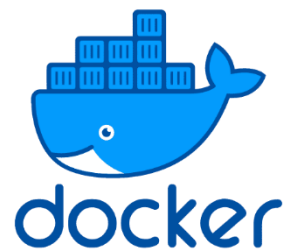




TO-DO-APPLIKATION

Von Jayden Hey



Inhaltsverzeichnis

Inhalt

| | |
|-----------------------------|---|
| Inhaltsverzeichnis | 1 |
| Github-Fork erstellen | 2 |
| Klonen des Repository | 2 |
| Installationen..... | 3 |
| README.md erstellen | 3 |
| Verwendung von Git | 4 |
| Docker-Container | 4 |

Github-Fork erstellen

Zuerst habe ich mich in meinem Githubaccount eingeloggt um dann das vorgegebene Repository «*docker-nodejs-sample*» zu Forken (kopieren) auf mein Repository. Als das erledigt war, habe ich über das Gitbash Terminal und diesem Code: (git clone <https://github.com/Jayden251/docker-nodejs-sample.git> cd docker-nodejs-sample) das Fork auf meinem Laptop geklont.

Klonen des Repository

1. Öffne das Terminal in Gitbash.
2. Navigiere zu dem Verzeichnis, in dem Sie das Projekt speichern möchten:
`cd/c/Dokumentablage/010_ZLI/003_Netzwerkgrundlagen/`
3. Klonen Sie das Repository:
`git clone https://github.com/Jayden251/docker-nodejs-sample.git`
4. Wechseln Sie in das Projektverzeichnis:
`cd docker-nodejs-sample`

```
C:\Dokumentablage\010_ZLI\003_Netzwerkgrundlagen\docker-nodejs-sample> git clone https://github.com/Jayden251/docker-nodejs-sample.git
Cloning into 'docker-nodejs-sample'...
remote: Enumerating objects: 64, done.
remote: Counting objects: 100% (37/37), done.
remote: Compressing objects: 100% (33/33), done.
remote: Total 64 (delta 6), reused 4 (delta 4), pack-reused 27 (from 1)
Receiving objects: 100% (64/64), 1.71 MiB | 4.48 MiB/s, done.
Resolving deltas: 100% (7/7), done.
```

Installationen

1. Git: <https://git-scm.com/>
2. Node.js: <https://nodejs.org>
3. Docker: <https://docs.docker.com/get-docker/>

README.md erstellen

1. Öffne das Terminal in Gitbash.
2. Navigiere zu dem Verzeichnis, in dem Sie das Projekt speichern möchten:
`cd/c/Dokumentablage/010_ZLI/003_Netzwerkgrundlagen/`
3. Erstellen des README's:
`touch README.md`
4. Wechseln Sie in das Projektverzeichnis:
`cd docker-nodejs-sample`

Verwendung von Git

1. Fügen Sie Änderungen zur Staging-Area hinzu:
Git add.
2. Erstellen Sie einen Commit mit einer Beschreibung:
git commit -m 'Nachricht'
3. Pushen Sie Ihre Änderungen zu Ihrem GitHub-Repository:
git push origin main

Docker-Container

1. CMD öffnen
2. Speicher Ort finden:

Cd C:\Dokumentablage\010_ZLI\003_Netzwerkgrundlagen\
docker-nodejs-sample

3. Docker init ausfüllen:
Docker init
4. Fragen:

Node

18.0.0

npm

node src/index.js

3000

```
Let's get started!
Warning - The following Docker files already exist in this directory:
- .dockerignore
- Dockerfile
- compose.yaml
- README.Docker.md
? Do you want to overwrite them? Yes
? What application platform does your project use? Node
? What version of Node do you want to use? 18.0.0
? What version of Node do you want to use? 18.0.0
? Which package manager do you want to use? npm
? What command do you want to use to start the app? [tab for suggestions] node s
? What command do you want to use to start the app? node src/index.js
? What port does your server listen on? 3000
? What port does your server listen on? 3000

Created - .dockerignore
Created - Dockerfile
Created - compose.yaml
Created - README.Docker.md

- Your Docker files are ready!
Review your Docker files and tailor them to your application.
Consult README.Docker.md for information about using the generated files.

Warning - The following files required to run your application were not found.
Create them before running your application:
- package.json
- package-lock.json

What's next?
Start your application by running - docker compose up --build
Your application will be available at http://localhost:3000
```