

> sessions/featured/



Continuous Integration with GitHub Actions

May 7th 2020



@ppremk

Sr. Implementation Engineer



@dechyper

Solutions Architect



@iamhughes

Sr. DevOps Engineer

Agenda

- Overview
- Core Concepts of GitHub Actions
- Work Session 1
- Work Session 2
- Work Session 3

The screenshot shows the GitHub Actions landing page. At the top, there's a header with the GitHub logo and a link to <https://github.com/features/actions>. Below the header, there are tabs for **Repositories**, **Packages**, **People**, and **Projects**. A prominent feature is the **Pinned repositories** section, which includes cards for **starter-workflows**, **toolkit**, **setup-node**, **javascript-action**, **typescript-action**, and **labeler**. Each card provides a brief description, the number of stars and forks, and the language used. Below the pinned repositories, there's a search bar with placeholder text "Find a repository...", and dropdown menus for "Type: All" and "Language: All". On the right side, there's a sidebar titled "Top languages" showing TypeScript, JavaScript, PowerShell, and Dockerfile, each accompanied by a small colored icon. Another sidebar titled "Most used topics" lists actions, github, github-actions, and release.

GitHub Actions
Automate your GitHub workflows
<https://github.com/features/actions> Verified

Repositories **Packages** **People** **Projects**

Pinned repositories

starter-workflows Accelerating new GitHub Actions workflows
2.5k stars, 1.8k forks
TypeScript

toolkit The GitHub ToolKit for developing GitHub Actions.
1.7k stars, 622 forks
TypeScript

setup-node Set up your GitHub Actions workflow with a specific version of node.js
449 stars, 218 forks
TypeScript

javascript-action Create a JavaScript Action with tests, linting, workflow, publishing, and versioning
278 stars, 91 forks
JavaScript

typescript-action Create a TypeScript Action with tests, linting, workflow, publishing, and versioning
335 stars, 82 forks
TypeScript

labeler An action for automatically labelling
271 stars, 100 forks
TypeScript

Find a repository... Type: All Language: All

setup-node
Set up your GitHub Actions workflow with a specific version of node.js
449 stars, 218 forks, 39 issues, 18 releases, Updated 5 minutes ago

toolkit
The GitHub ToolKit for developing GitHub Actions.
1,657 stars, 622 forks, 41 issues, 14 releases, Updated 3 hours ago

Top languages

- TypeScript
- JavaScript
- PowerShell
- Dockerfile

Most used topics

- actions
- github
- github-actions
- release

Overview

- Scope of the Workshop
- Continuous Integration
- GitHub Actions + Continuous Integration

- Level 200 ~ 300
- Using GitHub Actions
- Workflows vs GitHub Actions
- Hands on Learning Lab Course on the topic
- Precursor to the Continuous Delivery Workshop

- Martin Fowler

GitHub Facts

40M+

Developers

100M+

Private and public
repositories

1000s

Open Source
Communities

1B+

Contributions per
year

2M+

Organizations

50%

Fortune 500
companies

YAHOO!

 CISCO

 Walmart

 COMCAST



 British Gas

 SAP

 salesforce

 NAVER

 HSBC

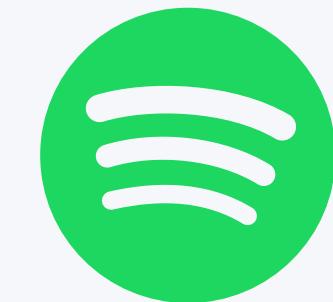
 Adobe

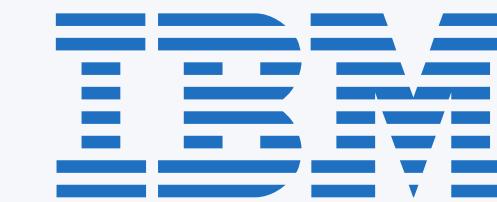
 WELLS FARGO

 hp

 Apple

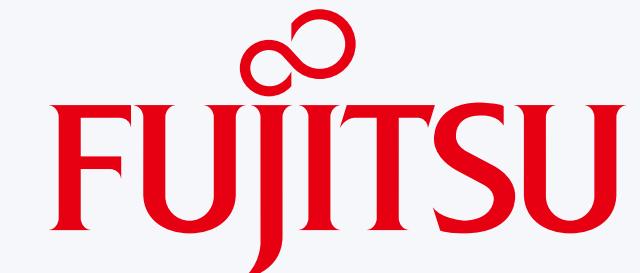
 Capital One



 IBM

 PayPal



 FUJITSU

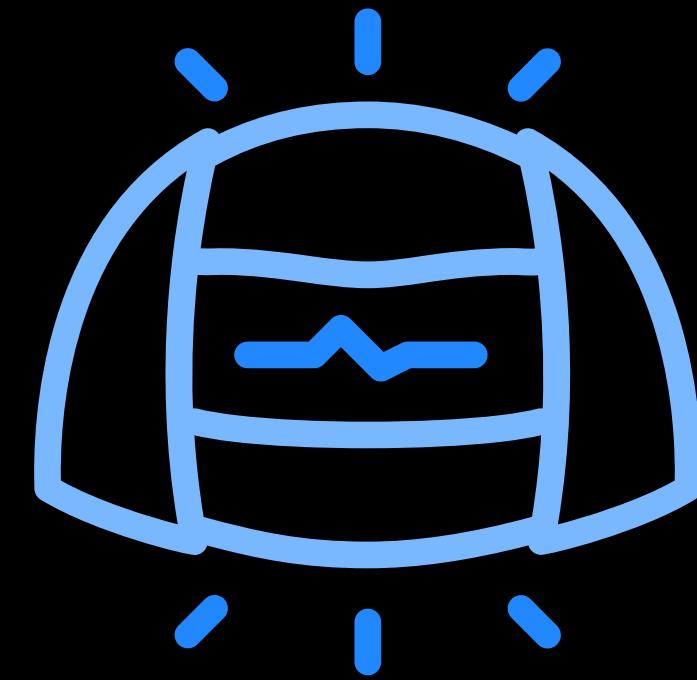
Success Drivers



Common
Philosophy



Asynchronous
Communication

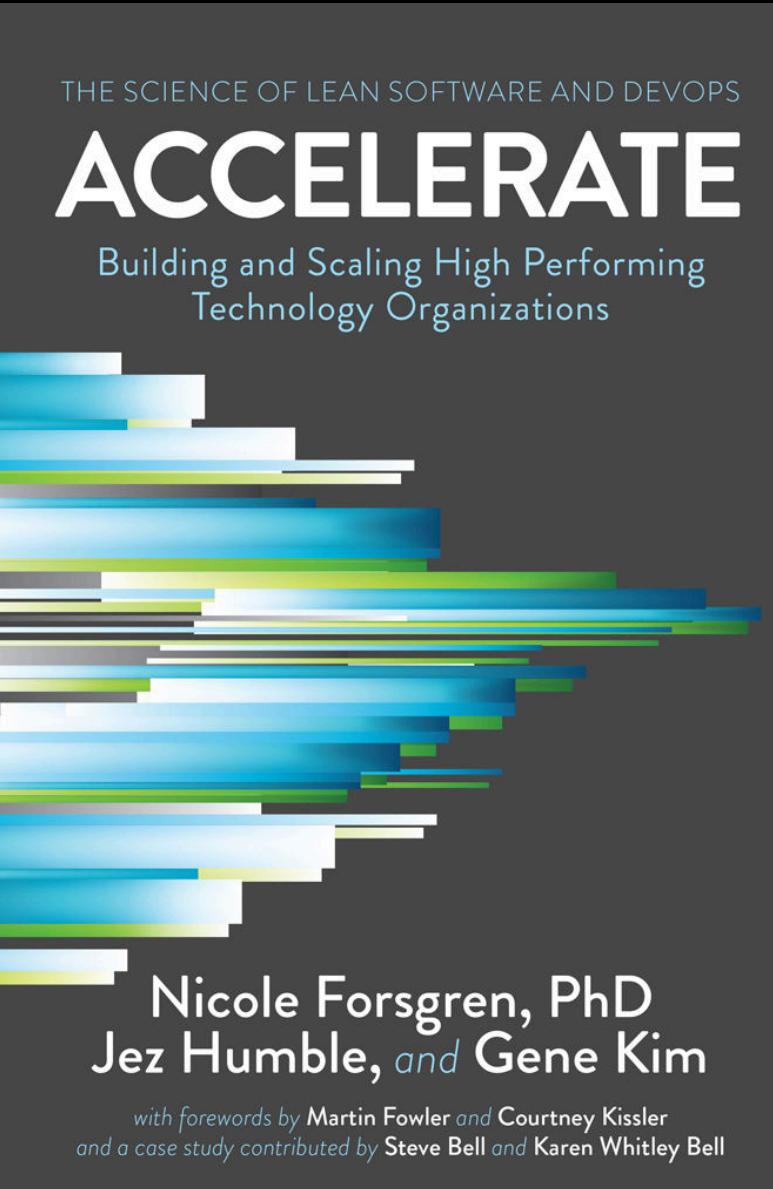


ChatOps



Continuous
Delivery

```
$ curl https://api.github.com/octocat
```



Drivers of Continuous Delivery

- Version Control
- Deployment Automation
- Continuous Integration
- Trunk-Based Development
- Test Automation
- Test Data Management
- Shift Left on Security
- Loosely Coupled Architecture
- Empowered Teams
- Monitoring
- Proactive Notification



Continuous Integration with GitHub Actions

Core Concept of GitHub Actions

- Configuration as Code
- GitHub Actions in a Nutshell
- Actions vs Workflow
- Actions Definition and Explanation

Configuration as Code for (Infrastructure) Continuous Integration



GitHub Actions (Nut Shell)

The image displays three side-by-side screenshots of the GitHub Actions interface for the repository `github/training-kit`.

Screenshot 1 (Left): Repository Overview

This screenshot shows the main repository page for `github/training-kit`. It includes:

- A search bar at the top.
- Links for `Code`, `Issues 4`, `Pull requests 4`, and `Actions`.
- Information about 3,082 commits, 19 branches, and 0 pull requests.
- A branch dropdown set to `actions-for-ci`.
- A link to view pull request #698.
- A message stating "This branch is 4 commits ahead, 10 commits behind master".
- A list of recent commits by `hectorsector` and others, including changes to `.github`, `_includes`, `_layouts`, `_pages`, `assets`, and `downloads`.

Screenshot 2 (Middle): Code Editor

This screenshot shows the code editor for the file `training-kit/.github/workflows/main.yml` in the `actions-for-ci` branch. The code defines a workflow named "CI for GitHub Pages site" that runs on push events. The workflow has a single job named "build" that uses the `actions/checkout@v1` action and runs on an Ubuntu-Latest container. The code is as follows:

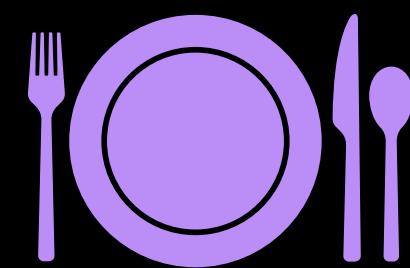
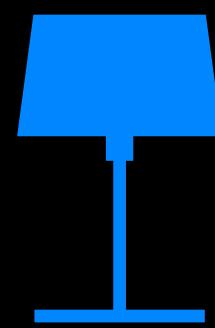
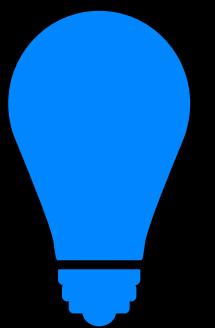
```
name: CI for GitHub Pages site
on: [push]
jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v1
      # Ruby-specific steps
      - name: Set up Ruby 2.6
        run: |
          curl -s https://www.ruby-lang.org/gems/ruby-2.6.3.tar.gz | tar xz
          cd ruby-2.6.3
          ./configure
          make
          make install
```

Screenshot 3 (Right): Build Log

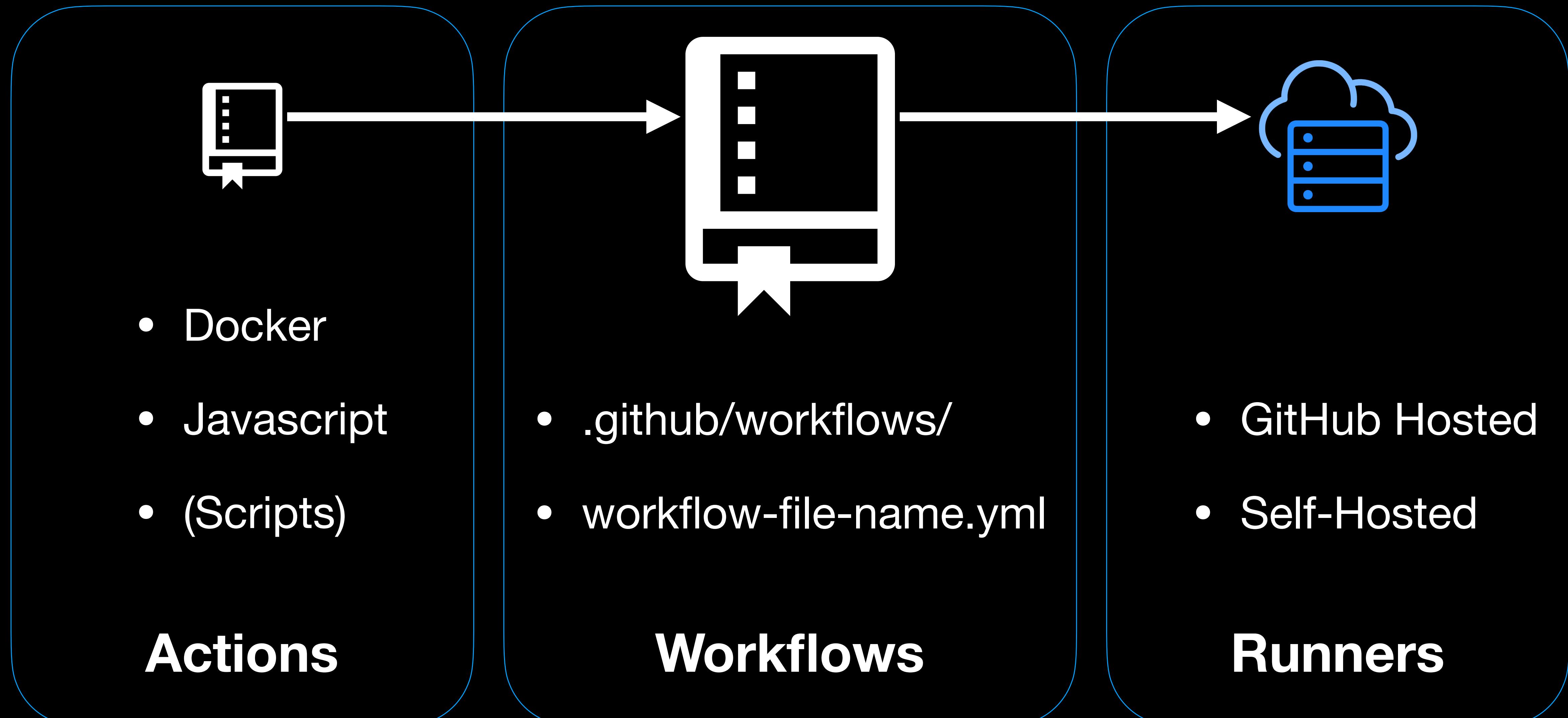
This screenshot shows the build log for the "CI for GitHub Pages site" workflow. The build was triggered by a push event and completed successfully 25 days ago in 4m 9s. The log details the following steps:

- Set up job
- Run actions/checkout@v1
- Set up Ruby 2.6
- Build and test with Rake
- Install node modules
- Build the site in the jekyll.builder container
- Complete job

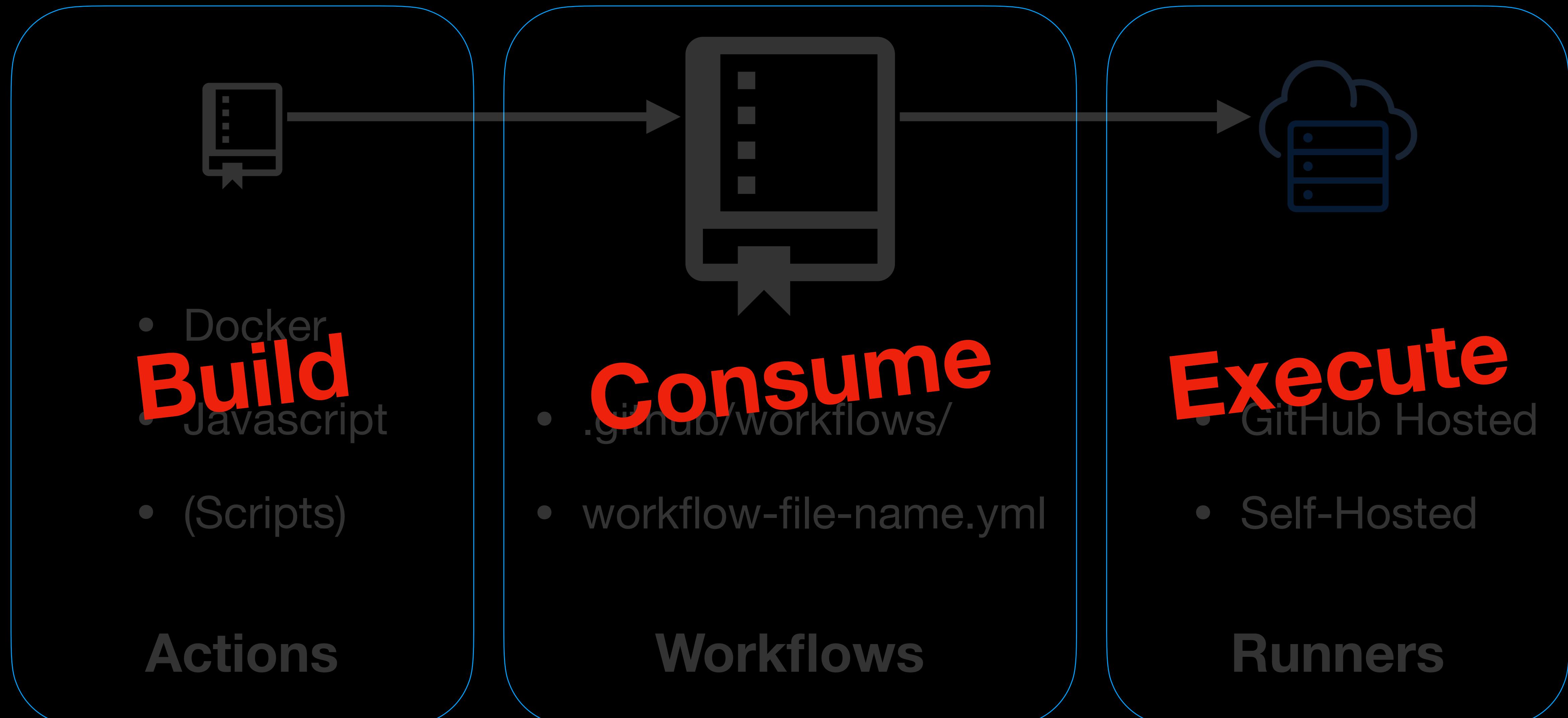
Actions vs Workflows



Lay of the Land



Lay of the Land



Actions definitions and explanation

Workflow file example

```
name: Greet Everyone
# This workflow is triggered on pushes to the repository.
on: [push]

jobs:
  build:
    # Job name is Greeting
    name: Greeting
    # This job runs on Linux
    runs-on: ubuntu-latest
    steps:
      # This step uses GitHub's hello-world-javascript-action: https://github.com/actions/hello-world-javascript-action@v1
      - name: Hello world
        uses: actions/hello-world-javascript-action@v1
        with:
          who-to-greet: 'Mona the Octocat'
          id: hello
      # This step prints an output (time) from the previous step's action.
      - name: Echo the greeting's time
        run: echo 'The time was ${{ steps.hello.outputs.time }}.'
```



The screenshot shows the GitHub Actions interface for the octocat/hello-world-workflow. It highlights several components:

- Workflow status:** Shows the workflow is up-to-date with a green checkmark.
- Last run:** Shows the most recent successful run 16 hours ago.
- Workflow / Check suite:** Shows the workflow name and trigger (push).
- Job / Check run:** Shows the specific job run on Linux.
- Step:** Points to the "Hello world action step" which ran successfully.
- Log view:** Shows the command output, including the greeting and the current time.
- Run time:** Shows the duration of each step.
- Download logs / More options:** Buttons for managing logs and other options.

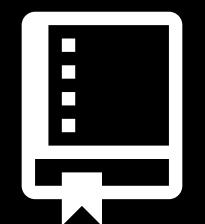
Work Session 1

- Prerequisites
- Installing Learning Lab
- Register for Using GitHub Actions Continuous Integration course
- Required Steps

- The full story
- GitHub Actions Composition
- What's a “Job”

Prerequisites

1. GitHub account with a verified email address
2. Install GitHub Learning Lab on your account



<https://git.io/JfsxF>

Install Learning Lab

1. Go to lab.github.com
2. Click **Sign in with GitHub**
3. **Accept** the terms
4. **Install GitHub Learning Lab**
(we recommend on your account / all repos)

Wait for the next step

The screenshot shows the GitHub Learning Lab landing page. At the top, there's a navigation bar with links for 'Learning Lab', 'For Organizations', and 'Teach on Learning Lab'. Below the header, a large blue button says 'Sign in with GitHub'. To the right of the button is a cartoon GitHub cat with glowing blue eyes. The text 'GitHub Learning Lab' is prominently displayed above a description: 'Learn new skills by completing fun, realistic projects in your very own GitHub repository. Get advice and helpful feedback from our friendly Learning Lab bot.' Below the cat, there's a section titled 'Featured courses' with three course cards visible.

Register for *Using GitHub Actions for CI*

1. Go to <https://git.io/Jewra> or search for “GitHub Actions”
2. On the course landing page, click **Join this course**
3. Click **Start**

The screenshot shows a web browser window with the URL `lab.github.com` in the address bar. The page title is "Using GitHub Actions for CI". The sub-header reads "Set up continuous integration with GitHub Actions". Below the header, there are two "Join this course" buttons: one labeled "Join this course" and another labeled "Join this course" with a "Private" lock icon. A large text box contains the following description: "GitHub Actions makes it easier than ever to incorporate continuous integration (CI) into your repositories. This course will get you from zero-to-CI by setting up two workflow files to automate the way you work." At the bottom, a section titled "After taking this course, you'll be able to:" lists the following bullet points:

- Describe CI and why it is necessary
- Use and customize a templated workflow
- Create CI workflows that match the team's needs and behaviors
- Use the repository's source code and build artifacts (like compiled source code) across jobs in a workflow



Work session 1

Complete only the following steps:

- ✓ Use a templated workflow
- ✓ Run a templated workflow
- ✓ Add your first test
- ✓ Read an Actions log
- ✓ Fix the test
- ✓ Share the workflow with the team

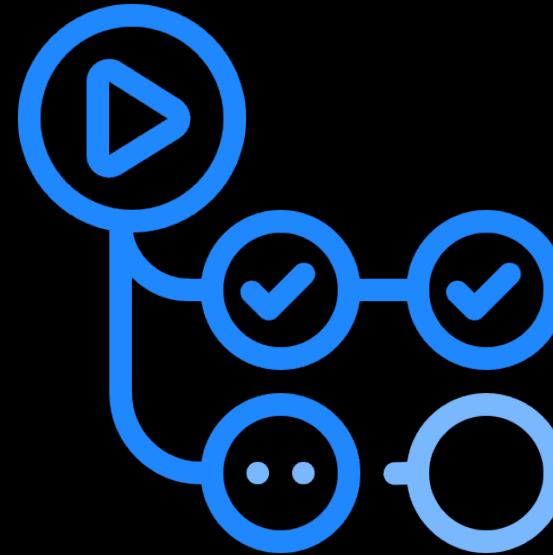
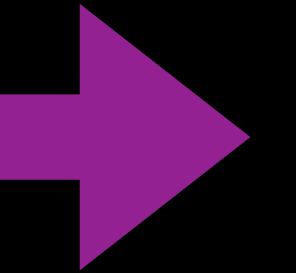


Course Steps

Course steps

- 1 Use a templated workflow
Create a pull request with a templated workflow Start
- 2 Run a templated workflow
Wait for GitHub to run the templated workflow and report back the results
- 3 Add your first test
Add your first test script for CI to pick up
- 4 Read an Actions log
Tell the bot which test is failing so we can fix it
- 5 Fix the test
Edit the file that's causing the test to fail
- 6 Share the workflow with the team
Merge the pull request containing your first workflow so the entire team can use it
- 7 Create a custom GitHub Actions workflow
Edit the existing workflow with new build targets

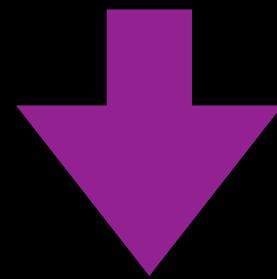
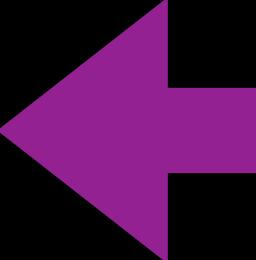
The complete story



Some event happens on GitHub

Examples: push, open a PR, comment on an issue

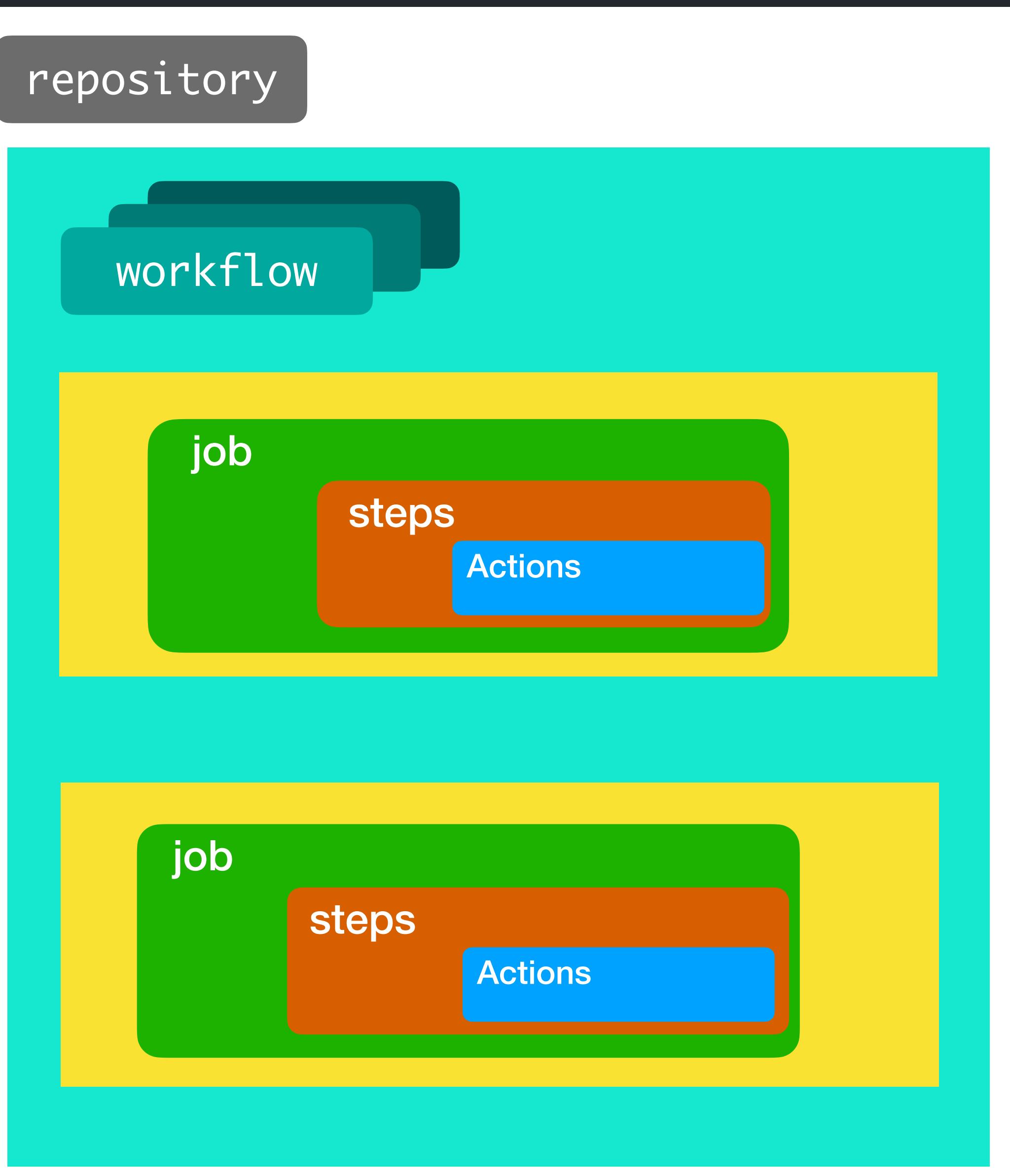
The event triggers a workflow



Jobs in a workflow run in a virtual environment

The workflow utilizes actions or scripts

GitHub Actions Composition



What's a “job”?

repository

workflow

job

Virtual Environment

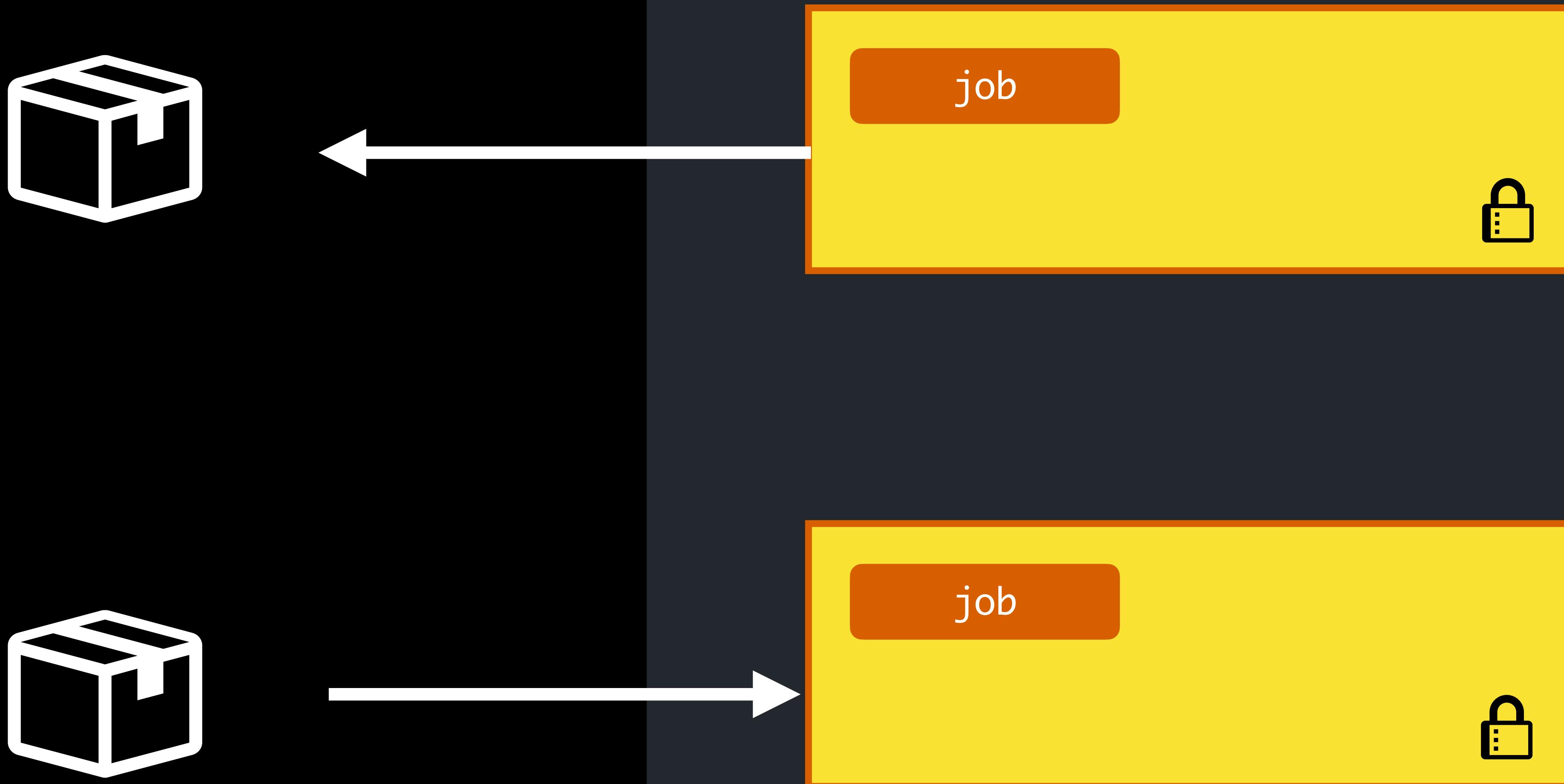


Virtual Environment

job



Sharing artefacts between “job”



Work Session 2

- Required Steps
- Multiple Workflows



Work session 2

Complete only the following steps:

- ✓ Create a custom GitHub Actions workflow
- ✓ Target a Windows environment
- ✓ Use multiple jobs
- ✓ Run multiple jobs
- ✓ Upload a job's build artifacts
- ✓ Download a job's build artifacts
- ✓ Share the improved CI workflow with the team



Course Steps

- 7 Create a custom GitHub Actions workflow
Edit the existing workflow with new build targets
- 8 Target a Windows environment
Edit your workflow file to build for Windows environments
- 9 Use multiple jobs
Edit your workflow file to separate build and test jobs
- 10 Run multiple jobs
Wait for the result of multiple jobs in your workflow
- 11 Upload a job's build artifacts
Use the upload action in your workflow file to save a job's build artifacts
- 12 Download a job's build artifacts
Use the download action in your workflow file to access a prior job's build artifacts
- 13 Share the improved CI workflow with the team
Merge the pull request with the improved workflow
- 14 Automate the review process
Add a new workflow file to automate the team's review process

Beyond CI for code defects

```
1 # This workflow will do a clean install of node dependencies, build the source code and run tests across different versions of node
2 # For more information see: https://help.github.com/actions/language-and-framework-guides/using-nodejs-with-github-actions
3
4 name: Node CI
5
6 on: [push]
7
8 jobs:
9   - uses: actions/upload-artifact@master
10  needs: build
11    name: webpack artifacts
12    path: public/
13
14 runs-on: ubuntu-latest
15
16 steps:
17   - uses: actions/checkout@v1
18   - name: npm install and build webpack
19     run: |
20       npm install
21       npm run build
22
23 test:
24   - uses: actions/download-artifact@master
25   with:
26     name: webpack artifacts
27     path: public
28   - name: npm install, and test
29   runs-on: ubuntu-latest
30
31 strategy:
32   matrix:
33     os: [ubuntu-latest, windows-2016]
34     node-version: [8.x, 10.x]
35
36 steps:
37   - uses: actions/checkout@v1
38   - name: Use Node.js ${matrix.node-version}
39     uses: actions/setup-node@v1
40     with:
41       node-version: ${matrix.node-version}
42   - name: npm install, and test
43     run: |
44       npm install
45       npm test
46   env:
47     CI: true
```

Job

Build Matrix

Multiple workflows

Work Session 1 + 2

Work Session 3

Workshop 2

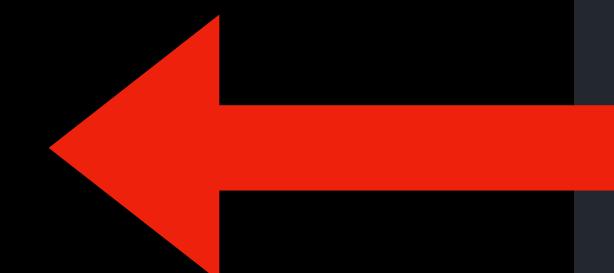
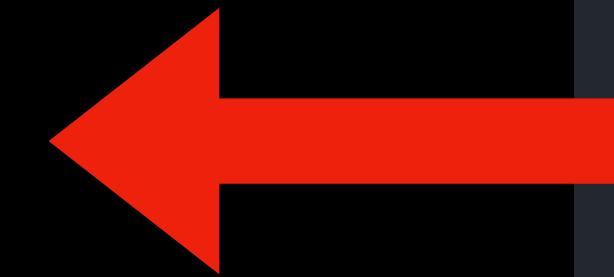
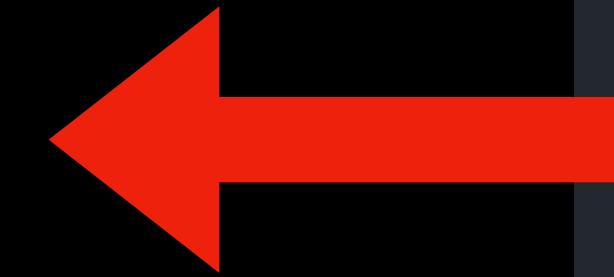
CD with Actions

repository

ci workflow

approval workflow

deployment workflow



Work Session 3

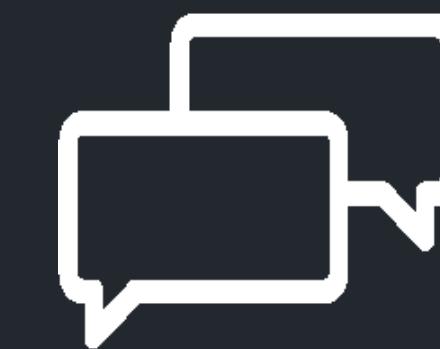
- Required Steps



Work session 3

Complete only the following steps:

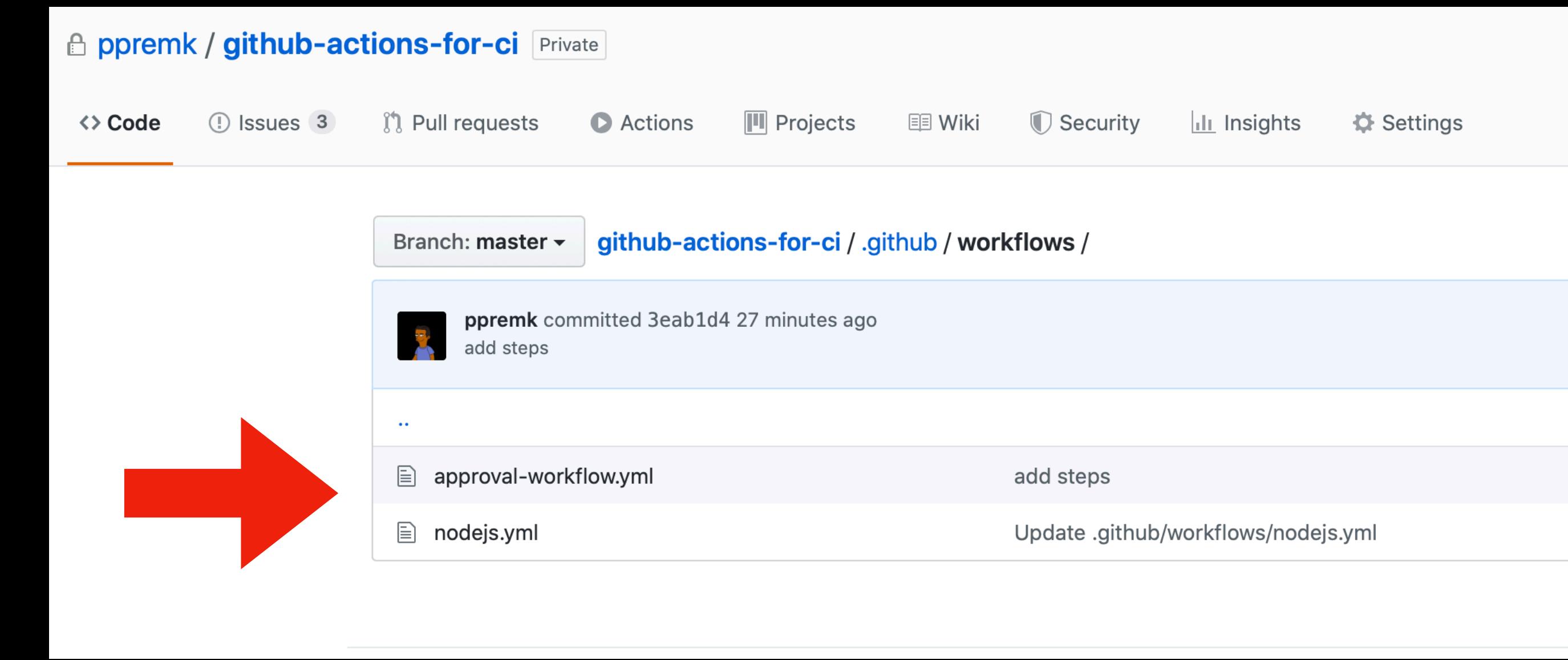
- ✓ Automate the review process
- ✓ Use an action to automate pull request reviews
- ✓ Create an approval job in your new workflow
- ✓ Automate approvals
- ✓ Use branch protections



Course Steps

- 14 Automate the review process
Add a new workflow file to automate the team's review process
- 15 Use an action to automate pull request reviews
Use the community action in your new workflow
- 16 Create an approval job in your new workflow
In your new workflow file, create a new job that'll use the community action
- 17 Automate approvals
Use the community action to automate part of the review approval process
- 18 Use branch protections
Complete the automated review process by protecting the master branch

A more complete pipeline



```
13 lines (13 sloc) | 376 Bytes
Raw Blame History
1 name: Team awesome's approval workflow
2 on: pull_request_review
3 jobs:
4   labelWhenApproved:
5     runs-on: ubuntu-latest
6     steps:
7       - name: Label when approved
8         uses: pullreminders/label-when-approved-action@master
9         env:
10           APPROVALS: "2"
11           GITHUB_TOKEN: ${{ secrets.GITHUB_TOKEN }}
12           ADD_LABEL: "approved"
13           REMOVE_LABEL: "awaiting%20review"
```

Bonus

- Migrating to GitHub Actions

```
pipeline {
    agent none
    stages {
        stage('Run Tests') {
            parallel {
                stage('Test On MacOS') {
                    agent { label "macos" }
                    tools { nodejs "node-12" }
                    steps {
                        dir("scripts/myapp") {
                            sh(script: "npm install -g bats")
                            sh(script: "bats tests")
                        }
                    }
                }
                stage('Test On Linux') {
                    agent { label "linux" }
                    tools { nodejs "node-12" }
                    steps {
                        dir("script/myapp") {
                            sh(script: "npm install -g bats")
                            sh(script: "bats tests")
                        }
                    }
                }
            }
        }
    }
}
```

Jenkins Pipeline

```
name: demo-workflow
on:
  push:
jobs:
  test:
    runs-on: ${matrix.os}
  strategy:
    fail-fast: false
  matrix:
    os: [macos-latest, ubuntu-latest]
steps:
  - uses: actions/checkout@v1
  - uses: actions/setup-node@v1
    with:
      node-version: 12
  - run: npm install -g bats
  - run: bats tests
  working-directory: scripts/myapp
```

GitHub Actions Workflow

Why Actions

- 50 Million developers building Actions for your CI (CI as Code)
- Open and component based design for CI
- A single platform and a simplified way of working
- OS agnostic CI
- GitHub security in your CI
- Use GitHub Open API to drive Actions

Resources

- <https://help.github.com/en/actions>
- <https://help.github.com/en/actions/configuring-and-managing-workflows>
- <https://help.github.com/en/actions/language-and-framework-guides>
- <https://help.github.com/en/actions/migrating-to-github-actions>
- <https://help.github.com/en/actions/hosting-your-own-runners>
- <https://github.com/actions>
- <https://github.com/actionsdesk>

