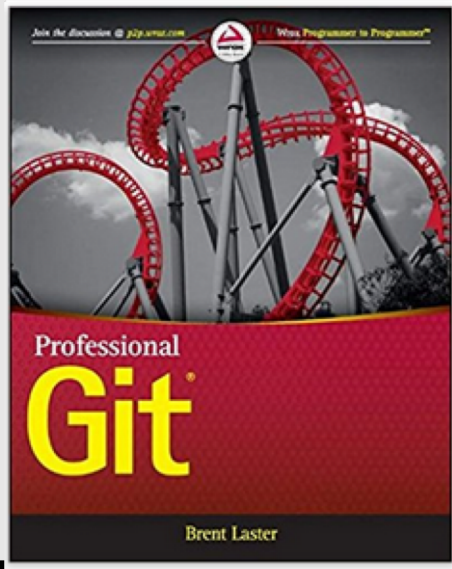
techskillstransformations.com
getskillsnow.com

Professional Git 1st Edition

by Brent Laster (Author)

★★★★★ 7 customer reviews

Look inside ↓



@techupskills

techupskills.com | techskillstransformations.com

© 2023 Brent C. Laster &
Tech Skills Transformations LLC