**Dokumentation Docker**

Inhalt

[1.Repo klonen und Fork erstellen 3](#_Toc181368438)

[1.1Fork erstellen 3](#_Toc181368439)

[1.2 Klonen 3](#_Toc181368440)

[2.Entwicklungsumgebung erstellen 3](#_Toc181368441)

[2.1 3](#_Toc181368442)

[3.Readme.md erstellen 3](#_Toc181368443)

[4.Verwendung von Git (Commit, Push) 4](#_Toc181368444)

[4.1 Git "commands" 4](#_Toc181368445)

[Git Commit/Push 4](#_Toc181368446)

# 1.Repo klonen und Fork erstellen

## 1.1Fork erstellen

Eine Fork erstellt man ganz einfach. Im [docker-nodejs-sample](https://github.com/ICT-BLJ/docker-nodejs-sample) von Reto drückt man auf Fork und gibt ein paar Dinge wie zu Beispiel den Namen an und schon ist man Fertig.

Ein Bild, das Text, Screenshot, Software, Multimedia-Software enthält.

Automatisch generierte Beschreibung

## 1.2 Klonen

Um die Repo zu klonen gibt man Lediglich diesen Befehl ein: «git clone <https://github.com/habasch44/docker-nodejs-sample.git>» und dann hat man eine Lokale Kopie

Ein Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung

# 2.Entwicklungsumgebung erstellen

## 2.1

Wir arbeiten hauptsächlich mit GitHub, Visual Studio und Docker.

# 3.Readme.md erstellen

Mithilfe von diesen vier Befehlen konnte ich ein Markdown Readme Erstellen.

touch readme.txt

git add readme.txt

mv README.txt README.md

git commit -m ""

# 4.Verwendung von Git (Commit, Push)

## 4.1 Git "commands"

git add .

git commit -m "initial commit" (anfangs)

git commit -m "KW$$"

git push

git checkout -b bugfix

git commit

git checkout main

git commit

objective

git merge bugfix

git checkout bugfix

git merge main

## 4.2 Git Commit/Push

Git commit erfasst du einen Snapshot der aktuell bereitgestellten Änderungen des Projekts

Mit git Push kann man einen lokalen Branch in ein anderes Repository verschieben, was eine bequeme Methode zur Veröffentlichung von Beiträgen ist.

# 5.Erstellung und Nutzung von Docker-Containern

## 5.1 Erstellung von Docker-Containern

Dockerfile erstellen (falls es noch nicht vorhanden ist): Dieses enthält alle Befehle, um das Projekt in einem Container auszuführen.

Image erstellen: docker build -t <name> .

Container starten:

docker run -d -p 3000:3000 <name>

Docker-Compose verwenden: Um mehrere Container zu orchestrieren, erstellen Sie eine `docker-compose.yml` und starten alle Container mit:

docker-compose up --build