

Abschlussarbeit

Dokumentation

Donart Aslani

Fertiggestellt am 30.10.2024

Inhaltsverzeichnis

Klonen des Repositories	3
Einrichtung der Entwicklungsumgebung	3
Erstellung der README.md	3
Verwendung von Git (Commit, Push).....	4
Erstellung und Nutzung von Docker-Containern.....	6

Klonen des Repositories

1. Navigieren Sie auf GitHub zur Hauptseite des Repository
2. Klicke oberhalb der Liste der Dateien auf Code.
3. Kopieren sie den SSH URL.
4. Öffnen sie Git Bash.
5. Gib git clone ein, und füge dann die zuvor kopierte URL ein.
7. Drücke die EINGABETASTE, um den lokalen Klon zu erstellen.

```
donar@ThinkPad-Donart MINGW64 ~  
$ cd abschlussarbeit_git  
  
donar@ThinkPad-Donart MINGW64 ~/abschlussarbeit_git  
$ git clone git@github.com:donart-as/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 | 3.55 MiB/s, done.  
Resolving deltas: 100% (7/7), done.  
  
donar@ThinkPad-Donart MINGW64 ~/abschlussarbeit_git  
$ |
```

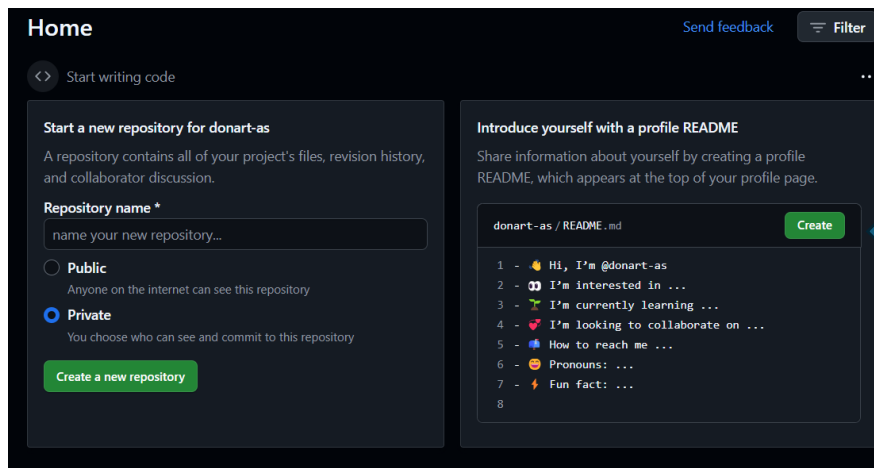
Einrichtung der Entwicklungsumgebung

Um diese Projekte starten zu können müssen wir uns bei GitHub anmelden und Docker Desktop installieren. Ausserdem benötigen wir auch CMD und Git bash, um code auszuführen.

Erstellung der README.md

Variante 1

1. Gehen sie auf GitHub Startseite.
2. Klicken sie auf **Introduce yourself with a profile README**, um einen README.md zu erstellen.



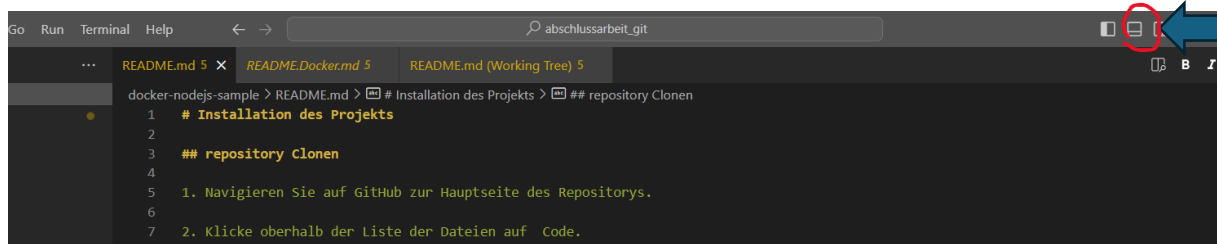
3. Sobald die Nachricht angezeigt wird, kann man codes hinzufügen oder einen clone generieren.

Variante 2

Eine andere Variante ist auch einfach das Dokument Namen auf README.md zu ändern.

Verwendung von Git (Commit, Push)

1. Öffnen sie auf Visual Studio Code den Terminal



2. Geben sie den Ordner ein, an dem sie es gespeichert hatten mit dem Code `cd ...`

```
PS C:\Users\donar\abschlussarbeit_git> cd .\docker-nodejs-sample\
```

3. Um eine Übersicht über die Veränderungen zu haben müssen sie **git Status** eingeben.

```

PS C:\Users\donar\abschlussarbeit_git\docker-nodejs-sample> git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   README.md

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .dockerignore
        Dockerfile
        README.Docker.md
        code-button.webp
        "entertastedr\303\274cken.png"
        gitcloneeingeben.png
        linkkopieren.png
        websitelocalhost3000.png

no changes added to commit (use "git add" and/or "git commit -a")

```

4. Tippen sie jetzt den Code git add.

5. Kurz vor dem Schluss müssen sie den Code git commit -m «Beschreibung»

```

PS C:\Users\donar\abschlussarbeit_git\docker-nodejs-sample> git commit -m "Readme.md hochladen"
[main a077862] Readme.md hochladen
9 files changed, 147 insertions(+), 48 deletions(-)
create mode 100644 .dockerignore
create mode 100644 Dockerfile
create mode 100644 README.Docker.md
create mode 100644 code-button.webp
create mode 100644 "entertastedr\303\274cken.png"
create mode 100644 gitcloneeingeben.png
create mode 100644 linkkopieren.png
create mode 100644 websitelocalhost3000.png

```

6. Zum Schluss muss man mit dem Code git Push tippen, um den ganzen Vorgang zu hochladen und Github neustarten.

```

PS C:\Users\donar\abschlussarbeit_git\docker-nodejs-sample> git push
Enumerating objects: 13, done.
Counting objects: 100% (13/13), done.
Delta compression using up to 22 threads
Compressing objects: 100% (11/11), done.
Writing objects: 100% (11/11), 216.58 KiB | 2.00 MiB/s, done.
Total 11 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:donart-as/docker-nodejs-sample.git
   2a77b9d..a077862  main -> main

```

spec	first commit	last year
src	Update postgres.js	last year
.dockerignore	Readme.md hochladen	54 minutes ago
.gitignore	first commit	last year
Dockerfile	Readme.md hochladen	54 minutes ago
README.Docker.md	Readme.md hochladen	54 minutes ago
README.md	Readme.md hochladen	54 minutes ago
code-button.webp	Readme.md hochladen	54 minutes ago
entertastedr\303\274cken.png	Readme.md hochladen	54 minutes ago
gitcloneeingeben.png	Readme.md hochladen	54 minutes ago
linkkopieren.png	Readme.md hochladen	54 minutes ago
package-lock.json	first commit	last year
package.json	first commit	last year
websitelocalhost3000.png	Readme.md hochladen	54 minutes ago

Erstellung und Nutzung von Docker-Containern

1. Zuerst müssen sie die Anweisungen der docs.docker.com befolgen.
2. Öffnen sie die kommandozeile.
3. Gehen sie in den entsprechenden Ordner hinein mit dem Code `cd ./datei`.

```
C:\Users\donar\abschlussarbeit_git>cd docker-nodejs-sample
```

4. Als nächstes tippen sie `docker init`, um diese Fragen zu beantworten.

```
C:\Users\donar\abschlussarbeit_git\docker-nodejs-sample>docker init

Welcome to the Docker Init CLI!

This utility will walk you through creating the following files with sensible defaults for your project:
- .dockerignore
- Dockerfile
- compose.yaml
- README.Docker.md

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 index.js) node src/index.js

? 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

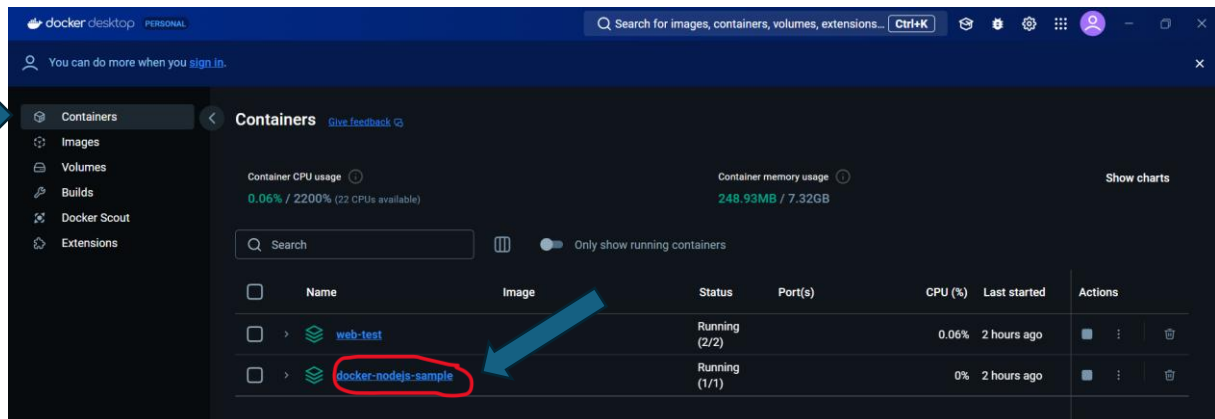
→ Your Docker files are ready!
Review your Docker files and tailor them to your application.
Consult README.Docker.md for information about using the generated files.

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

5. Sobald sie diese Fragen beantwortet haben können sie `docker compose up --build` eingeben, damit man es danach auf Docker Desktop anzeigen kann.

```
C:\Users\donar\abschlussarbeit_git\docker-nodejs-sample>docker compose up --build
[+] Building 9.8s (12/12) FINISHED
=> [server internal] load build definition from Dockerfile
=> => transferring dockerfile: 1.21kB
=> [server] resolve image config for docker-image://docker.io/docker/dockerfile:1
=> CACHED [server] docker-image://docker.io/docker/dockerfile:1@sha256:865e5dd094beca432e8c0ald5e1c465db5f998dca4e439981029b3b81fb39ed5
=> [server internal] load metadata for docker.io/library/node:18.0.0-alpine
=> [server internal] load .dockerignore
=> => transferring context: 663B
=> [server stage-0 1/4] FROM docker.io/library/node:18.0.0-alpine@sha256:469ee26d9e00547ea91202a34ff2542f984c2c60a2edbb4007558ccb76b56df2
=> [server internal] load build context
=> => transferring context: 3.50kB
=> CACHED [server stage-0 2/4] WORKDIR /usr/src/app
=> [server stage-0 3/4] RUN --mount=type=bind,source=package.json,target=package.json --mount=type=bind,source=package-lock.json,target=package-
=> [server stage-0 4/4] COPY . .
=> [server] exporting to image
=> => exporting layers
=> => writing image sha256:017b9f6fc08d34a444181b9c36df188aa5fad89b5f3ac98f59f5495399c1fa93
=> => naming to docker.io/library/docker-nodejs-sample-server
=> [server] resolving provenance for metadata file
[+] Running 2/2
✔ Network docker-nodejs-sample_default Created
✔ Container docker-nodejs-sample-server-1 Created
Attaching to server-1
server-1 | Using sqlite database at /tmp/todo.db
server-1 | Listening on port 3000
server-1 exited with code 0
```

Um zu schauen, ob es angezeigt wird, können sie in Docker Desktop auf Container drücken.



6. Sobald man den Container gestartet hat, kann man die entsprechende Website Angabe auf Google Chrome eingeben, damit es erscheint. Ohne das Starten des Containers funktioniert auch nicht die Website.

