

Kurs: TINF13B4

Inhalt

- [1 Teilnehmer](#)
- [2 Aufgabenbeschreibung](#)
- [3 Funktionsbeschreibung / Technologien](#)
 - [3.1 Übersicht](#)
 - [3.2 Onlineshop](#)
 - [3.3 Datenbank](#)
 - [3.3 Server](#)
- [4 Fazit](#)

1 Teilnehmer

Andreas Bissinger, Sidney Eisenhut

2 Aufgabenbeschreibung

Während der Durchführung des Kurses “Verteilte Systeme” im 6. Semester, sollte ein Internetshop erstellt werden, welcher als verteiltes System aufgesetzt wird. Folgende Requirements werden gestellt:

- Internetshop
 - ◆ Benutzersystem, Warenkorb
 - ◆ Bestellen, Einsehen von laufenden Bestellungen und deren Bestätigung
- Ausfallsicheres CDN-System
 - ◆ horizontales Skalieren Begründung: Schutz vor vielbesuchten Webseiten, die bei Nennung des Shops zu viel Druck ausüben und das System lahm legen (als Bsp. heise.de genannt)
 - ◆ Mindestausstattung an Servern, Möglichkeit der Zuschaltung weiterer Server (bei Bedarf, so schnell wie möglich)
 - ◆ ständige Verfügbarkeit der Anwendung
 - ◆ vollständige Absturz- und Zugriffssicherheit auch bei fehlerhafter Benutzung
 - ◆ auch bei Absturz (z.B. Hardwarefehler) eines Servers sollen Benutzer ihre Session nicht verlieren

Das Endergebnis wird im Laufe der letzten Vorlesung vorgestellt. Deadline: 14.04.2016

3 Funktionsbeschreibung / Technologien

3.1 Übersicht

Für die Umsetzung des Verteilten Systeme-Projekts wurden unterschiedliche Technologien diskutiert.

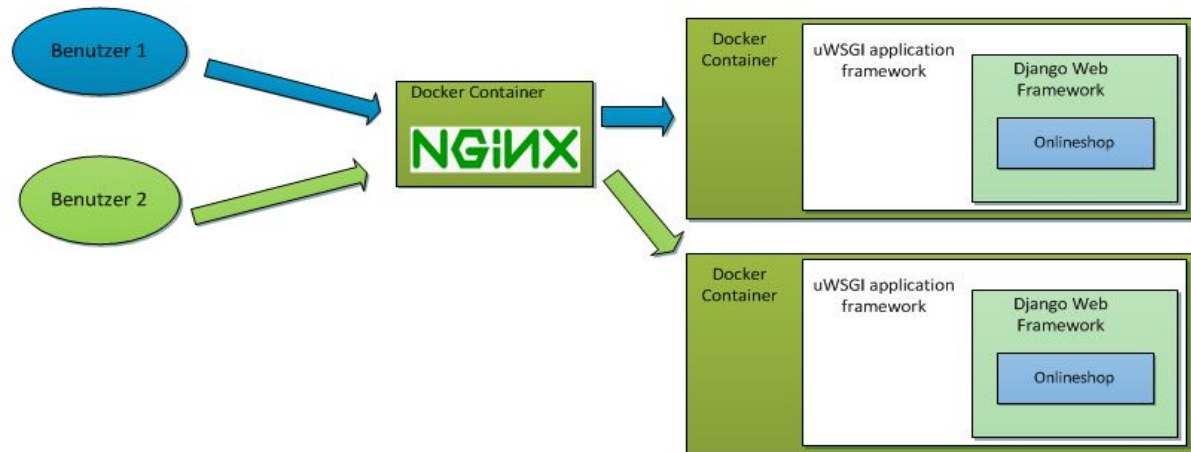


Abbildung 1: Benutzerzugriff auf die Application Server Quelle: eigene Abbildung

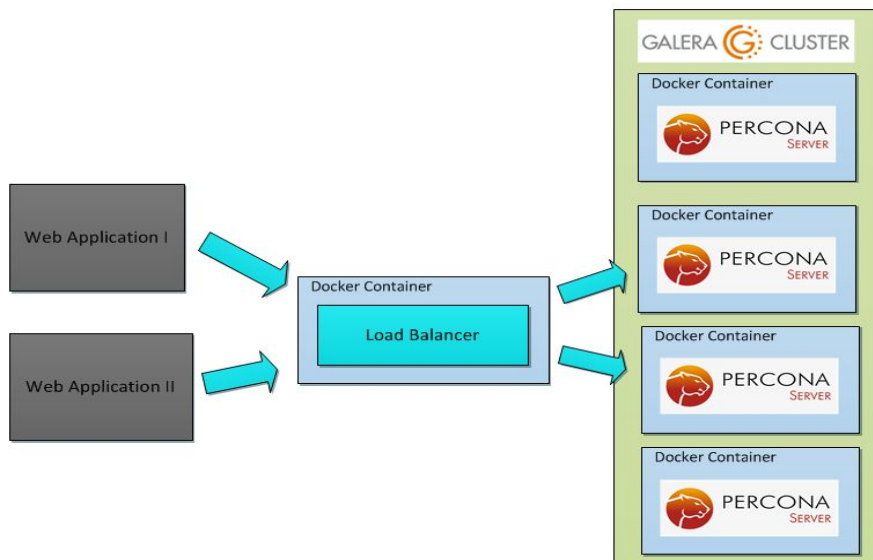


Abbildung 2: Zugriff der Web Applications auf die Datenbank als verteiltes Cluster

Verwendet wurden hierfür folgende Technologien, die im Folgenden genauer dargestellt werden:

- Galera Percona Cluster
- Django web framework
- Docker
- uWSGI
- nginx load balancer

3.2 Internetshop

Für den Internetshop wurden folgende Technologien verwendet:

- HTML
- CSS
- JavaScript

Kombiniert wurden diese Technologien durch die Verwendung des Web Application Frameworks Django. Programmiersprache für Django ist Python, was auch im Backend verwendet wird.

Der Internetshop bietet die folgenden geforderten Funktionalitäten:

- Waren können angezeigt werden
- User können sich im Internetshop einloggen
- Session-Handling solange der User eingeloggt ist
- Warenkorb zur Speicherung ausgewählter Produkte
- Bestellungen können aufgegeben werden, wobei Verbindungen zur Datenbank bzw. dem Datenbankcluster hergestellt werden

3.3 Datenbereitstellung:

Zur Bereitstellung der Daten, mit welchen der Onlineshop gespeist werden soll, findet das multi-master Datenbank Cluster System Galera Cluster Anwendung. Dieses dient der Verwaltung und transparenten Operationen auf den Datenbanken, wobei sich die Webshop Anwendung nicht darum kümmert, welche Datenbank angesprochen wird. Die eigentlichen Datenbanken werden durch einen Percona Server, welcher eine Art MySQL Server ist realisiert. Dieses System glänzt mit seinen hervorragende Performanz, Lastenverteilung und Transparenz der Operationen auf die Daten. Außerdem ist gewährleistet, dass nicht das komplette System ausfällt, sollte ein Percona Server aus beliebigen Gründen abstürzen.

3.4 Server-System:

Verwendet werden zwei Arten von Servern: Zum einen die Software Nginx für Web-Server, zum anderen Docker. Mit Hilfe der Module für die Lastenverteilung sowie Proxy kann Nginx als Lastverteiler agieren und User auf die jeweiligen Server, welche auf Dockercontainern laufen, weiterleitet. Auf diesen läuft jeweils der Internetshop.

Über unabhängige Docker-Container laufen folgende Komponenten:

- Load Balancer Nginx
- Application Server mit Webshop
- Percona Datenbankserver
- Galera Cluster Load Balancer (DB)

4 Fazit

Internetshop: Wurde wie gewünscht erstellt, Basisanforderungen werden erfüllt. Sessions werden verwaltet und Bestellungen sind einsehbar.

Skalierbarkeit und Ausfallsicherheit: Das erstellte System ist sehr gut skalierbar. Über den Befehl Docker Run können neue Application Server einfach hinzugefügt werden. Da diese über einen Load Balancer verwaltet werden, werden User beim Ausfall des Systems automatisch auf die Server umverteilt -> ausfallsicher

Downtime: Unser System gewährleistet eine Zero Downtime durch die Verwendung des Load Balancers und einer variablen Anzahl an Servern.