IT TOPICS 3

Github Actions & Workflows

WAT ZIJN ACTIONS

- Automatiseren van veelvoorkomende zaken
- Automatisch code testen
- Automatisch code builden
- ..

VOORDELEN ACTIONS

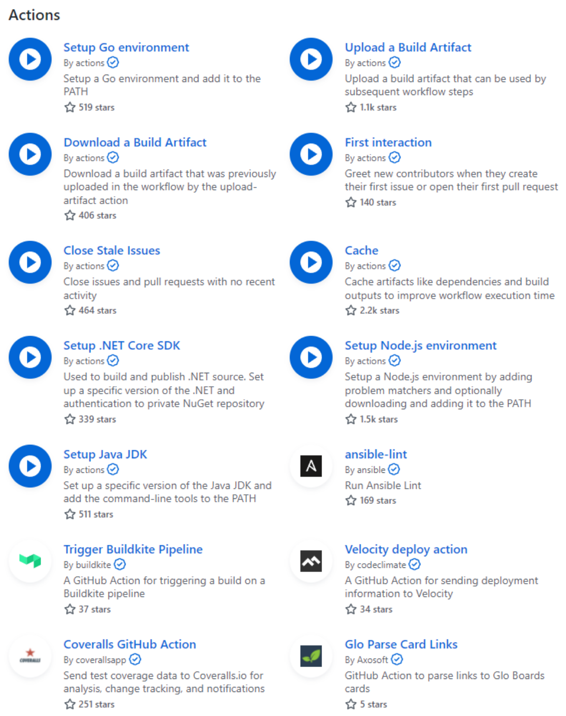
- Volledig geïntegreerd met GitHub
- Werken met elk GitHub event
- Community-powered workflows
- Elk platform, elke taal, elke cloud

FUNCTIONALITEITEN

- Linux, macOS, Windows en containers
- Matrix builds
- Uitgebreide logging
- Ingebouwd secret-management
- Easy to write, easy to share

VOORBEELDEN

- https://github.com/marketplace?type=actions



Actions

Setup Go environment
By actions
Setup a Go environment and add it to the PATH
☆ 519 stars

Upload a Build Artifact
By actions
Upload a build artifact that can be used by subsequent workflow steps
☆ 1.1k stars

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

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

Close Stale Issues
By actions
Close issues and pull requests with no recent activity
☆ 464 stars

Cache
By actions
Cache artifacts like dependencies and build outputs to improve workflow execution time
☆ 2.2k stars

Setup .NET Core SDK
By actions
Used to build and publish .NET source. Set up a specific version of the .NET and authentication to private NuGet repository
☆ 339 stars

Setup Node.js environment
By actions
Setup a Node.js environment by adding problem matchers and optionally downloading and adding it to the PATH
☆ 1.5k stars

Setup Java JDK
By actions
Set up a specific version of the Java JDK and add the command-line tools to the PATH
☆ 511 stars

ansible-lint
By ansible
Run Ansible Lint
☆ 169 stars

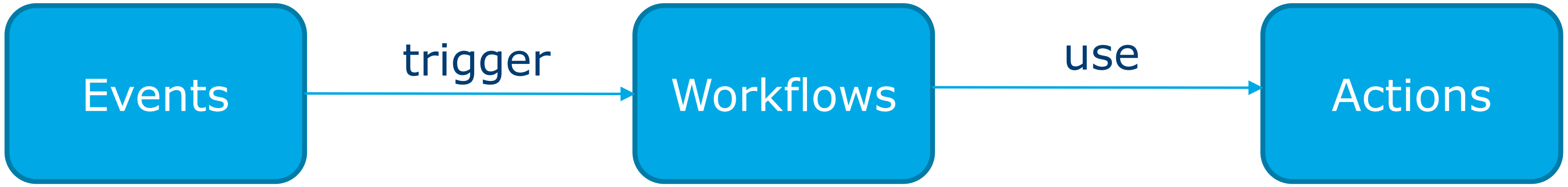
Trigger Buildkite Pipeline
By buildkite
A GitHub Action for triggering a build on a Buildkite pipeline
☆ 37 stars

Velocity deploy action
By codeclimate
A GitHub Action for sending deployment information to Velocity
☆ 34 stars

Coveralls GitHub Action
By coverallsapp
Send test coverage data to Coveralls.io for analysis, change tracking, and notifications
☆ 251 stars

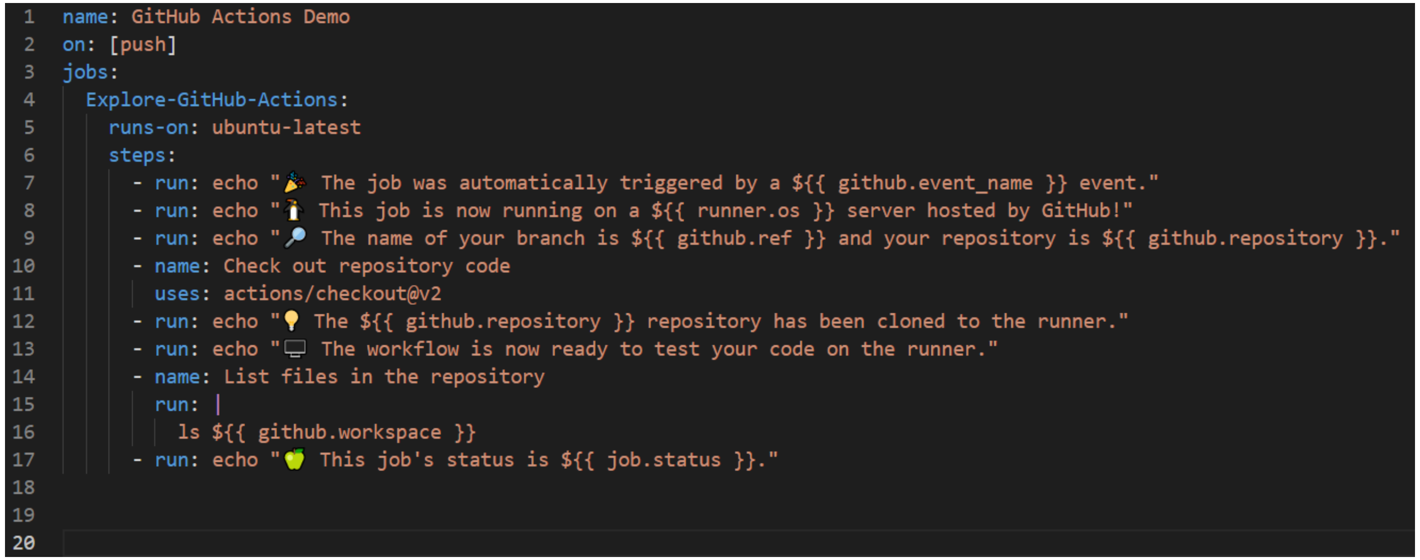
Glo Parse Card Links
By Axosoft
GitHub Action to parse links to Glo Boards cards
☆ 5 stars

COMPONENTEN



COMPONENTEN - EVENTS

- GitHub triggered events
  - Push
  - pull_request
  - Public
- Scheduled events
  - Schedule
- Manuele trigger
  - workflow_dispatch

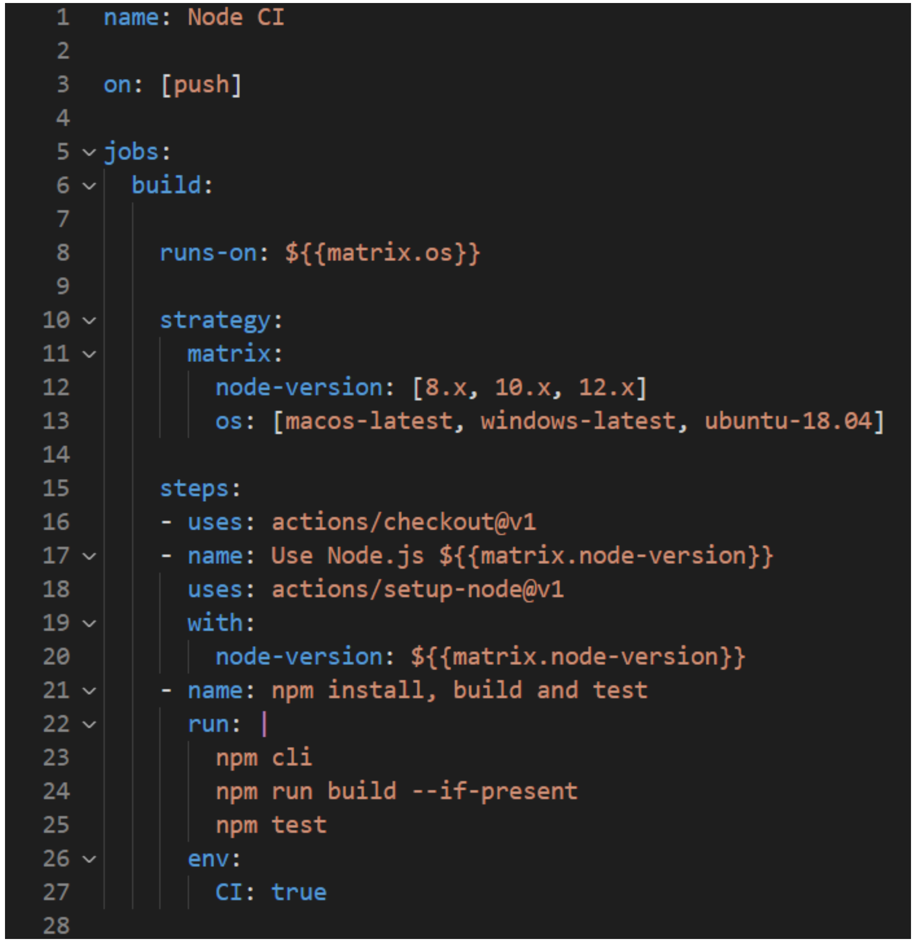


```yaml
name: GitHub Actions Demo
on: [push]
jobs:
  Explore-GitHub-Actions:
    runs-on: ubuntu-latest
    steps:
      - run: echo "🎉 The job was automatically triggered by a ${{ github.event_name }} event."
      - run: echo "🐧 This job is now running on a ${{ runner.os }} server hosted by GitHub!"
      - run: echo "🔎 The name of your branch is ${{ github.ref }} and your repository is ${{ github.repository }}."
      - name: Check out repository code
        uses: actions/checkout@v2
      - run: echo "💡 The ${{ github.repository }} repository has been cloned to the runner."
      - run: echo "🖥 The workflow is now ready to test your code on the runner."
      - name: List files in the repository
        run: |
          ls ${{ github.workspace }}
      - run: echo "🍏 This job's status is ${{ job.status }}."



```

https://docs.github.com/en/actions/using-workflows/events-that-trigger-workflows

COMPONENTEN - WORKFLOWS

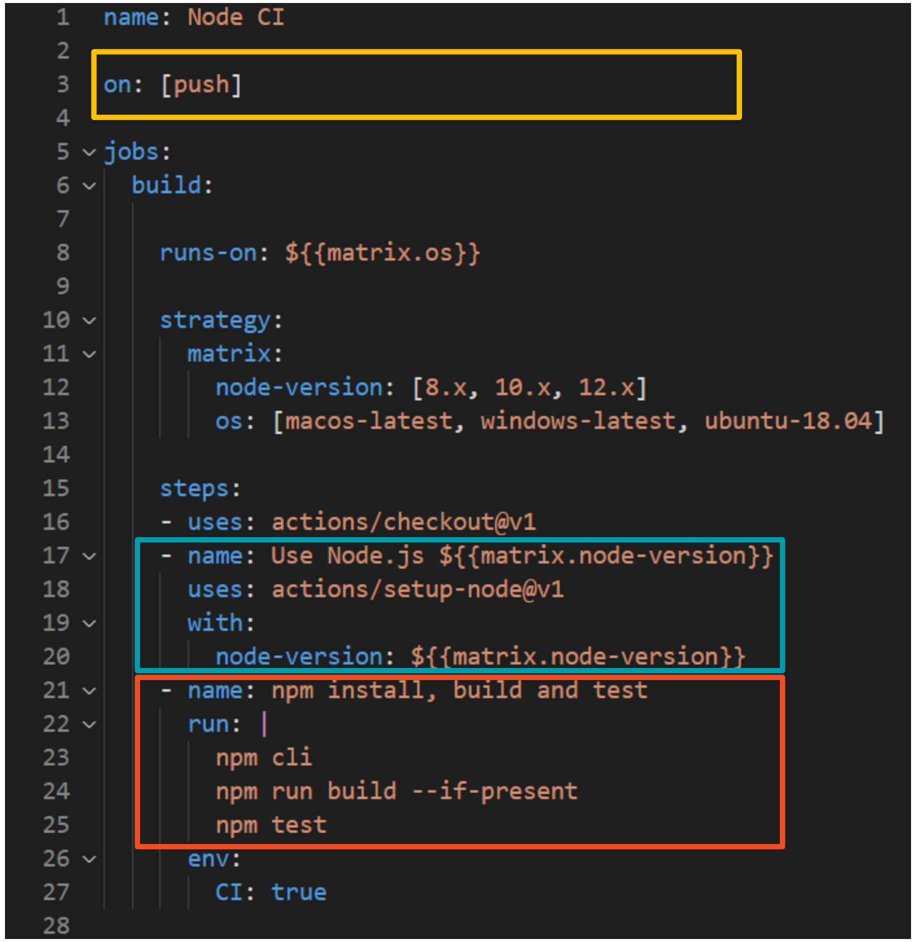
- Pipelines
- Processen omzetten in code
- .yaml syntax
- .github/workflows



```yaml
name: Node CI

on: [push]

jobs:
  build:

    runs-on: ${{matrix.os}}

    strategy:
      matrix:
        node-version: [8.x, 10.x, 12.x]
        os: [macos-latest, windows-latest, ubuntu-18.04]

    steps:
    - uses: actions/checkout@v1
    - name: Use Node.js ${{matrix.node-version}}
      uses: actions/setup-node@v1
      with:
        node-version: ${{matrix.node-version}}
    - name: npm install, build and test
      run: |
        npm cli
        npm run build --if-present
        npm test
      env:
        CI: true

```

COMPONENTEN - WORKFLOWS

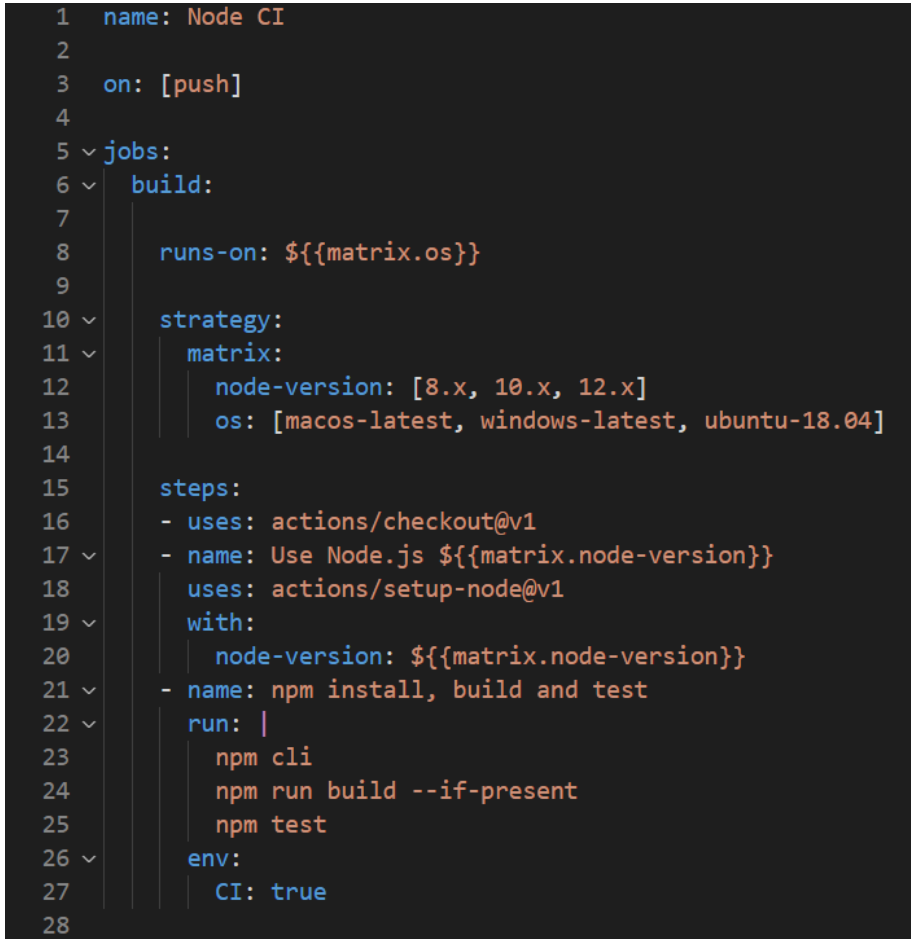
- Workflows hangen alles samen
- Acties in bepaalde volgorde
  - Luisteren voor events
  - Bestaande actions uitvoeren
  - Of shell scripts uitvoeren



```
name: Node CI

on: [push]

jobs:
  build:

    runs-on: ${{matrix.os}}

    strategy:
      matrix:
        node-version: [8.x, 10.x, 12.x]
        os: [macos-latest, windows-latest, ubuntu-18.04]

    steps:
    - uses: actions/checkout@v1
    - name: Use Node.js ${{matrix.node-version}}
      uses: actions/setup-node@v1
      with:
        node-version: ${{matrix.node-version}}
    - name: npm install, build and test
      run: |
        npm cli
        npm run build --if-present
        npm test
      env:
        CI: true
```

COMPONENTEN - WORKFLOWS

- De actions runnen in VM's
  - Linux, Windows of Mac
- Log output
- Artifacts bouwen
- Secret store voor elke repo of organisatie



```yaml
name: Node CI

on: [push]

jobs:
  build:

    runs-on: ${{matrix.os}}

    strategy:
      matrix:
        node-version: [8.x, 10.x, 12.x]
        os: [macos-latest, windows-latest, ubuntu-18.04]

    steps:
    - uses: actions/checkout@v1
    - name: Use Node.js ${{matrix.node-version}}
      uses: actions/setup-node@v1
      with:
        node-version: ${{matrix.node-version}}
    - name: npm install, build and test
      run: |
        npm cli
        npm run build --if-present
        npm test
      env:
        CI: true

```

COMPONENTEN - ACTIONS

	JavaScript action	Container action
Virtual environment	Linux, MacOS, Windows	Linux
Taal	Alles wat compiled naar JS	Elke taal
Snelheid	++	+
Gebruiksvriendelijkheid	++	+

COMPONENTEN - ACTIONS

- https://github.com/marketplace/actions/close-stale-issues

ACTIONS MAKEN

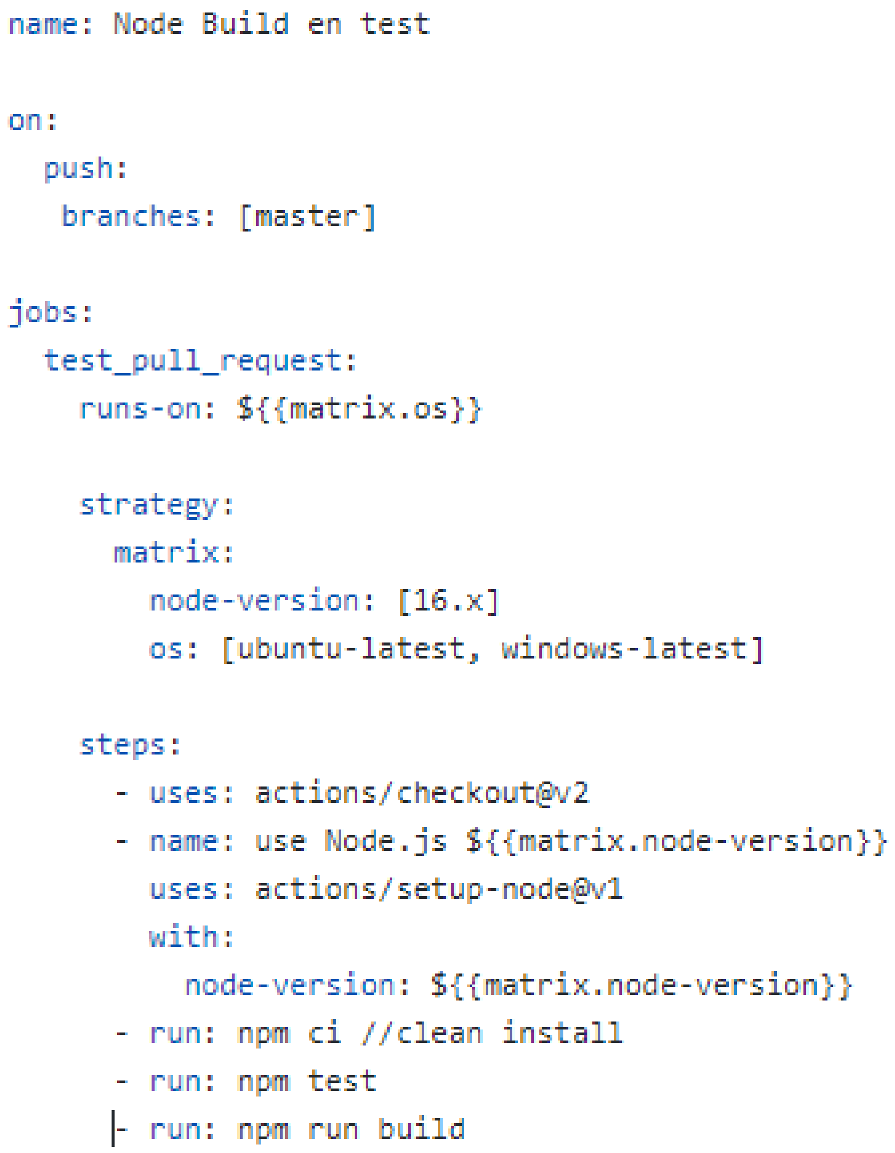
- Manier 1
  - Alles in 1 action
  - Moeilijk onderhoudbaar
  - Één grote brok
- Manier 2
  - Opsplitsen in meerdere actions
  - Onderhoudbare stukjes

ACTIONS EN WORKFLOWS DEMO

Github Actions & Workflows

DEMO

- Nieuwe workflow:
  - Elke push op master / main (afhankelijk wat je gebruikt) gaat api example testen en builden
  - Enkel testen op lunix en windows
  - Node versie 16.x



```
name: Node Build en test

on:
  push:
   branches: [master]

jobs:
  test_pull_request:
    runs-on: ${{matrix.os}}

    strategy:
      matrix:
        node-version: [16.x]
        os: [ubuntu-latest, windows-latest]

    steps:
      - uses: actions/checkout@v2
      - name: use Node.js ${{matrix.node-version}}
        uses: actions/setup-node@v1
        with:
          node-version: ${{matrix.node-version}}
      - run: npm ci //clean install
      - run: npm test
      - run: npm run build
```

DEMO

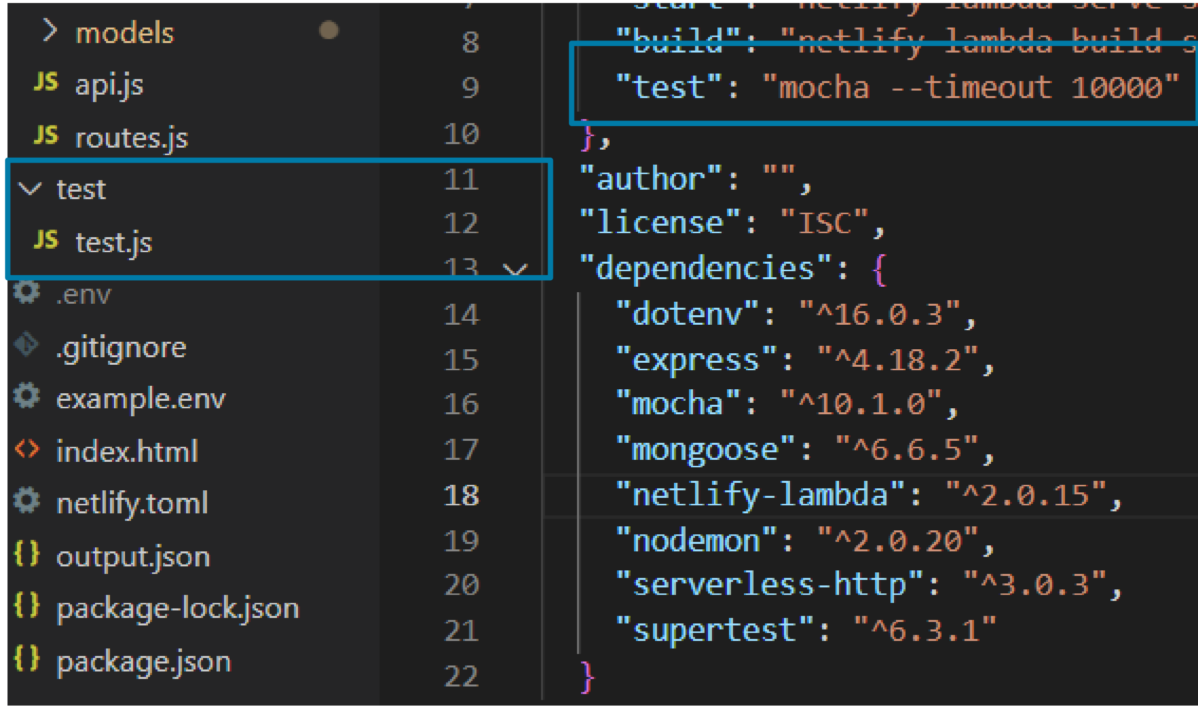
- API_example aanpassen in VS code
  - Map *test* toevoegen met test.js
  - Packages installeren voor de test:

    npm i mocha supertest
  - Package.json aanpassen:

    "test":"mocha –timeout 10000"

Mocha: JS test framework

Supertest: http-test module

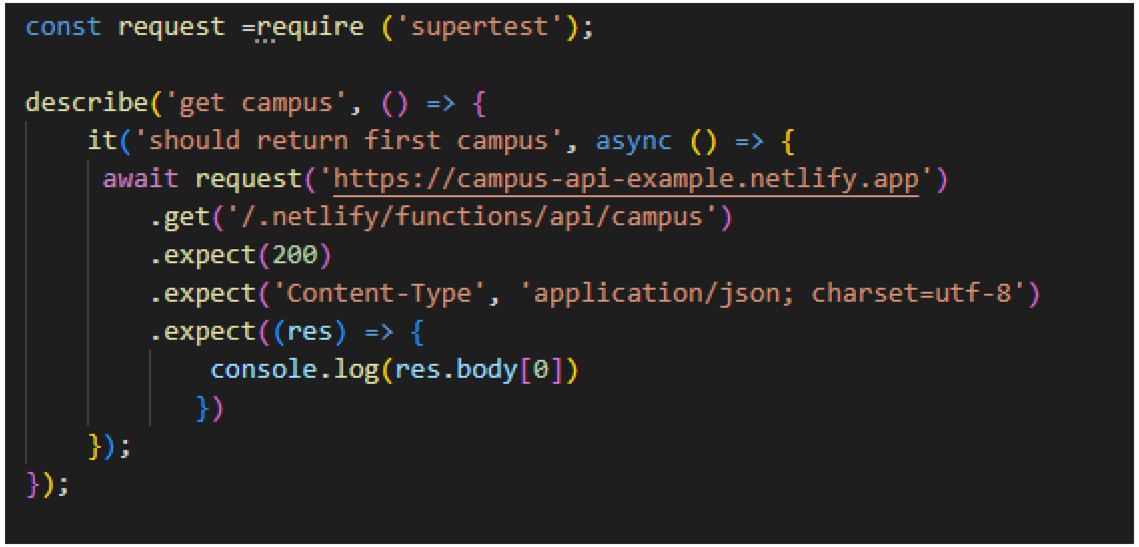


```
> models
JS api.js
JS routes.js
v test
  JS test.js
.env
.gitignore
example.env
index.html
netlify.toml
output.json
package-lock.json
package.json
```

```
    "build": "netlify-lambda build s
    "test": "mocha --timeout 10000"
  },
  "author": "",
  "license": "ISC",
  "dependencies": {
    "dotenv": "^16.0.3",
    "express": "^4.18.2",
    "mocha": "^10.1.0",
    "mongoose": "^6.6.5",
    "netlify-lambda": "^2.0.15",
    "nodemon": "^2.0.20",
    "serverless-http": "^3.0.3",
    "supertest": "^6.3.1"
  }
```

DEMO

- Test.js aanpassen:
  - Supertest toevoegen
  - Mocha
  - Supertest



```
const request =require ('supertest');

describe('get campus', () => {
    it('should return first campus', async () => {
     await request('https://campus-api-example.netlify.app')
        .get('/.netlify/functions/api/campus')
        .expect(200)
        .expect('Content-Type', 'application/json; charset=utf-8')
        .expect((res) => {
            console.log(res.body[0])
           })
    });
});
```

- Pushen naar github
- Check log

TROUBLESHOOTING ACTIONS

- Workflow editor
- Action-debugging
  - ACTIONS_STEP_DEBUG
  - ACTIONS_RUNNER_DEBUG
- VS Code extension
- Lokaal debuggen: nektos/act

ACTIONS BEST PRACTICES

- Versionering
- Documentatie
- Unit testing
- Metadata onderhouden
- Uploaden naar Marketplace

CI

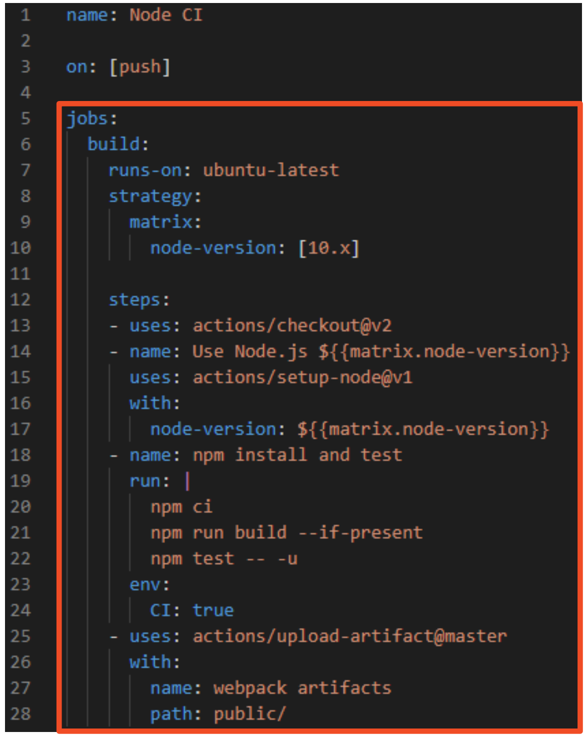
Github Actions & Workflows

WAT IS CI

- Continuous integration
- Code van verschillende developers mergen
- Applicatie builden
- Applicatie testen
- Applicatie bouwen tot een deploybaar pakket

CI WORKFLOW

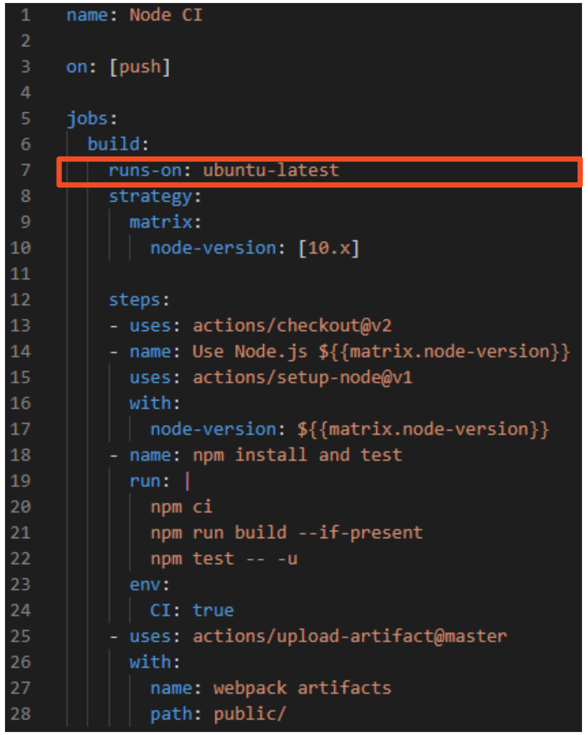
- Single job with 4 steps
- Ubuntu
- Matrix build
- Bestaat uit aparte actions
  - checkout
  - Setup
  - Npm shell script
  - Artifact uploaden



```yaml
name: Node CI

on: [push]

jobs:
  build:
    runs-on: ubuntu-latest
    strategy:
      matrix:
        node-version: [10.x]

    steps:
    - uses: actions/checkout@v2
    - name: Use Node.js ${{matrix.node-version}}
      uses: actions/setup-node@v1
      with:
        node-version: ${{matrix.node-version}}
    - name: npm install and test
      run: |
        npm ci
        npm run build --if-present
        npm test -- -u
      env:
        CI: true
    - uses: actions/upload-artifact@master
      with:
        name: webpack artifacts
        path: public/
```

CI WORKFLOW

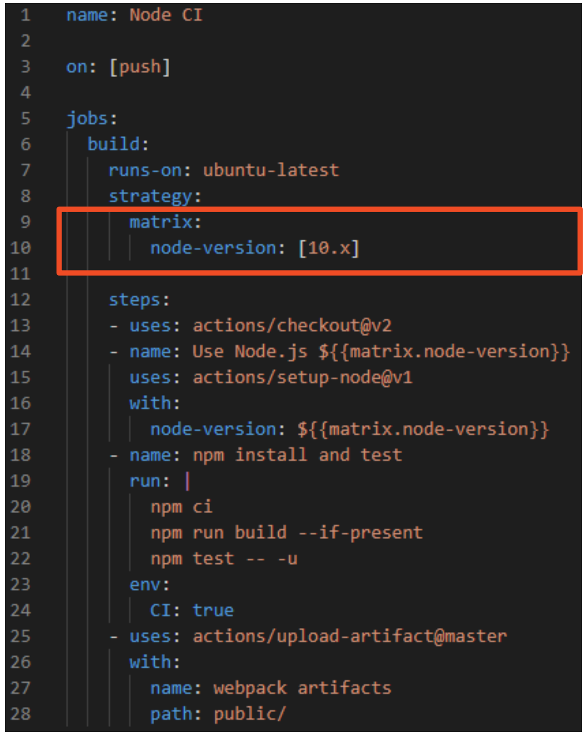
- Single job with 4 steps
- Ubuntu
- Matrix build
- Bestaat uit aparte actions
  - checkout
  - Setup
  - Npm shell script
  - Artifact uploaden



```yaml
name: Node CI

on: [push]

jobs:
  build:
    runs-on: ubuntu-latest
    strategy:
      matrix:
        node-version: [10.x]

    steps:
    - uses: actions/checkout@v2
    - name: Use Node.js ${{matrix.node-version}}
      uses: actions/setup-node@v1
      with:
        node-version: ${{matrix.node-version}}
    - name: npm install and test
      run: |
        npm ci
        npm run build --if-present
        npm test -- -u
      env:
        CI: true
    - uses: actions/upload-artifact@master
      with:
        name: webpack artifacts
        path: public/
```

CI WORKFLOW

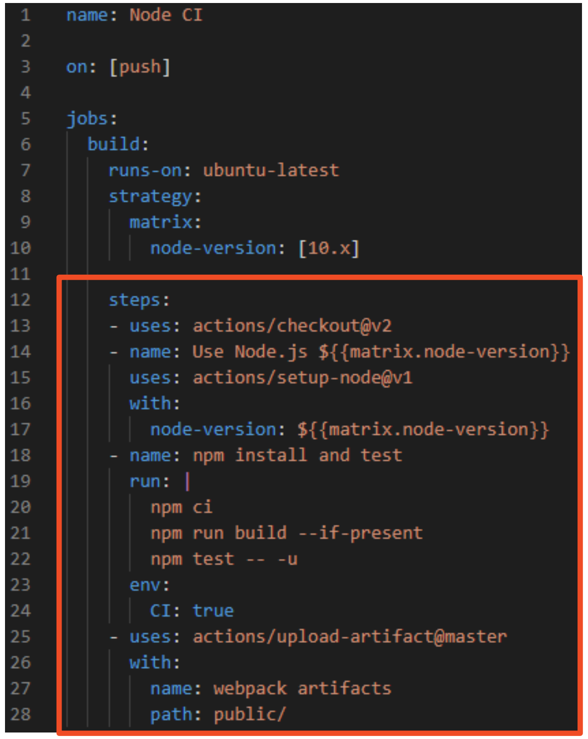
- Single job with 4 steps
- Ubuntu
- Matrix build
- Bestaat uit aparte actions
  - checkout
  - Setup
  - Npm shell script
  - Artifact uploaden



```yaml
name: Node CI

on: [push]

jobs:
  build:
    runs-on: ubuntu-latest
    strategy:
      matrix:
        node-version: [10.x]

    steps:
    - uses: actions/checkout@v2
    - name: Use Node.js ${{matrix.node-version}}
      uses: actions/setup-node@v1
      with:
        node-version: ${{matrix.node-version}}
    - name: npm install and test
      run: |
        npm ci
        npm run build --if-present
        npm test -- -u
      env:
        CI: true
    - uses: actions/upload-artifact@master
      with:
        name: webpack artifacts
        path: public/
```

CI WORKFLOW

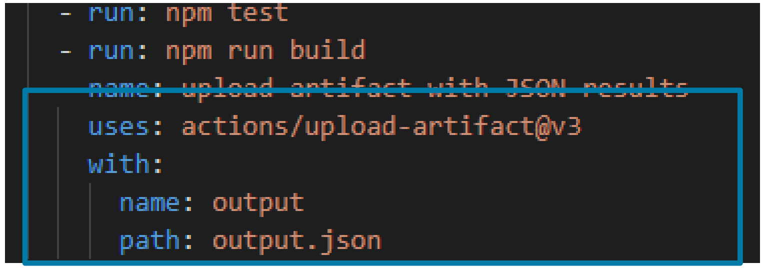
- Single job with 4 steps
- Ubuntu
- Matrix build
- Bestaat uit aparte actions
  - checkout
  - Setup
  - Npm shell script
  - Artifact uploaden



```yaml
name: Node CI

on: [push]

jobs:
  build:
    runs-on: ubuntu-latest
    strategy:
      matrix:
        node-version: [10.x]

    steps:
    - uses: actions/checkout@v2
    - name: Use Node.js ${{matrix.node-version}}
      uses: actions/setup-node@v1
      with:
        node-version: ${{matrix.node-version}}
    - name: npm install and test
      run: |
        npm ci
        npm run build --if-present
        npm test -- -u
      env:
        CI: true
    - uses: actions/upload-artifact@master
      with:
        name: webpack artifacts
        path: public/
```

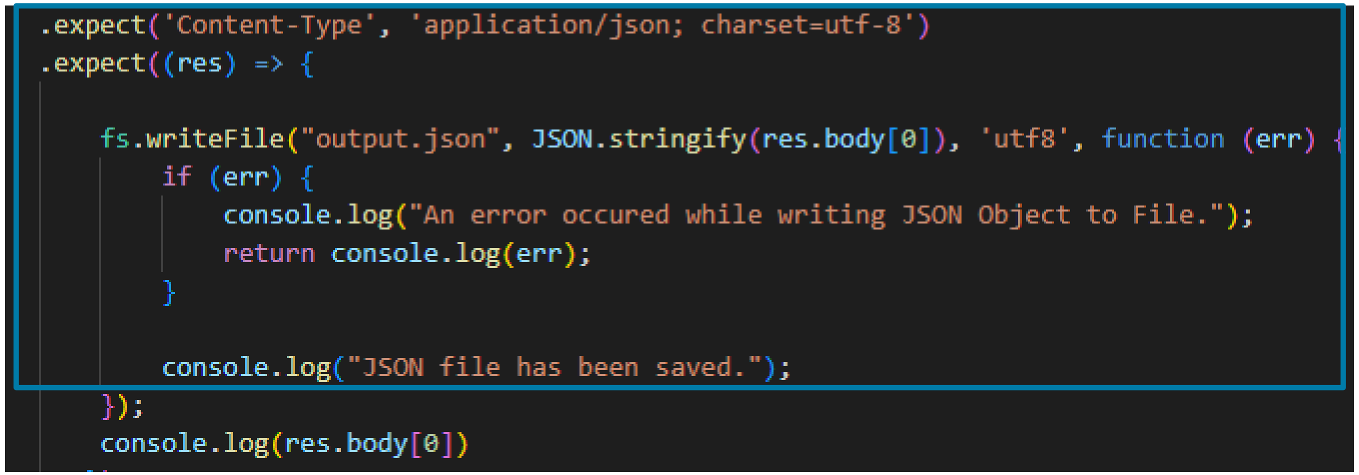
DEMO

- Output van test-actie
- Workflow aanpassen:
  - Artifact uploaden



```
  - run: npm test
  - run: npm run build
    name: upload artifact with JSON results
    uses: actions/upload-artifact@v3
    with:
      name: output
      path: output.json
```

- Test.js aanpassen in VSC



```
.expect('Content-Type', 'application/json; charset=utf-8')
.expect((res) => {

    fs.writeFile("output.json", JSON.stringify(res.body[0]), 'utf8', function (err) {
        if (err) {
            console.log("An error occured while writing JSON Object to File.");
            return console.log(err);
        }

        console.log("JSON file has been saved.");
    });
    console.log(res.body[0])
```

- Pushen naar github
- Log bekijken en artifact downloaden

CD

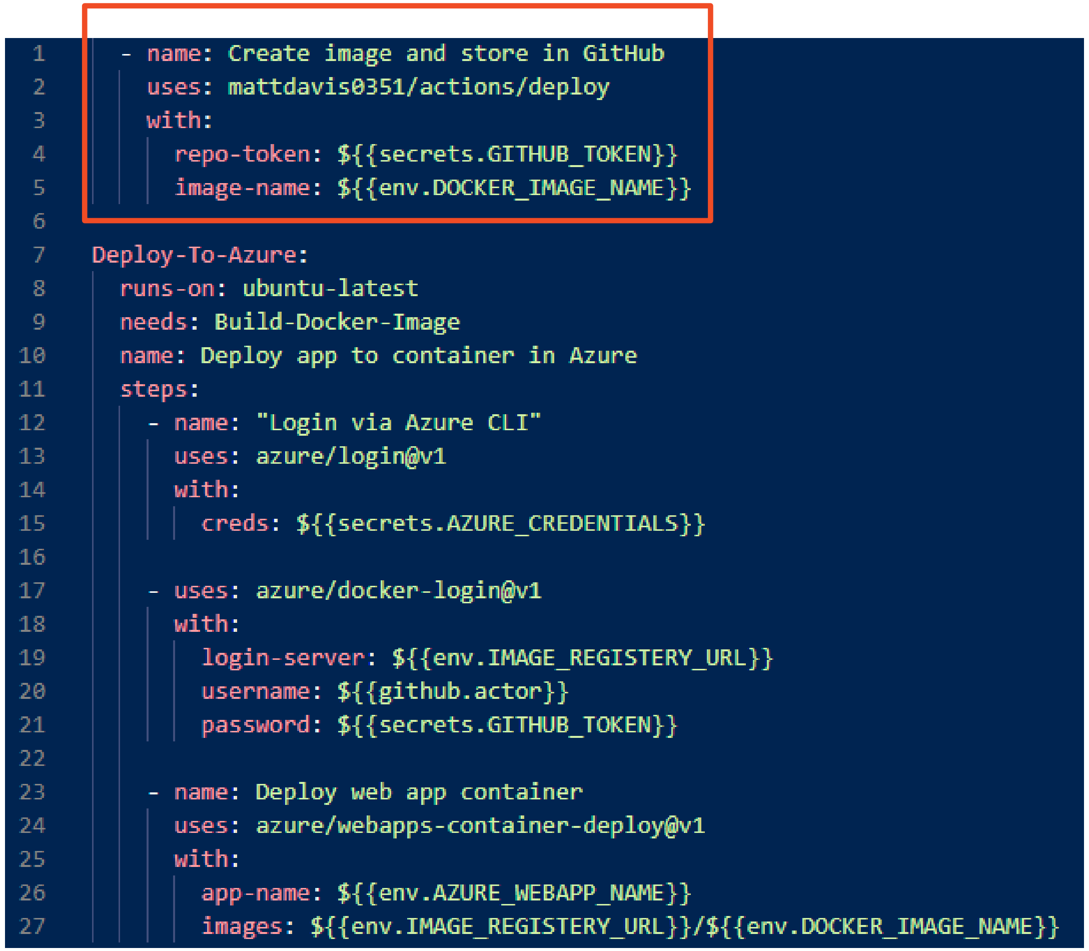
Github Actions & Workflows

WAT IS CD

- Continuous deployment
- Automatisch testen
- Constant online houden van applicatie

CD WORKFLOW

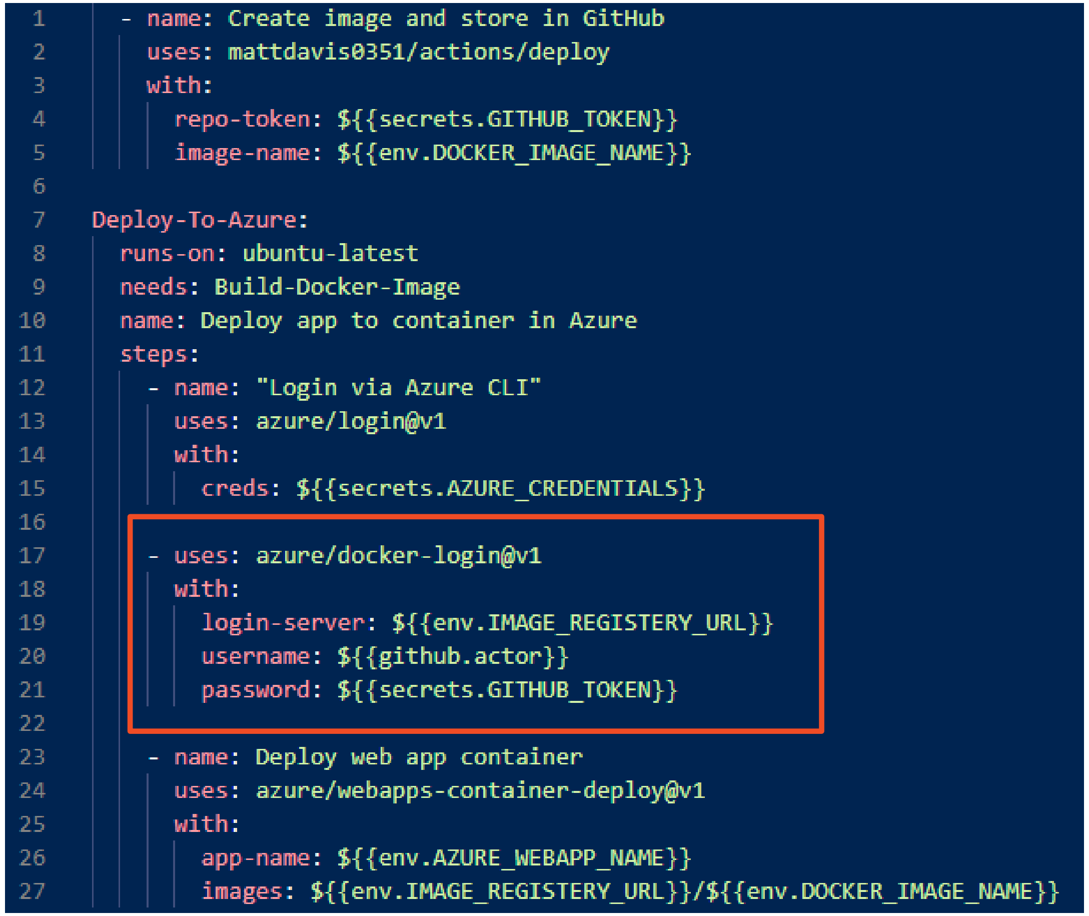
- Image opslaan in GitHub
- Jobs in verschillende envs
- Gebruik maken van Docker image
  - Deployed de container naar Azure



```
  - name: Create image and store in GitHub
    uses: mattdavis0351/actions/deploy
    with:
      repo-token: ${{secrets.GITHUB_TOKEN}}
      image-name: ${{env.DOCKER_IMAGE_NAME}}

Deploy-To-Azure:
  runs-on: ubuntu-latest
  needs: Build-Docker-Image
  name: Deploy app to container in Azure
  steps:
    - name: "Login via Azure CLI"
      uses: azure/login@v1
      with:
        creds: ${{secrets.AZURE_CREDENTIALS}}

    - uses: azure/docker-login@v1
      with:
        login-server: ${{env.IMAGE_REGISTERY_URL}}
        username: ${{github.actor}}
        password: ${{secrets.GITHUB_TOKEN}}

    - name: Deploy web app container
      uses: azure/webapps-container-deploy@v1
      with:
        app-name: ${{env.AZURE_WEBAPP_NAME}}
        images: ${{env.IMAGE_REGISTERY_URL}}/${{env.DOCKER_IMAGE_NAME}}
```

CD WORKFLOW

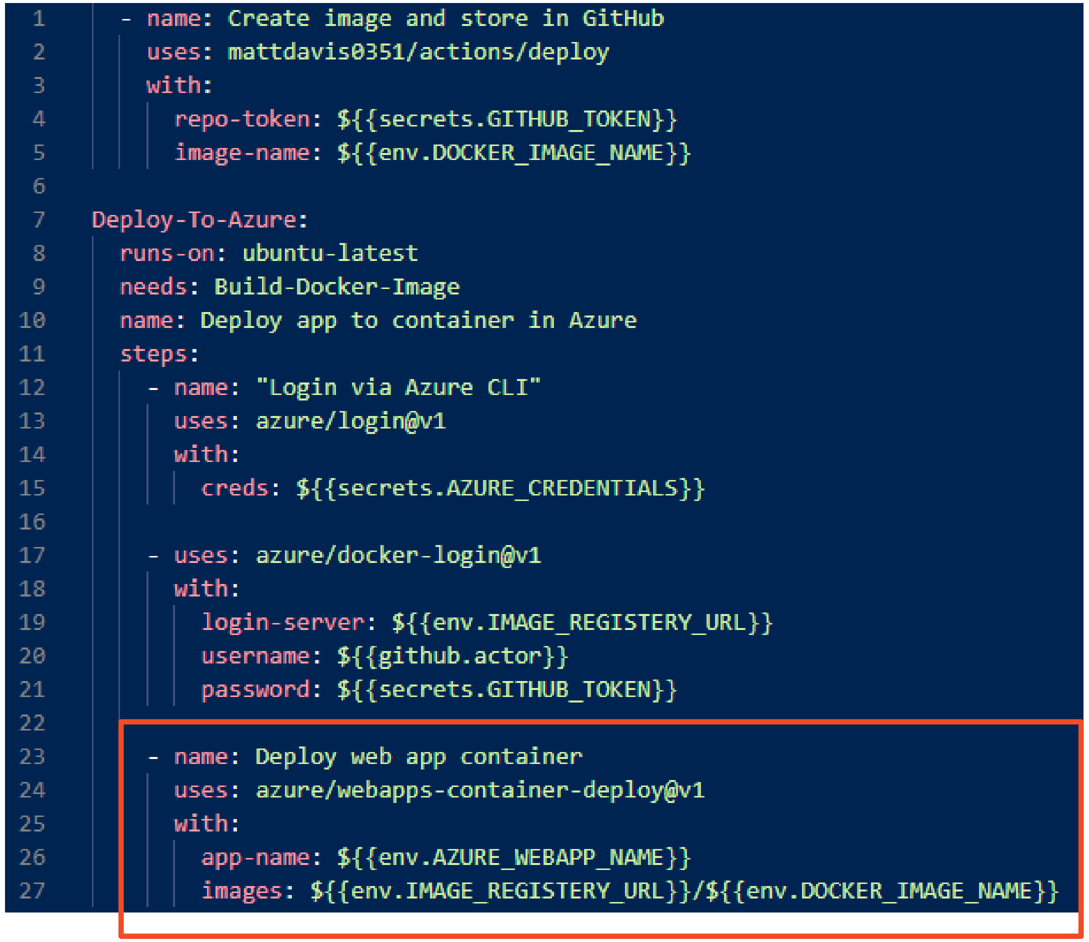
- Image opslaan in GitHub
- Jobs in verschillende envs
- Gebruik maken van Docker image
  - Deployed de container naar Azure



```yaml
  - name: Create image and store in GitHub
    uses: mattdavis0351/actions/deploy
    with:
      repo-token: ${{secrets.GITHUB_TOKEN}}
      image-name: ${{env.DOCKER_IMAGE_NAME}}

Deploy-To-Azure:
  runs-on: ubuntu-latest
  needs: Build-Docker-Image
  name: Deploy app to container in Azure
  steps:
    - name: "Login via Azure CLI"
      uses: azure/login@v1
      with:
        creds: ${{secrets.AZURE_CREDENTIALS}}

    - uses: azure/docker-login@v1
      with:
        login-server: ${{env.IMAGE_REGISTERY_URL}}
        username: ${{github.actor}}
        password: ${{secrets.GITHUB_TOKEN}}

    - name: Deploy web app container
      uses: azure/webapps-container-deploy@v1
      with:
        app-name: ${{env.AZURE_WEBAPP_NAME}}
        images: ${{env.IMAGE_REGISTERY_URL}}/${{env.DOCKER_IMAGE_NAME}}
```

CD WORKFLOW

- Image opslaan in GitHub
- Jobs in verschillende envs
- Gebruik maken van Docker image
  - Deployed de container naar Azure



```yaml
  - name: Create image and store in GitHub
    uses: mattdavis0351/actions/deploy
    with:
      repo-token: ${{secrets.GITHUB_TOKEN}}
      image-name: ${{env.DOCKER_IMAGE_NAME}}

Deploy-To-Azure:
  runs-on: ubuntu-latest
  needs: Build-Docker-Image
  name: Deploy app to container in Azure
  steps:
    - name: "Login via Azure CLI"
      uses: azure/login@v1
      with:
        creds: ${{secrets.AZURE_CREDENTIALS}}

    - uses: azure/docker-login@v1
      with:
        login-server: ${{env.IMAGE_REGISTERY_URL}}
        username: ${{github.actor}}
        password: ${{secrets.GITHUB_TOKEN}}

    - name: Deploy web app container
      uses: azure/webapps-container-deploy@v1
      with:
        app-name: ${{env.AZURE_WEBAPP_NAME}}
        images: ${{env.IMAGE_REGISTERY_URL}}/${{env.DOCKER_IMAGE_NAME}}
```

ENVIRONMENTS

Github Actions & Workflows

ENVIRONMENTS

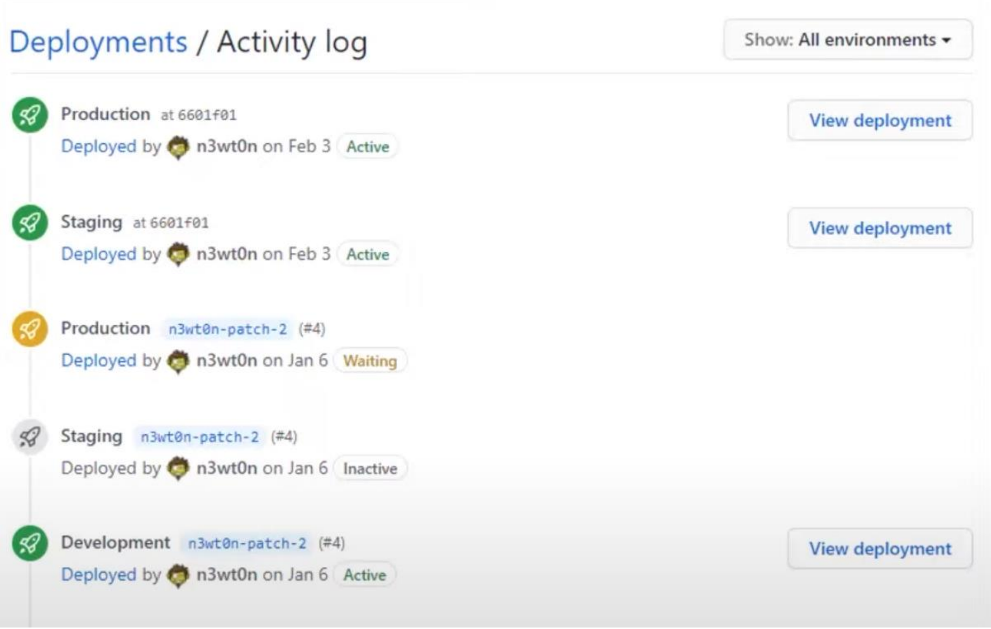
- Logische onderscheiding tussen omgevingen
- Dev – Test – QA – Production
- Ganse omgeving of een deel
- Kunnen eigen secrets hebben
- Hebben protection rules

PROTECTION RULES

- Nodige reviewers
- Bepaalde wachttijd instellen
- Bepaalde branches toelaten
- Mogelijkheid om derde partijen te laten beslissen

DEPLOYMENT LOGS

- Één of meerdere environments
- Volledige geschiedenis van deployments
- Status van deployments



Deployments / Activity log

Show: All environments ▾

Production at 6601f01
Deployed by n3wt0n on Feb 3 Active
View deployment

Staging at 6601f01
Deployed by n3wt0n on Feb 3 Active
View deployment

Production n3wt0n-patch-2 (#4)
Deployed by n3wt0n on Jan 6 Waiting

Staging n3wt0n-patch-2 (#4)
Deployed by n3wt0n on Jan 6 Inactive

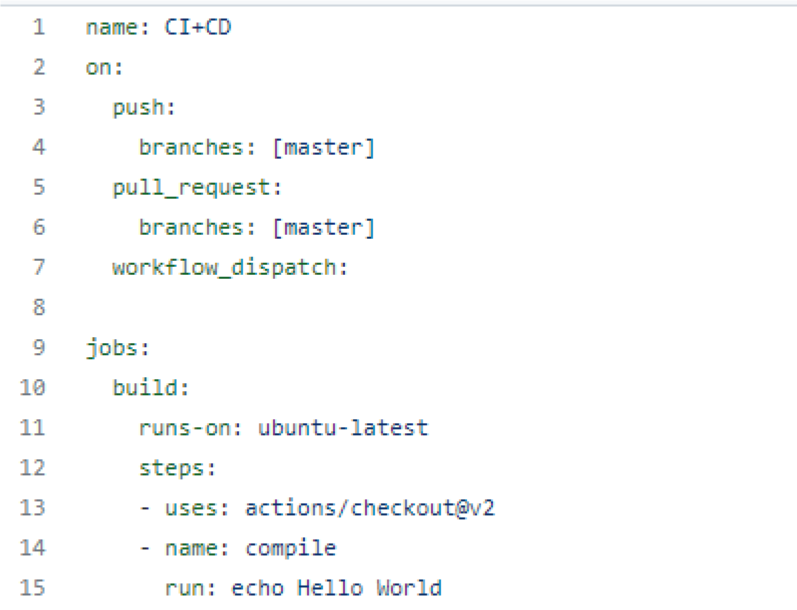
Development n3wt0n-patch-2 (#4)
Deployed by n3wt0n on Jan 6 Active
View deployment

ENVIRONMENTS DEMO

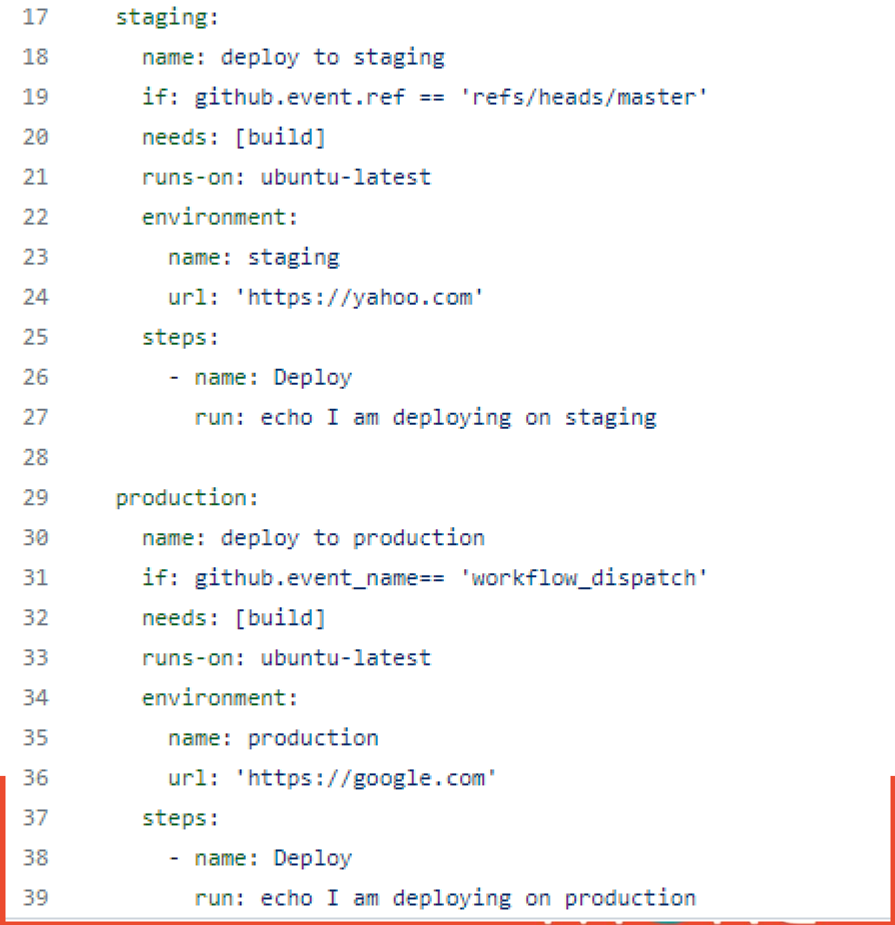
Github Actions & Workflows

DEMO

- Maak staging en production aan als nieuwe environments
- Nieuwe workflow CI/CD.yml



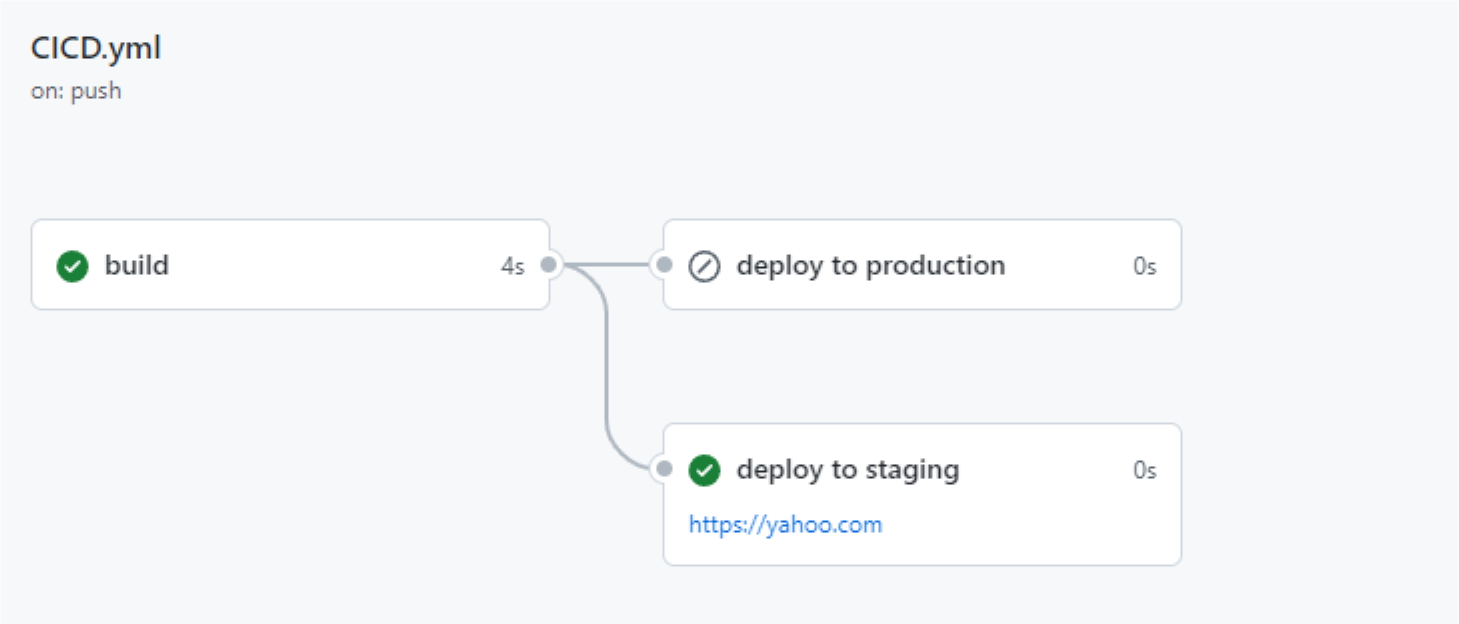
```
name: CI+CD
on:
  push:
    branches: [master]
  pull_request:
    branches: [master]
  workflow_dispatch:

jobs:
  build:
    runs-on: ubuntu-latest
    steps:
    - uses: actions/checkout@v2
    - name: compile
      run: echo Hello World
```



```
  staging:
    name: deploy to staging
    if: github.event.ref == 'refs/heads/master'
    needs: [build]
    runs-on: ubuntu-latest
    environment:
      name: staging
      url: 'https://yahoo.com'
    steps:
      - name: Deploy
        run: echo I am deploying on staging

  production:
    name: deploy to production
    if: github.event_name== 'workflow_dispatch'
    needs: [build]
    runs-on: ubuntu-latest
    environment:
      name: production
      url: 'https://google.com'
    steps:
      - name: Deploy
        run: echo I am deploying on production
```

DEPLOYMENT LOGS

- Resultaat:



CICD.yml
on: push

✓ build 4s

⊘ deploy to production 0s

✓ deploy to staging 0s
https://yahoo.com

RUNNERS

Github Actions & Workflows

WAT IS EEN RUNNER

A runner is a server that has the GitHub Actions runner application installed. You can use a runner hosted by GitHub, or you can host your own. A runner listens for available jobs, runs one job at a time, and reports the progress, logs, and results back to GitHub. GitHub-hosted runners are based on Ubuntu Linux, Microsoft Windows, and macOS, and each job in a workflow runs in a fresh virtual environment. For information on GitHub-hosted runners, see "About GitHub-hosted runners." If you need a different operating system or require a specific hardware configuration, you can host your own runners.

RUNNERS

GitHub hosted

- Automatische updates
- Onderhouden door GitHub
- Clean instance voor elke job
- Duidelijke pricing

https://docs.github.com/en/billing/managing-billing-for-github-actions/about-billing-for-github-actions

Self hosted runners

- Open source
- Meer customisatie
- Eigen verantwoordelijkheid
- GitHub raadt dit af voor publieke repositories

RUNNER GROUPS

- Groeperen van self-hosted runners
- Configureren op Enterprise en/of organisation niveau
- Scope naar specifieke orgs en repos
- Runners kunnen wisselen per groep
- Kan maar tot één groep tegelijk behoren

SECRET MANAGEMENT

Github Actions & Workflows

SECRETS

- Geheime variabelen
- Api keys
- Wachtwoorden
- Logins
- Configuratie

SECRETS

Organization

- Secret management zonder duplicatie
- Worden doorgegeven aan repo's
- Kunnen scoped worden voor specifieke repo's
- Niet beschikbaar met gratis licentie

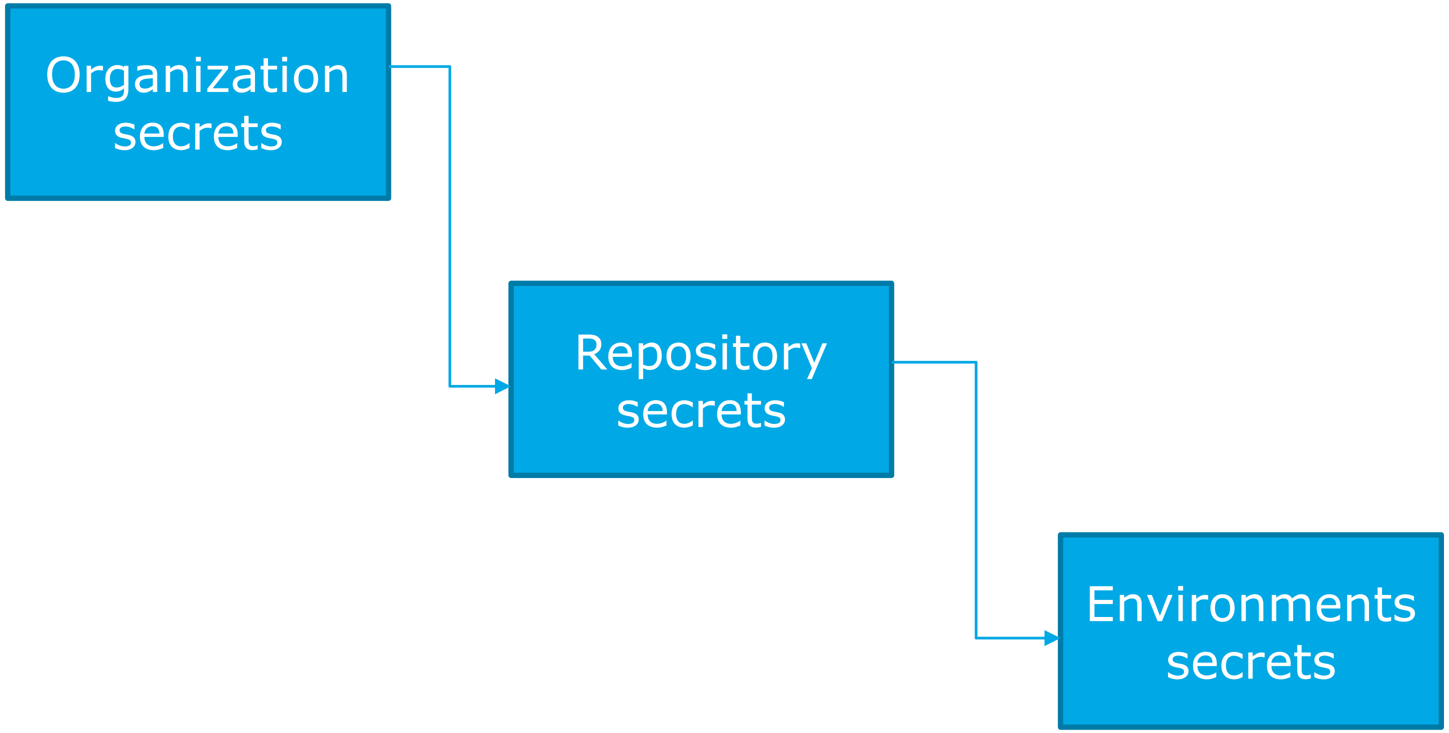
Repository

- Standaard scoped voor een specifieke repo
- Kunnen org secrets overschrijven
- Beschikbaar met gratis licentie

Environments

- Scoped voor een specifiek environment
- Overschrijven org en repo secrets
- Enkel specifieke users met toegang

SECRETS



Organization secrets

Repository secrets

Environments secrets

SECRETS

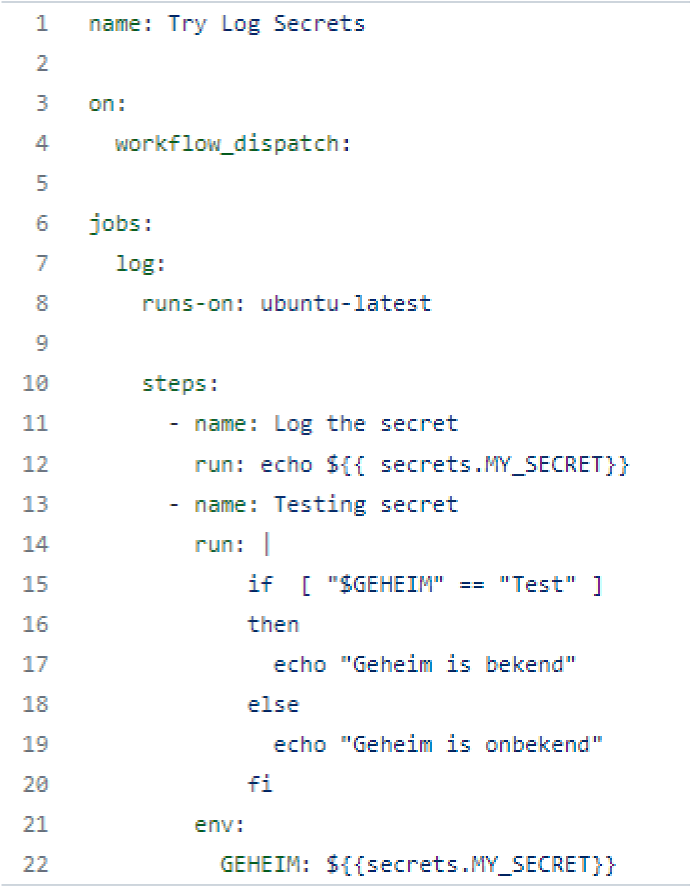
- Beperkingen
  - Kunnen niet gelezen worden in apps
  - Worden niet doorgegeven aan forked repo's
  - Maximum van 100 secrets / workflow
  - Gelimiteerd tot 64kb

SECRET MANAGEMENT DEMO

Github Actions & Workflows

DEMO SECRETS

- Maak een nieuwe secret aan MY_SECRET
- Nieuwe TrySecrets.yml workflow
  - Testen of we de waarde van secret kunnen achterhalen



```
name: Try Log Secrets

on:
  workflow_dispatch:

jobs:
  log:
    runs-on: ubuntu-latest

    steps:
      - name: Log the secret
        run: echo ${{ secrets.MY_SECRET}}
      - name: Testing secret
        run: |
            if  [ "$GEHEIM" == "Test" ]
            then
              echo "Geheim is bekend"
            else
              echo "Geheim is onbekend"
            fi
        env:
          GEHEIM: ${{secrets.MY_SECRET}}
```

OEFENINGEN

Github Actions & Workflows

OEFENINGEN ACTIONS 1

- Maak in VisualStudio een C# console app project met naam GithubActions_Oef1
- Gebruik Git commando’s in de package manager console om het project in een nieuwe github-repo GithubActions_oef1 te uploaden.
- Maak een nieuwe workflow om de applicatie te testen en te builden voor verschillende OS’en telkens er een nieuw pull_request gebeurd.
- Maak een nieuwe branch *testing* met een index.html in de root. Activeer een PR om de testing branch te mergen met de main branch en check de log.

OEFENINGEN ACTIONS 2

- Maak een GitHub repo *GitHubActions_oef2* met een public/index.html webpagina. Typ hierin wat tekst.
- Maak een workflow om deze folder te deployen op github pages. Bestudeer goed de handleiding van de acties.
- Test met username.github.io/GithubActions_oef2/public/

LECTUUR

https://www.redhat.com/en/topics/devops/what-is-ci-cd