Moduldokumentation

Modul Application Performance Management (apm)

Simon Wächter

2018

Inhalt

[1 Einleitung 2](#_Toc507246323)

[1.1 Einleitung 2](#_Toc507246324)

[1.2 Lernziele 2](#_Toc507246325)

[1.3 Prüfungen 2](#_Toc507246326)

[2 Woche 1 3](#_Toc507246327)

[2.1 Allgemeines 3](#_Toc507246328)

[2.1.1 Über dem Dozenten 3](#_Toc507246329)

[2.1.2 Der normale Betrieb 3](#_Toc507246330)

[2.1.3 Was dürfen sie erwarten 4](#_Toc507246331)

[2.2 Spass mit Netzwerken 5](#_Toc507246332)

[2.2.1 Ein anderes «Normdokument» 5](#_Toc507246333)

[2.3 Einführung 5](#_Toc507246334)

[2.3.1 Was ist eine Cloud 5](#_Toc507246335)

[2.3.2 Die Servicemodelle einer Cloud 5](#_Toc507246336)

[2.3.3 Der typische Aufbau einer Virtualisierungsinfrastruktur 6](#_Toc507246337)

[2.3.4 Die Skalierungsarten 6](#_Toc507246338)

[2.3.5 Ein typischer Loadbalancer 7](#_Toc507246339)

[2.3.6 Aufgaben und Übungen 8](#_Toc507246340)

[3 Woche 2 9](#_Toc507246341)

# Einleitung

## Einleitung

Dieses Dokument stellt die Moduldokumentation für das Modul apm dar. Allfällige Unterlagen sind im Modulordner zu finden.

## Lernziele

Das Modul beinhaltet folgende Lernziele:

* Studierende dieses Moduls können für eine Applikation Performanceziele aus nicht funktionalen Anforderungen ableiten. Sie sind in der Lage die nötigen Performancemessungen zu planen und durchzuführen.
* Studierende dieses Moduls können Messresultate richtig interpretieren und die daraus notwendigen Massnahmen am Deployment, an der Konfiguration sowie an der Implementation der Applikation vorschlagen bzw. durchführen.
* Studierende dieses Moduls kennen verschiedene Massnahmen zur Performanceoptimierung und können diese auch effektiv einsetzen.
* Studierende dieses Moduls verstehen die Anforderungen von Betreibern von Data Centers. Sie sind in der Lage sinnvolle Quality of Service-Parameter auszuhandeln.

## Prüfungen

Die Modulnote setzt sich zu 100% aus zwei Semesterprüfungen zu je 50% zusammen.

# Woche 1

## Allgemeines

### Über dem Dozenten



### Der normale Betrieb



Jeder Wochenblock richtet sich nach folgendem Muster:

* Unterrichtsblock (2 Lektionen Theorie): Dieser Unterricht wird in Referatsform gehalten. Sie können ihn attraktiver gestalten wenn
  + Sie den zu behandelnden Stoff bereits aufgearbeitet haben.
  + Fragen zum Stoff stellen.
* Übungsblockblock (1 Lektion Einzelarbeit): In diesem Block erhalten Sie jeweils eine Aufgabe, die sie bearbeiten werden. Sie benötigen hierfür einen Computer mit Internetanbindung und eine virtuelle Maschine mit einem Linux.
* Die Hausaufgaben: Die Hausaufgaben bauen üblicherweise auf den Übungsaufgaben auf. Sie sind eine Vertiefung und Vorbereitung auf die Prüfung. Wenn sie die Aufgaben gut lösen, sollten sie wesentlich bessere Noten im Assessment haben.

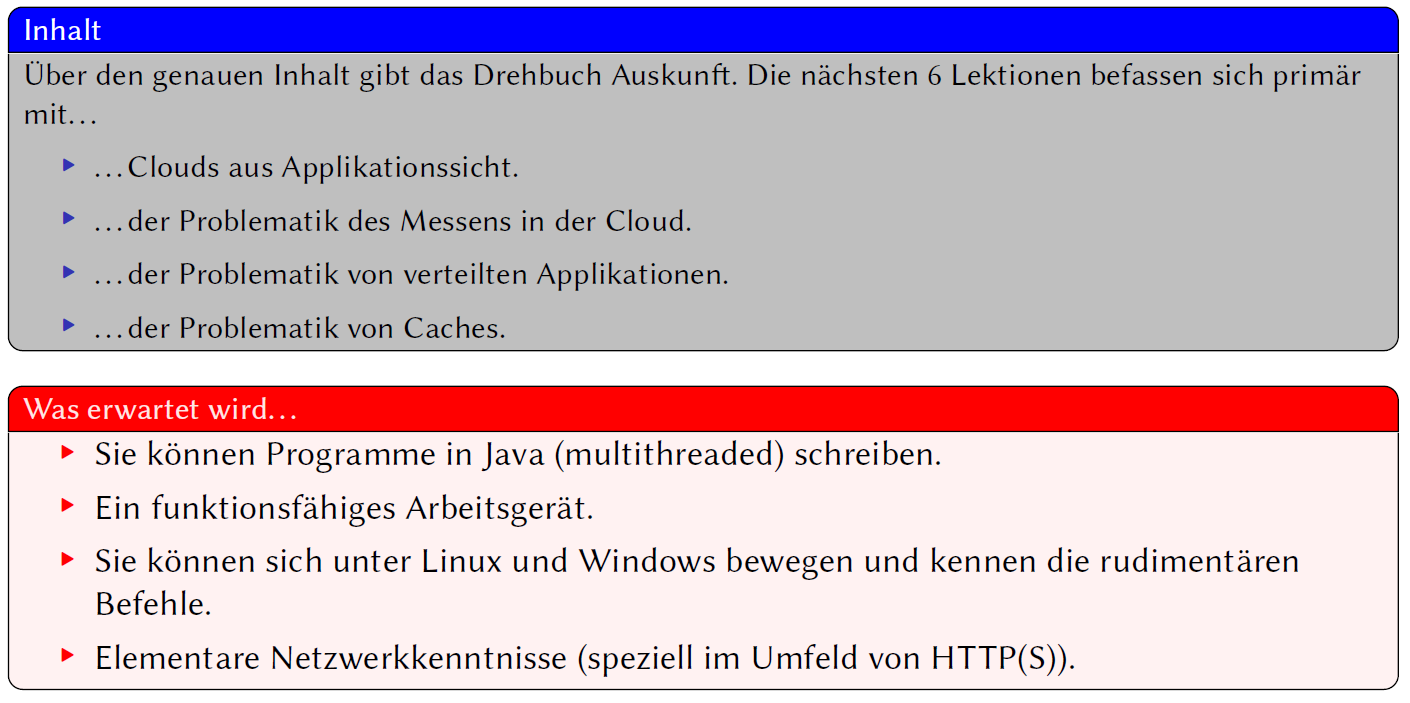
Sie finden an folgenden Orten die Informationen über dieses Modul:

* Im AD
* Drehbuch
* Wenn Sie weitere Quellen benötigen, sind sie in den Abgegebenen Unterlagen verlinkt oder aber sie sind direkt in das PDF-Dokument eingebettet.

Die Slides befinden sich jeweils mindestens 3 Arbeitstage vor der Lektion in der Ablage (im “AD”). Die Aufgaben gliedern sich immer gleich:

* Sie arbeiten den Stoff jeweils vor.
* Sie lösen die Übung (Wenn sie in der Lektion nicht bis zum Ende kommen).
* Sie lösen die Aufgaben.
* Das erste Assessment ist ohne Hilfsmittel.

### Was dürfen sie erwarten

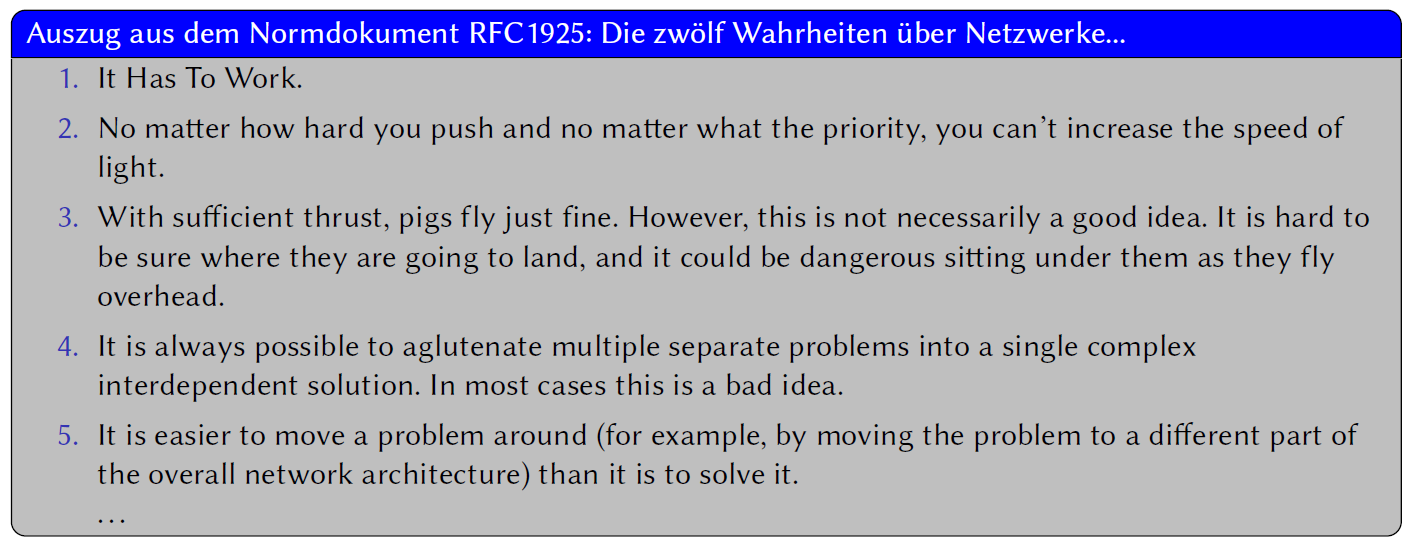


Gewisse Dinge sind für die folgenden Lektionen vorausgesetzt:

* Sie können Programme in Java (Multithreaded) schreiben. Dies bedeutet, Sie kennen die Grundlagen von Threads, Semaphoren, Monitoren, Synchronisation und Interprozess-Kommunikation.
* Ein funktionsfähiges Arbeitsgerät. Es funktionieren alle gängigen Betriebssysteme wie Mac, Windows oder Linux. Es ist gut 8GB RAM zu haben (die Übungen laufen auch mit 4 aber holpriger)
* Sie können sich unter Linux und Windows bewegen und kennen die rudimentären Befehle.
* Elementare Netzwerkkenntnisse (speziell im Umfeld von HTTP(S)).

## Spass mit Netzwerken

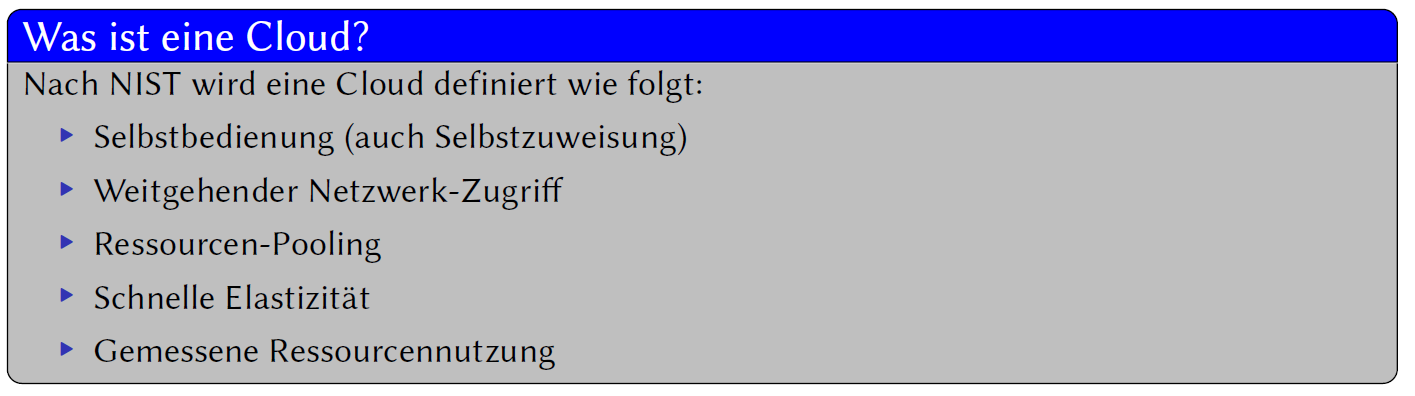
### Ein anderes «Normdokument»



Der vorliegende Auszug stammt aus dem Normen-Dokument RFC1925. Nicht alle Normen sind gleich ernst gemeint und es ist sicher von Bedeutung zu wissen, dass obiges Dokument am 1. April eingereicht worden ist.

## Einführung

### Was ist eine Cloud



Nach NIST wird eine Cloud definiert wie folgt:

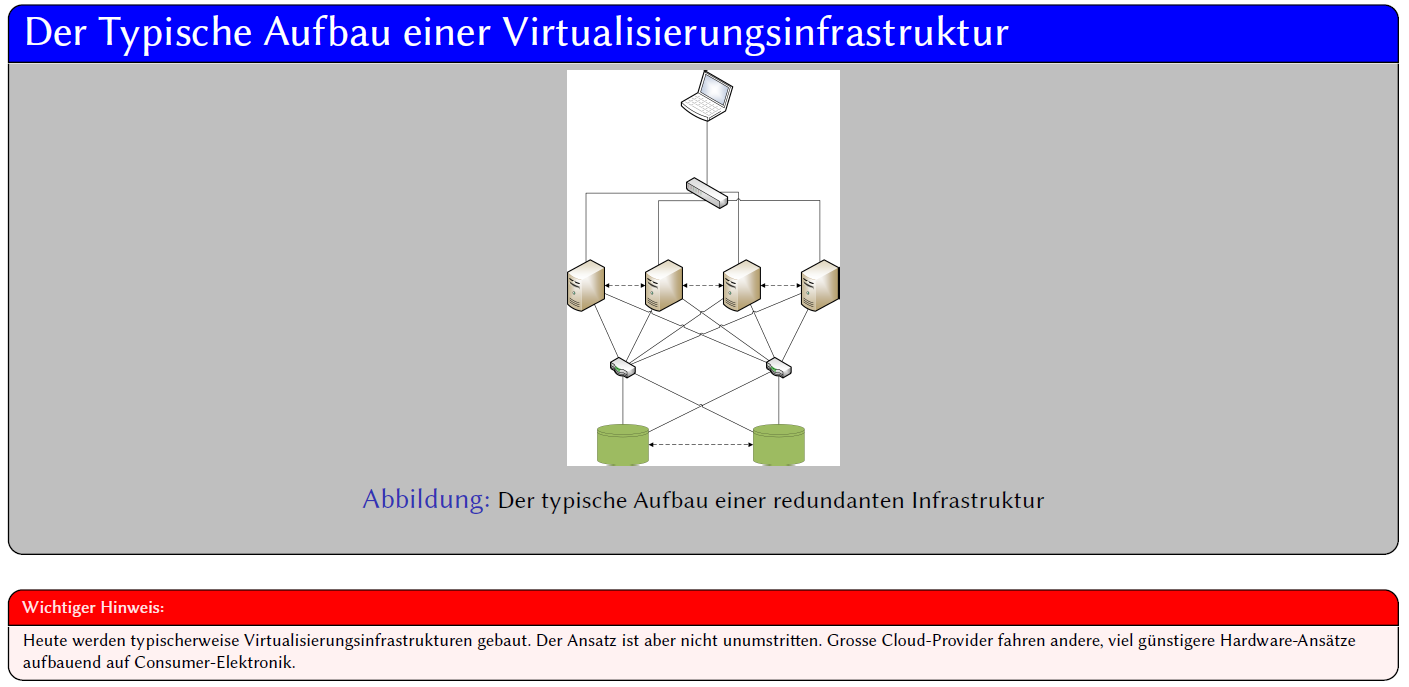
* Selbstbedienung (auch Selbstzuweisung): Das Bereitstellen Ressourcen (z.B. CPUs, Disk-Kapazität, Arbeitsspeicher) läuft automatisch ohne Interaktion mit dem Service Provider ab.
* Weitgehender Netzwerk-Zugriff: Die Services sind mit Standard-Mechanismen über das Netz verfügbar
* Ressourcen-Pooling: Die Ressourcen werden vom Anbieter gebündelt in Pools. Die Benützer können Ressourcen aus diesen Pools beziehen ohne dass sie wissen, wo genau sich die Ressourcen Befinden.
* Schnelle Elastizität: Services können schnell bezogen und auch wieder zurückgegeben werden.
* Gemessene Ressourcennutzung: Die Ressourcennutzung wird gemessen und überwacht.

### Die Servicemodelle einer Cloud



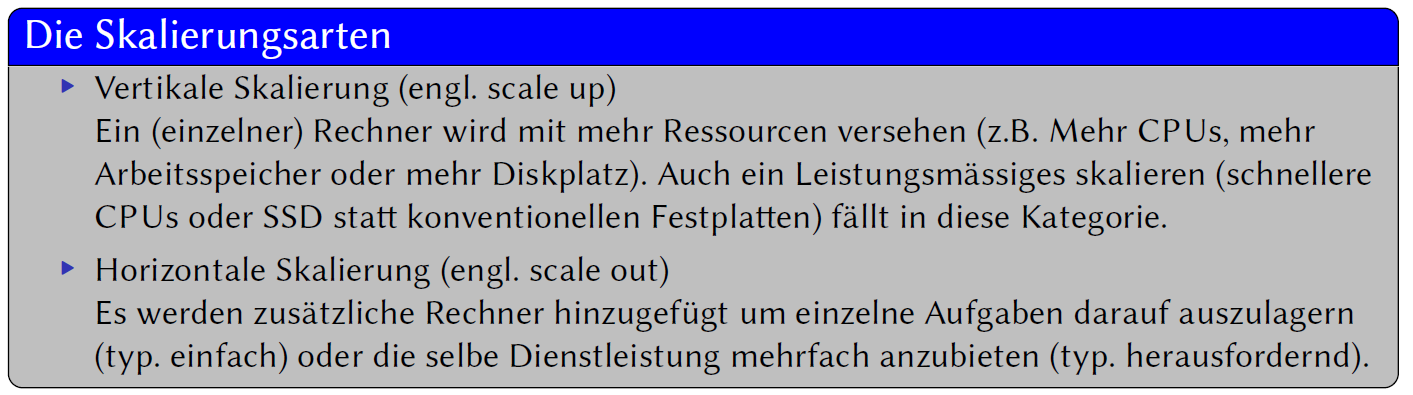
* Infrastructure as a Service (IaaS): Es werden nur die Hardware-Ressourcen zur Verfügung gestellt (CPU, RAM, Disk). Für Betriebssysteme oder Software ist der Kunde zuständig. Manchmal wird auch das Bereitstellen eines Betriebssystems hinzugerechnet.
* Platform as a Service (PaaS): Es wird eine Infrastruktur (siehe IaaS) und mindestens das Betriebssystem bereitgestellt. Typischerweise können auch generische Dienste wie ein SQL-Server oder ein Webserver vorinstalliert sein. Umstritten ist, ob auch die Lizenzen dazu gehören (NIST schweigt sich hierüber aus)
* Software as a Service (SaaS): Weitergehendes bereitstellen einer fertig vorkonfigurierten Software.

### Der typische Aufbau einer Virtualisierungsinfrastruktur



Die typische Virtualisierungsinfrastruktur wird (nach Lehrbuch) aufgebaut wie auf der Grafik gezeigt. Das ist aber aus heutiger Sicht nicht mehr zeitgemäss. Heute werden möglichst günstige Server/Computer verwendet. Der Interne Datenspeicher wird für die virtuellen Maschinen (VMs) verwendet. Eine komplizierte Replikation sorgt dafür, dass Daten zeitnah auch auf anderen Hosts zur Verfügung stehen.

### Die Skalierungsarten



Die typischen Skalierungsarten aus Softwaresicht sind:

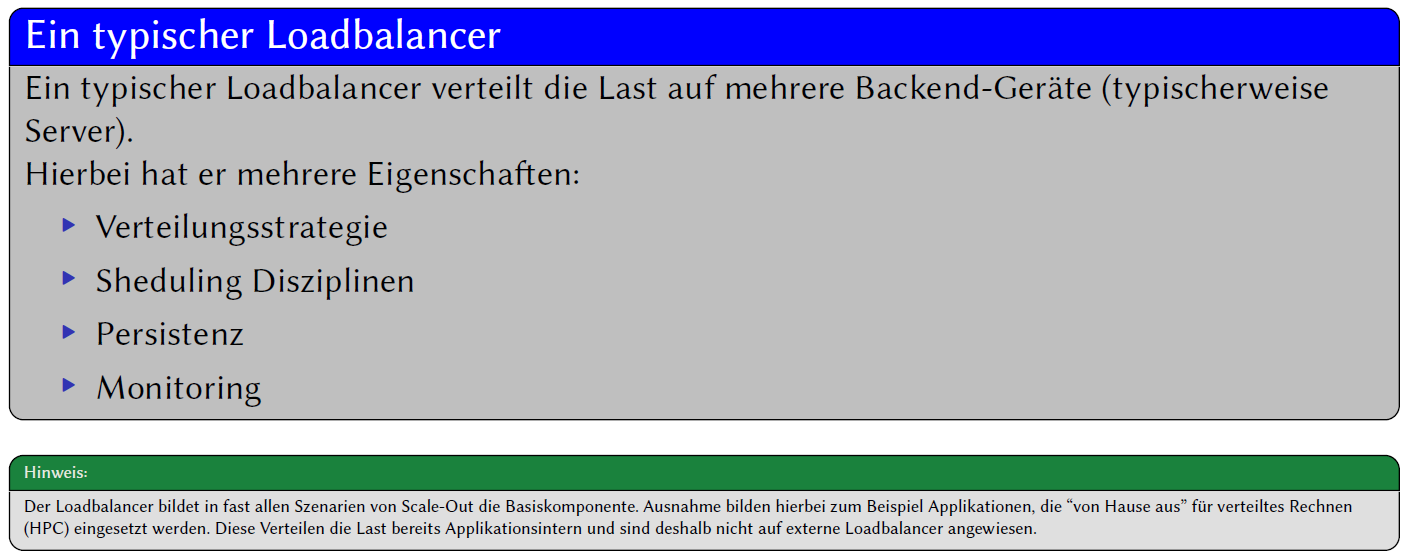
* Vertikale Skalierung (engl scale up): Ein (einzelner) Rechner wird mit mehr Ressourcen versehen (z.B. Mehr CPUs, mehr Arbeitsspeicher oder mehr Diskplatz). Auch ein Leistungsmässiges skalieren (schnellere CPUs oder SSD statt konventionellen Festplatten) fällt in diese Kategorie.
* Horizontale Skalierung (engl. scale out): Es werden zusätzliche Rechner hinzugefügt um einzelne Aufgaben darauf auszulagern (typ. einfach) oder die selbe Dienstleistung mehrfach anzubieten (typ. herausfordernd).

Eine vertikale Skalierung gerät normalerweise schnell an seine Grenzen. Server können nur mit einer begrenzten Anzahl CPUs (wegen der Steckplätze), RAM (auch wegen der Steckplätze) und Festplatten ausgerüstet werden (wobei es am einfachsten ist grosse Festplatten anzuschliessen).

Eine grosse Disk bietet grosse Probleme, die gemeinhin nicht erwartet werden. Eine Festplatte von 1TB Grösse dauert bei einer vollständigen Übertragung über ein 1Gb Netzwerk (z.B. bei einem vollständigen Backup) rund 2.5h (Idealannahme). Stellen sie sich die Datenmengen vor, die während eines Wochenendes (dann werden typischerweise die Full-Backups gemacht) gesichert werden können.

Virtuelle Maschinen sind typischerweise klein (aus Effizienzgründen).

### Ein typischer Loadbalancer



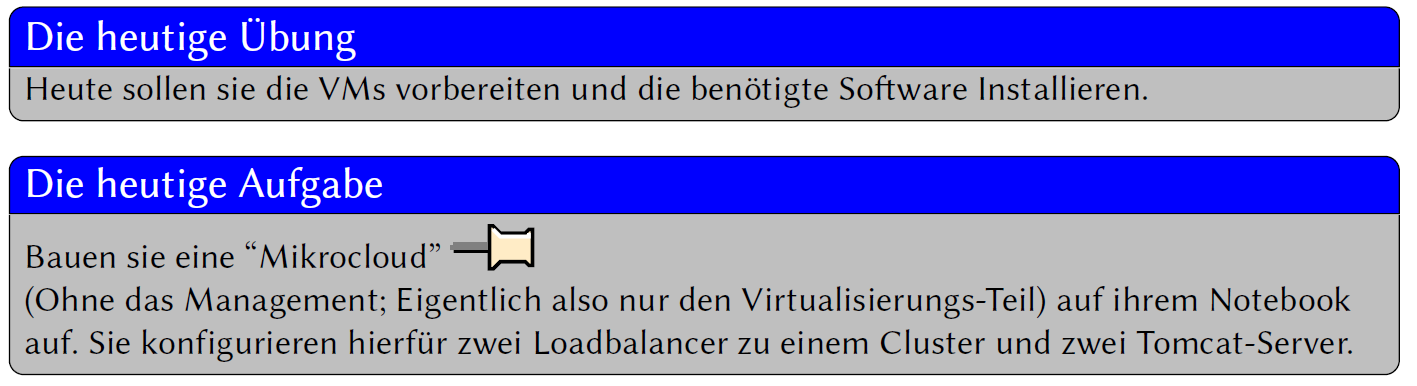
Ein typischer Loadbalancer verteilt die Last auf mehrere Backend-Geräte (typischerweise Server). Hierbei hat er mehrere Eigenschaften:

* Verteilungsstrategie: Nach welcher Strategie werden neu ankommende Verbindungen verteilt. Typische Strategien sind “Round robin” (via DNS oder direkt) oder “load distribution”
* Sheduling Disziplinen: Nach welchem Grundsatz werden Backends zugeteilt. Typische Strategien sind hier beispielsweise FIFO (First In First Out) oder Earliest Deadline First.
* Persistenz: Wie werden eingehende Verbindungen von bekannten Clients identifiziert, damit sie immer an das Selbe Backend geleitet werden. Das kann nach Protokollkriterien erfolgen (z.B. Senderadresse) oder nach Inhalt (z.B. ein spezielles Cookie ist gesetzt).
* Monitoring: Wie werden die Backends überwacht. Über einen simplen Ping, nimmt er TCP-Verbindungen auf einem bestimmten Port an oder über eine vollständige Prüfung respektive Statusabfrage.

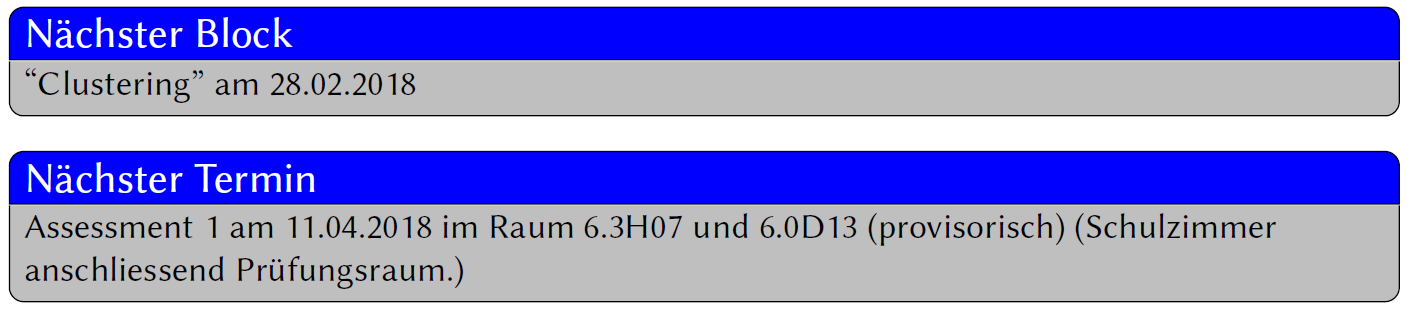
Das sind aber nur die Grundlegenden Eigenschaften. Fortschrittlichere Geräte können auch arbeiten wie intelligentes Routing oder Authentisierung übernehmen. Das sind aber beides streng genommen Eigenschaften eines sogenannten Reverse-Proxies.

Wir legen den Fokus auf sogenannte Serverside-Loadbalancer. Die Clientside-Lodabalancer spielen nur eine extrem untergeordnete Rolle in der Praxis.

### Aufgaben und Übungen



Sorgen Sie dafür, dass Ihre Arbeitsmittel nächstes Mal einsatzbereit sind.



# Woche 2