

Hochschule Bremerhaven

Fachbereich II
Management und Informationssysteme
Wirtschaftsinformatik B.Sc.

Modul
Vernetzte Systeme

Analyse verschiedener Dockerkonstellationen

Vorgelegt von:	Junior Ekane	MatNr. 40128
	Florian Quaas	MatNr. 39952
	Herman Tsago	MatNr. 00000
	Steve Aguiwo II	MatNr. 40088

Vorgelegt am: 19. Februar 2024

Dozent: Prof. Dr. Oliver Radfelder

Dozentin: Prof. Dr. Karin Vosseberg

Inhaltsverzeichnis

1	Einleitung	3
1.1	Vorstellung des Moduls Vernetzte Systeme	3
2	Projektablauf	3
2.1	Vorstellung der Projektidee	3
2.2	Projektorganisation und Teamarbeit	3
3	Werkzeuge	3
3.1	Verwendete Werkzeuge	3
4	Docker	6
4.1	Was ist ein Docker?	6
4.2	Was ist ein Netzwerk?	6
4.3	Was ist ein Image?	6
4.4	Was ist ein Container?	6
4.5	Noch ein Punkt?	6
4.6	Wie haben wir Docker verwendet?	6
5	Ergebnis	6
5.1	Etwas	6
5.2	noch etwas	6
6	Lasttest	6
6.1	Was haben wir analysiert?	6
6.2	Wie haben wir das analysiert?	6
6.3	Endergebnis	6
7	Reflexion	6
8	Fazit	6
	Literaturverzeichnis	7
	Selbstständigkeitserklärung	7

1 Einleitung

1.1 Vorstellung des Moduls Vernetzte Systeme

2 Projektablauf

2.1 Vorstellung der Projektidee

2.2 Projektorganisation und Teamarbeit

Für unsere Auseinandersetzung zum Thema "Analyse verschiedener Dockerkonstellationen" im Rahmen der Veranstaltung "Vernetzte Systeme" haben wir uns gezielt für einen strukturierten Projektplan, der sich an agilen Methoden orientiert. Zum Entwerfen und Fertigstellung haben wir einen Sprint von drei Wochen gemacht inklusiv das Verfassen vom Bericht. Regelmäßig haben wir uns alle zwei Tage getroffen und da wurde immer über die Fortschritte gesprochen und neue Aufgabe verteilt. Die ersten Tage habe wir erstmal düber ausgetauscht, was wir machen wollen und wie wir es tun möchten?

Die Implementierung haben wir in mehreren iterativen Phasen unterteilt, wobei das Kanban-System zur Visualisierung und Verwaltung des Arbeitsflusses eingesetzt wurde. Hierfür haben wir die Webanwendung Trello(Verlinkung fehlt) verwendet. In der App haben wir über einen Projekt-Dashboard verfügt, auf dem unsere Aufgaben in drei Kategorien aufgelistet waren wie:

- Zu erledigen
- In Arbeit
- Erledigt

Dies ermöglichte eine flexible Anpassung an wechselnde Anforderungen und eine kontinuierliche Verbesserung der Implementierung. Die agilen Prinzipien unterstützten uns dabei, schnell auf Herausforderungen zu reagieren und eine effektive Implementierung sicherzustellen.

3 Werkzeuge

3.1 Verwendete Werkzeuge

Bei so einem Projekt, da wo es draun geht mit mehreren entfernten Systemen zu arbeiten, braucht man genauso wie bei anderen Projekten auch Werkzeuge, mit denen man seinen Entwurf

auf die Beine bringen kann.

Hierfür hatten wir schon aus der Vorlesung einige Vorgaben bekommen wie zum Beispiel die Verwendung einer Docker-Umgebung, woran wir uns auch entsprechend eingehalten haben. aber damit unser Entwurf unserer Vorstellung entspricht haben wir auch noch andere Werkzeuge gebraucht, damit wir zum Beispiel unsere mit der Programmiersprache Java entwickelte Mini-Anwendung laufen lassen können, Dateneingaben speichern können, das Einkommen von mehreren Anfragen zu gleicher Zeit verwalten können und auch dann das Verhalten unserer Mini-Anwendung beobachten können. Hier sind unsere Werkzeuge aufgelistet:

- **Git:** Es ist tatsächlich so, dass man oft alleine entwickelt, wenn es sich um solche minimale Anwendungen wie bei uns handelt, doch ist es aber auch nicht ungewöhnlich, dass man mit anderen Leuten kooperieren muss und wenn dies der Fall ist, muss auf irgendeine Art sichergestellt werden, dass alle Leute auf einem gemeinsamen Ordner arbeiten, dessen Inhalt jeder verwalten und auch nach Bedarf zurücksetzen kann. Hierzu ist Git das perfekte Werkzeug, denn es ermöglicht eine Versionsverwaltung von Dateien, sodass man auch auf einen vergangenen Inhalt zugreifen kann, wenn gemerkt wird, dass etwas schiefgelaufen ist oder dass man etwas gelöscht hat, das im Code von großem Belang war.
- **Apache Tomcat:** Es handelt sich hierbei um eine Open-Source-Implementierung verschiedener Spezifikationen wie etwa der **jakarta Servlet**, **jakarta Anmerkungen**, **jakarta Authentifizierung** und noch mehr. Es gehört der Apache Software Foundation. Mit einem Tomcat-Server wird möglich eine Anwendung, die ursprünglich in der Programmiersprache Java geschrieben wurde, als Webanwendung laufen zu lassen, was bei uns tatsächlich erforderlich war.
- **Docker:** Es ist ein revolutionäres Tool, das die Art und Weise, wie Anwendungen entwickelt, ausgeliefert und ausgeführt werden, verändert hat. Es ermöglicht Entwicklern, ihre Anwendungen samt Abhängigkeiten in sogenannten Containern zu verpacken. Diese Containerisierung erleichtert die Bereitstellung und den Betrieb von Anwendungen, da sie auf jedem System laufen können, das Docker unterstützt, unabhängig von der Umgebung. Dies sorgt für Konsistenz über Entwicklung, Test und Produktion hinweg und vermeidet das berühmte „Bei mir lokal funktioniert es“ Problem. Für unser Projekt ist Docker essentiell, da es uns ermöglicht, verschiedene Konstellationen mit MariaDB, Redis und anderen Tools effizient zu orchestrieren und zu analysieren.
- **MariaDB:** Dieses Werkzeug muss keiner mehr vorgestellt werden, der sich schon zumindest einmal mit Datenbanken auseinander gesetzt hat. MariaDB ist eine sehr berühmte Open-Source-Datenbank, die als Abspaltung von MySQL entstanden ist, mit dem Ziel, vollständige Open-Source-Freiheit zu gewährleisten. Sie bietet eine hohe Kompatibilität mit MySQL, was bedeutet, dass Anwendungen, die für MySQL geschrieben sind, in der Regel ohne Änderungen mit MariaDB funktionieren. In unserem Projekt nutzen wir

MariaDB zur Speicherung von Daten, die von unserer Servlet-Anwendung verwendet werden.

- **Redis:** Es geht hierbei um ein in-memory Datenstrukturspeicher also eine NoSQL-Datenbank, der als Datenbank, Cache und Message Broker verwendet werden kann. Es unterstützt Datentypen wie Strings, Hashes, Listen, Sets und mehr, was es sehr flexibel macht. Redis ist bekannt für seine hohe Geschwindigkeit und Effizienz bei Operationen mit Daten im Speicher. Bei uns hatten wir die Idee gehabt, dass wir mit Redis eine Warteschlange bauen können. redis kennen wir schon zusammen mit MariaDB und K6 aus dem zweiten Semester, in dem wir eine kurze Einführung in jeden Themen gehabt haben.
- **Tcpdump:** Aus der vorletzten Veranstaltung kennengelernt ist Tcpdump ein mächtiges Kommandozeilen-Tool zur Netzwerküberwachung und -analyse. Es ermöglicht es, den Datenverkehr auf einem Netzwerk zu erfassen und zu analysieren, was für die Diagnose von Netzwerkproblemen oder für Sicherheitsanalysen unerlässlich ist. In unserem Projekt verwenden wir tcpdump, um den Datenverkehr zwischen den verschiedenen Containern zu analysieren und zu verstehen, wie unsere Anwendungen kommunizieren und wie sich die Netzwerkleistung auf die Gesamtleistung der Anwendung auswirkt.
- **Ping:** Sehr beghert ist Ping ein einfaches, aber äußerst nützliches Netzwerk-Utility, das verwendet wird, um die Verfügbarkeit und die Latenz (Ping-Zeit) zwischen zwei Netzknoten zu testen. Es sendet ICMP „Echo Request“-Nachrichten an das Ziel und erwartet „Echo Reply“-Nachrichten zurück. In unserem Docker-basierten Setup nutzen wir Ping, um die Netzwerkverbindung zwischen den Containern sowie die Verbindung zur Außenwelt zu überprüfen, was für die Fehlersuche und Leistungsoptimierung unerlässlich ist.
- **K6:** Es ist ein modernes, Open-Source-Performance-Testing-Tool, das für Entwickler konzipiert wurde, um die Leistung und Skalierbarkeit von Anwendungen in einer DevOps-Umgebung zu testen. Es ermöglicht das Schreiben von Tests in JavaScript, was die Erstellung realistischer Test-Szenarien vereinfacht. In unserem Projekt verwenden wir K6, um die Belastbarkeit und Performance unserer Dockerkonstellationen unter verschiedenen Bedingungen zu testen, um sicherzustellen, dass unsere Anwendungen auch unter Last zuverlässig und effizient laufen. Effektiv wurde K6 für die Testphase bei uns verwendet. Mehr dazu unten bei Lasttest.

Mit diesen Werkzeugen, die uns entweder sehr vertraut oder mit denen wir noch einstiegen, haben wir unsere eigene Wege gefunden, wie die mit einander kombiniert werden, um unsere Projektidee umsetzen zu können und Erfolg zu erzielen.

4 Docker

4.1 Was ist ein Docker?

4.2 Was ist ein Netzwerk?

4.3 Was ist ein Image?

4.4 Was ist ein Container?

4.5 Noch ein Punkt?

4.6 Wie haben wir Docker verwendet?

5 Ergebnis

5.1 Etwas

5.2 noch etwas

6 Lasttest

6.1 Was haben wir analysiert?

6.2 Wie haben wir das analysiert?

6.3 Endergebnis

7 Reflexion

8 Fazit

Selbstständigkeitserklärung

Wir versichern, die von uns vorgelegte Arbeit selbstständig verfasst zu haben. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder nicht veröffentlichten Arbeiten anderer entnommen sind, haben wir als entnommen kenntlich gemacht. Sämtliche Quellen und Hilfsmittel, die wir für die Arbeit benutzt haben, sind angegeben. Die Arbeit haben wir mit gleichem Inhalt bzw. in wesentlichen Teilen noch keiner anderen Prüfungsbehörde vorgelegt.

Bremerhaven, den 19. Februar 2024

Unterschrift:

Junior Ekane

Florian Quaas

Steve Aguiwo

Herman Tsago