



Berufsbegleitender Studiengang
Wirtschaftsinformatik, 3. Semester

Hausarbeit
im Rahmen der Lehrveranstaltung
IT-Infrastruktur

über das Thema

**Kubernetes - Eine Revolution bei der Bereitstellung von
Softwarelösungen**

Betreuer : Dr. Arthur Dill

Autor : Carlo Nölle
Matrikelnr. : 492765
Sürther Hauptstraße 47
50999 Köln

Abgabe : 27. Februar 2020

Inhaltsverzeichnis

1	Einleitung	1
1.1	Aktualität und Relevanz	1
1.2	Zielsetzung	1
1.3	Aufbau der Arbeit	1
1.4	Hinweise	1
2	Der Hintergrund	2
3	Fazit	4
	Anhang	5

Abbildungsverzeichnis

1	Verzeichnisstruktur der \LaTeX -Dateien	2
---	--	---

Tabellenverzeichnis

Abkürzungsverzeichnis

1 Einleitung

Diese Arbeit umfasst das Themengebiet rund um die Softwarelösung Kubernetes, welche das Bereitstellen und Pflegen von weiterer Software vereinfacht. Es wird hierbei ein flexibler und skalierbarer Ansatz gewählt der sich von anderen Techniken abhebt.

1.1 Aktualität und Relevanz

Steigende Nachfrage sorgt für immer neue Herausforderungen in der Art und Weise wie eine App (Applikation) oder Website online bleibt. Wer durch Überlastung offline geht verliert potenzielle Kunden. Kubernetes ist das Trendthema Nummer 1 für Systemintegranten und Infrastrukturbetreibende und wird von vielen Marktführern eingesetzt. Dadurch und durch die Open Source Lizenz unter dem das Projekt steht ist die Software für viele interessant.

1.2 Zielsetzung

Kubernetes sollte heute an keinem Softwareunternehmen vorbeigehen. Natürlich ist der Aufwand immer dem Nutzen entgegen zu setzen. Mittels dieser Arbeit wird versucht einen Vergleich zu herkömmlichen Methoden zu ziehen und einen Ausblick auf die Chancen und Risiken des Einsatzes der Software zu geben.







1.3 Aufbau der Arbeit

TODO

1.4 Hinweise

Im weiteren Verlauf dieser Hausarbeit wird aus Gründen der besseren Lesbarkeit das generische Maskulinum verwendet. Weibliche und anderweitige Geschlechteridentitäten werden dabei ausdrücklich mitgemeint, soweit es für die Aussage erforderlich ist.

Abbildung 1: Verzeichnisstruktur der \LaTeX -Dateien

Name	Änderungsdatum	Typ	Größe
 abbildungen	29.08.2013 01:25	Dateiordner	
 kapitel	29.08.2013 00:55	Dateiordner	
 literatur	31.08.2013 18:17	Dateiordner	
 skripte	01.09.2013 00:10	Dateiordner	
 compile.bat	31.08.2013 20:11	Windows-Batchda...	1 KB
 thesis_main.tex	01.09.2013 00:25	LaTeX Document	5 KB

Quelle: Eigene Darstellung

- Einleitung - Worum geht es (Ist-Situation) - Warum wird es benötigt (Kubernetes vs. VMs) - Chancen und Risiken - Änderung bestehender Prozesse - Ausgewählte Aspekte (Lösung Vorstellen) - Umziehen bestehender Infrastruktur - Enterprise Readiness - Lösung erklären (Handlungsaspekte) - Neue Aufgabenverteilung für Admin und Entwickler (Devops) - Skalierbarkeit - Fazit

2 Der Hintergrund

In der Softwareentwicklung gibt es irgendwann den Punkt, bei dem man eine fertige Website oder App programmiert hat. Nun gilt es, dieses Produkt erreichbar zu machen, entweder nur ausgewählten Personen oder gleich der gesamten Öffentlichkeit. Diesen Schritt nennt man Bereitstellung, oder auch zu Englisch Deployment. In dieser Arbeit fokussieren wir uns auf eine Website, die der Öffentlichkeit zugänglich gemacht werden soll.

Diese fertige Website soll über das öffentliche Internet, auch WWW (World-wide-web), von jedem internetfähigen Gerät abrufbar gemacht werden. Dazu benutzt man einen Computer, der die Website bereit hält um sie an Endgeräte auszuliefern. Man nennt solch einen Computer Server, und Endgeräte sind die Clients. Server für Websites nennt man Webserver.

Ein solcher Server hat jedoch harte Grenzen was die Kapazität betrifft. Man spricht von Last auf den Server, welche durch zu viele Clients die gleichzeitig auf der Website sind erzeugt wird. Wann diese Grenzen erreicht sind, hängt davon ab, auf wie viel Hardwareressourcen der Server zurückgreifen kann, also Arbeitsspeicher (RAM) und Prozessor- (CPU) Leistung. Der Server muss die Anfragen verarbeiten können die er erhält, und wenn die

Ressourcen schon durch andere Arbeitsschritte ausgelastet sind kann der Server neue Anfragen nicht mehr entgegennehmen.

Eine nicht stark frequentierte Website braucht keinen starken Server, jedoch besteht immer die Möglichkeit, dass durch verschiedene Faktoren der sogenannte Traffic, die Menge Daten die über das Netzwerk geschickt werden, rapide ansteigt und somit der Server überlastet wird. Durch nicht beantwortete Anfragen kann bei einem Onlineshop beispielsweise Geld verloren gehen, was natürlich nicht passieren soll.

Ein anderes Problem ist eigentlich ein ganz simples. Wenn die Hardware auf dem der Server läuft einen Defekt hat, ist die Website gar nicht mehr erreichbar. Um dieses Problem zu umgehen dupliziert man einfach den Server auf eine andere Hardwarebasis. Somit hat man zwei Server, die die gleiche Website hosten (bereit halten).

Nur funktioniert das Internet so, dass eine Domain (Url, z.B.: beispiel.de) auf eine Adresse im Internet zeigt. Solch eine Adresse nennt man IP, und sie ist einzigartig im gesamten Netz. Da wir aber jetzt auf eine Adresse zeigen, können somit nicht zwei Server antworten, sondern man landet immer bei dem der die Adresse trägt. Dafür gibt es dann Reverse-Proxy's, oder auch Load-Balancer. Im Grunde machen die nichts anderes, als eine Adresse zu haben, und die ankommende Anfrage an definierte Ziele weiter zu leiten.

Der Name Load-Balancer nimmt den positiven Nebeneffekt schon vorweg, den solch ein Proxy kann eine Anfrage nicht bloß zufällig an eine von n vielen Adressen weiterleiten, sondern auch nach verschiedenen Faktoren agieren. Beispielsweise kann der Load-Balancer sich merken, wie viele Verbindungen gerade auf einem Server laufen, um neue besser an einen Server weiter zu leiten der weniger zu tun hat. Somit sind die zwei Hauptprobleme beseitigt.

Das soeben erklärte Prinzip nennt man Horizontale Skalierung, da ein Server neben den anderen gestellt wird (Horizontal), um somit mehr Last abfangen zu können (Skalierung). Vertikale Skalierung ist sehr ähnlich, man verteilt die Last auf mehrere Server, jedoch nicht auf verschiedene Hardware. Man installiert z.B. zwei Webserver auf einem Computer. Dies kann einen Vorteil bringen was die Lastverträglichkeit angeht, da manchmal eine Software nicht die kompletten Hardwareressourcen alleine benutzen darf, jedoch bringt es keine Ausfallsicherheit.

Durch Horizontale Skalierung erhält man also Ausfallsicherheit, da die Website redundant verfügbar ist, und die Last wird gleichmäßig verteilt auf alle Maschinen, wobei jederzeit weitere eingegliedert werden können. Ausfallsicherheit wird auch Hochverfügbarkeit genannt, da selbst im Falle eines Hardwaredefekts die Website immernoch erreichbar wäre. Wahre Hochverfügbarkeit würde erreicht werden, wenn die Replikate des Servers noch

an verschiedenen physischen Standorten stehen würden, wodurch man Faktoren höherer Gewalt wie Stromausfällen davon kommt. Infrastruktur, so wird die Hardware/Software Lösung hinter der Website/App genannt, ist ein großes Thema und es gibt viele verschiedene Ansätze die gleichen Probleme zu lösen.

3 Fazit

Wünsche Euch allen viel Erfolg für das 7. Semester und bei der Erstellung der Thesis. Über Anregungen und Verbesserung an dieser Vorlage würde ich mich sehr freuen.

Anhang

Anhang 1: Beispielanhang

Dieser Abschnitt dient nur dazu zu demonstrieren, wie ein Anhang aufgebaut sein kann.







Anhang 1.1: Weitere Gliederungsebene

Auch eine zweite Gliederungsebene ist möglich.

Anhang 2: Bilder

Auch mit Bildern. Diese tauchen nicht im Abbildungsverzeichnis auf.

Abbildung 2: Beispielbild

Name	Änderungsdatum	Typ	Größe
 abbildungen	29.08.2013 01:25	Dateiordner	
 kapitel	29.08.2013 00:55	Dateiordner	
 literatur	31.08.2013 18:17	Dateiordner	
 skripte	01.09.2013 00:10	Dateiordner	
 compile.bat	31.08.2013 20:11	Windows-Batchda...	1 KB
 thesis_main.tex	01.09.2013 00:25	LaTeX Document	5 KB

Ehrenwörtliche Erklärung

Hiermit versichere ich, dass die vorliegende Arbeit von mir selbstständig und ohne unerlaubte Hilfe angefertigt worden ist, insbesondere dass ich alle Stellen, die wörtlich oder annähernd wörtlich aus Veröffentlichungen entnommen sind, durch Zitate als solche gekennzeichnet habe. Ich versichere auch, dass die von mir eingereichte schriftliche Version mit der digitalen Version übereinstimmt. Weiterhin erkläre ich, dass die Arbeit in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde/Prüfungsstelle vorgelegen hat. Ich erkläre mich damit **einverstanden/nicht einverstanden**, dass die Arbeit der Öffentlichkeit zugänglich gemacht wird. Ich erkläre mich damit einverstanden, dass die Digitalversion dieser Arbeit zwecks Plagiatsprüfung auf die Server externer Anbieter hochgeladen werden darf. Die Plagiatsprüfung stellt keine Zurverfügungstellung für die Öffentlichkeit dar.

Köln, 27.2.2020

(Ort, Datum)

A handwritten signature in black ink, consisting of a large, stylized 'H' followed by a series of loops and a final flourish.

(Eigenhändige Unterschrift)