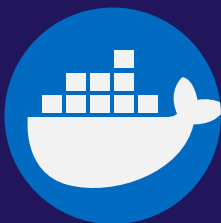




Docker Grundlagen Kursheft

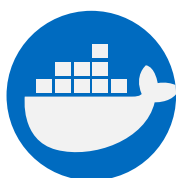
Kapitel: Docker Compose Projekt



DATAMICS
machine intelligence consulting services

Inhaltsverzeichnis

Inhaltsverzeichnis	1
1. Nodejs Anwendung	2
2. MongoDB mit Docker Compose	3
3. Starte die Docker Anwendung	4



1. Nodejs Anwendung

Erstelle eine leere Dockerfile Datei:

```
touch Dockerfile
```

Füge den folgenden Code in die Datei ein:

```
FROM node:8

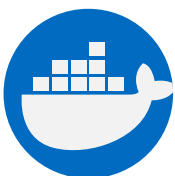
# Erstellt ein "app" Verzeichnis
WORKDIR /usr/src/app

# Installiert die Anwendungspakete
COPY package*.json ./
RUN npm install

# Kopiert den Quellcode
COPY . .

#Zeigt den Port an und startet die Anwendung
EXPOSE 8080
CMD [ "npm", "start" ]
```

Du kannst jetzt ganz einfach deine Anwendung mit dem folgenden Befehl starten:



```
docker build -t mongo-app .
```

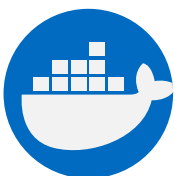
2. MongoDB mit Docker Compose

Wie du schon zuvor gelernt hast, startet das nur unsere nodejs Anwendung ohne MongoDB. Daher erstellen wir jetzt eine `docker-compose` Datei, welche die Anwendung zusammen mit MongoDB startet:

```
touch docker-compose.yml
```

Füge den folgenden Code in der Datei ein:

```
version: "3"
services:
  web:
    build: .
    ports:
      — "8080:8080"
    depends_on:
      — mongo
  mongo:
    image: mongo
    ports:
      — "27017:27017"
```



Jetzt hast du 2 Services in einer Datei definiert. Ein Service startet die nodejs Anwendung in einem Container mit dem Port 8080 und der andere Service startet einen Container mit einer MongoDB Instanz.

3. Starte die Docker Anwendung

Starte jetzt den Service:

```
docker-compose up --build
```

Öffne einen Browser und gib die folgende Links ein, um deine Docker Anwendung anzuschauen:

- <http://localhost:8080/users>
- <http://localhost:8080/user-create>

Hinweis: Schaue dir die Video Lektionen an, um die Browsereingaben zu sehen.

Beende deine Anwendung wieder mit dem folgenden Befehl:

```
docker-compose down
```

