

Container – Übungsaufgaben Lösungen

In diesem Assessment wirst du eine “Hallo Welt!” Node.js Anwendung ausführen.

1. Starte einen Node.js Server mit den folgenden Bedingungen:

- Lass den Container im Hintergrund laufen
- Gib dem Container einen Namen
- Lasse den Container automatisch entfernen, nachdem er beendet wurde
- Verwende das `node` Image mit der Version/Tag `8`

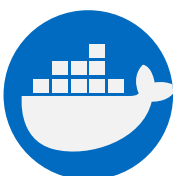
```
docker run -d --rm --name mein_nodejs node:8
```

2. Die `hallo.js` Datei ist im gleichen Verzeichnis wie dieses Dokument. Stelle sicher, dass die Datei in den Container übertragen wird. Zum Beispiel in das Verzeichnis `/usr/src/app`

```
docker run -d --rm --name mein_nodejs -v "$PWD":/usr/src/app  
node:8
```

3. Setze das Arbeitsverzeichnis auf das Verzeichnis mit der `hallo.js`

```
docker run -d --rm --name mein_nodejs -v "$PWD":/usr/src/app  
-w /usr/src/app node:8
```



4. Starte die Anwendung mit dem Befehl `node hallo.js`

```
docker run -d --rm --name mein_nodejs -v "$PWD":/usr/src/app  
-w /usr/src/app node:8 node hallo.js
```

5. Übergib die Umgebungsvariable `"NODE_ENV=production"`. Damit erkennt Node.js, dass es sich um einen produktiven Prozess handelt.

```
docker run -d --rm --name mein_nodejs -e  
"NODE_ENV=production" -v "$PWD":/usr/src/app -w /usr/src/app  
node:8 node hallo.js
```

6. Limitiere den Arbeitsspeicher auf 300 MB und das Swap auf 1 GB

```
docker run -d --rm --name mein_nodejs -e  
"NODE_ENV=production" -m "300M" --memory-swap "1G" -v  
"$PWD":/usr/src/app -w /usr/src/app node:8 node hallo.js
```

7. Lass uns jetzt die Docker log Dateien anschauen, um zu sehen ob die Anwendung gestartet wurde.

Hinweis: Führe dazu nochmal den Befehl ohne den `--rm` Parameter aus

```
docker logs mein_nodejs
```

