



docker

# PFLICHTENHEFT

Diplomarbeit Informatiktechniker HF

## THEMA

Aufbau und Analyse eines Web-Service  
Clusters mit Docker

Philipp Dubach  
TEKO TIN-O-13a

# Inhaltsverzeichnis

<b>1. VERSIONSNACHWEIS</b>	<b>3</b>
<b>2. PROJEKTINFORMATIONEN</b>	<b>4</b>
<b>3. KONKRETISIERUNG</b>	<b>5</b>
<b>3.1. EINLEITUNG</b>	<b>5</b>
3.1.1. THEMA	5
3.1.2. GRUNDLAGE	5
3.1.3. MOTIVATION	5
3.1.4. ALTERNATIVEN	6
<b>3.2. KONKRETISIERUNG DES ZIELS</b>	<b>8</b>
3.2.1. KOMPENDIUM	8
3.2.2. EVALUIERUNG DER KOMPONENTEN	8
3.2.3. REALISIERUNG DER INFRASTRUKTUR	10
3.2.4. ANALYSE DES SYSTEMS	10
3.2.5. ERFOLGSKRITERIEN	10



## 2. Projektinformationen

Thema	Aufbau und Analyse eines Web-Service Clusters mit Docker
Autor/Diplomand	Philipp Dubach, Bogenstrasse 3, CH-5614 Sarmenstorf AG
Studium	Informatiktechniker HF
Schule	TEKO Olten, Belchenstrasse 9, CH-4600 Olten SO
Klasse	TIN-O-13a
Diplombetreuer	Graziano Feline, Fachdozent TEKÖ Olten

### Termine

13.6.2016	Themeneingabe
14.8.2016	Pflichtenheft einreichen an Diplombetreuer
19.9.2016	Start Diplomarbeit
26-30.9.2016	Vorzeigetermin 1 mit Diplombetreuer
10-14.10.2016	Vorzeigetermin 2 mit Diplombetreuer
31.10.2016	Abgabe Diplomarbeit
18/19.11.2016	Präsentation Diplomarbeit

## 3. Konkretisierung

### 3.1. Einleitung

#### 3.1.1. Thema

Die Studie soll Einblicke in die aktuell florierende Technologie der Container-Virtualisierung **Docker** geben. Anhand eines konkreten Projektes wird die Technologie erläutert. Es wird aufgezeigt wie ein Clustersystem aufgebaut und erarbeitet wird. Am Ende der Studie wird die Infrastruktur des Systems analysiert und getestet mittels Benchmark und Failover-Szenarien um die Hochverfügbarkeit zu prüfen. Die Studie soll zudem ein Kompendium sein um die nötigen Kenntnisse der Technologie zu erlernen.

#### 3.1.2. Grundlage

Vor ein paar Monaten habe ich erstmals etwas von der Container-Virtualisierung gehört. Wenn ich mich recht erinnere, brachte mich die Java IDE Eclipse auf das Thema. Eclipse präsentierte auf Ihrer Webpage „Eclipse Che“ welches anstatt einer Desktop-Applikation im Webbrowser läuft. Ich wollte natürlich wissen, auf welche Art das Ganze installiert werden kann und wie die Technologie funktioniert. Eclipse hatte gleich 3 Varianten zur Verfügung gestellt um das System zu installieren. Einmal als Image für Virtualbox, zum Zweiten als Cloud-Variante auf den Servern von Codenvy.com und drittens als native Version mittels Container auf einem eigenen Server. Ich habe alle 3 Varianten ausprobiert und war begeistert. Ich habe dadurch gemerkt, dass eigentlich alle Varianten mit Linux im selben Container laufen. Bei den Varianten eins und zwei wird einfach eine virtuelle Linux Instanz mit Virtualbox erstellt und gestartet. Der Container welcher danach gestartet wird, war bei allen Varianten der Gleiche. Beim Durchlesen der Dokumentation und beim Testen von Eclipse Che bin ich dann auch auf Docker gestossen. Ich fing an mich für das Thema Docker zu interessieren. In der Zwischenzeit habe ich viel Zeit damit verbracht mich mit der dieser Technologie zu befassen. Ich habe Container virtualisiert wie Wordpress, Nginx, Ubuntu und MySQL. Das Internet und diversen Bücher waren meine steten Begleiter, sobald ich Zeit hatte mich damit zu befassen.

#### 3.1.3. Motivation

Da mich das Thema so in den Bann gezogen hat und ich ein Thema für die Diplomarbeit suchen durfte, war das Thema Docker nebst den anderen Themen hoch im Kurs. Ich überlegte mir, welches Arbeit für mich persönlich den grössten Nutzen und Mehrwert generiert. Somit habe ich mich für eine Studie mit Docker entschieden. Nicht nur um eine Arbeit schreiben, sondern um mich persönlich weiterzubilden. Das Thema habe ich bewusst

so gewählt, dass es für mich eine hohe Herausforderung darstellt. Es soll dazu dienen, dass ich mich noch tiefer in diese Materie einarbeite. Da ich in der Firma Simeta AG für die Systemadministration, das Prozessmanagement und Evaluierungen von Software Konzepten verantwortlich bin, bin ich der Auffassung, dass es für zukünftige berufliche Projekte sehr Wertvoll sein wird.

### **3.1.4. Alternativen**

Die Docker Technologie steht natürlich nicht alleine da. Es gibt diverse Alternativen für Virtualisierungen. Drei davon werde ich hier erläutern.

#### **3.1.4.1. System-Virtualisierung Hypervisor Typ-1**

Bei der Typ-1 System-Virtualisierung läuft ein Hypervisor-Betriebssystem nativ auf der Hardware. Mit diesem Betriebssystem ist es durch eine Virtualisierungsschicht möglich weitere Betriebssysteme gleichzeitig zu betreiben. Der sogenannte Hypervisor vermittelt dann den Eindruck als würden alle Systeme eigenständig laufen. Den virtuellen Maschinen können CPU Kerne, Speicher in Form von RAM, Netzwerkkarten und virtuelle Festplatten zugeteilt werden. Somit sind die virtuellen Computer abgekoppelt vom Host-Betriebssystem und nutzen Ihre eigenen Ressourcen. Jede virtuelle Maschine benötigt ein komplettes Betriebssystem um Lauffähig zu sein. Das Host-Betriebssystem kümmert sich dann in der Regel um das Managen der virtuellen Maschinen und muss zwingend in Betrieb sein.

Hier ist eine Liste der bekanntesten Typ-1 Hypervisor-Betriebssysteme:

- Microsoft Hyper-V Server
- Citrix XenServer
- VMWare vSphere
- KVM

#### **3.1.4.2. System-Virtualisierung Hypervisor Typ-2**

Der Hypervisor Typ-2 ist sehr ähnlich wie Typ-1. Im Gegensatz zu Typ-1 ist das Host-Betriebssystem bei Typ-2 jedoch ein vollwertiges Betriebssystem und nutzt alle verfügbaren Ressourcen, welches das Host-Betriebssystem anbietet. Um eine virtuelle Umgebung aufzubauen, ist es notwendig eine zusätzliche Virtualisierungssoftware zu installieren. Diese Software organisiert dann die Ressourcen für die virtuellen Maschinen. Die Zuteilung von Hardwareressourcen ist bei Typ-2, wie auch Typ-1 identisch. Die Performance ist bei beiden Systemen gleichwertig, jedoch sind bei Typ-2 weniger Ressourcen für virtuelle Maschinen verfügbar, weil das Host-Betriebssystem mehr Ressourcen in Anspruch nimmt als bei einem reinen Hypervisor-System.

Untenstehend eine Auflistung von bekannten Typ-2 Virtualisierungsanwendungen:

- VMWare Workstation
- Oracle VirtualBox
- Parallels Workstation
- Microsoft Hyper-V

#### **3.1.4.3. Container-Virtualisierung**

Die dritte Art der Virtualisierung setzt ein Host-Betriebssystem voraus, welches auf Linux basiert. Anders als bei den Hypervisor-Systemen wird der Focus direkt auf die Virtualisierung der Anwendung gesetzt. Zum Beispiel ein DHCP Dienst oder ein Web Server. Die Anwendungen, oder anders ausgedrückt die Container, nutzen so direkt den Kernel, das System und die Hardware des Host-Systems. Es muss somit nicht zusätzlich ein komplettes Betriebssystem virtualisiert werden um einen Dienst oder eine Anwendung auszuführen. Die Container sind wie bei den Hypervisor-Systemen vom Hostsystem abgekoppelt. Sie können somit auch eigene IP-Adressen und Ports haben. Dieser Ansatz hat den Vorteil, dass die Systeme performanter sind weil Sie weniger Ressourcen benötigen. Eine Emulation von Grafik oder Hardware findet dabei nicht statt. Der Nachteil ist jedoch, dass die Vielfalt der Anwendungen deutlich eingeschränkter ist. Es kann nur Software verwendet werden, welche auf einem Linux-System lauffähig ist. Grösstenteils basieren die Container auf Open-Source Software und sind zudem kostenlos. Klassische Desktop Anwendungen sind nicht realisierbar. Die Container-Virtualisierung zielt auf Dienste und Webanwendungen.

Folgende bekannte Systeme gibt es:

- Docker
- LXC Linux Containers
- LXD Linux Containers (Nachfolger LXC)
- OpenVZ
- Linux VServer
- Solaris Containers
- FreeBSD Jails

#### **3.1.4.4. Vergleich der Alternativen zu Docker**

Das Konzept der 3 Typen von Virtualisierungstechniken ist unterschiedlich. Je nach Verwendungszweck ist es besser einen Hypervisor zu nutzen. Direkte Alternativen sind deshalb lediglich die Container-Virtualisierungstechniken. Als direkten Konkurrenten sehe ich LXD Containers. Docker baut noch auf LXC Containers auf, was aber nicht unbedingt einen Nachteil sein muss, da es dadurch ein bereits sehr grosses Angebot und weniger Kompatibilitätsprobleme gibt. Docker hat mittlerweile eine der grössten Community-Gruppe

und eine sehr hohe Akzeptanz erreicht. Aktuell gibt es wöchentlich neue Updates und Zusatztools um mit Docker zu arbeiten. Es wächst und wächst. Sogar Microsoft will für sein neues Server Betriebssystem Docker Container unterstützen. Deshalb sehe ich ein sehr grosses Potenzial für die Zukunft von Docker.

## **3.2. Konkretisierung des Ziels**

Für die Konkretisierung des Ziels habe ich mein Projekt unterteilt in Teilziele, welche ich erreichen will. Damit laufe ich nicht in Gefahr vom Thema abzuschweifen. Ich werde hier meine Teilziele konkretisieren und die Abgrenzungskriterien setzen.

### **3.2.1. Kompendium**

Wie bereits erwähnt, möchte ich ein Kompendium für Docker schreiben. Ich werde aber nicht jedes Detail ausführlich erläutern, sondern gezielt auf Details eingehen, welche im Alltag benötigt werden und für die Studie notwendig sind. Das Kompendium soll ein Schnellstart für Personen mit Docker sein welche die Technologie noch nicht kennen. Um mit Docker zu arbeiten, benötigt man ein Basiswissen von Linux. Ich werde die für Docker nötigen Befehle aus dem Linux Bereich erläutern. Das Kompendium wird durch das ganze Projekt führen und ist nicht ein einzelner Teil der Studie. Jedoch soll ein roter Faden durch die Arbeit entstehen. Die komplette Installation von einem Linux-System wird nicht Teil der Studie sein, da jedes Linux System ein bisschen anders ist.

### **3.2.2. Evaluierung der Komponenten**

Die Evaluierung der Komponenten gilt als Vorbereitung für die „Realisierung der Infrastruktur“ für meine Studie. Die Hauptbereiche sind unterteilt in Teilevaluierungen. Ich werde hier eine Konkretisierung zu den Komponenten schreiben welche Teil der Evaluierung sind. Da die Evaluierung ein grosser Teil der Studie wird, werde ich hier erläutern was genau evaluiert werden soll.

#### **3.2.2.1. Betriebssystem**

Es sollen geeignete Betriebssysteme evaluiert werden, welche mit Docker umgehen können. Dies gilt für einen Hauptserver welcher alles verwaltet, sowie für zusätzliche Hosts welche für das Clustersystem unerlässlich sind. Der Hauptserver in unserem Fall der Cluster-Manager wird alle Hosts verwalten und für das Fail-Over zuständig sein. Zudem übernimmt er die Überwachungsaufgabe des ganzen Systems. Die Hosts sollen ein schlankes Betriebssystem erhalten damit sie schnell in Betrieb genommen werden können. Diese Systeme sollen verglichen und evaluiert werden, damit das ganze System möglichst effizient betrieben



werden kann. In der Evaluierung der Hostsysteme gehe ich auf die Grösse des Systems sowie auf die Initialisierungsgeschwindigkeit der nötigen Komponenten für Docker ein. Beim Hauptserver wiederum geht es um Benutzerfreundlichkeit der Verwaltungsaufgaben und die Sicherheit.

#### **3.2.2.2. Hardware**

Die Hardware muss wie bei jeder Software einer Mindestanforderung genügen. Das ist auch bei Docker der Fall. Da eines der Teilziele ein Clustersystem ist, werde ich hier definieren was für eine Hardwareanforderung sinnvoll ist. Der Nutzen von Hardware ist ein entscheidender Faktor um eben Ressourcen genauso einzusetzen wie sie benötigt werden. Bei der Container-Virtualisierung kann man sehr viel Einfluss darauf nehmen, dass keine Ressourcen verschwendet werden. Die Evaluierung der Hardware ist auf die Studie ausgerichtet und wird mehrere Hardwarekomponenten beinhalten, da mehrere Maschinen dazu benötigt werden um einen Cluster aufzubauen.

#### **3.2.2.3. Infrastruktur**

Als Basis für meine Studie soll ein Container mit einem Web Service implementiert werden. Dieser soll dann im Cluster auf mehreren Maschinen laufen und skaliert werden. Es muss also Evaluert werden, wie die Infrastruktur des Clusters aussehen, und welche Services darin laufen sollen. Ist die Grundlage ein Container welcher bereits besteht oder soll ein Web Service Containerfähig gemacht werden? Denkbar wäre auch ein bestehender Container für meine Zwecke umzubauen. Was für einen Nutzen soll der Container haben? Wie soll er implementiert werden? Wie sieht die Infrastruktur aus? Besteht der Container aus einem oder mehreren Containern? Durch diese Evaluierung sollen diese Fragen beantwortet werden.

#### **3.2.2.4. Clustertechnologie**

Um einen Cluster mit Docker zu erstellen gibt es Methoden wie Sand am Meer. Fast wöchentlich tauchen neue oder auf eine bestehende Technologie basierende Methoden im Netz auf. Meine Aufgabe ist es, eine für die Studie passende Technologie zu wählen. Dazu werde ich diese untereinander vergleichen. Wichtig für diesen Punkt ist es, eine stabil laufende und benutzerfreundliche Technologie zu wählen.

Die aktuell bekanntesten Tools dafür möchte ich hier aufzählen:

- Docker Swarm
- Google Kubernetes
- Apache Mesos
- Rancher Cattle

### **3.2.2.5. Analysemethodik**

Im Vorfeld muss ich evaluieren, wie ich die Performance des Clusters und der Skalierung testen und überwachen kann. Da ich ein Clustersystem aufbaue, soll auch getestet werden was bei einem Ausfall einer Maschine geschieht. Um dies zu analysieren, benötige ich diverse Tools. Ein Stress-Test Tool um den Service zu überlasten und ein Monitoring-System welches die CPU und die Netzauslastung der Container überwacht und rapportiert.

### **3.2.3. Realisierung der Infrastruktur**

Nachdem alle Komponenten bewertet worden sind, komme ich zu der Realisierung des Systems. Daraus wird eine Schritt für Schritt Anleitung und gleichzeitig der Hauptbestandteil des Kompendiums. Ich werde grossen Wert darauflegen, alle Schritte möglichst genau zu beschreiben um mich an die Evaluierung welche ich machen werden zu halten. Es ist gut möglich, dass es kleine Abweichungen geben kann, jedoch sollten diese Abweichungen so gering wie möglich gehalten werden. Nach diesem Kapitel der Studie, sollte es Personen mit Informatikkenntnissen möglich sein, diese Infrastruktur komplett nachzubauen.

### **3.2.4. Analyse des Systems**

Das Schlusskapitel meiner Studie ist die Analyse der ganzen Infrastruktur, welche realisiert werden soll. Dies setzt voraus, dass die Realisierung erfolgreich war. Hier wird sich zeigen, ob die Infrastruktur des Clusters auch wirklich einen Nutzen hat. Hier muss ich mich komplett überraschen lassen, da ich bisher ein solche Infrastruktur noch nie aufgebaut habe. Ich erhoffe mir natürlich durch den Cluster eine bessere Performance, aber ich werde trotzdem ganz neutral an die Sache herangehen um das zu beurteilen.

### **3.2.5. Erfolgskriterien**

- Bewilligung für die Studie erhalten
- Alle Komponenten erfolgreich evaluiert
  - Betriebssysteme
  - Hardware
  - Infrastruktur
  - Clustertechnologie
  - Analysemethodik

- Infrastruktur erfolgreich realisiert
  - Aufsetzen der Betriebssysteme
  - Docker-Systeme aufgesetzt
  - Container implementiert
  - Clustersystem lauffähig
- Analyse des Systems abgeschlossen
  - Analysesoftware installiert
  - Auswertung des Systems gemacht
- Kompendium geschrieben