Hochschule Esslingen
University of Applied Sciences

Bachelorarbeit

CSP Evaluierung auf Basis von IaC Unterstützung und Metriken

im Studiengang Softwaretechnik und Medieninformatik
der Fakultät Informationstechnik
Wintersemester 2021/2022

Julian Schallenmüller

Zeitraum: 15.10.2021 - 15.01.2022
Prüfer: Prof. Dr.-Ing. Kai Warendorf
Zweitprüfer: Prof. Dr. rer. nat. Mirko Sonntag

Firma: Noavtec Consulting GmbH
Betreuer: Dipl.-Ing. (BA) Matthias Häussler

Ehrenwörtliche Erklärung

Hiermit versichere ich, die vorliegende Arbeit selbstständig und unter ausschließlicher Verwendung der angegebenen Literatur und Hilfsmittel erstellt zu haben.

Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Esslingen, den 9. November 2021 _____

Unterschrift

todo: 9. November 2021 - 15:41 Uhr

Sperrvermerk

Die nachfolgende Bachelorarbeit enthält vertrauliche Daten der Noavtec Consulting GmbH. Veröffentlichungen oder Vervielfältigungen dieser Arbeit – auch nur auszugsweise – sind ohne ausdrückliche Genehmigung der Noavtec Consulting GmbH nicht gestattet. Diese Arbeit ist nur den Prüfern sowie den Mitgliedern des Prüfungsausschusses zugänglich zu machen.

Zitat

„“Showing a strong success and visible benefits is key to getting others to agree to try your way of doing things.”“

Frederic Rivain

Vorwort

Dank an die Firma und die Firmenmitarbeiter, max. 1/2 Seite

Kurz-Zusammenfassung

„Aushängeschild"“ der Arbeit, max 1 Seite

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

1.1 Einleitung

Eines der wichtigsten Schlagworte im Zeitalter der fortschreitenden Digitalisierung ist der Begriff des Cloud Computings. Auch in Bereichen der Industrie wie der Finanz- und Versicherungsbranche die sich zu großen Teilen aufgrund von Sicherheits- und anderen Bedenken lange Zeit gegen die Nutzung Cloud-basierter Systeme entschieden hatte gewinnt das Thema mehr und mehr Relevanz. Die Nutzung von Cloud-Technologien verspricht die Möglichkeit schneller auf Anforderungen von Kunden reagieren zu können, kostengünstige und flexible Skalierung der eigenen Rechenkapazitäten, Einsparungen durch den Wegfall eigener IT-Infrastruktur Fachleute und mehr.
Gemeinsam mit der Eröffnung neuer Möglichkeiten bringt die Einführung neuer Technologie auch immer eine Reihe eigener Herausforderungen mit sich. Für eine erfolgreiche und gewinnbringende Einführung dieser ist es essentiell diese zu verstehen und die passende Denkweisen und Werkzeuge mit denen die aufkommenden Probleme gelöst werden können entsprechend einzusetzen.

1.2 Motivation und Ziele der Arbeit

Das Thema mit dem sich diese Arbeit befassen wird ist das automatisierte Managen und Bereitstellen von Cloud Computing Resourcen, zusammengefasst unter dem Begriff Infrastructure as Code (IaC). Die Grundlagenkapitel werden zu diesem Zweck auf den technischen Kontext und die Relevanz von Cloud Computing, IaC und dem Software Tool Terraform eingehen. Es wird erläutert werden an welcher Stelle die ausgewählten Plattformen und Software zum Einsatz kommen, welche Probleme dadurch gelöst werden, wo deren Vorteile und Grenzen liegen sowie welche Alternativen existieren und wo ergänzende Werkzeuge zum Einsatz kommen können.
Den Kern der Arbeit bildet ein Vergleich der ausgewählten Cloud Service Provider auf Basis der Unterstzützung Von IaC mit Terraform. Hierfür wird die Infrastruktur einer Schulung der Firma Novatec Consulting GmbH in der die wichtigsten grundlegenden Cloud Computing Resourcen Verwendung finden auf verschiedenen Plattformen mit Terraform bereitgestellt und einem Vergleich unterzogen. Als Vergleichsmetriken werden Kriterien der Norm ISO/IEC 25000 herangezogen um eine fachgerechte und neutrale Bewertung zu gewährleisten. Dabei soll auch in Betracht gezogen werden wie hoch der Aufwand für die

Migration eines bestehenden Systems zu IaC ausfällt. Nach der Auswertung der Vergleichsergebnisse werden die gewonnenen Erkentnisse zusammengefasst, weitere untersuchenswerte Aspekte aufgeführt und ein Ausblick auf aktuelle ud Zukünftige Entwicklungen gegeben. Durch die sehr aktuelle Relevanz des Themas werden sich mit hoher Wahrscheinlichkeit auch in Zukunft neue Software und Technologien durchsetzen, alte verdrängt und neue Lösungsansätze für bestehende Herausforderungen und Probleme etabliert werden. Diese Arbeit wird jedoch nur die aktuell relevantentesten PLattformen und Tools betrachten um eine Eintscheidungsbasis für deren Einsatz zum aktuellen Zeitpunkt bieten zu können.

2 Grundlagen

2.1 Funktionsprinzip, Vorteile und Herausforderungen des modernen Cloud Computings

2.1.1 Definition und Funktionsweise

Das National Institute of Standards and Technology (NIST) der Vereinigten Staaten von Amerika definiert Cloud Computing im Abstract der NIST SP-800-145 folgendermaßen:

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. This cloud model is composed of five essential characteristics, three service models, and four deployment models.

Cloud Computing beschreibt ein Modell das es ermöglicht ortsunabhängig, zweckdienlich und zeitunabhängig auf einen konfigurierbaren Pool an Computerresourcen (Netzwerke, Server, Datenspeicher, Anwendungen und Services) zuzugreifen die schnell und mit minimalem Aufwand und minimaler notwendiger Interaktion bereitgestellt und wieder abgebaut werden können. Dieses Cloud Modell beschreibt fünf essentielle Charakteristiken, drei Servicemodelle und vier Bereitstellungsmodelle.

Weiter definiert das Dokument die fünf Charakteristiken in den folgenden Punkten:

On-demand-self-service: Der Nutzer kann eigenmächtig die benötigten Resourcen automatisch bereitstellen, es wird keine menschliche Interaktion benötigt.

Broad network access: Auf Leistungen wird über das normale Internet mit standard Mechanismen die die Nutzung von Thin Clients und Fat Clients (Smartphones, Tablets, Laptops oder Workstations) fördern zugegriffen.

Resource Pooling:: Die Computerresourcen des Anbieters werden in einem gemeinsamen Pool für mehrere Kunden in einem mandantentauglichen Modell bereitgestellt, physische

und virtuelle Resourcen werden nach dynamisch zugewiesen und entsprechend der Nachfrage neu verteilt. Es wird eine empfundene Ortsunabhängigkeit hergestellt indem der Nutzer kein genaues Wissen darüber besitzt wo sich dessen Resources befinden, auf höherem Level wie beispielsweise dem Staat, der Region oder auch Rechenzentrum kann der Ort vom Nutzer spezifiziert werden. Die bereitgestellten Resourcen beinhalten zum Beispiel Datenspeicher, Rechenleistung, Rechenspeicher und Netzwerkbandbreite.

Rapid Elasticity: Rechenkapazitäten werden dehnbar bereitgestellt und abgebaut, teilweise automatisch, um entsprechend der Nachfrage schnell hoch- und wieder zurück skalieren zu können, Rechenkapazitäten erscheinen dadurch unbegrenzt und zu jeder Zeit in jedem Umfang bereitgestellt werden.

Measured Service: Cloud systeme kontrollieren und optimieren Resourcennutzung automatisch mithilfe eines Messungs-systems dass auf einer abstrakten Ebene den entsprechenden Service (Datenspeicher, Rechenleistung, Benutzerkonten) überwacht, kontrolliert und Bericht erstattet um sowohl für Anbieter als auch Kunden Transparenz herzustellen.

Es wird zwischen drei grundlegende Cloud Service Modellen unterschieden: Infrastructure as a Service (IaaS), Platform as a Service (PaaS) und Software as a Service (SaaS).

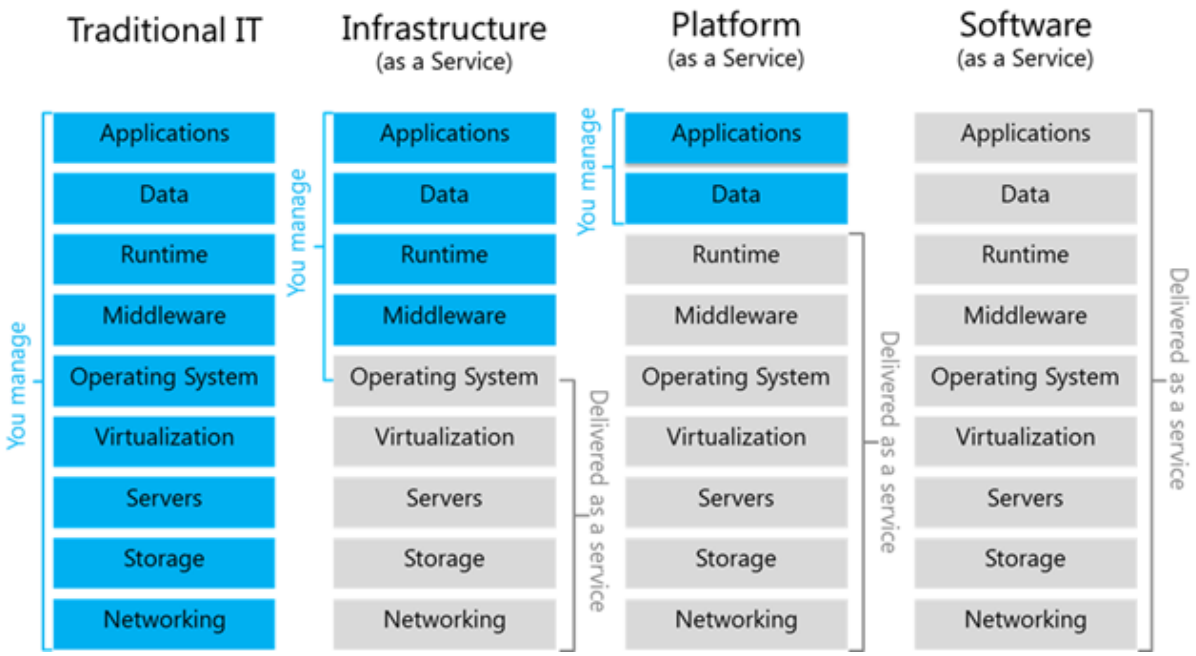

Abb. 2.1: Die grundlegenden Cloud Service Modelle im Überblick

Infrastructure as a Service: Der Nutzer hat die Fähigkeit Rechenleistung, Datenspeicher Netzwerke und weitere fundamentale Computerresourcen bereitzustellen und beliebige Software darauf zu betreiben, dazu können Betriebssysteme und Anwendungen gehören.

Die darunterliegende Infrastruktur wird vom Anbieter betrieben, der Nutzer kann aber eingeschränkte Kontrolle über bestimmte Komponenten haben, dazu gehören beispielsweise Firewalls.

Platform as a Service: Der Nutzer verfügt über die Fähigkeit seine eingkaufte oder selbst erstellten Anwendungen auf der Cloud Infrastruktur zu betreiben, die notwendige Umgebung die über Sprachen, Bibliotheken, Tools und Services verfügt wird vom Anbieter bereitgestellt. Die darunter liegende Infrastruktur mit Netzwerken, Servern, Betriebssystemen und Speicher wird vom Anbieter betrieben, der Nutzer hat die Kontrolle über die Anwendung und Konfiguration der Umgebung in der die Anwendung betrieben wird.

Software as a Service: Dem Nutzer wird der Zugriff auf die vom Anbieter in der Cloud Infrastruktur betriebenen Softwareanwendung gewährt. Auf diese wird mithilfe eines Thin oder Fat Clients zugegriffen, dabei kümmert sich der Nutzer nicht um den Betrieb und die Konfiguration der darunterliegenden Cloud Infrastruktur wie Netzwerke, Server, Betriebssystem, Speicher und die Anwendung selbst mit Ausnahme eingeschränkter Nutzereinstellungen.

Die von NIST unterschiedenen grundlegenden Service Modellen können noch weiter differenziert werden. Ein Beispiel hierfür ist das Modell Function as a Service (FaaS) als Subset von PaaS. FaaS erlaubt es Programmcode auszuführen ohne sich um die Bereitstellung weiterer Infrastruktur kümmern zu müssen wie es der Betrieb eines gewöhnlichen Microservices verlangt.

In Art der Bereitstellung eines Cloud Services werden vier grundlegende Modelle unterschieden; Public, Private, Hybrid und Community Cloud Modelle.

Private Cloud: Die Cloud Infrastruktur wird ausschließlich für die Nutzung durch eine einzige Organisation mit mehreren Nutzern bereitgestellt. Besitz und Betrieb liegen dabei entweder bei der selben Organisation, einer Drittpartei oder einer Kombination beider, die Infrastruktur kann dabei On- oder Off Premise betrieben werden.

Public Cloud: Die Public Cloud steht für die Nutzung durch die allgemeine Öffentlichkeit bereit. Die Cloud Infrastruktur befindet sich im Besitz eines Unternehmens, Bildungseinrichtung, Regierungsorganisation oder einer Kombination aus diesen und wird auch von der selben Organisation On Premise betrieben.

Community Cloud: Eine Community Cloud wird von einer Gemeinschaft von Nutzern mit gemeinsamen Anliegen eingesetzt. Der Besitz und Betrieb liegen dabei bei einem oder mehreren Mitgliedern dieser Gemeinschaft, einer Drittpartei und kann Off oder On Premise betrieben werden.

Hybrid Cloud: Die Hybrid Cloud besteht aus einer Kombination der beschriebenen Modelle (Public, Private und Community). Diese bilden dabei eigene Instanzen die aber durch standardisierte oder proprietäre Schnittstellen den Transfer von Daten und Anwendungen zwischen den Instanzen erlauben.

2.1.2 Zu erwägende Vor- und Nachteile des Einsatzes von Cloud Computing

Vorteile und Treiber der Adoption von Cloud Computing

Wirtschaftliche Vorteile: Ein Vorteil in der Nutzung von Cloud Computing kann darin liegen dass ein Großteil der für den Betrieb notwendigen Infrastruktur nicht mehr vom Unternehmen selbst eingakuft, eingerichtet und betrieben werden muss (vgl. linke Seite Abb 2.1). Potentiell hohe Kosten die bereits vor der Inbetriebnahme eines Systems mit einem höheren Risiko aufgewendet werden mussten stellen in Form von individuell geringeren laufenden Beträgen ein deutlich reduziertes Risiko da.
Sofern der Einsatz von Cloud Computing in einer sinnvollen und korrekten Weise erfolgt können je nach Fall die Gesamtkosten um einen hohen Anteil reduziert werden.
Die Gesamtkostenersparnisse stehen auch im Zusammenhang mit Skaleneffekten die für große Cloudanbieter gelten. Der Betrieb eines einzelnen Servers ist im Verhältnis mit bedeutend höheren Kosten verbunden als das hinzufügen eines äquivalenten System zu einem Rechenzentrum im Betrieb von AWS oder einem vergleichbaren Anbieter.

Skalierbarkeit: Besonders für schnell wachsende Unternehmen ist die Option der schnellen Skalierbarkeit einer der prominentesten Vorteile der Cloud. Es kann nicht nur auf vorhersagbare Anstiege (zum Beispiel ausgelöst durch eine Verkaufsaktion) sondern auch auf unvorhersehbare Ereignisse reagiert werden. Zusätzlich ist es möglich diese Skalierung nicht nur bis zu einem bestimmten Limit, sondern nahezu unendlich zu betreiben. Wichtig ist auch dass sowohl auf steigende als auch sinkende Nachfrage reagiert werden kann.

Resilienz: In einem worst case Szenario kann ein ganzes Rechenzentrum durch unvorhergesehen Ereignisse wie beispielsweise Brände, Naturkatastrophen oder anderes vollständig zerstört werden. Selbst wenn Backup-Rechenzentren verfügbar sind ist eine übertragung der Operationen kein trivialer Ablauf und birgt oft nicht außer Acht zu lassende Risiken. Die Flexibilität der Cloud erlaubt es die gesamte Infrastruktur mit sehr geringem Aufwand in nicht betroffene Regionen zu verlagern und die Kontinuität der Geschäftstätigkeiten mit minimaler Unterbrechung aufrecht zu erhalten.

Security: Security Aspekte können sowohl einen Vor- als auch Nachteil von Cloud Computings darstellen. Hier sollen zuerst Vorteile dargelegt werden, potentielle Probleme werden im nächsten Unterkapitel beschrieben. Die technischen Möglichkeiten und besonders auch die Wahrnung des Themas Sicherheit in der Cloud unterlagen und unterliegen auch immernoch einem deutlichen Wandel. Cloud Anbieter investieren viele Resourcesn in Sicherheit und stellen dem Nutzer zum Beispiel bereits sicher implementierte Verschlüsselungen zur Verfügung oder bieten einen gewissen Schutzen vor Denial-of-Service Angriffen durch die natürliche Skalierbarkeit.

Nachteile und Risiken

Netzwerkabhängigeit: Da der Zugriff auf Cloud Dienste über das Internet erfolgt entsteht dadurch entsprechend auch eine hohe Abhängigkeit. Eine stabiele und schnelle Netzwerkanbindung ist vorraussetzung dass effektiv gearbeitet werden kann, bei lokal gehosteten Systemen ist diese Abhängigkeit entsprechend geringer.

Vendor Lock-in: Bei der Nutzung eines Cloud Anbieters entsteht die Gefahr sich zu sehr in Abhängigkeit eines einzelenen Anbieters zu begeben. Im Fall einer Änderung der Nutzungsbedingungen oder einer Änderung im Bezahlmodell die den eignen Interessen stark entgegen steht besteht die Gefahr bereits so abhängig von diesem Anbieter zu sein dass die Kosten eines Wechsels derart hoch ausfallen dass man gezwungen ist die Bedingugen zu akzeptieren.

Security und Privacy: Sicherheitsrisiken sind einer der meistgenannten Gründe die gegen Cloud Computing sprechen, besonders im Fall der Nutzung einer Public Cloud. Die Gefahr dass Daten in die Hände dritter gelangen kann zum zum Beispiel nicht vollständig ausgeschlossen werden. Da die Verantwortung über die Sicherheit der Daten dem Cloud Anbieter unterliegt kann es auch zu Problemen in Hinsicht der Privateheit der Daten kommen, sollte eine Regierungsorganisation Zugriff auf bestimmte Daten eines Nutzer verlangen könnten diese ohne dessen Einverständnis gewährt werden.

Kosten: Auch wenn die Nutzung von Cloud Computing mit dem Vorteil geringerer Kosten beworben ist dies nicht zwingend garantiert. Werden die vorhandenen Systeme ungünstig verwendet, zum Beispiel bleiben viele gebuchte Ressourcen ungenutzt und IP-Adressen unverwendet, können hohe Kosten entstehen, auch in der Phase des Übergangs zu Cloud Computing können höhere Kosten entstehen als in einem vergleichbaren Zeitraum davor.

2.1.3 Grundlegende Erklärung von DevOps

Die exakte Definition und Abgrenzung des Begriffes DevOps ist ein Thema über das es keine uniform akzeptierte allein gültige Definition und Abgrenzung gibt. Im allgemeinen aber bezeichnet DevOps eine stärkere Vereinigung der Entwicklungs- (Dev) und Betriebs- (Ops) Teams eines Softwareprojekts durch die Anwendung einer effektiveren Arbeitskultur und -philosophie mit neuen Methoden, Werkzeugen und Prozessen. Verschiedene Organisationen legen hierbei in ihrer Definition die Schwerpunkte auf unterschiedliche Aspekte. Als Beispiel betont Amazon hierbei besonders die schnelle Auslieferung neuer Produkte (äbility to deliver applications and service at high velocity"), Microsoft die Kollaboration zwischen den beteiligten Teams ("DevOps enables formerly siloed roles - development, IT operations, quality engineering, and security - to coordinate and collaborate to produce better, more reliable products.").
Das DevOps Reasearch and Assessment (DORA) Team hat über sieben Jahre mit 32.000 Beteiligten die Praktiken und Fähigkeiten untersucht die hohe Leistungsfähigkeit bei Entwicklung und Auslieferung von Sofware fördern. Eine Übersicht über die Erkenntnisse

dieser Studie ist als Diagramm im Anhang ... zu finden.
Infrastructure as Code spielt als Werkzeug für die automatisierte Bereitstellung der benötigten Computerresourcen eine wichtige Rolle im DevOps Prozess.

GitOps als Werkzeug für Continuous Delivery im Rahmen von DevOps

GitOps wird von Gitlab als ein betriebliches Rahmenkonzept (operational Framework) defniert das DevOps Best Practices in der automatiserten Bereitstellung von Infrastruktur anwendet. GitOps benötigt dabei drei Hauptkomponenten:
Infrastructure as Code, Merge Requests und CI/CD. Organisationen die mit einer ausgereiften Anwendung der DevOps Kultur arbeiten können über GitOps hunderte Male pro Tag neuen Code auf den Produktionsservern installieren. Im praktischen Teil dieser Arbeit wird GitOps verwendet um da es eine einfache Integration mit Github ermöglicht, Github selbst wird als Versionskontrollsystem verwendet da es weithin etabliert und breit unterstützt ist und von der Novatec Consulting GmbH bevorzugt eingesetzt wird.

2.1.4 Überblick über die wichtigsten Public Cloud Service Provider

Da der Kern der Arbeit die Infrastructure as Code Unterstützung der wichtigsten Cloud Service Provider in deren Public Cloud untersucht soll hier ein kurzer Überblick über den aktuellen Markt in diesem Bereich gegeben werden.

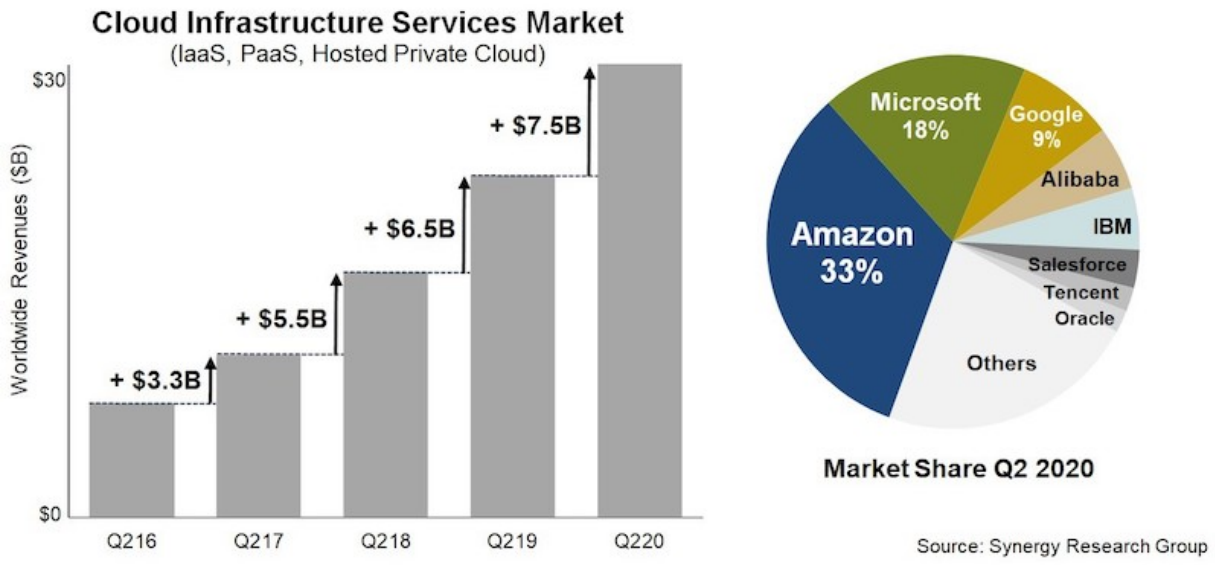

Abb. 2.2: CSP Market Share Q2 2020 nach Umsatz

1. Amazon: Amazon besitzt mit 33% den größten Anteil am Markt. Die Amazon Elastic Compute Cloud (EC2) ist die älteste der aktuell großen Cloud Computing Plattformen, der erste Release fand im August 2006 statt.

2. Microsoft: Microsoft stellt den zweitgrößten Anteil am Markt mit 18%, der erste Release von Microsoft Cloud Computing Service Microsoft Azure erfolgte im Oktober 2008. Gemeinsam besaßen Amazon und Microsoft 2020 über 50% des gesamten Marktes.

3. Google: Google's Google Cloud Platform steht mit 9% an dritter Stelle. Google Compute Engine, die IaaS Komponente der Cloud Services von Google wurde im Juni 2012 veröffentlicht.

4. Alibaba Cloud: Die viertgrößte Cloud Plattform ist die Alibaba Cloud der Alibaba Group. Alibaba Cloud bietet zusätzlichzu seinem Elastic Compute Service (ECS) einen dedizierte GPU basierten Service an.

5. IBM Cloud: Die aktuell fünftgrößte Cloud Computing Plattform ist die IBM Cloud von IBM. Bis Juni 2013 eine eigene Firma unter dem Namen Softlayer wurde diese von IBM übernommen und 2017 gemeinsam mit den anderen Cloud Services der Firma in ein einheitliches Portfolio unter dem Namen IBM Coud aufgenommen.

Abb. 2.3: CSP Gartner Magic Quadrant August 2020

Der Gartner Magic Quadrant für Cloud Service Provider bietet einen groben Überblick über den Umfang der Angebote (Completeness of Vision) und die Ausgereiftheit einer Plattform (Ability to Execute). Deutlich erkennbar ist hier die Vormachtsstellung von Amazon gegenüber Microsoft und Google sowie der Abstand von Google zum nächst größten Anbieter Alibaba Cloud. TODO: Irgendwas mit IBM Ability to Execute vs Marktanteil verglichen mit Oracle

2.2 Infrastructure as Code

Infrastructure as Code beschreibt einen Ansatz zur Automatiesierung der Bereitstellung von Infrastruktur basierend auf Methoden aus der Softwareentwicklung (vgl Kief Morris Infrastructure as Code S.4).
Statt eines manuellen Aufbaus und direkter Konfiguration der einzelnen Komponenten werden maschinenlesbare Dateien verfasst welche dann von einem IaC Tool eingelesen und verarbeitet werden. Dabei kommen bevorzugt deklarative Sprachen zum Einsatz deren höhere Abstraktion mehr Flexibilität als ein imperativer Ansatz erlaubt. (vgl. https://docs.microsoft.com/us/devops/deliver/what-is-infrastructure-as-code)

2.2.1 Technologischer Wandel und das Cloud Age Mindset

Durch die Technologien der Cloud ist es möglich eine gewünschte IT-Infrastruktur sehr viel schneller bereitzustellen als zuvor möglich. Statt des Einkaufs, Anschließens und Einrichtens eines physischen Servers das, je nach Szenario einen Zeitraum von mindestens mehreren Stunden oder Tagen bis zu Wochen gedauert hätte können virtuelle Resourcen in der Cloud in wenigen Minuten bereitgestellt werden. Der schnellere Ablauf wird durch die Automatisierung von Prozessen wie etwa der Bereitstellung der Infrastruktur mithilfe von IaC Tools noch verstärkt, nicht nur bei der initialen Mit diesen Veränderungen wird das Management und die Erweiterung der bestehenden Systeme jedoch nicht unbedingt einfacher (vgl IaC Kief Morris S.3), die Verwendung der alten IT-Governance Modelle (TODO Fußnotenerklärung) die sich bisher bewährt haben sind jedoch aufgrund der Veränderungen ungeeignet. Kief Morris stellt die fundamentalen Unterschiede des Arbeitens mit Cloud-Technologien mithilfe der folgenden Tabelle dar.

Iron Age	Cloud Age
Cost of change is high	Cost of change is low
Changes represent failure (changes must be “managed,” “controlled”)	Changes represent learning and improvement
Reduce opportunities to fail	Maximize speed of improvement
Deliver in large batches, test at the end	Deliver small changes, test continuously
Long release cycles	Short release cycles
Monolithic architectures (fewer, larger moving parts)	Microservices architectures (more, smaller parts)
GUI-driven or physical configuration	Configuration as Code

Abb. 2.4: Iron Age vs Cloud Age

Veränderungen in der ’Iron Age’ sind aufwändig und teuer und stellen ein Risiko dar, es wird versucht diese Risikopunkte zu reduzieren daher werden viele Veränderung gebüdelt

getestet und eingeführt wodurch lange Release-Zyklen entstehen. Die Architekturen die dadurch befördert werden sind eher monolytisch die Konfiguration erfolgt eher mithilfe von GUI gesteuerten Programmen oder physischen, zum Beispiel wenn ein neuer Server in ein Netzwerk eingebunden wird. Veränderungen in der Cloud stellen fast genau das Gegenteil dar, daher wird erkennbar dass eine auf die 'Iron Age' zugeschnittene Arbeitsweise für die Cloud nicht effektiv ist, es ist ein neues Cloud Age Mindset das die rechte Spalte der Tabelle verinnerlicht erforderlich um die Vorteile der Cloud wirklich effektiv nutzen zu können.

2.2.2 Vorteile von IaC und durch IaC gelöste Probleme des manuellen Provisionings

Kein Configuration Drift durch einheitliche Codebase: Configuration drift bezeichnet eine über die Zeit wachsende Abweichung zweier ursprünglich identischer Systeme. Wird ein gleiches System, zum Beispiel ein Applikationsserver, in verschiedenen Umgebungen eingesetzt stellen diese oftmals auch leicht verschiedene Anforderungen auf die dann mit Optimierungen, etwa in Form spezifischer Konfigurationsdetails, reagiert werden kann. Wird nun das ursprüngliche System geupdated werden individuelle, oft undokumentierte Anpassungen nicht berücksichtigt und ein Update kann unbequeme Konsequenze nach sich ziehen. Werden alle Veränderungen in einer einheitlichen Codebase verwaltet und Updates häufig vorgenommen verhindert man starken Configuration Drift der über Zeit stattfindet

Wiederverwendbarkeit durch einheitlichen Code: Ein weiterer Vorteil der durch die Verwendung einer einheitlichen Codebase entsteht ist die Wiederverwendbarkeit und Repoduzierbarkeit eines Systems. Wenn ein identisches System an einer anderen Stelle aufgebaut werden soll oder sollte das System aus unvorhergesehen Gründen in seiner Gesamtheit verloren gehen kann es schnell und mit verhältnismäßig geringem Aufwand reproduziert werden. Ein dazu passender Ausdruck in Bezug auf Server ist 'Cattle not Pets'. Statt sich individuell und mit großem Aufwand um einzelne Server zu kümmern wie man es etwa mit dem eigenen Haustier tut sollten Server wie leicht ersetzbares Vieh behandelt werden.

Schnelleres Provisioning durch Cloud: Ein bereits häufig angesprochener Vorteil ist die schnelle Bereitstellung, auf tiefere Erklärung kann daher hier verzichtet werden.

Schnellerer Profit: Schnellere Bereitstellung der Hardware und schnellere Auslieferung von Software die durch DevOps Methoden begünstigt werden erzeugen entsprechend auch schneller einen Mehrwert.

Einheitliches Tooling in Dev, Ops und mehr: Verwendung von IaC fördert die einheitlicheres Tooling in allen Bereichen die mit einem Softwareprodukt in Zusammenhang stehen. Cloud Technologien fördern und ermöglichen diese Vereinheitlichung, manuelles Provisioning ohne ein Cloud Modell erschwert dies oder macht es sogar unmöglich.

Stärkere Automatisierung im Arbeitsablauf: Automatisierung bedeutet immer einen gewissen upfront Overhead, jedoch kann auf längere Sicht deutlich von automatisierten Abläufen profitiert weren. Ein Beispiel hierfür sind automatisierte und in eine CI/CD Pipeline integrierte Tests.

Höhere Zuverlässigkeit und Sicherheit durch schnelle Updates: Von eine Struktur die schnelle und häufige Veränderungen fördert kann auch die Sicherheit und Zuverlässigkeit eines Systems profitieren. Auf Sicherheitsrisiken kann in kurzer Zeit und ohne viel Aufwand reagiert werden, eine unsichere Funktion kann etwa schnell durch eine sichere Variante ausgetauscht werden, mögliche damit verbundene Probleme können dann in automatisierten Tests erkannt werden wodurch das System zuverlässig bleibt.

Schnellere Fehlersuche und -behebung:

Fehler die dennoch auftreten können dann durch den Einsatz kleinerer Komponenten schneller isoliert, gefunden und behoben werden. Monolithische Systeme erschweren dies durch dadurch dass weniger klar isolierte Komponenten vorhanden sind die häufig mehr Abhängigkeiten aufweisen und daher schwerer als einzelne Einheit getestet werden können.

2.2.3 Argumente gegen IaC/Gründe der Ablehnung

Kief Morris bennent drei häufige Argumente die gegen die Einführung von IaC genannt werden, diese sollen Im Kontext der gennanten Vorteile betrachtet werden um deren Ursachen und Validität festzustellen.

- Veränderungen werden nicht oft durchgeführt wodurch sich Automatisierung nicht lohnt
- Zuerst Infrastruktur bauen, Automatisierung erfolgt später - Man müsse zwischen Geschwindigkeit der Auslieferung und Qualität wählen

2.2.4 Die Schlüsselmetriken und Kernverfahren zum erfolgreichen Einsatz von IaC

Metriken:

- Delivery Lead time: Zeit für Implementierung,test auslieferung von Veränderungen am Produktionssystem

- Deployment Frequency: Wie oft wird deployed

- Change fail percentage: Aka wie gut wird getestet

- Mean time to restore: Wie lang bis System wieder läuft

Kernverfahren:

- Define everything as Code

- Continuously Test and Deliver all work in Progress

- Build small simple pieces

2.2.5 Warum und für was genau wird Terraform hier verwendet?

- Typen von IaC Tools, Terraform Provisioning tool vs Configuration Management

- Terraform wird als Provisioning Tool verwendet da es darauf stark spezialisiert - Terraform unterstützt viele Provider - Modularität

Limitierungen weil Nischenprodukt

- Konstrukte wie Schleifen oder If else sind unintuitiv oder nicht vorhanden - Kein try catch error handling - Kein eingebautes rollback - Aktuell können keine Skripte aufgrund bestehender Infrastruktur automatisch erstellt werden

https://medium.com/cloudnativeinfra/when-to-use-which-infrastructure-as-code-tool-665af289fbde - Alternativen zu Terraform und Ergänzungen

3 Aufbau und Untersuchung

Beschreibung der HW- und SW-Realisierung

3.1 High-level Aufbau der Infrastruktur des Versuchsobjekts

Beispiel Text

3.2 Zu analysiernde Aspekte und Eigenschaften

3.3 Konkreter Aufbau in Microsoft Azure

3.4 Konkreter Aufbau in Amazon AWS

3.5 Konkreter Aufbau in Google Cloud Platform

3.6 Literaturverweise

Verweise im Text: [1] und [Gun04].

4 Ergebnisse

4.1 Bewertung Azure

4.2 Bewertung AWS

4.3 Bewertung GCP

4.4 Resultate und Vergleichsmatrix

„Neuigkeiten“ Messergebnisse

5 Schluss

Ergebnis-Bewertung, Zusammenfassung und Ausblick

5.1 Gewonnenen Erkenntnisse

5.2 Zusammenfassung der Arbeit

5.3 Mögliche weitere untersuchenswerte Aspekte

5.4 Aktuelle und zukünftige Entwicklungen

A Kapitel im Anhang

Alles was den Hauptteil unnötig vergrößert hätte, z. B. HW-/SW-Dokumentationen, Bedienungsanleitungen, Code-Listings, Diagramme

Literaturverzeichnis

[1] Thomas Nonnenmacher, LaTeX Grundlagen - Setzen einer wissenschaftlichen Arbeit Skript, 2008,http://www.stz-softwaretechnik.de; *(Bei STZ Internetseite unter Publikationen - Skripte) [V. 2.0 26.02.08]*

[Gun04] Karsten Günther, LaTeX2 — Das umfassende Handbuch, Galileo Computing, 2004, http://www.galileocomputing.de/katalog/buecher/titel/gp/titelID-768; *1. Auflage*