Cloud computing

Autor: Björn Böing

Inhaltsverzeichnis

Abkürzungsverzeichnis

AMI	Amazon Machine Image
API	Application Programming Interface
ARN	Amazon Resource Name
ASP	Application Service Provider
AWS	Amazon Web Services
AZ	Availability Zone
BSI	Bundesamt für Sicherheit in der Informationstechnik
CSA	Cloud Security Alliance

DbaaS	Database as a Service
DDoS	Distributed-Denial-of-Service
DNS	Domain Name System
EC2	Elastic Compute Cloud
EULA	End-User Licence Agreement
EU-DSGVO	EU-Datenschutz-Grundverordnung
FaaS	Function as a Service
GaaS	Games as a Service
HPE	Hewlett Packard Enterprise
HTTP	Hypertext Transport Protocol
IaaS	Infrastructure as a Service
IAM	Identity and Access Management
IdP	Identity Provider
IMAP	Internet Message Access Protocol
IoT	Internet of Things
IoTaaS	IoT as a Service
IIoT	Industrial Internet of Things
KI	Künstliche Intelligenz
NASA	National Aeronautics and Space Administration
NIST	National Institute of Standards and Technology
MIT	Massachusetts Institute of Technology
MQTT	Message Queuing Telemetry Transport
OAuth	Open Authentication
OPC	Open Platform Communication
OSI	Open Systems Interconnection
PaaS	Platform as a Service
QoS	Quality of Service
REST	Representational State Transfer
SaaS	Software as a Service
SOA	Service orientierte Architektur
SSH	Secure Shell
STS	Security Token Service
S3	Simple Storage Service
VM	Virtuelle Maschine
VMM	Virtuelle Maschine Monitor
XaaS	Everything as a Service

Einleitung

In diesem Kapitel werden eine Vielzahl von Facetten und Themengebieten vorgestellt und teils detailliert betrachtet, die zu dem Oberbegriff "*Cloud Computing*" gehören. Entstanden ist dieses Kapitel als Ausarbeitung für das Modul "*Spezielle Gebiete zum Software Engineering*" (Sommersemester 2018) des Master Studiengangs Informatik, an der Fachhochschule Minden - Campus Minden.

Definition

Für den Begriff "Cloud Computing" gibt es keine Definition, die sich zu diesem Zeitpunkt allgemeingültig durchsetzen konnte, jedoch ähneln sich die meisten häufig in den Kernpunkten. Die Definition des US-amerikanischen National Institute of Standards and Technology (NIST) wird in vielen Publikationen und Vorträgen verwendet und lautet:

"*Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.*" [NIST11]

Neben dem Kernpunkt aller Definitionen, dass Rechnerressourcen über ein Netzwerk bereitgestellt und genutzt werden können, stellt das NIST mit dieser Definition eine schnelle und einfache Verwendung in den Fokus. Das Bundesamt für Sicherheit in der Informationstechnik (BSI) geht noch einen Schritt weiter und sagt, dass Angebote und Nutzung der Cloud Dienstleistungen ausschließlich über definierte technische Schnittstellen und Protokolle abläuft. Die direkte Interaktion mit den Anbietern ist nicht vorgesehen. [BSI18]

Geschichtliches

Die Bedeutung von "Cloud Computing" wie sie heute existiert und was damit verbunden wird, hat sich seit der ersten Verwendung nicht viel verändert. Erstmal verwendete die Compaq, Weltmarktführer für PC-Systeme der 1990er, den Begriff 1996 in einem internen Dokument. Schon davor wurde der Begriff "Cloud" und die dazugehörige Abbildung verwendet, um unter anderem das Internet, Telekommunikation und verteilte Anwendungen darzustellen. [REGA11]

Populär wurde der Begriff "Cloud Computing", als Amazon 2006 ihre Elastic Compute Cloud (EC2) auf den Markt brachte. In den anschließenden Jahren brachten auch Unternehmen wie Google, Microsoft, IBM und auch Oracle vergleichbare Produkte auf den Markt, um den neu erzeugten Bedarf nach externer und mietbarer Rechenleistung zu bedienen. Das erste open-source Projekt, welches das Erstellen von privaten und hybrid Clouds ermöglichte, war das OpenNebula Projekt der National Aeronautics and Space Administration (NASA) und wurde 2008 veröffentlicht. [FOOT17] [IBMJ09]

Grundlagen & weiterführende Konzepte

In diesem Abschnitt werden zum einen grundlegende Informationen, wie die Service- und Bereitstellungsmodelle im Cloud Bereich, vorgestellt, aber andererseits auch auf weiterführende Konzepte wie Edge Computing eingegangen. Darüberhinaus werden verschiedene Nachteile & Begrenzungen betrachtet, die Cloud Computing mit sich bringt.

Charakteristika

Das NIST stellt für einen Großteil von IT-Bereichen Definitionen bereit und kümmert sich hauptsächliche darum, Standards vorzuschlagen und auszuarbeiten, die von einem breitgefächerten Spektrum an Industrien und staatlichen Einrichtungen eingesetzt werden sollen. In der Definition des NIST zu Cloud Computing werden die folgenden fünf Kern-Charakteristika beschrieben [NIST11]:

- **On-demand self-service**
  Ein Nutzer kann ohne menschliche Interaktion (also eigenständig) die zugänglichen Ressourcen, wie Serverinstanzen und Speicher, verwalten.
- **Broad network access**
  Die angebotenen Funktionen sind über das Netzwerk und mittels standardisierter Mechanismen zu erreichen und darauf ausgerichtet Client-Plattformen wie Smartphones, Tablets oder Laptops zu unterstützen.
- **Resource pooling**
  Die Ressourcen des Anbieters sind darauf ausgelegt von mehreren Kunden parallel genutzt zu werden. Dies wird erreicht, indem sowohl die physischen als auch die virtuellen Ressourcen einem Kunden automatisiert zugewiesen und entzogen werden.
- **Rapid elasticity**
  Funktionen können elastisch bereitgestellt und freigegeben werden, um eine Skalierung zu ermöglichen, die sich (manchmal auch automatisiert) den Umständen entsprechend anpasst. Dem Nutzer erscheinen die Ressourcen häufig als unbegrenzt und können dadurch zu beliebigen Zeitpunkten, in beliebigen Mengen angefordert werden.
- **Measured service**
  Die Nutzung von Cloud Systemen wird automatisiert überwacht, um beispielsweise den genutzten Speicher, die genutzte Bandbreite oder die Anzahl der aktiven Benutzer zu messen. Diese können sowohl von Seiten des Nutzers, als auch vom Anbieter aus transparent überwacht, kontrolliert und bekanntgegeben werden.

Eine Definition der "*Cloud Security Alliance*" (CSA) nennt neben der Charakteristika zu "rapid elasticity" und "on-demand self-service" die folgenden Eigenschaften:

- **Service orientierte Architektur** (SOA)
  Eine der Grundvoraussetzungen für Cloud Computing. Die Cloud-Dienste werden in der Regel über sogenannte Representational State Transfer (REST)-Schnittstellen angeboten.

- **Mandantenfähigkeit**
  Nutzer teilen sich in einer Cloud-Umgebung gemeinsame Ressourcen, weshalb diese mandantenfähig sein müssen.

- **Pay per Use**
  Es müssen nur Ressourcen bezahlt werden, die auch tatsächlich in Anspruch genommen werden. Ausnahmen bilden Flatrate-Modelle.

Zu diesen verbreiteten Charakteristika sind über die Jahre weitere Faktoren hinzugekommen, die je nach Anwendungsfall mehr oder weniger bedeutsam sind.

[BSI18] [NIST11]

Servicemodelle

Auf Basis von Cloud Computing Technologien sind in den vergangenen Jahren eine Vielzahl verschiedener Projekte und Produkte entstanden, die vor allem darauf abzielen dem Nutzer Funktionalitäten flexibel und skalierbar zur Verfügung zu stellen. Da Nutzer in der Regel keine Funktionalitäten wirklich kaufen, sondern eher mieten, wird in diesem Zusammenhang von "Services", also Diensten, gesprochen. In diesem Zusammenhang ist auch die Begrifflichkeit "as a Service" entstanden, mit denen Cloud Computing Produkte häufig betitelt werden. Die finanzielle Abrechnung läuft bei Cloud Services oft über das bereits erwähnte *"Pay per Use"* Prinzip ab.

Die Abbildung 1 zeigt den Zusammenhang der drei verbreitetsten Servicemodelle *"Infrastructure as a Service" (IaaS)*, *"Plattform as a Service" (PaaS)* und *"Software as a Service" (SaaS)* in Form eines Mengendiagramms. In den nächsten Abschnitten sollen diese drei Modelle und weitere vorgestellt und deren Anwendungsfälle betrachtet werden.

Abb. 1: IaaS vs PaaS vs SaaS - Quelle: [GASS16]

Infrastructure as a Service (IaaS)

Mit IaaS wird das Bereitstellen von IT-Infrastrukturen mittels high-level Application Programming Interfaces (APIs) beschrieben, welche Details und Funktionen auf eine höhere Abstraktionsebene anheben, um deren Benutzung zu vereinfachen. Typische Infrastrukturen, die als Service bereitgestellt werden sind Rechenleistung und Speicherplatz.

Über sogenannte "Hypervisor" oder "Virtual Machine Monitor" (VMM) werden auf einem Computer eine oder mehrere virtuelle Maschinen (VMs) gestartet und verwaltet. Der Computer, auf dem die VMs laufen, wird "Host" genannt, während die laufenden VMs als "Guest" bezeichnet werden. Anzumerken ist, dass das Betriebssystem des Host-Systems nicht einschränkt, welche Betriebssysteme auf den VMs genutzt werden können. Im Bereich von Cloud Computing werden Orchestrierungs Technologien genutzt, um unter anderem die Entscheidung auf welchem Host eine VM laufen soll und auch das Verknüpfen von neuen VMs und freiem Speicher zu automatisieren. Dadurch wird ermöglicht, dass eine Vielzahl von Benutzer neue VMs eigenständig aufsetzen und nutzen können, ohne dass Interaktionen mit einem Dritten notwendig sind. In der Regel stellen Anbieter von IaaS ein Webportal zur Verfügung, über das neue Maschinen mit ein paar Klicks erzeugt werden können und dabei sowohl das Betriebssystem als auch die Rechen- und Speicherleistung festgelegt werden. [SHAW17] [ROUS17a]

Die NIST hat bereits 2011 standardisierte Definitionen zu den weitverbreitesten Servicemodellen veröffentlicht. Das IaaS Modell wird wie folgt definiert:

> *"The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage or*

control the underlying cloud infrastructure but has control over operating systems, storage, and deployed applications; and possibly limited control of select networking components (e.g., host firewalls)." [NIST11]

Auch diese Definition beinhaltet, dass die grundlegende Infrastruktur vom Anbieter verwaltet und bereitgestellt wird und der Nutzer trotzdem die Kontrolle über das Betriebssystem, den Speicher und installierte Software hat. Die NIST merkt allerdings an, dass häufig die Kontrolle über Netzwerkkomponenten beschränkt sind und nennt dazu Einstellungen an der Firewall als Beispiel.

Platform as a Service (PaaS)

Das PaaS Modell erweitert das vorhergehend beschriebene IaaS Modell mit vorinstallierter Software (häufig speziell Middleware), die für die Entwicklung und den Betrieb von Anwendungen notwendig sind und übernimmt weitere Aufgaben, wie das Konfigurieren des Betriebssystems, von Datenbanken und von technischen Bibliotheken. Das bedeutet, dass beispielsweise Server aufgesetzt werden können, die die JavaScript Laufzeitumgebung Node.js schon installiert haben und der Server somit lediglich die auszuführende Software erhalten muss. [GASS16] [WATT17]

Vorrangiges Ziel von PaaS ist es die Entwicklung von Softwareanwendungen zu vereinfachen und zu beschleunigen. Dies wird erreicht, indem Entwickler sich nicht detailliert mit dem Erzeugen und Konfigurieren von Testumgebungen befassen müssen, sondern auf fertige Lösungen zurückgreifen beziehungsweise auf einfache Weise diese Lösungen individuell anpassen können. Diese Kernpunkte sind auch von der NIST in ihrer Definition von PaaS zu finden:

"*The capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages, libraries, services, and tools supported by the provider. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly configuration settings for the application-hosting environment.*" [NIST11]

Fotango, eine Tochterfirma von Canon Europe, brachte 2006 als weltweit erster Anbieter einen Dienst online, der das umsetzte, was später als PaaS bekannt wurde. Mit "*Zimki*" wurde damals eine Plattform angeboten, die alle Nebenaufgaben für die Entwicklung von JavaScript automatisiert umsetzte und dabei das "*Pay per Use*" Prinzip für die finanzielle Abrechnungen nutze. Obwohl Zimki Profit einspielte und sich einer wachsenden Nutzerzahl erfreuen konnte, entschied sich Canon den Dienst am 24.12.07 einzustellen. [FORR06] [MARK07]

Software as a Service (SaaS)

Mit SaaS stellt ein Anbieter das volle Paket, bestehend aus Infrastruktur, Plattform und auch Anwendungssoftware, für die direkte Nutzung bereit. Dabei steht die Anwendersoftware im Vordergrund und wird häufig über einen Thin-Client, wie einen Internetbrowser, bedient. Durch diesen Ansatz muss ein Anwender keinerlei produktspezifische Software installieren, sondern kann die volle Funktionalität jederzeit auf Abruf nutzen. Aus diesem Grund wird SaaS auch als "*On Demand Software*" bezeichnet und gilt als das komplette Gegenteil zur klassischen "*On Premises Software*", bei der ein Produkt auf dem Nutzerrechner vollständig installiert werden muss, bevor es genutzt werden kann. [GASS16] [WATT17]

Im Vergleich zu IaaS, was sich eher an IT-Administratoren richtet und zu PaaS, was vorrangig von Software-Entwicklern genutzt wird, findet bei SaaS die Interaktion direkt mit dem Nutzer statt. Als Beispiel sind die Google Apps wie Docs, Spreadsheets und Presentation zu nennen, deren Funktionalität vollständig über den Browser zu nutzen sind und keinerlei Zusatzinstallationen benötigen. [GASS16]

Die NIST definierte 2011 SaaS wie folgt:

"*The capability provided to the consumer is to use the provider's applications running on a cloud infrastructure. The applications are accessible from various client devices through either a thin client interface, such as a web browser (e.g., web-based email), or a program interface.The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user-specific application configuration settings.*" [NIST11]

Die Grundidee, dass einem Kunden die Aufgaben zur Wartung und Bezahlung von IT-Infrastrukturen und Plattformen abgenommen werden, ist nicht erst mit SaaS entstanden. Bereits in den 1990ern sind sogenannte "*Application Service Provider*" (ASP) entstanden, die Softwareanwendungen über Netzwerke bereitstellten. Auch ASPs sahen ein Geschäftsmodell darin, dem Kunden die Arbeit abzunehmen, die abseits von der eigentliche Bedienung und Verwendung einer Softwarelösung notwendig ist. Stattdessen war es nur notwendig Client-Software zu installieren, was gerade für kleine und mittelständische Unternehmen eine enorme Kostenersparnis darstellte. [BIAN20]

Im Vergleich zu ASPs stellen SaaS Anbieter in der Regel ihre eigene Software über die Cloud bereit, anstatt Software von Dritten anzubieten. Darüber hinaus wird für SaaS-Lösungen in den meisten Fällen einzig ein Internetbrowser benötigt und keine separat installierte Clientanwendung. Was den Unterschied in Wartung und Betrieb betrifft, so stellten ASPs für jeden ihrer Kunden eine eigene Instanz der gewünschten Anwendung zur Verfügung, während modere SaaS Lösungen, mit einem mandantenfähiges System, mehrere Kunden über eine einzige Instanz versorgen können. Sowohl ASP als auch SaaS unterstützen, durch die Zentralisierung der laufenden Softwareanwendungen, das Prinzip von Continuous Delivery. Dies bedeutet, dass die Anwendungen, die über das Netzwerk bereitgestellt werden, in deutlich höherer Frequenz Updates erhalten können, ohne dabei den Endnutzer mit einbeziehen zu müssen. Anzumerken ist, dass SaaS Continuous Delivery stärker unterstützt als ASPs, da dort die Client-Software in der Regel keine Updates benötigt. [BIAN20]

Function as a Service (FaaS)

Der Begriff FaaS ist aus dem Bereich der *"serverless"* Architekturen entstanden. Initial wurde mit serverless beschrieben, was heute mit PaaS gemeint ist. Also, dass eine Anwendung und die dazugehörigen Server, von Dritten bereitgestellt und gewartet werden, statt diese Aufgaben selbst zu erledigen.

Das heutige Verständnis einer serverless Architektur beschreibt allerdings tatsächlich einen Ansatz, der sich vom klassischen Server entfernt. Gemeint ist, dass nicht ein Prozess eine lange Zeit darauf wartet, dass eine Anfrage gestellt wird und dieser Prozess möglicherweise auch mehrere Anfragen abwickelt. Stattdessen setzt der FaaS Ansatz darauf, dass die geforderten Ressourcen innerhalb von Millisekunden hochfahren, anschließend eine Anfrage behandeln und dann wieder herunterfahren. Dies wird über Event-Mechanismen umgesetzt, die zur Ausführung einer bestimmten Funktion führen statt, dass ein Prozess auf Anfragen warten muss. Im Vergleich zu PaaS wird die Nutzung von FaaS meist pro Ausführung bepreist statt pro verstrichener Zeit in dem der Dienst aktiv war.

Mit FaaS werden mittlerweise vor allem Microservices realisiert, die durch eine Orchestrierung von verschiedenen Funktionalitäten umfassende Aufgabe erledigen. Amazon ist mit AWS Lamba einer der bekanntesten Anbieter von FaaS, obwohl hook.io es weltweit als erster anbot.

[AVRA16] [HAN17]

Everything as a Service (XaaS)

Nachdem die vorangehend beschriebenen Servicemodelle immer weiter an Bekanntheit und Beliebtheit gewannen, kamen weitere Technologien die mittels Cloud Computing *"as a Service"* angeboten wurden. All diese verschiedenen Modelle und Technologien werden unter *"Everything as a Service"* (XaaS) zusammen gefasst. Heutzutage werden die verschiedensten Anwendungen und Technologien über die Cloud als Service bereitgestellt wie beispielsweise:

- Internet of Things (IoT) as a Service (IoTaaS)
- Database as a Service (DbaaS)
- Blockchain as a Service (BaaS)
- Games as a Service (GaaS)

Genau wie die vorhergehenden Modelle basieren auch diese darauf, dass der Endnutzer möglichst wenig Aufwand betreiben muss, um sie zu nutzen.

[ROUS17b] [WIKI18]

Bereitstellungsmodelle

Durch die große Verbreitung und die verschiedensten Anwendungsgebiete sind auch die verschiedensten Formen der Bereitstellung von Cloud Computing entstanden. Die vier verbreitesten werden im Folgenden vorgestellt.

Public

Eine *"Public Cloud"* beschreibt eine öffentlich zugängliche Cloud-Infrastruktur, bei der die benötigten Ressourcen von einem Dritten bereitgestellt und gewartet werden. Die Nutzung von Public Clouds kann zwar kostenlos sein oder zumindest bis zu einem gewissen Grad, ist in der Regel allerdings mit direkten Kosten an den Anbieter verbunden. Dieses Bereitstellungsmodell ist mit Abstand am weitesten vertreten und wir sowohl von Privatleuten als auch von Unternehmen verwendet, um Arbeit auszulagern und die eigene Produktivität zu steigern. [INNO17] [NIST11]

Private

Unternehmen, die die Cloud-Infrastruktur eines Dritten allein nutzen oder die die Bereitstellung und Wartung selbstständig erledigen wollen, greifen häufig auf eine *"Private Cloud"* zurück. Diese erlaubt es ihnen die vorhergehend beschriebenen Servicemodelle zentral zu betreiben und müssen so nicht zwingend auf einen Dritten vertrauen. Ein Unternehmen kann somit sowohl Betreiber als auch Nutzer einer Cloud sein und darüberhinaus auch Zweigstellen Zugriff auf die Private Cloud ermöglichen. [INNO17] [NIST11]

Community

Eine *"Community Cloud"* siedelt sich zwischen Public und Private Clouds an. Die genutzte Infrastruktur wird zwischen mehreren Unternehmen oder Teilnehmern geteilt, ist aber nicht öffentlich zugänglich. In der Regel schließen sich Unternehmen zusammen, die beispielsweise die gleichen Vorraussetzungen an Privatsphäre, Performanz und Sicherheit stellen und durch den Zusammenschluss die entstehenden Kosten aufteilen wollen. Community Clouds können innerhalb des Zusammenschlusses bereitgestellt, aber auch durch einen Dritten bezogen werden. Über ein mandantenfähiges System können einzelne Unternehmen separat von einander agieren, es lassen sich aber trotzdem zentrale Datenquellen erzeugen, auf die mehrere Teilnehmer Zugriff haben. Ein Beispiel für einen Zusammenschluss sind Banken, bei denen mehrere Tochterfirmen eigenständig arbeiten, aber trotzdem zentrale Datenquellen mit einbeziehen. [INNO17] [NIST11]

Hybrid

Beim Einsatz von "*Hybrid Clouds*" werden zwei oder mehr Clouds miteinander verbunden, die unterschiedliche Bereitstellungsmodelle besitzen können. Unternehmen können dadurch beispielsweise sensible Daten mit einer Private Cloud schützen und gleichzeitig andere Daten durch eine Public Cloud öffentlich zugänglich machen. Außerdem ermöglichen Hybrid Clouds einen schritt- oder teilweisen Umstieg von on premise Anwendungen zur Cloud. Dies wird vor allem von Unternehmen genutzt, die einen Umstieg zur Cloud nicht in einem einzigen Schritt stemmen können. Das unter Geschichtliches erwähnte Projekt OpenNebula der NASA ist ein Beispiel für die gleichzeitige Nutzung von Private und Public Cloud. Erstere nutzt die NASA, um Forschung und Entwicklung unter Verschluss zu halten, während letztere dazu genutzt wird, um Datensätze mit anderen Unternehmen und der Öffentlichkeit auszutauschen. [INNO17] [NIST11]

Edge Computing

Mit dem Begriff "Edge Computing" wird der Ansatz beschrieben, mit dem die Intelligenz eines Netzwerkes zur Datenquelle verschoben wird. Gemeint ist, dass Rechenleistung und Speicher an die "Edge", also die Kante, eines Netzwerkes gebracht werden, um vor Ort die dort entstehenden Daten zu verarbeiten. Edge Computing befindet sich derzeit auf dem aufsteigenden Ast und wird immer häufiger eingesetzt, um vor allem im IoT-Bereich eine bessere Kontrolle und Performanz zu erreichen. Einige wichtige Entitäten und Begriffe für Edge Computing sind:

- **Edge device**
  Geräte, die Daten erzeugen, wie beispielsweise Sensoren und industrielle Maschinen.
- **Edge**
  Der "*Rand*" des betrachteten Netzwerkes, der je nach Anwendungsgebiet völlig unterschiedlich ist. In der Telekommunikation könnte es ein Sendemast oder Handy sein, im Automobilbereich ein Auto.
- **Edge gateway**
  Eine zentrale Stelle, von der die Verbindung zu Bereichen außerhalb eines Edge Netzwerkes ermöglicht wird.
- **Fat client**
  Im Bereich von Edge Computing ist Software gemeint, die auf den Edge devices Daten verarbeiten können. Das Gegenstück zu "*thin clients*", die Daten nur versenden können.
- **Edge computing equipment**
  Edge Computing nutzt Hardware und Technologien, die bereits in anderen Bereichen eingesetzt wird, als auch welche, die speziell für diese und ähnliche Szenarien entwickelte wurde. Hersteller wie Cisco produzieren gezielt Netzwerk-Hardware, die besonders zuverlässig und belastbar sind (sowohl die Technik als auch das Gehäuse).
- **Mobile edge computing**
  Der Ausbau von Edge computing im Bereich der Telekommunikation, speziell in 5G Szenarien.

Der Wandel, der durch Edge Computing vollzogen wird, ist in mehreren Bereichen sehr vielversprechend und bringt einige Vorteile mit sich.

Zum einen senkt es die Latenz, die eine Anwendung oder ein Gerät für eine Entscheidungsfindung oder ein Ergebnis braucht. Eine Anfrage quer über den Globus zu schicken, wo eventuell weitere Anfragen entstehen, ehe eine Antwort zurück kommt, dauert deutlich länger, als auf alle benötigten Mittel vor Ort zugreifen zu können. Laut Matthew Lynley von "techcrunch.com" [LYNL18] entwickelt Amazon möglicherweise Chips für Amazon Echo, um genau diesem Latenz-Problem entgegen zu wirken. Diese Chips sollen die Informationen und Anfragen, die in die Cloud geschickt werden müssen, senken, um so die Antwortzeit drastisch zu reduzieren.

Zum anderen bietet Edge Computing die Möglichkeit über ein richtiges Management die Sicherheit, beispielsweise von Nutzern und deren Geräten, zu steigern. Spätestens nach dem Distributed-Denial-of-Service (DDoS) Angriff auf die Dyn im Oktober 2016 [STAT16] ist die Sicherheit von Edge Geräten, wie sie häufig für IoT-Netzwerke benutzt werden, ein großes Thema. Damals konnte ein riesiges Botnetz, das zum Großteil aus IoT-Geräten bestand, die Verwendung des Internets dramatisch stören. Die Mirai Malware übernahm die Kontrolle von Geräten, deren Nutzernamen und Passwörter denen der Werkseinstellungen entsprachen und nutzte die so gesammelte Rechenleistung für eine DDoS Attacke gegen den Domain Name Service (DNS)-Betreiber der USA.

Ebenso wie Werkseinstellungen von Nutzernamen und Passwörter, bringen auch veraltete Betriebssysteme und Software Sicherheitsrisiken mit sich. Laufende IoT-Geräte werden nur selten mit aktuellen Updates ausgestattet und beinhalten wenige Sicherheitsmechanismen. Statt händisch einzelne Geräte zu updaten sollte stattdessen ein zentrales Management diese Aufgabe übernehmen und für mehr Sicherheit am Netzwerkrand sorgen. Genauso wie Webbrowser meist verdeckt Updates erhalten oder Smartphone-Besitzer auf neue Versionen hingewiesen werden, sollte es auch bei Edge-Geräten der Fall sein.

Neben den Sicherheitsaspekten kann Edge Computing auch dabei helfen ein weiteres Problem zu lösen, das durch IoT entstanden ist. Die enorme Menge an Daten, die durch IoT-Geräte anfallen und versendet werden sollen, bringen die Bandbreite eines Netzwerks an ihre Grenzen. Statt alle Daten zur Verarbeitung und Speicherung in die Cloud zu senden, können intelligente Edge Geräte dabei helfen die Daten zu filtern und nur bedeutsame Informationen über die Leitung zu senden. Zur Filterung sollen vor allem künstliche Intelligenzen (KIs) auf die Endgeräte gelangen.

Getrieben wird der Wandel zum Edge Computing vor allem durch die Industrie. Mit der Einführung von Industrial Internet of Things (IIoT) generieren, versenden und analysieren Unternehmen ihre Prozesse und senken so beispielsweise ihre unerwarteten Ausfallzeiten. Obwohl bei der riesigen Datenmenge die starke Rechenleistung und große Speicherkapazitäten der Cloud eine sehr zentrale Rolle spielen, so bietet Edge Computing weitere Möglichkeiten die Performanz des IIoT weiter zu verbessern. Möglich ist dies vor allem durch die niedrigen Preise von Geräten und Sensoren zur Herstellung von Edge Geräten, die darüber hinaus immer weniger Platz benötigen. Ebenso dient die steigende Anzahl an

Anwendungsgebieten von IIoT und die modernen Technologien für maschinelles Lernen und Analysen als treibende Kraft für den Einsatz von Edge Computing in der Industrie.

Die Zentralisierung, auf der Cloud Computing aufbaut, wird durch Edge Computing etwas aufgelockert. Das heißt nicht, dass es überflüssig wird, sondern vielmehr, dass Aufgaben an den Rand des Netzwerkes abgeben werden. Je nach Anwendungszenario ist diese Verschiebung stärken oder schwächer ausgeprägt. In Extremfällen kann Edge Computing aber auch vollständig ohne die Anwendung von zentralem Cloud Computing stattfinden.

[BUTL17] [GEDI18] [MILL18] [FELD17]

Fog Computing

Genau wie beim Edge Computing, wird auch beim Fog Computing Rechenleistung, Speicher und allgemein die digitale Intelligenz zurück zum Rand des Netzwerkes verschoben. Dies soll ebenfalls für geringere Latenzen, eine höhere Sicherheit und geringere Auslastung der Bandbreite sorgen.

Der Begriff "Fog Computing" wurde 2013 erstmals von Cisco bei einer Publizierung verwendet und wird bis heute auch vorrangig von Cisco geprägt. Am 19. November 2015 wurde das "OpenFog Consortium" gegründet, in welchem Unternehmen wie Cisco Systems, ARM Holdings, Dell, Intel, Microsoft und die Princeton University zusammen arbeiten, um Fog Computing zu verbreiten und zu standardisieren. Die bisherige Arbeit brachte die sogenannte "OpenFog Reference Architecture" hervor, welche die acht Säulen einer OpenFog Architektur detailliert betrachtet [OPEN17]:

- Sicherheit
- Skalierbarkeit
- Offenheit
- Selbstständigkeit
- Programmierbarkeit
- RAS (Zuverlässigkeit, Erreichbarkeit und Wartbarkeit)
- Agilität
- Hierarchie

Die Unterschiede zu Cloud und Edge Computing sind die Nähe zum Endverbraucher, die dichte geografische Verteilung und die mobile Einsatzmöglichkeit. Das standardmäßige Prinzip von Cloud Computing wird dahingehend verändert, dass durch sogenannte Fog-Nodes mehrere Endgeräte verknüpft werden und diese die Kommunikation zur Cloud übernehmen, statt die Endgeräte direkt. Durch die erhöhte Leistung sind die Fog-Nodes in der Lage Aufgaben der Cloud (z.B. Filterung von gesammelten Daten) zu übernehmen und darüberhinaus auch als eine Art von "vorgelagerter Cloud" betrachtet werden kann.

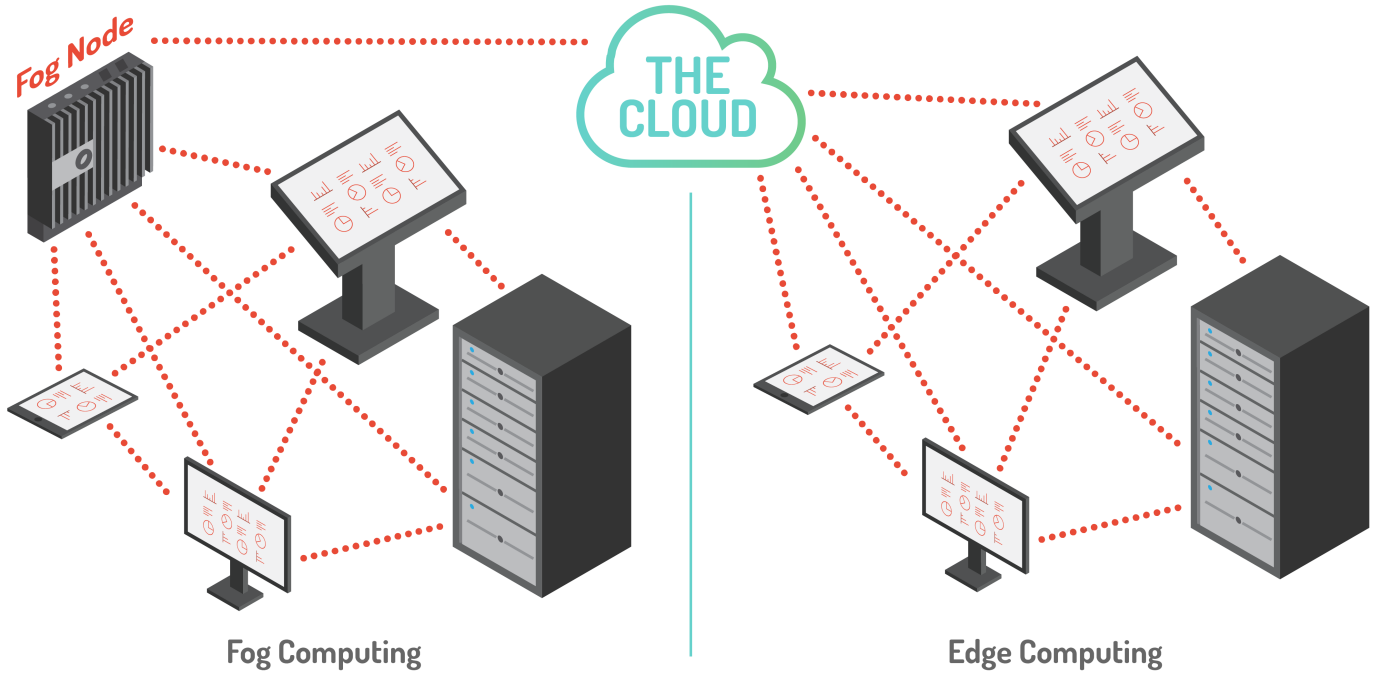

Abb. 2: Edge vs Fog Computing - Quelle: [ELLE17]

Im Vergleich zu Edge Computing, wo die Endgeräte (z.B. IoT-Dinge) eine stärkere Rechenleistung, Speicherkapazität und Intelligenz erhalten, bleiben die Endgeräte im Fog-Computing meist leistungsschwach. Stattdessen sind Gateways oder Fog-Nodes mit starker Leistung ausgestattet, um die bereits genannten Vorteile auszuspielen. Die Abbildung 2 zeigt genau diesen Unterschied, in einer vereinfachten Darstellung.

Bei Fog Computing sind mehrere Schritte nötig, ehe Daten an die Cloud gesendet werden. Davor werden sie:

1. über standardmäßig I/O Mechanismen ausgelesen.
2. von einem Open Platform Communication (OPC)-Server oder einer IoT-Node in ein Internet Protokoll wie Message Queuing Telemetry Transport (MQTT) oder Hypertext Transfer Protocol (HTTP) konvertiert.
3. an eine Fog-Node im Netzwerk verschickt, wo sie gefiltert und/oder verarbeitet werden.

Ebenso wie Edge Computing kann auch Fog Computing unabhängig von Cloud Computing umgesetzt werden. In der Praxis wird aber auch Fog Computing lediglich als Ergänzung zur Cloud genutzt, um vor allem Endnutzern eine erhöhte "*Quality of Service*" (QoS) zu bieten.

Die schrittweise Verarbeitung und Konvertierung von gesammelten Daten stellt einen komplexeren Ansatz dar, als es bei Edge Computing der Fall ist. Darüberhinaus entstehen durch Fog Computing weitere potenzielle "*Points of Failure*".

[CISC13] [ELLE17] [OPEN17] [OPTO18]

Nachteile & Begrenzungen

Cloud Computing bietet durch die Erschwinglichkeit, Effektivität und Skalierbarkeit viele Vorteile für Unternehmen. Kosten, Arbeitsaufwand und auch benötigtes Know-How können von einen Anbieter übernommen werden und ermöglicht den Unternehmen dadurch, sich auf das eigene Tagesgeschäft zu fokussieren. Trotz alledem bringt der Umstieg zur Cloud auch Nachteile und Sicherheitsrisiken mit sich.

Distributed-Denial-of-Service (DDoS) Angriffe

Obwohl Cloud Anbieter über eine riesige Menge an Ressourcen verfügen und Kunden dadurch eine flexible Skalierung angeboten werden kann, so sind DDoS Angriffe auf Cloud Systeme mittlerweile durchaus denkbar. Dies wird vor allem durch die stetig steigende Menge von IoT Geräten ermöglicht, von denen viele über mangelhafte Sicherheitsmechanismen verfügen.

Der bereits im Abschnitt Edge Computing angesprochene Angriff auf den amerikanischen DNS-Betriber Dyn im Jahr 2016 zeigt, dass auch große Organisationen nicht immer vor DDoS sicher sind. Ganz davon abgesehen, dass Kunden die Option zur automatischen Skalierung zusätzlich buchen müssen, können Cloud Server trotzdem mit genügend Traffic in die Knie gehen oder zumindest eine geringere QoS bieten.

Unternehmen, die ihre eigene Cloud Infrastruktur aufbauen, können einerseits durch geringere Ressourcen deutlich angreifbarer sein, andererseits können Angriffe auf Cloud Systeme wie beispielsweise AWS oder Azure dazu führen, dass viele Kunden gleichzeitig von einem DDoS Angriff betroffen sind.

Und selbst, wenn der Cloud Anbieter die geeigneten Maßnahmen gegen solche Angriffe trifft, können auch weitere, unglückliche Ereignisse dafür sorgen, dass der Dienst eines Unternehmens nicht erreichbar ist. Im April 2017 waren große Teile des Simple Storage Service (S3) von Amazon für kurze Zeit nicht erreichbar, was in dieser Zeit zu einem Schaden von ca. 150 Millionen US-Dollar führte. Ursache war laut Amazon ein Tippfehler beim Entfernen von S3 Servern, der dazu führte, dass deutlich mehr Server heruntergefahren wurden, als geplant. Cloud Nutzer sollten die Risiken von offline Zeiten kalkulieren und Ausfallpläne bereithalten, um Verluste möglichst gering zu halten. [HERS17]

[BRAD16] [UTLE18]

Layer 8 Fehler

Der Nutzer wird häufig als achte Schicht über dem OSI-Netzwerk-Modell bezeichnet und genau wie in allen anderen Systemen können Fehler von Nutzern erhebliche Folgen haben. Viele Cloud Anbieter ermöglichen den Zugriff zur Cloud über beinahe jedes Endgerät wie PC, Laptops, Tablets und Smartphones. Der Verlust oder die Kompromittierung dieser Geräte kann Unbefugten den Zugang zu gesicherten Cloud Systemen ermöglichen und kann somit durchaus zu Sicherheitsproblemen führen.

Durch den großen Bekanntheitsgrad von Cloud Systemen sind diese seit einiger Zeit auch das Ziel von "*Phishing*" und "*Social Engineering Angriffen*". Ersteres kann beispielsweise in Form von Emails sein, die den Empfänger auf präparierte Webseiten locken, wo dann vertraulichen Informationen preisgeben werden sollen und dadurch in die Hände des Angreifers fallen. Bei Social Engineering Angriffen wird gezielt versucht Mitarbeiter von der Vertrauenswürdigkeit des Angreifers zu überzeugen, sodass der Angreifer sensible Informationen oder Zugänge erhält.

Ein aktueller Fall von fehlerhaftem Nutzerverhalten stellt der Fall der Marketing Firma Exactis dar. Wie WIRED am 27.06.2018 veröffentlichte war eine Datenbank der Firma Exactis mit rund 340 Mil. sensiblen Datensätzen für eine längere vollkommen öffentlich und ungeschützt online erreichbar. Diese Datensätze beinhalten zwar keine Informationen zu Kreditkarten oder Sozialversicherungsnummern, dafür allerdings sehr private Charakteristika der Betroffenen. Enthalten sind unter anderem Informationen zu Telefonnummer und Adresse, aber auch zu Interessen, Angewohnheiten und die Geschlechter der Kinder zu einer Person. [GREE18]

[UTLE18] [WOOD13]

Mangelhafte Backups

Ein weiterer Sicherheitsaspekt, der zu betrachten gilt, ist die Wichtigkeit von Backups. Viele Cloud Betreiber bieten einen automatisierten Service an, der von den laufenden Systemen Backups macht. Dafür muss allerdings sichergestellt werden, dass auch alle relevanten Daten adäquat synchronisiert werden und nicht verloren gehen. Gerade für den Fall, dass ein System von Ransomware befallen und verschlüsselt wird sind Backups der einzige Weg, um dem Angreifer nicht ausgeliefert zu sein. Das Massachusetts Institute of Technology (MIT) sieht Ransomware als einer der sechs größten Gefahren, denen Unternehmen 2018 gegenüber stehen. Durch Cloud Computing sind in den vergangenen Jahren enorme Mengen an Daten zentralisiert worden und stellen dadurch interessante Ziele für Angreifer dar. Vor allem kleinere Cloud Anbieter könnten Opfer von Angriffen werden, da ihre Ressourcen weitaus eingeschränkter sind. [GILE18] [UTLE18]

Systemfehler

Wird beim Entwickeln von Softwareanwendungen nicht ein gewisser Qualitätsstandard eingehalten, kann dies zu Sicherheitslücken führen. Da bilden Anwendungen in der Cloud beziehungsweise Anwendungen die für das Bereitsstellen in der Cloud verwendet werden, keine Ausnahme. Möchte ein Anbieter beispielsweise mehrere Kunden mit der selben Instanz versorgen, so setzt dieser auf ein mandantenfähiges System. Fehler innerhalb solchen Anwendungen können allerdings dafür sorgen, dass die Daten der Kunden nicht sauber von einander getrennt sind und unbefugte Zugriffe ermöglicht werden. [UTLE18]

Entscheiden sich Unternehmen ihre Cloud Systeme bei einem Dritten online zu stellen, machen diese sich auch automatisch abhängig. Bei der Wahl des Anbieters sollten sich Unternehmen Gedanken machen, was für Auswirkungen es hat, sollte ein System mal nicht erreichbar sein. Vor allem die großen Cloud Anbieter können sich allerdings damit rühmen eine Erreichbarkeit von über 99% vorweisen zu können. [CLOU18]

2014 erhielt ein Angreifer Zugang zu der AWS Management Konsole des Softwareunternehmens Code Space. Dieses Unternehmen bot Quellcode Repositories und Projekt Management Dienste an und setzte dabei fast vollständig auf Infrastrukturen und Anwendungen von AWS. Der Angreifer erpresste Code Space und löschte einen Großteil der Daten und Infrastruktur des Unternehmens, als dieses nicht kooperierte. Dabei gingen auch die Backups, die das Unternehmen für Notfälle erstellt hatte verloren, was das Unternehmen sehr schwer traf und in den Medien mit dem *"Mord"* an das Unternehmen verglichen wurde. [VENE14]

Privatsphäre

Die vorhergehend erläuterten Sicherheitsaspekte richten sich vor allem an Unternehmen, die ihre Cloud-Infrastruktur von einem Dritten beziehen. Aber auch auf Endnutzer und vor allem auf deren Privatsphäre, hat der Wandel zur Cloud Auswirkungen.

Nutzer teilen durch den steigenden Einsatz von Cloud-Anwendungen, viele private Informationen mit Dritten und wissen häufig gar nicht, wie diese Informationen verarbeitet werden und von wem sie eingesehen werden können. Dies wird vor allem in den Medien als *"gläsernde Bürger"* bezeichnet. Darüberhinaus können Cloud Anbieter, genau wie Provider von anderen IT-Services, gesammelte Daten weitergeben oder müssen dies teilweise sogar, wenn es von der Regierung eingefordert wird. Ein Auszug aus den AGBs von Dropbox zeigt, dass der Anbieter, ähnlich wie andere, sich vorbehält Daten mit Dritten auszutauschen:

> *"**Recht und Ordnung sowie öffentliches Interesse** – Wir können Ihre Daten auch Dritten offenlegen, wenn dies nach unserem Ermessen sinnvoll und notwendig scheint, um (a) geltenden Gesetzen, Vorschriften, rechtlichen Verfahren oder angemessenen Anfragen von Behörden Folge zu leisten, (b) einen Menschen vor dem Tod oder schwerer körperlicher Verletzung zu schützen, (c) Dropbox oder unsere Nutzer vor Betrug oder Missbrauch zu schützen, (d) die Rechte, das Eigentum, die Sicherheit oder die Interessen von Dropbox zu schützen oder (e) Aufgaben auszuführen, die im öffentlichen Interesse liegen."* [DROP18]

Solche Regelungen geben Cloud Anbietern viel Handlungsfreiraum, sodass in extremen Situationen auch private Informationen ohne Durchsuchungsbeschluss offen gelegt werden können. Auf der anderen Seite können Anbieter in den meisten Ländern nicht dazu gezwungen werden Informationen an staatliche Behörden weiter zu geben. So hat ein amerikanisches Gericht entschieden, dass Microsoft nicht dazu gezwungen werden kann, Daten eines Nutzer an den amerikanischen Staat weiterzugeben, wenn diese Daten auf einem Server im Ausland liegen. Microsoft hatte sich damals geweigert die Emails eines Nutzern frei zu geben, da sie auf einem irischen Server gespeichert waren. [EDWA17]

Die seit dem 25.05.2018 wirksame EU-Datenschutz-Grundverordnung (EU-DSGVO) hat großen Einfluss auf die zu leistenden Datenschutzmaßnahmen von Unternehmen. Diese Grundverordnung beinhaltet zum einen, dass Unternehmen der EU dazu verpflichtet werden, dass personenbezogene Daten künftig innerhalb der EU-Grenzen gespeichert und *"nach datenschutzrechtlichen Bestimmungen und Vorgaben"* verarbeitet werden müssen. Ob diese Vorgaben eingehalten werden, wird von Datenschutzexperten geprüft und Verstöße können dabei mit bis zu vier Prozent des Jahreseinkommens verantwortet werden. Außerdem sind Unternehmen auch dafür verantwortlich, dass die EU-DSGVO von Drittanbietern eingehalten wird, bei denen ein Unternehmen Daten speichert. Dabei sind vor allem Unternehmen, die Cloud-Dienste nutzen betroffen. Des Weiteren müssen Einzelpersonen über jede Datenerhebung ausdrücklich und ausführlich informiert werden, was die Transparenz zum Umgang mit persönlichen Daten verbessern soll. Die EU-DSGVO sieht auch vor, dass Daten auf Verlangen hin, möglichst unverzüglich, gelöscht oder korrigiert werden müssen. [EUDS18] [NETW18]

Inflexibilität und geringere Kontrolle

Neben den bereits erläuterten Sicherheitsaspekten, bei denen vor allem Abhängigkeit zu einem Dritten, Risiken in Sachen Sicherheitlücken und Privatsphäre herausstechen, gibt es in Sachen Cloud Computing weitere Nachteile und auch Begrenzungen.

Im Vergleich zu standardmäßigen on premise Anwendungen und selbstständig verwalteten IT-Infrastrukturen, schränken Clouddienste die Kontrolle und Flexibilität ein. Über die End-User Licence Agreement (EULA) können Anbieter enorme Einschränkungen machen, was Kunden mit den genutzten Diensten machen können.

Darüber hinaus werden Cloud Anwendungen in der Regel nur in einer einzigen Version zur Verfügung gestellt, um dem Kunden möglichst zeitnah Updates liefern zu können. Dies hat allerdings zur Folge, dass auf individuelle Probleme und Forderungen von Kunden kaum noch eingegangen werden kann. Bei kleineren mit mittelständigen on-premise Anwendungen ist es noch häufig der Fall, dass ein Unternehmen Wünschen von besonders treuen Kunden entgegen kommt und individuelle Lösungen entwickelt.

Ein weiterer Nachteil von Cloud Computing ist das sogenannte "*Vendor Lock-In*". Damit ist gemeint, dass der Wechsel von einem Cloud Anbieter zum nächsten häufig nicht problemlos möglich ist, was zu zusätzlichen Kosten führt. Cloud Architekturen, die auf die Infrastruktur eines speziellen Anbieters zugeschnitten sind, könnte beim Wechsel zu einem anderen Anbieter nur mit Kompromissen oder zusätzlichen Entwicklungsaufwand übernommen werden. Dies kann zu zusätzlichen Sicherheitsrisiken führen.

Zusammenfassend ist zu sagen, dass viele Unternehmen durch die Skalierbarkeit, Agilität und dem "*Pay per Use*" Prinzip von Cloud Computing profitieren können. Jedoch sollte für jeden Anwendungsfalls das geeignete Servicemodell gewählt, die Risiken stetig abgewägt und ausreichend Sicherheitsmaßnahmen getroffen werden.

[LARK18] [WARD18]

Amazon Web Services (AWS)

Mit "*Amazon Web Services*" (AWS) hat Amazon im März 2006 eine Plattform für die Öffentlichkeit gestartet, mit der Cloud Computing Dienste flexibel und nach Bedarf von Einzelpersonen, Unternehmen und Regierungen bezogen werden können. Mit 34% Marktanteil konnte Amazon 2017 allein durch AWS einen Umsatz von 17,4 Milliarden US-Dollar verbuchen. [RAMA17]

Die Dienste können zentral über die "*AWS Management Console*" erreicht und konfiguriert werden. Dabei reicht ein Internetbrowser als Client aus oder sogar eine AWS spezifische Smartphone App, die allerdings nur lesenden Zugriff ermöglicht. Über die Management Konsole kann auf alle der über 90 Dienste zugegriffen werden, die in die folgenden Bereiche aufgeteilt sind [AWS18a]:

- Datenverarbeitung
- Speicherung
- Datenbank
- Migration
- Netzwerk und Bereitstellung von Inhalten
- Developer-Tools
- Verwaltungs-Tools
- Medienservices
- Maschinelles Lernen
- Analysen
- Sicherheit, Identität und Compliance
- Services für Mobilgeräte
- AR und VR
- Anwendungsintegration
- Customer Engagement
- Unternehmensproduktivität
- Desktop- und App-Streaming
- Internet of Things
- Entwicklung von Spielen
- Software
- AWS-Kostenmanagement

Wie auch bei anderen Cloud Computing Anbietern, ergeben sich die Kosten für den Nutzer durch eine Kombination von Verbrauch, den Spezifikationen zu Hardware/Betriebssystem/Software/Netwerk, der benötigten Erreichbarkeit, Redundanz, Sicherheit und dem gewünschten Kundenservice. Dabei setzt auch Amazon auf das "*Pay per Use*" Prinzip und rechnet Kosten monatlich ab. Viele Dienste wie EC2, S3, AWS Lambda und Amazon RDS können auf einer kostenlosen Basis, mit eingeschränktem Kontingent, genutzt werden. Darüberhinaus können Institutionen, Lehrbeauftragte und Studenten sich über "*AWS Educate*" zusätzlich einem Netzwerk anschließen, welches bei der Ausbildung der "*nächsten Generation von IT und Cloud Experten*" beitragen soll und die nötigen Ressourcen zum cloudbezogenen Lernen bereitstellen. AWS Educate ermöglicht es außerdem Studenten sich ein Kontingent von 40 US-Dollar zu sichern, das frei für die AWS Dienste verwendet werden kann. [AWS18b] [AWS18c] [AWS18d]

In den nachfolgenden Abschnitten sollen zum einen einige Grundlagen und allgemeine Hinweise zur Verwendung von AWS erläutert und zum anderen die bekanntesten Dienste vorgestellt werden.

Grundlagen

Durch einen erfolgreichen Login zur AWS Management Console gelangt man zu einem Dashboard, wie es die Abbildung 3 zeigt. In rot wurden die wichtigsten Menüoptionen markiert und kurz beschrieben.

Der Menüpunkt **Services** öffnet die Gesamtübersicht zu allen Diensten die AWS anbietet und erlaubt eine weitere Navigation zu den AWS-Diensten.

Über einen Klick auf den eigenen **Nutzernamen** (in diesem Fall "User") sind alle Einstellungen zu erreichen, die mit dem eigenen Account zusammenhängen. Dazu gehören persönliche Informationen, die Organisationen der man unter Umständen angehört, die aktuelle Kostenübersicht und die Einsicht zu sicherheitsrelevanten Informationen wie Passwort, Multi-Faktor-Authentifizierung und privaten Schlüsseln.

Der letzte Menüpunkt erlaubt es die Region zu wechseln in der Aktionen ausgeführt werden sollen. Die Abbildung 3 zeigt "Ohio" an, es können aber verschiedene Regionen aus den östlichen USA, den westlichen USA, Asien/Pazifik, der EU und auch Südamerika gewählt werden. Dieser Menüpunkt ist von besonderer Wichtigkeit, wenn es darum geht sicherzustellen, dass Dienste unter anderem datenschutzrechtlich richtig gestartet werden. Außerdem ist anzumerken, dass zum aktuellen Zeitpunkt (Juni 2018) die Verwendung des kostenlosen Kontingents auf die vier Regionen in den USA beschränkt sind.

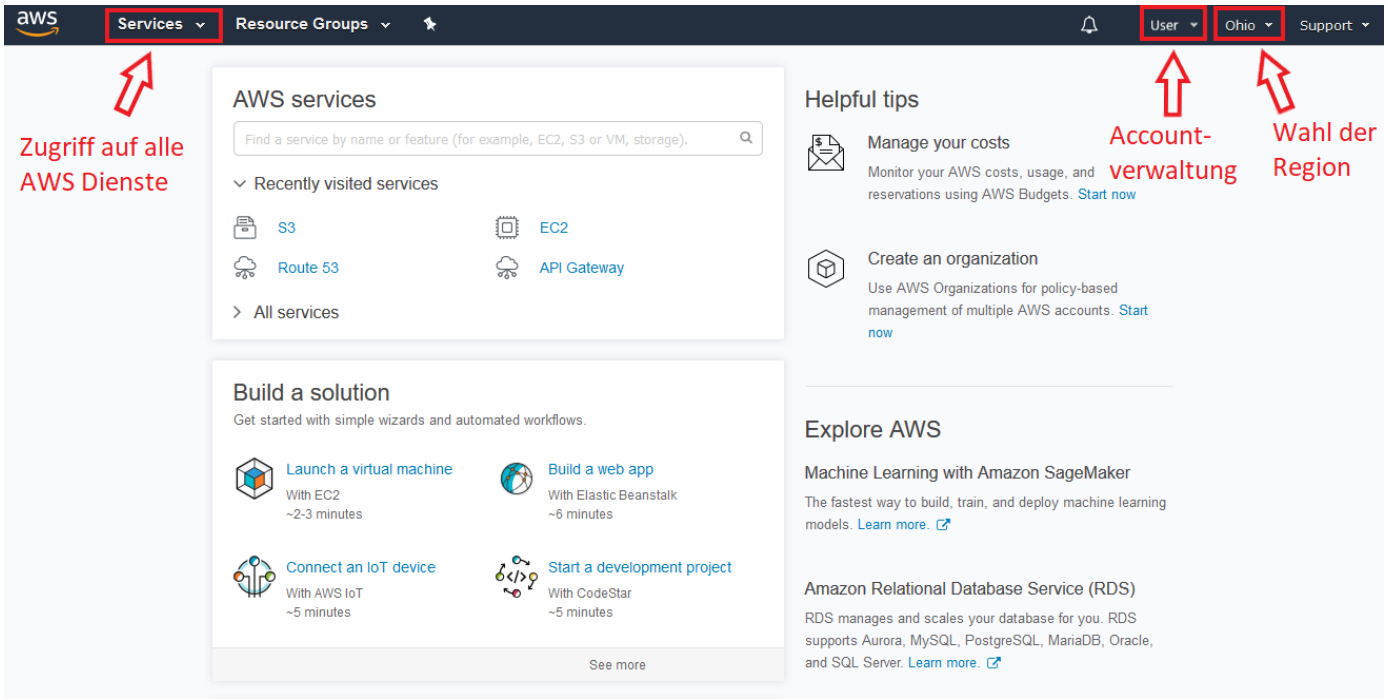

Abb. 3: AWS Management Console - Quelle (verändert): [AWS18e]

Die Erstellung eines AWS Accounts ist prinzipiell vollkommen kostenlos. Für eine standardmäßige Registrierung ist allerdings die Angabe einer Kreditkarte notwendig. Studenten können dies über AWS Educate umgehen.

In den USA hat Amazon zwei sogenannte "*Pop-up Lofts*" eröffnet, in denen AWS Kunden, Start-Up Unternehmen oder eigenständige Entwickler komplett kostenfrei professionelle Hilfestellung zu den AWS-Infrastrukturen erhalten können. Dabei können vor Ort 60-minütige Sitzungen angenommen werden, im denen speziell geschultes Personal alle Fragen in Bezug auf AWS beantwortet und individuelle Hinweise gibt. Außerdem finden in diesen Pop-up Lofts in regelmäßigen Abständen Events statt, die in drei Kategorien unterteilt sind, um Anfänger, Fortgeschrittene und Experten zusätzliche Schulungsmöglichkeiten anzubieten. [AWS18f] [AWS18g]

Identity and Access Management (IAM)

Unternehmen und Organistationen, die AWS-Dienste nutzen, können durch die Verwendung von Amazons Accountsverwaltung "*Identity and Access Management*" (IAM) ein individuelles Rechtesystem auf die AWS-Infrastruktur abbilden. Dabei bietet IAM die Möglichkeit verschiedene Benutzer, Benutzergruppen und Rollen zu definieren und diesen sehr detailliert den Zugriff auf Ressourcen erlauben oder verbieten. Diese Benutzerverwaltung wird von Amazon kostenfrei angeboten und soll unter anderem mittels Best Practices (externer Link) eine sichere Verwendung der AWS-Dienste innerhalb von Unternehmen und Organistationen bieten.

Die Granularität in denen Rechte vergeben werden können reicht dabei von einem vollen Zugriff auf die AWS Management Console, über den Zugriff auf vereinzelte Dienste und Einstellungsmöglichkeiten bis zur Beschränkung darauf, dass nur einzelne Funktionen eines Dienstes aufgerufen werden können. Hinzu kommt, dass Zugriffe auf bestimmte Uhrzeiten und Zeiträume beschränkt werden können und Nutzern auch die Verwendung von Multi-Faktor-Authentifizierung vorgeschrieben werden kann. Abbildung 4 zeigt beispielhaft ein IAM Dashboard mit mehrere Benutzer und verschiedenen Gruppen.

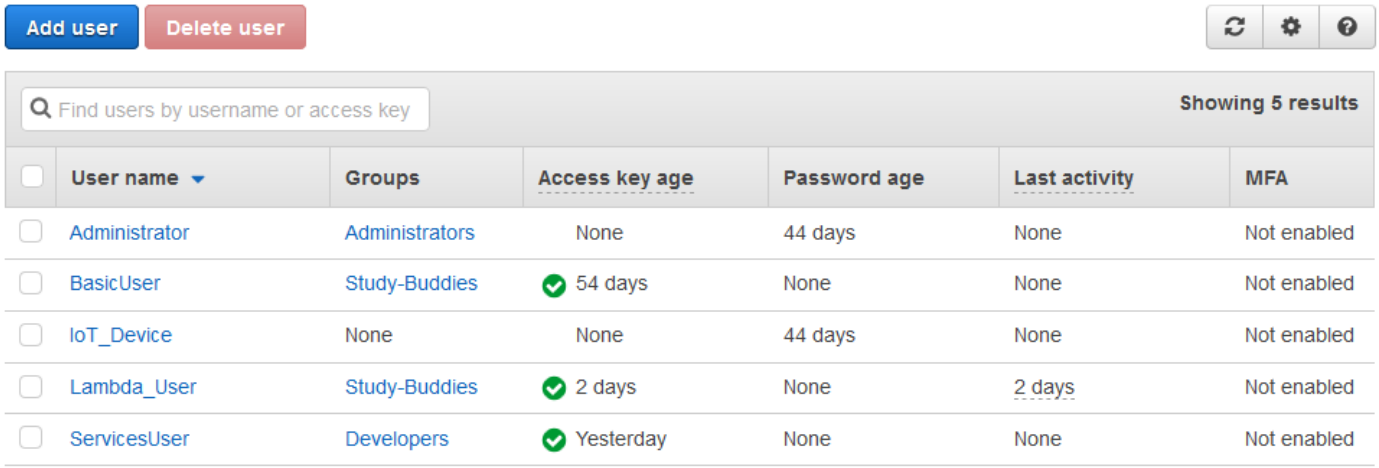

Abb. 4: AWS IAM Dashboard - Quelle: [AWS18h]

Bei der Erstellung von neuen IAM-Benutzern können unter anderem Passwörter und Zugangsschlüssel vordefiniert werden, aber auch temporäre Anmeldedaten erzeugt werden. Temporäre Zugangsdaten können auch IAM-Benutzer und AWS-Dienste erhalten, die normalerweise keinen Zugriff auf AWS-Ressourcen einer Organisation haben. Hierbei kommen die IAM-Rollen ins Spiel die einer AWS-Ressource zugewiesen werden müssen. IAM-Benutzer und AWS-Dienste können diese definierten Rollen annehmen und anschließend temporäre Anmeldeinformationen erhalten, mit denen Aufrufe von AWS-APIs erfolgen können. Dabei ist zu beachten, dass ein AWS-Dienst nur eine IAM-Rolle zugewiesen bekommen kann, aber mehrere andere Dienste und Anwendungen diese Rolle einnehmen können. AWS-Ressourcen können über sogenannte "*Amazon Resource Names*" (ARNs) eindeutig über alle AWS-Kunden hinaus identifiziert werden. Diese ARNs werden unter anderem dafür verwendet, dass in den IAM Einstellungen Dienste hinterlegt werden können, die eine IAM-Rolle einnehmen dürfen. ARNs können beispielsweise das folgende Format besitzen:

```
<!-- Elastic Beanstalk application version -->
arn:aws:elasticbeanstalk:us-east-1:123456789012:environment/My App/MyEnvironment

<!-- IAM user name -->
arn:aws:iam::123456789012:user/David

<!-- Amazon RDS instance used for tagging -->
arn:aws:rds:eu-west-1:123456789012:db:mysql-db

<!-- Object in an Amazon S3 bucket -->
arn:aws:s3:::my_corporate_bucket/exampleobject.png
```

[AWS18i] [AWS18j] [AWS18n]

Elastic Compute Cloud (EC2)

Der "*Elastic Compute Cloud*" (EC2) Dienst, stellt einen der ältesten und beliebtesten Dienste dar, den AWS anbietet und bildet das Servicemodell PaaS dar. Über diesen Dienst können virtuelle Rechenkapazitäten auf einfache Weise erstellt, gestartet, gestoppt und verwaltet werden. Hierbei kann auf fertige Instanz-Templates, sogenannte Amazon Machine Images (AMIs), zurückgegriffen werden. Bezahlt werden müssen nur die Instanzen, die tatsächlich auch laufen. Das Erstellen einer EC2 Instanz ist in sieben Schritte aufgeteilt, die nachfolgend detailliert vorgestellt werden:

1. Amazon Machine Image (AMI) wählen

Offiziell werden von Amazon 35 verschiedene AMIs unterstützt, die auf Linux oder Windows basieren. Diese unterscheiden sich, neben dem Betriebssystem, vor allem in der Software, die schon vorinstalliert ist. Außerdem können auch AMIs gewählt werden, die von der AWS-Community bereitgestellt werden. Hier stehen über 30.000 verschiedene Templates zur Verfügung, mit denen eine EC2 Instanz gestartet werden kann.

2. Instanz Typ wählen

Der Instanz Typ gibt an, mit welcher Hardware eine Instanz ausgestattet ist. Diese sind in die folgenden fünf "Familien" unterteilt:

- *General Purpose*
  Instanzen dieser Familie haben eine gute Balance zwischen CPUs, RAM, Speicher und Netzwerkleitung und sind für viele Anwendungen eine gute Wahl, die wenig bis moderaten Speicher und Rechenleistung benötigen.

- *Compute optimized*
  Namensgeben sind diese Instanzen darauf ausgelegt Aufgaben mit hohen Anforderungen an CPU und RAM haben. Im Durchschnitt stehen pro CPU 2 GB RAM zur Verfügung, was hochskaliert bis zu maximal 72 CPUs und 144 GB RAM. Nicht außer Acht zu lassen ist die Netzwerkanbindung dieser Instanzen, die bis zu 25 Gigabit betragen kann.

- *GPU graphics*
  Mit den GPU graphics Instanzen bietet AWS virtuelle Rechner an, die für Aufgaben im Grafikbereich gedacht sind. Diese sind mit starken GPUs, hohem RAM und starker Netzwerkverbindung ausgestattet. Die GPU graphics Instanzen können mit bis zu 64 CPUs, 976 GB RAM und 8 NVIDIA K80-Hochleistungs-GPUs ausgestattet werden. Letzte besitzen pro Einheit 2.496 parallele Verarbeitungskerne und 12 GB GPU-Speicher.

- *Memory optimized*
  Die Memory fokussierten EC2 Instanzen richten sich an Kunden, die große Datenbanksysteme oder Cachingmechanismen bereitstellen wollen. Instanzen dieser Familie zeichnen sich durch sehr hohen RAM aus, der bis zu 1952 GB betragen kann.

- *Storage optimized*
  Für Anwendungen, die bestimmte Anforderungen an I/O Operationen und Speicherkapazitäten haben, sind die Speicher optimierten Instanzen von AWS geeignet. Je nach Instanz Typ können bis zu 24 HDDs zu je 2048 GB Speicher oder 8 SSDs zu je 1900 GB Speicher ausgewählt werden.

3. Instanz detailliert konfigurieren

Die nachfolgende Abbildung 5 gibt einen Überblick über den dritten Schritt der Erzeugung einer EC2 Instanz. Hier kann zum einen die Anzahl der Instanzen festgelegt werden, die mit den vorherigen Konfigurationen gestartet werden sollen. Zum anderen können Einstellungen zur Netzwerkanbindung, Verhalten beim Stoppen beziehungsweise Herunterfahren der Instanz gemacht werden und auch eine IAM-Rolle festgelegt werden, mit der andere Anwendungen automatisiert Anmeldedaten erhalten können (siehe Identity and Access Management (IAM)).

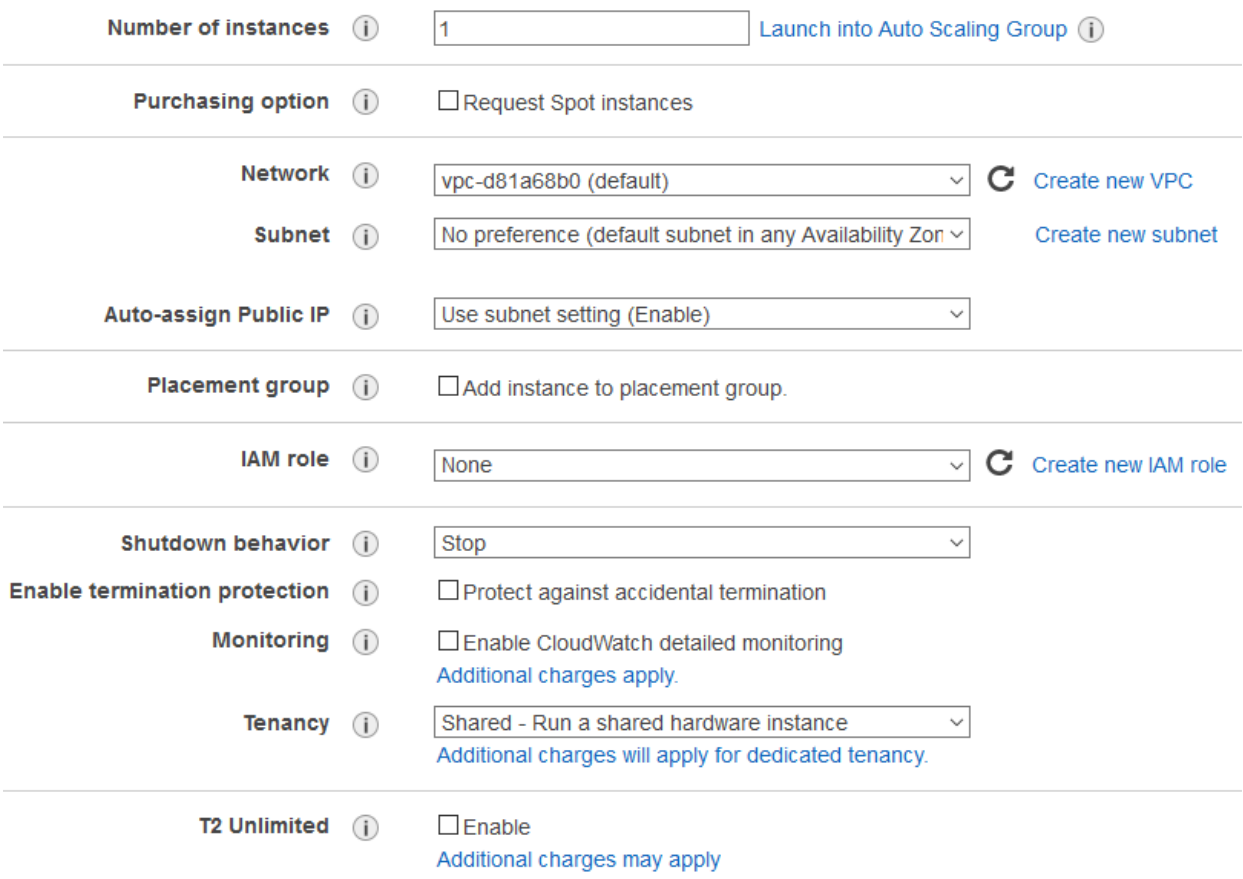

Abb. 5: Amazon EC2 Configure Instance Details - Quelle: [AWS18k]

Besonders hervorzuheben ist der letzte Menüpunkt "*T2 Unlimited*". Diese Option ist nur für die T2 Instanzen der General Purpose Familie verfügbar und ermöglicht eine Skalierbarkeit der verfügbaren CPU-Leistung, sollten Anwendungen mehr Leistung benötigen als von der Instanz vorgesehen.

4. Speicher hinzufügen

Die Speichereinstellungen zu einer EC2 Instanz ermöglichen es die größen und Performanztypen der gewünschten Speicher festzulegen. Darüberhinaus kann festgelegt werden, ob Daten verschlüsselt gespeichert werden sollen und ob Speichermedien über die Lebenszeit einer EC2 Instanz hinaus existieren sollen. Außerdem können Speicherzustände von existierenden Snapshots wiederhergestellt werden, die in S3 Buckets gelagert sind.

5. Tags hinzufügen

Um bei einer größeren Anzahl von EC2 Instanzen nicht den Überblick zu verlieren, können diese mit Tags versehen werden. Diese werden durch einfache Key-Value-Paare dargestellt und können somit beispielweise Auskünfte zu Besitzer, Verwendungszweck und zugehörigen Projekten direkt ersichtlich machen.

6. Sicherheitsgruppen konfigurieren

Die Sicherheitsgruppen beschreiben die Firewall Einstellungen, die zu einer EC2 Instanz gehören. Hier können fertige Verbindungstypen hinzugefügt werden, um gängige Verbindungen wie Secure Shell (SSH), HTTP und Internet Message Access Protocol (IMAP) zu erlauben. Es können aber auch gezielt Konfigurationen zu bestimmten Protokollen und Ports gemacht werden, sowie IP-Adressen beziehungsweise IP-Adressräume festgelegt werden die auf die Instanz zugreifen dürfen.

7. Übersicht und Starten der Instanz

Im letzten Schritt können alle vorher getätigten Einstellungen nochmal in übersichtlicher Form geprüft werden, bevor die Instanz gestartet wird. Tendenziell sind nur Schritt 1 und 2 notwendig, um zu Schritt 7 zu gelangen, da die anderen Schritte standardmäßig vorkonfiguriert sind. Es ist allerdings in der Regel sinnvoll eine Instanz für einen spezifischen Anwendungsfall zu individualisieren.

[AWS18k] [AWS18l]

Simple Storage Service (S3)

Amazon "*Simple Storage Service*" (S3) ist ein Dienst, um beliebige Mengen von Daten zu speichern und abzurufen. Dabei können alle Daten im nativen Format gesichert und direkt im S3, mit leistungsstarken Analysen, verarbeitet werden. AWS selbst sagt über S3, dass "*selbst die strengsten rechtlichen Anforderungen*" erfüllt werden und er "*von Grund auf für eine Beständigkeit von 99,999999999% entwickelt*" wurde.

Durch Dienste wie Amazon Athena oder Amazon Redshift Spectrum können komplexe Abfragen auf unterschiedlich großen und unstrukturierten Datenmengen ausgeführt werden. Ersteres ermöglicht Abfragen mit Standard-SQL, ohne, dass eine zusätzliche Infrastruktur verwaltet werden muss. Letzteres ist gezielt für Analysen von riesigen, unstrukturierten Datenmengen optimiert, wie sie beispielsweise im Data-Warehouse Bereich auftreten. Selbst Abfragen auf Datensätze im Exabyte-Bereich, also über 1000 Petabyte, liefern laut AWS schnelle Ergebnis und sind möglich, ohne dass Daten extrahiert werden müssen. Die Geschwindigkeit wird vor allem durch Abfrageoptimierung erreicht, mit der die Abfragen auf Tausenden Knoten parallel ausgeführt werden.

In vielen Preiskategorieren von S3 und auch beim Dienst Amazon Glacier, werden gespeicherte Daten automatisch auf mindestens drei physischen Einrichtungen repliziert. AWS unterscheidet dabei zwischen "*Regionen*" und "*Availability Zones*" (Verfügbarkeitszonen, AZs). Die Abbildung 6 zeigt dabei die schon existierenden Regionen mit gelben Kreisen und die geplanten Regionen mit türkisen Kreisen. Die Zahl zu einer Region gibt an, wieviele AZs in dieser Regionen zur Verfügung stehen. Zonen sind physisch von anderen isoliert (mindestens 10km Entfernung) und über leistungsstarke Netzwerke miteineinander verbunden. Insgesamt bietet die AWS-Cloud derzeit 55 AZs an, die auf 18 Regionen aufgeteilt sind und plant vier weitere Regionen mit zusammen 12 AZs.

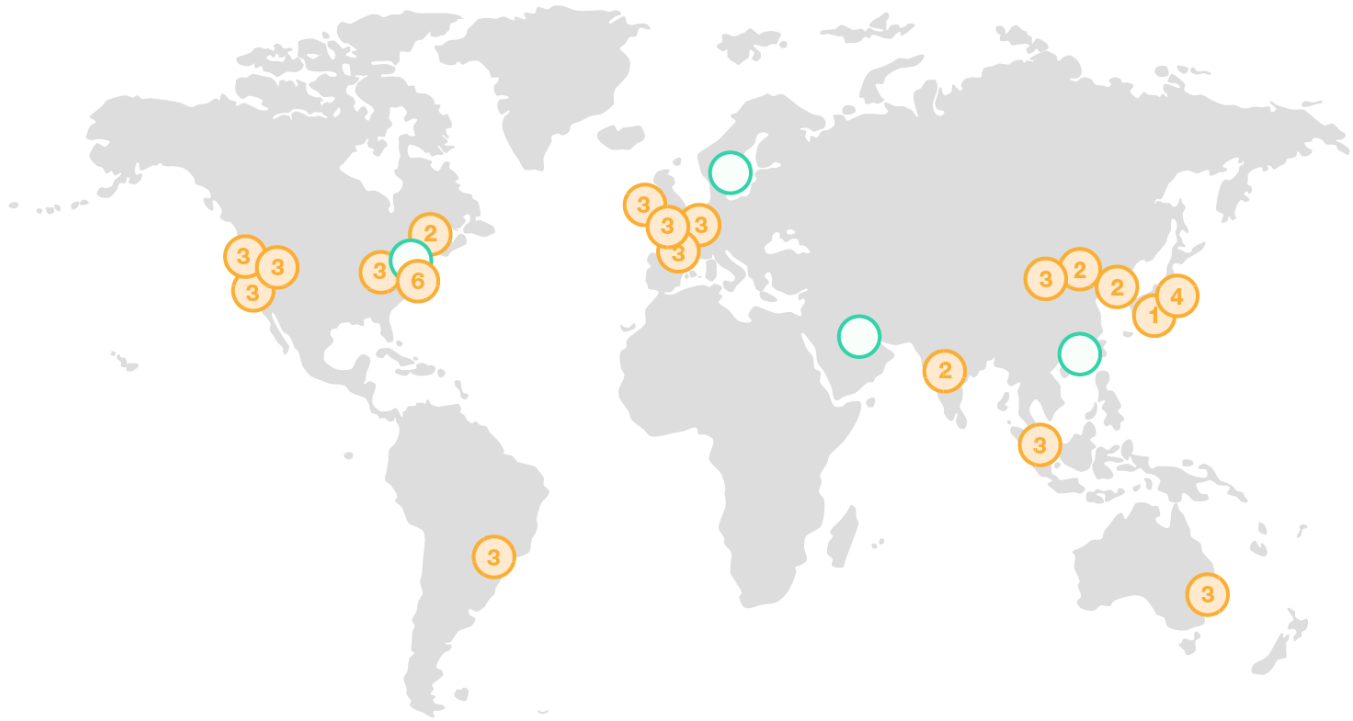

Abb. 6: Globale AWS-Infrastruktur - Quelle: [AWS18m]

Bei der Nutzung von S3, werden sogenannte "*Buckets*" erzeugt, die als Datentöpfe zu sehen sind. Das heißt es müssen keine Datenbanken händisch angelegt, konfiguriert und verwaltet werden. S3 kann ähnlich wie Google Drive über den Browser genutzt werden, um Daten hochzuladen, zu suchen und zu verwalten. Es können Ordner und Tags angelegt werden, um die Daten zu sortieren und klassifizieren. Darüberhinaus können Buckets und auch einzelende Ordner/Daten sowohl öffentlich als auch privat sein oder für spezielle AWS Accounts freigegeben werden. Die ARN eines Bucket oder auch die von einzelnen Ressourcen innerhalb eines Buckets ermöglichen es außerdem, dass Anwendungen und andere AWS-Dienste diese gezielt verwenden können.

Amazon S3 kann einerseits in die meisten AWS-Dienste integriert werden, aber auch durch eine Großzahl von Systemintegratoren und unabhängigen Softwarelieferanten mit weiteren Softwareanwendungen zusammengeführt werden. Diese helfen dabei *"gängige Sicherungs-, Wiederherstellungs- und Archivierungslösungen, die besten Big Data-Lösungen und innovative Lösungen zur Notfallwiederherstellung"* umzusetzen.

Gartner, Inc. äußert sich wie folgt zu Amazon S3:

> *"Amazon S3 ist gemessen an den verwalteten Daten der größte Objektspeicherservice, der auf einer öffentlichen Cloud basiert. AWS weiß besser als jeder andere Anbieter, wie Kunden groß angelegte Speicherservices in der öffentlichen Cloud einsetzen."*

[AWS18m] [AWS18o] [AWS18p]

AWS Lambda

Das bereits erläuterte FaaS Modell wird von Amazon mittels AWS Lambda angeboten. Dabei wird vollständig auf das Erzeugen von Infrastrukturen, wie Server, verzichtet und es kann sich gezielt auf die eigentliche Entwicklung von Funktionen konzentriert werden. Bezahlt werden muss nur für die Rechenzeit, die eine Funktion benötigt, wenn sie ausgeführt wird und auch nur dann, wenn sie tatsächlich ausgeführt wird. Neben dem Wegfallen von Verwaltungsaufgaben, bezogen auf Server, bietet AWS Lambda eine automatische Skalierung der hinterlegten Funktionen, indem jede Anfrage separat und parallel ausgeführt wird. Für die Entwicklung von Funktionen werden die folgenden Programmiersprachen beziehungsweise Lautzeitumgebungen unterstützt:

- C# (.NET Core 1.0 und 2.0)
- Go 1.x
- Java 8
- Node.js (4.3, 6.10 und 8.10)
- Python (2.7 und 3.6)

Während die ersten drei lokal entwickelt werden müssen und anschließend in einer AWS Lambda Funktion hinterlegt werden können, steht für Node.js und Python ein Online-Editor bereit. Dieser Online-Editor ermöglicht die Bearbeitung von mehreren Dateien innerhalb eines Projektes mit Syntaxhighlighting und automatischer Codevervollständigung. Eine einfache AWS Lambda Funktion, hier in JavaScript geschrieben, sieht wie folgt aus:

```
exports.handler = (event, context, callback) => {
    // Succeed with the string "Hello world!"
    callback(null, 'Hello world!');
};
```

Bei der Erstellung von AWS Lambda Funktionen kann einerseits mit dem oben gezeigten Beispiel begonnen werden oder andererseits auch auf sogenannte Blueprints zurückgegriffen werden. Diese stellen von anderen Entwicklern veröffentliche Funktionen dar, die mit unterschiedlichen Lizenzen versehen sind. Darüberhinaus können auch vollständige Repositories genutzt werden, die beispielsweise fertige Erweiterungen von Amazon Alexa darstellen. Über das Webinterface können des Weiteren auch Test-Events hinterlegt werden, um die entwickelten Funktionalitäten zu überprüfen.

Damit Funktionen ausgeführt werden können, muss mindestens ein sogenannter "*Trigger*" definiert werden. Der Aufruf einer bestimmten REST-Schnittstelle stellt ein klassischen Beispiel dar, mit denen Serverless Anwendungen umgesetzt werden. Das bedeutet, dass Anwendungen REST-Anfragen schicken, hinter denen kein großes, umfangreiches Backend steht, sondern einzelne Funktionen. Die REST-API dient in solchen Fällen einzig der Orchestrierung verschiedener Funktionen.

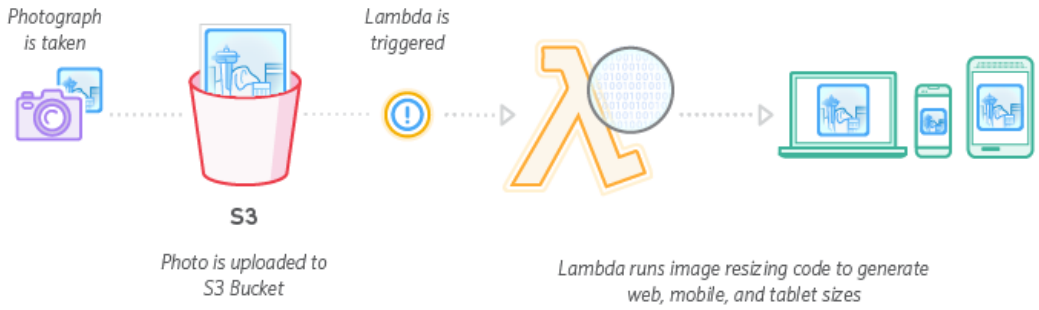

Abb. 7: AWS Lambda Beispiel: Image Thumbnail Creation - Quelle: [AWS18q]

Neben direkten Aufrufen, kann AWS Lambda auch auf speziell hinterlegte Events reagieren. Zu diesen gehören beispielsweise Nachrichten an ein Mobile/IoT Backend, Änderungen in einem Repository von AWS CodeCommit und auch Dateiablagen in einem S3 Bucket. Funktionen von AWS Lambda können außerdem Zugriff auf weitere APIs und Ressourcen haben, die mittels einer passenden IAM-Rolle zugewiesen werden.

Die Abbildung 7 zeigt das beispielhafte Zusammespiel von Amazon S3 mit AWS Lambda. Dieses Szenario wird von "*The Seattle Times*" genutzt, um Bildgrößen automatisch an unterschiedliche Geräte wie Desktop-PCs, Tablets und Smartphones anzupassen. In diesem Beispiel reagiert AWS Lambda auf das Hinzufügen von neuen Fotos in einen S3 Bucket und führt daraufhin Code aus, der zu dem neuen Bild die passenden Größen generiert.

[AWS18q] [AWS18r]

Cloud Design Patterns

Wie in allen Bereichen der Softwareentwicklung, gibt es auch für Cloud Computing Leitfäden für die Verwendung und Implementierung von Anwendungen, genannt "*Design Patterns*". Diese Patterns sind in der Regel so gestaltet, dass sie die Lösung zu einem konkreten Problem darstellen oder aber als generelle Hinweise zu sehen sind, die die Entwicklung und den Betrieb von Softwareanwendungen vereinfachen sollen.

Das Buch "*Cloud Design Patterns*" der Microsoft patterns & practive group ([HOME14]) beinhaltet insgesamt 24 Design Patterns, die jeweils ein oder mehrere Problemfelder im Bereich Cloud Computing beschreiben und Lösungen erläutern. Diese Problemfelder sind:

- **Erreichbarkeit**
  Indikator für den Zeitraum, in dem ein System oder eine Anwendung erreichbar ist und funktioniert. Wird in der Regel in Prozent der Uptime eines System angegeben. Systemfehler, Angriffe und Auslastungen sind unter anderem Faktoren, die die Erreichbarkeit beeinflussen.

- **Daten Management**
  Stellt einen Kernfaktor im Cloud Bereich dar und beeinflusst viele Qualitätsmerkmale. Typischerweise werden Daten dezentral gelagert, um eine gute Performanz, Skalierbarkeit und Erreichbarkeit zu erreichen. Dies bringt allerdings Herausforderungen mit sich, wie beispielsweise Datenkonsistenz und Synchronisierung.

- **Design und Implementierung**
  Beim Designen und Implementieren von Cloudanwendungen müssen eine Vielzahl von Entscheidungen getroffen werden, die unter anderem Wartbarkeit, Wiederverwendbarkeit und Bereitstellung betreffen. Diese Entscheidungen haben enorme Auswirkungen auf die Gesamtkosten und Qualität eines Produktes

- **Benachrichtigungen**
  Durch die Dezentralisierung der Cloud wird eine Infrastruktur benötigt, die Benachrichtigungen und Kommunikation zwischen einzelnen Komponenten, Diensten und Anwendungen erlaubt. Dies wird häufig über asynchrone Nachrichten abgewickelt, was sehr vorteilhaft ist, aber auch Aufgaben zu Themen wie Idempotenz mit sich bringt.

- **Management und Überwachung**
  Da Cloudanwendungen in der Regel in einem externen Rechenzentrum laufen und/oder auf Basis von IaaS/PaaS von Drittanbieter, können Probleme zum Management und zur Überwachung der bezogenen Ressourcen enstehen. Es müssen Informationen bereitgestellt werden, anhand derer der Status von laufenden Systemen überwacht werden kann.

- **Performanz und Skalierbarkeit**
  Performanz stellt einen Indikator für die Reaktionsfähigkeit eines Systems dar, die für die Ausführung einer Aufgabe erreicht wird. Skalierbarkeit hingegen ist der Indikator dafür, wieviel Last ein System verträgt, ohne, dass die Performanz darunter leidet. Vor allem im Cloud Computing Bereich müssen Systeme automatisch hoch- und herunterskalieren können, um unvorhergesehenen Lastspitzen entgegen zu wirken, aber auch, um bei geringer Last Kosten zu sparen.

- **Stabilität**
  Die Stabilität eines Systems beschreibt die Fähigkeit, Ausfälle zu erkennen und diese abzufangen. Die Vernetzung von Cloudanwendungen bringt viele Abhängigkeiten mit sich, sodass Systemausfälle von zentralen Diensten zu ernsthaften Problemen und Inkonsistenzen führen kann. Stabile Systeme erkennen Ausfälle frühzeitig und wirken diesen schnell und effektiv entgegen.

- **Sicherheit**
  Die öffentliche Zugänglichkeit von Cloudanwendungen, über das Internet, bringt eine Vielzahl von Gefahren mit sich, die durch geeigneten Sicherheitmaßnahmen abgewendet werden müssen. Diese Maßnahmen müssen schadhafte Angriffe unterbinden, Ungefugten den Zugriff verweigern und sensible Daten schützen.

In den nachfolgenden Abschnitten sollen vier der von Microsoft veröffentlichen Design Pattern vorgestellt werden und dabei auch externe Quellen und Anregungen mit einfließen.

[HOME14]

Cache-aside

Anwendungen nutzen Cachingmechanismen, um wiederholte Anfragen auf einen Datenspeicher abzufangen und dadurch die Performanz zu steigern. Allerdings muss sichergestellt werden, dass gecachte Daten immer möglichst aktuell sind und veraltete Daten nicht zu Inkonsistenzen führen.

Das Cache-aside Pattern stellt eine Lösung zu diesem Problem dar und befasst sich deshalb mit den Problemfelder "Daten Management" und "Performanz und Skalierbarkeit". Durch die Implementierung eines *"write through"* Caches werden I/O Operationen direkt auf dem Cache ausgeführt. Die Abbildung 8 verbildlicht das Cache-aside Pattern, welches auf drei Schritten basiert. Als erstes wird versucht, die gewünschten Daten aus dem Cache zu lesen beziehungsweise dort zu verändern (**1**). Sind die Daten dort nicht hinterlegt, werden sie aus dem Datenspeicher geholt (**2**) und gleichzeitig im Cache hinterlegt (**3**).

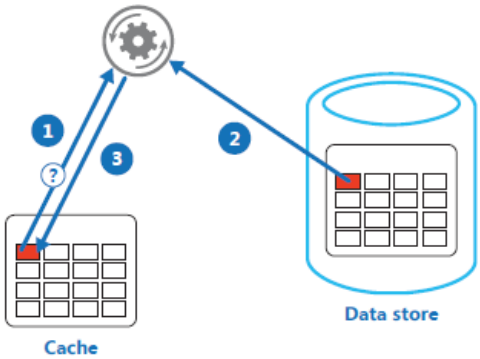

Abb. 8: Das Cache-aside Pattern - Quelle: [HOME14]

Bei der Implementierung müssen passende Entscheidungen dazu getroffen werden, wie lange Daten im Cache verweilen können. Dieses Zeitintervall sollte auf den jeweiligen Anwendungsfall zugeschnitten sein, sodass der Datenspeicher entlastet wird, aber auch gleichzeitig Daten

im Cache nicht veralten. Caching ist am effektivsten für relativ statische Daten oder häufige Leseoperationen.

Darüberhinaus müssen Regeln festgelegt werden, wann Daten aus dem Cache entfernt werden, sollte dieser an seine Speicherkapazität gelangen. Häufig werden hier die Daten entfernt deren letzer Zugriff am weitesten her ist. Wichtig ist außerdem geeignete Aktionen einzuleiten, sollten externe Prozesse den Datenspeicher verändern können, um Inkonsistenzen zu vermeiden.

[EVAN14] [HOME14]

Compensating Transaction

Die Cloud vernetzt häufig eine ganze Reihe von dezentralen Anwendungen und Dienste miteinander, die vor allem Daten verändern. Dabei werden in der Regel mehrere Operationen parallel ausgeführt und miteinander gekoppelt. Transaktions-Mechanismen werden dazu genutzt, um die Konsistenz in Systemen zu erhalten, indem bereits ausgeführte Aktionen rückgängig gemacht werden, sollte eine Operation innerhalb eines Transaktionsfensters fehlschlagen. Die Besonderheiten von Cloud-Infrastrukturen sorgen allerdings dafür, dass sehr strenge Transaktionen zu Verlusten in Performanz führen können. Müssen Dienste auf eine Rückmeldung warten, dass eine Transaktion mit X Operationen bei Y anderen Diensten erfolgreich durchgeführt werden konnte, gehen einige der Cloud-Vorteile verloren. Darüberhinaus können Operationen häufig nicht einfach rückgängig gemacht werden, da die Informationen schon von weiteren Anwendungen verwendet oder wieder verändert worden sein. Außerdem könnte sich der Zustand von Diensten in einer Service orientierten Architektur (SOA) durch Teiltransaktionen bereits geändert haben.

Das Compensating Transaction Pattern setzt auf dem "*Eventual Consistency Modell*" auf und stärkt die Stabilität eines Systems. Die Transaktion in diesem Pattern überschreiben nicht einfach den aktuellen Status eines Dienstes oder eines Datenspeichers mit den Informationen wie sie vorher waren, sondern bilden einen intelligenten Prozess, der alle Operationen der betroffenen Instanzen mit einbezieht.

Ein weitverbreitetes Verfahren zur Implementierung von eventuell konsistenten Operationen, basiert auf der Verwendung von sogenannten "*Workflows*". In diesen Workflows werden Informationen gespeichert, die abbilden, was getan werden muss, um die Ausführung einer Operation wieder rückgängig zu machen. Im Endeffekt sind Workflows die Transaktionen um tieferliegende Transaktionen herum. Somit ist es auch möglich, dass die Workflow Transaktionen fehlschlagen. In solchen Fällen muss das System dazu in der Lage sein von den fehlschlagenden Operationen erneut zu starten und diese wiederholt zu versuchen. In extremen Fällen, in denen keine automatische Wiederherstellung eines konsistenten Zustanden möglich ist, muss das System Alarm schlagen und möglichst viele Informationen zu dem Vorfall liefern.

[HOME14] [ROBI13]

Federated Identity

Typischerweise müssen Mitarbeiter eines Unternehmens mit mehr als nur einer Anwendung arbeiten, die von verschiedenen Anbietern bezogen werden. Es ist keine Seltenheit, dass die benötigten Anmeldeinformationen sich bei diesen Anwendungen unterscheiden, weil die Anbieter unterschiedliche Richtlinien umsetzen. Für Nutzer hat dies zur Folge, dass sie sich verschiedene Informationen merken und diese auch in Verbindung mit den richtigen Anwendungen verwenden müssen. Da Zugangsdaten innerhalb eines Unternehmens häufig von zentralen Positionen aus verwaltet werden, muss dafür gesorgt werden, dass diese den Überblick behalten und unter anderem auch Nutzerdaten löschen, wenn ein Mitarbeiter das Unternehmen verlässt. In größeren Firmen kann die Nutzer- und Zugangsdatenverwaltung zu einer enormen Aufgabe heranwachsen, die bei fehlerhafter Ausführung zu Sicherheitsrisiken führen kann.

Eine mögliche Lösung zu diesen Problemen bildet das Federated Identity Pattern. Hierbei werden Authentifizierungsmechanismen implementiert, die getrennt vom eigentlichen Quellcode abstrahiert und die eigentliche Authentifizierung an eine vertrauenswürdige dritte Instanz abgibt. Durch diese Trennung wird die Entwicklung vereinfacht, ermöglicht es Nutzern zwischen mehrerer "*Identity Providers*" (IdPs) zur Authentifizierung wählen zu können und verringert außerdem die administrativen Tätigkeiten der Nutzerverwaltung.

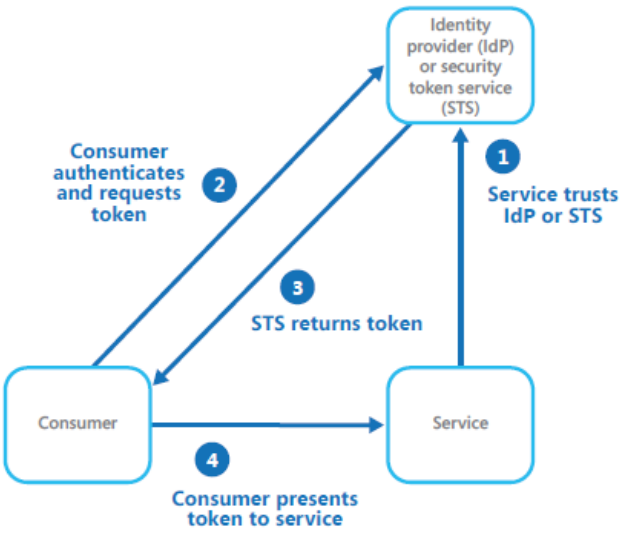

Abb. 9: Das Federated Identity Pattern - Quelle: [HOME14]

Die Abbildung 9 zeigt den Ablauf einer Authentifizierung eines Benutzers gegenüber eines IdPs, um Zugriff zu einem Dienst zu erhalten. Dabei bildet das Vertrauen eines Dienstes zu einem IdP die Basis (**1**). Möchte ein Nutzer nun einen Dienst nutzen, so authentifiziert dieser sich gegenüber dem IdP und fordert ein Token an (**2**). Sofern die Authentifizierung erfolgreich ist, übergibt der IdP dem Nutzer ein Token (**3**) mit dem dieser Zugang zu dem angefragten Dienst erhält (**4**).

Die Funktion eines IdPs können auch Security Token Services (STSs) übernehmen. Heutzutage wird Authentifizierung auch häufig mittels "*Open Authentication*" (OAuth) umgesetzt. Dies ermöglicht es Nutzern mit der Verwendung eines bestehenden Kontos von zum Beispiel Facebook, Microsoft, Google und Amazon zur Identifizeriung und Authentifizierung gegenüber einem Dritten zu verwenden. Ansätze wie OAuth steigern die Sicherheit für den Nutzer, da sensible Informationen zu Nutzername und Passwort nicht für neue Dienste eingegeben werden müssen und dadurch im Netz verteilt werden würden.

[GORD12] [HOME14]

Sharding

Das Speichern von Daten auf einem einzigen zentralen Server kann, vor allem bei großen Cloudanwendungen, von folgenden Einschränkungen betroffen werden:

- **Mangelnder Speicherplatz**
  Gerade bei großen Anwendungen kann die Datenmenge in kurzer Zeit enorm anwachsen. Zwar kann mit weiteren Festplatten der verfügbare Speicher erhöht werden, dies stellt aber auf langfristiger Sicht nicht immer eine endgültige Lösung dar.
- **Fehlende Rechenleistung**
  Wird ein Datenspeicher von einer Vielzahl von Nutzern gleichzeitig abgefragt, kann vor allem bei großen Datenmengen die Antwortzeit drastisch sinken und in Extremfällen auch zu Timeouts führen. Die Steigerung dieser Kapazitäten ist nur in manchen Anwendungsszenarien eine langfristige Lösung.
- **Netzwerk Bandbreite**
  Reicht auch die Rechenleistung eines Systems aus, so kann sich der Flaschenhals auf die Bandbreite eines Netzwerkes verschieben. In diesem Fall reichen die Kapazitäten der Netzwerk Hardware nicht aus, um die angeforderten Daten in akzeptabler Zeit zu übermitteln.
- **Geographische Anforderungen**
  Es kann notwendig sein, dass beispielsweise aus rechtlichen Gründen oder zur Steigerung der Performanz, die Datenspeicher in der selben Region angesiedelt sind, von denen der Nutzer auf diese zugreifen will. Zur Erinnerung: Die EU-DSGVO legt fest, dass personenbezogene Informationen von Bürgern der EU nur auf Servern innerhalb der EU gespeichert werden dürfen.

Das Aufteilen eines Datenspeichers in horizontale Partionen oder "*Shards*", stellt eine Lösung zu den genannten Problemen dar. Jeder dieser Shards besitzt den selben Aufbau, hält aber unterschiedliche Datensätze. Welche Daten ein Shard enthalten soll, muss im Voraus, zum Beispiel anhand von Attributes der Daten, festgelegt werden. Aus diesen Attributen bildet sich der "*Shard Key*" mit dem ein Shard eindeutig adressiert werden kann. Hierbei ist es wichtig, dass die Schlüssel auf den jeweiligen Anwendungsfall zugeschnitten sind, um eine höhere Performanz bei häufig verwendeten Abfrage-Typen zu erreichen.

Der Zugriff auf Daten wird über eine zentrale Sharding Logik abgewickelt, die Anfragen von Anwendungen entgegen nimmt und über verschiedene Strategien herausfindet, welcher Shard die geforderten Daten besitzt:

- Die **Lookup Strategie** nutzt den Shard Key für ein direkten Mapping zu virtuellen oder physischen Partitionen auf denen die Daten gespeichert sind.
- Bei der **Range Strategie** werden verwandte Datensätze zusammen auf den Shards gespeichert und dem Shard Key entsprechend sortiert. Diese Strategie kann vor allem für häufigen Anfragen sinnvoll sein, die Datensätze über einen bestimmten Bereich, zum Beispiel alle Bestellungen eines Monats, anfordern.
- Namensgebend nutzt die **Hash Strategie** ein Hashverfahren, um zu bestimmen, auf welchem Shard ein Datensatz gespeichert werden soll. Dieser Ansatz zielt darauf ab die Daten gleichmäßig zu verteilen und so Hot Spots zu vermeiden. Häufig werden dafür zufällige Elemente mit in das Hashverfahren eingebunden.

Bei der Wahl der richtigen Sharding Strategie ist vor allem eine genaue Betrachtung des Anwendungsfalles notwendig, um darauf basierend die passende Strategie und die Zusammenstellung der Shard Keys festzulegen. In einem mandantenfähigen System könnten, bei der Range Strategie, beispielsweise die Shards so genutzt werden, dass jeder Shard eine bestimmte Anzahl an Mandanten enthält. Wenn eine definierte Grenze erreicht ist, wird ein neuer Shard hinzugeschaltet. Sind die ersten Mandanten des Systems gleichzeitig auch die aktivsten, so kann dies zu Hot Spots auf den jeweiligen Shards führen. Wird stattdessen die Mandanten-ID gehasht und zur differenzierten Speicherung genutzt, so können frühzeitig schon mehrere Shards aktiv und die Last somit besser verteilt sein.

[HOME14] [KERS18]

Zusammenfassung

Obwohl der Cloud Computing Bereich schon viele Jahre produktiv im Einsatz ist, sind die Ideen und Konzepte noch nicht ausgeschöpft. Immer neue Technologien und Dienste ermöglichen vollkommen neue Einsatzgebiete und Anwendungsansätze für die Cloud. Anbieter wie Amazon erlauben es immer mehr Unternehmen ihre digitalen Prozesse und Softwareanwendungen in die Cloud zu verlagern, um dabei in den Genuss der immer schwerer wiegenden Vorteile von Cloud-Infrastrukturen zu kommen.

Nachteile und Begrenzungen der Cloud können in manchen Fällen durch "neue" Bereiche wie Edge und Fog Computing ausgebessert werden. Viele anfängliche Probleme sind seit der ersten Cloud-Dienste wie Amazon EC2 längst gelöst und durch Best Practices und Design Patterns können viele Risiken schon durch Designentscheidungen umgangen werden.

Neuheiten und aktuelle Trends

Die Hewlett Packard Enterprise (HPE) hat im Juni 2018 die "*Greenlake Hybrid Cloud*" vorgestellt. Dieses Produkt soll dabei helfen Kosten zu sparen, indem automatisiert die richtigen Cloud-Ressourcen ausgewählt werden. Das Marktforschungsinstitut Gartner sagt, dass bis 2020 über 40% der Kunden zu viel für ihre Public Cloud bezahlen. Die Greenlake Hybrid Cloud ist derzeit mit Microsoft Azure und AWS kompatibel und stellt eine Art von Cloud dar, die sich selbst konfiguriert. [NICK18]

Ein neues Buzzword im Bereich Cloud Computing stellt der Begriff "*Cloud Native Application*" dar. Gemeint sind damit Anwendungen, die sich von ihrem Design, ihrer Entwicklung und der Form mit der sie bereitgestellt werden, von Natur aus in der Cloud wohl fühlen. Vor allem Anwendungen aus dem Microservice Bereich, die auf Basis von Containern bereitgestellt werden und für verteilte Anwendungsfälle konzipiert sind, können als Cloud Native Application bezeichnet werden. Geprägt wird dieser Begriff vor allem durch die "*Cloud Native Computing Foundation*". Ebenso spielen aber Technologien wie Kubernetes mit ein, die eine hochgradig skalierbare Bereitstellung von Container-basierten Anwendungen ermöglichen. [KUBE18] [OWEN17]

Literaturverzeichnis

[AVRA16]	Avram, Abel ; InfoQ, 25.06.2016: FaaS, PaaS, and the Benefits of Serverless Architecture
	URL: https://www.infoq.com/news/2016/06/faas-serverless-architecture (abgerufen am 27.06.2018)
[AWS18a]	Amazon AWS: Produkte
	URL: https://aws.amazon.com/de/?nc2=h_lg (abgerufen am 30.06.2018)
[AWS18b]	Amazon AWS: AWS Free Tier
	URL: https://aws.amazon.com/free/ (abgerufen am 30.06.2018)
[AWS18c]	Amazon AWS: AWS Customer Agreement
	URL: https://aws.amazon.com/agreement/ (abgerufen am 30.06.2018)
[AWS18d]	Amazon AWS ; AWS Educate: Teach Tomorrow's Cloud Workforce Today
	URL: https://aws.amazon.com/education/awseducate/ (abgerufen am 30.06.2018)
[AWS18e]	Amazon AWS: AWS Management Console
	URL: https://us-east-2.console.aws.amazon.com/console/home?region=us-east-2 (abgerufen am 30.06.2018)
[AWS18f]	Amazon AWS: AWS Pop-up Loft
	URL: https://aws.amazon.com/start-ups/loft/ (abgerufen am 30.06.2018)
[AWS18g]	Amazon AWS: AWS Pop-up Loft FAQ
	URL: https://aws.amazon.com/start-ups/loft/faq (abgerufen am 30.06.2018)
[AWS18h]	Amazon AWS: AWS IAM Dashboard
	URL: https://console.aws.amazon.com/iam/home?region=us-east-2#/users (abgerufen am 30.06.2018)
[AWS18i]	Amazon AWS: Verwalten von Rollen
	URL: https://aws.amazon.com/de/iam/details/manage-roles/?nc1=f_ls (abgerufen am 30.06.2018)
[AWS18j]	Amazon AWS: AWS Identity and Access Management (IAM)
	URL: https://aws.amazon.com/iam/?nc2=h_m1 (abgerufen am 30.06.2018)
[AWS18k]	Amazon AWS: EC2 Launch Wizard
	URL: https://us-east-2.console.aws.amazon.com/ec2/v2/home?region=us-east-2#LaunchInstanceWizard: (abgerufen am

	30.06.2018)
[AWS18l]	Amazon AWS: EC2
	URL: https://aws.amazon.com/de/ec2/ (abgerufen am 30.06.2018)
[AWS18m]	Amazon AWS: Globale AWS-Infrastruktur
	URL: https://aws.amazon.com/de/about-aws/global-infrastructure/ (abgerufen am 01.07.2018)
[AWS18n]	Amazon AWS: Amazon Resource Names (ARNs) and AWS Service Namespaces
	URL: https://docs.aws.amazon.com/general/latest/gr/aws-arns-and-namespaces.html (abgerufen am 01.07.2018)
[AWS18o]	Amazon AWS: Was ist ein Cloud-Objektspeicher?
	URL: https://aws.amazon.com/de/what-is-cloud-object-storage/ (abgerufen am 01.07.2018)
[AWS18p]	Amazon AWS: Amazon S3
	URL: https://aws.amazon.com/de/s3/ (abgerufen am 01.07.2018)
[AWS18q]	Amazon AWS: AWS Lambda
	URL: https://aws.amazon.com/de/lambda/ (abgerufen am 01.07.2018)
[AWS18r]	Amazon AWS: AWS-Fallstudie: The Seattle Times
	URL: https://aws.amazon.com/de/solutions/case-studies/the-seattle-times/ (abgerufen am 01.07.2018)
[BIAN20]	Bianchi, Alessandra ; Inc., 01.04.2000: Say good-bye to software as we know it and hello to ASP start-up
	URL: https://www.inc.com/magazine/20000401/18093.html (abgerufen am 26.06.2018)
[BSI18]	Bundesamt für Sicherheit in der Informationstechnik (BSI): Cloud Computing Grundlagen
	URL: https://www.bsi.bund.de/DE/Themen/DigitaleGesellschaft/CloudComputing/Grundlagen/Grundlagen_node.html (abgerufen am 03.05.2018)
[BUTL17]	Butler, Brandon ; Network World, 21.09.2017: What is edge computing and how it's changing the network
	URL: https://www.networkworld.com/article/3224893/internet-of-things/what-is-edge-computing-and-how-it-s-changing-the-network.html (abgerufen am 29.06.2018)
[BRAD16]	Bradbury, Danny ; The Register, 21.06.2016: DDoS, the cloud and you
	URL: https://www.theregister.co.uk/2016/07/21/ddos_the_cloud_and_you/ (abgerufen am 29.06.2018)
[CISC13]	Cisco, 2013: Fog Computing, Ecosystem, Architecture and Applications
	URL: https://web.archive.org/web/20140922061044/http://www.cisco.com/web/about/ac50/ac207/crc_new/university/RFP/rfp13078.html (abgerufen am 29.06.2018)
[CLOU18]	Cloud Harmony: CloudSquare Service Status
	URL: https://cloudharmony.com/status (abgerufen am 29.06.2018)
[DROP18]	Dropbox DE: Datenschutzrichtlinien
	URL: https://www.dropbox.com/de/privacy (abgerufen am 29.06.2018)
[EDWA17]	Edwards, Elaine: The Irish Times, 16.10.2017: US supreme court to hear appeal in Microsoft warrant case
	URL: https://www.irishtimes.com/business/technology/us-supreme-court-to-hear-appeal-in-microsoft-warrant-case-1.3257825 (abgerufen am 29.06.2018)
[ELLE17]	Elle, Jessica ; Forestgiant, 05.04.2017: Fog VS Edge Computing
	URL: https://forestgiant.com/articles/fog-vs-edge/ (abgerufen am 26.06.2018)
[EUDS18]	Intersoft Consulting: Datenschutz-Grundverordnung DSGVO
	URL: https://dsgvo-gesetz.de/ (abgerufen am 29.06.2018)
[EVAN14]	Evans, Chris ; ComputerWeekly, 04.2014: Write-through, write-around, write-back: cache explained
	URL: https://www.computerweekly.com/feature/Write-through-write-around-write-back-Cache-explained (abgerufen am 01.07.2018)

[FELD17]	Felde, Christian ; Blog of Christian Felde, 20.12.2017´: On edge architecture
	URL: https://blog.cfelde.com/2017/12/on-edge-architecture/ (abgerufen am 27.05.2018)
[FOOT17]	Foote, Keith D. ; Dataversity, 22.06.2017: A Brief History of Cloud Computing
	URL: http://www.dataversity.net/brief-history-cloud-computing/ (abgerufen am 27.05.2018)
[FORR06]	Forrest, Brady ; O'Reilly Radar, 25.09.2006: Zimki, hosted JavaScript environment
	URL: http://radar.oreilly.com/2006/09/zimki-hosted-javascript-enviro.html (abgerufen am 26.06.2018)
[GASS16]	Gassner, Heinz ; Smart Industry Forum, 02.12.2016: What Do We Actually Mean By: IaaS, PaaS, SaaS?
	URL: https://smartindustryforum.org/what-do-we-actually-mean-by-iaas-paas-saas/ (abgerufen am 21.06.2018)
[GEDI18]	GE Digital: What is Edge Computing?
	URL: https://www.ge.com/digital/blog/what-edge-computing#edge-computing-vs-cloud-computing-3 (abgerufen am 27.05.2018)
[GILE18]	Giles, Martin ; MIT Technology Review, 02.01.2018
	URL: https://www.technologyreview.com/s/609641/six-cyber-threats-to-really-worry-about-in-2018/ (abgerufen am 29.06.2018)
[GORD12]	Gordon, Whitson ; lifehacker, 13.06.2012: Understanding OAuth: What Happens When You Log Into a Site with Google, Twitter, or Facebook
	URL: https://lifehacker.com/5918086/understanding-oauth-what-happens-when-you-log-into-a-site-with-google-twitter-or-facebook (abgerufen am 01.07.2018)
[GREE18]	Greenberg, Andy ; WIRED, 27.06.2018: Marketing firm Exactis leaked a personal info database with 340 million records
	URL: https://www.wired.com/story/exactis-database-leak-340-million-records/ (abgerufen am 29.06.2018)
[HAN17]	Han, Bowei ; Medium, 05.11.2017: An Introduction to Serverless and FaaS (Function as a Service)
	URL: https://medium.com/@BoweiHan/an-introduction-to-serverless-and-faas-functions-as-a-service-fb5cec0417b2 (abgerufen am 27.06.2018)
[HERS17]	Hersher, Rebecca ; National Public Radio, 03.03.2017: Amazon And The $150 Million Typo
	URL: https://www.npr.org/sections/thetwo-way/2017/03/03/518322734/amazon-and-the-150-million-typo?t=1529930367722 (abgerufen am 29.06.2018)
[HOME14]	Homer, Alex; Sharp, John; Brader, Larry; Narumoto, Masashi; Swanson, Trent ; Microsoft, 04.03.2014: Cloud Design Patterns
	URL: https://www.microsoft.com/en-us/download/details.aspx?id=42026 (abgerufen am 01.07.2018)
[IBMJ09]	IBM Journal of Research and Development, 07.2009: The Reservoir model and architecture for open federated cloud computing
	URL: https://ieeexplore.ieee.org/document/5429058/ (abgerufen am 19.06.2018)
[INNO17]	Innocent, Johnson ; DZone / Cloud Zone, 09.03.2017: Cloud Computing Deployment Models
	URL: https://dzone.com/articles/cloud-computing-deployment-models (abgerufen am 28.06.2018)
[KERS18]	Kerstiens, Craig ; Citrusdata, 10.01.2018: Database sharding explained in plain English
	URL: https://www.citusdata.com/blog/2018/01/10/sharding-in-plain-english/ (abgerufen am 01.07.2018)
[KUBE18]	Kubernetes.io: Production-Grade Container Orchestrierung
	URL: https://kubernetes.io/ (abgerufen am 01.07.2018)
[LARK18]	Larkin, Andrew ; Cloudacademy, 26.06.2018
	URL: https://cloudacademy.com/blog/disadvantages-of-cloud-computing/ (abgerufen am 29.06.2018)
[LYNL18]	Lynley, Matthew ; techcruch.com: Amazon may be developing AI chips for Alexa
	URL: https://techcrunch.com/2018/02/12/amazon-may-be-developing-ai-chips-for-alexa/ (abgerufen am 27.05.2018)
[MARK07]	Markham, Gervase ; Hacking for Christ, 25.09.2007: Zimki Shuts Down
	URL: http://blog.gerv.net/2007/09/zimki_shuts_down/ (abgerufen am 26.06.2018)
[MILL18]	Miller, Paul ; The Verge, 07.05.2018: What is edge computing?
	URL: https://www.theverge.com/circuitbreaker/2018/5/7/17327584/edge-computing-cloud-google-microsoft-apple-amazon

	(abgerufen am 27.05.2018)
[NETW18]	Network-Karriere, 18.01.2018: EU-Datenschutz für personenbezogene Daten ab 2018 wirksam - Server-Standort entscheidend!
	URL: https://www.network-karriere.com/2018/01/18/eu-datenschutz-f%C3%BCr-personenbezogene-daten-ab-2018-wirksam-server-standort-entscheidend/ (abgerufen am 29.06.2018)
[NICK18]	Nickel, Oliver ; Golem.de, 20.06.2018: Software automatisiert die Ressourcen in der Cloud
	URL: https://www.golem.de/news/hpe-greenlake-hybrid-cloud-software-automatisiert-die-ressourcen-in-der-cloud-1806-135051.html (abgerufen am 01.07.2018)
[NIST11]	National Institute of Standards and Technology (NIST) - The NIST Definition of Cloud Computing 2011
	URL: https://csrc.nist.gov/publications/detail/sp/800-145/final (abgerufen am 03.05.2018)
[OPEN17]	OpenFog, 09.02.2017: OpenFog Reference Architecture for Fog Computing
	URL: https://www.openfogconsortium.org/wp-content/uploads/OpenFog_Reference_Architecture_2_09_17-FINAL.pdf (abgerufen am 10.06.2018)
[OPTO18]	OPTO 22: Fog Computing vs. Edge Computing
	URL: http://info.opto22.com/fog-vs-edge-computing (abgerufen am 29.06.2018)
[OWEN17]	Owen, Ken ; Cloud Native Computing Foundation, 15.05.2017: Developing Cloud Native Applications
	URL: https://www.cncf.io/blog/2017/05/15/developing-cloud-native-applications/ (abgerufen am 01.07.2018)
[RAMA17]	Rama, Galdys ; AWS insider, 01.08.2017: Report: AWS Market Share Is Triple Azure's
	URL: https://awsinsider.net/articles/2017/08/01/aws-market-share-3x-azure.aspx (abgerufen am 30.06.2018)
[REGA11]	Regalado, Antonio ; MIT Technology Review, 31.10.2011: Who Coined 'Cloud Computing'?
	URL: https://www.technologyreview.com/s/425970/who-coined-cloud-computing/ (abgerufen am 27.05.2018)
[ROBI13]	Robinson, Paul ; JBossDeveloper, 19.04.2013: Compensating Transactions: When ACID is too much
	URL: https://developer.jboss.org/wiki/CompensatingTransactionsWhenACIDIsTooMuch (abgerufen am 01.07.2018)
[ROUS17a]	Rouse, Margaret ; TechTarget, 09.2017: Infrastructure as a Service (IaaS)
	URL: https://searchcloudcomputing.techtarget.com/definition/iaas-IaaS (abgerufen am 21.06.2018)
[ROUS17b]	Rouse, Margaret ; TechTarget, 11.2017: XaaS (Everything as a Service)
	URL: https://searchcloudcomputing.techtarget.com/definition/XaaS-anything-as-a-service (abgerufen am 27.06.2018)
[SHAW17]	Shaw, Keith ; NetworkWorld, 19.12.2017: What is a hypervisor?
	URL: https://www.networkworld.com/article/3243262/virtualization/what-is-a-hypervisor.html (abgerufen am 21.06.2018)
[STAT16]	Statt, Nick ; The Verge: How an army of vulnerable gadgets took down the web today
	URL: https://www.theverge.com/2016/10/21/13362354/dyn-dns-ddos-attack-cause-outage-status-explained (abgerufen am 27.05.2018)
[UTLE18]	Utley, Gary ; The CWPS Blog, 12.05.2018: 6 Most Common Cloud Computing Security Issues
	URL: https://www.cwps.com/blog/cloud-computing-security-issues (abgerufen am 29.06.2018)
[VENE14]	Venezia, Paul ; InfoWorld, 23.06.2014: Murder in the Amazon cloud
	URL: https://www.infoworld.com/article/2608076/data-center/murder-in-the-amazon-cloud.html (abgerufen am 29.06.2018)
[WARD18]	Ward, Susan ; The balance small business, 15.04.2018: The Cons of Cloud Computing
	URL: https://www.thebalancesmb.com/disadvantages-of-cloud-computing-4067218 (abgerufen am 29.06.2018)
[WATT17]	Watts, Stephen ; BMC Blogs, 22.09.2017: SaaS vs PaaS vs IaaS: What's The Difference and How To Choose
	URL: https://www.bmc.com/blogs/saas-vs-paas-vs-iaas-whats-the-difference-and-how-to-choose/ (abgerufen am 26.06.2018)
[WIKI18]	Wikipedia, 26.06.2018: As a service
	URL: https://en.wikipedia.org/wiki/As_a_service (abgerufen am 27.06.2018)
[WOOD13]	Wood, Peter ; SlideShare, 28.05.2013: Attacking the cloud with social engineering

URL: https://www.slideshare.net/PeterWoodx/attacking-the-cloud-with-social-engineering (abgerufen am 29.06.2018)