# Projekt Anleitung



Inhalt

[Projekt Anleitung 1](#_Toc181366608)

[1. Repository klonen 2](#_Toc181366609)

[2. Entwicklungsumgebung einrichten 2](#_Toc181366610)

[3. README.md erstellen 2](#_Toc181366611)

[4. Git verwenden (Commit und Push) 3](#_Toc181366612)

[5. Docker-Container erstellen und verwenden 3](#_Toc181366613)

# 1. Repository klonen

Um das Repository zu kopieren, geben Sie den folgenden Befehl ein und ersetzen Sie <SSH-Schlüssel> mit dem SSH-Link des gewünschten Repositories:

git clone <SSH-Schlüssel>

Sollten Sie noch keinen SSH-Schlüssel besitzen, können Sie diesen in den GitHub-Einstellungen erstellen und dann Ihrem Konto hinzufügen.

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

Automatisch generierte Beschreibung

Ein Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung

# 2. Entwicklungsumgebung einrichten

Nach dem Klonen des Projekts sollten Sie die folgenden Schritte zur Einrichtung der Entwicklungsumgebung durchführen:

Öffnen Sie das Verzeichnis des geklonten Projekts mit:

cd <repository-ordner>

Installieren Sie die benötigten Pakete (zum Beispiel für eine Node.js-Umgebung) mit:

npm install

Falls das Projekt Umgebungsvariablen benötigt, legen Sie optional eine. env-Datei an.

# 3. README.md erstellen

Die Datei README.md dient als Dokumentation für das Projekt und sollte üblicherweise folgende Inhalte enthalten:

Projektbeschreibung: Eine kurze Zusammenfassung des Projekts.

Installationsanweisungen: Beschreibung zum Klonen und Installieren der notwendigen Pakete.

Nutzungsanweisungen: Erläutert, wie das Projekt gestartet und verwendet wird.

Beispiel:

markdown

# Projekt Setup

## Installation

```bash

npm install

Beitragen: Hinweise für Entwickler, wie sie zum Projekt beitragen können.

# 4. Git verwenden (Commit und Push)

Befolgen Sie diese Schritte, um Änderungen zu committen und hochzuladen:

Fügen Sie die Änderungen zur Staging-Area hinzu:

git add.

Erstellen Sie einen Commit, um die Änderungen zu speichern:

git commit -m "Beschreibung der Änderungen"

Laden Sie die Änderungen ins Repository hoch:

git push origin main

Hinweis: Wenn Sie einen anderen Branch verwenden, ersetzen Sie main durch den Namen des gewünschten Branches.

# 5. Docker-Container erstellen und verwenden

Mit Docker können Sie Anwendungen in Containern isoliert bereitstellen. Die wesentlichen Schritte dazu sind:

Erstellen Sie, falls noch nicht vorhanden, eine Dockerfile, die alle Schritte für die Containerisierung enthält.

Bauen Sie das Docker-Image:

docker build -t <image-name**>.**

Starten Sie den Container:

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

Dieser Befehl führt den Container im Hintergrund aus (-d) und leitet Port 3000 des Containers auf Port 3000 des Hosts weiter.

Um mehrere Container gleichzeitig zu verwalten, erstellen Sie eine docker-compose.yml und starten alle Container mit:

docker-compose up --build