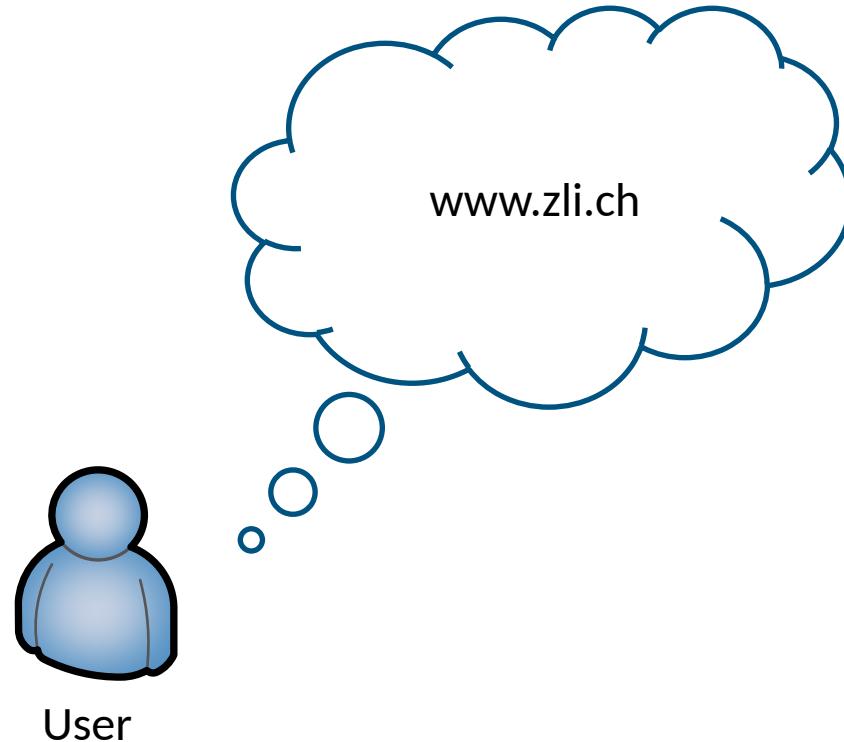
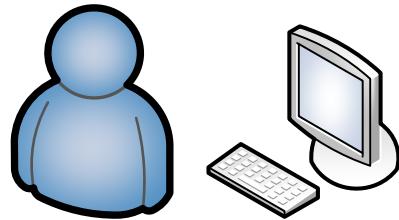


Block 1

Bereitstellung von Daten und Applikationen im Internet

Block 1



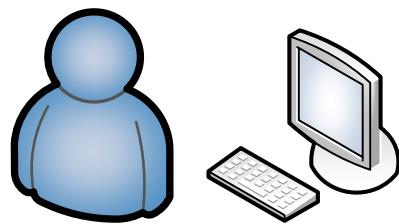


User

Client



Browser



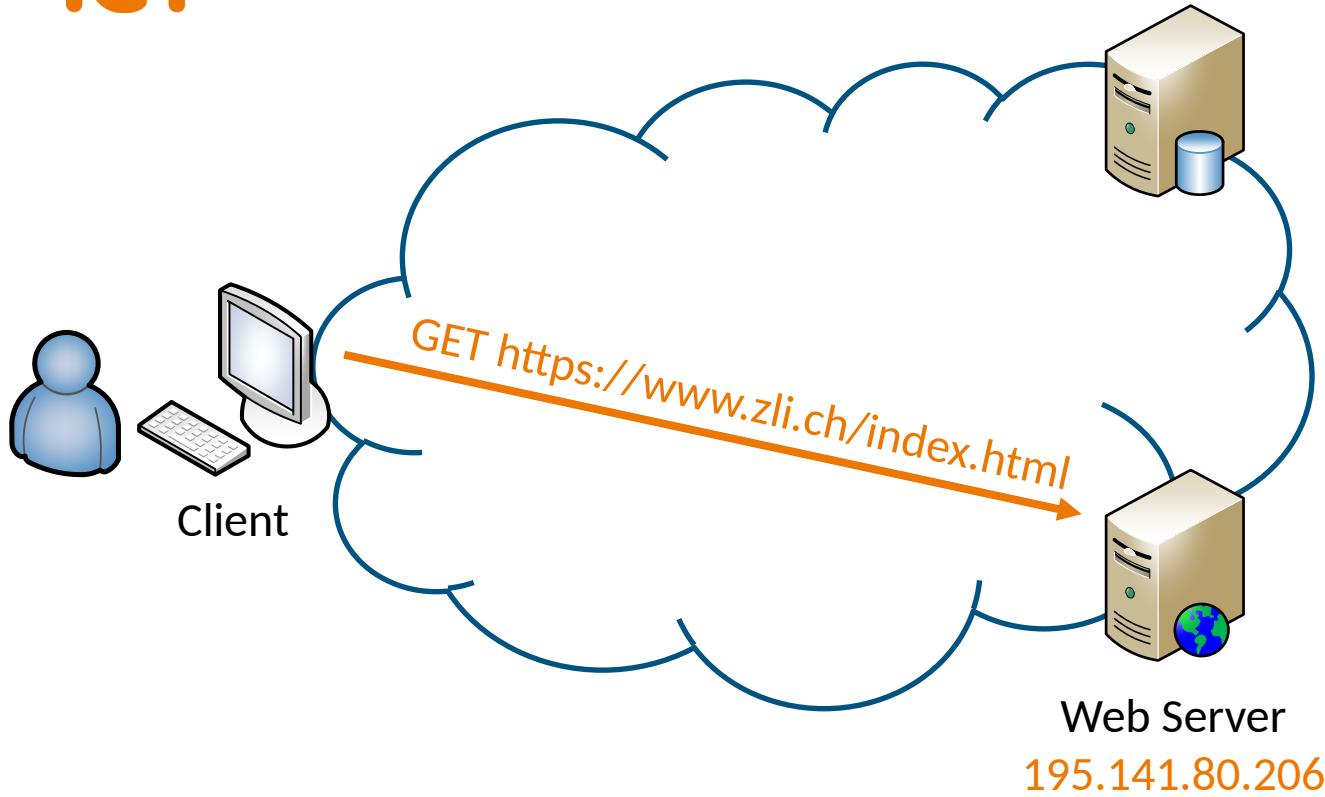
User

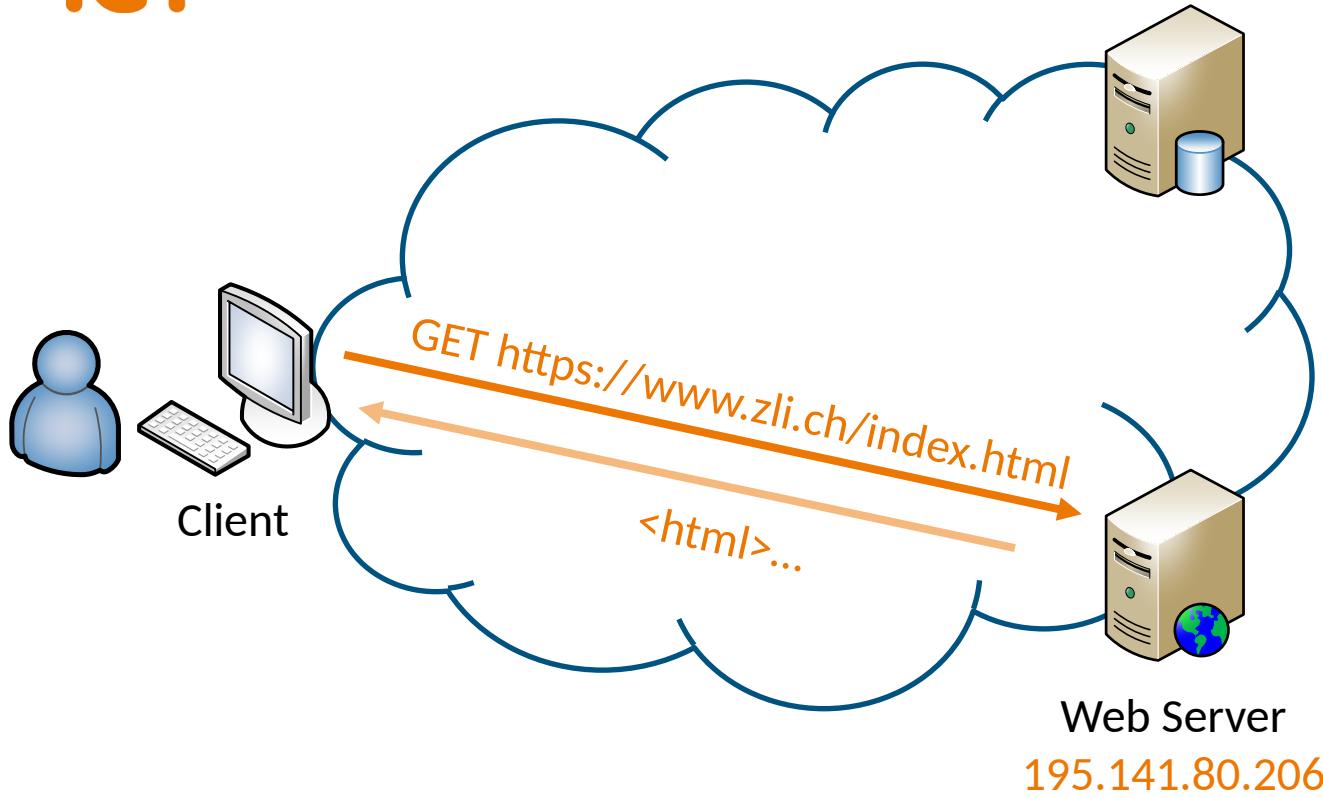
Client

HTTP Anfragen «Klassisch»

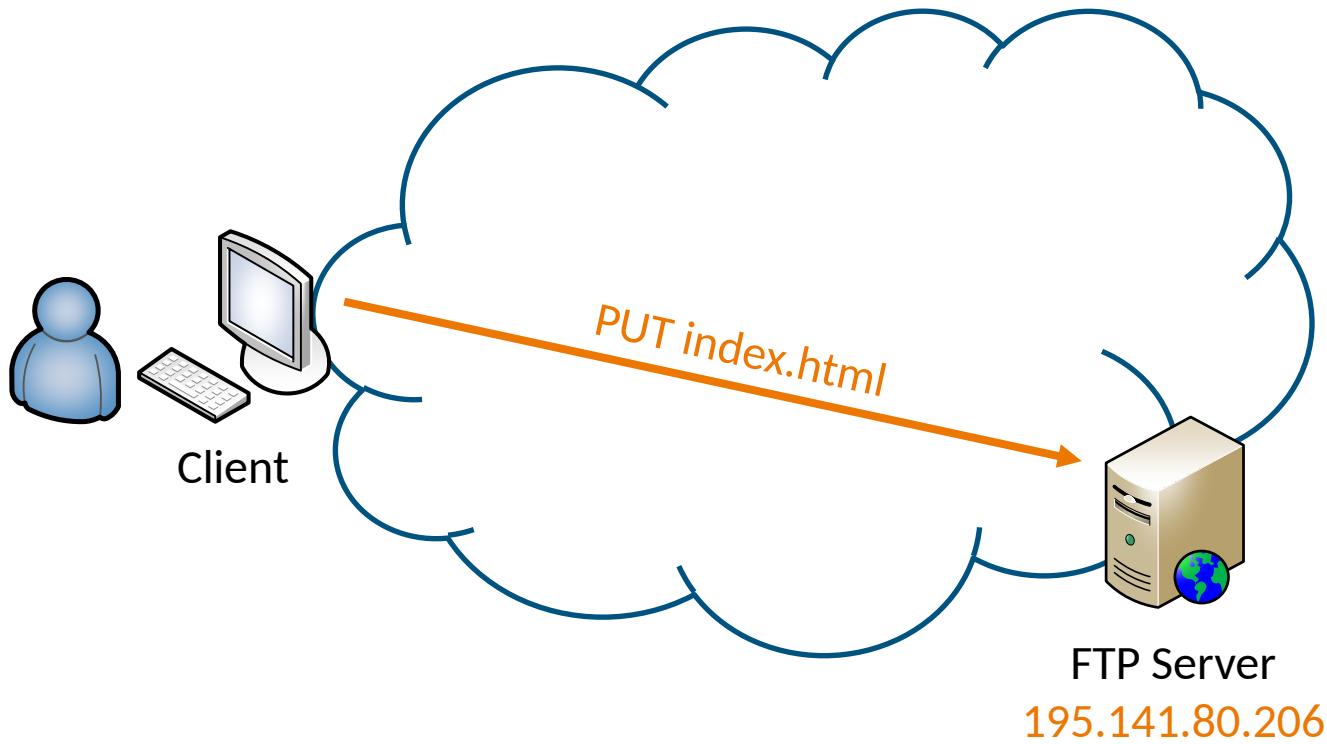








Deployment «Klassisch»

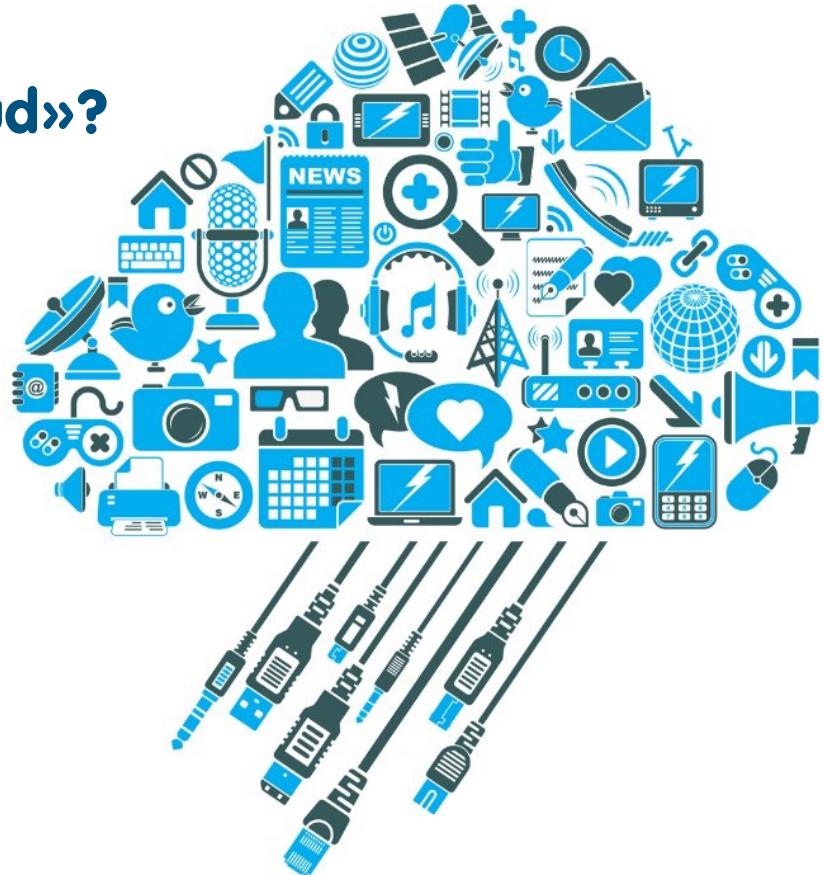


Begriffe und Definitionen

Block 1

Was versteht man unter «Cloud»?

Was versteht man unter «Cloud»?



Wie ist der Begriff entstanden? Wieso heisst es «Cloud»?

Wie ist der Begriff entstanden? Wieso heisst es «Cloud»?

- › Historischer wurde das Internet schon immer als «Wolke» dargestellt
- › Der Verbund von Computern besteht aus ganz vielen kleinen Teilen, wie bei einer Wolke. Ein einzelner Wassertropfen verschwindet im grossen Ganzen, aber gemeinsam ist es eine Wolke.



Ist der Begriff «Cloud» ein synonym für Internet?

- › Ja!
- › «It's just someone else's computer»
- › Cloud = Dienst/Service, die im Internet ausgeführt werden

Wie wird der Begriff definiert?

Wie wird der Begriff definiert?

Cloud Computing ist ein Modell zur Ermöglichung eines allgegenwärtigen, komfortablen und bedarfsgerechten Netzzugangs zu einem gemeinsam genutzten Pool konfigurierbarer Rechenressourcen (z. B. Netze, Server, Speicher, Anwendungen und Dienste), die mit minimalem Verwaltungsaufwand oder minimaler Interaktion mit dem Dienstanbieter schnell bereitgestellt und freigegeben werden können. Dieses Cloud-Modell besteht aus **fünf wesentlichen Merkmalen**, **drei Dienstmodellen** und **vier Bereitstellungsmodellen**.

Woran erkennt man eine Cloud?

Die fünf Merkmale nach NIST

Merkmale des Cloud Computing (1/5)

1. On demand self-service

Nutzung auf Abruf - Der Nutzer hat jederzeit Zugriff auf die Cloud-Dienste.

Merkmale des Cloud Computing (2/5)

2. Broad network access

Zugriff mit bekannten Technologien - Der Nutzer kann über Internetverbindung und entsprechende Clients auf die Dienste zugreifen.

Merkmale des Cloud Computing (3/5)

3. Ressource pooling

Zusammenlegung von Ressourcen - Die Serverkapazitäten werden gebündelt in einer einheitlichen Cloud zur Verfügung gestellt.

Merkmale des Cloud Computing (4/5)

4. Rapid elasticity

Ressourcenanpassung - Die zur Verfügung stehenden Ressourcen werden angepasst und es entsteht der Eindruck eines unbegrenzten Speicherplatzes.

Merkmale des Cloud Computing (5/5)

5. Measured service

Überwachung des Dienstes - Die einzelnen Cloud-Server werden immer wieder überwacht und optimiert.

Welche Dienstleistungen gibt es in der Cloud?

Welche Dienstleistungen gibt es in der Cloud?

- › E-Mail
- › Web-Server
- › Dropbox
- › iCloud
- › etc.



N

ne Dienstle



Google

Office



Wo wird die Infrastruktur für die Cloud betrieben? Die vier Deployment-Modelle

Deployment-Modelle: Public

Public
<p>In diesem Cloud-Computing-Modell stellt ein Drittpartei-Anbieter über das Internet der Öffentlichkeit Cloud-Computing-Ressourcen zur Verfügung. Mit der Public Cloud müssen Unternehmen nicht ihre eigenen Cloud-Server inhouse aufbauen oder warten.</p> <p>EIGENSCHAFTEN: Multi-Tenant-Architektur Nutzungsorientiertes Abrechnungsmodell.</p> <p>HAUPTANBIETER: AWS, Microsoft Azure, Google Cloud Platform</p>

- › Bereitstellung im **geteilten** Rechenzentrum
- › Keine Kontrolle über die Infrastruktur

Deployment-Modelle: Private

Private
Ein Cloud-Computing-Modell, mit dem ein Unternehmen proprietäre Architektur nutzt sowie Cloud-Server, die in ihrem eigenen Rechenzentrum laufen.
EIGENSCHAFTEN: Single-Tenant-Architektur On-Premises-Hardware Direkte Kontrolle über die zugrundeliegende Cloud-Infrastruktur
HAUPTANBIETER: HPE, VMware, Dell EMC, IBM, Red Hat, Microsoft, OpenStack

- › Bereitstellung im **eigenen** Rechenzentrum
- › Volle Kontrolle über die Infrastruktur

Deployment-Modelle: Hybrid

Hybrid

Ein Cloud-Computing-Modell, bestehend aus einem Mix an On-Premises, Private Cloud und Drittpartei-Public-Cloud-Services mit Orchestrierung zwischen den beiden Plattformen.

EIGENSCHAFTEN:

Cloud-Bursting-Fähigkeiten
Public- und Private-Cloud-Umgebungen

HAUPTANBIETER:

Mischung aus
Public- und Private-Cloud-Anbietern

- › Mischung aus Public- und Private-Cloud Anbietern

Deployment-Modelle: Community

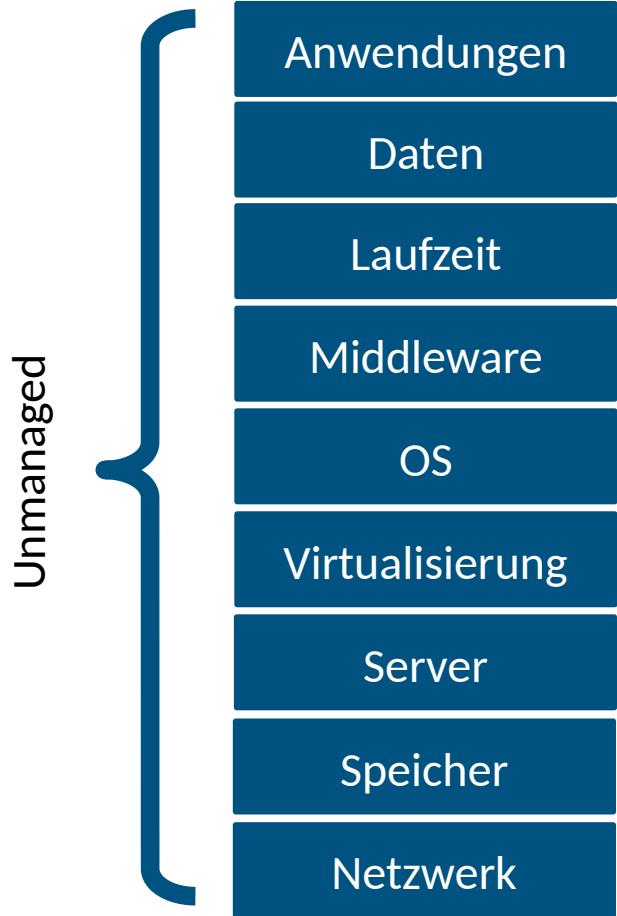
- › Mehrere Firmen schliessen sich zusammen um gemeinsam eine Cloud zu betreiben

Was wird zur Verfügung gestellt?

Die vier Service-Modelle

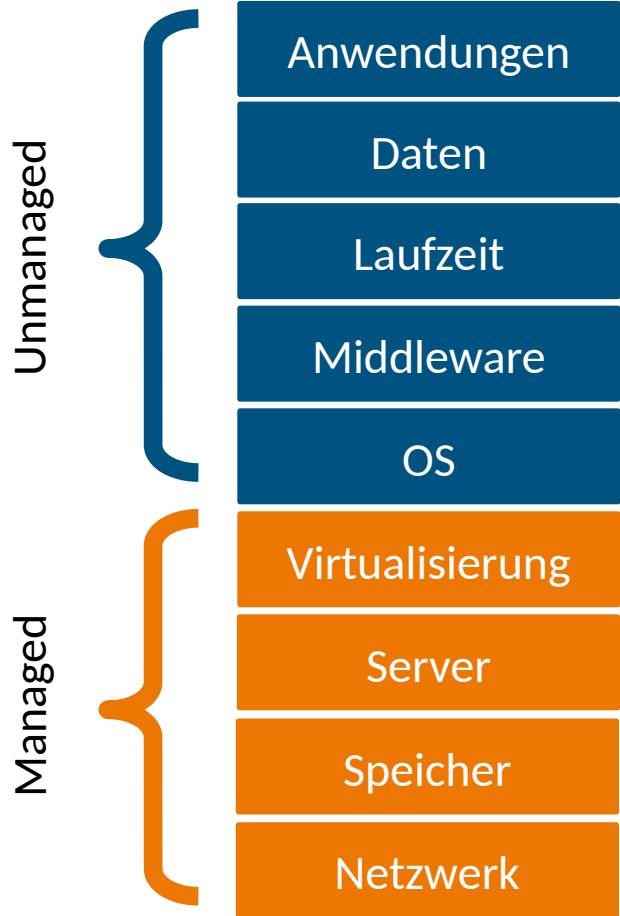
On Premises

- › Alles wird selbst gekauft, installiert, verwaltet und gewartet.
- › z.B.: Applikationen laufen auf den eigenen Servern im Keller.



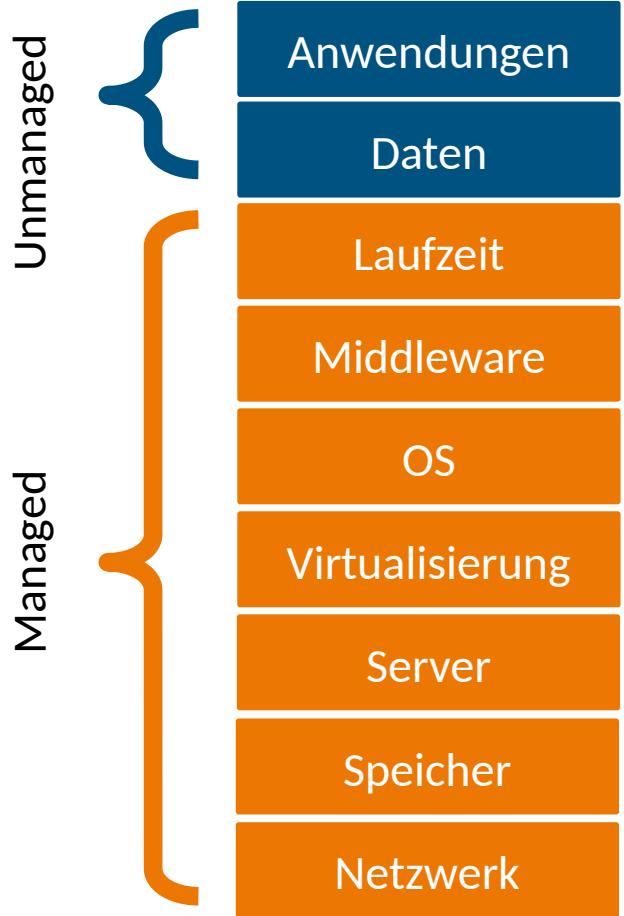
Service-Modelle: IaaS

- › Physische Server werden durch den Anbieter verwaltet und gewartet. Betriebssystem und Software werden selbst verwaltet.
- › z.B.: Applikationen laufen auf Virtual Private Server bei Hetzner, Nodes bei Cloudscale, Amazon E2C



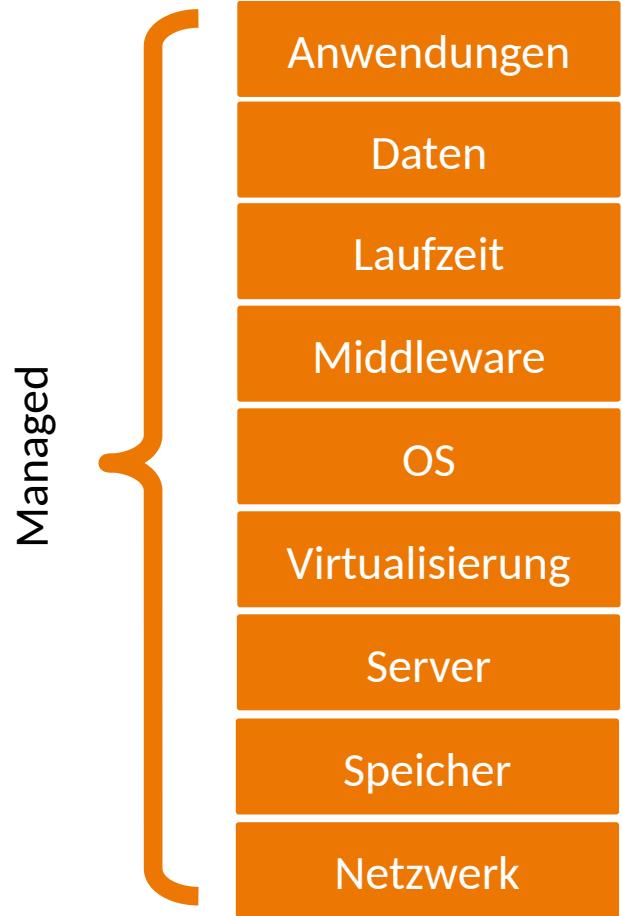
Service-Modelle: PaaS

- › Laufzeitumgebung für Applikationen wird durch Anbieter zur Verfügung gestellt. Applikation wird selbst verwaltet
- › z.B.: Applikationen laufen auf Heroku



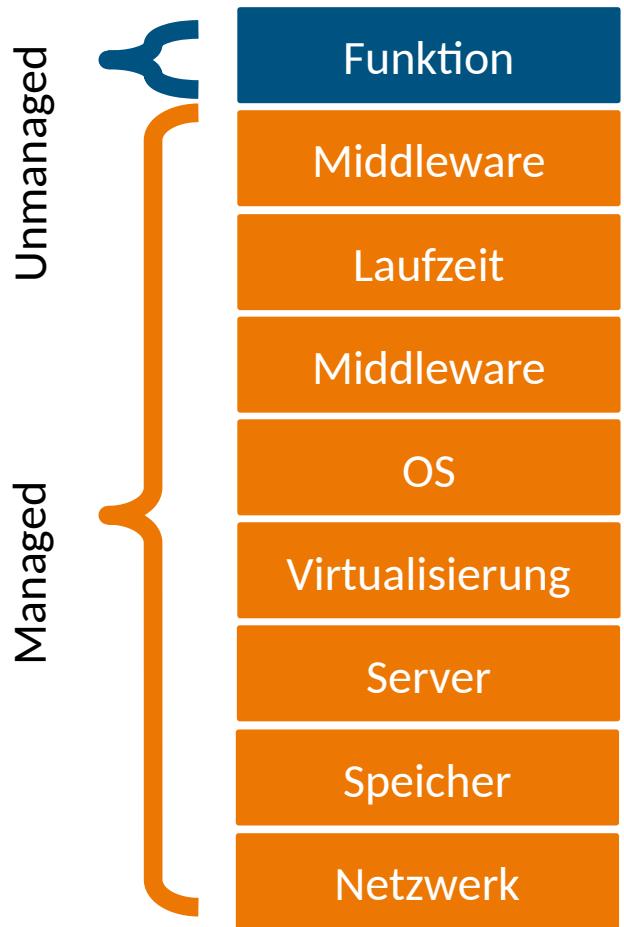
Service-Modelle: SaaS

- › Die gesamte Applikation wird durch den Anbieter zur Verfügung gestellt.
- › z.B.: Google Docs, Dropbox



*Service-Modelle: FaaS

- › Laufzeitumgebung inkl. Middleware (z.B. HTTP Server) wird zur Verfügung gestellt.
Nur die eigentliche Funktion selbst wird selbst verwaltet
- › z.B.: Google Cloud Functions, Firebase



Welche Vorteile haben Cloud-Modelle?

Vorteile von Cloud-Lösungen

- › Bezahlung nach Verbrauch
- › Geringe Gesamtkosten
- › Geringer Wartungsaufwand
- › Weltweiter Zugang



Vorteile von Cloud-Lösungen

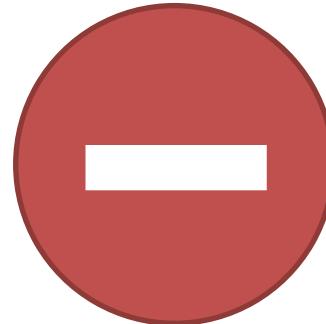
- › Garantierte Betriebskontinuität und Verfügbarkeit
- › Hohe Flexibilität und Elastizität
- › Kurze Markeinführungszeit



Welche Nachteile haben Cloud-Modelle?

Nachteile von Cloud-Lösungen

- › Bezahlung nach Verbrauch
- › Sehr hohe Komplexität
- › Weniger Kontrolle



Block 2

Cloud Anbieter

Block 2

Welche Cloud-Anbieter gibt es?

Welche Cloud-Anbieter gibt es?

- › Amazon AWS
- › Google Cloud Service GCS
- › Microsoft Azure
- › Alibaba Cloud
- › Cloudscale, Flow, ...

Cloud Anbieter: Amazon Web Services AWS



Gründung	2006
Sitz	Seattle USA
Umsatz	45 Mrd. USD
Dienste	> 200
Rechenzentren	77

Cloud Anbieter: Microsoft Azure



Gründung	2008
Sitz	Redmond USA
Umsatz	51 Mrd. USD
Dienste	> 200
Rechenzentren	> 160

Cloud Anbieter: Alibaba Cloud



Gründung	2009
Sitz	Hangzhou CN
Umsatz	9 Mrd. USD
Dienste	
Rechenzentren	24

Cloud Anbieter: Google Cloud Platform



Gründung	2008
Sitz	Mountain View USA
Umsatz	12 Mrd. USD
Dienste	
Rechenzentren	24

Cloud Anbieter: Cloudscale



Gründung	2008
Sitz	Zürich
Umsatz	?
Dienste	
Rechenzentren	2

Welche Cloud-Anbieter gibt es?

- › Demo Amazon, Cloudscale, Flow

Welche Cloud Dienste betreibt Ihre Firma selbst?

Welche Cloud Dienste betreibt Ihre Firma selbst?

- >
- >
- >

Applikationen deployen mit git

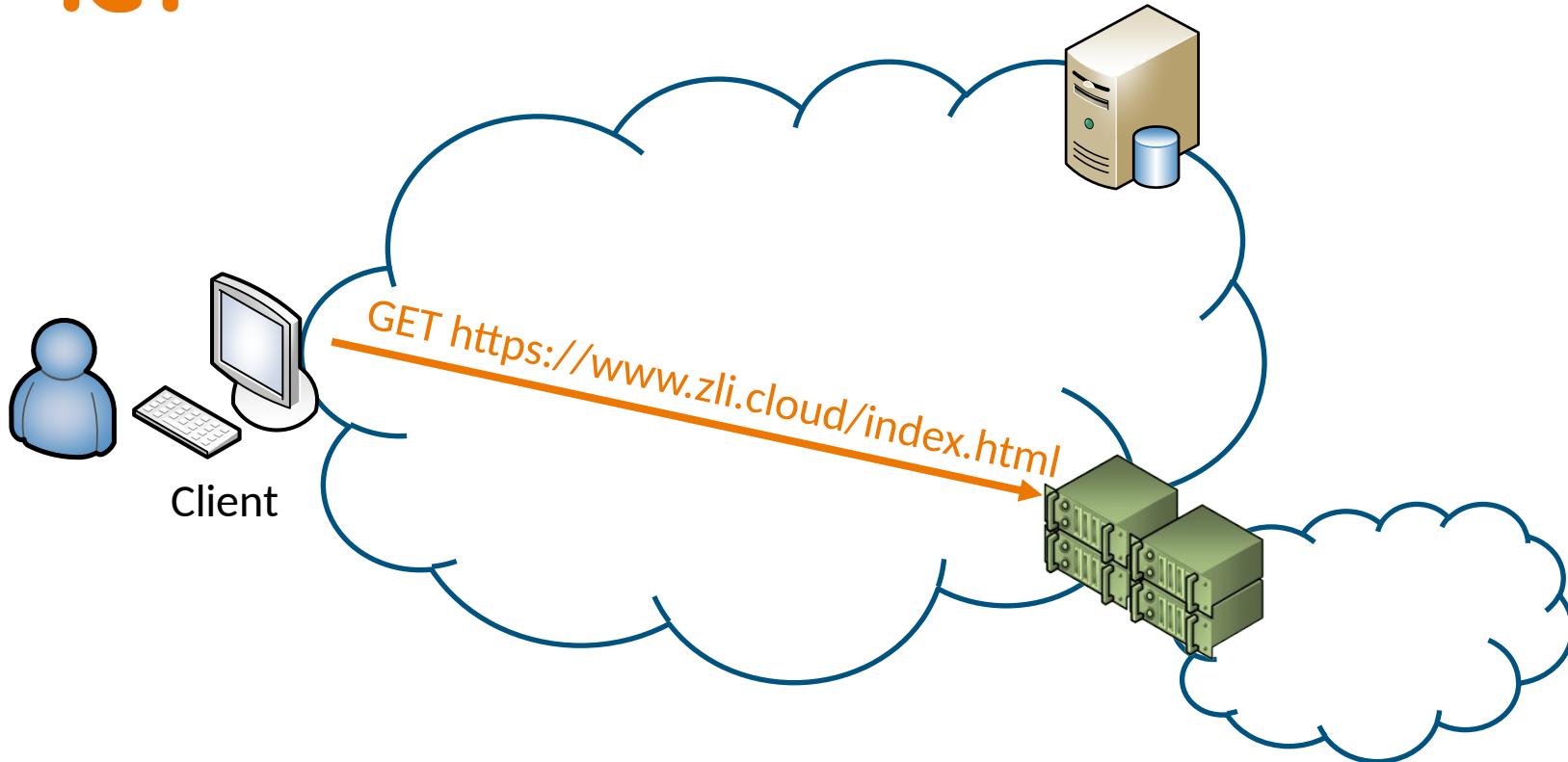
Block 2

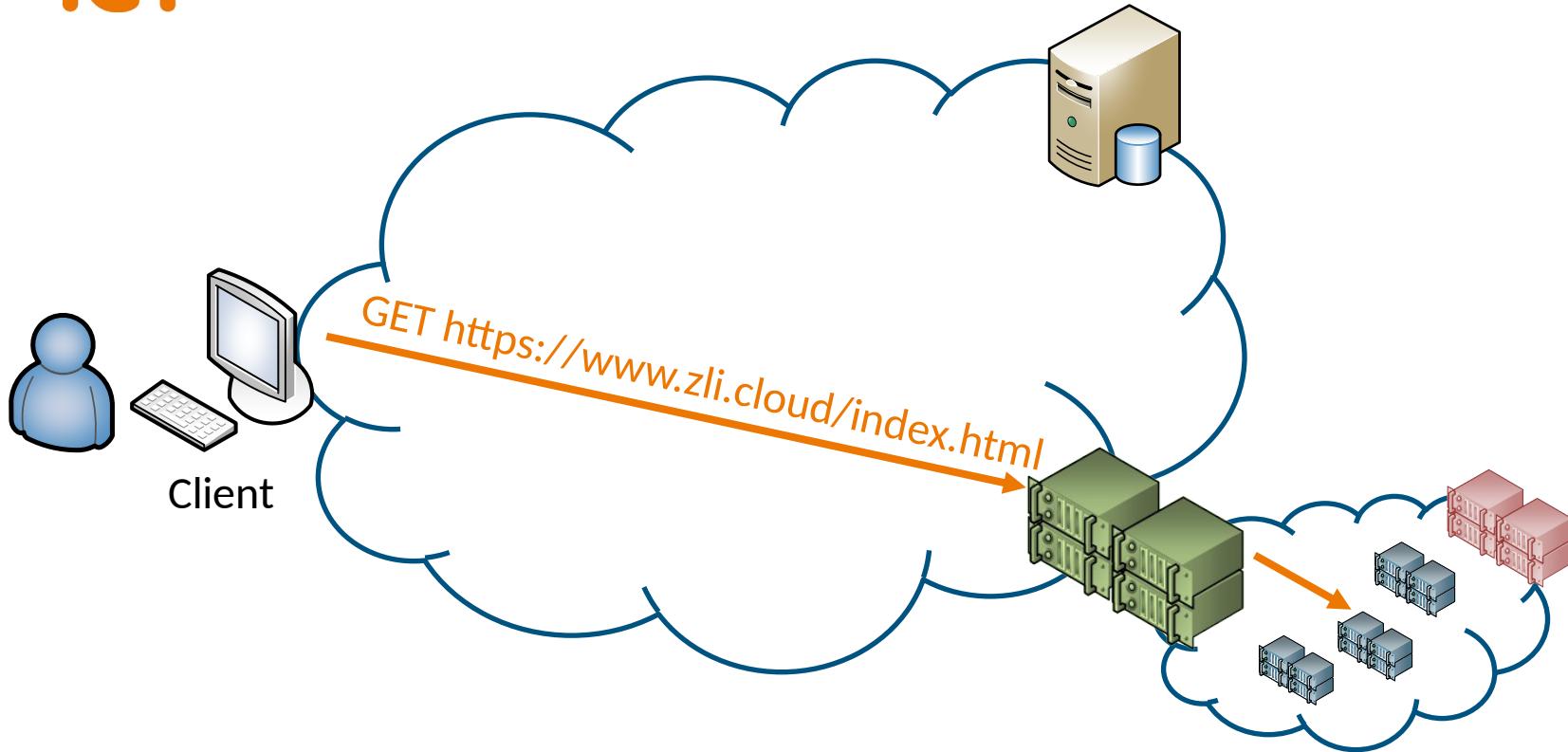
Wie kommt eine Applikation in die Cloud?

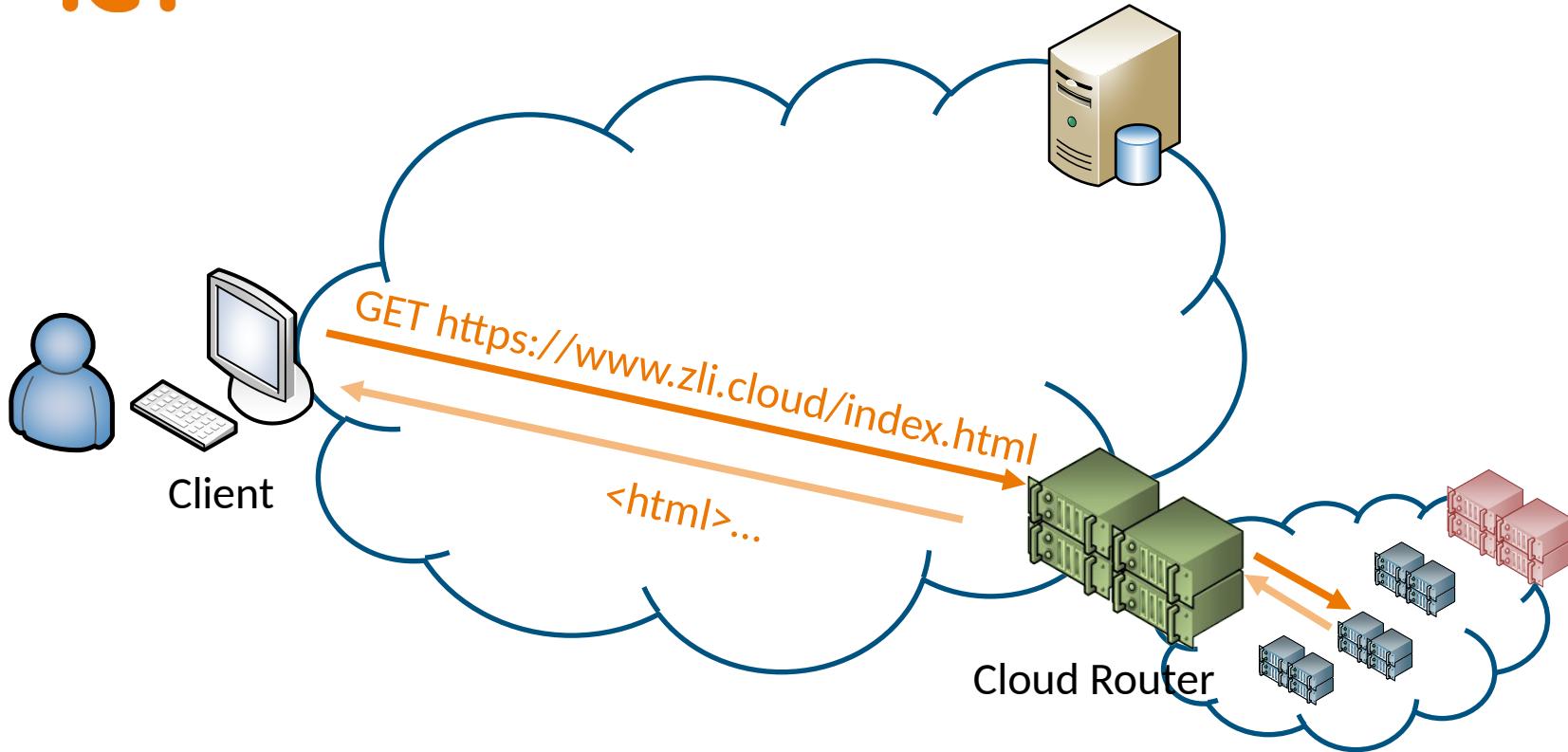
HTTP Anfragen «Cloud»



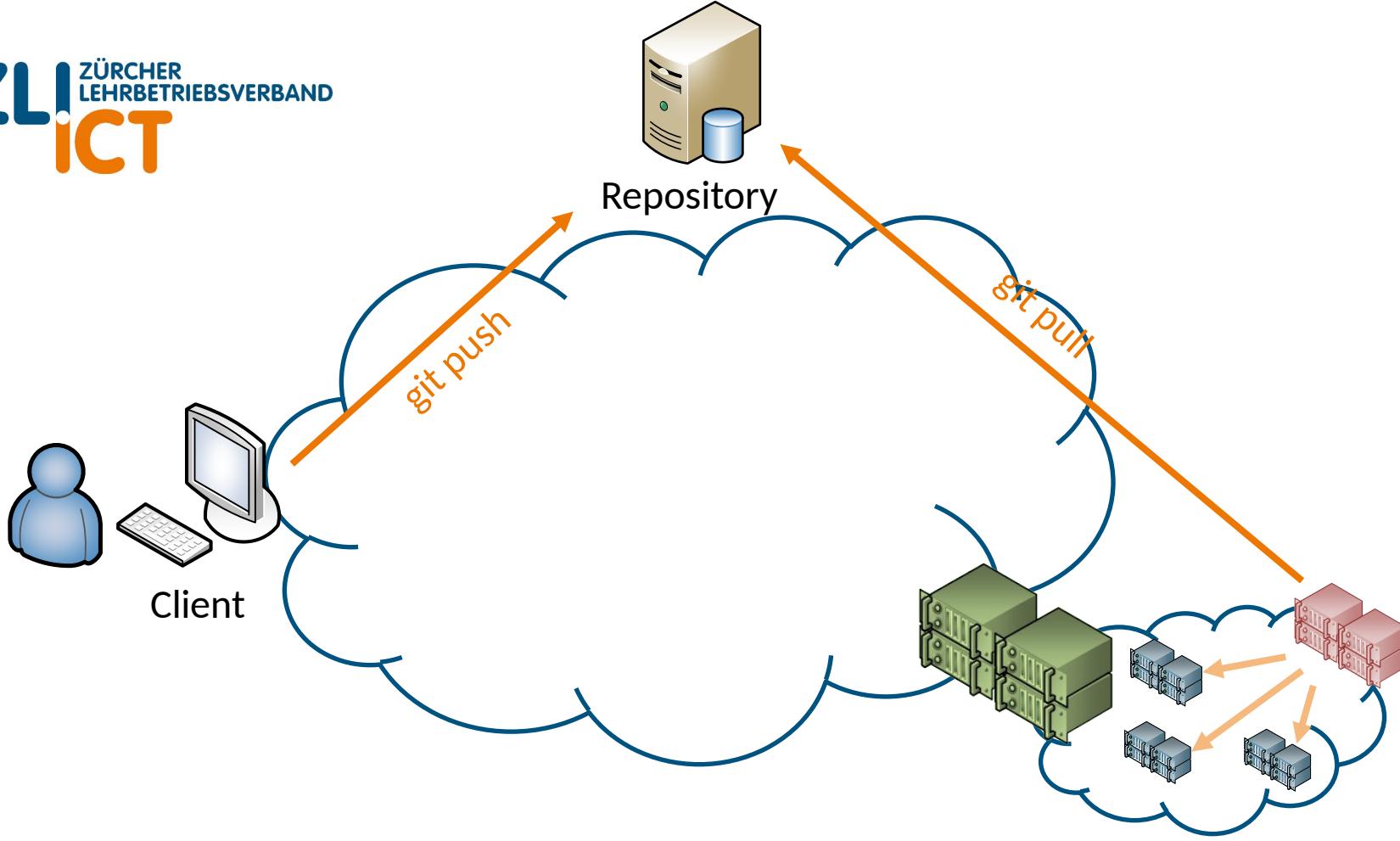








Deployment «Cloud»



Wie werden technologische Beiträge in der Cloud geteilt
bzw. zur Verfügung gestellt? (1/3)

- › Code Repositories
 - z.B. GitHub, GitLab, usw.

GitHub

 Bitbucket



GitLab



Azure DevOps

Wie werden technologische Beiträge in der Cloud geteilt
bzw. zur Verfügung gestellt? (2/3)

- › Binary-/ Container Repositories
z.B. dockerhub



Wie werden technologische Beiträge in der Cloud geteilt
bzw. zur Verfügung gestellt? (3/3)

- › Anbieterspezifische Marketplaces



Block 3

Einführung in die Container Technologie

Block 3

Was ist Container-Technologie oder Container-Virtualisierung?

- › Eine standardisierte Software-Einheit.
- › Die Container-Virtualisierung erlaubt es, eine Anwendung unabhängig von einem Host-Betriebssystem zu betreiben
- › Im Container verpackt ist also eine komplette Anwendung inklusive Code, Abhängigkeiten und Konfigurationen in einem klar festgelegten Format.

Was sind die Vor – und Nachteile der Technologien?

Virtuelle Server	Container
Prozessisolierung auf Hardwareebene	Prozessisolierung auf Betriebssystemebene
VM bietet vollständige Isolierung der Anwendungen vom Hostbetriebssystem	Container können gewisse Ressourcen mit dem Host-Betriebssystem teilen
Jeder virtuelle Server hat ein eigenes Betriebssystem	Jeder Container kann Betriebssystem-Ressourcen gemeinsam nutzen
Hochfahren in Minuten	Hochfahren in Sekunden
Mehr Ressourcenverbrauch	Geringerer Ressourcenverbrauch
Vorkonfigurierte VMs sind schwer zu finden und zu verwalten	Vorgefertigte Container für Anwendungen bereits verfügbar

Was sind die Vor – und Nachteile der Technologien?

Virtuelle Server	Container
Die Anpassung von vorkonfigurierten VMs erfordert Arbeit	Benutzerdefiniertes Setup mit Containern ist einfach zu erstellen
VMs sind in der Regel grösser, da sie ein vollständiges Betriebssystem enthalten.	Container sind klein und haben nur eine Container-Engine auf dem Host-Betriebssystem
VMs können leicht auf einen neuen Host verschoben werden	Container werden zerstört und neu erstellt, anstatt sie zu verschieben
Die Erstellung von VMs nimmt relativ viel Zeit in Anspruch	Container können in Sekundenschnelle erstellt werden
Virtualisierte Apps sind schwieriger zu finden und es dauert länger, sie zu installieren und auszuführen.	Containerisierte Anwendungen können innerhalb weniger Minuten gefunden und installiert werden

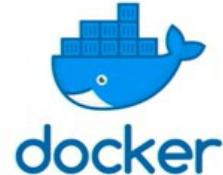
**Welche Produkte kennen Sie im Zusammenhang mit
virtuellen Servern und Container?**

Welche Produkte kennen Sie im Zusammenhang mit virtuellen Servern und Container?

Virtual Machines



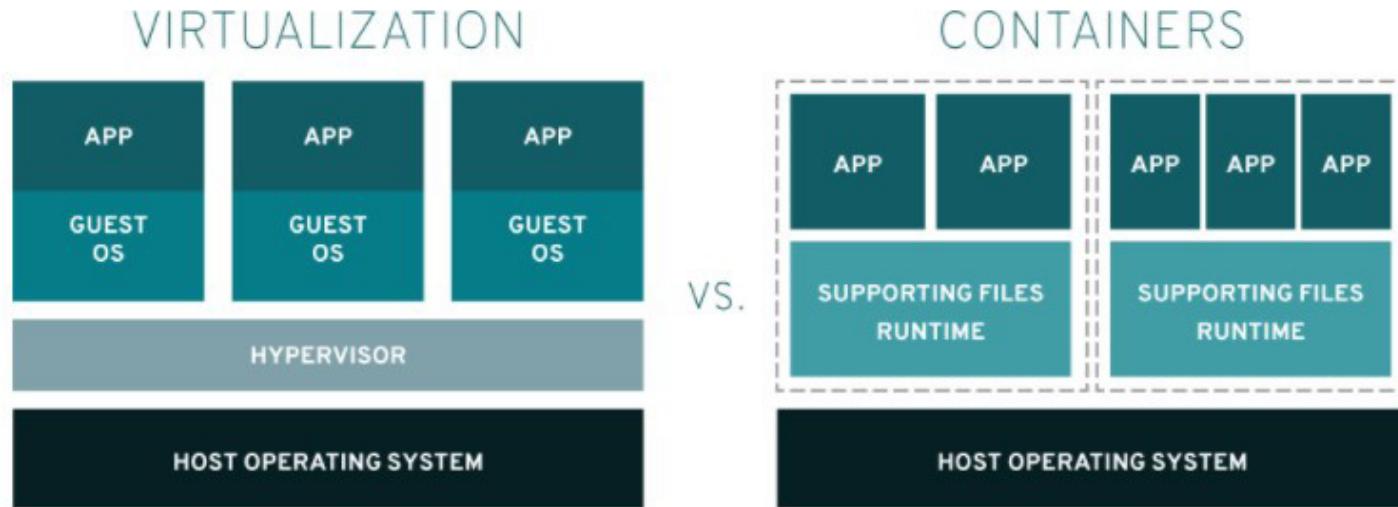
Containers



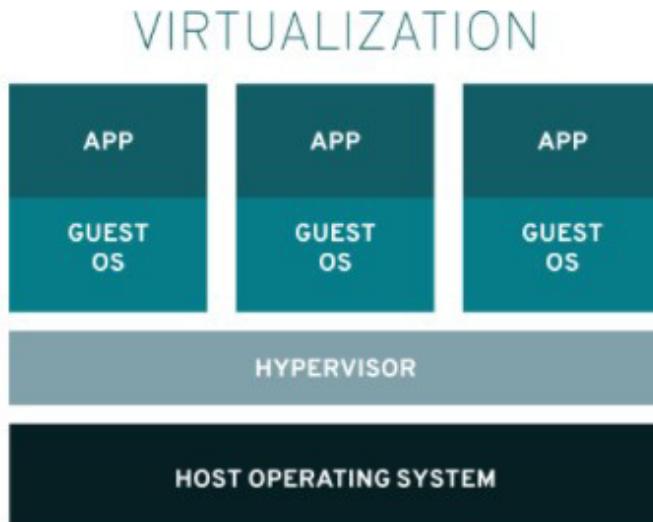
Wie unterscheiden sich die Technologien in Bezug auf Bereitstellung, Speicherplatz, Portabilität, Effizienz und Betriebssystem (Kernel)?

Dimensionen	Virtuelle Server	Container
Bereitstellung	Minuten	Millisekunden
Speicherplatz	Mehr Platz, da die Daten für das Betriebssystem dupliziert werden	Weniger Platz, da Container Images geteilt werden. Es wird lediglich die Applikation gespeichert
Portabilität	Virtuelle Server können auf Hardwarelevel verschoben werden solange der identische Hypervisor verwendet wird	Container können auf Laptops, auf Hardware im Rechenzentrum oder in der Cloud betrieben werden
Effizienz	Benötigt mehr RAM, CPU und Speicherplatz	Geringerer Ressourcenverbrauch
Betriebssystem (Kernel)	Dediziert	Geteilt

Können virtuelle Server durch Container ersetzt werden?

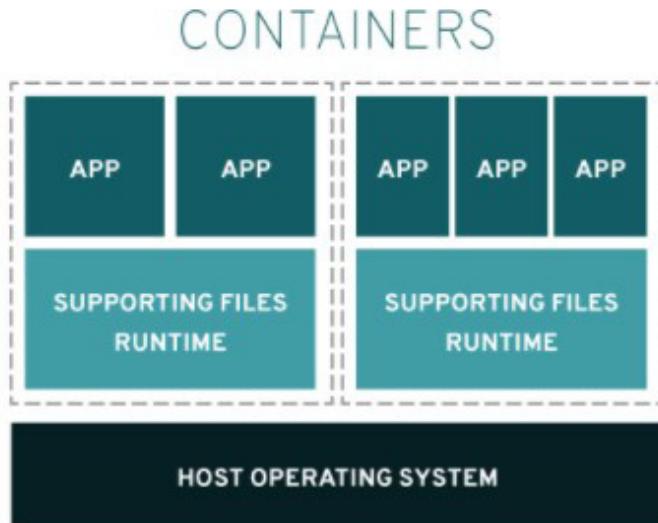


Können virtuelle Server durch Container ersetzt werden?



- Traditionelle, Legacy- und monolithische Workloads unterzubringen
- Riskante Entwicklungszyklen zu isolieren
- Infrastrukturressourcen (wie Netzwerke, Server und Daten) zu provisionieren
- ein Betriebssystem unter einem anderen Betriebssystem (z. B. Unix unter Linux) auszuführen

Können virtuelle Server durch Container ersetzt werden?



- cloudnative Apps zu entwickeln
- Microservices zu paketieren
- DevOps- oder CI/CD-Praktiken einzuführen
- skalierbare IT-Projekte in einer uneinheitlichen IT-Umgebung zu verschieben, die unter einem gemeinsamen Betriebssystem läuft

Self-Managed vs. Fully Managed Containers

Self-Managed vs. Fully Managed Containers

Self-Managed	Fully Managed
Volle Kontrolle über System und Daten	Kleinste Kontrolle über System und Daten
Höchste System Administrationskosten und Komplexität	Niedrigste System Administrationskosten und Komplexität
Zusätzliche Hardwarekosten	Keine Hardwarekosten
Zusätzliche DB und OS Lizenzkosten	Keine zusätzlichen DB und OS Lizenzkosten notwendig, bereits in der Miete inbegriffen
Kein SLA	Klar definiertes SLA

Self-Managed vs. Fully Managed Containers

Self-Managed	Fully Managed
Keine automatische Skalierung	Automatische Skalierung gemäss SLA
DR und Backup müssen zusätzlich konfiguriert werden	DR und Backup gemäss SLA
Updates müssen manuell eingepflegt werden	Updates werden automatisch eingespielt
Vollständige Transparenz des Systems	Starkes Vendor-Locking
Marginales Vendor-Locking	Wenig Transparenz des Systems

Block 4

Container Orchestrierung

Block 4

Warum braucht man Container-Orchestrierung?

Warum braucht man Container-Orchestrierung?

Provisionierung und Deployment

Konfiguration und Planung

Ressourcenzuweisung

Container-Verfügbarkeit

Skalieren oder Entfernen von Containern

Load Balancing und Traffic Routing

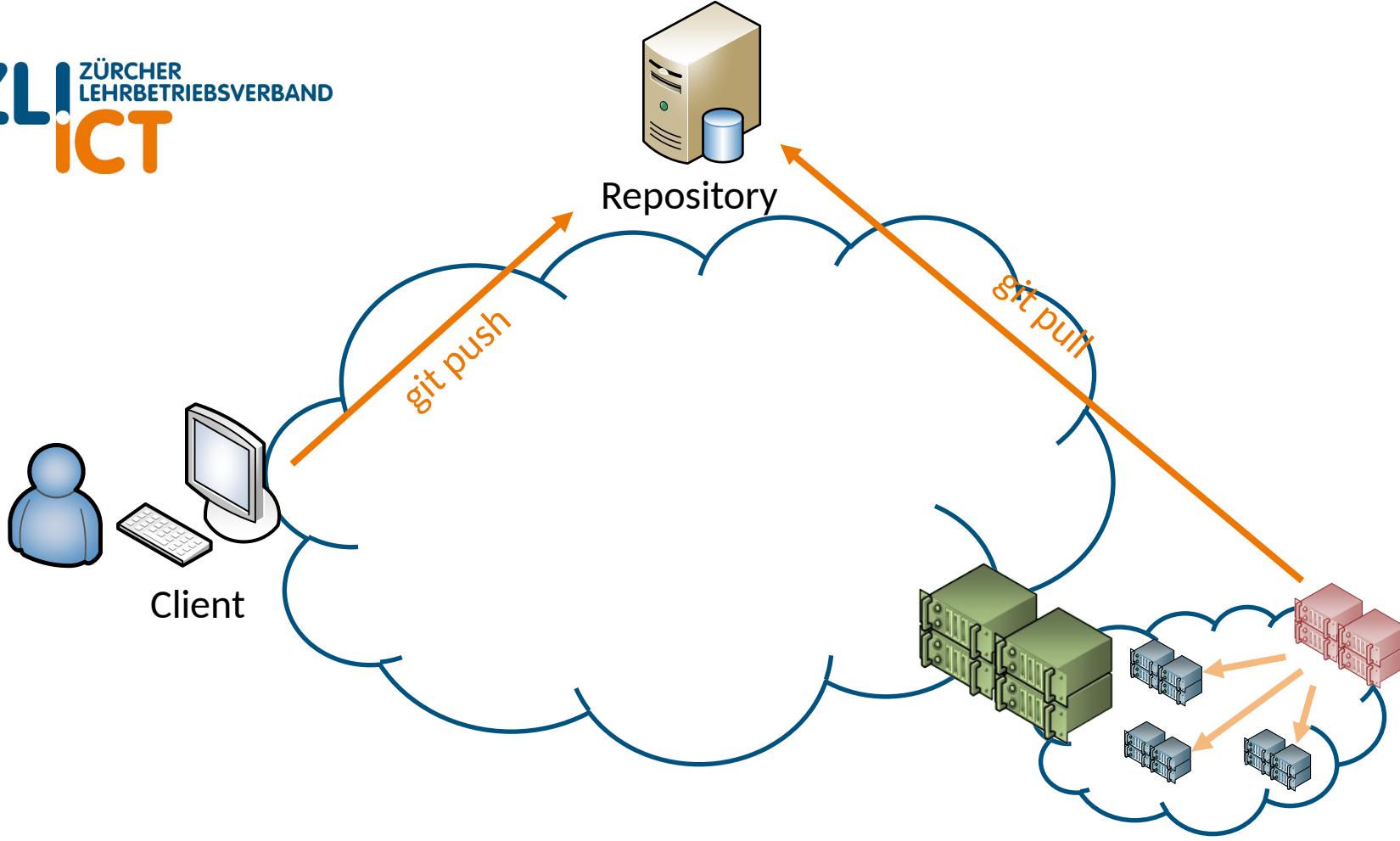
Load Balancing und Traffic Routing

Überwachen des Containerzustands

Konfigurieren von Anwendungen

Konfigurieren von Anwendungen

Sichern von Interaktionen zwischen Containern



Wie funktioniert Container-Orchestrierung

Beschreibung der **Konfiguration** einer Anwendung in einer YAML- oder JSON-Datei. Die Konfigurationsdatei teilt dem Konfigurationsmanagement-Tool mit, wo sich die Container-Images befinden, wie ein Netzwerk eingerichtet wird und wo Protokolle gespeichert werden.

Wie funktioniert Container-Orchestrierung

Beim **Deployment** eines neuen Containers plant das Container-Management-Tool automatisch das Deployment in einem Cluster, berücksichtigt alle definierten Anforderungen oder Einschränkungen und findet den richtigen Host. Das Orchestrierungs-Tool verwaltet dann den Lifecycle des Containers basierend auf den Spezifikationen, die in der Erstellungsdatei festgelegt wurden.

Wie funktioniert Container-Orchestrierung

Patterns verwalten die Konfiguration, den Lifecycle und die Skalierung von containerbasierten Anwendungen. Diese wiederholbaren Patterns sind die Tools, die ein Entwickler zum Erstellen vollständiger Systeme benötigt.

Wie funktioniert Container-Orchestrierung

Die Container-Orchestrierung kann in jeder **Umgebung** verwendet werden, in der Container ausgeführt werden, z. B. auf lokalen Servern und Public oder Private Cloud-Umgebungen

Welche Container-Orchestrierung Technologien kennen Sie?

Welche Container-Orchestrierung Technologien kennen Sie?



kubernetes



**RED HAT[®]
OPENSHIFT**



**Amazon
EKS**



Azure Kubernetes Service (AKS)



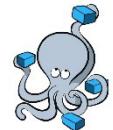
Google Kubernetes Engine



RANCHER[®]



MARATHON



**docker
Compose**



minikube



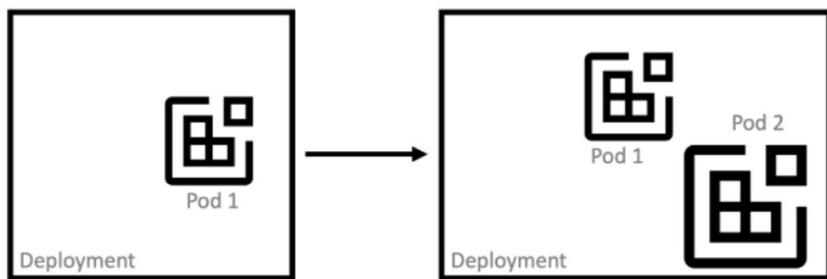
CLOUDIFY



Nomad

Was versteht man unter «scaling» von Containern?

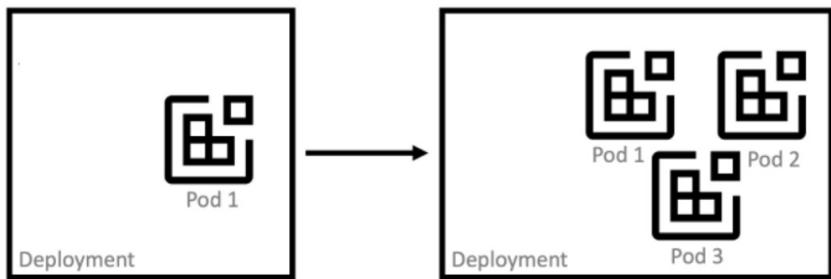
Vertikale Skalierung (Scale up)



Einige Pods können **mehr Ressourcen** nutzen können als andere.

Vertikale Skalierung eignet sich in diesem Zusammenhang eher für zustandsbehaftete Anwendungen, bei denen die Anfragen von einem bestimmten Pod bearbeitet werden müssen. Bei zustandslosen Anwendungen gibt es diese Anforderung nicht, so dass es einfacher ist, die horizontale Skalierung zu verwenden, und Sie haben den Vorteil, dass jeder Pod auf dieselbe Weise definiert ist.

Horizontale Skalierung (Scale out)



Es werden **mehr Instanzen** der Pods gestartet.

Horizontale Skalierung ist das, was Kubernetes am besten kann. Dafür wurde es ja auch entwickelt! Pods, die in einer Kubernetes-Bereitstellung ausgeführt werden, können mühelos repliziert werden, sodass eine horizontale Skalierung innerhalb von Sekunden möglich ist. Der in Kubernetes integrierte Load Balancer ermöglicht eine nahtlose Konfiguration. Mit dem horizontalen Pod-Autoscaler (HPA) können Sie die minimale und maximale Anzahl der Pods angeben, die Sie ausführen möchten, sowie die CPU- oder Speicherauslastung, die Ihre Pods anstreben sollten.