

---

# Laborprotokoll

## Docker

---

Systemtechnik Labor  
5YHITM 2016/17, Gruppe A

Maximilian Seidl, Christoph Miko

Note:  
Betreuer: M.Schabel

Version 0.1  
Begonnen am 20. Januar 2017  
Beendet am 20. Januar 2017

# Inhaltsverzeichnis

<b>1</b>	<b>Einführung</b>	<b>1</b>
1.1	Ziele . . . . .	1
1.2	Aufgabenstellung . . . . .	1
<b>2</b>	<b>Ergebnisse</b>	<b>2</b>
2.1	Installieren von Docker auf Arch Linux . . . . .	2
2.1.1	Vorbereitung . . . . .	2
2.1.2	Installation . . . . .	2
2.1.3	Starten von Docker . . . . .	2
2.2	Installieren eines Docker-Image . . . . .	2
2.3	MySQL auf der Docker-Machine . . . . .	3
2.4	Probleme . . . . .	3
2.5	Erstellen eines eigenen Dockerfile (Image) . . . . .	3
2.5.1	Syntax des Dockerfiles . . . . .	3
2.5.2	Fertiges Dockerfile . . . . .	4
2.5.3	Probleme . . . . .	4
2.6	Verbundene Images (Wordpress) . . . . .	4
2.6.1	Verwenden des Images . . . . .	4
2.6.2	Probleme . . . . .	5

# 1 Einführung

Diese Übung behandelt sich mit Docker-Images.

## 1.1 Ziele

Manuelle Erstellung

- Linux-Basisimage aufsetzen
- Dienst Installieren (mysql/postgresql/mongodb/couchbase/...)

Automatische Erstellung

- Dockerfile, obige Image erstellen.

Verbundene Images

- Starten und Konfigurieren des WordPress-Container

Abgabe des Protokolls (pdf) pro Gruppe - max 3. Personen / Gruppe.

## 1.2 Aufgabenstellung

Manuelle Erstellung

- Starten Sie mit einem Linux-Basisimage (z.B. Debian oder Ubuntu)
- Installieren Sie einen Dienst nach Wahl (mysql / postgresql / mongodb / couchbase / ..) und konfigurieren Sie diesen
- Konfigurieren Sie den Container so, dass dieser automatisch den angegebenen Dienst startet

Automatische Erstellung

- Erstellen Sie ein Dockerfile, welches das obige Image automatisch erstellt
- Laden Sie die folgenden Images: mysql, wordpress
- Konfigurieren und Starten Sie den WordPress-Container so, dass dieser die Datenbank aus dem MySQL-Container verwendet ([https://hub.docker.com/\\_/wordpress/](https://hub.docker.com/_/wordpress/))

## 2 Ergebnisse

### 2.1 Installieren von Docker auf Arch Linux

#### 2.1.1 Vorbereitung

Die Installation von Docker auf Arch Linux kann auf zwei verschiedene Arten durchgeführt werden. Entweder über das Community-package oder über das package aus dem AUR. Docker benötigt verschiedene Dependencies:

- bridge-utils
- device-mapper
- iproute2
- sqlite

#### 2.1.2 Installation

Um das normale package zu installieren reicht ein simples

```
1 $ sudo pacman -S docker
```

Listing 1: Installieren des Docker packages über Pacman

Oder im pack AUR

```
1 $ yaourt -S docker-git
```

Listing 2: Docker aus dem AUR

#### 2.1.3 Starten von Docker

Nach der Installation wird ein **systemd** service erstellt und wird folgendermaßen gestartet:

```
1 $ systemctl start docker
```

Listing 3: Starten von Docker

### 2.2 Installieren eines Docker-Image

Um eine neues Docker-Projekt anzulegen muss ein neuer Ordner angelegt werden und danach ein neues File erstellen mit dem Namen: **Dockerfile**

```
1 $ mkdir dockerstuff
2 $ cd dockerstuff
3 $ nano Dockerfile
```

Listing 4: Erstellen des Docker-Project root

Zu Beginn fügt man nur den Namen des Images ein:

```
1 $ FROM ubuntu
```

Listing 5: Dockerfile erster Eintrag

Jetzt kann das neue Image gebuildet werden. Dieser Befehl passiert mit einer Namensgebung

```
1 $ docker build -t testproj
```

Listing 6: Builden des Images

Um eine Verbindung zum Docker-Container herzustellen muss er noch gestartet werden:

```
1 $ docker run -it ubuntu
```

Listing 7: Starten des Containers

## 2.3 MySQL auf der Docker-Machine

Um die manuelle Erstellung abzuschließen wird auf der Maschine noch ein Tutorial durchgeführt, wie man MySQL installiert. MySQL-Installationsanweisung

```
1 $ systemctl enable mysql
```

Listing 8: MySQL beim Start ausführen

## 2.4 Probleme

Um auf dem Docker-Container Packages installieren zu können muss zunächst ein **update und upgrade** des Systems vorgenommen werden.

## 2.5 Erstellen eines eigenen Dockerfile (Image)

### 2.5.1 Syntax des Dockerfiles

Um das Dockerfile richtig zu konfigurieren müssen gewisse Syntax-Regeln eingehalten werden.

- **# COMMENT** mit # wird ein Kommentar eingeleitet
- **INSTRUCTIONS** Dieser Befehl hebt hervor, welche Parameter und andere Vorbereitung getroffen werden müssen um ein Dockerfile lauffähig zu machen.
- **FROM** Ist immer die erste Zeile, da es das Image festlegt welches verwendet wird.
- **MAINTAINER** Legt den Autor des Dockerfiles fest.
- **ENV** Setzt die Umgebungsvariablen in der Form [key] [value]. Der value kann irgendein String sein, zum Beispiel auch IP-Adressen, URLs.

- Es gibt verschiedene Varianten ein Umgebungsvariablen zu setzen: **ADD**, **COPY**, **ENV**, **EXPOSE**, **LABEL**, **USER**, **WORKDIR**, **VOLUME**, **STOPSIGNAL** und **ON-BUILD**.
- **RUN** Führt ein command aus. Zum Beispiel führt **RUN apt-get update && apt-get install mysql**, führt ein update durch und installiert mysql.
- **EXPOSE** Definiert den Port auf dem die Virtuelle Maschine laufen soll.
- **CMD** Kann nur einmal in einem Dockerfile vorkommen. Die Syntax ist **CMD ["executable","param1","param2"]**
- **ENTRYPOINT** beschreibt welcher command beim Start ausgeführt werden soll.

### 2.5.2 Fertiges Dockerfile

```

1      # Pull base image.
2      FROM ubuntu:14.04
3
4      # Install.
5      RUN \
6      sed -i 's/# \(\.*multiverse$\)\1/g' /etc/apt/sources.list && \
7      apt-get update && \
8      apt-get -y upgrade && \
9      apt-get install -y build-essential && \
10     apt-get install -y software-properties-common && \
11     apt-get install -y byobu curl git htop man unzip vim wget && \
12     apt-get install -y mysql-server && \
13     rm -rf /var/lib/apt/lists/*
14
15     # Set environment variables.
16     ENV HOME /root
17
18     # Define working directory.
19     WORKDIR /root
20
21     #RUN service mysql restart && /tmp/setup.sh
22
23     # Define default command.
24     ENTRYPOINT service mysql restart && bash

```

Listing 9: Dockerfile des selbsterstellten Images

### 2.5.3 Probleme

Damit die Installationsprompt automatisch durchgeführt wird kann man beim Installieren von Packages ein **-y** als Option hinzugefügt werden. Dies bewirkt eine **noninteractive-Installation**

## 2.6 Verbundene Images (Wordpress)

### 2.6.1 Verwenden des Images

Ein lauffähiges **docker-compose** wird benötigt damit der Wordpress-Container funktioniert.

```

2         version: '2'
4         services:
6             wordpress:
8                 image: wordpress
10                ports:
12                - 8000:80
14                environment:
16                WORDPRESS_DB_PASSWORD: example
18
20             mysql:
22                 image: mariadb
24                 environment:
26                 MYSQL_ROOT_PASSWORD: example

```

Listing 10: docker-compose.yml

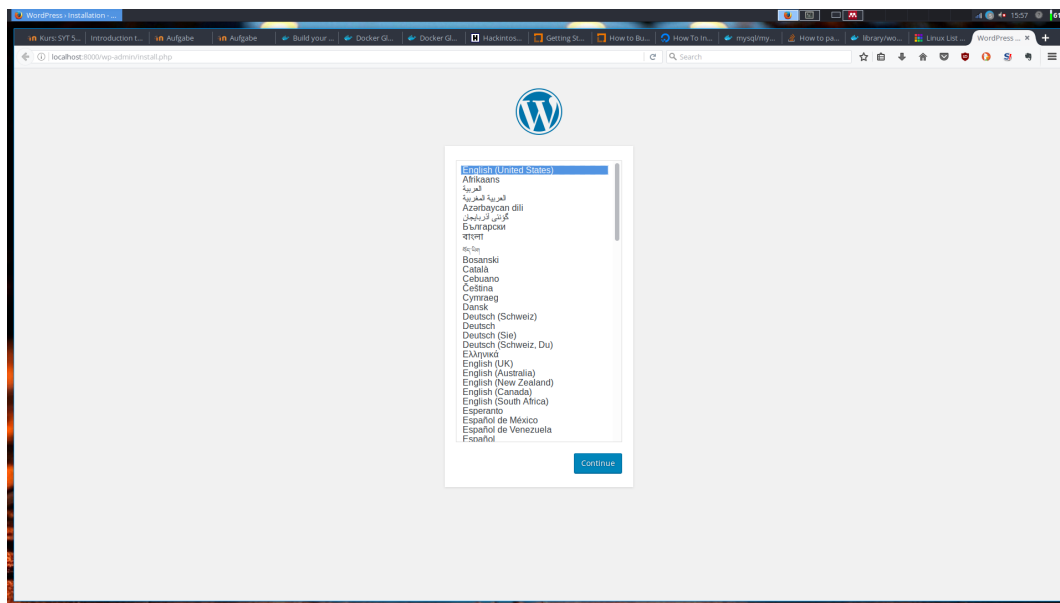


Abbildung 1: Ergebnis des Wordpress-Containers

### 2.6.2 Probleme

Damit der **docker-compose** einwandfrei funktioniert wird noch eine Dependency benötigt, da sonst diese Fehlermeldung auftritt:

```
1 $ pkg_resources.DistributionNotFound: The 'colorama>=0.3.7' distribution was not found
   and is required by docker-compose
```

Listing 11: pip Installationsanweisung

Die Fehlermeldung wird behoben, indem man ein Package mit dem **Python Installer pip** installiert.

```
1 $ pip install colorama
```

Listing 12: pip Installationsanweisung

### 3 Quellen

Wordpress-Docker: [https://hub.docker.com/\\_/wordpress/](https://hub.docker.com/_/wordpress/)

Docker-Setup: <https://www.linux.com/news/getting-started-docker>

Docker-Image: <https://www.linux.com/learn/how-build-your-own-custom-docker-images>

Docker-Arch Linux: <http://docs-stage.docker.com/engine/installation/linux/archlinux/>



## Listings

1	Installieren des Docker packages über Pacman . . . . .	2
2	Docker aus dem AUR . . . . .	2
3	Starten von Docker . . . . .	2
4	Erstellen des Docker-Project root . . . . .	2
5	Dockerfile erster Eintrag . . . . .	3
6	Builden des Images . . . . .	3
7	Starten des Containers . . . . .	3
8	MySQL beim Start ausführen . . . . .	3
9	Dockerfile des selbsterstellten Images . . . . .	4
10	docker-compose.yml . . . . .	5
11	pip Installationsanweisung . . . . .	5
12	pip Installationsanweisung . . . . .	5

## Abbildungsverzeichnis

1	Ergebnis des Wordpress-Containers . . . . .	5
---	---	---