Cloud Expert

Cloud Deployments: Github Actions

DE HOGESCHOOL
MET HET NETWERK

Elfde-Liniestraat 24, 3500 Hasselt, www.pxl.be

Pipelines in Github Actions

Recap
Github Actions Components
Workflows: Basics
Jobs
Marketplace
Triggers
Example

Wat is een Pipeline?

Definitie “deployment pipeline”:

(first defined by Jez Humble and David Farley in their book *Continuous Delivery: Reliable Software Releases Through Build, Test, and Deployment Automation*)

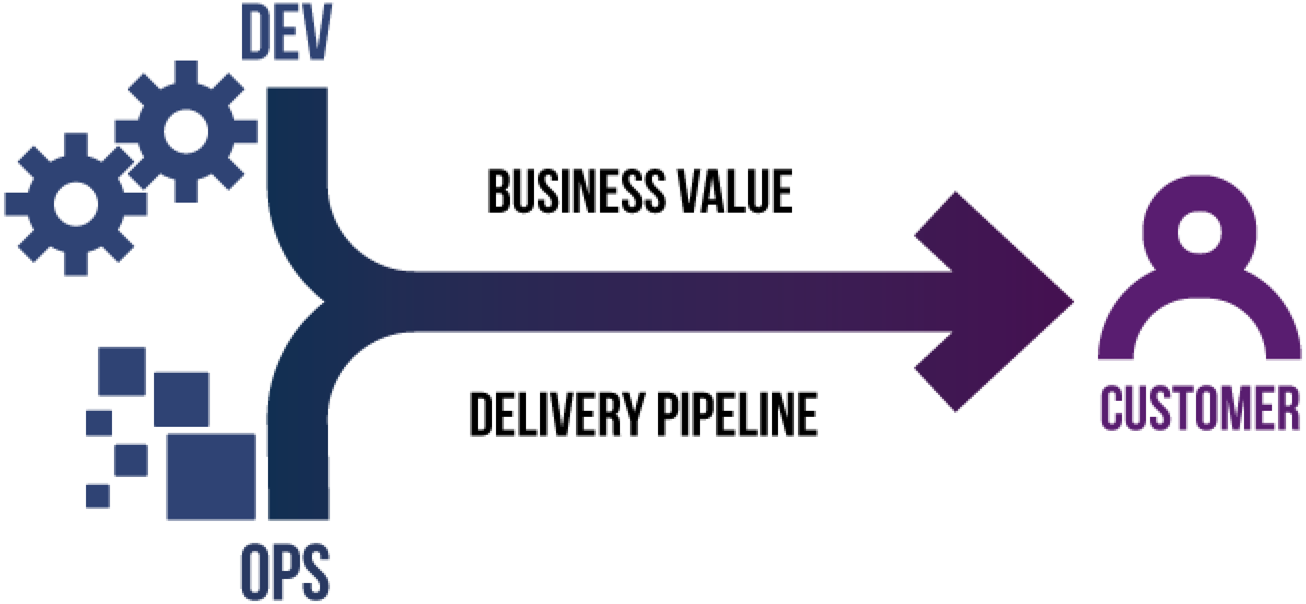
It ensures that all code checked in to version control is automatically built and tested in a production-like environment.

Sleutelwoorden:

- Alle Code
- Versiebeheer
- **Automatisch gebouwd**
- Automatisch getest
- Productie-waardige omgeving

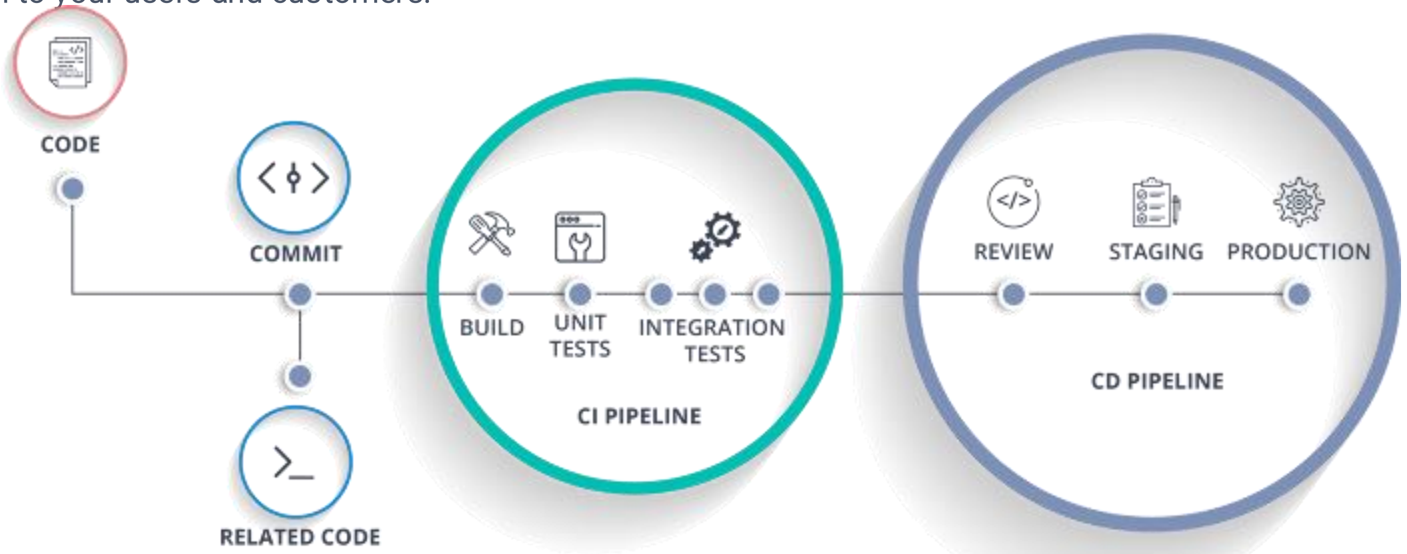
THE FIRST WAY

CREATE A DELIVERY SYSTEM



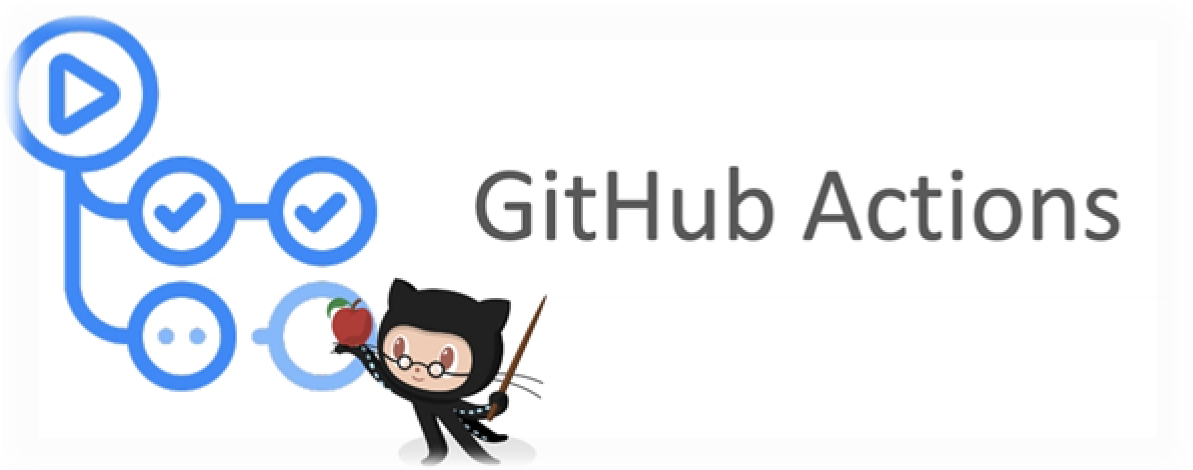
Recap: CI/CD pipeline

A ***continuous delivery (CD) pipeline*** is an automated expression of your process for getting software from version control right through to your users and customers.



Every change to your software (committed in source control) goes through a complex process on its way to being released. This process involves building the software in a reliable and repeatable manner, as well as progressing the built software (called a "build") through multiple stages of testing and deployment.

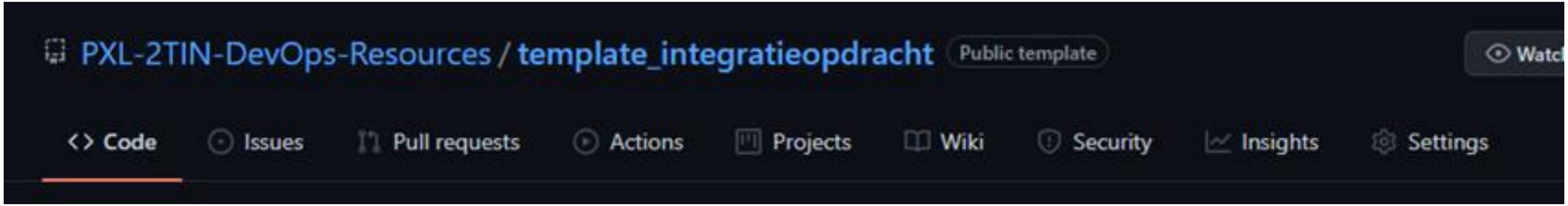
Github Actions: Overview



https://docs.github.com/en/actions

Github ecosysteem

- Veel meer dan enkel code repositories
  - Issue tracking
  - Kanban & projectomvolging
  - Wiki & documentatie
  - Releases
  - Security scans / dependency scans
  - CI/CD
  - ….



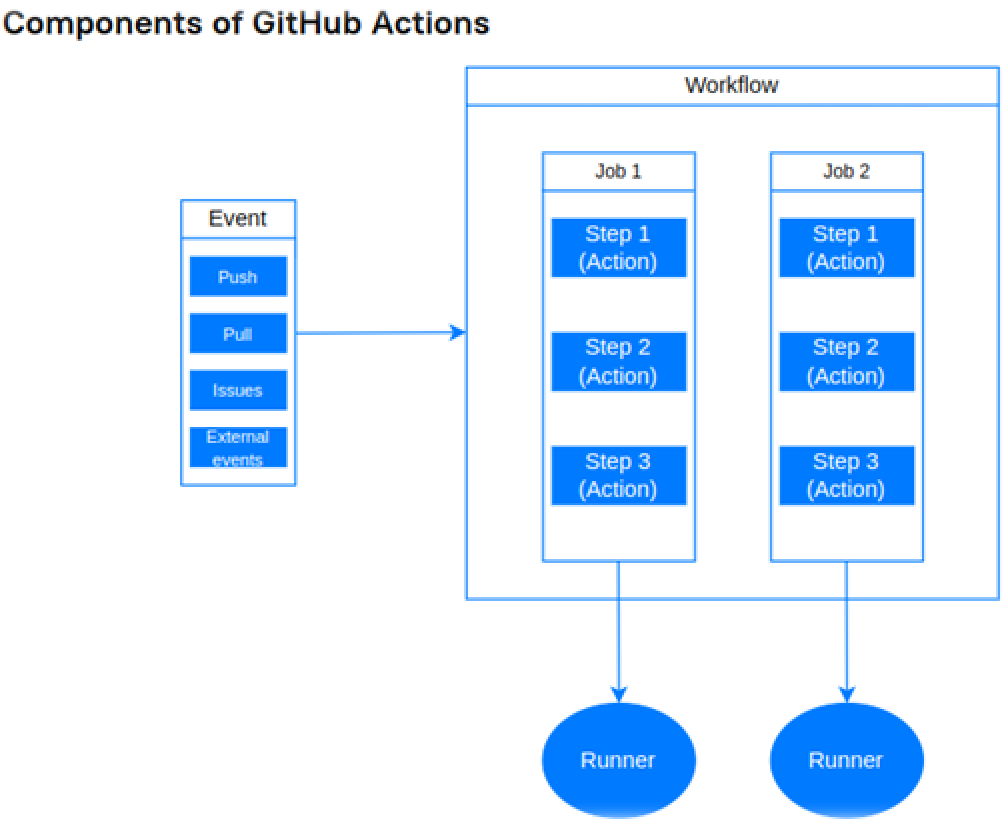
Github Actions: Workflows

Een workflow is een configureerbaar geautomatiseerd proces at déén of meerdere **jobs** gaat uitvoeren

- Gedefinieerd door een YAML bestand in de repository
- Kan uitgevoerd worden door triggers / events / manueel in de reppository.
- Staat in de .github/workflows folder

https://docs.github.com/en/actions/writing-workflows/quickstart

Github Actions: Components

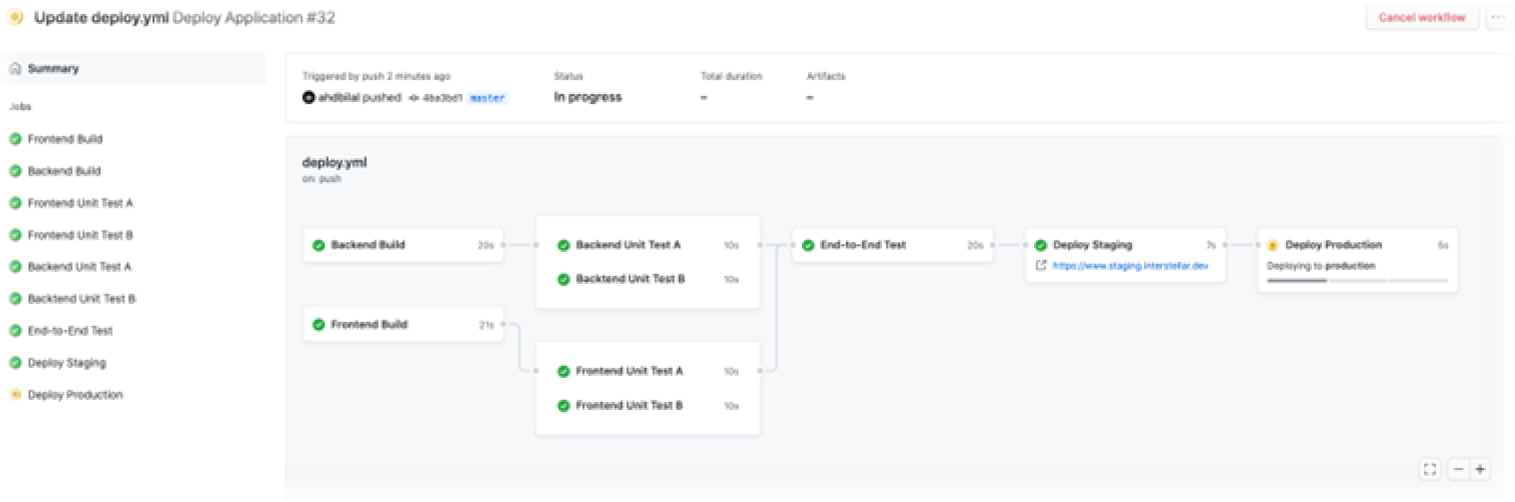


https://docs.github.com/en/actions/about-github-actions/understanding-github-actions

Github Actions: Workflows

Een repository kan meerdere workflows hebben, elk met hun eigen set van taken zoals bv:

- Builden & testen van pull requests
- Deployment van de applicatie bij een nieuwe release



Update deploy.yml Deploy Application #32

Summary

Jobs

- Frontend Build
- Backend Build
- Frontend Unit Test A
- Frontend Unit Test B
- Backend Unit Test A
- Backtend Unit Test B
- End-to-End Test
- Deploy Staging
- Deploy Production

Triggered by push 2 minutes ago: ahdbilal pushed -o- 4ba3bd1 master

Status: In progress

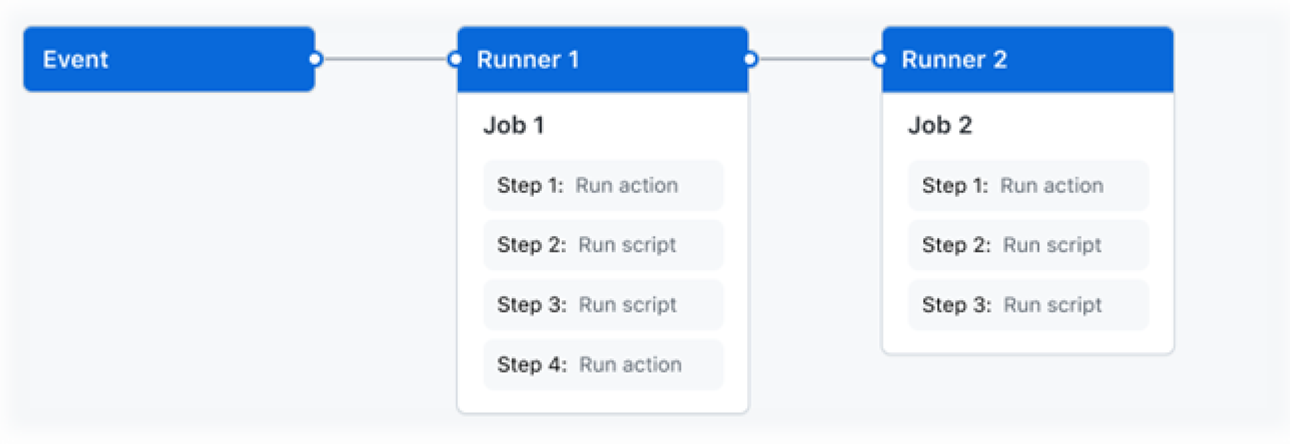
Total duration: –

Artifacts: –

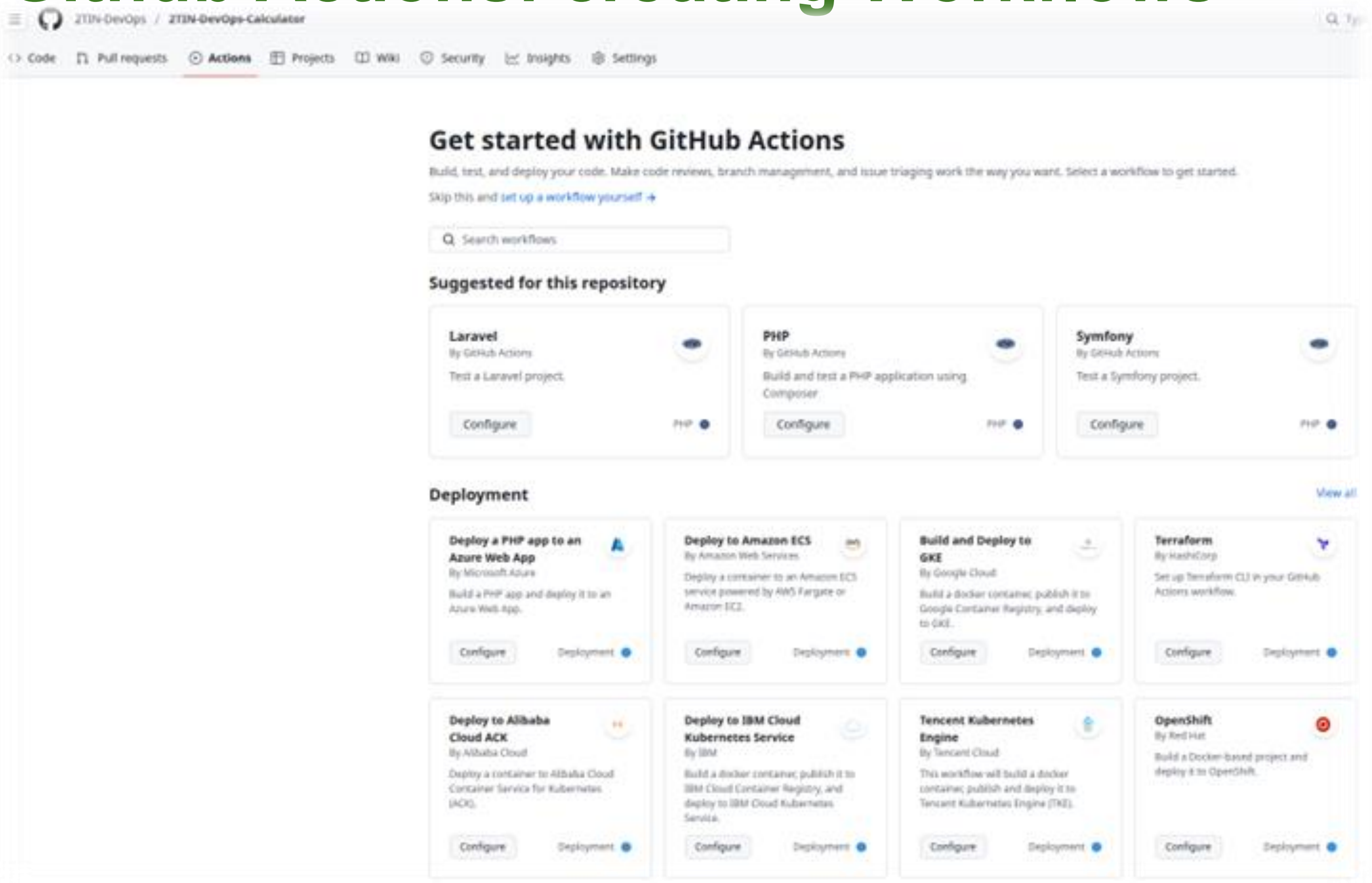
deploy.yml
on: push

Github Actions: Workflow Basics

- Eén of meerdere events die de workflow starten
- Eén of meerdere jobs die elks op een runner worden uitgevoerd met één of meerdere steps
- Elke step kan een script uitvoeren of gebruik maken van een extensie je helpt bij het uitvoeren van een set van instructies (Zie marketplace later)



Github Actions: creating Workflows



Get started with GitHub Actions

Build, test, and deploy your code. Make code reviews, branch management, and issue triaging work the way you want. Select a workflow to get started.

Skip this and set up a workflow yourself →

Search workflows

Suggested for this repository

Laravel
By GitHub Actions
Test a Laravel project.
Configure — PHP ●

PHP
By GitHub Actions
Build and test a PHP application using Composer
Configure — PHP ●

Symfony
By GitHub Actions
Test a Symfony project.
Configure — PHP ●

Deployment

View all

Deploy a PHP app to an Azure Web App
By Microsoft Azure
Build a PHP app 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 ●

Github Actions: creating Workflows

Een workflow maken:

- Gebruik de wizard in de "actions" tab (zoals te zien in de vorige slide)
- Doe het manueel: Maak een nieuwe yaml file in de .github/workflows folder

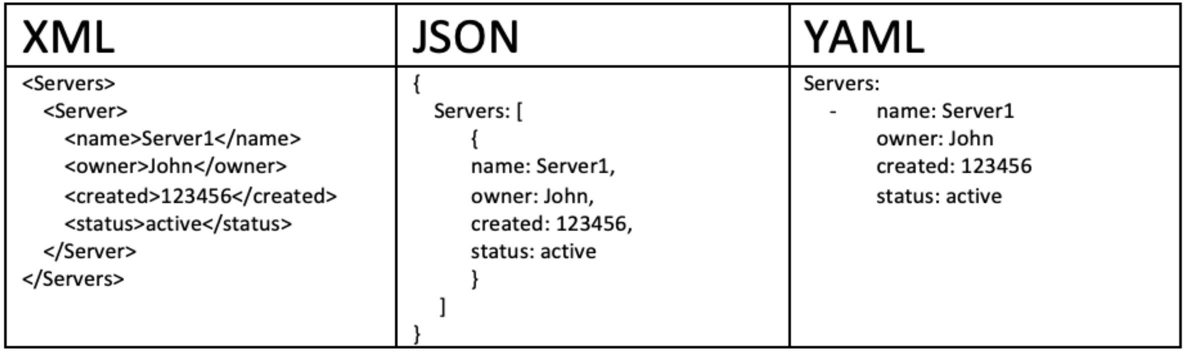
Github detecteert automatisch Actions workflows in de repository als je ze opslaat in de map .github/workflows.

Je mag de naamgeving van deze bestanden zelf kiezen, maar ze moeten eindigen op .yml of .yaml. Yaml is een markup taal die we voornamelijk gebruiken voor configuratiebestanden

https://docs.github.com/en/actions/writing-workflows/quickstart

Yaml

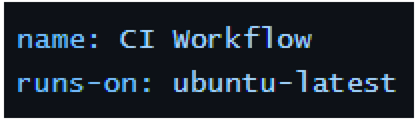
Definitie: "**YAML** is een voor mensen leesbaar bestandsformaat, dat gebruikt wordt voor onder andere configuratiebestanden en in applicaties voor data-opslag en verzending. Het formaat bestaat sinds 2001, en gebruikt sinds 2006 de bestandsextentie .yaml. De opmaak gebeurt met **spaties**, en niet met tabs"



XML	JSON	YAML
<Servers>	{	Servers:
<Server>	Servers: [	- name: Server1
<name>Server1</name>	{	owner: John
<owner>John</owner>	name: Server1,	created: 123456
<created>123456</created>	owner: John,	status: active
<status>active</status>	created: 123456,	
</Server>	status: active	
</Servers>	}	
	]	
	}	

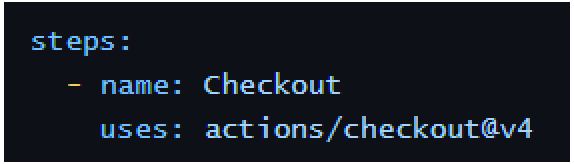
Yaml cheatsheet

Key-value pairs:



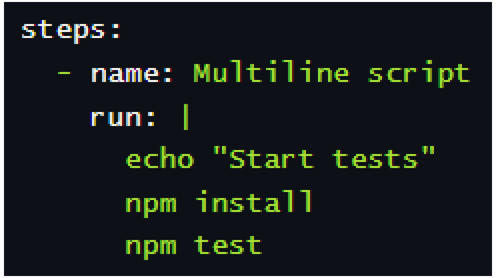
```yaml
name: CI Workflow
runs-on: ubuntu-latest
```

Lijsten:



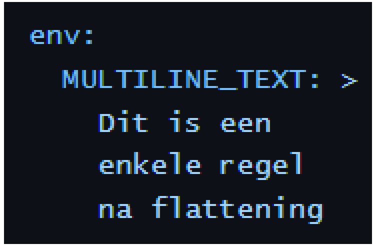
```yaml
steps:
  - name: Checkout
    uses: actions/checkout@v4
```

Multiline script:



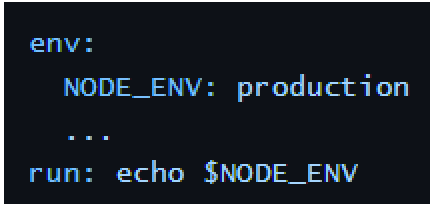
```yaml
steps:
  - name: Multiline script
    run: |
      echo "Start tests"
      npm install
      npm test
```

Multiline tekst:



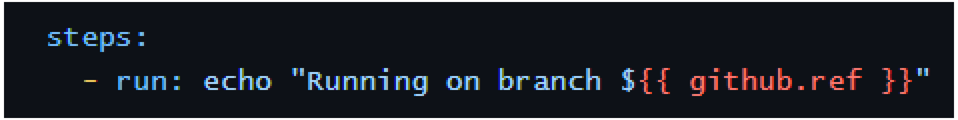
```yaml
env:
  MULTILINE_TEXT: >
    Dit is een
    enkele regel
    na flattening
```

Variabelen:



```yaml
env:
  NODE_ENV: production
  ...
run: echo $NODE_ENV
```

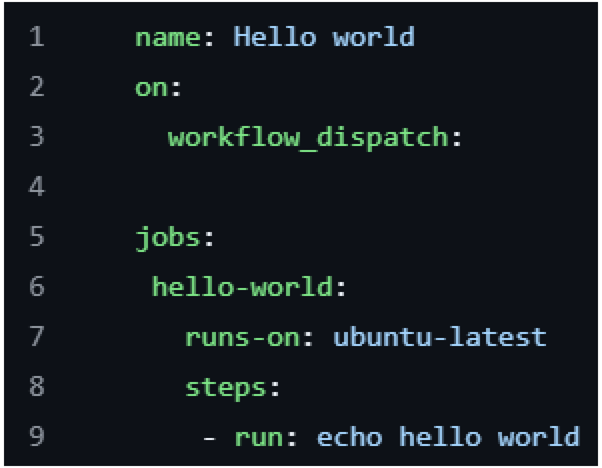
Github variabelen:



```yaml
steps:
  - run: echo "Running on branch ${{ github.ref }}"
```

Hello world

- Navigeer naar de Github actions tab en maak een nieuwe workflow.yml file aan
- Voorzie volgende inhoud:

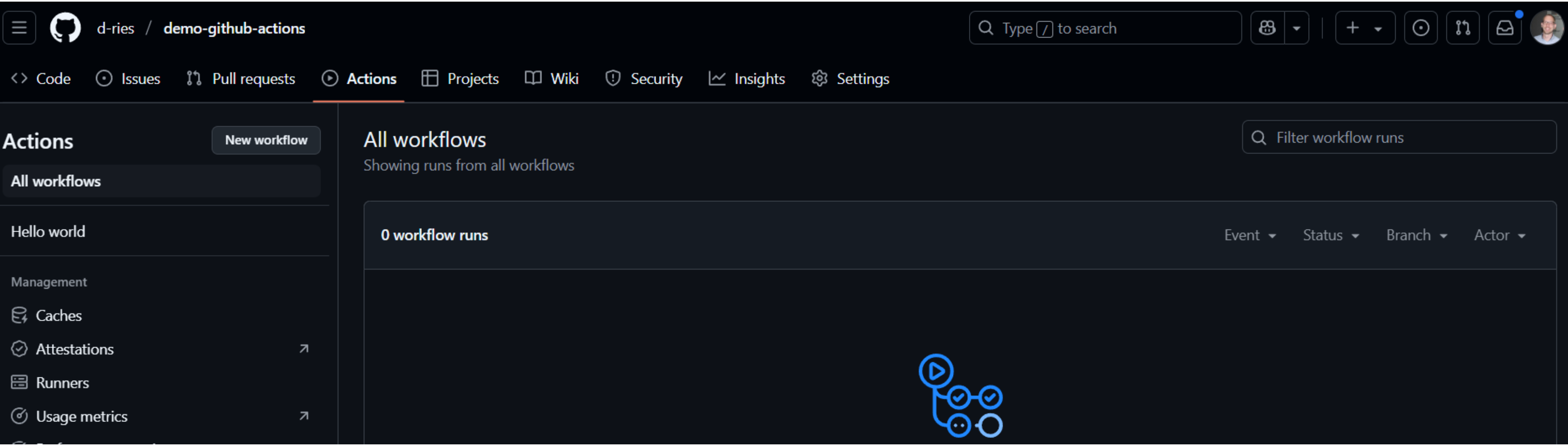


```yaml
name: Hello world
on:
  workflow_dispatch:

jobs:
 hello-world:
   runs-on: ubuntu-latest
   steps:
    - run: echo hello world
```

Hello world

- Onder de actions tab zie je nu de "hello world" workflow staan
  - Dashboard met algemene info over je workflow / pipelines & runs

    = Visuele feedback



d-ries / demo-github-actions

Type / to search

Code · Issues · Pull requests · Actions · Projects · Wiki · Security · Insights · Settings

Actions — New workflow

All workflows

Hello world

Management

Caches

Attestations

Runners

Usage metrics

All workflows

Showing runs from all workflows

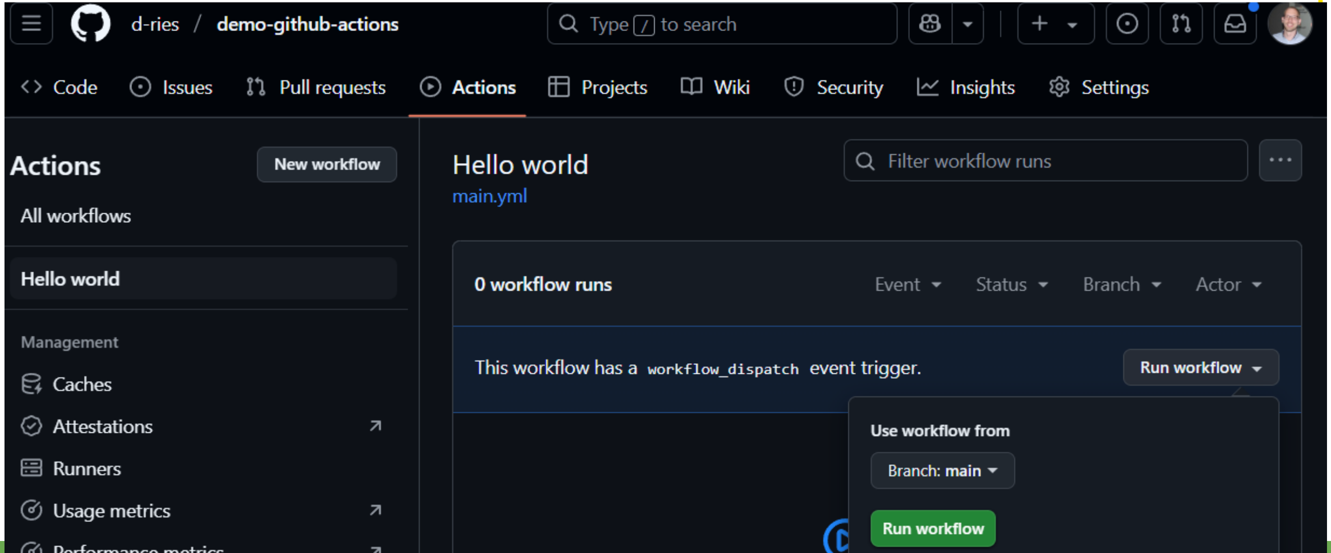
Filter workflow runs

0 workflow runs

Event · Status · Branch · Actor

Hello world

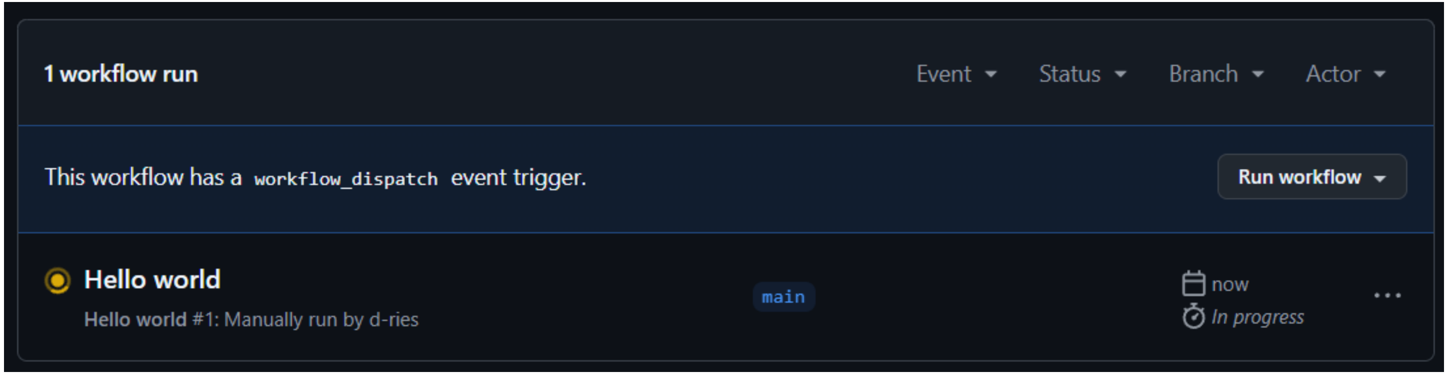
- Elke workflow heeft zijn eigen dashboard met een historiek van de workflow runs
- Het hello world voorbeeld kan je handmatig starten met de knop "run workflow)



This workflow has a `workflow_dispatch` event trigger.

Hello world

- Realtime feedback over vorige en huidige build pogingen
  - Gebruik van kleuren, identifier #, timestamps, welke gebruiker deze start, …
  - Later ook linken met bv commit digests



1 workflow run

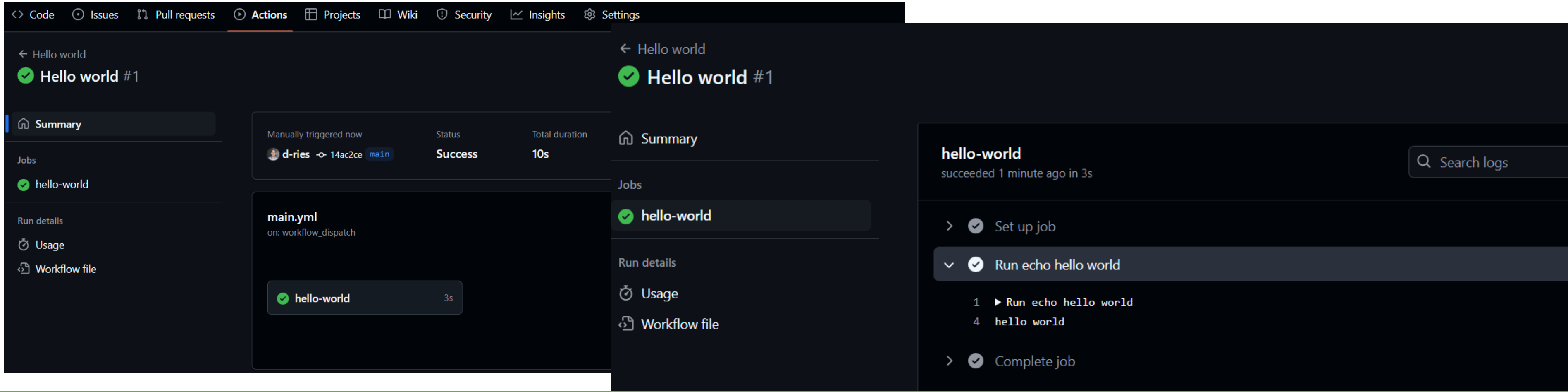
This workflow has a `workflow_dispatch` event trigger.

Hello world

Hello world #1: Manually run by d-ries

Hello world

- Elke workflow run heeft nog eens zijn eigen dashboard met informatie over die specifieke run:
    - Algemene metrics, volledige log van alle uitgevoerde jobs & steps met hun output
    - Eventuele opgeslagen artifacts (later meer)



Hello world #1

Manually triggered now — d-ries 14ac2ce main

Status: Success

Total duration: 10s

main.yml
on: workflow_dispatch

hello-world 3s

Hello world #1

hello-world
succeeded 1 minute ago in 3s

Set up job

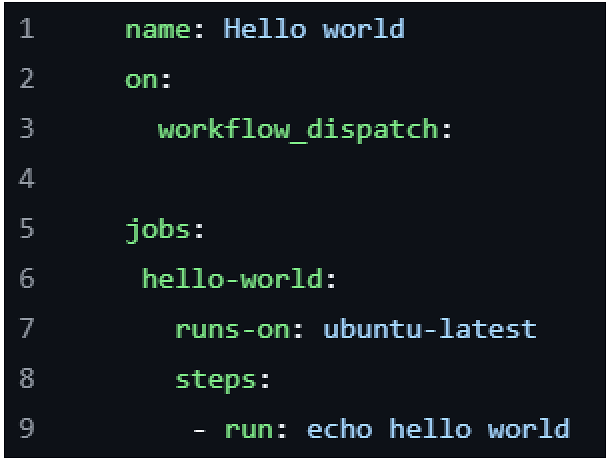
Run echo hello world

```
▶ Run echo hello world
hello world
```

Complete job

Structuur Workflow

- Vaste structuur
  - Definitie pipeline: Algemene metadata van de pipeline zoals de naam
  - Triggers: Op welke manier start de pipeline
  - Jobs: Welke verschillende grote blokken bevat deze pipeline
    - Steps: Een job heeft één of meerdere stapjes
  - Structuur uitbreidbaar met extra benodigdheden, bovenstaande is het minimum

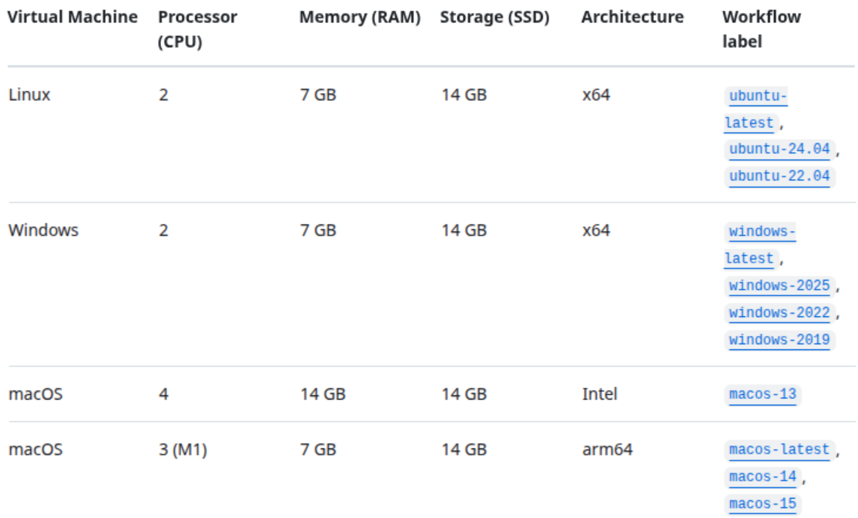


```yaml
name: Hello world
on:
  workflow_dispatch:

jobs:
 hello-world:
   runs-on: ubuntu-latest
   steps:
    - run: echo hello world
```

Runs on ?

De Github Actions gebruikt achterliggend VMs voor de CI omgeving:

- Runs-on: Linux, Windows, MacOS
- Gratis voor public repos
- Aangerekend per lopende minuut voor private repos



Virtual Machine	Processor (CPU)	Memory (RAM)	Storage (SSD)	Architecture	Workflow label
Linux	2	7 GB	14 GB	x64	`ubuntu-latest`, `ubuntu-24.04`, `ubuntu-22.04`
Windows	2	7 GB	14 GB	x64	`windows-latest`, `windows-2025`, `windows-2022`, `windows-2019`
macOS	4	14 GB	14 GB	Intel	`macos-13`
macOS	3 (M1)	7 GB	14 GB	arm64	`macos-latest`, `macos-14`, `macos-15`

Kan gebruikt worden voor ALLE soorten jobs:

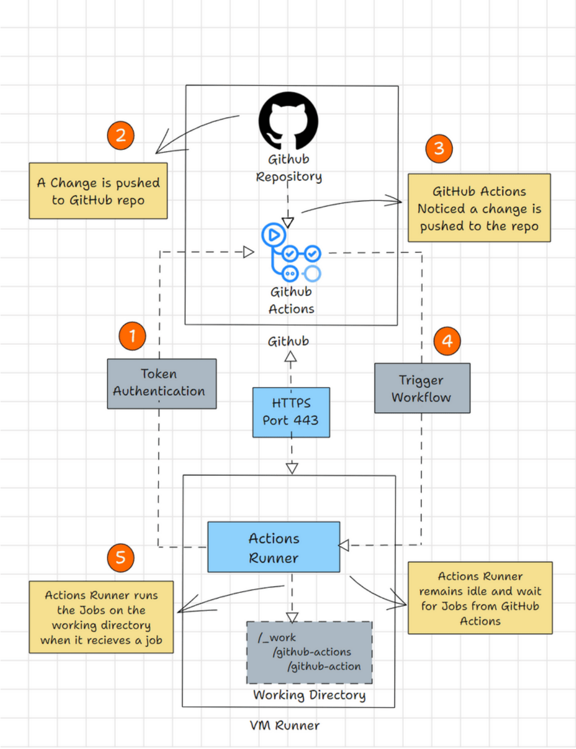
- Builds
- Unit testing
- Functionele testen (zie volgend hoofdstuk)

https://docs.github.com/en/actions/using-github-hosted-runners/using-github-hosted-runners/about-github-hosted-runners

Runs on ?

OF je host je eigen runner:

- Geen Github kost
- Volledige controle
- Soms nodig in usecases waar omgevingen en/of data niet direct toegankelijk zijn


2
A Change is pushed to GitHub repo
Github Repository
3
GitHub Actions Noticed a change is pushed to the repo
Github Actions
Github
1
Token Authentication
HTTPS Port 443
4
Trigger Workflow
Actions Runner
5
Actions Runner runs the Jobs on the working directory when it recieves a job
Actions Runner remains idle and wait for Jobs from GitHub Actions
/_work
/github-actions
/github-action
Working Directory
VM Runner

https://docs.github.com/en/actions/hosting-your-own-runners/managing-self-hosted-runners/about-self-hosted-runners

Jobs

- Een job is een verzameling stappen die uitgevoerd wordt
  - Alle stappen van een job worden op één virtuele machine uitgevoerd
- Opsplitsing in meerdere jobs mogelijk
  - Jobs kunnen afhankelijk zijn van elkaar
  - separation of concerns
- **Gebruik meerdere jobs buiten de scope van het vak**

https://docs.github.com/en/actions/writing-workflows/choosing-what-your-workflow-does/using-jobs-in-a-workflow

Github Actions: Workflow jobs

Jobs:

- Runs-on:

  Omgeving: Linux, Windows, MacOS

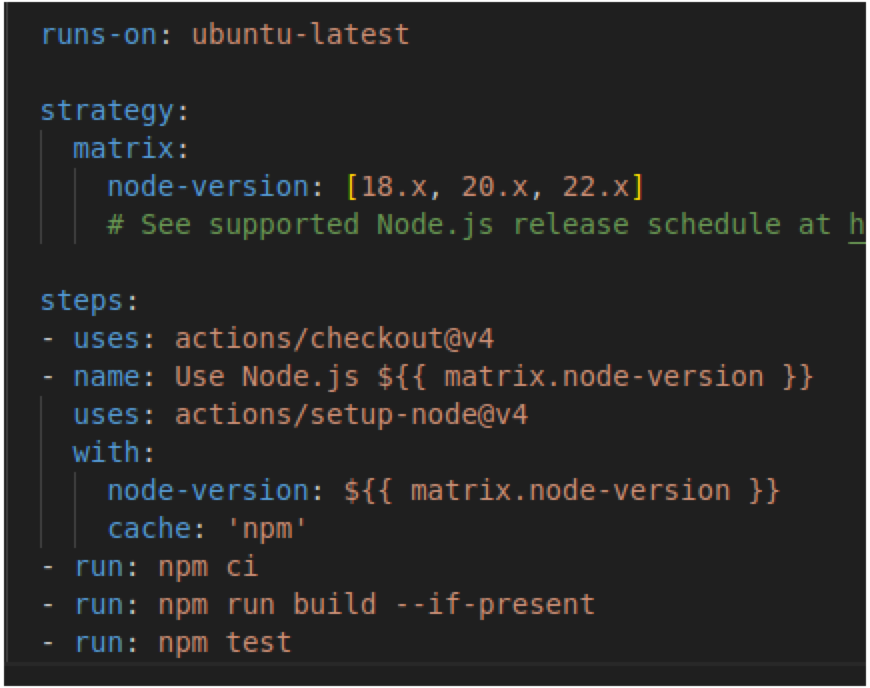
- Strategy:

  Welke versie van iets willen we gebruiken

- Steps:

  uses: externe, predefined action

  run: CLI commando

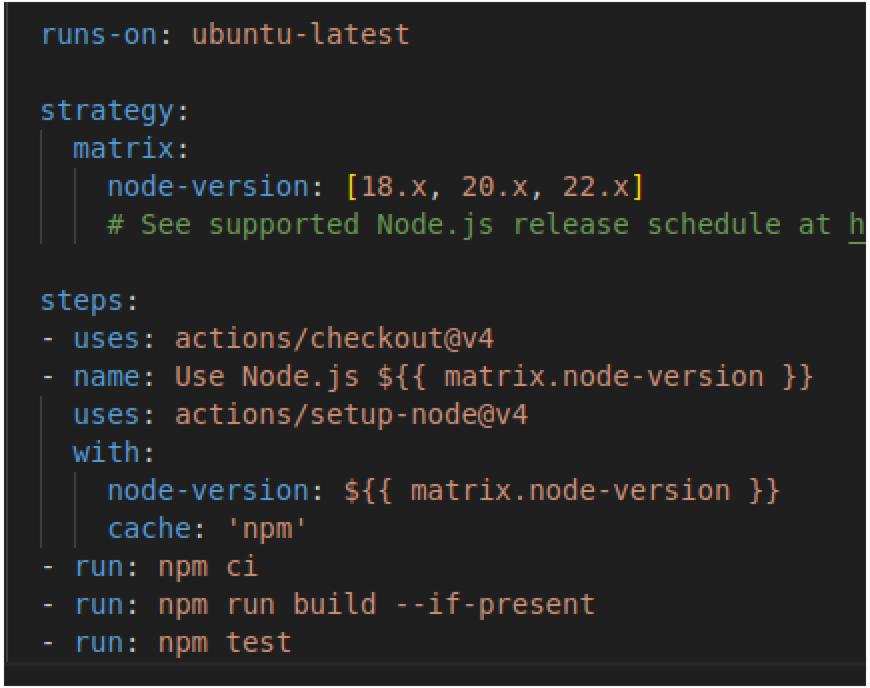


```yaml
runs-on: ubuntu-latest

strategy:
  matrix:
    node-version: [18.x, 20.x, 22.x]
    # See supported Node.js release schedule at h

steps:
- uses: actions/checkout@v4
- name: Use Node.js ${{ matrix.node-version }}
  uses: actions/setup-node@v4
  with:
    node-version: ${{ matrix.node-version }}
    cache: 'npm'
- run: npm ci
- run: npm run build --if-present
- run: npm test
```

https://docs.github.com/en/actions/writing-workflows/choosing-what-your-workflow-does/using-jobs-in-a-workflow

Steps

- Een job bestaat uit één of meerdere steps
    - ○ Opsplitsing van taken van een job
    - ○ Worden sequentieel uitgevoerd
    - ○ Output van steps kan doorgegeven worden (via systeem variabele, zie docs)
    - ○ Bij het falen van een step stopt de job



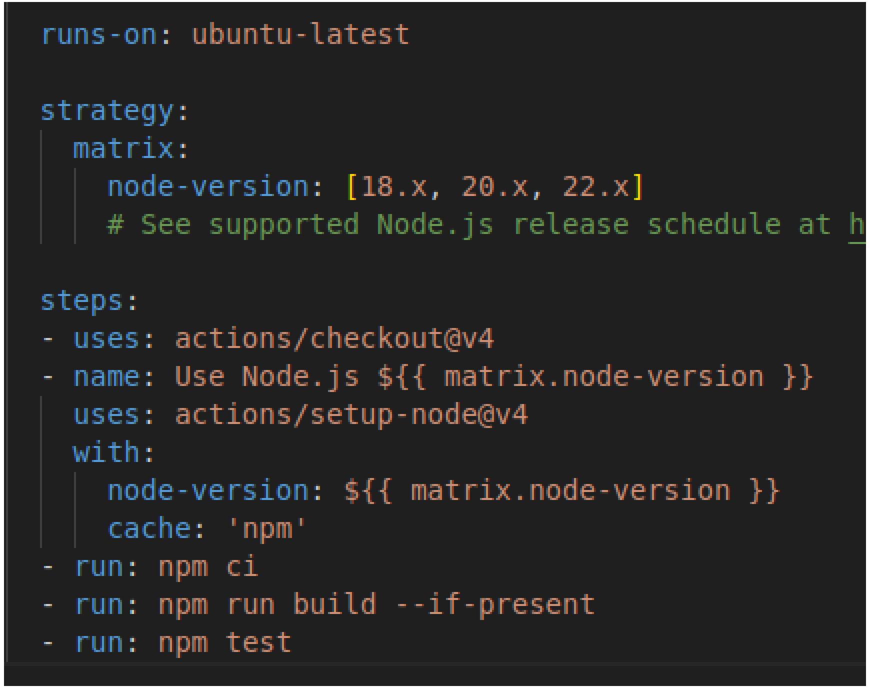
```yaml
runs-on: ubuntu-latest

strategy:
  matrix:
    node-version: [18.x, 20.x, 22.x]
    # See supported Node.js release schedule at h

steps:
- uses: actions/checkout@v4
- name: Use Node.js ${{ matrix.node-version }}
  uses: actions/setup-node@v4
  with:
    node-version: ${{ matrix.node-version }}
    cache: 'npm'
- run: npm ci
- run: npm run build --if-present
- run: npm test
```

Github Actions: Workflow steps

- Steps:

  uses: externe, predefined action

  run: CLI commando



```yaml
runs-on: ubuntu-latest

strategy:
  matrix:
    node-version: [18.x, 20.x, 22.x]
    # See supported Node.js release schedule at h

steps:
- uses: actions/checkout@v4
- name: Use Node.js ${{ matrix.node-version }}
  uses: actions/setup-node@v4
  with:
    node-version: ${{ matrix.node-version }}
    cache: 'npm'
- run: npm ci
- run: npm run build --if-present
- run: npm test
```

Uses zijn externe github scripts die verified zijn

Bvb actions/checkout@v4:

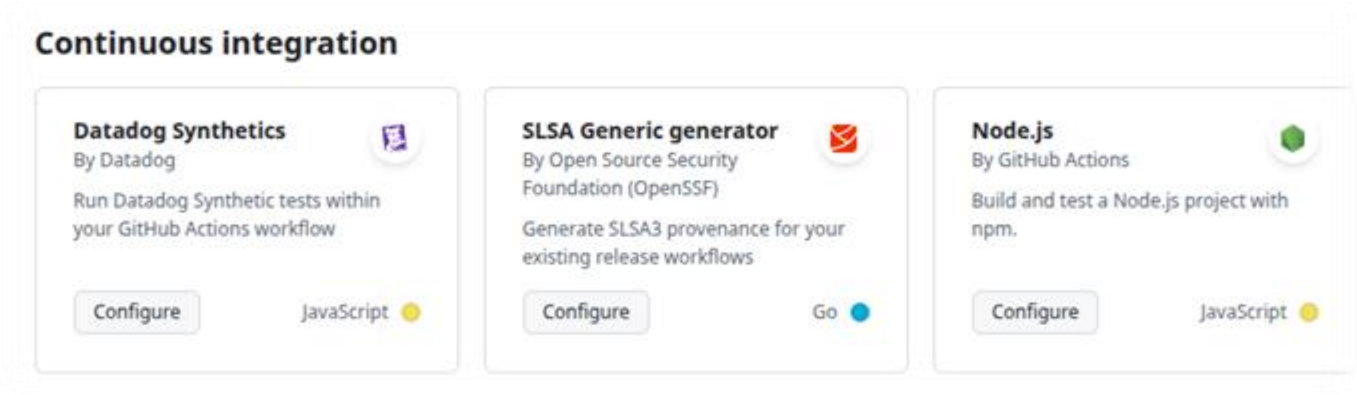
https://github.com/actions/checkout

Default usage, maar kan custom parameters aanvaarden, zie setup-node bvb.

https://docs.github.com/en/actions/writing-workflows/choosing-what-your-workflow-does/using-jobs-in-a-workflow

Github Actions: Marketplace

You are not special....use the marketplace



Continuous integration

Datadog Synthetics
By Datadog
Run Datadog Synthetic tests within your GitHub Actions workflow

SLSA Generic generator
By Open Source Security Foundation (OpenSSF)
Generate SLSA3 provenance for your existing release workflows

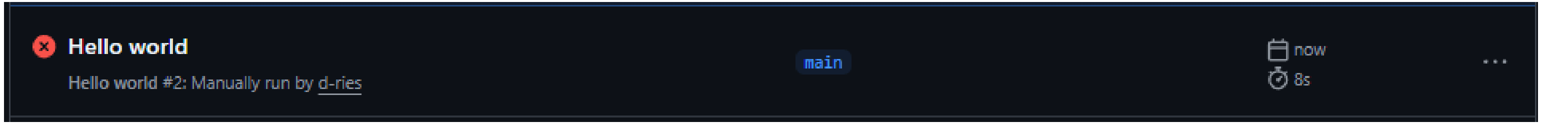
Node.js
By GitHub Actions
Build and test a Node.js project with npm.

For our devops calculator we know we have a node application, so lets use the proposed action.

https://docs.github.com/en/actions/writing-workflows/choosing-what-your-workflow-does/using-jobs-in-a-workflow

Errors & debugging

- Bij foutmeldingen in commando's of plugins stopt de workflow
  - Digital andon cord
- Visuele feedback dat er iets is misgegaan



☒ Hello world

Hello world #2: Manually run by d-ries

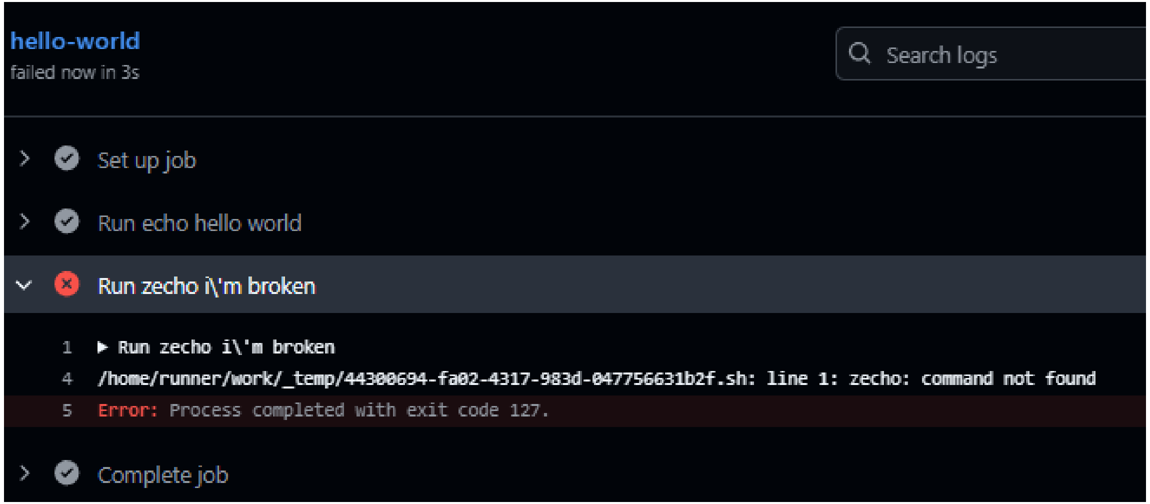
main

now

8s

Errors & debugging

- Technische feedback is terug te vinden in de rapportering van je workflow run
  - Per step kan je kijken wat er goed gaat en misloopt
  - Foutmeldingen



hello-world
failed now in 3s

- Set up job
- Run echo hello world
- Run zecho i\'m broken

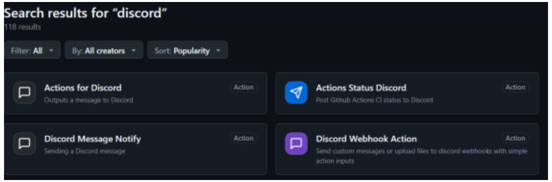
```
▶ Run zecho i\'m broken
/home/runner/work/_temp/44300694-fa02-4317-983d-047756631b2f.sh: line 1: zecho: command not found
Error: Process completed with exit code 127.
```

- Complete job

Dependencies

- Specifieke commando's / tooling nodig?

1. Controle of er een bestaande marketplace action is



Search results for "discord"

118 results

Filter: All ▾ By: All creators ▾ Sort: Popularity ▾

Actions for Discord Action
Outputs a message to Discord

Actions Status Discord Action
Post Github Actions CI status to Discord

Discord Message Notify Action
Sending a Discord message

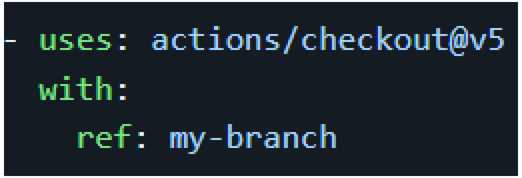
Discord Webhook Action Action
Send custom messages or upload files to discord webhooks with simple action inputs

2. Indien niet handmatig installeren in de runner voor gebruik OF gebruik self hosted runner met nodige depencencies

Checkout code

- Stap één van een Github actions run is vaak het binnentrekken van de code van de repository
- Volgen logica van de vorige slide:

https://github.com/marketplace/actions/checkout

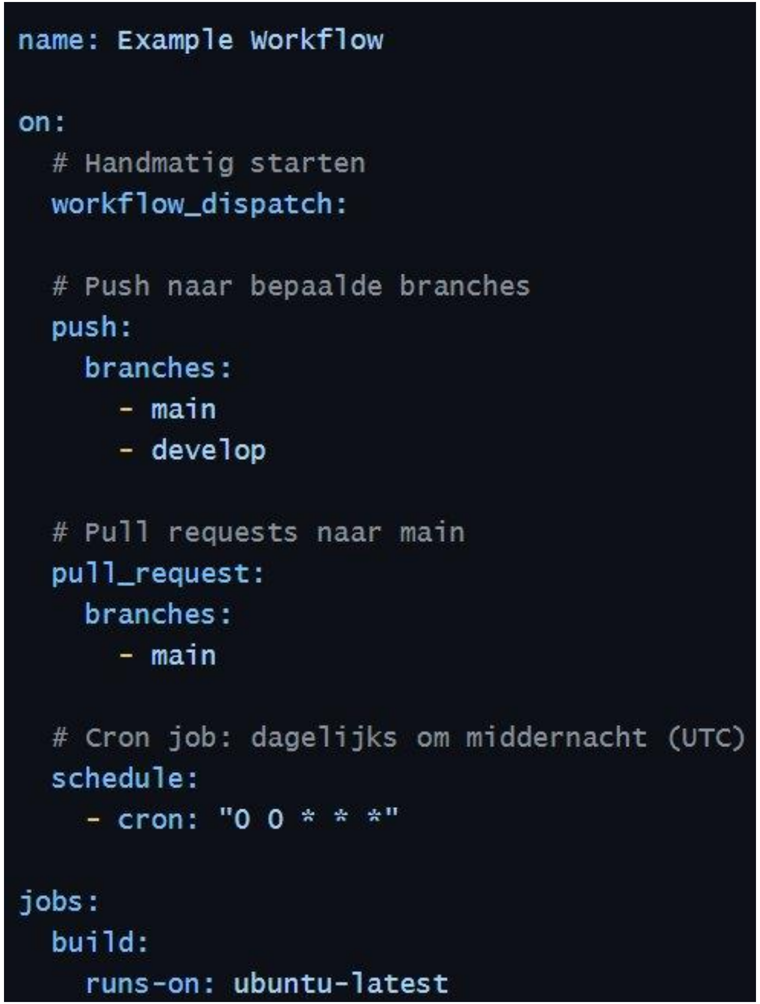


```
- uses: actions/checkout@v5
  with:
    ref: my-branch
```

Triggers

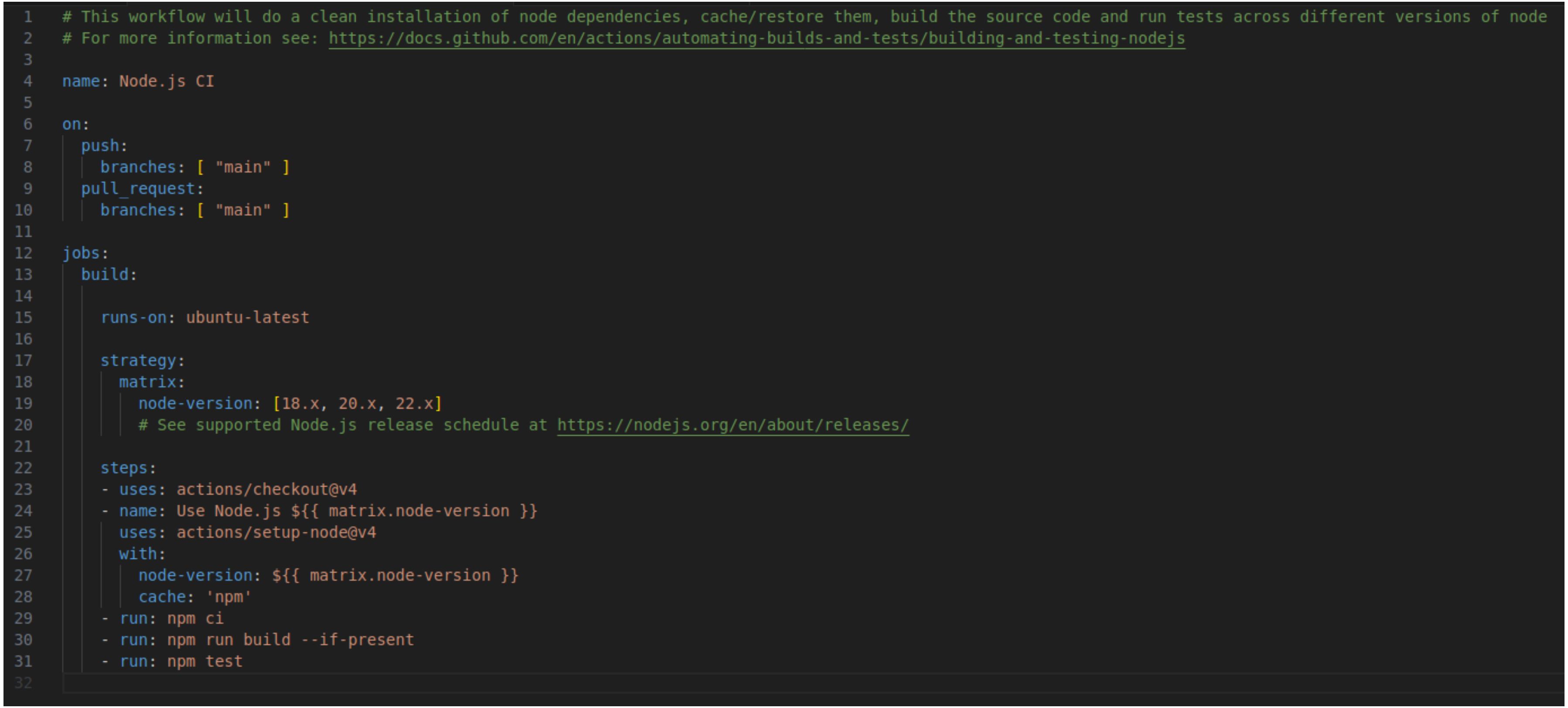
- Het on keyword bepaald wanneer de pipeline gerund wordt
    - Bij bepaalde events op bepaalde branches
    - Bij het aanmaken / commenten van issues
    - Op vaste tijdstippen
    - Manueel
    - …

Triggers



```yaml
name: Example Workflow

on:
  # Handmatig starten
  workflow_dispatch:

  # Push naar bepaalde branches
  push:
    branches:
      - main
      - develop

  # Pull requests naar main
  pull_request:
    branches:
      - main

  # Cron job: dagelijks om middernacht (UTC)
  schedule:
    - cron: "0 0 * * *"

jobs:
  build:
    runs-on: ubuntu-latest
```

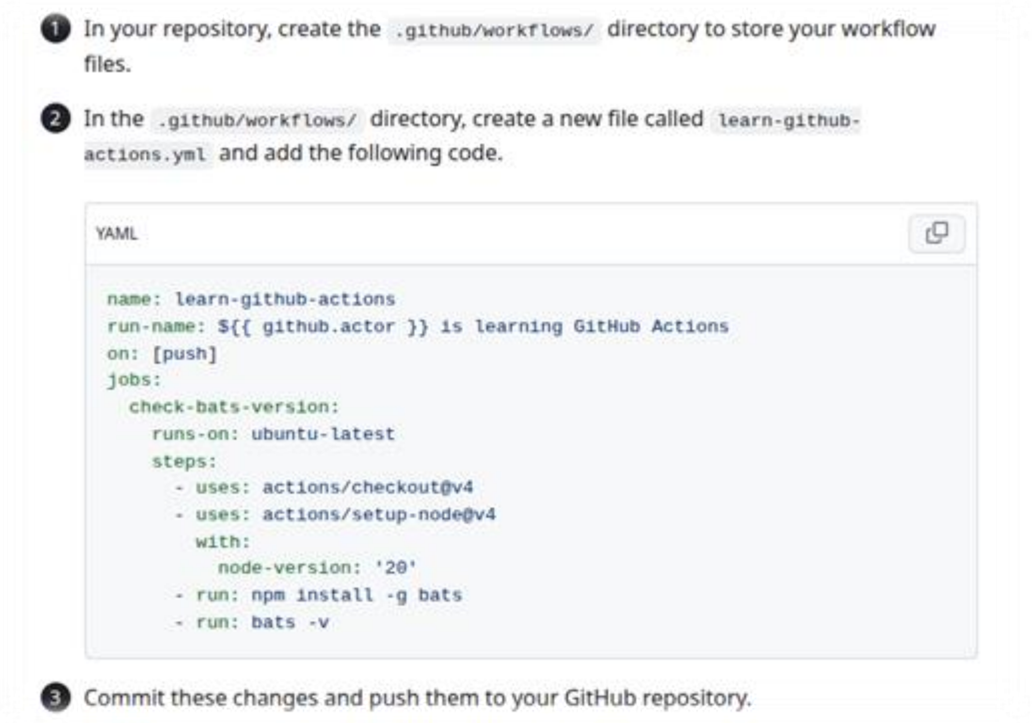
Github Actions: Workflow



```yaml
# This workflow will do a clean installation of node dependencies, cache/restore them, build the source code and run tests across different versions of node
# For more information see: https://docs.github.com/en/actions/automating-builds-and-tests/building-and-testing-nodejs

name: Node.js CI

on:
  push:
    branches: [ "main" ]
  pull_request:
    branches: [ "main" ]

jobs:
  build:

    runs-on: ubuntu-latest

    strategy:
      matrix:
        node-version: [18.x, 20.x, 22.x]
        # See supported Node.js release schedule at https://nodejs.org/en/about/releases/

    steps:
    - uses: actions/checkout@v4
    - name: Use Node.js ${{ matrix.node-version }}
      uses: actions/setup-node@v4
      with:
        node-version: ${{ matrix.node-version }}
        cache: 'npm'
    - run: npm ci
    - run: npm run build --if-present
    - run: npm test
```

https://docs.github.com/en/actions/writing-workflows/choosing-what-your-workflow-does/using-jobs-in-a-workflow

Github Actions: example



1. In your repository, create the `.github/workflows/` directory to store your workflow files.
2. In the `.github/workflows/` directory, create a new file called `learn-github-actions.yml` and add the following code.

YAML

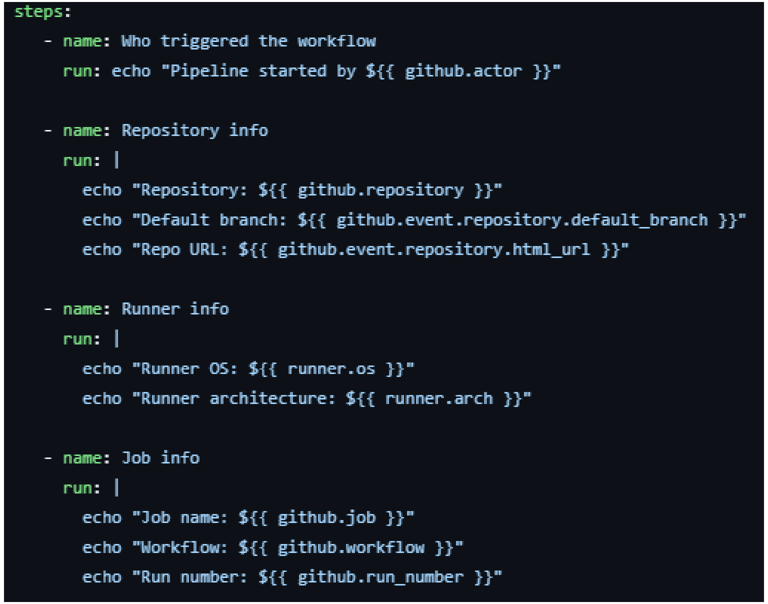
```yaml
name: learn-github-actions
run-name: ${{ github.actor }} is learning GitHub Actions
on: [push]
jobs:
  check-bats-version:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v4
      - uses: actions/setup-node@v4
        with:
          node-version: '20'
      - run: npm install -g bats
      - run: bats -v
```

3. Commit these changes and push them to your GitHub repository.

https://docs.github.com/en/actions/use-cases-and-examples/creating-an-example-workflow

Github Actions: Environment variables

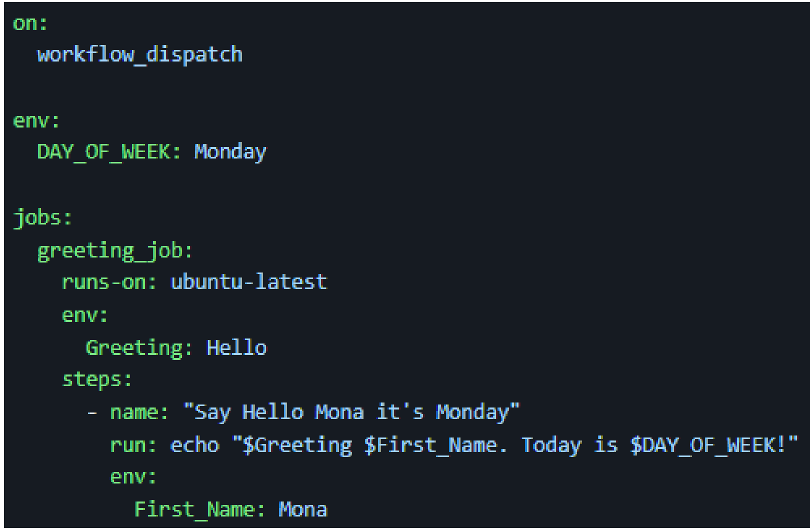
- Github voorziet een hele hoop systeemvariabelen over de context van je workflow run
- Alle info over :
  - workflow run
  - Repository
  - Job
  - Environment & secrets
  - ...



```yaml
steps:
  - name: Who triggered the workflow
    run: echo "Pipeline started by ${{ github.actor }}"

  - name: Repository info
    run: |
      echo "Repository: ${{ github.repository }}"
      echo "Default branch: ${{ github.event.repository.default_branch }}"
      echo "Repo URL: ${{ github.event.repository.html_url }}"

  - name: Runner info
    run: |
      echo "Runner OS: ${{ runner.os }}"
      echo "Runner architecture: ${{ runner.arch }}"

  - name: Job info
    run: |
      echo "Job name: ${{ github.job }}"
      echo "Workflow: ${{ github.workflow }}"
      echo "Run number: ${{ github.run_number }}"
```

Github Actions: Environment variables

- Daarnaast is het mogelijk om zelf (environment) variabelen aan te maken
- Om variabelen aan te maken maken we gebruik van het `env` keyword
  - Dit kan je op verschillende plaatsen gebruiken



```
on:
  workflow_dispatch

env:
  DAY_OF_WEEK: Monday

jobs:
  greeting_job:
    runs-on: ubuntu-latest
    env:
      Greeting: Hello
    steps:
      - name: "Say Hello Mona it's Monday"
        run: echo "$Greeting $First_Name. Today is $DAY_OF_WEEK!"
        env:
          First_Name: Mona
```

Github Actions: Secret management

- Wat als we gevoelig informatie in onze workflow willen gebruiken?
    - Username, passwords, tokens, private keys, …
- **In plaintext in onze YML files = security risk!**

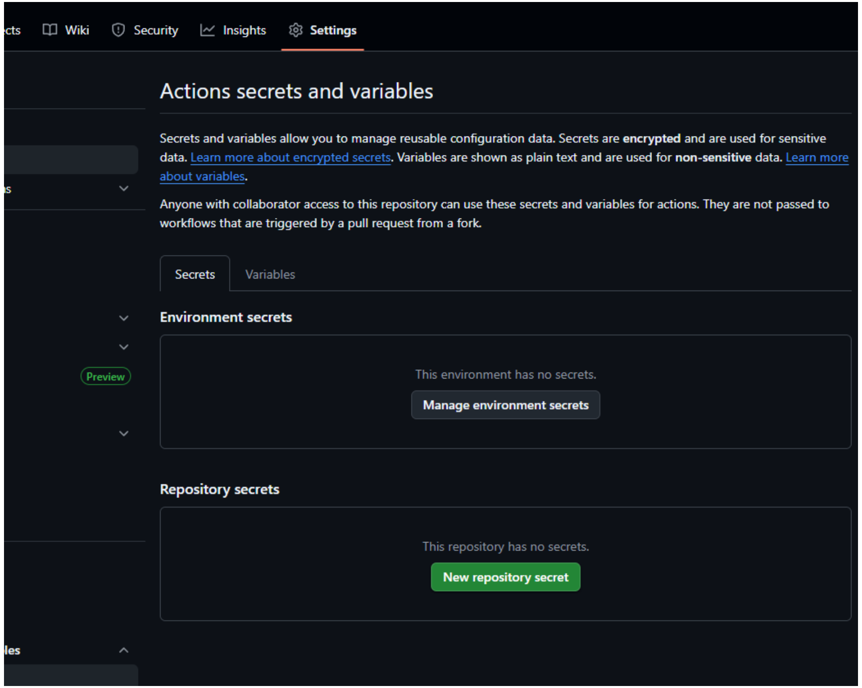
Github Actions: secrets

voorbeeld: verbinden met een VM in de cloud om onze code te deployen:

```
jobs:
deploy:
runs-on: ubuntu-latest
steps:
- uses: actions/checkout@v2
- name: Deploy to EC2
  env:
    PRIVATE_KEY: "----–BEGIN OPENSSH PRIVATE KEY----–B3BlbnNzaC...AI4fVFeKj9AliW2Jgaxeg==----–END OPENSSH PRIVATE KEY----–"

  HOST: 52.48.75.998
  USER: ec2-user
  run: |
  echo "$PRIVATE_KEY" > github-ec2.pem && chmod 600 github-ec2.pem
  ssh -o StrictHostKeyChecking=no -i github-ec2.pem ${USER}@${HOST} '
```

Github Actions: Secret management

- Gebruik maken van ingebouwde secret store van Github
- Kluis die onze gevoelige data beschermt en injecteert waar nodig
- Admin rechten op repository nodig



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.

Environment secrets

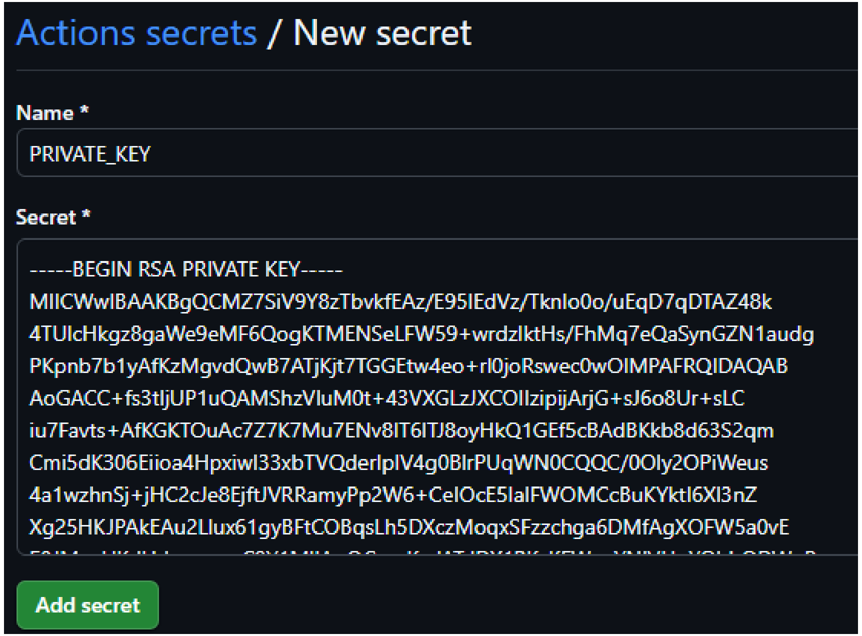
This environment has no secrets.

Repository secrets

This repository has no secrets.

Github Actions: Secret management

- Gebruik maken van ingebouwde secret store van Github
- Kluis die onze gevoelige data beschermt en injecteert waar nodig
- Admin rechten op repository nodig



Actions secrets / New secret

Name *

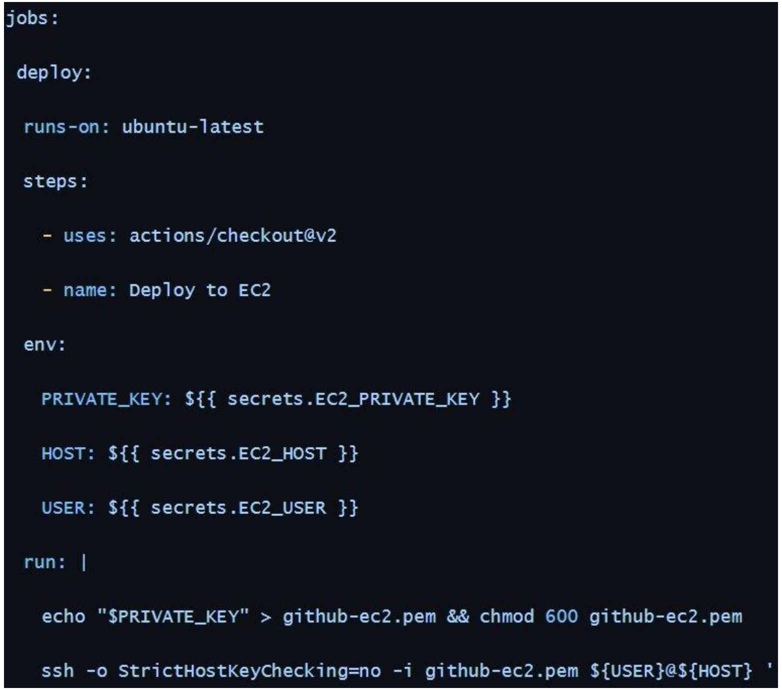
PRIVATE_KEY

Secret *

```
-----BEGIN RSA PRIVATE KEY-----
MIICWwIBAAKBgQCMZ7SiV9Y8zTbvkfEAz/E95IEdVz/Tknlo0o/uEqD7qDTAZ48k
4TUIcHkgz8gaWe9eMF6QogKTMENSeLFW59+wrdzIktHs/FhMq7eQaSynGZN1audg
PKpnb7b1yAfKzMgvdQwB7ATjKjt7TGGEtw4eo+rl0joRswec0wOIMPAFRQIDAQAB
AoGACC+fs3tIjUP1uQAMShzVluM0t+43VXGLzJXCOIIzipijArjG+sJ6o8Ur+sLC
iu7Favts+AfKGKTOuAc7Z7K7Mu7ENv8IT6ITJ8oyHkQ1GEf5cBAdBKkb8d63S2qm
Cmi5dK306Eiioa4HpxiwI33xbTVQderIpIV4g0BIrPUqWN0CQQC/0Oly2OPiWeus
4a1wzhnSj+jHC2cJe8EjftJVRRamyPp2W6+CeIOcE5IaIFWOMCcBuKYktI6XI3nZ
Xg25HKJPAkEAu2Llux61gyBFtCOBqsLh5DXczMoqxSFzzchga6DMfAgXOFW5a0vE
```

Github Actions: Secret management

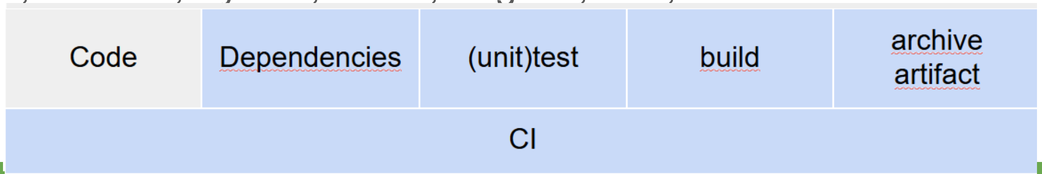
- In een workflow file zijn secrets aanspreekbaar via ${{ secrets.NAAMSECRET }}
- Integratie via environment variables



```
jobs:
 deploy:
  runs-on: ubuntu-latest
  steps:
   - uses: actions/checkout@v2
   - name: Deploy to EC2
  env:
   PRIVATE_KEY: ${{ secrets.EC2_PRIVATE_KEY }}
   HOST: ${{ secrets.EC2_HOST }}
   USER: ${{ secrets.EC2_USER }}
  run: |
   echo "$PRIVATE_KEY" > github-ec2.pem && chmod 600 github-ec2.pem
   ssh -o StrictHostKeyChecking=no -i github-ec2.pem ${USER}@${HOST} '
```

Github Actions: Continious integration

- CI is niet zwart wit:
    - Elke technologiestack heeft zijn eigen stappen en tooling
    - Elk bedrijf hecht meer of minder waarde aan bepaalde stappen
    - Doel blijft hetzelfde: hoge velocity & snelle/eenvoudige deployments
- Niet nodig om het wiel opnieuw uit te vinden
    - Maak gebruik van de Github actions marketplace
    - Online veel voorbeelden & praktische cases te vinden voor populaire technologiestacks zoals Java, .NET, NodeJS, Python, React, Angular, Vue, ...



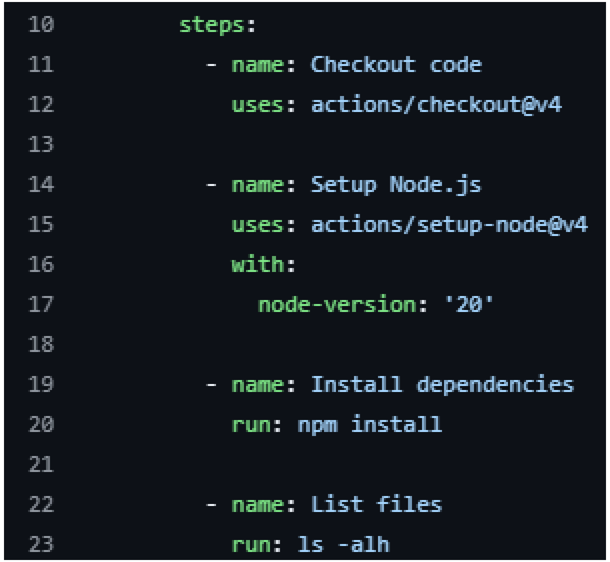
Code	Dependencies	(unit)test	build	archive artifact
CI				

Github Actions: pipeline – Installing dev dependencies

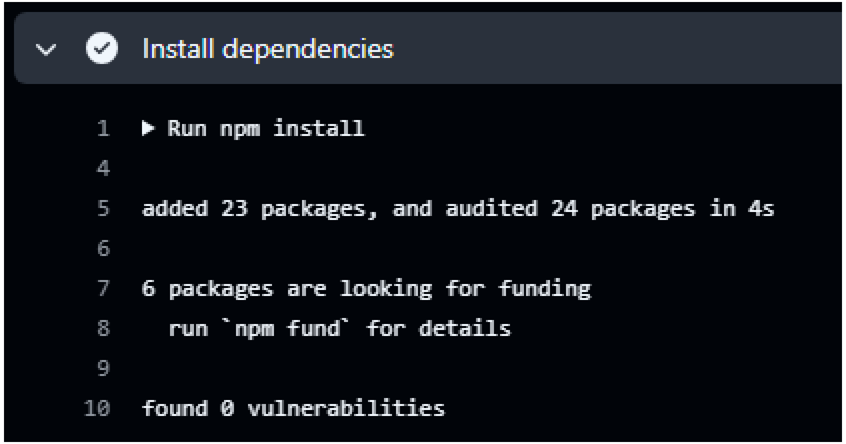
- Vaak gebruiken we tools om 3rd party packages & modules te installeren:
    - Npm voor NodeJS
    - Pip voor Python
    - Composer voor PHP
    - …
- Gebruik Marketplace plugins voor integratie van deze tools waar mogelijk!

Github Actions: pipeline – Installing dev dependencies

- NodeJS voorbeeld



```yaml
    steps:
      - name: Checkout code
        uses: actions/checkout@v4

      - name: Setup Node.js
        uses: actions/setup-node@v4
        with:
          node-version: '20'

      - name: Install dependencies
        run: npm install

      - name: List files
        run: ls -alh
```



Install dependencies

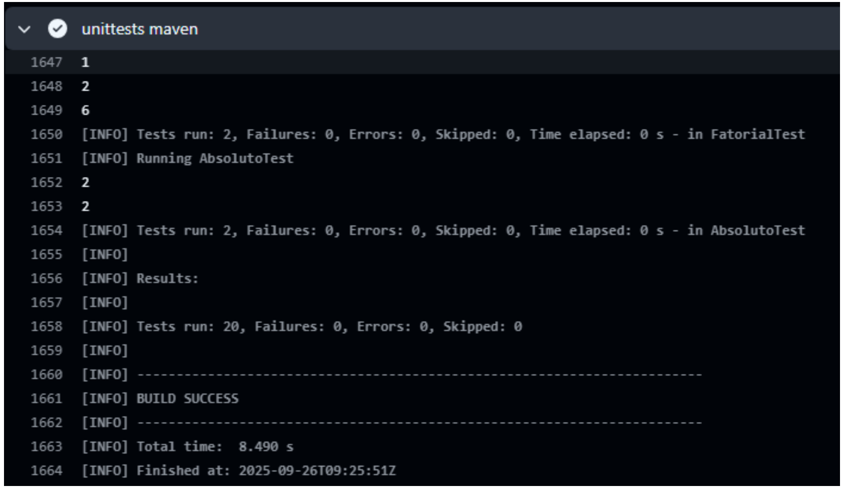
```
▶ Run npm install

added 23 packages, and audited 24 packages in 4s

6 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

Github Actions: pipeline - Unit testing

- Hangt af van de testrunner van de applicatie
  - Java => JUnit
  - NodeJS => Jest
- Vaak integratie vanuit andere talen naar JUnit
- Testrunners hebben verschillende export mogelijkheden voor rapportering

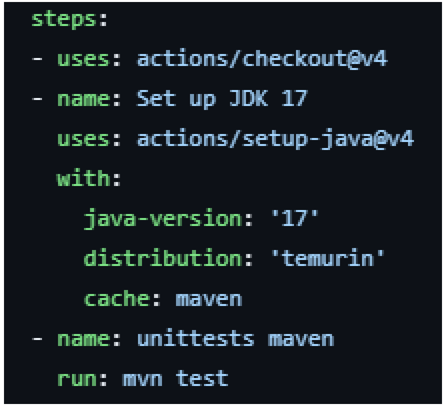
Github Actions: pipeline - Unit testing

- Testen runnen a.d.h.v. testunner
- Default output in console
    - Minder overzichtelijk



✓ unittests maven

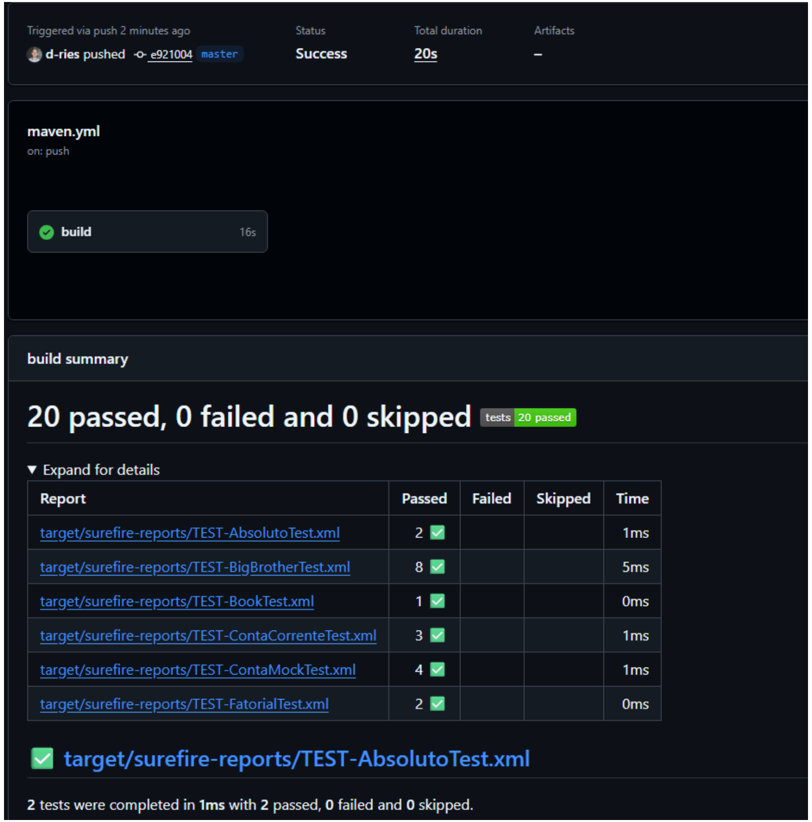
```
1
2
6
[INFO] Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0 s - in FatorialTest
[INFO] Running AbsolutoTest
2
2
[INFO] Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0 s - in AbsolutoTest
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 20, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  8.490 s
[INFO] Finished at: 2025-09-26T09:25:51Z
```



```yaml
steps:
- uses: actions/checkout@v4
- name: Set up JDK 17
  uses: actions/setup-java@v4
  with:
    java-version: '17'
    distribution: 'temurin'
    cache: maven
- name: unittests maven
  run: mvn test
```

Github Actions: pipeline - Unit testing

- Rapportering uit de console trekken a.d.h.v. test-reporter
  - Integratie met Github UI
  - Link aan build poging



Triggered via push 2 minutes ago
d-ries pushed e921004 master

Status
Success

Total duration
20s

Artifacts
–

maven.yml
on: push

build summary

20 passed, 0 failed and 0 skipped

▼ Expand for details

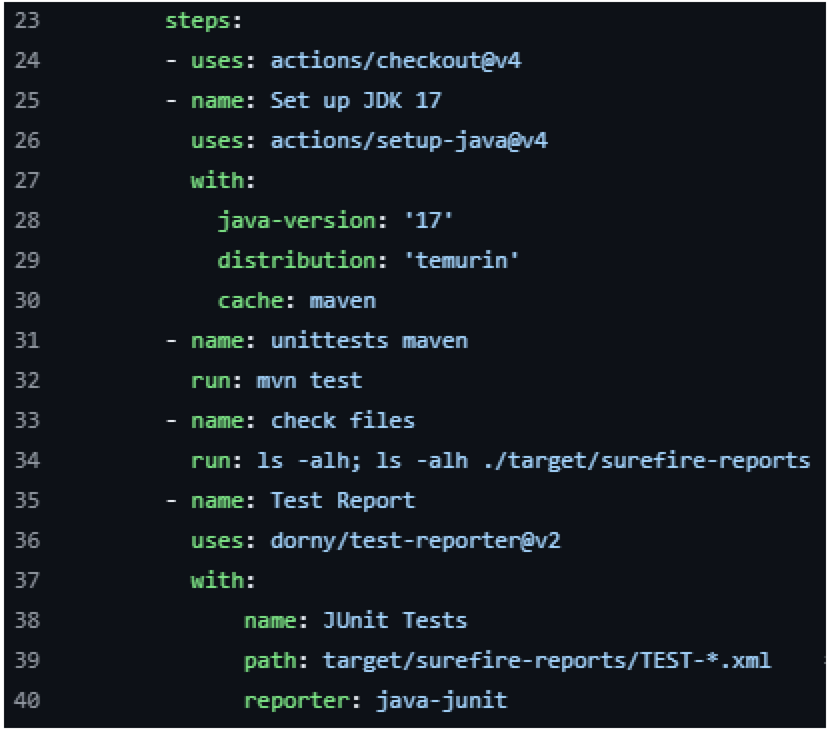
Report	Passed	Failed	Skipped	Time
target/surefire-reports/TEST-AbsolutoTest.xml	2 ✓			1ms
target/surefire-reports/TEST-BigBrotherTest.xml	8 ✓			5ms
target/surefire-reports/TEST-BookTest.xml	1 ✓			0ms
target/surefire-reports/TEST-ContaCorrenteTest.xml	3 ✓			1ms
target/surefire-reports/TEST-ContaMockTest.xml	4 ✓			1ms
target/surefire-reports/TEST-FatorialTest.xml	2 ✓			0ms

✓ target/surefire-reports/TEST-AbsolutoTest.xml

2 tests were completed in **1ms** with **2** passed, **0** failed and **0** skipped.

Github Actions: pipeline - Unit testing

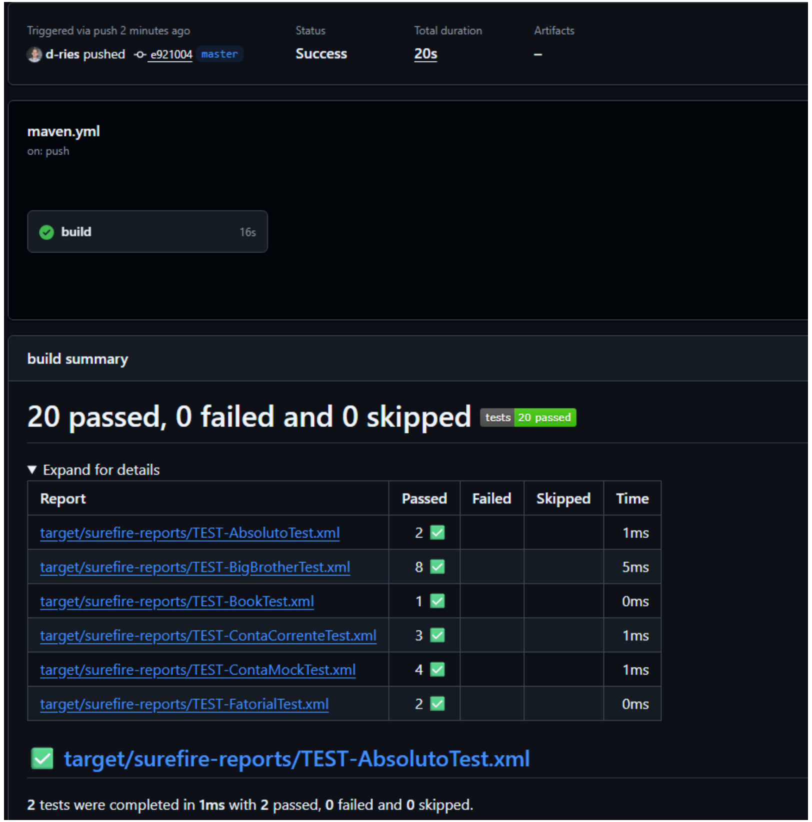
- Rapportering uit de console trekken a.d.h.v. test-reporter
    - Integratie met Github UI
    - Link aan build poging



```
    steps:
    - uses: actions/checkout@v4
    - name: Set up JDK 17
      uses: actions/setup-java@v4
      with:
        java-version: '17'
        distribution: 'temurin'
        cache: maven
    - name: unittests maven
      run: mvn test
    - name: check files
      run: ls -alh; ls -alh ./target/surefire-reports
    - name: Test Report
      uses: dorny/test-reporter@v2
      with:
          name: JUnit Tests
          path: target/surefire-reports/TEST-*.xml
          reporter: java-junit
```

Github Actions: pipeline - Unit testing

- Rapportering uit de console trekken a.d.h.v. test-reporter
  - Integratie met Github UI
  - Link aan build poging



Triggered via push 2 minutes ago
d-ries pushed -o- e921004 master

Status
Success

Total duration
20s

Artifacts
–

maven.yml
on: push

build 16s

build summary

20 passed, 0 failed and 0 skipped

tests 20 passed

▼ Expand for details

Report	Passed	Failed	Skipped	Time
target/surefire-reports/TEST-AbsolutoTest.xml	2 ✓			1ms
target/surefire-reports/TEST-BigBrotherTest.xml	8 ✓			5ms
target/surefire-reports/TEST-BookTest.xml	1 ✓			0ms
target/surefire-reports/TEST-ContaCorrenteTest.xml	3 ✓			1ms
target/surefire-reports/TEST-ContaMockTest.xml	4 ✓			1ms
target/surefire-reports/TEST-FatorialTest.xml	2 ✓			0ms

✓ target/surefire-reports/TEST-AbsolutoTest.xml

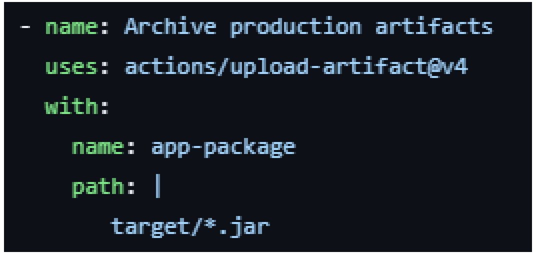
2 tests were completed in **1ms** with **2** passed, **0** failed and **0** skipped.

Jenkins - pipeline - Artifacts

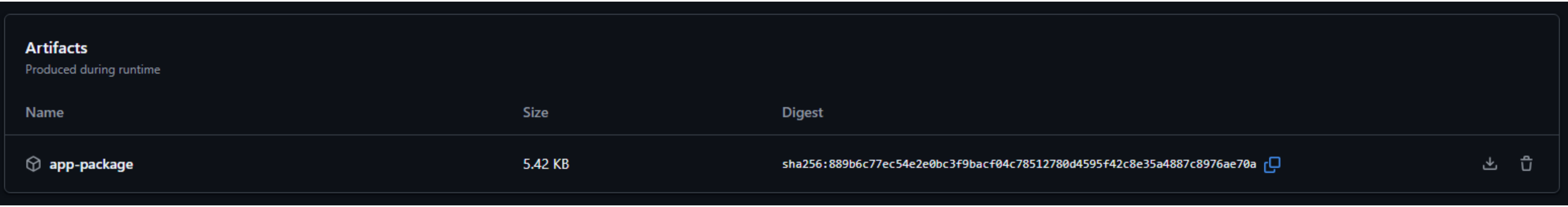
- Na het uitvoeren van je workflow run wordt alle data verwijderd
- Artifacts zijn bestanden die je wil bewaren na het uitvoeren van je workflow run
    - Uitvoerbare file(s) van je applicatie
    - Logs
    - Test resultaten
    - Rapporten

Jenkins - pipeline - Artifacts

- Na het uitvoeren van je workflow run wordt alle data verwijderd
- Artifacts zijn bestanden die je wil bewaren na het uitvoeren van je workflow run



```yaml
- name: Archive production artifacts
  uses: actions/upload-artifact@v4
  with:
    name: app-package
    path: |
      target/*.jar
```



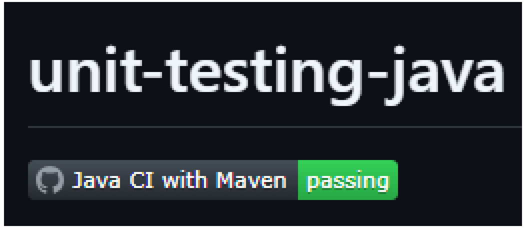
Artifacts

Produced during runtime

Name	Size	Digest
app-package	5.42 KB	sha256:889b6c77ec54e2e0bc3f9bacf04c78512780d4595f42c8e35a4887c8976ae70a

Extra integraties: Readme CI information

- Feedback over CI workflow kan rechtstreeks geïntegreerd worden in markdown files van de repository zoals README.md



unit-testing-java

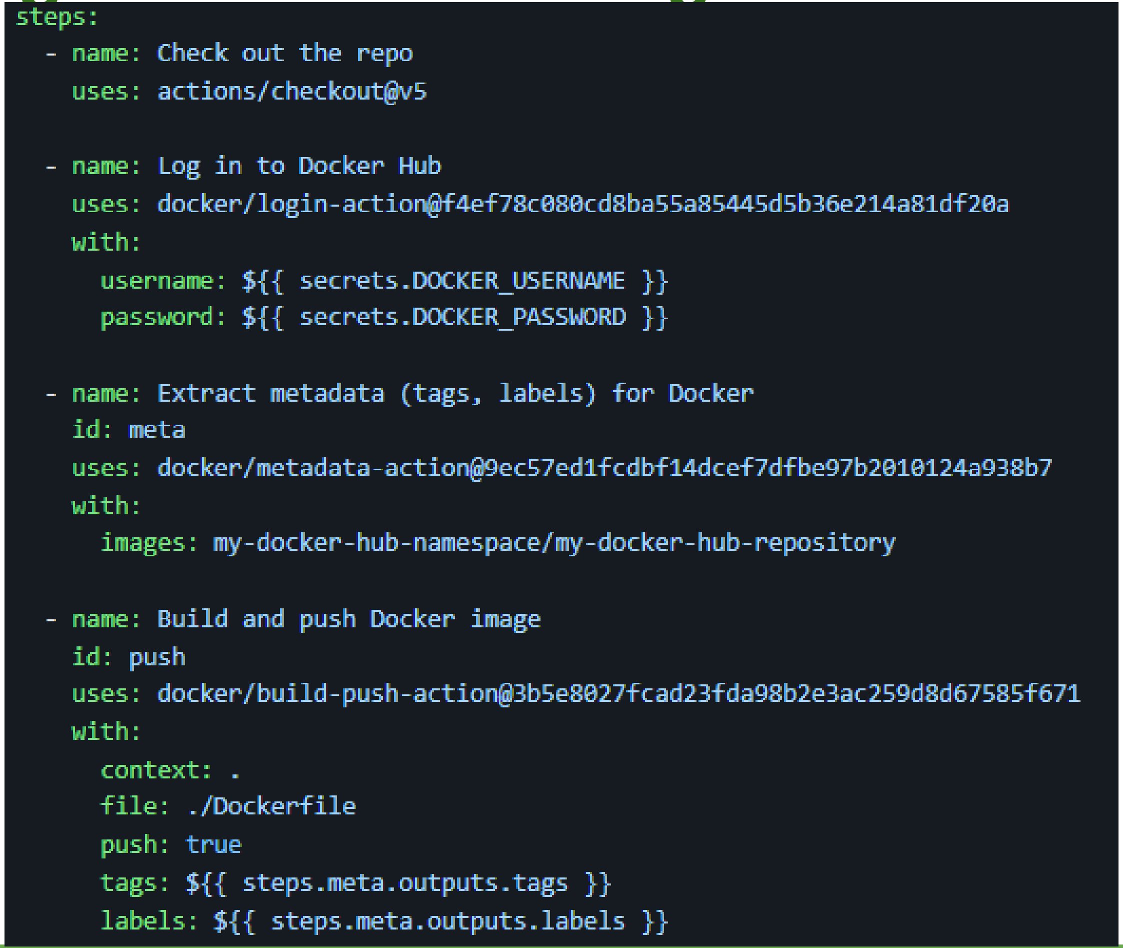
- Voorzien van volgende URL in markdown file:

![CI](https://github.com/<OWNER>/<REPO>/actions/workflows/<WORKFLOW_FILE>/badge.svg)

Extra integraties: Publishing artifacts

- Vaak worden artifacts uit de GH Actions flow gehaald en in een extern systeem gestoken
- Vaak Docker containers met tags die teruglinken aan workflow runs / commits
- Integratie met Dockerhub of Github container registry
    - Authenticatie a.d.h.v. Github secrets!
- Dockerfile moet aanwezig zijn in repository van de applicatie

Extra integraties: Publishing artifacts



```yaml
steps:
  - name: Check out the repo
    uses: actions/checkout@v5

  - name: Log in to Docker Hub
    uses: docker/login-action@f4ef78c080cd8ba55a85445d5b36e214a81df20a
    with:
      username: ${{ secrets.DOCKER_USERNAME }}
      password: ${{ secrets.DOCKER_PASSWORD }}

  - name: Extract metadata (tags, labels) for Docker
    id: meta
    uses: docker/metadata-action@9ec57ed1fcdbf14dcef7dfbe97b2010124a938b7
    with:
      images: my-docker-hub-namespace/my-docker-hub-repository

  - name: Build and push Docker image
    id: push
    uses: docker/build-push-action@3b5e8027fcad23fda98b2e3ac259d8d67585f671
    with:
      context: .
      file: ./Dockerfile
      push: true
      tags: ${{ steps.meta.outputs.tags }}
      labels: ${{ steps.meta.outputs.labels }}
```

Assignments

Lab 4 – Cloud Deployments

HOGESCHOOL PXL