

# Amazon Elastic Compute Cloud

Peter Gehrt

Hochschule Mannheim - Fakultät für Informatik

Matrikelnummer: 0625379

`peter.gehrt@stud.hs-mannheim.de`

**Zusammenfassung** Neben Büchern, Elektronikprodukten und Kleidung bietet Amazon nun auch stundenweise Rechenleistung in Form virtueller Server oder Datenspeicher an. Kunden können in der Amazon Elastic Compute Cloud (EC2) virtuelle Server für ihre eigenen Anwendungen mieten und dabei über einen Webservice skalierbare Anwendungslösungen entwickeln. EC2 bietet hiermit eine einsteigerfreundliche Möglichkeit der professionellen Migration von produktiven Systemumgebungen und einen hochverfügbaren, kostengünstigen und sicheren Weg für datenintensive Berechnungen.

EC2 ist Bestandteil der Amazon Web Services (AWS) und somit eine Anwendung von Cloud-Computing sowie Infrastructure as a Service (IaaS). Das Marktforschungsinstitut Gartner setzte Cloud-Computing in seiner letzten Prognose der Top10 Technologies auf den zweiten Platz und prognostizierte, dass im Jahr 2010 über 80 Prozent der Unternehmen diese Technologie als essenziellen Bestandteil einsetzen werden [1]. Amazon realisiert als einer der ersten Anbieter Cloud-Computing-Lösungen, die sich durch einen sehr einfachen Arbeitsablauf, On-Demand Zugriff auf scheinbar unendliche Ressourcen, zeitbasierte Abrechnung und minimale eigene IT Anforderungen im eigenen Unternehmen auszeichnen [2]. Dennoch klingt EC2 wie eine bereits seit Jahren eingesetzte Technik, denn so genannte virtuelle Server sowie Rootserver werden schon länger von Hostinganbietern angeboten. Negative Stimmen wie Larry Ellison, Gründer des Oracle Softwarekonzerns, erklären Cloud-Computing sei keine Zukunftstechnologie, sondern ein bereits seit Jahren eingesetztes Mittel <sup>1</sup>. Von welchen Vorteilen profitiert das eigene Unternehmen durch konkrete Projekte wie EC2 und welche Risiken birgt die neue Technik?

---

<sup>1</sup> The interesting thing about cloud computing is that we've redefined cloud computing to include everything that we already do ... The computer industry is the only industry that is more fashion-driven than women's fashion. Maybe I'm an idiot, but I have no idea what anyone is talking about. What is it? It's complete gibberish. It's insane. When is this idiocy going to stop? (Larry Ellison) [3], [4]

## 1 Entstehung

Das breite Spektrum der verfügbaren Cloud-Computing Angebote lässt sich in zwei Gruppen unterteilen. Bei den sogenannten *Public Clouds* werden einzelne Anwendungen in der Cloud laufen gelassen aber auch große Teile der unternehmenseigenen Infrastruktur ausgelagert. Dem gegenüber stehen Modelle, die es einzelnen Firmen ermöglichen Ihre IT-Infrastruktur als *Private Cloud* nur unternehmensintern für eigene Zwecke verfügbar zu machen.

Amazon betreibt sehr große verteilte Rechenzentren in den Vereinigten Staaten sowie Europa und muss für die eigenen Kerndienste deutliche Überkapazitäten für Stoßzeiten vorhalten. Diese verfügbaren Ressourcen liegen aber in den restlichen Zeiten brach. Durch die Konsolidierung mittels XEN entstand eine einheitliche virtualisierte Serverlandschaft und zur automatischen Steuerung musste eine einfache Programmierschnittstelle (API) erstellt werden. Diese ursprünglich für interne Arbeitsabläufe entwickelten Schnittstellen wurden ab 2006 auch für externe Benutzung angeboten und ständig erweitert. Seitdem können Benutzer durch eine einfache Anmeldung an die Amazon Web Services, bei der nur eine Kreditkarte erforderlich ist, diese Dienstleistungen in Form von virtualisierten Servern nutzen. Seit Bestehen von EC2 wurden insgesamt 15.5 Millionen Instan-

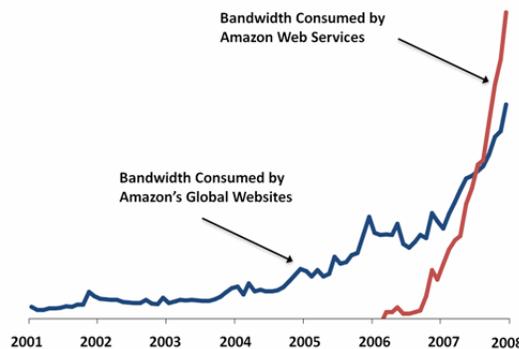


Abbildung 1. Amazon Bandbreitennutzung

zen gestartet, allein im September 2009 waren es über 50.000 Stück und Ende Oktober 2009 setzte Amazon circa 40.000 Hardwareserver nur für EC2 ein. Damit übertrafen die AWS-Benutzer bereits 2007 (Abbildung 1<sup>2</sup>) die von Amazon selbst benötigte Bandbreite. [5]

<sup>2</sup> <http://aws.typepad.com/aws/2008/05/lots-of-bits.html>

## 2 Eigenschaften

Durch die Migration eines Servers aus einem herkömmlichen Rechenzentrum zu EC2 versprechen sich die meisten Anwender neben einer flexiblen Skalierung und einem damit verbundenen Preisvorteil noch weitere Vorteile wie hohe Sicherheit und Performance.

Nach einer Definition der Handelsbundesbehörde der Vereinigten Staaten des National Institute of Standards and Technology (NIST) ist Cloud-Computing eine Umwandlung von einmaligen Investitionskosten in variable laufende Kosten [6]. Dieses Prinzip wird unter dem Schlagwort *pay as you go* zusammengefasst und lässt sich mit den bei Mobiltelefonen bekannten Prepaid Tarifen vergleichen. Es werden weder einmalige Investitionskosten noch ein monatlicher Grundbetrag fällig, sondern es werden nur Leistungen berechnet, die wirklich verbraucht werden. Bei EC2 versteht man darunter die stundenweise Abrechnung pro laufender Rechnerinstanz. Auch wenn damit die Kosten höher als beim Betrieb der Anwendung im eigenen Rechenzentrum zu sein scheinen, muss man dennoch die Elastizität und den Risikoübergang aller damit verbundenen Faktoren auf den Cloud-Computing Anbieter mit einbeziehen. Zusätzlich hat man weniger Verwaltung und Wartungsaufwand, denn es sind bereits sämtliche Kosten in die Laufzeit der Amazon Instanz bereits eingerechnet und brauchen nicht weiter mit einkalkuliert werden. Bei der Umsetzung eines umfangreichen Projektes oder der Erörterung einer möglichen Migration der eigenen Anwendung in die Cloud empfiehlt sich in der Planungsphase der Einsatz des Simple Monthly Calculator [7]. Somit können alle Kombinationen von Amazon Services im Voraus berechnet und angepasst werden.

Neuen Untersuchungen zufolge liegt die gemittelte *Serverauslastung* in herkömmlichen Rechenzentren nur zwischen 5 und 20 Prozent [8] [9]. Der Anwender einer EC2 kann durch Überwachung und getriggerte Aktionen die abgerufene Leistung jederzeit den eigenen Bedürfnissen anpassen und dadurch fast eine 80-prozentige Auslastung erreichen. Durch das Preismodell von Amazon sind die Kosten für den Betrieb von einer Instanz über 100 Stunden oder den Parallelbetrieb von 100 Instanzen für eine einzige Stunde gleich. Somit arbeitet eine Anwendung in der EC2 zeitsparender und effizienter je besser sie parallelisierbar ist. Das Risiko der Überprovisionierung sinkt und im Gegensatz zu den bisherigen Verfahren trägt man bei EC2 nicht das Risiko der Unterproduktion. Dadurch wird der dauerhafte Betrieb von 500 Servern überflüssig, wenn im Mittel nur 100 notwendig sind. Zusätzliche saisonale Schwankungen wie bei Webshops zu Weihnachtszeiten üblich fallen dabei noch mehr ins Gewicht. Gerade für junge Start-up-Unternehmen ist diese dynamische Skalierung ein sehr wichtiges Argument [9]. Schließlich ist auch eine Zusammenarbeit mit Green IT möglich: Durch die Konsolidierung werden weniger Server benötigt und man unterstützt aktuelle politische Bestrebungen die Systeme möglichst umweltschonend einzusetzen.

Durch die Bereitstellung einer solchen Umgebung für die Öffentlichkeit musste Amazon aber auch die eigenen Datensicherheit und die des Netzwerks für die

Benutzer gewährleisten. Durch Sicherheitsmechanismen von XEN werden die einzelnen Gäste erfolgreich voneinander getrennt und durch die im Hypervisor des XEN laufende Firewall lässt nur explizit erlaubte Verbindungen zu. Seit Ende 2008 bietet Amazon auch ein *Service Level Agreement* (SLA) an, in denen Benutzer genau definierte Leistungen und Verfügbarkeiten garantiert werden. Anhand dieser SLA kann jeder Anwender mit seinem eigenen Rechenzentrum vergleichen, ob er diese Gewährleistungen selbst geben kann.

Eine Gefahr der Sicherheit in heutigen Unternehmen ist der sogenannte Single Point Of Failure. Dabei wird die Verfügbarkeit einer Anwendung durch einen einzelnen kritischen Punkt in der Hardware oder Software gefährdet [10]. Der menschliche *Fehlerfaktor* beim Austausch von Hardwarekomponenten oder falsche Reaktionen auf Störfälle ist durch die weitgehende Automatisierung fast ausgeschlossen. Bei den wenigen verbleibenden Eingriffen durch Servicepersonal kann Amazon auf sehr erfahrene und gut ausgestattete Techniker und Administratoren zurückgreifen. Defekte Komponenten oder Softwareanpassungen lassen sich unkomplizierter und schneller bearbeiten, wobei ein spezialisierter Mitarbeiter bei Amazon deutlich mehr Server administrieren kann als in einem herkömmlichen Rechenzentrum. Amazon unterhält Rechenzentren auf verschiedenen Kontinenten und auch innerhalb dieser Regionen bietet es mehrere Verfügbarkeitszonen. Diese sind untereinander abgeschottet und bieten eine hohe Verfügbarkeit und Redundanz über alle Ebenen. Dadurch kann Amazon alle kritischen Komponenten mehrfach bereitstellen, von der Netzwerkanbindung über Server und Datenhaltung kann alles in der virtuellen Umgebung kurzzeitig oder auf Dauer repliziert werden. Somit ist die Datenhaltung durch die mehrfache Redundanz und getrennte Zonen in vielen Fällen deutlich sicherer als in einem unternehmensinternen Rechenzentrum. Zusätzlich kann die Ausfallsicherheit einer erstellten Serverlandschaft durch Mechanismen wie flexibel zuweisbare IP-Adressen oder virtuelle Load-Balancer deutlich erhöht werden. Um die damit gebotene Fehlertoleranz weiter zu erhöhen, ist die kombinierte Nutzung von verschiedenen CC Anbieter vorstellbar.

Cloud-Computing Anbieter wie Amazon oder Google setzen deutlich mehr Ressourcen in Form von Mitarbeitern oder Geldmitteln im Kampf gegen Cyberkriminalität ein, als es ein normales Unternehmen vermag. Durch die im Netzwerk bereits eingebauten Schutzmechanismen wie Firewalls und abgesicherte Subnetze wird ein Großteil von Angriffen wie DoS, Man-in-the-Middle, IP-Spoofing, Portscans und Paket-Sniffing vereitelt. Ein Angriff auf eine in EC2 aufgebaute Architektur ist deutlich schwerer als auf ein alleinstehendes Rechenzentrum, da Amazon den genauen Standort der eigenen Rechenzentren geheim hält, um keine Angriffsfläche zu bieten. Man ist gegen kompromittierende Angriffe durch Einbruchversuche oder Netzwerkattacken besser gewappnet. Eine neue Untersuchung des Berkley RAD Institutes [11] zeigt, dass man DoS-Attacks durch Hochskalierung sehr effektiv entgegenwirken kann. Unter einer DoS-Attacke versteht man einen Überlastungsangriff, der den Ausfall von Netzwerkdiensten zur Folge hat. Hierbei setzen die Angreifer heutzutage auf angemietete Bot-Netzwerke [12]. Die Angreifer bezahlen in der Regel für die verwendeten Bot-

Netzwerke deutlich mehr Geld pro Stunde als der angegriffene AWS-Benutzer. In diesem Fall kann man einen Angriff einfach durch kurzzeitige Hochskalierung aussitzen und warten, bis der Angreifende aufgibt [8].

Viele Anwender von Cloud-Computing versprechen sich durch die Illusion von fast unendlicher Performance einen *Geschwindigkeitsvorteil*. Die von Amazon eingesetzte Serverhardware bietet eine möglichst hohe Leistungsdichte und stammt aus dem High End Sektor. Die I/O- und CPU-Komponenten müssen auch für die Amazon internen Zwecke Hochverfügbarkeit, einen hohen Durchsatz und geringe Zugriffszeiten bieten. Zusätzlich kann durch die zonenbasierte Aufteilung mittels CloudFront der angesprochene Service sehr nah am Endanwender platziert werden und erreicht damit niedrige Zugriffszeiten für die Endanwender. Auch die von Amazon angebotenen Mechanismen wie Autoscaling zur Anpassung der genutzten EC2 Kapazitäten oder Elastic-Load-Balancer bringen einen Administrator in die Position, schnell und wirksam Schwachstellen zu finden und zu verbessern.

Eine *Untersuchung* der EC2 Instanzen im Rahmen dieser Semesterarbeit zeigte überdurchschnittlich gute Performance und Ladezeiten sowie hohen Netzwerkdurchsatz. Die Performance einer einzelnen Instanz definiert Amazon anhand der Elastic Compute Unit (ECU), welche die vergleichbare Rechengeschwindigkeit eines Intel Xeon/Opteron 1.2 GHz Prozessors aus dem Jahr 2007 aufweisen soll. Durch Testmessungen mit verschiedenen Benchmark-Programmen lässt sich diese Definition einfach belegen. Hierzu wurde die Prozessor- und Arbeitsspeicher-Performance einer Small-Instanz (Singlecore Xeon mit 1.2 GHz und 2 GB Arbeitsspeicher) und eine Quadruple-Instanz (Xeon X5550 mit 3 GHz und 8 Kernen sowie 68 GB Arbeitsspeicher) durch Berechnung von großen Primzahlen gemessen und mit den aktuellen Spitzenwerten der jeweiligen Benchmark-Programme verglichen. Mit einem Spitzenwert in wprime [13] von 7 Sekunden für 32 MB Primzahlen erreichte der Quadruple-Server die Performance eines aktuellen Intel Core i7 920 mit 4 GHz und damit fast einen neuen Spitzenwert. Die Small-Instanz brauchte mit 170 Sekunden deutlich länger und ist lässt sich mit einem 2 GHz Athlon-64 Prozessor vergleichen. Ein Test der Kernelkompilierungszeit bestätigte mit 33 Sekunden einer Quadruple-Instanz im Gegensatz zu 604 Sekunden einer Small-Instanz diese Einschätzung. Die Festplatten einer Small-Instanz erreichen in den Messungen einen Durchsatz von 300 MB/s, eine Quadruple-Instanz von über 860 MB/s.

Die Ping-Antworten der Instanzen betragen innerhalb des Internets zwischen 30 und 40 Millisekunden und befanden sich innerhalb des Amazon-Netzwerkes mit unter einer Sekunde fast im Bereich der Messungenauigkeit. Gleichzeitig durchgeführte Netzwerkdurchsatztests zeigten, dass die Performance direkt mit der verwendeten Instanzklasse skaliert. Eine Small-Instanz erreichte im Schnitt 20-50 Mbit/s, wogegen eine Large-Instanz mit 90Mbit/s bereits die Referenzstelle - einen Server der Universität Mannheim - auslastete. Innerhalb des AWS Netzwerkes erreichte der Durchsatz je nach Instanzklasse zwischen 400 und 940 Mbit/s und damit fast volle Gigabitgeschwindigkeit. Das vollständige Messpro-

tokoll findet sich im Anhang. Eine Ermittlung der Java-Performance mittels verschiedenen Sortieralgorithmen wurde von Christian Schwarz unter [14] beschrieben.

### 3 Beispiele

Heutzutage fallen in den meisten Anwendungen immer größere Datenmengen an, zum Beispiel produzieren Londons Verkehrskameras bis zu 8 TB und das LHC bis zu 20 TB Daten jeden Tag. Datacruncher verarbeiten diese Daten gleichzeitig durch viele parallele Batchabläufe und werden besonders in Analyseprozessen eingesetzt. [15]

Aber auch einmalige Geschäftsprozesse werden durch das Instanzmodell von Amazon profitieren. Die amerikanische Tageszeitung The New York Times benötigte im Jahr 2008 eine kurzfristige Lösung zur Konvertierung von über 4 Millionen alten Artikeln in einheitliche Bilddateien und entschied sich für EC2. Es wurden mehrere große EC2 Instanzen gemietet und die Konvertierungen in nur 36 Stunden für 260 US-Dollar durchgeführt. Im Vergleich zur Konzeption, Beschaffung und Aufbau eines eigenen Rechenzentrums ein sehr günstiger Preis. [16]

Schnell skalierende Instanzen können in bereits vorhandenen Rechenzentren als Spitzenlast-Dämpfer agieren. Das eigene Rechenzentrum deckt die durchschnittlichen Lasten ab und externe Ressourcen aus EC2 die Anfragen zu Spitzenzeiten. Anwendungsszenarios die durch die Migration in die Amazon Cloud am stärksten profitieren sind zurzeit jedoch junge Start-up-Unternehmen die ihre zukünftige IT-Anforderungen weder planen noch vorzeitig bezahlen können. Ein Beispiel für die gut funktionierende Skalierung von Amazon ist Animoto, eine Web 2.0 Anwendung welche aus hochgeladenen Bildern vollautomatisch Videos für die Benutzer erstellt. Animoto lässt ihre Server in EC2 laufen. Durch einen starken Zulauf Mitte 2008 musste Animoto von 50 auf über 400 Server in wenigen Tagen skalieren, das entspricht einer Verdopplung alle 12 Stunden. Aber ebenso konnte das Unternehmen im Anschluss ihre Ressourcen wieder runterskalieren.

Das Foto und Videoportal SmugMug betreibt selbst keine Infrastrukturdienste. Es bedient sich nur bei anderen Anbietern wie OpenID für die Benutzeranmeldung, EC2 als Serverdienstleister und S3 für die Datenhaltung. Mit dieser Kombination von externen Services erwirtschaftete SmugMug im Jahr 2005 einen Gewinn von über 10 Millionen US-Dollar und beschäftigte dabei weniger als 20 Angestellte. Weitere Szenarien und Fallbeispiele sind bei Amazon unter Case-Studies zu finden [17].

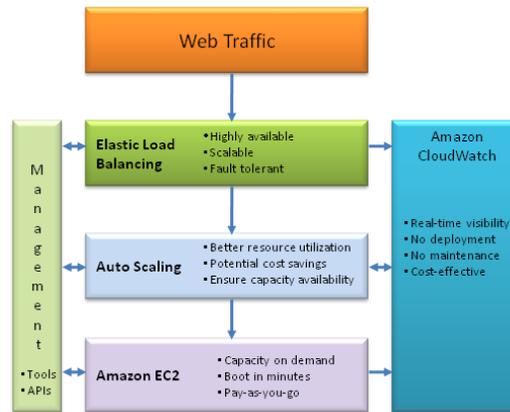


Abbildung 2. Aufbau Amazon Webservices

## 4 EC2

Auf der untersten Ebene (Abbildung 2<sup>3</sup>) bietet AWS neben EC2 auch transparente Speicherlösungen wie S3 oder verteilte Datenbankmanagementsysteme wie SimpleDB oder RDS an. Darauf basieren die virtuellen Server von EC2 sowie die zugehörigen Monitoring- und Skalierungs-Services. Auf der obersten Ebene erwachsen daraus fertige Dienstleistungslösungen von Amazon für Webindizierung, komplexe Algorithmen wie MapReduce und Crowdsourcing-Marktplätze wie MechanicalTurk.

### 4.1 Programmierung

Zur Steuerung aller angebotenen Dienstleistungen bietet Amazon einen WSDL-Webservice an, welchen man durch eigene Programme in Java, C, Python, Ruby oder PHP aufrufen kann. Um den Einstieg zu erleichtern, findet sich in der mitgelieferten Amazon Java Bibliothek über 130 selbsterklärende Beispiele. Anfang November 2009 wurde eine .NET Bibliothek für C veröffentlicht. Entwickeln von Amazon Webservices und EC2 Anwendungen bietet Amazon ein Plug-in für die Entwicklungsumgebung Eclipse an (AWS Toolkit for Eclipse). Zur Erhöhung der Sicherheit bietet sich der zertifikatsbasierte Zugriff für alle Webservice-Zugriffe an, die Kommandos selbst können zusätzlich für die Übertragung in unsicheren Netzwerken verschlüsselt werden.

Als Vereinfachung bietet Amazon für alle gängigen Betriebssysteme die sogenannte *Command Line Tools* an, die für alle verfügbaren Webservices direkt

<sup>3</sup> <http://aws.typepad.com/aws/2009/05/new-aws-load-balancing-automatic-scaling-and-cloud-monitoring-services.html>

ausführbare Befehle mitliefern. Alternativ wird mit ElasticFox eine Erweiterung für den Browser Mozilla Firefox angeboten, womit man alle Befehle direkt bedienen kann. Zusätzlich bietet dieses Plug-in immer die aktuellsten Neuerungen des Webservice-Befehlssatzes, wie benutzerdefinierte Tags für einzelne Serverinstanzen um eigene Instanzen markieren zu können [19].

Der einfachste Weg führt derzeit über die Amazon *Management Console*, welche fast alle Webservices abdeckt. Infolge des nun folgenden Arbeitsablaufes werden zusätzlich die Command Line Befehle genannt, um auch selbst geschriebene Skripte zu ermöglichen. Unterstützt wird man dabei von der sehr umfangreichen Dokumentation von Amazon [18].

## 4.2 Zugang

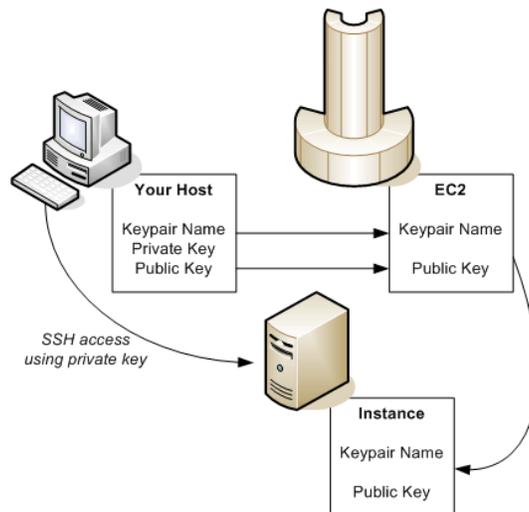


Abbildung 3. Amazon Schlüsselsystem

Zunächst benötigt man ein Konto, welches man unter der AWS-Startseite [20] unter Angabe einer Kreditkartennummer eröffnen kann. Über die AWS-Startseite hat man nun jederzeit und von überall Zugriff auf die umfangreichen Dokumentationsseiten, eine aktuelle Kostenübersicht mit Einzelabrechnung und die derzeit laufenden oder verfügbaren Services. Unter Security Credentials finden sich die für die spätere Bedienung notwendigen *Zugangsschlüssel und Zertifikate*. Der AccessKey (Abbildung 3<sup>4</sup>) ist der öffentliche Schlüssel der den Benutzeraccount eindeutig identifiziert und bildet mit dem geheimen Secret Access Key das

<sup>4</sup> <http://docs.amazonwebservices.com/AWSEC2/latest/DeveloperGuide/index.html?ami-from-existing-image.html>

asymmetrische Schlüsselpaar für spätere Verifikationen. Der private Schlüssel wird mit `ec2-add-keypair` erstellt und dient zum Login auf die einzelnen Rechnerinstanzen. Der weitere Arbeitsablauf zum Aufsetzen einer EC2 Instanz stellt sich

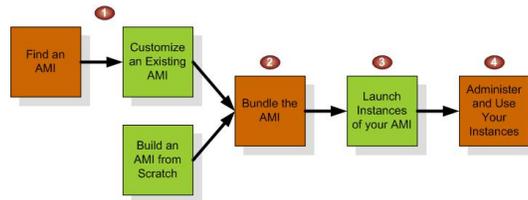


Abbildung 4. Amazon Arbeitsablauf beim Aufsetzen einer Instanz

wie in Abbildung 4<sup>5</sup> dar.

### 4.3 Regionen und Zonen

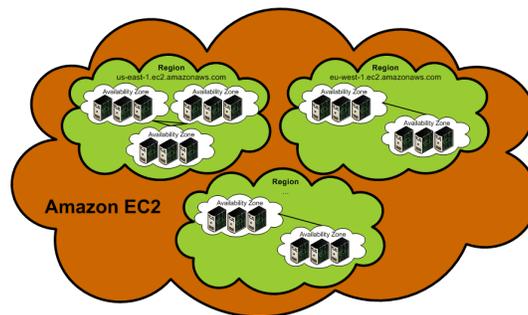


Abbildung 5. Regionenmodell

Um eine Instanz zu starten, muss man sich zunächst für eine Region entscheiden. Zurzeit bietet Amazon die Regionen US-East und EU-West an, eine weitere Region Asia-Pacific ist für Anfang 2010 geplant. Die gewünschte Region kann man mit `ec2-describe-regions` möglichst nah am späteren Endanwender platzieren. So ist ein in der EU-Region gestarteter Web-Server aus Deutschland deutlich schneller zu erreichen als aus der US Region, allerdings wird der Zugriff

<sup>5</sup> <http://docs.amazonwebservices.com/AWSEC2/latest/DeveloperGuide/index.html?concept-flow.html>

von einer Region in eine andere wie normaler Internettraffic abgerechnet. Jede Region wird in eigenständige Availability-Zonen aufgeteilt (Abbildung 5<sup>6</sup>). Diese sind räumlich und logisch isolierte ausfallsichere Rechenzentren. Somit kann man durch Aufteilung der Instanzen auf verschiedene Availability-Zonen innerhalb der gleichen Region partielle Ausfälle vermeiden.

#### 4.4 Amazon Machine Images

Hat man sich für eine Region und eine oder mehrere Availability-Zonen entschieden, ist der nächste Schritt sich über Launch Instances eine virtuelle Maschine einzurichten. Eine verfügbare virtuelle Maschine wird bei Amazon durch ein Amazon Machine Image (AMI) repräsentiert und ist ein in EC2 lauffähiges Image, welches aus einem Betriebssystem und vorinstallierter Software besteht. Mit verschiedenen Betriebssystemen wie Unix/Linux und Windows bietet Amazon eine umfangreiche Anwendungsgrundlage. Neben den direkt von Amazon bereitgestellten AMIs finden sich auch Community AMIs, welchen von anderen Anwendern und Firmen bereitgestellt werden. Da Amazon am Anfang fast vollständig auf Open Source Software und Betriebssysteme setzte, sind die Softwareanbieter nun nachgezogen und bieten verschieden integrierte Lösungen an. Es existieren vorlizenzierte Paid AMI Images für besondere Einsatzziele wie Datenbankserver von Oracle oder IBM. Diese werden direkt von den jeweiligen Unternehmen angeboten und werden nach eigenen Preismodellen abgerechnet. Im Gegensatz zu einer langwierigen Installation und Einrichtung eines eigenen Anwendungsservers fährt eine entsprechende Amazon Instanz in wenigen Minuten hoch und ist voll einsatzbereit. Viele große Softwareanbieter wie Oracle, IBM und Microsoft haben ihre bisherigen Lizenzmodelle angepasst und bieten ihre Produkte direkt in EC2 an. Durch kurzfristige Integration einer Testumgebung für die eigene Produktivumgebung bieten sich somit neue Timesaver und Anwendungsgebiete im Alltag eines Rechenzentrums. Durch Filterung nach Hersteller oder Betriebssystemplattform lässt sich die lange Liste auf die infrage kommenden Images reduzieren (`ec2-describe-images` o `Amazon | grep Fedora`). In der Weboberfläche bietet Amazon unter Quickstart eine Vorauswahl aller verfügbaren AMI.

Das so ausgewählte Image läuft auf einer von sieben verschiedenen Architekturumgebungen. Unterteilt in die drei Kategorien Standard, High-Memory und High-CPU gibt es jeweils mehrere Ausbaustufen, welche unterschiedliche Kosten verursachen. (Tabelle 1) Alle Instanzen mit mehr als 4 GB Arbeitsspeicher sind dabei 64-Bit Instanzen. Um laufende Kosten besser planen zu können, lassen sich auch bestimmte Instanzen auf für 6 oder 12 Monate fest reservieren. Mit einer solchen Reserved Instance sinkt der Preis pro Stunde. Sämtliche von Amazon angebotene Services wie Datenspeicher S3 oder Datentransfer werden durch ein getrenntes Preismodell abgerechnet.

<sup>6</sup> <http://docs.amazonwebservices.com/AWSEC2/latest/DeveloperGuide/index.html?concepts-regions-availability-zones.html>

**Tabelle 1.** Verschiedene Instanzklassen und das dazugehörige Preismodell

Instanz	ECU	Ram/HDD(GB)	Cent/Stunde
Small	1	1.7 / 160	8.5
Large	4	7.5 / 850	34
Extralarge	8	15 / 1690	68
High Memory Double	13	34 / 850	120
High Memory Quadruple	26	68 / 1690	240
High CPU Medium	5	1.7 / 350	17
High CPU Extra Large	20	7 / 1690	68

Für einen ersten Testserver bietet sich ein Standard Fedora 32-Bit Web-Server an. Für das KeyPair und die Firewall Security Groups können vorerst die empfohlenen Einstellungen gewählt werden. Eine Security Group besteht aus verschiedenen einander ergänzenden Regelsätze und ist vergleichbar mit einzelnen Regeln einer IP-Tables Firewall. Neben den wichtigsten Regelsätzen zu SSH, RDP, HTTP und FTP kann man jederzeit die Portbereiche für eigene Anwendungen ergänzen. Es lassen sich pro Instanz mehrere Gruppen zuweisen und somit erweitern. Die voreingestellte Gruppe blockt außer gruppeninternen alle externen Verbindungen. Der Assistent macht je nach gewählter Instanz sinnvolle Vorschläge. Die Firewall Einstellungen lassen sich nachträglich durch Änderung der Security Groups beeinflussen, diese gelten dann aber für alle damit verbundenen Instanzen (`ec2-authorize WEBSERVERGROUP p 80`). Die Startzeit der Instanz dauert je nach gewünschtem Image und Speicherplatzbedarf einige Sekunden bis zu wenige Minuten. Da einmal gestartete Instanzen von nun an lokal in der Region vorliegen wird ein zweiter oder erneuter Start deutlich schneller vonstattengehen (`ec2-run-instances INSTANCE`). Für Diagnose und Statusmeldungen bietet Amazon über das integrierte Console-Log die letzten Fehlermeldungen der laufenden Instanz. Bei Linux Instanzen zeigt diese die Ausgabe der Systemlogdatei `/var/log/messages`, unter Windows die letzten 3 Einträge des Event-Log.

Nun kann man sich auf die gestarteten Server mit Secure Shell (SSH) oder Windows Remote Desktop (RDP) einloggen. Da die bereitgestellten AMIs aus Sicherheitsgründen nicht über Standardpasswörter verfügen, ist die Verwendung der KeyPairs erforderlich. Diese KeyPairs werden beim Starten der ersten Instanz automatisch generiert und müssen bei der Verbindung angegeben werden. Zusätzliche Schlüssel können in der Weboberfläche unter KeyPairs erstellt und zugewiesen werden. Für die bekannte Secureshell-Anwendung Putty ist eine Umwandlung der Schlüssel erforderlich, das hierfür erforderliche Programm Putty-Gen finden sich unter [21]. Bei einer Windows Instanz wird beim ersten Start ein zufälliges Administrator-Passwort generiert, welches unter Angabe des privaten Schlüssels decodieren muss.

## 4.5 Speichervarianten

Nun ist es möglich den eigenen Server anzupassen, indem man zusätzliche Serverdienste installiert und vorhandene Services einrichtet. Über die Weboberfläche oder mittels `ec2-terminate-instances` kann man die eigenen Instanzen auch jederzeit endgültig heruntergefahren und damit komplett löschen. Dabei ist zu beachten, dass der interne Speicher (Instance Storage) einer AMI mit der gesamten Instanz beim herunterfahren sämtliche Daten verliert. Demnach ist einer der ersten Schritte zum sicheren persistenten Absicherung der eigenen Server der Einsatz von zusätzlichen Speicherlösungen von Amazon. Im Gegensatz zum internen Speicher einer Instanz sind Daten im Elastic-Block-Storage (EBS) oder Simple-Storage-Service (S3) nicht nach dem Herunterfahren einer Instanz verloren. Hierfür kann man EBS in beliebiger Größe von einigen GB bis mehrere TB als virtuelle Netzlaufwerke anbinden. Die Preise sind nach Durchsatz in GB

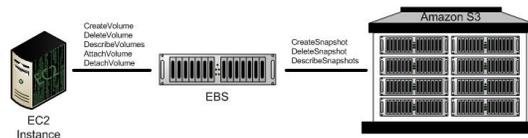


Abbildung 6. Elastic Block Storage

gestaffelt und können auch jederzeit wieder gelöscht werden. Wichtig ist darauf zu achten, dass man sich in der gleichen Region und Availability-Zone wie die Instanz befindet. Somit bietet sich EBS für die Installation von eigener Software oder Ablegen von Geschäftsdaten, Logdateien und komplette CD/DVD-Images an (Abbildung 6<sup>7</sup>). Es lassen sich auch inkrementelle Datensicherungen, sogenannte Snapshots von laufenden Instanzen für den späteren Gebrauch erstellen.

Nachdem man eine Instanz auf die eigenen Bedürfnisse angepasst hat, kann man diese in Zukunft entweder erneut in einer schnelleren Instanz oder auch mehrfach als Cluster starten lassen. Für diesen Zweck lassen sich eigene AMIs erstellen. Hierfür unterscheidet sich das AMI Bundle Verfahren für Windows und Linux und man benötigt das AMI Toolkit von Amazon, welches mit einer Schritt für Schritt Anleitung hier angeboten wird [18]. Das Verfahren benötigt je nach Größe des Servers etwa 10 bis 30 Minuten und das erstellte AMI wird in einen Speicherplatz bei Amazon S3 hochgeladen. Alternativ lässt sich auch ohne Instanz von Amazon eine AMI aus einem eigenen System bauen. Die Prozedur ist jedoch ungleich komplizierter, bietet dem Anwender allerdings mehr Freiheiten. Eine erstellte AMI ist vor dem ersten Gebrauch mit `ec2-register-s3bucket/image.manifest.xml` zu registrieren und man hat dabei die Wahl, ob

<sup>7</sup> <http://docs.amazonwebservices.com/AWSEC2/latest/UserGuide/index.html?ebs-api-overview.html>

diese nur privat oder auch für andere AWS Anwender nutzbar sein sollte. Durch ein Rechtemanagement kann man die eigenen AMIs auch für nur bestimmte Benutzer freigeben. Weiterhin kann man diese AMI auch kostenpflichtig anbieten, sodass sie in Zukunft bei Fremdzugriff pro Stunde am Umsatz beteiligt werden.

#### 4.6 Netzwerkservices

Jede gestartete Instanz besitzt zunächst einen privaten DNS Eintrag. Da es sich auf Dauer schlecht mit den langen komplizierten DNS-Adressen arbeiten lässt, kann man sich von Amazon eine einfachere IP-Adresse zuweisen lassen. Diese *Elastic IP* versteht sich als statische IP-Adressen, welche man dynamisch den eigenen Instanzen zuweisen kann. In Abbildung 7<sup>8</sup> sind die Web-Server durch

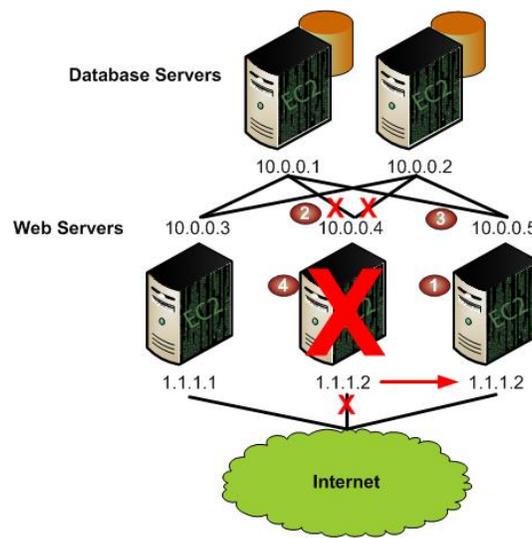


Abbildung 7. Elastic IP Address

elastische IP Adressen von Außen und die Datenbankserver nur durch private IP Adressen intern erreichbar. Falls eine Maschine durch einen Defekt oder Wartung ausfällt kann der Administrator die entsprechende Elastic IP auf einen anderen oder neuen Server übertragen. Somit hat der Benutzer transparenten Zugriff auf alle verfügbaren Dienste. Die IP-Adressen Knappheit der heutigen Zeit wird durch ein ungewöhnliches Kostenmodell berücksichtigt: Es werden nur beantragte, aber nicht zugewiesene IP-Adressen in Rechnung gestellt.

<sup>8</sup> <http://docs.amazonwebservices.com/AWSEC2/latest/DeveloperGuide/index.html?concepts-elastic-addressing.html>

Durch Überwachung der laufenden Instanzen kann der Administrator Informationen sammeln um die Entscheidung einer Hoch- oder Runterskalierung der eigenen Services zu begründen. Die Amazon Dienstleistungen CloudWatch und AutoScale ermöglichen ein exaktes Anpassen der Anwendung an die eigenen Anforderungen. Hierzu sammelt *CloudWatch* Rohdaten wie CPU Last, I/O-

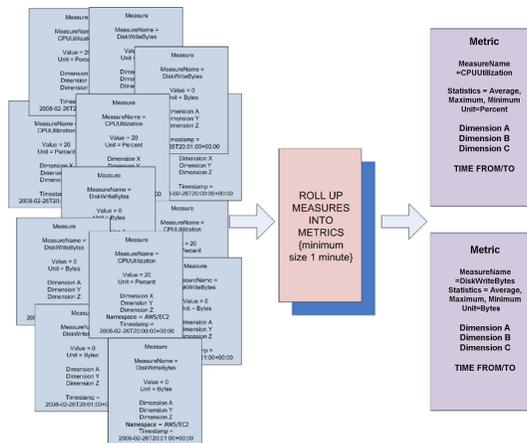


Abbildung 8. CloudWatch erstellt Metriken

Durchsatz von Festplatten und Netzwerkzugriffen und berechnet daraus in Echtzeit Metriken zum Zustand und Leistung der Instanzen (Abbildung 8<sup>9</sup>). Diese Metriken kann sich der Administrator in einfachen Diagrammen anzeigen lassen (Abbildung 9). Über die Webconsole bietet Amazon eine grundlegende Auswahl der wichtigsten Merkmale, wobei über die EC2 API eigene Abfragen definiert werden können. Damit liefert CloudWatch für die dynamische *Autoskalierung* die notwendigen Parameter. Cloud-Computing bietet im Vergleich zu herkömmliche Rechenzentren eine sehr kurzfristige Skalierung der eingesetzten Kapazitäten. Zurzeit erfolgt bei Amazon die Anpassung ausschließlich durch das Hinzufügen oder Entfernen von einzelnen Instanzen. Eine Beeinflussung der Performance während der Laufzeit durch Erhöhung der CPU-Leistung oder Erweiterung des Arbeitsspeichers ist nicht vorgesehen. Dieser Scale-Out Ansatz ist für Webanwendungen mit vielen gleichzeitigen Threads / Verbindungen geeignet.

Bei der dynamischen Skalierung unterscheidet man zwischen pro- und re-aktiver Skalierung. Die proaktive Skalierung veranlasst durch vordefinierte Regeln eine Anpassung der Instanzen bei zeitlich regelmäßigen Schwankungen. So ist eine Hochskalierung eines Webshops zu umsatzstarken Zeiten im Voraus durch entsprechende Skripte planbar. Amazon Web Services bietet durch die Metriken von

<sup>9</sup> <http://docs.amazonwebservices.com/AmazonCloudWatch/latest/DeveloperGuide/index.html?SvcIntro.html>

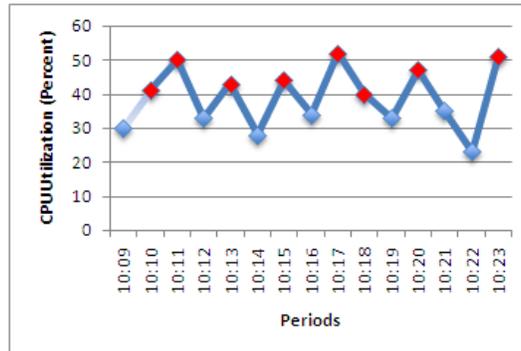


Abbildung 9. CloudWatch Diagramm in der Management-Console

CloudWatch zusätzlich reaktives Scaling (AutoScale) an. Sobald CloudWatch eine Serverauslastung registriert, kann dieser durch eine schnellere Instanz ersetzt oder mehr Speicher aus EBS / S3 zugesprochen werden [2]. Dadurch kann der Administrator mit EC2 einen deutlich geringeren Overhead der einzelnen Komponenten wie in herkömmlichen Rechenzentren erreichen. Eine einzelne AutoScale Funktion wird als Trigger bezeichnet und bestimmt aufgrund des auftretenden Zeitraumes (Periode) und Dauer des Durchbruchs (breachDuration) eine auszuführende Aktion. So kann bei einem kompletten Ausfall eines Servers dieser

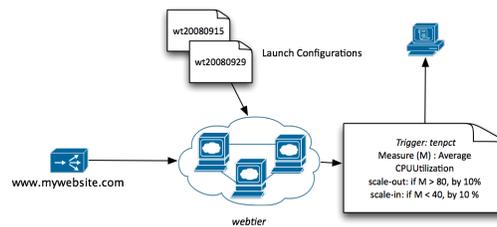
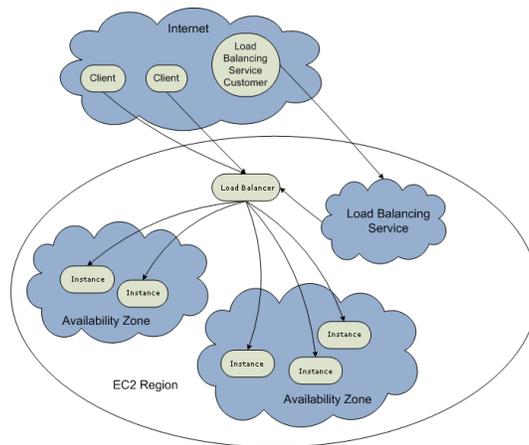


Abbildung 10. AutoScale Ablauf mit verschiedenen Startkonfigurationen

automatisch neu gestartet werden (Failover) oder bei hoher Auslastung der eingesetzten Ressourcen eine beliebige Anzahl neuer Instanzen hinzugefügt werden. Die eigene Infrastruktur lässt sich zu verschiedenen AutoScale Klassen gruppieren. Von AutoScale gestartete Instanzen können durch Parameter an bestimmte Startkonfigurationen wie eine Datenbankumgebung oder ein VPN-Netzwerk angepasst werden (Abbildung 10). Durch die schnelle Initialisierung ist auch ein vollständiges Anhalten von unbenötigten Instanzen vorstellbar, somit können wenig genutzte Ressourcen aus Kostengründen einfach heruntergefahren werden.

Eine umfangreichere Ausfallsicherung bietet Amazon mit zusätzlichen Infrastrukturservices wie Load-Balancer und virtuellen Netzwerken. Ein *Elastic-Load-Balancer* (ELB) funktioniert wie ein klassischer Netzwerk Load-Balancer indem er alle Benutzeranfragen auf verschiedene Server verteilt um eine möglichst gleichmäßige Auslastung zu erreichen und Spitzenlasten zu vermeiden (Abbildung 11<sup>10</sup>). Allerdings prüft ein Amazon ELB die einzelnen Instanzen aufgrund



**Abbildung 11.** Elastic Load Balancer verteilt Last in alle Zonen

der von CloudWatch aufbereiteten Metriken und verteilt die Netzwerklast entsprechend auf die Server innerhalb einer Region. Somit wird jede Zone innerhalb der Regionen gleich belastet. Zudem kann er bei Bedarf mittels Autoscaling auch neue Instanzen starten oder regruppieren.

Um die in EC2 gestarteten Instanzen schließlich transparent in die eigene Firmeninfrastruktur zu integrieren, bietet Amazon den Service *Virtual Private Cloud* an. Instanzen in einer eingerichteten VPC-Gruppe besitzen einen eigenen privaten IP-Adressraum und werden über einen VPN-Tunnel an das firmeneigene VPN Gateway angebunden (Abbildung 12<sup>11</sup>). Über diesen tauchen sie anschließend transparent im eigenen Netzwerk auf. Dabei ist neben den Kosten durch zusätzlichen Traffic zwischen Firmennetzwerk und Amazon aber auch die komplexe Einrichtung des VPN-Gateways zu beachten.

<sup>10</sup> <http://docs.amazonwebservices.com/ElasticLoadBalancing/latest/DeveloperGuide/index.html?SvcIntro.html>

<sup>11</sup> <http://docs.amazonwebservices.com/AmazonVPC/latest/GettingStartedGuide/index.html?BriefIntroduction.html>

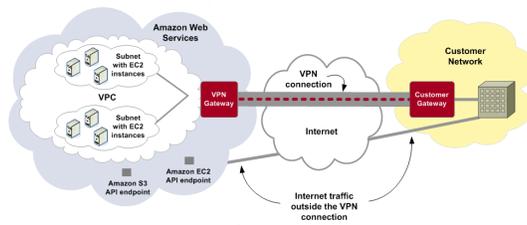


Abbildung 12. Anbindung einer Virtual Private Cloud

## 5 Risiken

Die allgemeinen Risiken im Bereich Sicherheit, Performance und Zukunftssicherheit von Cloud-Computing finden sich auch bei EC2 wieder. Die wichtigsten Kriterien dürften die Sicherheit der Daten sowie die Verfügbarkeit der Anwendung, des Netzwerk und des ganzen Systems sein.

*Daten* wie Mailkonten, Geschäftszahlen oder Benutzerangaben gelten als der wichtigste Teil eines Unternehmens. Im Zuge einer Migration zu EC2 werden diese aus der eigenen Hand gegeben und auf Servern eines anderen Unternehmens gelagert. Dieser Vorgang wird entweder von gesetzlichen Einschränkungen oder Unternehmensgrundsätzen sehr erschwert. Spätestens wenn die Daten einem fremden Unternehmen im Ausland anvertraut werden, in dem andere Datenschutzbestimmungen oder Gesetze wie zum Beispiel der amerikanische Patriot Act gelten, werden die Entscheidungsträger im Unternehmen Einwände haben. Wie würde sich das deutsche Amazon verhalten, wenn die amerikanische Regierung die Herausgabe bestimmter Daten der amerikanischen Muttergesellschaft Amazon verlangt? Allerdings ist sich Amazon dieser Problematik bewusst und bietet getrennte europäischen Rechenzentren. Außerdem empfiehlt sich die grundsätzliche Verschlüsselung der Daten für die Übertragung und Lagerung in der Cloud. Gängige Verschlüsselungsalgorithmen wie RSA oder AES bieten dabei hohe Sicherheit und geringen Performanceverlust.

Größer ist jedoch die Gefahr durch eine kompromittierte AMI Daten zu verlieren oder Einbruchstellen zu bieten. Da ein Update und das anschließende Neubündelung der entsprechenden Instanzen zeitaufwendig ist, wird es oftmals unterlassen. Ebenso ist das Abspeichern von Anmeldedaten für einen VPN-Tunnel oder sonstigen Passwörtern in einer selbst hergestellten AMI gefährlich. Eine umfangreiche Überlegung zum Thema Datensicherheit bei EC2 findet sich unter [22].

Durch die Veröffentlichung allgemein verbindlicher *Service Level Agreements* (SLA) seitens Amazon scheint vorerst die Verfügbarkeit des Gesamtsystems gegeben. Allerdings ist die Wirksamkeit einer SLA grundlegend von den vereinbarten Vertragsstrafen abhängig. Amazon bietet beim Bruch der Vertragsbedingungen einen 10-prozentigen Rabatt auf die Rechnung des letzten Jahres.

Im Fall einer Störung wird dieser Betrag allerdings in den wenigstens Fällen helfen können. Außerdem bleibt die Frage nach einem direkten Kontakt im Notfall offen. Durch aktuelle Datenverlustvorfälle wie von Microsoft Sidekick wird die Brisanz eines solchen Ausfalls deutlich: Bei einem schweren Fehler im System sind die Daten meistens unwiderruflich verloren [23]. Durch die gemeinsame Nutzung der Amazon Ressourcen mit tausend anderen Benutzern treten aber auch neue Probleme auf, wie ein weiterer Fall aus dem Jahr 2008 verdeutlicht. Bei der Aktualisierung einer internationalen Email-Spam und Malware Sperrliste wurden fälschlicherweise viele Amazon IP-Adressen gesperrt, sodass die Cloud-Anwendungen dieser Benutzer nicht mehr verfügbar waren [24].

Auch die Migration der Anwendungen in die Cloud kann *Softwareentwickler* vor neue Aufgaben stellen: Durch die nicht standardisierte API besteht die Gefahr eines Lock-In. Durch die unterschiedlichen Umgebungen lässt sich derzeit eine Anwendung oder Instanz nur sehr schwer von einem Cloud-Computing Anbieter zu einem anderen umziehen. Dieses Problem hat IBM erkannt und entwickelt eine Standard API für die anbieterübergreifenden Cloudsteuerung. Es bleibt abzuwarten, ob die verschiedenen Cloudanbieter diesen Vorschlag unterstützen oder sich isolieren. Als Anbieter sind große virtualisierte Systeme schwer zu testen, deshalb baut Amazon die Testbarkeit des Systems direkt als Attribut in das Produktivsystem ein und kann so durch entsprechend gesetzte Parameter Aufrufe in einen virtuellen Testaufbau umleiten. Bei Software die nicht in EC2 integriert angeboten wird hat man gegebenenfalls ein Lizenzproblem. Durch eine automatische Skalierung oder Umzug einer Instanz kann eine Anwendung ihre gültige Lizenz verlieren. Es ist aber durch die paid-AMIs absehbar, dass immer mehr Anbieter ihr Lizenzmodell anpassen.

Viele *Performanceprobleme* der Cloud lassen sich durch AutoScale lösen. Allerdings sind bis heute nicht alle Rechnerkomponenten so skalierbar wie Prozessoren und Arbeitsspeicher. Gerade bei Festplatten- oder Netzwerkoperationen reicht ein Hochskalieren nicht aus. Zudem werden diese Ressourcen von mehreren virtuellen Instanzen geteilt und können bei starker Inanspruchnahme zur Geschwindigkeitseinbußen führen. Im Bereich der Serverhardware wird immer mehr im Bereich von Skalierung und Parallelisierung erforscht und entwickelt. Die Ergebnisse die bei Multicore-Prozessoren in den letzten Jahren erreicht wurden, können auch bei anderen Komponenten wie Datenspeicher erwartet werden. Hier ist der aktuell entwickelte Flash-Speicher ein erster Schritt.

Ein Problem aus technischer Sicht ist der anfallende *Datentransfer* zwischen dem eigenen Unternehmen und der Cloud. Gerade bei MapReduce-Prozessen ist eine Größenordnung von mehreren Terabyte (TB) für eine Anwendung keine Seltenheit. Einer Untersuchung des Berkley RAD Institutes zufolge benötigt man für eine Datenübertragung von 10 TB zu Amazon S3 knapp 45 Tage und verursacht dabei über 1000 US-Dollar Kosten. Aber auch durch Neuerungen wie proprietäre Protokolle wie Aspera, die den Overhead herkömmlicher Verbindungen reduzieren und die Datenübertragungen im besten Fall verdreifachen, können nicht mit dem einfachen versenden von Festplatten mittels Paketunternehmen mit-

halten [25]. Der Windows Datenbankentwickler Jim Gray formulierte bereits vor einigen Jahren die Vermutung, dass auch zukünftig keine Datenübertragung mit einer Lkw-Ladung voller Festplatten mithalten kann und dieses Prinzip wird angesichts der schnellen Zunahme der Festplattendichten auch bei steigenden Datenübertragungsraten weiterhin gelten [26]. Für diesen Vorgang bietet Amazon eine eigene Dienstleistung an, der eingeschickte Festplatten einliest und in S3 abspeichert. Dieser Vorgang dauert für 1TB nur 1 Tag und kostet mit Versand und Einlesen nur 400 US-Dollar [8]. Eine alternative Möglichkeit für manche Projekte können die Public Data Sets über EBS Snapshots darstellen. Bereits vorhandene umfangreiche Datensätze in den Fachrichtungen Biologie, Chemie, Ökonomie und Wiki-Repositories sind unentgeltlich zugänglich und können weiterverarbeitet werden [27].

## 6 Die Konkurrenz im Vergleich

Im Vergleich zu alternativen Angeboten ist EC2 erst einmal gegen *herkömmliche Serverhosting* Angebote abzugrenzen. Im Test der Zeitschrift C'T wurden Rootrechner von deutschen Anbietern in Verfügbarkeit und Ladelastverteilung verglichen [28]. Laut diesem Test war der 1und1 Server innerhalb eines Monats vier Mal nicht erreichbar und die Ladezeit wurde in 15 Prozent der Fälle mit über 300ms als schlecht eingeschätzt. Billigere Anbieter hatten sogar bis zu 16-mal Ausfälle und vergleichbare Ladezeiten. Im Gegensatz dazu war EC2 im gleichen Zeitraum voll verfügbar [29]. Es wurde am 14. Oktober 2009 eine Störung der API in der EU-Region gemeldet, welche nach 37 Minuten von Amazon untersucht und behoben wurde. Auch die Untersuchung des Berkley RAD Institutes bestätigt die hohe Verfügbarkeit [8]. Damit steht EC2 als Cloudanbieter im Vergleich zu herkömmlichen Hostern sehr gut dar.

Zum Vergleich zu anderen *Cloud-Computing Anbietern* wurden verschiedene Anbieter auf der gleichen IaaS-Stufe untersucht. Viele andere Anbieter von Services auf Ebenen wie PAAS oder SAAS brauchten nicht betrachtet zu werden, da sich diese durch ihr völlig anderes Servicekonzept gegenseitig wenig Konkurrenz machen. Einer der Konkurrenten ist GoGrid. Dieser verfolgt einen ähnlichen Ansatz wie AWS und erlaubt dabei eine freie Zuweisung von Ressourcen innerhalb einer Instanz. Damit sind Größen wie Prozessorleistung, Arbeitsspeicher und Speicherplatz frei wählbar. Viele der heute in EC2 verfügbaren Möglichkeiten wie eine klar definierte SLA, die Webconsole oder VPC wurden zuerst von GoGrid eingeführt. Das Herstellen eigener AMIs ist bislang nicht möglich, sodass ein vollautomatisches hoch und runterskalieren der eingesetzten Komponenten nur unzureichend gelöst ist. Der Trendanalyse von Google zufolge erreicht GoGrid deutlicher weniger die Aufmerksamkeit der Anwender [30]. Weiter als GoGrid geht das junge Unternehmen Mosso, welches bei seiner PaaS-Lösung weder Root-Zugriff noch eine umfangreiche API bietet. Statt dessen versucht es durch fertige Hostingpakete die Vorteile des CC zu nutzen und die Skalierungsaufgaben selbst zu übernehmen. Somit bedient es sich IaaS und verbindet es mit administrativen

Aufgaben zu einer sehr wartungsfreien Gesamtlösung. Andere Cloud-Computing und Storage-Anbieter wie Rackspace, Joyent oder Flexiscale bieten weniger Optionen oder Features. Amazon bietet damit das derzeit umfangreichste Angebot an Cloud-Computing basierten Dienstleistungen [31].

Als einer der Vorreiter für einheitliche Cloud-Computing Umgebungen setzt IBM auf offene Standards sowie eine einheitliche API. IBM baute wegen rechtlichen Bedenken und vielen vertraulichen Daten eine private Cloud auf, um für eigene Projekte dynamische Teamumgebungen schaffen zu können. Die damit entstandene IBM BlueCloud ist bisher nicht für den freien Markt zugänglich, sondern bietet nur Anwendungspakete für eine geschlossene Benutzergruppe. Auch der große Betriebssystemhersteller Microsoft bietet mit Azure eine PaaS-Cloudlösung und damit eine Plattform für fertige Datenbankdienste und eine vollständige .NET-Framework Laufzeitumgebung an. Allerdings kann man Azure nicht auf eine bestehende IT-Infrastruktur aufsetzen, sondern ist auf ein von DELL geliefertes Gesamtpaket angewiesen. Damit entspricht Azure eher einer ganzheitlichen Umstrukturierung der eigenen IT-Struktur zu einer Private Cloud mit erheblichen Investitionskosten.

HP verfolgt mit den HP Pods die Idee der IT-Industrialisierung für den modularen Aufbau einer eigenen private Cloud-Lösung. Ein Pod ist ein stapelbarer Industriecontainer die der Kunde zur Skalierung der eigenen Infrastruktur nutzen und kombinieren kann. Diese ermöglichen je nach Variante mit über 10.000 CPU-Kernen oder 9 PB Speichervolumen eine sehr hohe Kompressionsrate. Zusätzlich will HP in Zukunft mit HP Dynamic Cloud auch Services anbieten.

## 7 Zusammenfassung und Ausblick

Durch die geringen Einstiegshürden erreichen auch Anfänger mit EC2 ohne umfangreiche Kenntnisse schnelle und robuste Lösungen. Durch die einfache Webconsole ist ein barrierefreier Zugang ohne vorheriges lesen der Dokumentation möglich. Erste Ergebnisse können mit der nahtlosen Integration von S3 und VPC früh in die eigene IT-Infrastruktur eingebunden werden. Infolge der voll automatisierbaren, schnell replizierenden und extrem hochverfügbaren Lösung lassen sich sehr schnell eigene Projekte migrieren, um damit Erfahrungen zu sammeln. Gerade die von Amazon veröffentlichte SLA dürfte sehr wirksam in der Überzeugung von Entscheidungsträgern in der eigenen Firma sein. Neben den Vorteilen wie Performance, Ausfallsicherheit und Kosten dürfen allerdings kritische Punkte wie Datensicherheit oder Übertragungsraten nicht vernachlässigt werden. Insbesondere junge Unternehmen werden durch die neue Art des Serverhostings sehr profitieren, aber auch bereits etablierte Unternehmen können Vorteile der Cloud nutzen.

## Literatur

1. Gartner Identifies Top Ten Disruptive Technologies for 2008 to 2012.  
<http://www.gartner.com/it/page.jsp?id=681107>
2. Reese, George. Cloud Application Architectures. O'Reilly.2009.
3. Larry Ellison. Oracle Analyst Day.  
<http://www.cloudave.com/link/oracle-cloudy-cloud-computing-plans-saas>
4. Larry Ellison im Videointerview über Cloud-Computing.  
<http://www.techfieber.de/2009/10/02/video-larry-lastert-warum-oracle-ceo-larry-ellison-cloud-computing-hasst/>
5. Amazon EC2 Resource Usage 2009.  
<http://www.jackofallclouds.com/2009/09/anatomy-of-an-amazon-ec2-resource-id/>
6. National Institute of Standards.  
<http://csrc.nist.gov/groups/SNS/cloud-computing/index.html>
7. Amazon Webservices Simple Monthly Calculator.  
<http://calculator.s3.amazonaws.com/calc5.html>
8. Above the Clouds: A Berkeley View of Cloud Computing.  
<http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.pdf>
9. Potenzialabschätzung von Rechenzentren.  
<http://www.energieeffizienz-im-service.de/rechenzentren/online-ratgeber/potenzialabschaetzung.html>
10. Single Point of Failure beherrschen.Computerwoche.  
<http://www.computerwoche.de/subnet/hp-intel/1889791/>
11. UC Berkeley. Electrical engineering and computer sciences (EECS).  
<http://www.eecs.berkeley.edu/department/>
12. CERT. Denial of Service Attacks.  
[http://www.cert.org/tech\\_tips/denial\\_of\\_service.html](http://www.cert.org/tech_tips/denial_of_service.html)
13. wPrime benchmarking application homepage.  
<http://www.wprime.net>
14. Schwarz, Christian. Cloud-Computing Seminar. Universität Heidelberg. 2009.  
[http://pvs.informatik.uni-heidelberg.de/Teaching/CLCP-09/CLCP\\_SS2009\\_Christian\\_Schwarz\\_EC2.pdf](http://pvs.informatik.uni-heidelberg.de/Teaching/CLCP-09/CLCP_SS2009_Christian_Schwarz_EC2.pdf)
15. Large Hadron Collider.  
<http://lcg.web.cern.ch/LCG/>
16. New York Times Archives Project.  
<http://open.blogs.nytimes.com/2008/05/21/the-new-york-times-archives-amazon-web-services-timesmachine/>
17. Amazon Webservices Case-Studies.  
<http://aws.amazon.com/solutions/case-studies/>
18. Amazon Webservices Technical Documentation.  
<http://aws.amazon.com/documentation/>
19. Mozilla Firefox Plugin Elasticfox.  
<http://sourceforge.net/projects/elasticfox/>
20. Amazon Webservices.  
<http://aws.amazon.com/>
21. PuTTY Secure Shell- und Telnet-Programm.  
<http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>
22. Überlegungen zu Amazon EC2 Instanzen als Ziel von Angriffen.  
<http://bitmuncher.wordpress.com/2008/09/15/uberlegungen-zu-amazon-ec2-instanzen-als-ziel-von-angriffen/>

23. Sidekick-Datenverlust wirft Schatten auf die Cloud.  
<http://www.heise.de/mobil/meldung/Sidekick-Datenverlust-wirft-Schatten-auf-die-Cloud-821328.html>
24. Amazon: Hey Spammers, Get Off My Cloud.  
[http://blog.washingtonpost.com/securityfix/2008/07/amazon\\_hey\\_spammers\\_get\\_off\\_my.html](http://blog.washingtonpost.com/securityfix/2008/07/amazon_hey_spammers_get_off_my.html)
25. Aspera stellt neues EC2 Datenübertragungsprotokoll vor.  
<http://gigaom.com/2009/09/10/aspera-bypasses-bandwidth-bottlenecks-to-ec2/>
26. Jim Gray über Datenübertragungsraten.  
<http://queue.acm.org/detail.cfm?id=864078>
27. Amazon Public Data Sets.  
<http://aws.amazon.com/publicdatasets/1>
28. CT, Ausgabe 23/2007, Seite 168.
29. Amazon Webservices Service Health Dashboard.  
<http://status.aws.amazon.com/>
30. Google Trendanalyse GoGrid - EC2.  
<http://www.google.de/trends?q=gogrid%2C+amazon+ec2ctab=0geo=alldate=allsort=0>
31. Comparing the Cloud: EC2, Mosso and GoGrid.  
<http://www.infoq.com/articles/cloud-comparison>

# Anhang

## Untersuchungen zur Performance von Amazon Webservices EC2

© Peter Gehrt - Stand: Oktober 2009

SYSTEM	kcbench (s)	hdparm (MB/s)	ping (s)	iperf (Mbit/s)	wprime (s)
Referenzserver			20	80	
EC2small	604	300	40	52	170
EC2Quadruple	33	800	40	90	7
AEC2 intern			1	936	

### Getestete Systeme:

EC2Quadruple:	8x X5550 @2.7ghz (8 cores) mit 68 GB Ram
EC2small:	Xeon E5430 @ 2.7 Ghz (Cores) mit 1.7 GB Ram
Referenzserver	Core2Duo E6400 @2.2Ghz mit 4GB Ram

### I/O

kcbench: EC2Quadruple	33s
kcbench: EC2small	604s
hdparm -tT EC2Quadruple:	800 MB/s
hdparm -tT EC2small:	300 MB/s

### Network

ping Universität > EC2small:	32 ms
ping DSL > EC2small:	38 - 48 ms
ping EC2small > EC2small:	0.5 ms
ping DSL > Universität:	19 - 35 ms

iperf Universität > EC2small:	18- 53 Mbit/s
iperf Universität > EC2Quadr	80 - 90 Mbit/s
iperf ec2 > ec2:	350 Mbit/s
iperf ec2 > ec2:	936 Mbit/s

### CPU

<http://www.wprime.net/?q=leaderboard>

wprime EC2Quadruple:	7s	entspricht laut Bestenliste einem Core i7 920 @ 3.6Ghz
	50 s	mit nur einem Thread
wprime EC2small:	170s	entspricht laut Bestenliste einem Athlon 64 @ 2Ghz

<http://www.winfuture-forum.de/index.php?showtopic=50598>

superpi 2M EC2Quadruple	34s	entspricht Core2Duo @ 2.6Ghz
superpi 2M EC2small:	208s	entspricht Athlon XP @ 2Ghz

iperf -s @ Small Instanz

-----  
Server listening on TCP port 5001  
TCP window size: 85.3 KByte (default)  
-----

```
[ 4] local 10.224.94.XXX port 5001 connected with 10.224.99.XXX port 42759
[ 4] 0.0-10.1 sec 392 MBytes 326 Mbits/sec
[ 5] local 10.224.94.XXX port 5001 connected with 10.224.99.XXX port 42760
[ 5] 0.0-10.0 sec 338 MBytes 282 Mbits/sec
[ 4] local 10.224.94.XXX port 5001 connected with 10.224.99.XXX port 42761
[ 4] 0.0-10.1 sec 357 MBytes 296 Mbits/sec
[ 5] local 10.224.94.XXX port 5001 connected with 10.224.99.XXX port 42762
[ 5] 0.0-10.0 sec 309 MBytes 258 Mbits/sec
```

iperf -s @ Quadruple Instanz

```
-----
Server listening on TCP port 5001
TCP window size: 85.3 KByte (default)
-----
```

```
[ 4] local 10.230.7.XXX port 5001 connected with 10.230.9.XXX port 49828
[ 4] 0.0-10.0 sec 1.09 GBytes 935 Mbits/sec
[ 5] local 10.230.7.XXX port 5001 connected with 10.230.9.XXX port 49829
[ 5] 0.0-10.0 sec 1.06 GBytes 910 Mbits/sec
[ 4] local 10.230.7.XXX port 5001 connected with 10.230.9.XXX port 49830
[ 4] 0.0-10.0 sec 1.09 GBytes 934 Mbits/sec
[ 5] local 10.230.7.XXX port 5001 connected with 10.230.9.XXX port 49831
[ 5] 0.0-10.0 sec 1.09 GBytes 936 Mbits/sec
```

hdparm /dev/sdb @ Quadruple Instanz

```
Timing cached reads: 16958 MB in 1.99 seconds = 8528.00 MB/sec
Timing buffered disk reads: 2434 MB in 3.03 seconds = 804.01 MB/sec
/dev/sdb:
Timing cached reads: 16884 MB in 1.99 seconds = 8489.35 MB/sec
Timing buffered disk reads: 2562 MB in 3.02 seconds = 847.93 MB/sec
```

hdparm /dev/sda2 @ Small Instanz

```
Timing cached reads: 4318 MB in 2.04 seconds = 2114.70 MB/sec
Timing buffered disk reads: 918 MB in 3.00 seconds = 305.77 MB/sec
/dev/sda2:
Timing cached reads: 4208 MB in 2.04 seconds = 2064.30 MB/sec
Timing buffered disk reads: 998 MB in 3.01 seconds = 331.86 MB/sec
```

I/O Durchsatztests mit bonnie++ @ Reverenzserver

```
Version 1.03c -----Sequential Output----- --Sequential Input- --Random-
-Per Chr- --Block-- -Rewrite- -Per Chr- --Block-- --Seeks--
Machine Size K/sec %CP K/sec %CP K/sec %CP K/sec %CP K/sec %CP /sec %CP
vsrv1 8000M 40946 81 43518 11 20438 5 38331 71 54691 8 229.4 0
-----Sequential Create----- -----Random Create-----Create-- --Read--- -Delete--
-Create-- --Read---Delete--
files /sec %CP /sec %CP /sec %CP /sec %CP /sec %CP /sec %CP
16 ++++++ +++ ++++++ +++ ++++++ +++ ++++++ +++ ++++++ +++ ++++++ +++
8000M,40946,81,43518,11,20438,5,38331,71,54691,8,229.4,0,16,+++++,+++,+++++,+++,+++++,+++,+++++
```

I/O Durchsatztests mit bonnie++ @ Small Instanz

```
Version 1.03 -Sequential Output-- --Sequential Input- --Random--Per Chr- --Block-- -Rewrite- -Per
```

```

Chr- --Block-- --Seeks--
Machine      Size K/sec %CP K/sec %CP K/sec %CP K/sec %CP K/sec %CP /sec %CP
ip-10-224-111 3408M 24869 37 67693 8 30600 1 26788 25 72091 1 250.1 0
-----Sequential Create----- -----Random Create-----
-Create-- --Read--- -Delete-- -Create-- --Read--- -Delete--
files /sec %CP /sec %CP /sec %CP /sec %CP /sec %CP /sec %CP
16 ++++++ +++ ++++++ +++ ++++++ +++ ++++++ +++ ++++++ +++ ++++++ +++
3408M,24869,37,67693,8,30600,1,26788,25,72091,1,250.1,0,16,+++++,+++,+++++,+++,+++++,+++,+++

```

```

[root@ip-SMALL-Instanz ]# kcbench -r
Linux running: 2.6.21.7-2.fc8xen
Compiler: gcc (GCC) 4.1.2 20070925 (Red Hat 4.1.2-33)
CPU: 1 x Intel(R) Xeon(R) CPU E5430 @ 2.66GHz
Memory: 1706 MByte
linux-2.6.20 (/usr/share/kcbench-data/linux-2.6.20)
Filling caches: This might take a while...
Filling caches: Done
Run 1 (-j 2): 1656 (P:39% R:603.68 U:208.19 S:33.18) 604 Sec
Run 2 (-j 2): 1660 (P:40% R:602.16 U:207.84 S:33.26)
Run 3 (-j 2): 1660 (P:40% R:602.21 U:207.86 S:33.34)

```

```

[root@ip-Quadruple-Instanz ~]# kcbench -r
Linux running: 2.6.21.7-2.fc8xen-ec2-v1.0
Compiler: gcc (GCC) 4.1.2 20070925 (Red Hat 4.1.2-33)
CPU: 8 x Intel(R) Xeon(R) CPU X5550 @ 2.67GHz
Memory: 70007 MByte
linux-2.6.20(/usr/share/kcbench-data/linux-2.6.20)
Filling caches: Done
Run 1 (-j 16): 30581 (P:702% R:32.70 U:179.48 S:50.34) 33 Sec
Run 2 (-j 16): 30257 (P:695% R:33.05 U:179.56 S:50.31)
Run 3 (-j 16): 30797 (P:709% R:32.47 U:179.97 S:50.31)

```