

Bachelorarbeit  
**Cloud Service Provider Evaluierung  
auf Basis von  
Infrastructure as Code Unterstützung**

im Studiengang Softwaretechnik und Medieninformatik  
der Fakultät Informationstechnik  
Wintersemester 2021/22

Julian Schallenmüller

**Zeitraum:** 15.10.2021-31.01.2022

**Prüfer:** Prof. Dr.-Ing. Kai Warendorf

**Zweitprüfer:** Prof. Dr. Rer. Nat. Mirko Sonntag

---

**Firma:** Novatec 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 24. Januar 2022 \_\_\_\_\_  
Unterschrift

# Sperrvermerk

Die nachfolgende Bachelorarbeit enthält vertrauliche Daten der Novatec Consulting GmbH. Veröffentlichungen oder Vervielfältigungen dieser Arbeit – auch nur auszugsweise – sind ohne ausdrückliche Genehmigung der Novatec 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

Ich möchte mich an dieser Stelle bei der Firma Novatec bedanken, in der ich seit meiner Zeit als Praktikant, danach als Werkstudent und nun auch während meiner Bachelorarbeit immer willkommen war und bei allen Herausforderungen und Problemen stets unterstützt wurde.

Besonderer Dank geht hierbei an die Mitarbeiter der PA TC und insbesondere an meinen Betreuer Matthias Häussler. Herr Häussler stand bereits vor und vor allem während meiner Zeit als Bachelorant immer für Fragen und Ratschläge, auch über die normalen Arbeitszeiten hinaus, bereit.

Genauso möchte ich mich bei Herrn Prof. Dr.-Ing Warendorf für die Betreuung meiner Bachelorarbeit von Seiten der Hochschule Esslingen bedanken. Durch seine schnelle und unkomplizierte Art der Kommunikation konnten alle organisatorischen Fragen und Aufgaben rund um die Bachelorarbeit immer schnell beantwortet und bewältigt werden.

# Kurz-Zusammenfassung

Im Rahmen dieser Arbeit wird ein Vergleich der Cloud Service Provider Microsoft und Google in Bezug auf deren Unterstützung von Infrastructure as Code mit Terraform durchgeführt.

Dieser Vergleich soll die Qualität der Unterstützung von deren Plattformen Microsoft Azure und Google Cloud Platform in einem gewöhnlichen Szenario evaluieren und die Frage nach der Einsatzreife beider Plattformen beantworten.

Zusätzlich soll diese Arbeit ermitteln wie einheitlich die konkrete Umsetzung von Infrastructure as Code mit Terraform in einem funktional gleichwertigen Szenario für verschiedene Plattformen ausfällt.

Für die Beantwortung dieser Fragen wurde ein beispielhaftes Infrastruktur-System auf beiden Plattformen mithilfe von Terraform implementiert und deployed. Die in der Evaluierung eingesetzten Bewertungskriterien wurden aus den Kriterien der des Softwarequalitätsmodell der ISO/IEC 25010 ausgewählt.

Zusätzlich zu der Untersuchung des Testsystems wurde eine Reihe unterstützender Versuche durchgeführt, durch diese konnte ein Teil der aufgeworfenen Fragen beantwortet und ein vollständigeres Bild bezüglich der Performance, Qualität und Unterschiede der untersuchten Plattformen aufgebaut werden.

Das Ergebnis der Untersuchung lässt darauf schließen dass beide Plattformen eine gute Unterstützung von Terraform bieten, Azure genießt hierbei einen leichten Vorsprung hinsichtlich Funktionsumfang und Performance.

Die Einheitlichkeit der Umsetzung fällt jedoch eher gering aus. Sie begrenzt sich auf die Anwendung einer einheitlichen Sprache und Bedienung durch Terraform, es ist jedoch weiterhin ein umfangreiches Verständnis der individuellen Cloud Plattform notwendig um diese erfolgreich einsetzen zu können.

Dennoch ist es sinnvoll Terraform für das Deployment von Infrastruktur zu nutzen, durch die Umsetzung der Prinzipien von Infrastructure as Code werden die Vorteile moderner Cloud Plattformen in vollem Umfang ausgeschöpft und die Wertschöpfungskette von der Idee zur Auslieferung an Kunden kann deutlich beschleunigt werden.

# Inhaltsverzeichnis

1	Einleitung	1
2	Grundlagen	3
2.1	Funktionsprinzip, Vorteile und Herausforderungen	3
2.1.1	Definition und Funktionsweise	3
2.1.2	Vor- und Nachteile des Einsatzes von Cloud Computing	6
2.1.3	Überblick über die wichtigsten Cloud Service Provider	8
2.2	Infrastructure as Code	10
2.2.1	Technologischer Wandel und das Cloud Age Mindset	11
2.2.2	Vorteile von Infrastructure as Code im Vergleich zu manuellem Infrastruktur-Provisioning	12
2.2.3	Herausforderungen im Einsatz von Infrastructure as Code	14
2.2.4	Die drei Kernverfahren von Infrastructure as Code	14
2.3	Funktionsprinzip und Rolle von Terraform im IaC-Anwendungsprozess	14
2.3.1	Funktionsweise von Terraform	14
2.3.2	Einführung in die Hashicorp Configuration Language	14
3	Evaluierungsanforderungen und Umsetzung	15
3.1	Evaluierungsanforderungen	15
3.1.1	Ziel der Evaluierung	15
3.1.2	Untersuchte Komponenten der Terraform Provider	15
3.1.3	Auswahl der Evaluierungskriterien	15
3.2	Umsetzung des Testsystems	15
3.2.1	Eingesetzte Software und Tools	15
3.2.2	High-Level Aufbau des Testsystems	15
3.2.3	Konkreter Aufbau auf Google Cloud Platform	15
3.2.4	Konkreter Aufbau auf Azure	15
3.2.5	Aufbau der ergänzenden Versuche	15
4	Ergebnisse und Bewertung	16
4.1	Evaluierung der Functional Completeness	16
4.2	Ergebnisse und Bewertung der Time Behaviour Tests	16
4.3	Ergebnisse und Bewertung der Recoverability Tests	16
4.4	Ergebnisse und Bewertung der Modifiability Tests	16

---

4.5	Evaluierung der Einheitlichkeit der Testsysteme für Azure und Google Cloud Platform . . . . .	16
5	Schluss	17
5.1	Fazit . . . . .	17
5.2	Ausblick . . . . .	17
A	Kapitel im Anhang	18
	Literaturverzeichnis	19



# Abbildungsverzeichnis

2.1	Die Cloud Service Modelle im Überblick[3]	5
2.2	CSP Market Share Q2 2021 nach Umsatz[14]	9
2.3	CSP Gartner Magic Quadrant Juli 2021[15]	10

# Tabellenverzeichnis

2.1	„Iron vs Cloud Age“ <a href="#">[17]</a>	11
-----	--	----

# 1 Einleitung

Eines der wichtigsten Schlagworte im Zeitalter der fortschreitenden Digitalisierung ist der Begriff des Cloud Computings. Cloud Computing spielt heute längst nicht mehr nur in der IT-Industrie eine wichtige Rolle. Selbst in Bereichen wie der Finanzbranche, die besonders hohen Sicherheitsansprüchen gerecht werden muss, findet Cloud-basierte Software eine zunehmende Verbreitung [1].

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 jedoch auch immer eine Reihe eigener Herausforderungen mit sich. Für den erfolgreichen und gewinnbringenden Einsatz dieser Technologie ist es daher essentiell diese zu verstehen und die neuen Herausforderungen mit angepasster Denkweise und neuen Werkzeugen anzugehen.

Das Thema mit dem sich diese Arbeit befassen wird ist das automatisierte Managen und Bereitstellen von IT-Infrastruktur-Ressourcen, ein Teil der größeren Fachthematik Infrastructure as Code (IaC). Die Grundlagenkapitel werden zu diesem Zweck auf den technischen Kontext und die Relevanz von Cloud Computing, IaC und das Software Tool Terraform eingehen. Es wird erläutert werden an welcher Stelle die entsprechenden Plattformen und Software zum Einsatz kommen, welche Probleme durch diese 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 verschiedener Cloud Service Provider unter dem zentralen Kriterium derer Unterstützung von IaC mit Terraform. Zu diesem Zweck wurde ein Infrastruktur-System auf Basis des Infrastruktur-Showcase der Firma Novatec<sup>1</sup> für ver-

---

<sup>1</sup> Link: <https://github.com/NovatecConsulting/technologyconsulting-showcase-infrastructure>

schiedene Cloud Plattformen in Terraform Code implementiert und auf diesen deployed. Als Grundlage für den Vergleich dienen eine Auswahl von Aspekten aus dem Softwarequalitätsmodell der ISO/IEC 25010. Da jedoch nicht alle enthaltenen Qualitätsaspekte der Norm für diesen konkreten Vergleich geeignet sind werden diese zunächst auf ihre Anwendbarkeit im vorliegenden Fall hin analysiert und bewertet, anschließend werden einige der relevantesten Kriterien ausgewählt und die Untersuchung anhand derer durchgeführt. Da sich durch Unterschiede zwischen den Cloud Plattformen, zum Beispiel bei Auswahl der Leistungsfähigkeit verwendeter Virtueller Maschinen, einige Auffälligkeiten ergeben werden zusätzliche Tests definiert und durchgeführt um ein insgesamt vollständigeres Bild liefern zu können.

Im Anschluss werden die Ergebnisse des Vergleichs bewertet und die daraus resultierenden Erkenntnisse zusammengefasst. Aus diesen Erkenntnissen kann dann ein Fazit gezogen werden das deren Bedeutung im Kontext des aktuellen Stands der Technik interpretiert und bewertet. Den Abschluss bildet ein Ausblick auf zusätzliche Themen die als nächstes betrachtet werden sollten wenn es darum geht Infrastructure as Code für das Infrastruktur-Management in einer realen Produktivumgebung einzusetzen. Diese Themen werden im Verlauf der Arbeit bereits angesprochen, können hier jedoch noch nicht in zufriedenstellendem Umfang betrachtet werden.

## 2 Grundlagen

### 2.1 Funktionsprinzip, Vorteile und Herausforderungen des modernen Cloud Computings

Um die Rolle von Infrastructure as Code und Terraform vollständig erläutern zu können sollte zuerst das grundlegende Funktionsprinzip und die verschiedenen Service- und Bereitstellungs-Modelle moderner Cloud Plattformen erklärt werden. Die am meisten verwendete Definition von Cloud Computing wurde vom National Institute of Standards and Technology der Vereinigten Staaten von Amerika veröffentlicht und wird im folgenden Kapitel zusammengefasst.

#### 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[2] 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 Computing Ressourcen (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 Computing Ressourcen automatisch bereitstellen, es wird keine menschliche Interaktion benötigt.

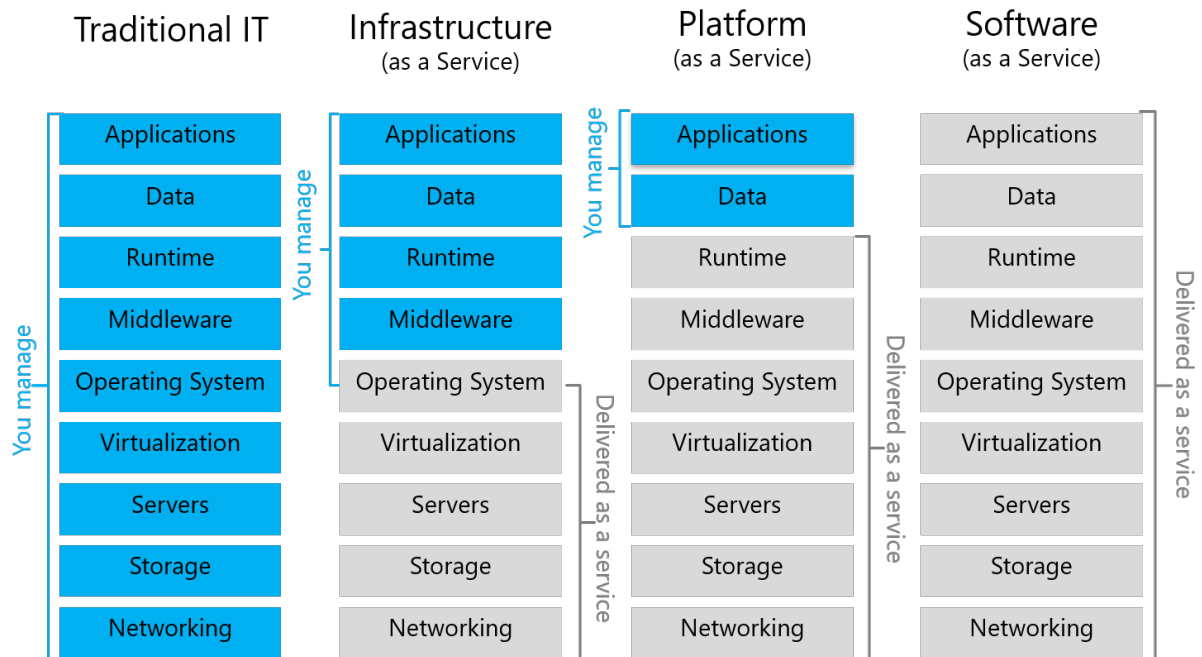
**Broad network access:** Auf Leistungen wird über das normale Internet mit standardmäßigen Mechanismen wie der Nutzung von Thin Clients und Fat Clients (Smartphones, Tablets, Laptops oder Workstations) zugegriffen.

**Resource Pooling:** Die Computing Ressourcen des Anbieters werden in einem gemeinsamen Pool für mehrere Kunden in einem Multi-Tenancy-fähigen Modell bereitgestellt, physische und virtuelle Ressourcen werden dynamisch zugewiesen und entsprechend der Nachfrage kontinuierlich neu verteilt. Es wird eine empfundene Ortsunabhängigkeit hergestellt indem der Nutzer kein genaues Wissen darüber besitzt wo sich dessen Ressourcen befinden, auf höherem Level wie beispielsweise dem Staat, der Region oder auch Rechenzentrum kann der Ort vom Nutzer spezifiziert werden. Die bereitgestellten Ressourcen beinhalten zum Beispiel Datenspeicher, Rechenleistung, Arbeitsspeicher 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 können zu jeder Zeit und in jedem Umfang bereitgestellt werden.

**Measured Service:** Cloud Systeme kontrollieren und optimieren Ressourcennutzung automatisch mithilfe eines Mess-Systems dass auf einer abstrakten Ebene den entsprechenden Service (Datenspeicher, Rechenleistung, Benutzerkonten, etc.) 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).



**Abb. 2.1:** Die Cloud Service Modelle im Überblick[3]

**Infrastructure as a Service:** Der Nutzer hat die Fähigkeit Rechenleistung, Datenspeicher Netzwerke und weitere fundamentale Computing Ressourcen 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 eingekaufte oder selbst erstellten Anwendungen auf der Cloud Infrastruktur zu betreiben, die notwendige Umgebung die über Sprachen, Bibliotheken, Tools und Services verfügt wird vom Cloud Service Provider (CSP) bereitgestellt. Die darunter liegende Infrastruktur mit Netzwerken, Servern, Betriebssystemen und Speicher wird vom CSP betrieben, der Nutzer hat die Kontrolle über Anwendung und Konfiguration der Umgebung in der die Anwendung betrieben wird.

**Software as a Service:** Dem Nutzer wird der Zugriff auf die vom CSP in der Cloud Infrastruktur betriebenen Softwareanwendung gewährt. Auf diese wird mithilfe eines Thin oder Fat Client zugegriffen, dabei kümmert sich der Nutzer nicht um den Betrieb und die Konfiguration der darunterliegenden Cloud Infrastruktur (Netzwerke, Server, Betriebssystem,

Speicher) und die Anwendung selbst mit Ausnahme eingeschränkter Nutzereinstellungen.

In Art der Bereitstellung eines Cloud Services werden vier grundlegende Modelle unterschieden; es existieren 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-Premises<sup>1</sup> 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-Premises 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-Premises 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 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

---

<sup>1</sup> Hardware ist lokal vor Ort/nicht vor Ort



Unternehmen selbst eingekauft, eingerichtet und betrieben werden muss (Abb. 2.1). Potentiell hohe Kosten die bereits vor der Inbetriebnahme eines Systems mit einem höheren Risiko aufgewendet werden müssen stellen in Form von individuell geringeren laufenden Beträgen ein deutlich reduziertes Risiko dar[4].

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[5].

Die Gesamtkostenersparnisse stehen auch im Zusammenhang mit Skaleneffekten<sup>1</sup> die für große Cloud Service Provider gelten. Der Betrieb eines einzelnen Servers ist im Verhältnis mit bedeutend höheren Kosten verbunden als das hinzufügen eines äquivalenten Systems zu einem Rechenzentrum im Betrieb von Microsoft oder einem vergleichbaren Anbieter[6].

**Skalierbarkeit:** Besonders für schnell wachsende Unternehmen ist die Möglichkeit schneller 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[7].

**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[8].

**Security:** Sicherheitsaspekte können sowohl einen Vor- als auch Nachteil von Cloud Computing darstellen. Hier sollen zuerst Vorteile dargelegt werden, potentielle Probleme sind im nächsten Abschnitt beschrieben.

Die technischen Möglichkeiten und besonders auch die Wahrnehmung des Themas Sicherheit in der Cloud unterlagen und unterliegen auch noch immer einem deutlichen Wandel. Cloud Anbieter investieren viele Ressourcen in Sicherheit und stellen dem Nutzer zum Beispiel bereits sicher implementierte Verschlüsselungen zur Verfügung oder bieten einen gewissen Schutz vor Denial-of-Service Angriffen durch ihre „natürliche“ Skalierbarkeit[9].

---

<sup>1</sup> Zusammenhang zwischen produziertem Ertrag und eingesetzter Ressourcen

### Herausforderungen und Risiken

**Netzwerkabhängigkeit:** Da der Zugriff auf Cloud Dienste über das Internet erfolgt entsteht dadurch entsprechend auch eine hohe Abhängigkeit. Stabile und schnelle Netzwerkanbindung ist eine kritische Voraussetzung für effektives Arbeiten[10], 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 einzelnen Anbieters zu begeben. Im Fall einer Änderung der Nutzungsbedingungen oder einer Änderung im Kostenmodell die den eigenen 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 Bedingungen zu akzeptieren[12].

**Security und Privacy:** Sicherheitsrisiken sind einer der meistgenannten Gründe die gegen Cloud Computing sprechen[11], besonders im Fall der Nutzung einer Public Cloud. Die Gefahr dass Daten in die Hände dritter gelangen kann zum beispielsweise nicht vollständig ausgeschlossen werden. Da die Verantwortung über die Sicherheit der Daten dem Cloud Anbieter unterliegt kann es auch zu Problemen hinsichtlich Privatheit der Daten kommen, sollte etwa eine Regierungsorganisation Zugriff auf bestimmte Daten eines Nutzer verlangen könnte dieser ohne dessen Einverständnis gewährt werden.

**Kosten:** Auch wenn die Nutzung von Cloud Computing mit dem Vorteil geringerer Kosten beworben wird, ist dies nicht zwingend garantiert. Werden die vorhandenen Systeme ungünstig verwendet, bleiben zum Beispiel viele gebuchte CPUs und IP-Adressen sowie weitere Ressourcen ungenutzt, können unnötig hohe Kosten entstehen. Auch während der Migrationsphase, in der möglicherweise beide Systeme parallel betrieben werden müssen, können höhere Kosten entstehen als in einem vergleichbaren Zeitraum davor. Nicht zuletzt stellt die Migration komplexer Systeme häufig eine große technische und dadurch auch finanzielle Herausforderung dar[13].

#### 2.1.3 Überblick über die wichtigsten Cloud Service Provider

In diesem Abschnitt soll ein knapper Überblick über die wichtigsten Public Cloud Service Provider gegeben werden. Abbildung 2.2 stellt den Marktanteil der aktuell umsatz-

stärksten Cloud Service Provider sowie das globale Umsatzwachstum insgesamt dar. Zu beachten ist dass in dieser Abbildung das Service Modell SaaS nicht beinhaltet ist.

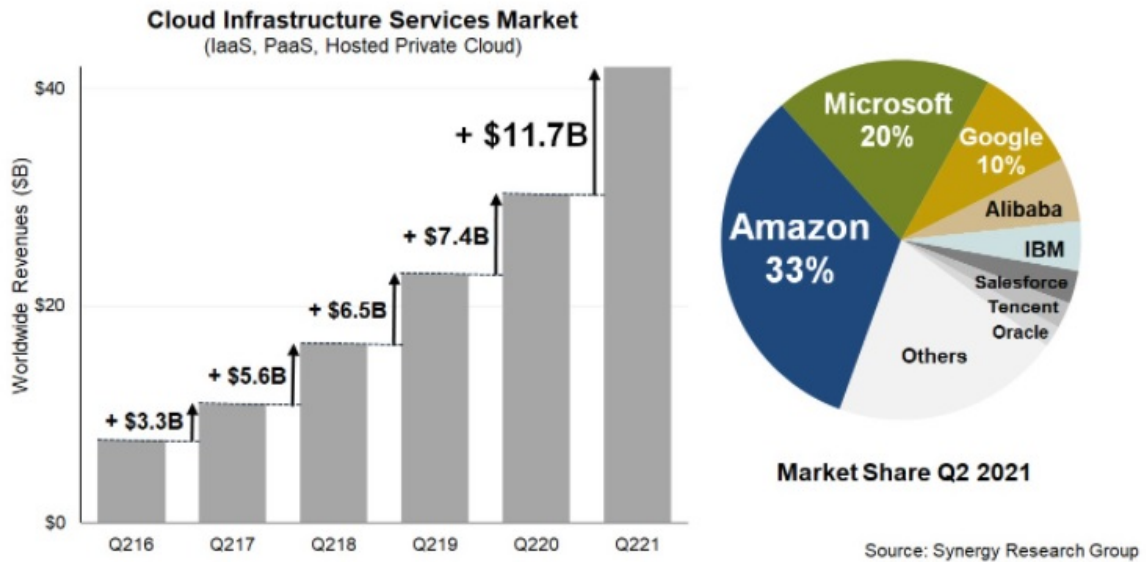


Abb. 2.2: CSP Market Share Q2 2021 nach Umsatz[14]

Aus Abb. 2.2 wird ersichtlich, dass Amazon, Microsoft und Google gemeinsam mehr als die Hälfte des Marktes beherrschen, Amazon allein stellt etwa ein Drittel aller Infrastruktur Services bereit.



**Abb. 2.3:** CSP Gartner Magic Quadrant Juli 2021[15]

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 Vormachtstellung von Amazon gegenüber Microsoft und Google, auffällig ist auch die Stellung von IBM die auf ein mögliches Problem in der Ausgereiftheit der IBM Cloud hinweisen könnte.

## 2.2 Infrastructure as Code

Automatisierung spielt in der modernen Softwareentwicklung eine wichtige Rolle. Automatisierte Unit- und Integration-Tests gehören inzwischen schon lange zum Standardrepertoire eines Softwareentwicklers. Die Verbreitung der DevOps-Kultur förderte in den

letzten Jahren auch die Automatisierung des Build- und Deployment-Prozesses und mit der steigenden Verbreitung Cloud-basierter Software erweitert sich diese Automatisierung nun auch auf das Provisioning der Infrastruktur.

Infrastructure as Code beschreibt im allgemeinen einen Ansatz zur Automatisierung des Infrastruktur-Provisionings basierend auf Methoden aus der Softwareentwicklung[16].

Statt eines manuellen Aufbaus und händischer 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[17].

### 2.2.1 Technologischer Wandel und das Cloud Age Mindset

Durch die Technologien der Cloud ist es heute möglich eine gewünschte IT-Infrastruktur sehr viel schneller bereitzustellen als zuvor. Statt Einkauf, Anschließen und Einrichten eines physischen Servers das, je nach Szenario einen Zeitraum von mindestens mehreren Stunden oder Tagen bis hin zu Wochen dauern kann, können virtuelle Ressourcen in der Cloud in wenigen Minuten verfügbar gemacht werden. Der schnellere Ablauf wird durch die Automatisierung von Prozessen wie der Bereitstellung der Infrastruktur mithilfe von IaC Tools noch verstärkt. Mit diesen Veränderungen wird das Management und die Erweiterung der bestehenden Systeme jedoch nicht unbedingt einfacher[18], die Anwendung von Arbeitsprozessen die sich bisher bewährt haben sind oftmals ineffektiv und verhindern das Ausnutzen des vollen Potentials der Cloud. Kief Morris, Autor von *Infrastructure as Code - Dynamic Systems for the Cloud Age*, stellt die fundamentalen Unterschiede des Arbeitens mit Cloud-Technologien mithilfe der folgenden Tabelle (Tab. 2.1) 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

**Tab. 2.1:** „Iron vs Cloud Age“[17]

Veränderungen im „Iron Age“ sind aufwändig und teuer und stellen ein Risiko dar, es wird versucht diese Risikopunkte zu reduzieren, daher werden viele Veränderung gebündelt getestet und eingeführt wodurch lange Release-Zyklen entstehen. Die Architekturen die dadurch befördert werden sind monolytisch, die Konfiguration erfolgt eher mithilfe von GUI-gesteuerten Programmen oder direkter physischer Interaktion, 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 das „Iron Age“ zugeschnittene Arbeitsweise für die Cloud nicht effektiv ist. Ein neues „Cloud Age Mindset“ das die rechte Spalte der Tabelle verinnerlicht ist erforderlich um die Vorteile der Cloud wirklich effektiv und in vollem Umfang nutzen zu können.

### 2.2.2 Vorteile von Infrastructure as Code im Vergleich zu manuellem Infrastruktur-Provisioning

**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 Umgebungen oftmals auch leicht verschiedene Anforderungen an diesen Server. Auf diese kann dann mit Optimierungen, etwa in Form spezifischer Konfigurationsdetails, reagiert werden. Wird nun das ursprüngliche Basis-System geupdated werden individuelle und oft undokumentierte Anpassungen nicht berücksichtigt wodurch ein Update unbequeme Konsequenzen nach sich ziehen kann. Werden alle Veränderungen in einer einheitlichen Codebase verwaltet und Updates häufig vorgenommen verhindert man starken Configuration Drift der ansonsten oftmals stattfindet[19].

**Wiederverwendbarkeit durch einheitlichen Code:** Ein weiterer Vorteil der durch die Verwendung einer einheitlichen Codebase entsteht ist die Wiederverwendbarkeit und Reproduzierbarkeit eines Systems. Wenn ein identisches System an einer anderen Stelle aufgebaut werden soll oder geht ein System aus unvorhergesehenen Gründen in seiner Gesamtheit verloren 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[20].

**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 weiteren Beteiligten Teams:**

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 werden. 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 Herausforderungen im Einsatz von Infrastructure as Code****2.2.4 Die drei Kernverfahren von Infrastructure as Code****2.3 Funktionsprinzip und Rolle von Terraform im IaC-Anwendungsprozess****2.3.1 Funktionsweise von Terraform****2.3.2 Einführung in die Hashicorp Configuration Language**



## 3 Evaluierungsanforderungen und Umsetzung

### 3.1 Evaluierungsanforderungen

#### 3.1.1 Ziel der Evaluierung

#### 3.1.2 Untersuchte Komponenten der Terraform Provider

#### 3.1.3 Auswahl der Evaluierungskriterien

### 3.2 Umsetzung des Testsystems

#### 3.2.1 Eingesetzte Software und Tools

#### 3.2.2 High-Level Aufbau des Testsystems

#### 3.2.3 Konkreter Aufbau auf Google Cloud Platform

#### 3.2.4 Konkreter Aufbau auf Azure

#### 3.2.5 Aufbau der ergänzenden Versuche

## 4 Ergebnisse und Bewertung

4.1 Evaluierung der Functional Completeness

4.2 Ergebnisse und Bewertung der Time Behaviour Tests

4.3 Ergebnisse und Bewertung der Recoverability Tests

4.4 Ergebnisse und Bewertung der Modifiability Tests

4.5 Evaluierung der Einheitlichkeit der Testsysteme für Azure und Google Cloud Platform

# 5 Schluss

## 5.1 Fazit

## 5.2 Ausblick

öö

## A Kapitel im Anhang

# Literaturverzeichnis

- [1] Asadi, S., Nilashi, M., Husin, A.R.C. et al. - Customers perspectives on adoption of cloud computing in banking sector. Inf Technol Manag 18, 305?330 (2017). [Online] Verfügbar unter: <https://doi.org/10.1007/s10799-016-0270-8> (Zugriff am: 23.01.2022)
- [2] Mell, Peter, and Tim Grance. - "The NIST definition of cloud computing."(2011). [Online] Verfügbar unter: <http://faculty.winthrop.edu/domanm/csci411/Handouts/NIST.pdf> (Zugriff am: 28.10.2021)
- [3] Chou, David. Cloud Service Models (IaaS, PaaS, SaaS) Diagram (2018). [Online] Verfügbar unter: <https://dachou.github.io/2018/09/28/cloud-service-models.html> (Zugriff am: 28.10.2021)
- [4] Lisdorf, Anders. - Cloud computing basics a non-technical introduction. Berkeley, CA: Apress, 2021. S.23.
- [5] Lisdorf, Anders. - Cloud computing basics a non-technical introduction. Berkeley, CA: Apress, 2021. S.24.
- [6] Lisdorf, Anders. - Cloud computing basics a non-technical introduction. Berkeley, CA: Apress, 2021. S.24-25.
- [7] Lisdorf, Anders. - Cloud computing basics a non-technical introduction. Berkeley, CA: Apress, 2021. S.27.
- [8] Lisdorf, Anders. - Cloud computing basics a non-technical introduction. Berkeley, CA: Apress, 2021. S.26.
- [9] Lisdorf, Anders. - Cloud computing basics a non-technical introduction. Berkeley, CA: Apress, 2021. S.25.

- [10] NANDGAONKAR, Suruchee V.; RAUT, A. B. A comprehensive study on cloud computing. International Journal of Computer Science and Mobile Computing, 2014, 3. Jg., Nr. 4, S. 737.
- [11] NANDGAONKAR, Suruchee V.; RAUT, A. B. A comprehensive study on cloud computing. International Journal of Computer Science and Mobile Computing, 2014, 3. Jg., Nr. 4, S. 737.
- [12] OPARA-MARTINS, Justice; SAHANDI, Reza; TIAN, Feng. Critical analysis of vendor lock-in and its impact on cloud computing migration: a business perspective. Journal of Cloud Computing, 2016, 5. Jg., Nr. 1, S. 1-18. [Online] Verfügbar unter <https://ieeexplore.ieee.org/abstract/document/7009018> (Zugriff am: 24.01.2022)
- [13] BAI, Kun, et al. What to discover before migrating to the cloud. In: 2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013). IEEE, 2013. S. 320. [Online] Verfügbar unter: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6573001> (Zugriff am: 24.01.2022)
- [14] Kein Autor. Enterprise spending on cloud infrastructure services grows to USD 42 billion in Q2, 2021: SRG (2021) [Online] Verfügbar unter: <https://www.cioandleader.com/article/2021/07/30/enterprise-spending-cloud-infrastructure-services-grows-usd-42-billion-q2-2021> (Zugriff am: 24.01.2022)
- [15] Bala, Raj; Gill, Bob; Smith, Dennis; Ji, Kevin; Wright, David. Magic Quadrant for Cloud Infrastructure and Platform Services (2021) [Online] Verfügbar unter: <https://www.gartner.com/doc/reprints?id=1-2710E4VR&ct=210802&st=sb> (Zugriff am: 24.01.2022)
- [16] Kief Morris. Infrastructure as Code. O'Reilly Media, 2. edition, 2020. S.4
- [17] Jacobs, Mike; Kaim, Ed. What is Infrastructure as Code? (2021) [Online] Verfügbar unter: <https://docs.microsoft.com/en-us/devops/deliver/what-is-infrastructure-as-code> (Zugriff am: 24.01.2022)
- [18] Kief Morris. Infrastructure as Code. O'Reilly Media, 2. edition, 2020. S.3
- [19] Kief Morris. Infrastructure as Code. O'Reilly Media, 2. edition, 2020. S.17-18
- [20] Kief Morris. Infrastructure as Code. O'Reilly Media, 2. edition, 2020. S.16