Gitlab CI - Praxis

Ziel

- Erste Schritte mit Gitlab (Ci)
- Automatisiertes bauen, testen und deployen einer React App
- Deploy zu Gitlab Pages

Schritte

1. Vorbereitungen

1.1 Gitlab Account erstellen

- Navigiere auf die Website gitlab.com und erstelle einen neuen Account (falls du noch keinen hast)

1.2 Hinterlege einen SSH Key

- Folge diesem Guide
- https://docs.gitlab.com/ee/user/ssh.html

1.3 Repository erstellen

- Klicke auf New Project -> Create blank project
- Vergebe einen Namen
- Clone das Projekt auf deinen Desktop

1.4 Öffne das neue Repo mit deinem Texteditor

```
cd new-repo
vim .
```

2. Erster Pipeline run

2.1 Die gitlab-ci.yml Datei

- Erstelle im Repository Root eine neue .gitlab-ci.yml Datei

```
stages:
  - build
  - test

build-job:
  stage: build
  script:
    - echo "Hello, $GITLAB_USER_LOGIN!"

test-job:
  stage: test
  script:
    - echo "This job tests something"
```

- Wir definieren also 2 stages, welche die Reihenfolge der Jobs definiert
- Außerdem werden zwei Jobs definiert: build-job und test-job
- Wenn zwei jobs in der gleichen stage sind, werden sie parallel ausgeführt
- $GITLAB_USER_LOGIN ist eine vordefinierte Variable, welche von Gitlab zur verfügung gestellt wird (https://docs.gitlab.com/ee/ci/variables/predefined_variables.html)
- Commite und pushe die Änderung

```
git add .
git commit -m "ci: add gitlab-ci"
git push
```

- Navigiere auf Gitlab im Repo nach Build -> Pipelines
- Stelle sicher, dass die Pipeline erfolgreich gestartet ist

3. React app

3.1 Create React App

- Um eine simple React App als Demo zu nutzen, greifen wir auf create-react-app zurück
- ALLE BEFEHLE SOLLEN IM REPO ROOT AUSGEFÜHRT WERDEN

```
npx create-react-app .
```

- Erstelle auch hier wieder einen neune commit

```
git add .
git commit -m "feat: add react app"
```

4. Build und Test Jobs

4.1 Build Job

- Ändere den build-job in der .gitlab-ci.yml Datei folgendermaßen ab

```
build-job:
  image: "node:lts"
  stage: build
  script:
    - npm ci # Install the npm dependencies
    - npm run build # Build the application
```

- Commite und pushe und überprüfe ob der Build erfolgreich ist

```
git add .
git commit -m "ci: add build step"
git push
```

4.2 Test Job

- Ändere den test-job in der .gitlab-ci.yml Datei folgendermaßen ab

```
test-job:
  image: "node:lts"
  stage: test
  script:
    - npm ci # Install the npm dependencies
    - npm run test # Test the application
```

- Commite und pushe und überprüfe ob der Test erfolgreich ist

```
git add .
git commit -m "ci: add test step"
git push
```

5. Deployment

5.1 Artifacts

- Um Dateien von einem Job an den nächsten zu übertragen, brauchen wir artifacts
- https://docs.gitlab.com/ee/ci/jobs/job_artifacts.html
- Wir wollen also den build Ordner (wo die statischen assets der React app: html, js, css drin liegen) aus dem build-job an unseren Deploy job übergeben

```
build-job:
  image: "node:lts"
  stage: build
  script:
    - npm ci # Install the npm dependencies
    - npm run build # Build the application
  artifacts:
    paths:
      - build/
```

5.2 Gitlab Pages

- Mehr infos: https://docs.gitlab.com/ee/user/project/pages/
- Um eine Website zu Gitlab Pages zu deployen, müssen wir:
  - Die statischen assets in eine public Ordner verschieben
  - Einen neune job mit dem Namen pages erzeugen
  - Eine neue deploy stage erzeugen
  - Ein neues Feld muss in der package.json gesetzt werden: homepage
  - Diese Domain folgt dem Schema: https://{GITLAB USER NAME}}.gitlab.io/{{REPO NAME}}
  - Nach dem ersten Pipeline Run, muss unter Deploy -> Pages -> Use unique Domain deaktiviert werden

```
stages:
  - build
  - test
  - deploy

...

pages:
  stage: deploy
  script:
    - rm -rf public
    - cp build/index.html build/404.html
    - mv build public
    - ls -la public
  artifacts:
    paths:
      - public/
```

package.json

```
{
    ...
    "name": "techstarter-ci-tutorial",
    "homepage": "techstarter-ci-tutorial",
    ...
}
```

- Commite und pushe und überprüfe ob die Pipeline durchläuft

```
git add .
git commit -m "ci: add gitlab pages"
git push
```