

This version of GitHub Enterprise was discontinued on 2023-03-15. No patch releases will be made, even for critical security issues. For better performance, improved security, and new features, [upgrade to the latest version of GitHub Enterprise](#). For help with the upgrade, [contact GitHub Enterprise support](#).

Using scripts to test your code on a runner

In this article

- Example overview
- Features used in this example
- Example workflow
- Understanding the example
- Next steps

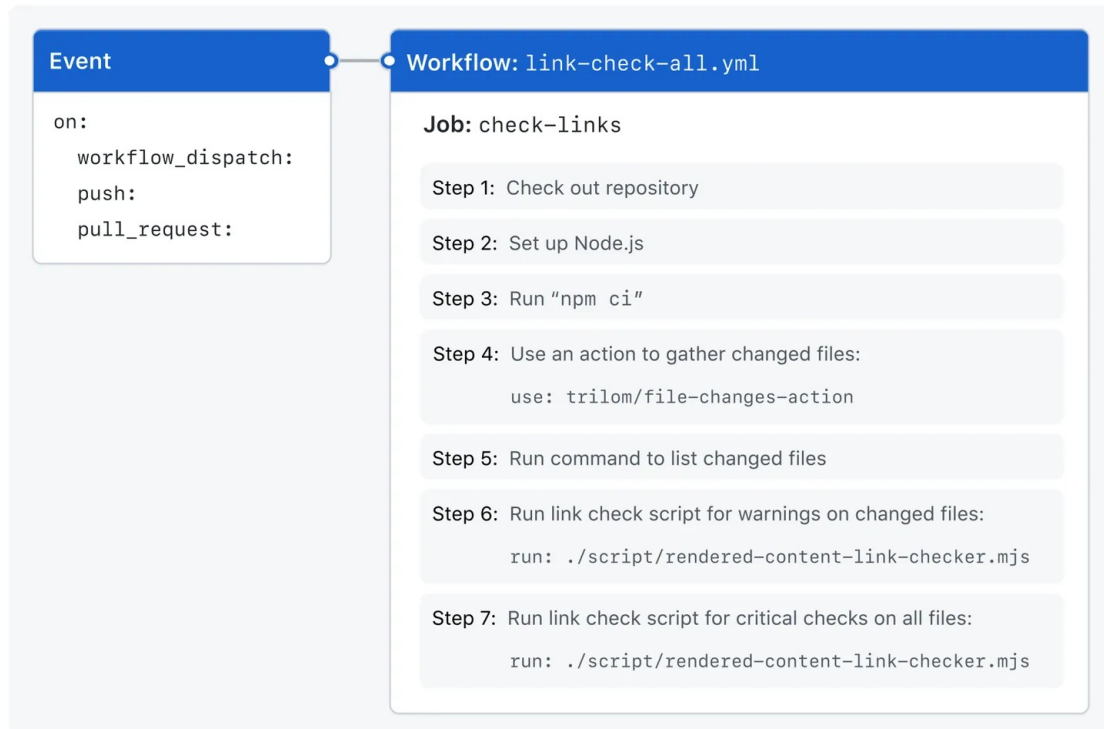
How to use essential GitHub Actions features for continuous integration (CI).

Note: GitHub-hosted runners are not currently supported on GitHub Enterprise Server. You can see more information about planned future support on the [GitHub public roadmap](#).

Example overview

This article uses an example workflow to demonstrate some of the main CI features of GitHub Actions. When this workflow is triggered, it automatically runs a script that checks whether the GitHub Docs site has any broken links.

The following diagram shows a high level view of the workflow's steps and how they run within the job:



Features used in this example [↗](#)

The example workflow demonstrates the following capabilities of GitHub Actions.

Feature	Implementation
Triggering a workflow to run automatically	push
Triggering a workflow to run automatically	pull_request
Manually running a workflow from the UI	workflow_dispatch
Setting permissions for the token	permissions
Controlling how many workflow runs or jobs can run at the same time	concurrency
Running the job on different runners, depending on the repository	runs-on
Cloning your repository to the runner	actions/checkout
Installing <code>node</code> on the runner	actions/setup-node
Using a third-party action	trilom/file-changes-action
Running a script on the runner	Using <code>./script/rendered-content-link-checker.mjs</code>

Example workflow [↗](#)

The following workflow was created by the GitHub Docs Engineering team. To review the latest version of this file in the [github/docs](#) repository, see [check-broken-links-github-github.yml](#).

Note: Each line of this workflow is explained in the next section at "[Understanding the example](#)."



```
name: 'Link Checker: All English'

# **What it does**: Renders the content of every page and check all internal links.
# **Why we have it**: To make sure all links connect correctly.
# **Who does it impact**: Docs content.

on:
  workflow_dispatch:
  push:
    branches:
      - main
  pull_request:

permissions:
  contents: read
  # Needed for the 'trilom/file-changes-action' action
  pull-requests: read

# This allows a subsequently queued workflow run to interrupt previous runs
concurrency:
  group: '${{ github.workflow }} @ ${{ github.event.pull_request.head.label || github
cancel-in-progress: true

jobs:
  check-links:
    runs-on: ${{ fromJSON(['ubuntu-latest', 'self-hosted'])[github.repository ==
    steps:
      - name: Checkout
        uses: actions/checkout@v2

      - name: Setup node
        uses: actions/setup-node@v2
        with:
          node-version: 16.13.x
          cache: npm

      - name: Install
        run: npm ci

      # Creates file "$/files.json", among others
      - name: Gather files changed
        uses: trilom/file-changes-action@a6ca26c14274c33b15e6499323aac178af06ad4b
        with:
          fileOutput: 'json'

      # For verification
      - name: Show files changed
        run: cat $HOME/files.json

      - name: Link check (warnings, changed files)
        run: |
          ./script/rendered-content-link-checker.mjs \
            --language en \
            --max 100 \
            --check-anchors \
            --check-images \
            --verbose \
            --list $HOME/files.json

      - name: Link check (critical, all files)
        run: |
          ./script/rendered-content-link-checker.mjs \
            --language en \
            --exit \
            --verbose \
            --check-images \
            --level critical
```

Understanding the example

The following table explains how each of these features are used when creating a GitHub Actions workflow.

Code	Explanation
<div><div>YAML</div><div><pre>name: 'Link Checker: All English'</pre></div></div>	The name of the workflow as it will appear in the "Actions" tab of the GitHub repository.
<div><div>YAML</div><div><pre>on:</pre></div></div>	The <code>on</code> keyword lets you define the events that trigger when the workflow is run. You can define multiple events here. For more information, see "Triggering a workflow."
<div><div>YAML</div><div><pre>workflow_dispatch:</pre></div></div>	Add the <code>workflow_dispatch</code> event if you want to be able to manually run this workflow from the UI. For more information, see workflow_dispatch .
<div><div>YAML</div><div><pre>push: branches: - main</pre></div></div>	Add the <code>push</code> event, so that the workflow runs automatically every time a commit is pushed to a branch called <code>main</code> . For more information, see push .
<div><div>YAML</div><div><pre>pull_request:</pre></div></div>	Add the <code>pull_request</code> event, so that the workflow runs automatically every time a pull request is created or updated. For more information, see pull_request .
<div><div>YAML</div><div><pre>permissions: contents: read pull-requests: read</pre></div></div>	Modifies the default permissions granted to <code>GITHUB_TOKEN</code> . This will vary depending on the needs of your workflow. For more information, see "Assigning permissions to jobs."
<div><div>YAML</div><div><pre>concurrency: group: '\${{ github.workflow }} @ \${{ github.event.pull_request.head</pre></div></div>	Creates a concurrency group for specific events, and uses the <code> </code> operator to define fallback values. For more information, see "Using concurrency."

YAML



```
cancel-in-progress: true
```

Cancels any currently running job or workflow in the same concurrency group.

YAML



```
jobs:
```

Groups together all the jobs that run in the workflow file.

YAML



```
check-links:
```

Defines a job with the ID `check-links` that is stored within the `jobs` key.

YAML



```
runs-on: ${ fromJSON('["ubuntu-latest", "self-hosted"]')[github
```

Configures the job to run on a GitHub-hosted runner or a self-hosted runner, depending on the repository running the workflow. In this example, the job will run on a self-hosted runner if the repository is named `docs-internal` and is within the `github` organization. If the repository doesn't match this path, then it will run on an `ubuntu-latest` runner hosted by GitHub. For more information on these options see "[Choosing the runner for a job](#)."

YAML



```
steps:
```

Groups together all the steps that will run as part of the `check-links` job. Each job in a workflow has its own `steps` section.

YAML



```
- name: Checkout
  uses: actions/checkout@v2
```

The `uses` keyword tells the job to retrieve the action named `actions/checkout`. This is an action that checks out your repository and downloads it to the runner, allowing you to run actions against your code (such as testing tools). You must use the checkout action any time your workflow will run against the repository's code or you are using an action defined in the repository.

YAML



```
- name: Setup node
  uses: actions/setup-node@v2
  with:
    node-version: 16.13.x
    cache: npm
```

This step uses the `actions/setup-node` action to install the specified version of the Node.js software package on the runner, which gives you access to the `npm` command.

YAML



```
- name: Install
  run: npm ci
```

The `run` keyword tells the job to execute a command on the runner. In this case, `npm ci` is used to install the npm software packages for the project.

YAML



```
- name: Gather files changed
  uses: trilom/file-changes-action@a6ca26c14274c33b15e6499323a
  with:
    fileOutput: 'json'
```

Uses the `trilom/file-changes-action` action to gather all the changed files. This example is pinned to a specific version of the action, using the `a6ca26c14274c33b15e6499323a` SHA.

YAML



```
- name: Show files changed
  run: cat $HOME/files.json
```

Lists the contents of `files.json`. This will be visible in the workflow run's log, and can be useful for debugging.

YAML



```
- name: Link check (warnings, changed files)
  run: |
    ./script/rendered-content-link-checker.mjs \
      --language en \
      --max 100 \
      --check-anchors \
      --check-images \
      --verbose \
      --list $HOME/files.json
```

This step uses `run` command to execute a script that is stored in the repository at `script/rendered-content-link-checker.mjs` and passes all the parameters it needs to run.

YAML



```
- name: Link check (critical, all files)
  run: |
    ./script/rendered-content-link-checker.mjs \
      --language en \
      --exit \
      --verbose \
      --check-images \
      --level critical
```

This step also uses `run` command to execute a script that is stored in the repository at `script/rendered-content-link-checker.mjs` and passes a different set of parameters.

Next steps

- To learn about GitHub Actions concepts, see "[Understanding GitHub Actions](#)."
- For more step-by-step guide for creating a basic workflow, see "[Quickstart for GitHub Actions](#)."
- If you're comfortable with the basics of GitHub Actions, you can learn about workflows and their features at "[About workflows](#)."

Legal