

# 1 Einleitung

Träumt nicht jeder an Informationstechnik interessierte Mensch von der allumfassenden sicheren und unbegrenzten Cloud-Lösung für einen intuitiven Workflow über alle Geräte und Plattformen hinweg? Der Autor möchte sich diesen Traum erfüllen und seinen eigenen Webserver betreiben, um dort verschiedene Dienste anbieten zu können. Aber Halt! Ist es nicht viel einfacher, diese Lösung an extern zu vergeben und die Dienste von Apple, Google und wie sie nicht alle heißen zu nutzen? Einfacher in jedem Fall, doch möchte man seine Daten nicht mit einem internationalen Konzern teilen, so führt kein Weg an der selbst gebauten Lösung vorbei. Des Weiteren ist es mit den heute verfügbaren Open-Source-Werkzeugen einfach geworden, einen eigenen Webserver zu betreiben und diesen universell zu nutzen. Beispielsweise ist auch die Gestaltung eines Webauftritts mit WordPress einfach realisierbar.

Im Folgenden wird zunächst das Problem näher erläutert und die möglichen bereits vorhandenen Lösungen mit der geplanten selbst betriebenen Lösung verglichen. Darauf aufbauend wird eine grobe Systemarchitektur skizziert, sowie das Projekt in Arbeitspakete unterteilt.

## 2 Motivation

Spätestens seitdem wir angefangen haben, täglich mehrere Computer verschiedener Größe vom Handy bis zur Full-Size-Workstation zu nutzen, haben wir uns bezüglich des Austausches von Informationen und Dateien einige Probleme eingehandelt: Ohne die Nutzung einer Cloud-Lösung ist es schwierig, den automatisierten Austausch zwischen den Geräten zu realisieren. Das Ergebnis ist oft doppelte und inkonsistente Datenhaltung auf unterschiedlichen Medien (USB-Stick, Handy, Laptop etc.). Die grundlegenden Probleme, zu denen im vorliegenden Projekt Lösungen erarbeitet werden sollen, lassen sich auf die folgenden drei Stichpunkte reduzieren.

1. Konsistenz: Wie kann ein einheitlicher Versionsstand zwischen den Geräten hergestellt werden?
2. Verfügbarkeit: Wie kann jedem Gerät zu jedem Zeitpunkt die vollständige Datenbasis zur Verfügung gestellt werden?
3. Vertraulichkeit: Es ist nicht gewünscht, persönliche Daten mit multinationalen Konzernen zu teilen. Wie kann dieses Ziel erreicht werden?

Die Lösung für Problem 1 und 2 scheint wie bereits angedeutet einfach: Die Nutzung einer Cloud-Lösung als verbindende Instanz zwischen allen Geräten. Hierzu gibt es bereits fertige Lösungen wie z.B. Google Drive, iCloud, Dropbox, OneDrive und viele mehr. All diese Fertiglösungen haben jedoch den Nachteil, dass der gesamte persönliche Datenbestand auf einen externen (amerikanischen) Server ausgelagert wird und die dahinter stehenden Unternehmen damit machen können, was sie möchten. Ist dies für den Nutzer akzeptabel steht der Nutzung einer solchen Cloud-Lösung nichts mehr im Wege. Ist man jedoch daran interessiert, selbst über seine Daten zu verfügen, muss etwas mehr Aufwand getrieben werden.

Auch für Home-Clouds gibt es fertige Lösungen. Zu nennen ist hierbei die Serverhardware von QNAP und Synology, welche eine Plug-and-Play-Lösung darstellen. Gegen die Nutzung einer solchen Plattform sprechen jedoch folgende Gründe:

- Preis-Leistungs-Verhältnis: Ein Synology Network-Attached-Storage-Server (NAS) mit einem Dual-Core Prozessor von Intel und 2GB RAM kostet ca. 450€. Für den selben Preis ist es möglich einen Server aus gebrauchten Teilen mit einem Hexa-Core-Xeon Prozessor und 64GB RAM zusammenzustellen.
- Unflexibel: Die Lösungen von QNAP und Synology sind *ausschließlich* als NAS konzipiert. Ein selbst aufgesetzter Server kann z.B. auch als Plattform für ein GitLab ein WordPress-Blog oder Ähnliches dienen.
- Begrenzte Erweiterbarkeit: Die Baugröße eines typischen Plug-and-Play-NAS erlaubt keine größeren Upgrades bezüglich Prozessor, RAM und Massenspeicher.

Die Motivation, eine eigene Plattform aufzusetzen, ist damit begründet. Im folgenden Kapitel wird kurz die Hard- und Software-Architektur beschrieben.

## 3 Details

### 3.1 Hardware

Die wichtigsten Komponenten des Servers lassen sich wie folgt darstellen:

- Fujitsu D3128-B25 Mainboard mit Intel C602 Server Chipsatz
- 8 x 8GB DDR3-RDIMM ECC Arbeitsspeicher
- Intel XEON E5-2560V2 Hexa-Core Server-CPU

All diesen Komponenten ist gemeinsam, dass sie auf Server-Anforderungen wie z.B. Dauerbetrieb optimiert sind. Des Weiteren sind die Komponenten gebraucht sehr günstig verfügbar.

## 3.2 Software

Als Betriebssystem soll ein Ubuntu Server 18.04 zum Einsatz kommen, begründet dadurch, dass der Autor bereits über einschlägige Erfahrung mit Ubuntu verfügt und somit die Einarbeitungszeit entfällt. Grundsätzlich ist aber auch der Einsatz einer anderen Linux-Distribution denkbar.

Eine Schlüsselposition nimmt die Virtualisierungssoftware Docker ein. Docker ist mit der Light-Version einer VM vergleichbar: Für eine Applikation wird zur Laufzeit die Infrastruktur wie die Laufzeitumgebung (im Fall von z.B. Python) oder sogar ein rudimentäres Betriebssystem (im Fall von z.B. NGINX als Webserver) bereit gestellt. Gekapselt wird jede Funktionalität in einem sog. Container, welcher *unabhängig* von der physischen Plattform läuft, die einzige Bedingung ist eine ähnliche Prozessorarchitektur und ein installiertes Docker. Der Vorteil dieser *Containervirtualisierung* liegt somit auch darin begründet, dass eine Applikation auf einer anderen Maschine entwickelt werden kann und beim Deployment auf dem Produktivsystem *garantiert* funktioniert. Abbildung 1 veranschaulicht die Rolle von Docker im Software-Stack.

## 3.3 Arbeitspakete

Zur Durchführung des Projektes müssen folgende Aufgaben zu den angegebenen Terminen erledigt werden:

### 3.3.1 Inbetriebnahme des Servers - Mitte November 2019

Da es sich bei dem Server um eine bildschirmlose Maschine handelt, muss das Betriebssystem automatisiert installiert werden. Hierfür eignet sich die Methode mittels Pre-seeding.

### 3.3.2 Deployment der Cloud - Anfang Dezember 2019

Die Inbetriebnahme der Cloudsoftware ist durch den Einsatz von Docker unkompliziert. Es existiert ein fertiges Docker-Image, welches durch wenige Befehle in Betrieb genommen werden kann. Wichtig ist hierbei auch die Verschlüsselung mit einem gültigem SSL-Zertifikat beispielsweise von Letsencrypt.

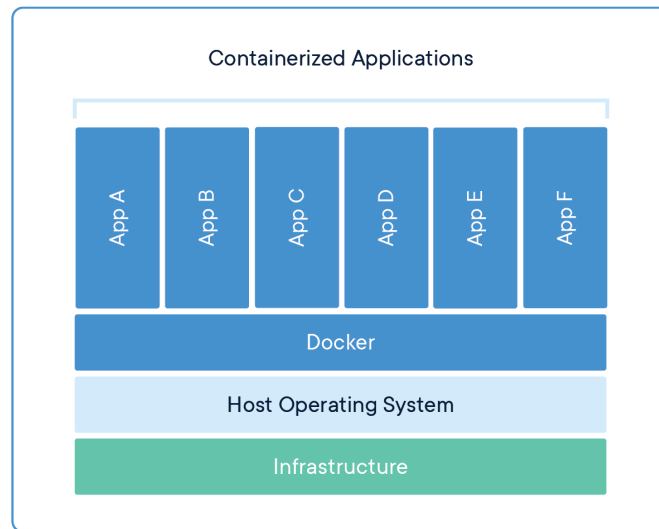


Abbildung 1: Docker als Instanz zwischen Applikation und Betriebssystem.  
Quelle Docker.com

### 3.3.3 Aufsetzen des Webauftritts - Weihnachten 2019

Zum unkomplizierten Design eines Webauftritts eignet sich die Open-Source-Software WordPress, welche ebenfalls als Docker-Container laufen kann. Da bezüglich WordPress und Front-End-Webdesign erst wenige Erfahrungen vorliegen, wird dieser Projektteil die meiste Zeit in Anspruch nehmen.

## 4 Zusammenfassung

Ziel des Projektes ist es, eine sichere und von überall erreichbare Plattform aufzubauen. Am Ende soll eine hochflexible und einfach zu erweiternde Server-Lösung stehen, welche sich leicht auf andere Hardware portieren lässt. Den Erweiterungsmöglichkeiten sind dank des modularen Systems von Docker lediglich die physikalischen Grenzen der Maschine gesetzt. Des Weiteren ist die eingesetzte Software zu 100% frei verfügbar und Open-Source.

## 5 Informationen zum Autor

Der Autor Thilo Wendt hat im Sommersemester 2019 erfolgreich sein Bachelorstudium an der TH-Nürnberg im Fach Elektrotechnik und Informationstechnik abgeschlossen. Der Studienschwerpunkt lag hierbei auf Informatik insbesondere im Bereich Hardware-Software-Co-Design und Low-Level-Programmierung. In seiner Freizeit beschäftigt sich Herr Wendt gerne mit Server-Projekten zum Hosting und der Gestaltung seines eigenen Web-Auftrittes.

Kontaktdaten:  
Thilo Wendt  
Mobil: +49 160 45 72 775  
E-Mail: wendtth65566@th-nuernberg.de