Ü101 - Container 2

Martin | 22.09.2023

Aufgabe 1
Im Repository unter https://github.com/martindubb/docker-task.git findet ihr eine einfache NodeJS App. Beschreibe kurz, was die App macht.

Die App, basierend auf dem bereitgestellten Code, ist eine einfache Webanwendung, die mit NodeJS und dem Express-Framework erstellt wurde. Hier ist eine kurze Beschreibung dessen, was die App macht:

Initialisierung und Konfiguration:

Die App verwendet die Module fs (zum Arbeiten mit Dateisystemen), express (ein Web-Framework für NodeJS) und body-parser (zum Parsen von eingehenden Anforderungsdaten).
Ein Logfile wird erstellt oder verwendet, wenn bereits vorhanden. Der Standardname des Logfiles ist "mylogfile.log", aber es kann durch eine Umgebungsvariable (LOGFILE) geändert werden. Beim Starten des Logfiles wird "Logfile started" in die Datei geschrieben.
Die App verwendet den body-parser-Middleware, um URL-codierte Daten zu parsen.
GET-Route /:

Wenn ein Benutzer die Hauptseite der App besucht, wird eine HTML-Seite angezeigt, die einen Abschnitt mit dem Titel "Testeingabe" und einen darunter liegenden Text (standardmäßig "Beispieltext") enthält.
Darunter befindet sich ein Formular, mit dem der Benutzer einen Text eingeben kann. Nach dem Absenden des Formulars wird die Eingabe an die POST-Route /Eingabe gesendet.
POST-Route /Eingabe:

Wenn das Formular abgesendet wird, nimmt diese Route die Eingabe des Benutzers entgegen.
Die Eingabe wird sowohl in der Konsole als auch im Logfile protokolliert.
Der zuvor angezeigte Text (unter "Testeingabe") wird durch die neue Eingabe des Benutzers ersetzt.

Nach dem Protokollieren der Eingabe wird der Benutzer zur Hauptseite umgeleitet.
Server:

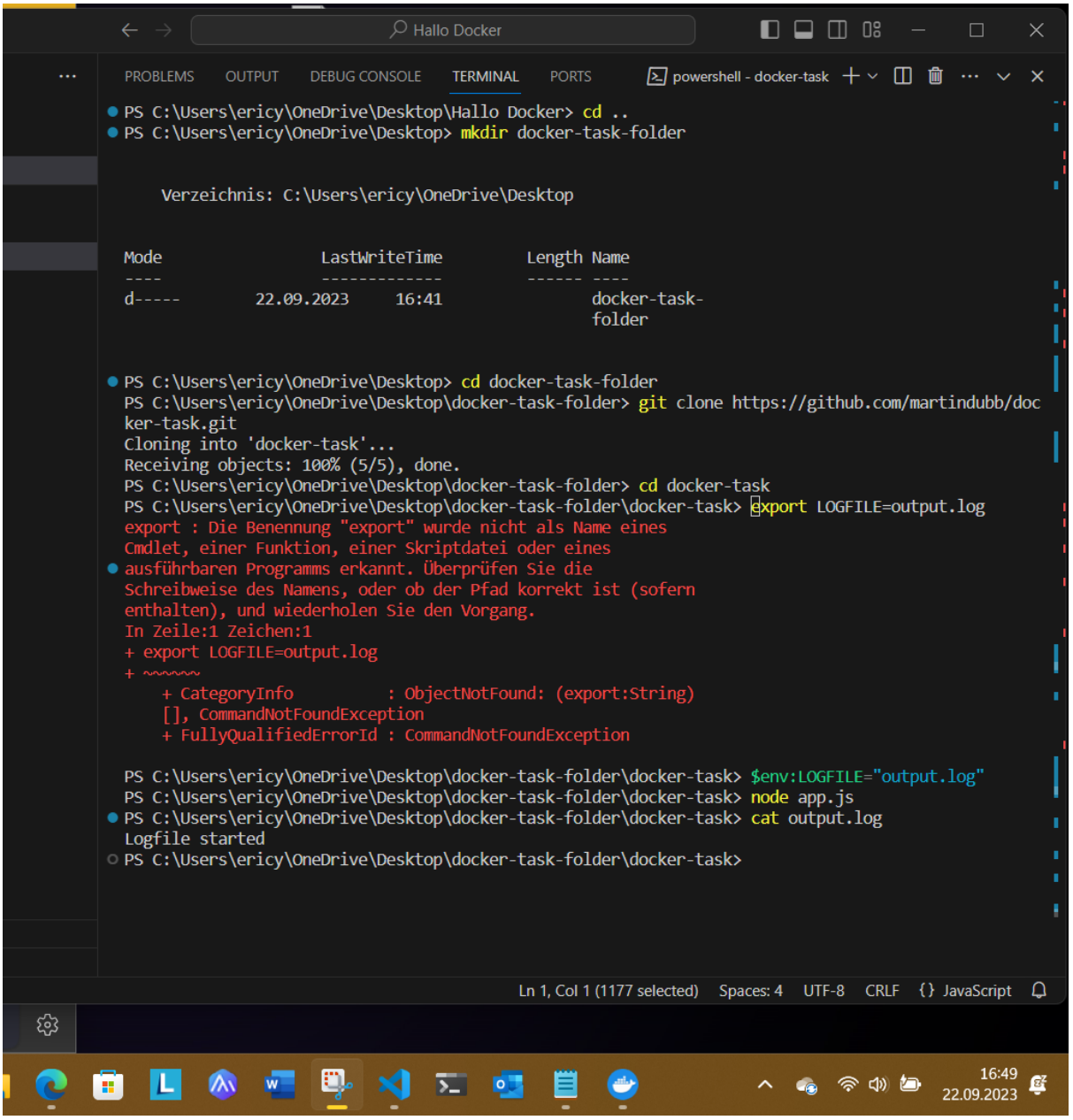
Die App hört auf Port 8080.
package.json:

Dies ist die Konfigurationsdatei für das Projekt. Es enthält Informationen über das Projekt wie den Namen, die Version und die Abhängigkeiten.
Die App hat zwei Hauptabhängigkeiten: body-parser und express.
Zusammenfassend ist die App eine einfache Webanwendung, die es dem Benutzer ermöglicht, einen Text einzugeben. Dieser Text wird dann sowohl in der Konsole als auch in einem Logfile protokolliert und auf der Hauptseite der App angezeigt.

Aufgabe 2

a) Klone das Repo auf deinen Rechner und starte die App.
b) Stelle sicher, dass die App den Log Output in eine Datei namens “output.log” schreibt, ohne den Code zu ändern! Wie kann das bewerkstelligt werden?
(Tipp: Zeile 6-8 in der app.js)

Erstelle einen Screenshot für die Abgabe.

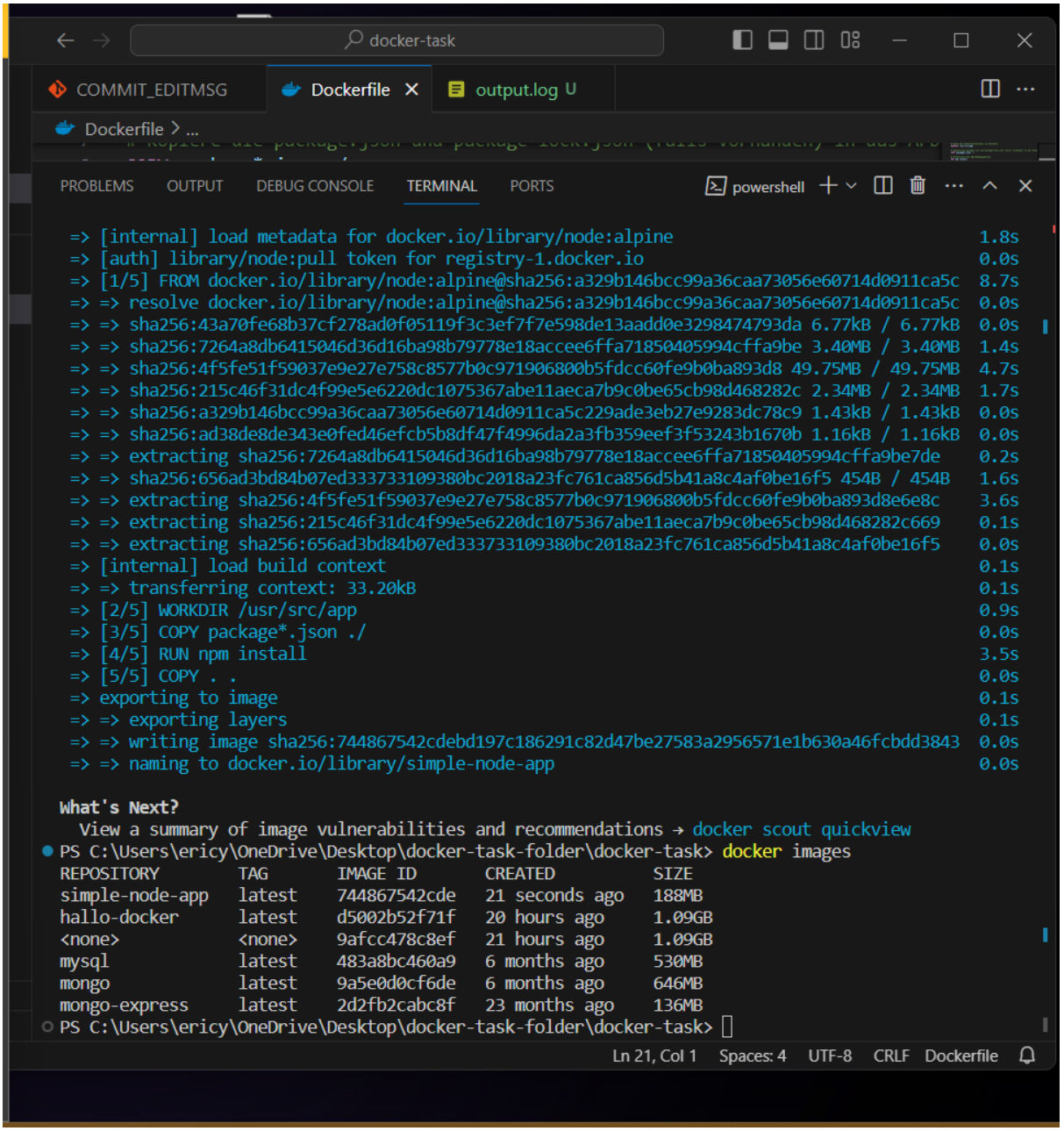


Aufgabe 3

Containerisiere die App!

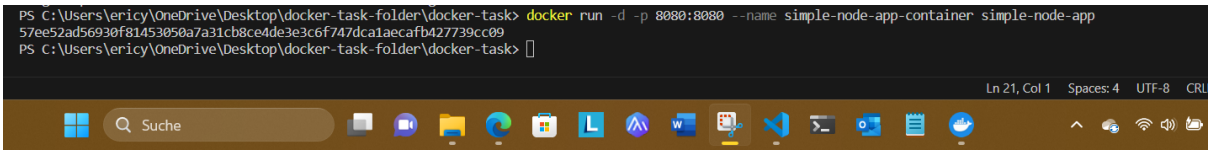
Schreibe ein Dockerfile, welches ein alpine-Image nutzt und in dem NodeJS zur Verfügung steht. Baue das Image und vergib den Namen “simple-node-app”. Liste alle Images von deinem Hostsystem auf.

Erstelle einen Screenshot für die Abgabe.



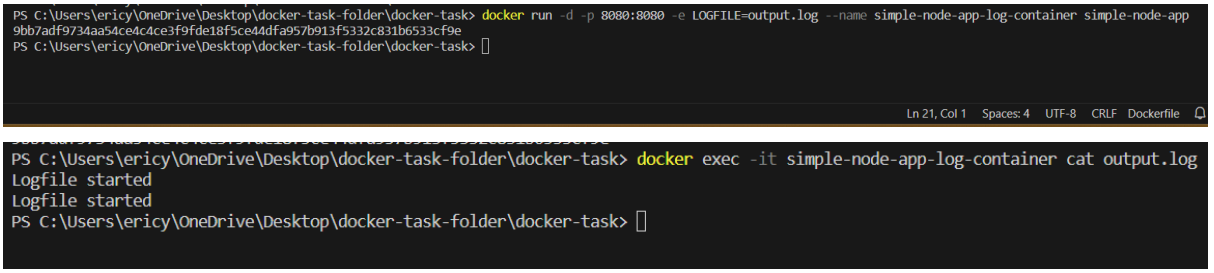
Aufgabe 4

a) Starte einen Container vom zuvor erstellten Image “simple-node-app”.
Stelle sicher, dass die App auf deinem Hostsystem erreichbar ist (Stichwort Port-Binding).
Erstelle einen Screenshot für die Abgabe und stoppe den Container wieder.

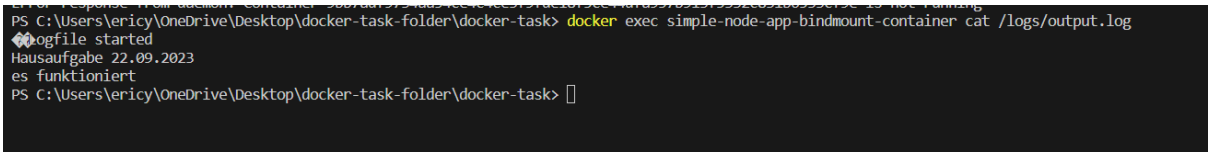


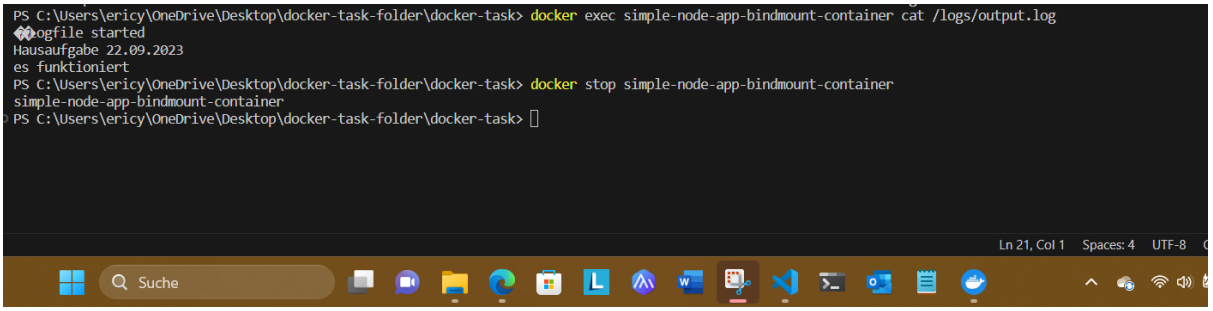
b) Starte einen Container und stelle wie in Aufgabe 2 sicher, dass die App den Log Output in eine Datei namens “output.log” schreibt, ohne den Code in der Datei app.js zu ändern. Welche Option muss dafür beim Start des Containers mitgegebenwerden? (Tipp: Umgebungsvariablen können auch im “docker run”-Befehl angegeben werden)

Erstelle einen Screenshot für die Abgabe und stoppe den Container wieder.



c) Erstelle einen separaten Ordner und eine Textdatei mit dem Namen “output.log” auf deinem Hostsystem. Nutze einen Bind-Mount, um den Ordner und die Datei im Container zur Verfügung zu stellen und starte den Container wie im Aufgabenteil b). Der Log Output sollte nun in der Datei auf deinem Hostsystem erscheinen!
Erstelle einen Screenshot für die Abgabe und stoppe den Container wieder.





Hilfen:

- Ports: https://docs.docker.com/engine/reference/commandline/run/#publish
- env: https://docs.docker.com/engine/reference/commandline/run/#env
- mount: https://docs.docker.com/engine/reference/commandline/run/#mount