
Raspi-Docker

hostname.

```
raspi-docker
```

samba Austauschlaufwerk.

```
/media/public/
```

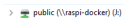
A small screenshot of a Windows File Explorer window. The address bar shows 'public (\\raspi-docker) (U)'. The main area is empty, indicating a network share.

Figure 1. Laufwerk in Windows

In dieses Austauschlaufwerk lege ich das aktuelle jar-file. Zum Beispiel *student-backend.jar* für den Rest-Service.

Diese Datei muss dann in das Docker-Projekt-Verzeichnis kopiert werden.

portainer Adresse.

```
http://raspi-docker:9000  
http://staecker.dyndns.org:9000/#/auth
```

1. Java Application, Rest-Service "Student-Backend"

Dockerordner liegt unter

```
/home/pirate/docker-projects/student-backend/
```

Hier liegt auch das Dockerfile.

Dockerfile.

```
FROM library/alpine ❶  
RUN mkdir /myVol ❷  
  
ENV TZ="Europe/Berlin"  
  
LABEL maintainer "staecker@t-online.de"  
LABEL description "Image for Spring Boot Rest Services"  
  
EXPOSE 8088 ❸
```

```

RUN apk update
RUN apk add openjdk8-jre ❹
RUN apk add --no-cache openssh
RUN apk add bash

WORKDIR /
# COPY /media/public/student-backend.jar /myvol/student-backend.jar
# CMD ["java", "-jar", "/media/public/student-backend.jar"]
CMD ["java", "-jar", "/myvol/student-backend.jar"] ❺

```

- ❶ Basisimage, welches verwendet wird
- ❷ Verzeichnis erstellen als später verwendeter mountpoint
- ❸ Port, welcher nach aussen gereicht wird
- ❹ Java Runtime installieren
- ❺ Java Applikation starten

1.1. Image File aus Dockerfile erzeugen

in das Docker-Projekt-Verzeichnis wechseln.

```
cd /home/pirate/docker-projects/student-backend/
```

Image aufgrund des Dockerfiles erzeugen.

```
docker build -t student-backend .
```

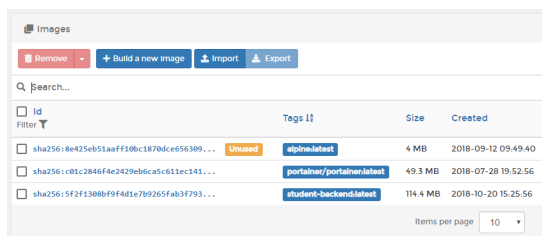


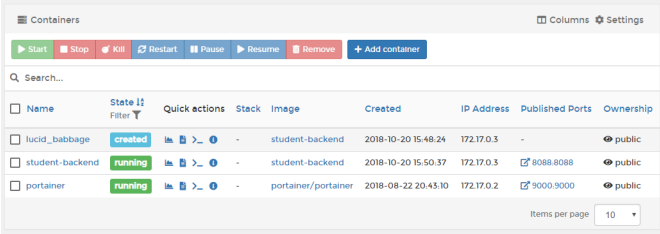
Figure 2. Image sollte im portainer zu sehen sein

1.2. Container starten

```
docker run --name student-backend --mount source=student-
manager,target=/myvol -p 8088:8088 student-backend
```

```
docker run --name student-backend -v /media/public:/myvol -p 8088:8088
student-backend
```

```
docker run -v /host/directory:/container/directory
```



Name	State	Quick actions	Stack	Image	Created	IP Address	Published Ports	Ownership
lucid_babbage	created	[icons]	-	student-backend	2018-10-20 15:48:24	172.17.0.3	-	public
student-backend	running	[icons]	-	student-backend	2018-10-20 15:50:37	172.17.0.3	8088:8088	public
portainer	running	[icons]	-	portainer/portainer	2018-08-22 20:45:10	172.17.0.2	9000:9000	public

Figure 3. container sollte in portainer zu sehen sein und laufen

1.3. Rest Service nutzen

alle Studenten abrufen (get).

```
staecker.dyndns.org:8088/students
```

einen Studenten abrufen (get mit Parameter).

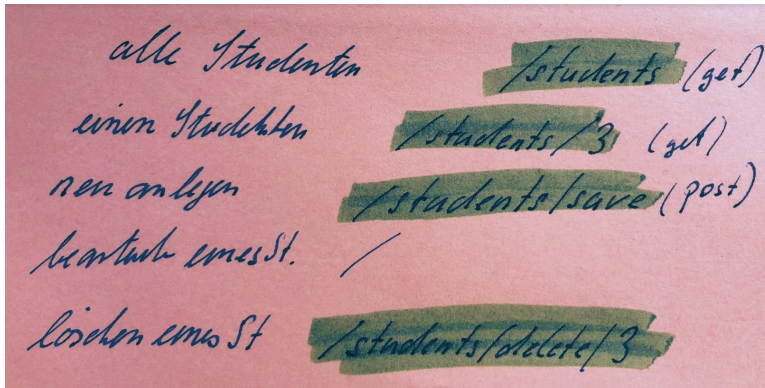
```
staecker.dyndns.org:8088/students/3
```

einen neuen Studenten anlegen (POST).

```
staecker.dyndns.org:8088/students/save
```

json im Body für den neuen Studenten.

```
{
  "firstname": "Bernd",
  "lastname": "Stromberg"
}
```



2. Internetseiten zum Thema Docker

<https://hub.docker.com/> Eine Auswahl von offiziellen und privaten Dockerimages.

2.1. Bücher