Fachhochschule Nordwestschweiz
Hochschule für Technik

Application Performance Management

FS 2021

Load Balancing

Michael Faes

Inhalt

Persönliches

Modul-Übersicht (1. Teil)

“Cloud”: Definitionen

Load Balancing

- Motivation
- LB-Methoden
- Monitoring (Health Checks)
- Persistenz
- Verschlüsselung

Übung

Persönliches

Michael Faes
michael.faes@fhnw.ch

Wohnort: Basel

Dazwischen: Reisen, Fotografie, Beach Volley, Kino, Jungschar

Bachelor ETH Zürich

Praktikum Canoo AG

Master ETH

Doktorat ETH

- **Programmiersprachen**
- **Parallelismus, Concurrency**
- Software-Engineering-Tools
- Education-Tools

Dozent FHNW 50%

Ausbildung Lehrdiplom Sek II

Modul-Übersicht 1. Teil

Cloud-Basics

- Loadbalancing
- Clustering
- Virtualisierung
- Caching

Cloud aus App-Sicht

Performance in der Cloud

- Mess-Setup, Auswerten, Interpretieren

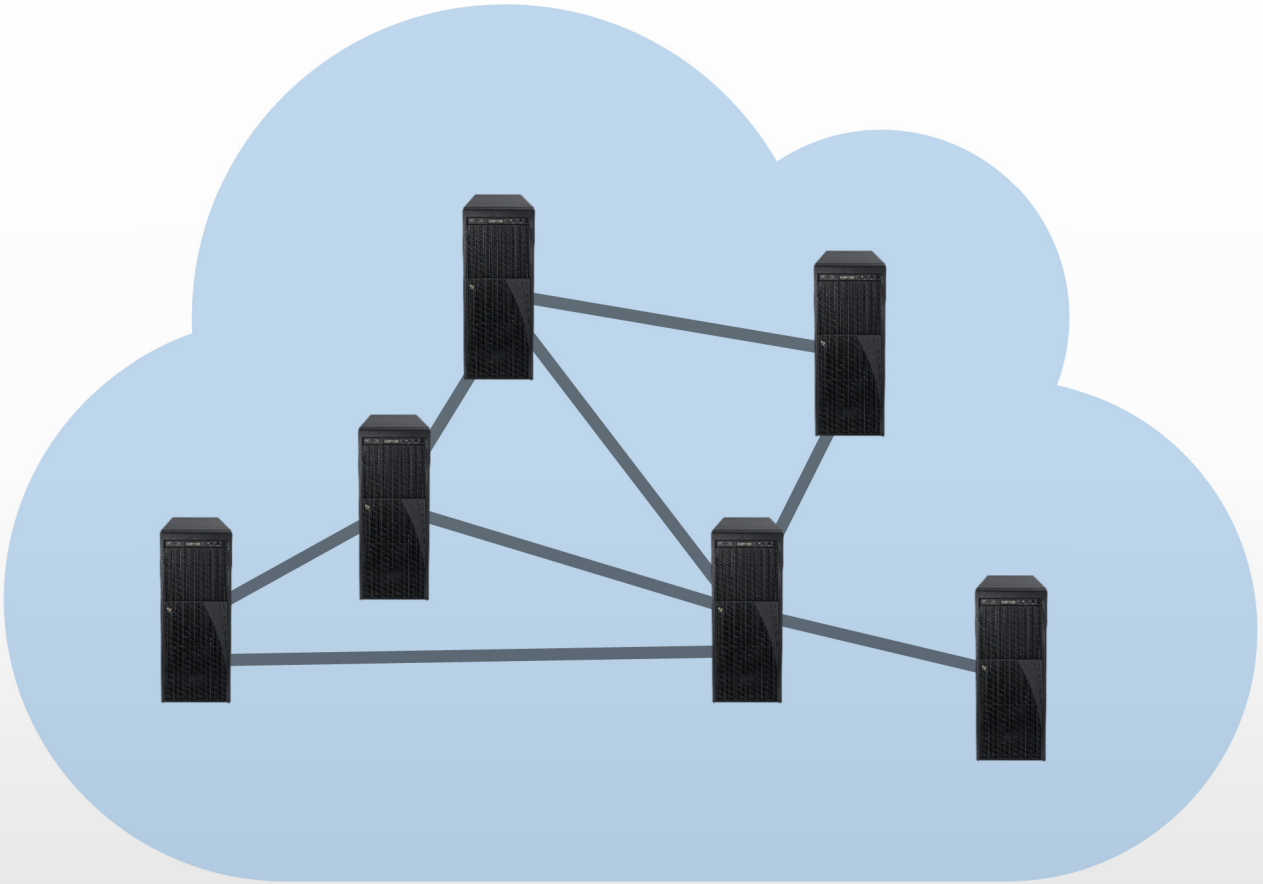
Praktisch!

Aufbau einer Mikro-Cloud

Entwickeln einer (sehr) einfachen Cloud-App

Umgang mit Last-Generator, Messungen durchführen

Was ist eigentlich eine “Cloud”?



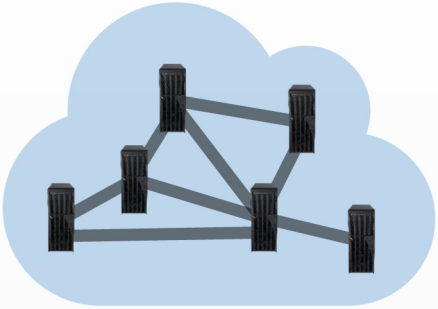
“I don’t like the term ‘cloud’; it’s nebulous.”

– Weissnichmehrwer

Was ist eigentlich eine “Cloud”?

Cloud Computing ist ein

Modell für “Selbstbedienung” von geteilten ***Rechen-Ressourcen*** *über ein Netzwerk. Beschaffung und Abgabe von Ressourcen findet schnell und ohne Provider-Interaktion statt.*

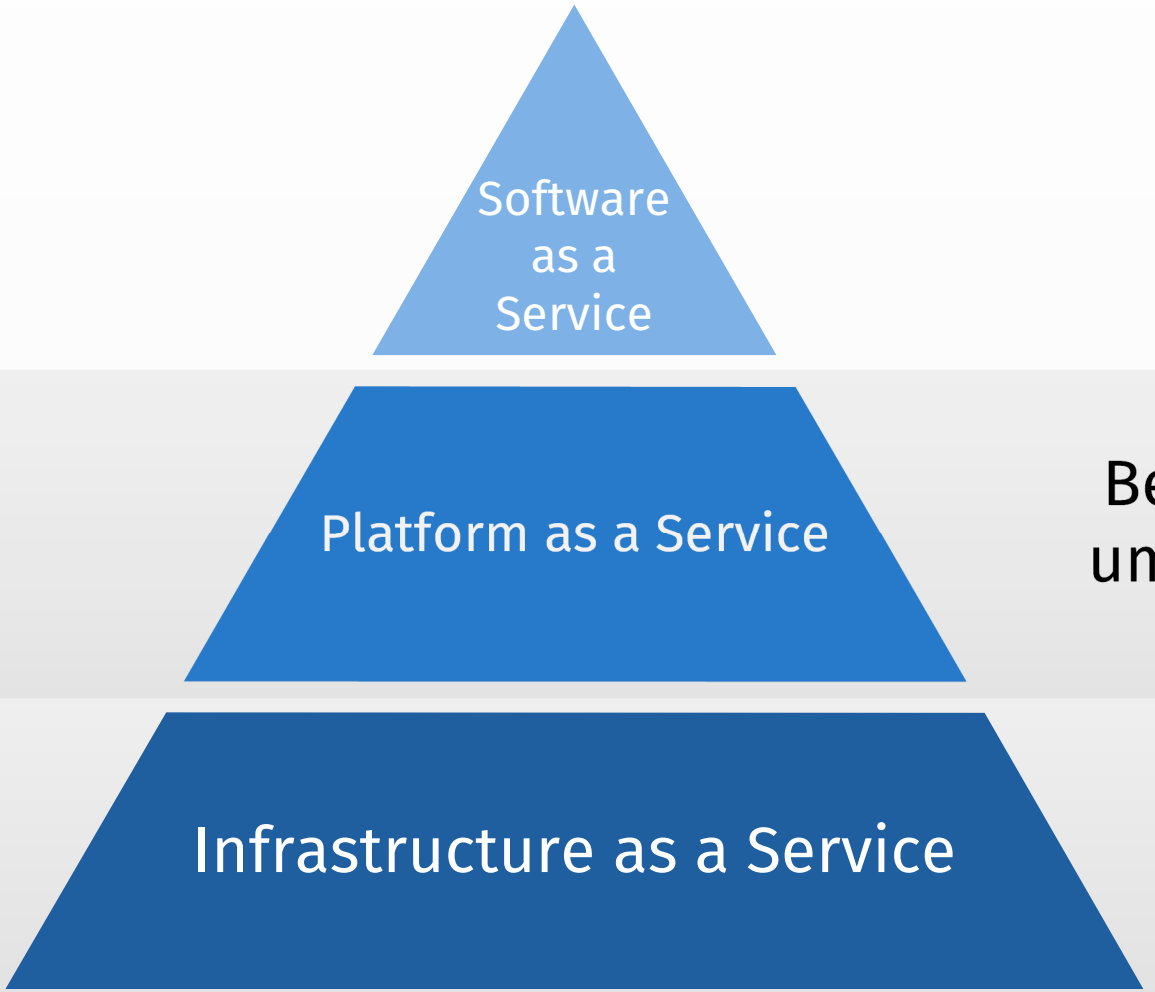


(ungefähre Definition von NIST)

5 Hauptmerkmale:

- Selbstzuweisung von Ressourcen
- Zugriff über Internet (Standard-Protokolle)
- Gemeinsame Ressourcen-Nutzung
- Schnelles Skalieren von Ressourcen
- Messen und Überwachen von Nutzung

Service-Modelle



Applikationen

Betriebssysteme, Laufzeit-umgebungen, Datenbanken

Computer, Netzwerke, Speicherplatz

Load Balacing

welcome to stack overflow

Home

PUBLIC

Stack Overflow

Tags

Users

Jobs

Teams+Slack
Q&A for work

Learn More

CAUTION CAUTION

~UNDER CONSTRUCTION~

Big changes for Y2K!

Go to the future

CAUTION CAUTION

Home

PUBLIC

Stack Overflow

Tags

Users

Jobs

Teams+Slack
Q&A for work

Learn More

How do I prepare for January 1st, 2000?

Ask Question

11

I've been hearing about the Millenium Bug (AKA the Y2K bug) and I'm not sure what to change about my program to prepare for it. I know it has to do with the dates, but there might also be something to do with leap years?

Here's what I know so far:

- On January 1st, 2000, programs that use double digits for years that assume 1900 will change from 1999 to 1900 instead of 2000. That shouldn't happen in my program.
- I should update any dates in my database that assume 1900 to explicitly specify the full year.
- February is a leap year?? So something should change?

Are these all the issues related to Y2K? I've been asking around on IRC, but everyone's giving me different answers.

y2k

share improve this question

edited yesterday

asked 2 days ago
Jon Chan ♦
552 ● 1 ● 5 ● 23

4 letmeaskjeevesthatforyou.com – Adam Lear ♦ 2 days ago

2 My company hired a consultant, but I don't think it'll matter when the apocalypse comes. I'm putting my savings into gold and canned goods. – Jon Ericson ♦ 2 days ago

1 possible duplicate: stackoverflow.com/questions/18200744/... – m0sa ♦ yesterday

Possible duplicate of What Does Y2K Compliant Mean? – Vinko Vrsalovic yesterday

add a comment

asked 2 days ago

viewed 1 times

active yesterday

BLOG

The Next CEO of Stack Overflow

5 Answers

active oldest votes

9

If you store your dates in Unix time as a signed 32-bit binary integer, that'll get you until 03:14:07 UTC on Tuesday, 19 January 2038 which should, like, be totally good enough.

share improve this answer

answered 2 days ago
Meg Risdal
106 ● 1

New contributor

1 That sounds perfect. I'll give that a shot! – Jon Chan ♦ yesterday

1 And if you need more, just store 2 signed 32-bit binary integers side by side in your SQL 7 table. That way you can use one as an overflow register and be good practically forever! – Haney ♦ yesterday

add a comment

Linked

5 What Does Y2K Compliant Mean?

Related

0 Preferred method for handling Y2K date conversions?

2 Win32: How to prevent DateTimePicker from accepting 2-digit years?

1 How Do I Convert a Y2K Date to SQL DATE In SSIS?

8 How to parse string dates with 2-digit year?

question feed

GUESTBOOK

Name:

Comment:

FIIIIIIIRRRSSSSTTTTTTT
@meg - 2019/04/01

I think they got haxxxed
@josh - 2019/04/01

Motivation

Das Web hat sich verändert

- Tausende → 100 Millionen Users
- Schnellere Endgeräte
- **Statisch → Dynamisch**

Web-Apps müssen skaliert werden!

Scale-up

Scale-out

Web-Apps müssen skaliert werden!

Scale-up	Scale-out
Teuer	Relativ günstig, flexibel
Wenig flexibel	Muss App an verteiltes Setting anpassen!
Einfach, App muss (evtl.) nicht angepasst werden	Prinzipiell immer machbar (aber nicht jede App ist parallelisierbar)
Irgendwann ist Grenze erreicht...	**Way to go!**

Scale-out braucht Load Balancing

Last muss auf mehrere Server verteilt werden!

Wie?

Einfache Methode: Benutzer “vorverteilen”

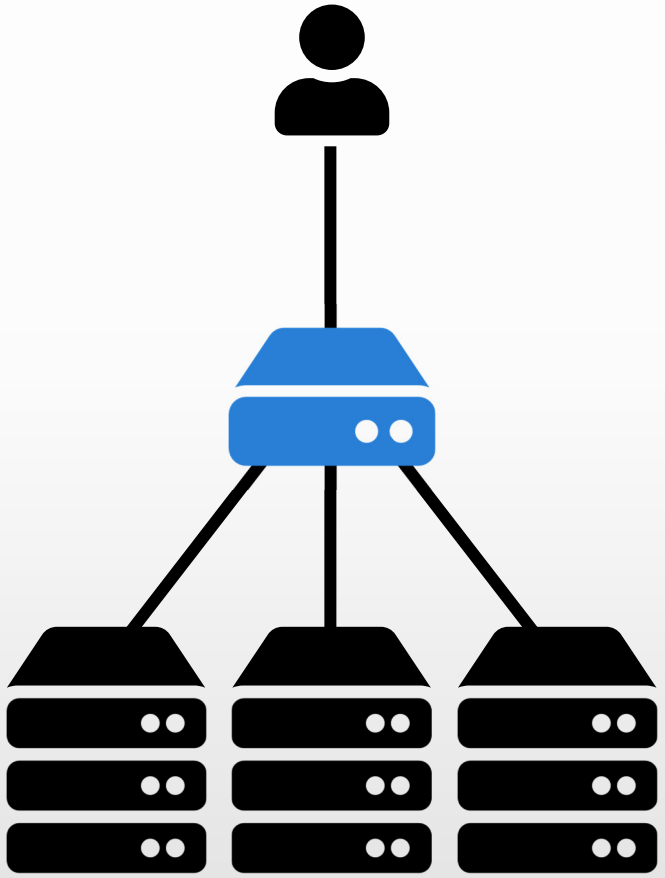
- DNS! Unterschiedliche Server-Adressen verteilen
- Nachteile: Wenig Kontrolle, keine Lösung für hohe Verfügbarkeit

NB: Load Balancing und Verfügbarkeit nicht dasselbe!

“Richtiges” Load Balancing

Eigentliche Server-Requests werden auf mehrere Server verteilt

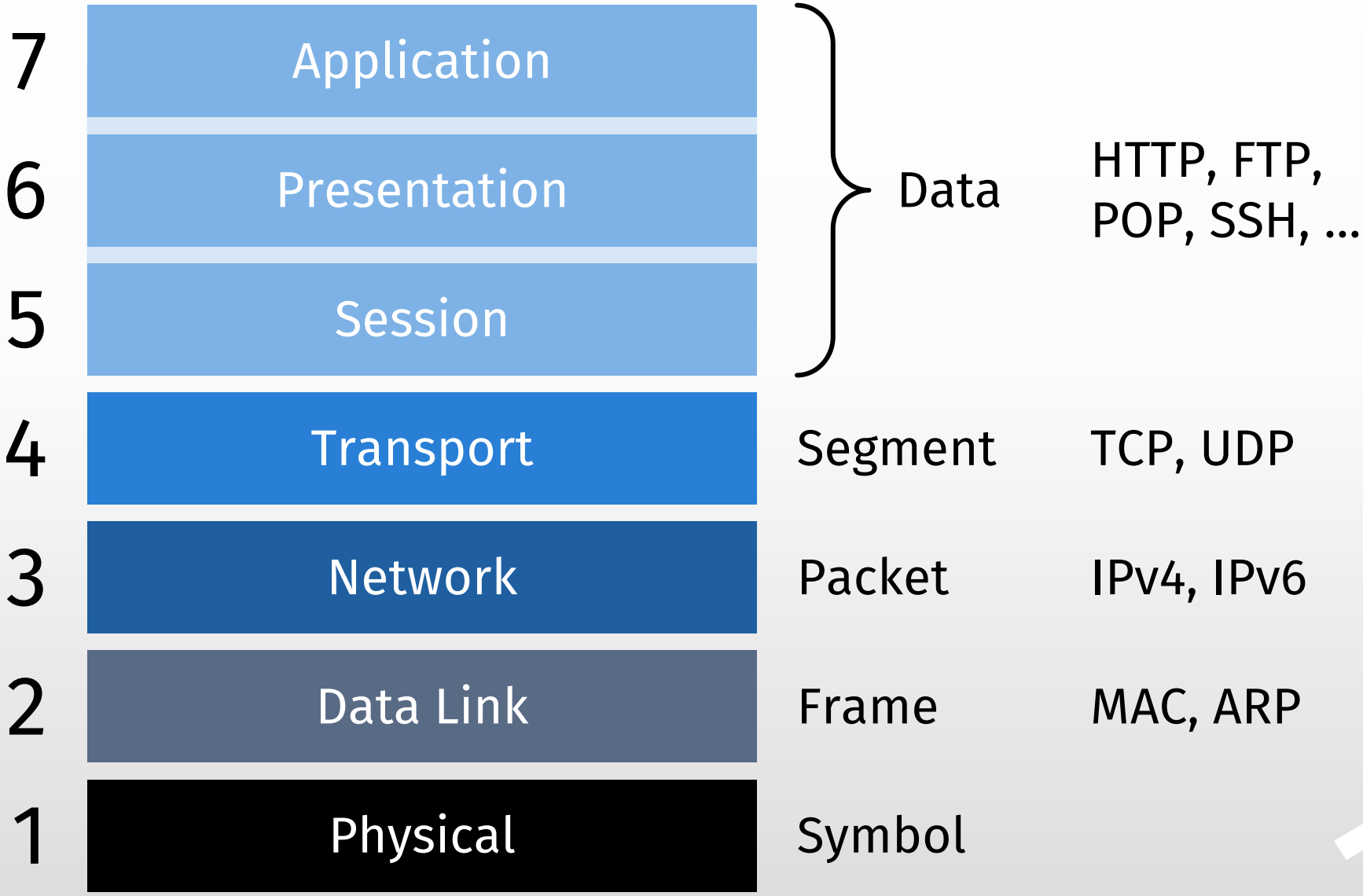
Neue Komponente: **Load Balancer**!



Separate Maschine (evtl. virtuell)

- Einstiegspunkt des Dienstes
- Verteilt eingehende Requests dynamisch
- Hardware- oder Software-basiert

Refresher: OSI Modell



LB-Methode: Direct Routing (L2/3)

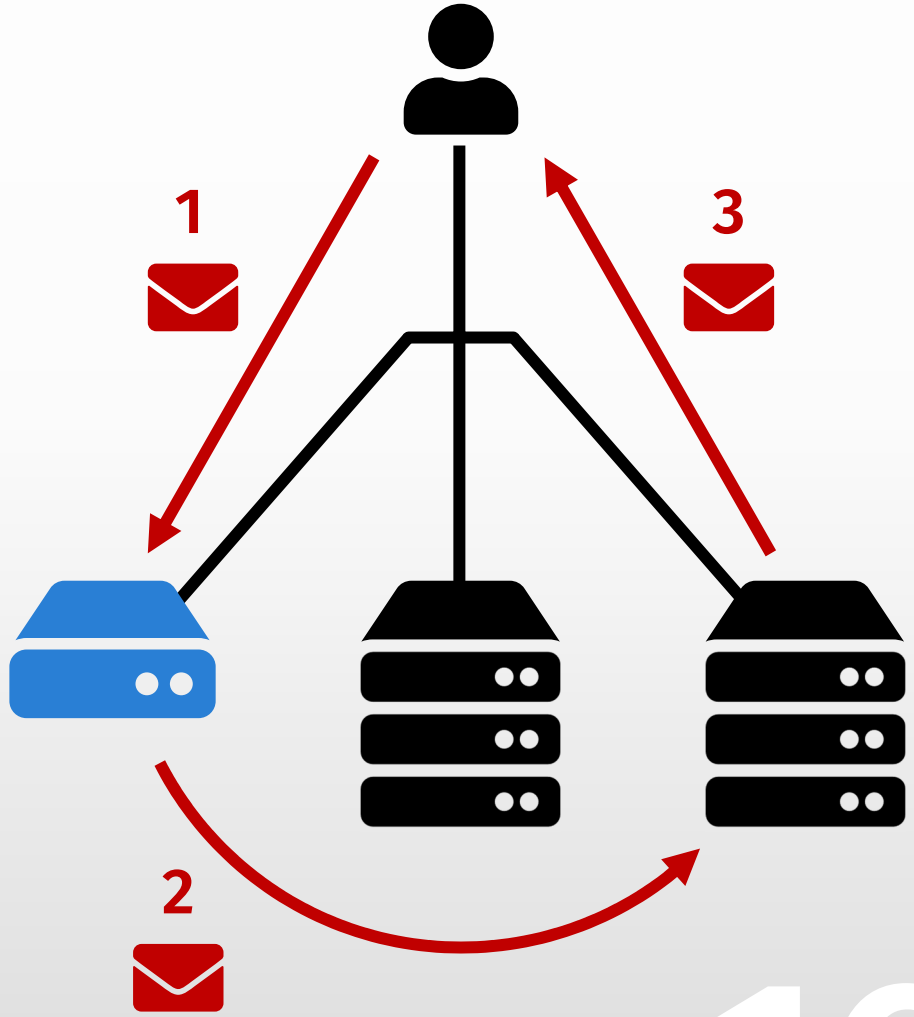
LB und Server im Gleichen Subnetz, alle haben gleiche IP konfiguriert

- Nur LB antwortet auf ARP-Requests, d.h. alle Packets landen bei ihm

LB ändert Frame, schickt Request an Server weiter

Server antwortet direkt an Client

Vorteil: Oberhalb von IP-Level (Layer 3+) muss nichts am Request geändert werden!



LB-Methode: NAT (Layer 3/4)

LB “trennt” Client von Server, welche in privatem Subnet sind

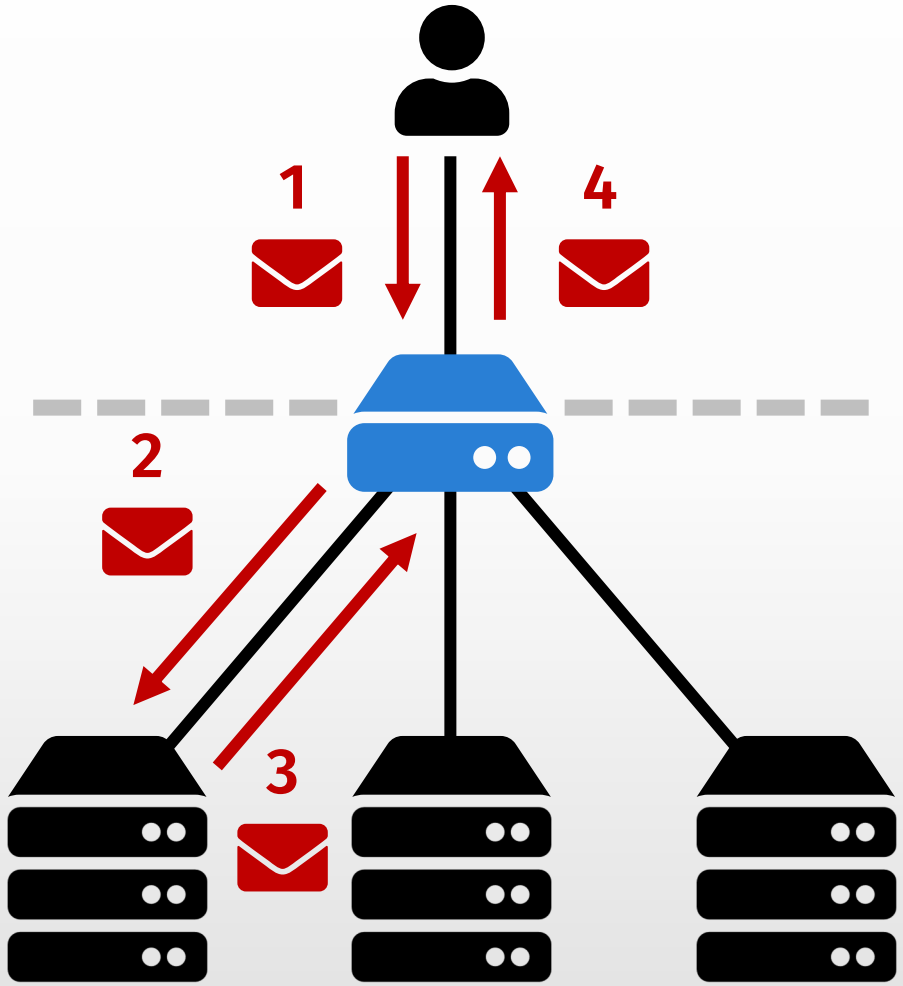
Wenn LB Request erhält, übersetzt er einfach Ziel-IP-Adresse

Vorteile:

- Einfachere Konfiguration
- Möglichkeit für Traffic Inspection

Nachteil:

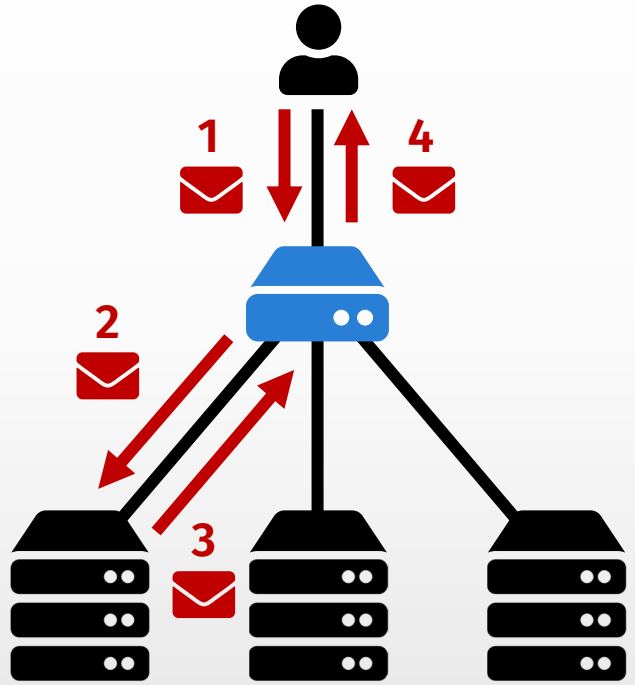
- Server hat andere IP als Client sieht, App muss evtl. angepasst werden



LB-Methode: Reverse Proxy (L7)

Trennung wie bei NAT, aber Requests werden auf Application Layer dekodiert

Beispiel: Client baut HTTP-Verbindung mit LB auf, LB mit Server



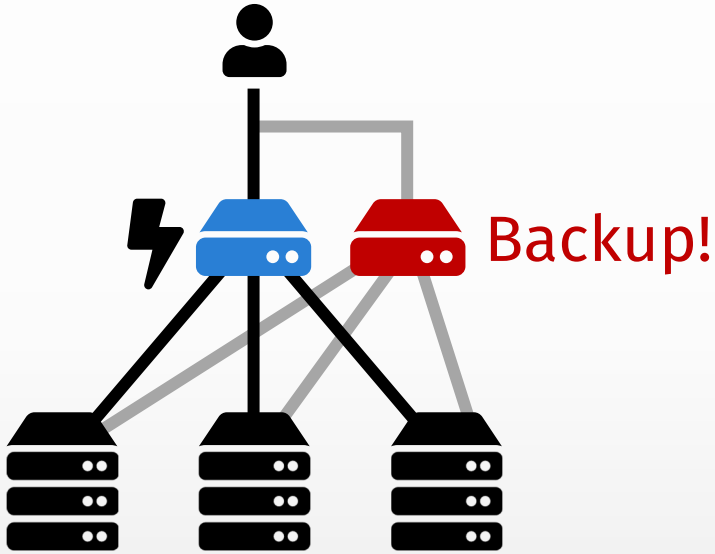
Vorteile:

- Balancing-Entscheidung basierend auf App-Informationen (z.B. Cookies) möglich
- LB kann TLS und Caching übernehmen

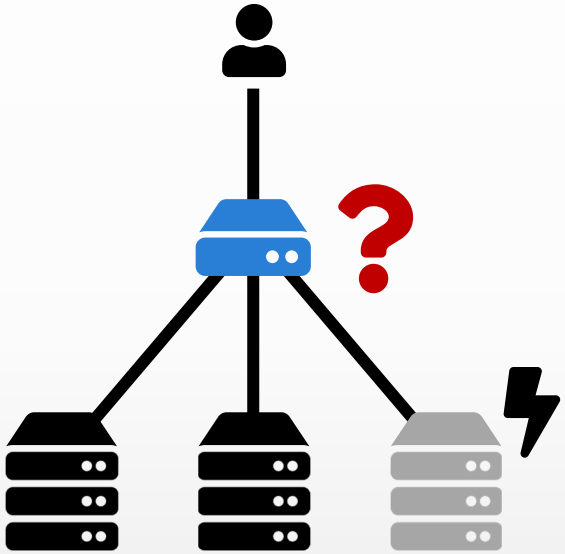
Nachteil:

- Ressourcen-Verbrauch!

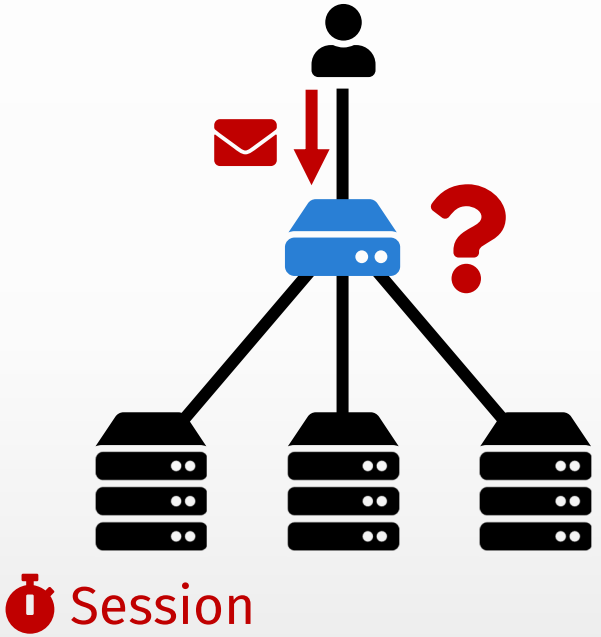
Load Balancing: Herausforderungen



LB-Verfügbarkeit



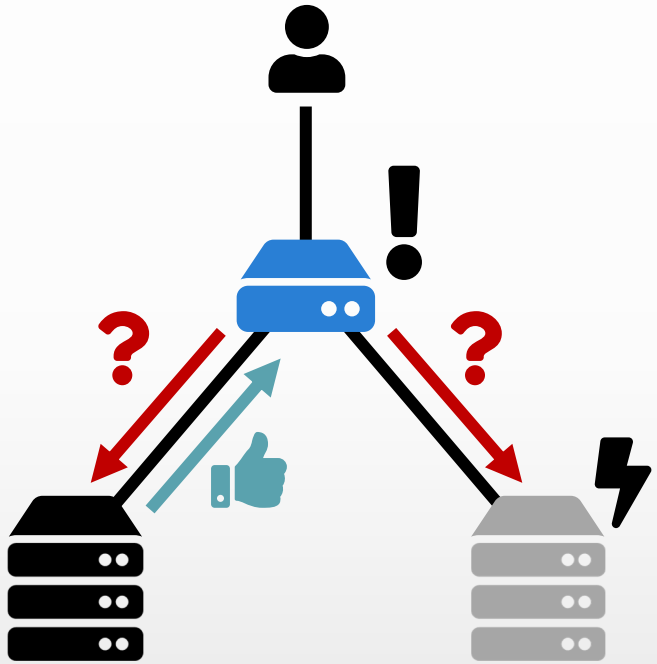
Monitoring
(Health Checks)



Persistenz

Monitoring

Damit LB weiss, welche Server verfügbar sind, führt er regelmässige “Health Checks” durch



Welche Art von Health Checks?

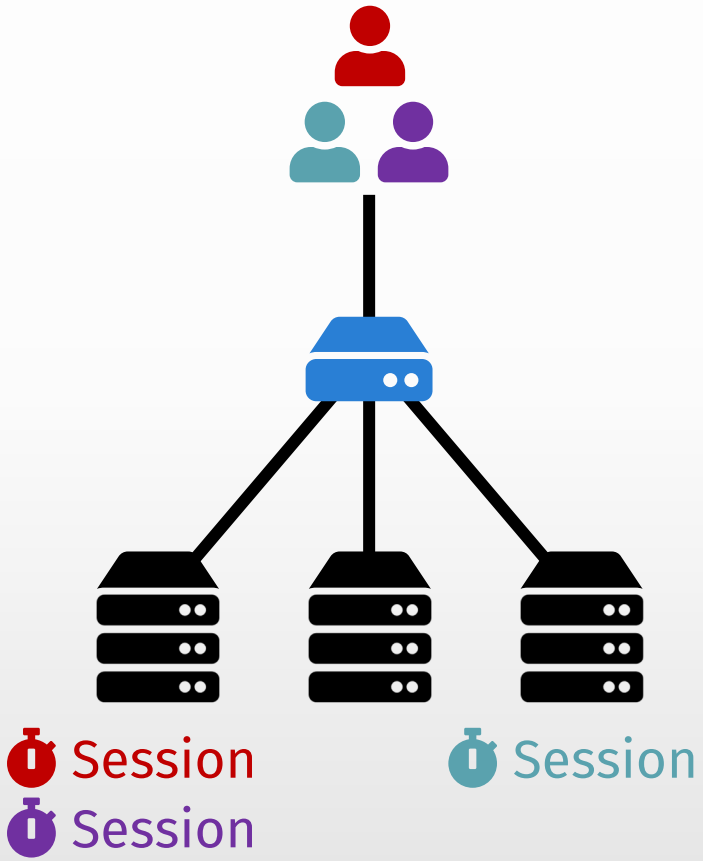
Pings

TCP-Verbindungsaufbau

HTTP-Requests

- Welche?

Persistenz



LB sollte sicher stellen, dass Client immer auf gleichem Server landet

(Zu) Einfach: Zu Beginn HTTP-3xx-Umleitung zu einem Server

- Geht nicht für getrennte Netze

Besser: LB muss Client–Server-Verhältnis lernen

Persistenz: Methoden

IP-Level

- LB erstellt Tabelle mit Zuordnung Client-IP → Server
- Einfach, aber keine Lösung für Clients mit wechselnder IP!

Cookie Learning (HTTP-Level)

- LB inspiziert HTTP-Request (Session-Cookie), erstellt Tabelle mit Zuordnung Session-ID → Server
- Probleme: 1) Endlicher Speicher, 2) Ausfall von Master-LB

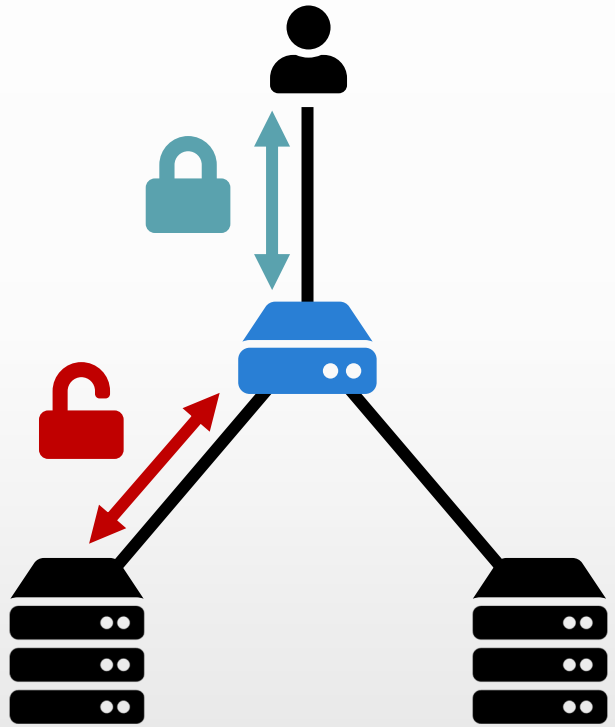
Cookie Insertion (HTTP-Level)

- LB fügt eigenes Cookie in HTTP-Messages ein
- Löst beide Probleme, aber ist rechenintensiver

Persistenz und HTTPS/TLS

HTTP-Level-Persistenz erfordert Einsicht/Änderung von HTTP-Messages.

Was, wenn Traffic verschlüsselt ist?



Mögl. Lösung: TLS kann auf Load Balancer gemacht werden

- LB ist Reverse Proxy, der zwischen HTTPS und HTTP “konvertiert”
- Entlastet Server, aber **be**lastet einzigen LB!

Persistenz und HTTPS/TLS

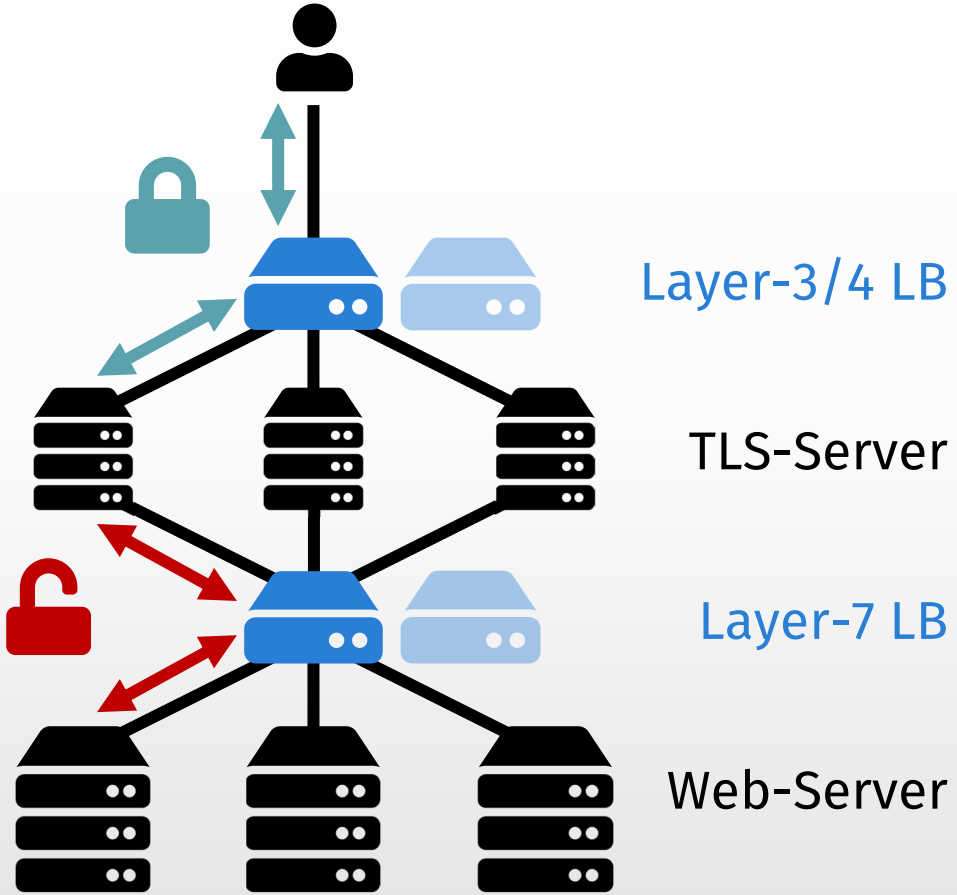
TLS auf Load Balancer: LB kann Bottleneck werden!

Mehrere Load Balancers?

- Nicht empfehlenswert: Farm von LBs verwalten ist schwierig
- Jeder LB muss Health Checks durchführen!

Lösung: TLS-Farm!

- Billig, entkoppelt, flexibel



Übung: Aufbau einer Mikro-“Cloud”

Übungsmaterial

Auf GitHub: https://github.com/apm-fhnw/apm-fs21

APM Woche 01

Übung

Klonen Sie dieses Git-Repository um die Übungsvorlage zu erhalten. Um neues Material zu erhalten, pullen Sie einfach wieder von diesem Repository. Wenn Sie Ihre eigenen Änderungen auf GitHub pushen wollen, forken Sie das Repository stattdessen.

1. App builden und starten

Builden Sie die Web-App im Ordner 'apm-app', indem Sie `mvnw package` darin aufrufen. Dafür muss die `JAVA_HOME` -Umgebungsvariable gesetzt sein. Als Alternative können Sie das 'apm-app'-Projekt als Maven-Projekt in Ihrer IDE importieren und den entsprechenden Maven-Befehl dort ausführen.

Wenn der Build erfolgreich war, führen Sie die Web-App mit folgendem Befehl aus: `mvnw spring-boot:run` . Sie sollten die Web-App nun unter localhost:8080 aufrufen können. Stoppen Sie die App mittels Ctrl+C.

2. App containerisieren

Damit wir die App einfach skalieren können, verpacken wir sie in einen Docker-Container. Dafür müssen Sie zuerst Docker installieren. Erstellen Sie dann im Projeckt-Ordner 'apm-app' eine Datei mit Namen 'Dockerfile' (ohne Dateiendung), welche ein Docker-Image definiert, und kopieren Sie folgenden Inhalt rein:

```
FROM openjdk:11-jre-slim-buster
RUN addgroup --system spring && adduser --ingroup spring --system spring
USER spring:spring
COPY target/*.jar app.jar
ENTRYPOINT ["java","-jar", "app.jar"]
```

Diese Befehle erstellen ein Image, welches die Java-Laufzeitumgebung und die JAR-Datei der Web-App enthält. Die Web-App wird zur Sicherheit mit einem Benutzer 'spring' ohne Root-Rechte ausgeführt. Erstellen Sie das Image, indem Sie folgenden Befehl auf der Kommandozeile aufrufen (dafür muss Docker im Hintergrund laufen):

```
docker build -t fhnw/apm-app .
```

Live-“Coding”

Aufgaben 1 – 3