FERDINAND PORSCHE
FERN FH

Nutzwertanalyse der verwalteten Kubernetes Services im Bereich der Hyperscaler (AKS, GKE und EKS)

Bachelorarbeit

eingereicht von: **Caglayan Baysal**
Matrikelnummer: 52006047

im Fachhochschul-Bachelorstudiengang Wirtschaftsinformatik (0470)
der Ferdinand Porsche FernFH

zur Erlangung des akademischen Grades <einer/eines>

Bachelor of Arts in Business

Betreuung und Beurteilung: DI Eszter Geresics-Földi, BSc MSc

Wiener Neustadt, Mai 2023

FERDINAND PORSCHE
FERN FH

Ehrenwörtliche Erklärung

Ich versichere hiermit,

1. dass ich die vorliegende Bachelorarbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Alle Inhalte, die direkt oder indirekt aus fremden Quellen entnommen sind, sind durch entsprechende Quellenangaben gekennzeichnet.
2. dass ich diese Bachelorarbeit bisher weder im Inland noch im Ausland in irgendeiner Form als Prüfungsarbeit zur Beurteilung vorgelegt oder veröffentlicht habe.

Wien, 12. Mai 2023

Unterschrift

Creative Commons Lizenz

Das Urheberrecht der vorliegenden Arbeit liegt bei Caglayan Baysal. Sofern nicht anders angegeben, sind die Inhalte unter einer Creative Commons <„Namensnennung - Nicht-kommerziell - Weitergabe unter gleichen Bedingungen 4.0 International Lizenz“ (CC BY-NC-SA 4.0)> lizenziert.
Die Rechte an zitierten Abbildungen liegen bei den in der jeweiligen Quellenangabe genannten Urheber: innen.

Die Kapitel 1 bis 3 der vorliegenden Bachelorarbeit wurden im Rahmen der Lehrveranstaltung „Bachelor Seminar 1“ eingereicht und am 06.02.2023 als Bachelorarbeit 1 angenommen.

Kurzzusammenfassung: Nutzwertanalyse der verwalteten Kubernetes Services im Bereich der Hyperscaler (AKS, GKE und EKS)

Diese wissenschaftliche Arbeit beschäftigt sich mit der Frage, welcher verwaltete Kubernetes Service, in ausgewählten Funktionen, wie CLI-Unterstützung; Spawn-Cluster Zeit; Kubernetes Versionsunterstützung; Monitoring; rollenbasierte Zugriffssteuerung; Überwachung der Knotenintegrität und Preisgestaltung die höchsten Nutzen erzielen. Der Grund für die Forschungsthesis sind die persönlichen Erfahrungen des Autors, um die Kundenanforderungen und die IT-Kompetenz der Kunden IT-Teams mit den Cloud Providern an erster Linie zu vergleichen und das IT-Team des Kunden und den Kunden selbst, auf die passende Cloud Enterprise Lösung zu unterstützen.

Um diese Forschungsfrage zu beantworten, wurde zuerst eine Literaturrecherche geführt und anschließend die jeweiligen Kubernetes Services als Prototyp bereitgestellt und auf die ausgewählten Funktionen analysiert. Anschließend wurde mittels einer Nutzwertanalyse, das Nutzen der einzelnen Funktionen bewertet.

Als Ergebnis stellte sich heraus, dass Google Kubernetes Engine die höchsten Nutzen erzielt und Azure Kubernetes Service ein sehr präziser Nachfolger von Google Kubernetes Engine ist, während Elastic Kubernetes Service die Rangfolge drei in bester Position, mit geringer Knappheit verteidigt.

Während der Bereitstellung von verwalteten Kubernetes Services stellte sich der Spawn-Cluster Zeit, als die größte Herausforderung. Vor allem die Bereitstellung des EKS-Clusters dauerte knapp 17 Minuten und dauerte um das Dreifache mehr an Zeit, bis der Cluster bereitgestellt worden ist in der Cloud Umgebung.

Schlagwörter:

Cloud Computing; Prototyping; AKS; EKS; GKE; Kubernetes, Nutzwertanalyse

Abstract: utility value analysis of the managed Kubernetes services in the area of hyperscalers (AKS, GKE and EKS)

This thesis addresses the question of which managed Kubernetes service provides the highest benefits in selected features. The motivation here was the authors personal experiences

in comparing customer requirements and IT competency of customer IT teams with cloud providers and supporting customers and their IT teams in choosing the appropriate cloud enterprise solution.

To answer the research question, a literature review was conducted, and the respective Kubernetes services were prototyped and analyzed for the selected features. Then, a utility analysis was used to evaluate the benefits of each feature.

The results showed that Google Kubernetes Engine provided the highest benefits, followed by Azure Kubernetes Service and Elastic Kubernetes Service.

The biggest challenge by the prototyping was the preparing of the EKS-Cluster. It took 17 minutes to provision, which was three times longer than other cloud environments.

Keywords:

Cloud Computing; Prototyping; AKS; EKS; GKE; Kubernetes, utility value analysis

Inhaltsverzeichnis

1. Einleitung

1.1 Ausgangssituation

Die Entwicklung der Technologie in der IT, hat dazu beigetragen, dass viele Unternehmen Ihre Anwendungen via Cloud Computing schnell in den Markt bringen. Dadurch ermöglicht die Cloud, dass Speichern von Daten und das Ausführen von Programmen, aus jedem Gerät, welches mit dem Internet verbunden ist. Unternehmen können anhand der Technologie viele Vorteile genießen. Vor allem können Sie die Anwendungen individuell anpassen und ihre Services dementsprechend auf die Anforderungen skalieren und sind flexibler als je zu vor.

Ausgeschlossen vom Unternehmensziel ist ein wichtiger Faktor, die Unternehmenseffizienz. Unternehmen müssen fürs Erste sich keine Gedanken um die Kosten oder um die Wartung der Architektur und der Infrastruktur von ihnen in den Markt eingeführten Anwendungen machen. Das Konzept von Cloud Computing ist es, dass Unternehmen Remote-Ressourcen nutzen. Dadurch sparen Unternehmen bei Kosten für den Server und bei Kosten für weitere Geräte.

Ein weiterer Punkt der die Unternehmenseffizienz befürwortet ist, dass „Pay-as-you-go-Prinzip“. Bedeutet schlicht und ergreifend, dass Kundinnen und Kunden nur für die tatsächlich verwendeten Ressourcen bezahlen. Das Zahlungsmodell ist somit nutzungsabhängig. Angesichts der oben angeführten Vorteile, weist auch die Benutzung von innovativer Technologie, klarerweise einen strategischen Nutzen für viele Unternehmen. Unternehmen können schneller agieren und erwerben dadurch einen klaren Wettbewerbsvorteil. Die allgemeine Infrastruktur wird von Cloud-Service-Provider betrieben, daher entsteht die Möglichkeit für Unternehmen, dass sie sich auf ihre eigentlichen Prioritäten fokussieren und somit die Arbeit optimieren.

Die Hyperscaler, wie Amazon Web Services (AWS), Microsoft Azure und Google Cloud sind die weltweit bekanntesten Anbieter von IT-Ressourcen bezogen auf Cloud Computing. Obwohl viele der Funktionen von Hyperscaler gleich aufgebaut sind, unterscheiden sie sich dennoch in einigen Anwendungen, wie zum Beispiel die verwaltete Kubernetes Services. Kubernetes ist ein Container-Orchestrierungstool und kann als ein Betriebssystem für die Cloud bezeichnet werden. Die Open-Source-Plattform dient zur Automatisierung der Bereitstellung, Skalierung und Verwaltung von Container-Anwendungen.

1.2 Zielsetzung der Arbeit

Das Ziel dieser Arbeit ist es, anhand einer Nutzwertanalyse einen groben Überblick, für die Entscheidung der verwalteten Kubernetes Dienstleistungen zu verschaffen und weiters die Erkenntnisse darüber zu erlangen, ob die Entscheidungsmethode, die Ausführung einer Nutzwertanalyse dafür gut geeignet ist.

Die Zielgruppe dieser Bachelorarbeit sind jene die eine Verknüpfung zur Cloud haben. Diese sind beispielsweise Cloud- Beraterinnen und Berater, -Entwicklerinnen und Entwickler, Data Engineer, -Architekten und allgemein Unternehmen, die Ihre Organisation mit dieser Innovation bereitstellen und von diesem Wettbewerbsvorteil profitieren möchten.

1.3 Forschungsfrage

Die Forschungsfrage, welche im Rahmen dieser Bachelorarbeit herauszufinden ist, lautet:

Welche der drei Hyperscaler (Azure, AWS und Google Cloud), in Bezug auf die Funktionen (CLI-Unterstützung; Spawn-Cluster Zeit; K8s Versionsunterstützung; Monitoring; rollenbasierte Zugriffssteuerung; Überwachung der Knotenintegrität und Preisgestaltung) erzielen den höchsten Nutzen?

1.4 Die Hypothese

Bezogen auf die Funktionen der verwalteten Kubernetes Services (CLI-Unterstützung; Spawn-Cluster Zeit; K8s Versionsunterstützung; Monitoring; rollenbasierte Zugriffssteuerung; Überwachung der Knotenintegrität und Preisgestaltung) erzielt Google Cloud den höchsten Nutzen.

1.5 Methode

Für die Beantwortung der Forschungsfrage, werden in allen drei Cloud – Provider (Azure, AWS und Google Cloud) eine Cloud Umgebung bereitgestellt, die exakt dieselben Anforderungen erfüllen. Die Prototyp – Modelle werden Schritt für Schritt in der Arbeit dargestellt und mithilfe der Prototyp – Modellen werden die jeweiligen Cloud Umgebungen auf die angeführten Funktionen wie, CLI-Unterstützung; Spawn-Cluster Zeit; Kubernetes Versionsunterstützung; Monitoring; rollenbasierte Zugriffssteuerung; Überwachung der Knotenintegrität

und Preisgestaltung untersucht und bemessen. Gegebenenfalls werden die Prototypen mit der Dokumentation des jeweiligen Hyperscalers ergänzt.

Die einzelnen Funktionen werden je nach ihrer Priorität und Wichtigkeit für die Cloud – Umgebung in der Tabelle der Nutzwertanalyse gewichtet und dargestellt. Je nach Gewichtung und Wichtigkeit für die Cloud – Umgebung werden diese bewertet.

1.6 Aufbau der Arbeit

Im ersten Kapitel der Bachelorarbeit befasse ich mich mit den Hintergrundinformationen, wie zum Beispiel, wie die Ausgangssituation aktuell ist; wie die Forschungsfrage entstanden ist; was das Ziel dieser Arbeit ist; mit welcher Methodik ich die Forschungsfrage beantworten möchten und für welche Zielgruppen diese Arbeit opportun werden könnte.

Kapitel zwei der Arbeit befasst sich mit den Basics. Hier werden die Grundlagen des Cloud-Computings geschildert und wichtige Begriffsdefinitionen erklärt, die relevant für die Nachvollziehbarkeit der Arbeit sind.

Anschließend folgt die Definition der Hyperscaler und Basisinformationen über Azure, AWS und Google Cloud Platform, danach befasse ich mich mit der Container Orchestrierung und diversen Orchestrierungstools und schließe das Kapitel mit Fokus auf Kubernetes, dessen Basis Architektur und dessen verwaltete Kubernetes Services: Azure Kubernetes Service; Amazon Elastic Kubernetes Service und Google Kubernetes Engine ab.

Nach Bezug auf dessen Aktualität auf Stand der Technik und Stand der Wissenschaft, wird im letzten Abschnitt von Teil 1 der Bachelorarbeit die Grundlagen der Nutzwertanalyse, die Erläuterung wichtiger Begriffsdefinitionen und die Beschreibung der Vorgehensweise in meinem Fall, näher erklärt.

Der zweite Teil der Bachelorarbeit beginnt mit der Differenzierung der Funktionen von verwalteten Kubernetes Services, die im Rahmen der Arbeit, für die Beantwortung der Forschungsfrage relevant sind. Im Anschluss dieser Differenzierung folgt die Begriffsdefinition der einzelnen Funktionen.

Nach der Definition findet die Beschreibung der Methode und anschließend die Beschreibung der Vorgehensweise. Das Kapitel wird mit Prototypen der verwalteten Kubernetes Services und mit der Darstellung der jeweiligen Funktionen in AKS, EKS und GKE beendet.

Das vorletzte Kapitel der Arbeit befasst sich mit dem Bewertungsmodell und der Ermittlung der höchsten Nutzen, die anhand des Assessments zu kalkulieren sind. Nachdem die Ergebnisse in einer Matrix dargestellt und evaluiert werden, folgt im letzten Kapitel ein Resümee der Forschungsfrage und Bewertung der Hypothese. Abschließend werden, im Rahmen der Bachelorarbeit, die weiteren Ausblicke vorgestellt.

2. Cloud Computing

Im Kapitel zwei der Arbeit werden die Grundlagen der gewählten Thematik erläutert. Dies beinhaltet im ersten Teil „Stand der Technik“ eine Definition der verwendeten Grundbegriffe. Im zweiten Teil befindet sich „Stand der Wissenschaft“. Anschließend findet die Überleitung zu den Cloud - Provider statt. Hierbei werden die Hyperscaler namens Azure, Amazon Web Service und Google Cloud Plattform genauer in Betracht gezogen.

2.1 Grundlagen

In folgendem Kapitel werden die Grunddefinitionen der verwendeten Begriffe geklärt. Dies beinhaltet zu Beginn die Definition einer Cloud allgemein, sowohl als allgemeine Begriffsdefinition als auch die Definition von der NIST „National Institute of Standards and Technolgy“. Danach werden die Hauptcharaktereigenschaften einer Cloud, die Bereitstellungsmodelle und die Cloud-Service Modelle untersucht. (Hurwitz et al. 2010)

2.2 Begriffsdefinition

Unter dem Begriff Cloud Computing ist allgemein klar bekannt, dass damit die nächste Evolutionsstufe des Internets gemeint ist. Die Cloud im Cloud Computing bietet also einen Service im Service an. Unter Service ist hierbei von der Rechenleistung bis hin zur Recheninfrastruktur, die Anwendungen und auch die Möglichkeit die Geschäftsprozesse bis hin zur persönlichen Zusammenarbeit eines Unternehmens zu verstehen, welche angeboten werden als eine Service. Dieser Service steht auch für jede Benutzerinnen und Benutzer, wann und wo die Benutzerinnen und Benutzer dies haben möchte, als Service zur Verfügung. (Hurwitz et al. 2010)

2.2.1 Die Definition von Cloud

Die Cloud selbst besteht aus einer Reihe von Hardware, wie die Netzwerke, Speicher, Dienste und Schnittstellen, die die Bereitstellung von Datenverarbeitungen als eine Dienst den Cloud-Usern ermöglichen. Somit ist zu verstehen, dass die Cloud-Dienste die Bereitstellung von Software, Infrastruktur, und dem Speicher, über das Internet, wie zum Beispiel entweder als eine separate Komponente oder als eine vollständige Plattform, dem User je nach Bedarf zur Verfügung steht zu betrachten. (Hurwitz et al. 2010)

Eine weitere Definition der Cloud, welche formell beschrieben ist und für eine einfache Erklärung der Definition dient, ist die der von „National Institute of Standards and Technology“ (NIST).

Auch wenn die Definition einer Cloud von der NIST, sich im Laufe der Zeit immer weiterentwickelt hat, wird dessen Beschreibung einer Cloud immer noch als die, der Standard betrachtet. (Reciprocity 2021)

Die NIST-Cloud-Definition besteht aus drei Hauptkomponenten, diese wären wie gefolgt:

- wichtige Cloud-Eigenschaften
- Cloud-Bereitstellungsmodelle
- Cloud-Service-Modelle

2.2.2 Wichtige Cloud Eigenschaften

Im Laufe der Zeit bildeten sich durch die Popularität der Cloud sehr viele false friends. Eines dieser false friends ist es gewesen von der Popularität der Cloud zu profitieren, darunter ist zu verstehen, dass Unternehmen und Dienstleister behaupteten Cloud-Dienste anzubieten, weil eine Anwendung webbasiert gewesen ist. Doch dabei könnte es sich nicht um eine Cloud-Anwendung befasst haben, denn damit eine die Anwendung selbst und der Service rund um die Anwendung, als eine echte Cloud-Implementierung betrachtet werden kann, muss sie bestimmte Merkmale aufweisen. Diese Merkmale sind in der NIST-Definition von Cloud Computing über die fünf Haupteigenschaften einer Cloud abgedeckt. Laut NIST wird erst dann von einer Cloud gesprochen, wenn dessen Anwendungen mit diesen fünf Haupteigenschaften harmonisieren. (Rountree and Castrillo 2014)

Diese lauten:

- On-Demand-Self-Service
- Broad network access
- Resource pooling
- Elasticity
- Measured service

OnDemand Self – Service

Unter dem Begriff OnDemand-Self-Service ist zu verstehen, dass ein User Zugriff auf ein Serviceangebot anfordern und erhalten darf, ohne dass ein Administrator oder eine IT- Supporter die Anfrage manuell bearbeiten und ermöglichen muss. Im nachfolgenden Kapitel wird erläutert, welche Vorteile sich in einer OnDemand Self - Service einer Cloud hervorheben.

Das heißt, die Anfrageprozesse und Erfüllungsprozesse einer „OnDemand“ sind alle automatisiert. Dies bietet sowohl für den Anbieter als auch für den Nachfrager des Dienstes den Vorteil, dass der Service automatisch bereitgestellt wird durch die eigene Freigabe der Funktion. Durch die Implementierung von User-Self-Service können Kundinnen und Kunden die gewünschten Dienste schnell erwerben und auch darauf zugreifen, aus diesem Grund ist dies auch als ein sehr attraktives Merkmal der Cloud angesehen. Dadurch sind auch die benötigten Ressourcenzugriffe sehr schnell und einfach.

Ein weiterer wichtiger Punkt ist, dass die Eigenschaft User Self-Service, den Verwaltungsaufwand für den Provider reduziert. Administratoren dieser Tätigkeit müssen sich nicht ständig, um das Erstellen von Benutzerinnen und Benutzern und das Verwalten von Benutzeranfragen quälen und können ihre tagtäglichen Aktivitäten befolgen, wie zum Beispiel Bearbeitung der User Prioritäten oder Restriktionen.

Der User-Self-Service wird in normalen Fällen über ein Benutzerportal durchgeführt. Es gibt mehrere einsatzbereite Benutzerportale, die sofort verwendet werden können, um die erforderliche Funktionalität bereitzustellen, aber in gewissen Fällen wird ein benutzerdefiniertes Portal benötigt.

Bei der Implementierung von User-Self-Service ist immer zu bedenken, dass sich eine potenzielle Nachgiebigkeit und ein regulatorisches Problem begeben kann. Compliance-Programme wie Sarbanes Oxley (SOX) erfordern häufig Kontrollen, Kontrollen, um zu verhindern, dass einzelne Benutzerinnen und Benutzer bestimmte Dienste nutzen oder bestimmte Aktionen ohne Genehmigung ausführen kann. Dies führt dazu, dass einige Prozesse nicht vollständig automatisiert werden können. Es ist wichtig, dass die Userinnen und User, welcher eine User-Self-Service bereitstellt, auch versteht, welche Prozesse bei der Implementierung von Self-Service in einer Cloud - Umgebung automatisiert werden können und welchen die Rechte und Zugriffe fehlen. (Rountree and Castrillo 2014)

Broad Network Access

Sinngemäß wird in der Cloud gestrebt, dass Userinnen und User auf ihre Ressourcen überall aus und jederzeit zugreifen können sollen. Im Kapitel „Broad Network Access“ wird die zweite Haupteigenschaft einer Cloud, laut NIST wiedergeben und ebenso dessen Nachvollziehbarkeit des breiten Netzwerkzuganges beschrieben.

Ein Cloud-Dienst sollte permanent leicht zugänglich sein, weil viele Benutzerinnen und Benutzer nur über eine grundlegende Netzwerkverbindung verfügen, um eine Verbindung zu den Diensten oder zu den Anwendungen herzustellen. In den meisten Fällen handelt es sich bei der verwendeten Verbindung um eine Art Internetverbindung. Obwohl Internetverbindungen in der Bandbreite zunehmen, sind sie im Vergleich dazu immer relativ langsam, wie LAN-Verbindungen (Local Area Network).

Summa summarum, ist dies auch schon der einzige schlagfertiger Grund, weshalb Provider nicht verlangen, dass Benutzerinnen und Benutzer über eine große Menge an Bandbreite verfügen sollen, um den Dienst zu nutzen.

Verbindungen mit begrenzter Bandbreite führen zum zweiten Teil dieser Anforderung:

Aus Gründen wie begrenzte Bandbreite sollten Cloud-Dienste entweder völlig auf Klienten freie Cloud-Dienst umstellen oder einen dünnen Klienten erfordern. Dünn im Sinne von, eines Netzwerk Klienten, welches wenige Ressourcen und Daten beim Verbinden zu einem Host oder einer Service benötigt.

Der Grund dieser Anforderung ist sehr einfach begründet, erstens kann das Herunterladen eines Klienten mit viel Ressourcen und Daten, sehr lange dauern, insbesondere bei einer Verbindung mit geringer Bandbreite.

Zweitens, wenn die Clientanwendung viel Kommunikation zwischen dem Clientsystem und den Diensten erfordert, können Benutzerinnen und Benutzer Probleme mit Latenz bei Verbindungen mit geringer Bandbreite haben.

Eine weitere Anforderung, die durch diese Komplexitäten einer Cloud entstanden ist, dass auf die Cloud-Dienste, von einer Vielzahl von Client-Geräten aus, zugegriffen können sollte. Laptops und Desktops sind nicht die einzigen Geräte, die zur Verbindung mit Netzwerken und dem Internet verwendet werden. Userinnen und User dieser Technologie haben das Bedürfnis auch über Tablets, Smartphones und eine Vielzahl anderer Optionen einen Zugriff zu erstellen. Daher müssen Cloud-Dienste all diese Geräte unterstützen.

Wenn der Dienst eine Client-Anwendung erfordert, muss der Anbieter möglicherweise plattformspezifische Anwendungen erstellen, wie Windows, Mac, iOS und Android.

Die Entwicklung und Wartung einer Reihe verschiedener Client-Anwendungen ist und kann kostspielig werden, daher ist es äußerst vorteilhaft, wenn die Cloud - Lösung so konzipiert werden kann, dass überhaupt kein Client erforderlich ist. (Rountree and Castrillo 2014)

Ressource Pooling

Die dritte Haupteigenschaft der NIST - Cloud - Definition, welche in diesem Kapitel geschildert wird, befasst sich hauptsächlich mit dem Fundamental des „PAY-AS-YOU-GO – Prinzips“.

Ressourcenpooling verhilft Kosten zu sparen und ermöglicht Flexibilität auf Seiten des Providers, das bedeutet, dass Ressourcen-Pooling auf die Tatsache basiert, dass Kundinnen und Kunden nicht ständig alle ihnen zur Verfügung stehenden Ressourcen benötigen. Dadurch lässt sich für den Provider der Cloud – Umgebung konstatieren, dass wenn Ressourcen nicht von einem Kundinnen und Kunden verwendet werden, der Provider diese Ressourcen, anstatt ungenutzt zu lassen, von einem anderen Kundinnen und Kunden, zu dessen Benutzung zur Verfügung stellen kann, um aus dieser Situation zu profitieren. Dies gibt Anbietern die Möglichkeit, viel mehr Kundinnen und Kunden zu bedienen, als sie es könnten, wenn jeder Kunde dedizierte Ressourcen benötigt. Eine solche Eigenschaft, wird häufig durch Virtualisierung erreicht.

Durch Virtualisierung können Cloud - Provider die Dichte ihrer Systeme erhöhen. Sie können mehrere virtuelle Sitzungen auf einem einzigen System hosten. In einer virtualisierten Umgebung werden die Ressourcen auf einem physischen System, in einem Pool platziert, der von mehreren virtuellen Systemen verwendet werden kann. (Rountree and Castrillo 2014)

Elasticity

In diesem Abschnitt wird die vorletzte Eigenschaft der NIST - Cloud - Definition beschrieben. Die Elastizität ist auch eigentlich auf einer herkömmlichen Umgebung verfügbar, aber hier wird versucht zu erklären, woran genau sie sich unterscheidet als die herkömmliche Umgebung einer Anwendung.

Schnelle Elastizität beschreibt die leicht zu wachsende Fähigkeit einer Cloud-Umgebung, um die Benutzeranforderungen zu erfüllen. Unter normalen Umständen besitzen Cloud-Bereitstellungen bereits die erforderliche

Infrastruktur, um die Servicekapazität zu erweitern. Wenn das System richtig konfiguriert ist, beschreibt die Elastizität nur das Hinzufügen weiterer Computerressourcen, wie Festplatten, Arbeitsspeicher und dergleichen. Der Knackpunkt hierbei ist, dass die Ressourcen zwar verfügbar sind, aber erst verwendet werden, nach dem „PAY-AS-YOU-GO Konzept, wenn sie benötigt werden. Dadurch kann der Provider und auch die Userinnen und User Verbrauchskosten ersparen, wie zum Beispiel die Strom- und Kühlungskosten.

Die Elastizität in der Cloud kann entweder durch Bereitstellungen wie die der Künstliche Intelligenz oder durch Clusterbildung und Orchestrierungen erreicht werden. Wenn die Ressourcennutzung einen bestimmten Punkt erreicht, wird ein Auslöser ausgelöst.

Der Auslöser startet automatisch den Prozess der Kapazitätserweiterung. Sobald die Nutzung nachgelassen hat, schrumpft die Kapazität nach Bedarf, um sicherzustellen, dass keine Ressourcen verschwendet werden. Außerdem ermöglicht die schnelle Elastizität der Cloud, das Bewältigen der „Burst-Kapazität“. Ist eine erhöhte Kapazität, die nur für kurze Zeit benötigt wird, wie beispielsweise, wenn eine Organisation am Ende des Geschäftsquartals eine erhöhte Kapazität für die Auftragsabwicklung benötigt als die herkömmliche Kapazität, die normalerweise bei einer Auftragsabwicklung benutzt wird.

Auf einer herkömmlichen Umgebung, wie beispielsweise in einem Server, müsste das Unternehmen über interne Kapazitäten verfügen, um diese Belastung zu bewältigen. Höchstwahrscheinlich würde dies bedeuten, dass Ressourcen immer verfügbar sind in einem Unternehmen, aber hauptsächlich für nur einen Bruchteil der Zeit verwendet werden.

In einer Cloud-Umgebung hingegen, kann die Organisation von den Vorteilen der Public Cloud-Ressourcen für den kurzen Zeitraum Gebrauch machen und muss nicht immer intern verfügbar sein. (Rountree and Castrillo 2014)

Measured Service

Die letzte Eigenschaft der NIST - Cloud - Definition, um der Definition etwas näher zu treten, ist das „Measured Service“. Laut NIST, sind Cloud-Dienste in der Lage, die Nutzungen die gebraucht und benutzt worden sind zu messen.

Diese können anhand verschiedener Metriken wie verwendeter Zeit, verwendeter Bandbreite und verwendeter Daten quantifiziert werden. Die gemessene Information ermöglicht eigentlich die „PAY-AS-YOU-GO“-Funktion der Innovation.

Sobald eine geeignete Metrik identifiziert wurde, wird eine Rate bestimmt. Dieser Satz wird verwendet, um zu bestimmen, wie viel einem Kundinnen und Kunden in Rechnung gestellt werden soll. Auf diese Weise wird dem Kundinnen und Kunden nach Verbrauch abgerechnet. Unbenutzte Tage dieses Konzeptes, werden natürlich in der Rechnungserstellung berücksichtigt. Mit begründetem Bedarf wird bei einer Benutzung eine Funktion; eine integrierte Systemkonfiguration oder einer Anwendung, die weitere minimal erforderte Teile und oder Konfigurationen dieser Umgebung aktiviert. Nutzung dieser Art des Service werden als eine messbare Servicefunktion gesehen. Dies könnten beispielsweise die VM's; das VPN-Tunnel; der virtueller Arbeitsspeicher oder die erforderte Speicherkapazität sein.

Bei einem gemessenen Service ist es sehr wichtig, dass Sie die damit verbundenen Kosten verstehen. Verständlicherweise werden bei hohen Kosten, die Infrastruktur u. die Architektur einer Anwendung erneut durchdacht, wo und wie Kosten erspart werden könnten. Ansonsten wird es üblicherweise sein, dass in den ersten paar Monaten unwillkommene Gebühren zu zahlen sind, die überraschend hohe Kosten beinhalten. (Rountree and Castrillo 2014)

2.2.3 Cloud Deployment Models

Laut NIST gibt es vier verschiedene Bereitstellungsmodelle der Cloud, die sogenannten Cloud Deployment Modelle, Modelle die zu klassifizieren nach ihre Verwaltungsbereichen sind. In diesem Kapitel werden die Cloud-Bereitstellungsmodelle dargestellt, um die Aufteilung der Cloud Computing Dienste leichter nachvollziehen zu können.

Zu Beginn ist klarzustellen, dass im Grunde genommen Clouds, die primären Ergebnisse von Cloud Computing sind. Sie können in Cloud Computing auch als parallele und verteilte Systeme angesehen werden, wie zum Beispiel als Systeme, die physische und virtuelle Computer nutzen, um Anwendungen bereitzustellen und als einheitliche Computerressource dargestellt werden zu können. Es ist zu bedenken, dass jede Firma seine eigene Anforderungen hat, auf welche exakten Dienste sie aus einer Cloud zugreifen möchten und wie viel Kontrolle sie über die Umgebung haben möchten. Die Antworten auf die Fragen „Welche Art von Cloud eine Organisation verwenden möchte?“ und „Auf welche Weise, eine Organisation die Tätigkeit ausüben möchte?“, ist sehr von der Organisation abhängig, da diese zum Schluss der Bereitstellung, die Entscheidungsträger sind. Um unterschiedlichen Anforderungen dieser Art, von vielen Organisation, Stand halten zu können, bietet die Cloud verschiedene

Bereitstellungsmodelle wie die Privat Cloud; Public Cloud; Hybrid Cloud und die Cloud Community. (Rountree and Castrillo 2014)

Private Cloud

In dieser Art der Cloud befinden sich die Systeme und Ressourcen, die die Anwendung bereitstellen, innerhalb der Organisation, welche dies benutzt. In einer Privat Cloud ist die Firma für das Management und die Verwaltung der verwendeten Systeme verantwortlich, sowie auch auf die Erbringung der eigenen Dienstleistung. Daraus resultiert sich, dass Betreiber von Privat Cloud, auch für alle Software- und Kundenanwendungen verantwortlich sind, die auf Endusersystemen sich befinden.

Der Zugriff auf eine Privat Cloud ist sehr stark abhängig wie sie bereitgestellt ist. Bei einer On-Premises Privat Cloud ist die Private Cloud vor Ort und Lokal gespeichert, ist die Private Cloud in einer Cloud so kann mittels Internet, über die Verwendung eines virtuellen privaten Netzwerks (VPN) zugegriffen werden. Andersrum wäre eine lokale LAN oder eine Wide Area Network (WAN) Zugriff zu befolgen.(Hurwitz et al. 2010)

Public Cloud

Public Clouds sind der erste Ausdruck von Cloud Computing. Sie sind eine Umsetzung des Cloud Computing, bei der die angebotenen Dienste jedem, von überall und zu jeder Zeit über das Internet zur Verfügung gestellt werden.

Aus struktureller Sicht handelt es sich um ein verteiltes System, das höchstwahrscheinlich aus einem oder mehreren miteinander verbundenen Rechenzentren besteht, auf denen die spezifischen Dienste implementiert sind, die von der Cloud angeboten werden.

Kundinnen und Kunden können sich einfach beim Cloud-Anbieter anmelden, ihre Zugangsdaten und Rechnungsdaten eingeben und die angebotenen Dienste nutzen. Public Clouds dienen eigentlich zur Minimierung der IT-Infrastrukturkosten und als einfach zu praktizierbare Option, zur Bewältigung von Lasten in der lokalen Infrastruktur. Sie sprechen hauptsächlich kleine Unternehmen an, die in der Lage sind, ihre Geschäfte, ohne große Investitionen zu starten, indem sie ihre IT-Anforderungen komplett auf die öffentliche Infrastruktur umstellen oder verlassen. Der größter Vorteil hierbei ist es gewesen, die Fähigkeit zu besitzen, Ressourcen entsprechend den Anforderungen des zugehörigen Geschäfts zu wachsen oder zu schrumpfen zu lassen, indem Firmen die Infrastruktur mieten oder abonnieren.

Ein grundlegendes Merkmal von Public Clouds ist die Mandantenfähigkeit. Dies ist eine grundlegende Voraussetzung, um eine effektive Überwachung der Benutzeraktivitäten zu ermöglichen und die gewünschte Leistung zu gewährleisten.

Mit Benutzerinnen und Benutzern ausgehandelte Quality-of-Services-Attribute und QoS-Management sind sehr wichtige Aspekte der Public Cloud. Aus diesem Grund ist ein erheblicher Teil der Softwareinfrastruktur, auf die Überwachung der Cloud gewidmet. Eine Public Cloud kann jede Art von Dienst anbieten wie zum Beispiel eine Infrastruktur; eine Plattform oder die Anwendung selbst. (Hurwitz et al. 2010)

Hybrid Cloud

Dieses Modell der Bereitstellung einer Cloud, ist als eine Kombination aus zwei oder mehreren anderen Cloud-Modellen zu sehen. Zwar sind die Cloud-Modelle inhaltlich nicht miteinander vermischt, vielmehr ist jede Cloud die in einer Hybrid Cloud benutzt wird separat präpariert und alle miteinander verbunden. Bei dem hybriden Modell ist die Umgebung ziemlich komplex dargestellt, daher bietet sie aber auch mehr Flexibilität bei der Erfüllung der Ziele einer Firma.

Private Clouds dagegen sind die perfekte Lösung, wenn es notwendig ist, die Verarbeitung von Informationen in den Räumlichkeiten einer Organisation zu halten oder die vorhandene Hard- und Softwareinfrastruktur zu nutzen. Einer der größten Nachteile privater Bereitstellungen ist die Unfähigkeit, nach Bedarf zu skalieren und Lasten effizient zu bewältigen. In diesem Fall ist es wichtig, die Funktionen von Public Clouds nach Bedarf zu nutzen. Vor allem aus diesen Gründen könnte eine Hybridlösung eine interessante Möglichkeit sein. Die Vorteile aus der Privat Cloud und Public Cloud vollkommen ausschöpfen zu wollen.

Hybrid Clouds ermöglichen es Unternehmen, vorhandene IT-Infrastrukturen zu nutzen, vertrauliche Informationen innerhalb der Räumlichkeiten zu verwalten und durch die Bereitstellung externer Ressourcen natürlich zu wachsen und zu schrumpfen. Das heißt, Ressourcen freizugeben, wenn sie nicht mehr benötigt werden um IT-Kosten zu ersparen.

In einer Hybrid Lösung beschränken sich dadurch die Sicherheits-bedenken nur auf den Public Teil der Cloud, der verwendet werden kann, um Operationen mit weniger strengen Einschränkungen durchzuführen. Ein Hybrid Modell ist daher als ein heterogenes verteiltes System, das aus einer privaten Cloud resultiert, die zusätzliche Dienste oder Ressourcen aus einer oder mehreren öffentlichen

Clouds integriert zu sehen. Mithilfe dieser Modellierung einer Cloud, werden in einer Cloud die Skalierbarkeitsprobleme behoben, indem Hybrid Clouds externe Ressourcen zur Überschreitung benutzen können. (Hurwitz et al. 2010)

Community Cloud

Auch als Semi-Public Cloud angesehen. Sie werden unter normalen Umständen von Mitgliedern einer ausgewählten Gruppe von Organisationen gemeinsam genutzt. Gemeinschaftliche Zwecke und Missionen führen sie zu einer Gruppierung in einer Community. Sie wollen als eine Organisation, keine Public Cloud nutzen, die freizugänglich ist und wollen auch nicht die einzigen Verantwortlichen für die Wartung der Cloud sein. In einer Cloud wie dieser können Verantwortungen mit anderen geteilt werden. (Hurwitz et al. 2010)

2.2.4 Cloud Service Models

Laut des nationalen Instituts für Standards und Technologie ist eine Cloud nicht nur eine Lösung für eine Umgebung, sondern dient auch einer Dienstleistungsservice. Dieser Abschnitt handelt sich, um die einzelnen Cloud Servicemodelle und ihre Funktionen. Die Clouds in Cloud Computing können alle IT-Dienste unterstützen, die als Dienstprogramm genutzt und über ein Netzwerk, höchstwahrscheinlich das Internet, bereitgestellt werden. Eine solche Charakterisierung umfasst ganz unterschiedliche Aspekte wie zum Beispiel die Infrastruktur; Plattformen; Anwendungen und Dienste. (Hurwitz et al. 2010)

Das Servicemodell ist in drei Modellen untergliedert:

Infrastructure as a Service (IaaS)

Die Infrastruktur als eine Service, im Grunde genommen sehr selbst erklärend, stellt im Klaren den Kundinnen und Kunden oder der User die grundlegende Infrastruktur als eine Dienst bereit. Dies könnten physische Maschinen, virtuelle Maschinen, Netzwerke, Speicher oder eine Kombination aus allen sein. IaaS-Implementierungen werden häufig verwendet, um die internen Rechenzentren einer Organisation zu ersetzen. Dies ermöglicht Organisationen mehr Flexibilität und dies zu geringen Kosten. (Rountree and Castrillo 2014)

Platform as a Service (PaaS)

Das PaaS Service-Modell stellt ein Betriebssystem; eine Entwicklungsplattform oder eine Datenbankplattform bereit. Die „PaaS“ ermöglichen es Unternehmen, Anwendungen zu entwickeln, ohne sich um den Aufbau der Infrastruktur kümmern zu müssen. Die zur Unterstützung der Entwicklungsumgebung

erforderlich sind. Allerdings ist hier wiederum zu bedenken, dass wenn bei einer Cloud Umgebung für eine PaaS-Implementierung entschieden wurde, ist die Cloud möglicherweise in den Tools eingeschränkt, die zum Erstellen der Anwendungen verwenden wird.

Heißt so viel wie, dass die Userinnen und User dieser PaaS-Implementierung, nur die bereitgestellten Applikationen und dessen Daten verwalten können und die restlichen Utensilien wie Laufzeit; Betriebssystem; Netzwerk; Speicher; Server; Art der Virtualisierung und die Middleware durch Cloud Provider verwaltet werden. (Rountree and Castrillo 2014)

Software as a Service (SaaS)

Software als eine Service oder SaaS stellt Anwendungs- und Datendienste bereit. Anwendungen, Daten und alle notwendigen Plattformen und Infrastrukturen werden vom Dienstleister bereitgestellt. SaaS ist das ursprüngliche Cloud-Service-Modell. Es bleibt nach wie vor das beliebteste Modell und bietet mit Abstand die meisten Anbieteroptionen. (Rountree and Castrillo 2014)

2.2.5 Allgemeine Definition von Containerisierung

Die weitverbreitetste Methodik für Virtualisierungen in der Cloud Welt, ist der Hypervisor. Sie ist die Softwareschicht, die in einem Hardwaregerät isolierte virtuelle Maschine erstellen und ausführen lässt. Doch aufgrund der Nachfrage an mehr Flexibilität, Skalierbarkeit und lukrativer Ressourcenmanagement, tendieren viele Cloud Provider zu Containerlösungen. In der IT-Welt haben sich somit die Containerisierungen rasant weiterentwickelt.

Es konnte eruiert werden, dass Virtualisierungen via Container ressourceneffizienter sind, als Virtualisierungen per virtuellen Maschinen. Im Gegensatz zu virtuellen Maschinen, welche per Gastbetriebssysteme durchgeführt werden, um die Anwendung bereitzustellen, verwendet diese Methode dasselbe Betriebssystem, in welchen der Container platziert ist.

Daher gelingt es in der Containerisierung, die Ressourcen, je nach Nutzung und Bedarf, zu verwalten. Dies lässt sich wiederum positiv anmerken, wenn für weitere Instanzen auf dem Server ausreichend Ressourcen zur Verfügung stehen, weil durch die Benutzung der Container, die Ressourcen kalibriert werden.

Die Hypervisor Technologie wurde erlischt, weil sie keinen redundanten Kern im Betriebssystem, vereinzelte Bibliotheken je Applikation und Binärdateien

aufweisen konnte, wie die Verwendung der Containerlösungen. Demgegenüber erfordert die alte Technologie, der Hypervisor, ein vollständiges Betriebssystem auf der VM, welches wiederum viele Ressourcen des eigentlichen Betriebssystems ausschöpft. (Zhang, Cheng, and Boutaba 2010)

2.2.6 Allgemeine Definition von Orchestrierung

Mit Laufe der Zeit hat sich die Methode der Containerisierung, in allen Ebenen des Anwendungslebenszyklus etabliert. Vor allem, hat sich das Etablissement sehr auf die Produktionsphase der Anwendung entwickelt. Jedoch bestehen zurzeit Anwendungen, wobei dessen Workloads in vielen Containern abgelagert werden müssen, auf unterschiedlichen Hosts. Solch komplexe Architekturmodellierungen, die verkapselte Anwendungen, in unterschiedlichen Hosts ausführen, fordern weitere Tools, wie Verwaltungstools, um den Überblick beizubehalten. Durch die Container Orchestrierung werden Orchestrierungen bereitgestellt, automatisiert, verwaltet, skaliert und die Vernetzung von Containern im gesamten Cluster gepflegt. So ein Verwaltungstool ist zum Beispiel Docker Swarm. Eine detaillierte Übersicht und weitere Beispiele von Container Orchestrierungstools, werden im Kapitel „Aktueller Stand der Technik“ genauer beschrieben. (Course Hero 2023)

2.3 Aktueller Stand der Technik

In diesem Abschnitt begebe ich mich auf den aktuellen Stand der Technik in Bezug auf die Bereitstellungsmöglichkeiten von Cloud-Computing Systemen. Zuerst wird das aktuell, allgemeingültige Architekturdesign von Rechenzentren beschrieben, anschließend wird die nächste Stufe der Cloud Computing - Ära, dass serverlose Computing, erläutert. Darüber hinaus werden die Grundfundamente der drei Hyperscaler (Azure, AWS und Google Cloud) dargestellt. Abschließend folgt ein Überblick über die Containerisierung- und Orchestrierungsmöglichkeiten in der Cloud.

2.3.1 Architekturdesign von Rechenzentren

Die Rechenzentren, welche die Rechenleistung und den Speicher einquartieren, haben eine große Bedeutung in Cloud Computing. Sie enthalten meistens Tausende von Geräten wie Server, Arbeitsspeicher, Switches und Router. Eine korrekte Planung dieser Netzwerkarchitektur ist ebenso von großer Bedeutung, da sie die Anwendungsleistung und den Durchsatz in einer solchen verteilten Computerumgebung stark beeinflussen. Nichtsdestotrotz sollten Rechenzentren

auch mit einer Skalierbarkeits- und Ausfallsicherheitsfunktion bestattet sein. Die Fundamente eines Rechenzentrums bestehen aus der Kern-, Aggregations- und Zugriffsschicht.

Wie auf der Abbildung 1 dargestellt, befindet sich die Zugriffsschicht am Ort, an dem die Rack Server mit dem Netzwerk verbunden sind. In der Regel bestehen Rechenzentren aus mehreren Rack-Servern, welche je mit 20-40 Servern bestückt sind. Die abgebildeten Rack Server dienen als Computer, die auf Langlebigkeit und auf ihre Leistungsfähigkeit optimiert sind. Ihre Funktionalität ist einfach, Zugänglichkeit auf Dienstprogramme oder Daten über ein Netzwerk. Aufgrund ihrer überdimensionalen Größe im Gegensatz zu einem herkömmlichen Computer, eignen sie sich ideal für IT-Lösungen. (Zhang, Cheng, and Boutaba 2010)

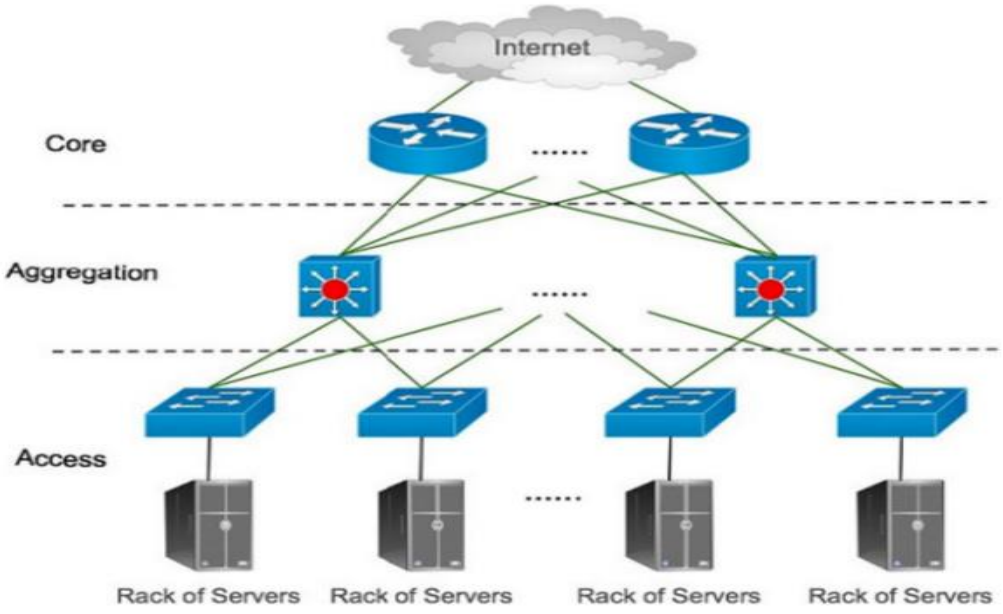


Abbildung 1 Architekturdesign von Rechenzentren

Ein solcher Rack-Server ist jeweils mit einem Access Switch mit einer 1-Gbit/s -Verbindung verbunden. Die Zugangsswitches sind üblicherweise mit zwei Aggregationsswitches für die Redundanz mit 10-Gbit/s -Verbindungen verbunden.

Die zweite Schicht, der sogenannte Aggregationsschicht, stellt in Cloud-Computing Umgebung die wichtigsten Funktionen bereit, wie zum Beispiel die

Domain-Service; Lokalisierungsservice, Server-Load-Balancing und weitere Services dergleichen.

Die Kernschicht und somit die erste Schicht der Architektur, bietet die Konnektivität zu mehreren Aggregationsswitches und ebenso eine robuste geroutete Bereitstellung ohne „Single Point of Failure“ (SPOF). In der Kernschicht verwalten die Kernrouter den Datenverkehr, in und aus den Rechenzentren. In der Praxis werden Standard-Ethernet Switches und – Router benutzt, um die Netzwerkinfrastruktur aufzubauen.(Zhang, Cheng, and Boutaba 2010)

2.3.2 Allgemeingültiges Design einer Netzwerkarchitektur für Rechenzentren

- Hohe einheitliche Kapazität:

Die maximale Rate eines Server-zu-Server-Verkehrsflusses sollte nur durch die verfügbare Kapazität auf den Netzwerkschnittstellenkarten der sendenden und empfangenden Server begrenzt werden, und das Zuweisen von Servern zu einem Dienst sollte unabhängig von der Netzwerktopologie sein . Es sollte einem beliebigen Host im Rechenzentrum möglich sein, mit jedem anderen Host im Netzwerk über die volle Bandbreite seiner lokalen Netzwerkschnittstelle zu kommunizieren. (Rountree and Castrillo 2014)

- Kostenlose Migration von virtuellen Maschinen (VM):

Durch die Virtualisierung kann der gesamte VM-Status über das Netzwerk übertragen werden, um eine VM von einer physischen Maschine auf eine andere zu migrieren. Ein Cloud-Computing-Hosting-Service kann VMs für statistisches Multiplexing oder dynamisch ändernde Kommunikationsmuster migrieren, um eine hohe Bandbreite für eng gekoppelte Hosts zu erreichen oder um eine variable Wärmeverteilung und Stromverfügbarkeit zu erreichen, des Rechenzentrums. Die Kommunikationstopologie sollte so gestaltet sein, dass sie eine schnelle Migration virtueller Maschinen unterstützt. (Rountree and Castrillo 2014)

- Elastizität:

Unter Elastizität darf die Anpassungsfähigkeit einer Cloud zu der jeweiligen Anwendungsanforderung verstanden werden. Führt diese zu einer Belastung kann es für Ausfälle führen. Ausfälle, werden wiederum in großem Umfang üblich sein. Aus diesem Grund sollte die Netzwerkinfrastruktur fehlertoleranter

gegenüber verschiedenen Arten von Serverausfällen, Verbindungsausfällen oder Rack-Server-Ausfällen sein. Um die Elastizität besser zu verinnerlichen, ist ein Rückblick an die Schichtenarchitektur (OSI-Modell) nötig.

Um die angepasste Elastizität je nach Anwendungsanforderung zu bewahren, sollte die Adressierung einer Nachricht, wie im Adressierungsschemata der Vermittlungsschicht der Schichtenarchitektur eingeordnet werden. Kommunikationsadressierungen wie Unicast- und Multicast-Kommunikationen sollten nicht in dem Umfang beeinträchtigt werden, der durch die zugrunde liegende physische Konnektivität zulässig ist. (Rountree and Castrillo 2014)

- Skalierbarkeit:

Die Netzwerkinfrastruktur muss auf eine große Anzahl von Servern skaliert werden können und eine schrittweise Erweiterung ermöglichen. Die Skalierbarkeit ist sehr eng verbunden mit der Elastizität einer Bereitstellung. Sie befähigen die Cloud- Umgebung, je nach Bedarf und Anforderung zusammenzuziehen und auszudehnen.

Angenommen, gewisse Workloads nehmen an Volumen zu, so werden die Elastizität und Skalierbarkeit aufgefordert, die Zuteilung und das Hinzufügen von weiteren Ressourcen beziehungsweise deren Rücknahme oder Umschichtung, wenn die Nachfrage sich reduziert. (Zhang, Cheng, and Boutaba 2010)

- Rückwärtskompatibilität:

Der Stand der Technik erlaubt es einer zur Verfügung gestellten Umgebung auch rückwirkend kompatibel zu sein. Die Netzwerkinfrastruktur ermögliche dies mithilfe von Switches und Router, die mit Ethernet und IP-Adressierungen arbeiten. In diesem Kontext bezieht sich die Rückwärtskompatibilität in der Umgebung auf ein physisches/virtuelles Hardware- oder Softwaresystem. Dadurch können Schnittstellen und Daten aus früheren Versionen des Systems oder mit anderen Systemen tadellos benutzt werden. (Zhang, Cheng, and Boutaba 2010)

2.3.3 Serverless Computing als Function as a Service (FaaS)

Der Begriff „FaaS“, ist zwar noch nicht weit ausgebreitet, aber dennoch stellt serverless Computing eine neue Ära in Cloud Computing dar. Gedacht ist der serverless Computing als ein Cloud-Service Modell, welches Flexibilität bei der Entwicklung, Bereitstellung und in Implementierung von Anwendungen bieten soll, ohne dass die Entwicklerinnen und Entwickler sich Gedanken über die

Bereitstellung von Servern machen muss. Es verhilft Entwicklerinnen und Entwicklern, sich auf ihr Kerngeschäft zu konzentrieren, anstatt ihre wertvolle Zeit in die Bereitstellung, Skalierung und Verwaltung von Servern zu investieren. FaaS, bietet eine vollständige Infrastruktur, die die Anwendung ausführt, indem eine Abstraktionsebene über der Cloud-Infrastruktur hinzugefügt wird.

Die angesagtesten Cloud Provider, die diese Funktion, Ihren Kundinnen und Kunden als ein serverloses Framework bereitstellen, sind unter anderem die AWS, Azure und Google Cloud Platform.

Cloud Provider, die IT-Ressourcen auf Basis des Cloud Computings anbieten, sind auch unter dem Namen „Hyperscaler“ sehr bekannt. Die Hauptaufgabe der Hyperscaler ist es, die horizontale Skalierung zu ermöglichen. Eine maximale Skalierung wie diese, ist nur möglich, wenn ein Maximum an Leistung und Durchsatz für die Benutzerinnen und Benutzer bereitgestellt wird und ausreichend Elastizität vorhanden ist. Außerdem müssen sie auch einen redundanten Aufbau besitzen, damit Fehler im System dennoch toleriert und die Cloud-Umgebung trotzdem eine hohe Verfügbarkeit leistet. (Red Hat 2020)

Im Klartext sind Hyperscaler, Systeme, die durch Cloud Computing kreiert sind. Systeme dieser Art können situationsbedingt mit Millionen von Servern, wie in einem Rechenzentrum, in einem Netzwerk verbunden und erweiterbar sein.

Es gibt vier Cloud-Anbieter, die allgemein als die bekanntesten Hyperscaler gelten, diese sind: „IBM; Amazon; Microsoft und Google“. In Public Cloud – Sektor ergeben die drei Hyperscaler (AWS; Azure und Google Cloud Platform) zusammen einen Gesamtmarktanteil von 75 Prozent. Aus diesem Grund wird im Rahmen dieser Bachelorarbeit der Fokus auf „Amazon Web Service (AWS); Microsoft Azure und Google Cloud Platform“ festgesetzt. Da alle drei Hyperscaler im Markt sehr beliebt sind, haben alle drei teils unterschiedliche vorteilhafte Funktionen, die gegebenenfalls qualitativ im Vergleich miteinander etwas differenzieren.

2.3.4 Die Grundlagen der Hyperscaler (Azure, AWS und Google Cloud)

Wie bereits im Kapitel 2.3.3 erwähnt, ist serverloses Computing ein Cloud-Computing Modell, in welchem der Code als Dienst aufrechterhalten wird. Der Vorteil hierbei ist, dass die User sich nicht um die zugrunde liegende Infrastruktur zu kümmern hat. Nichtsdestotrotz bedeutet dies nicht, dass in einer serverlosen Architektur, kein Server benötigt wird. In einer serverlosen Architektur, werden die Funktionen immer noch auf den Servern ausgeführt, doch mit dem

Unterschied, dass hierbei ein Dritter die Entwicklung verwaltet. Die Drittanbieter, die den Markt als Cloud Service Provider (CSP) dominieren sind, die drei Hyperscaler: Microsoft Azure (Azure), Amazon Web Service (AWS) und Google Cloud Platform (GCP).

Die Etablierung von AWS, Azure und Google Cloud

Azure:

Azure wurde 2010, als eine kompetente Cloud Computing Plattform für Unternehmen bereitgestellt. Seit der Gründung hat Azure große Leistung unter seinen Mitbewerbern einführen können.

AWS:

AWS ist die Tochtergesellschaft von „amazon.com“, die ebenso sowohl für Einzelpersonen, Unternehmen und für die Regierung, On-Demand und Cloud-Computing Plattform Lösungen auf kostenpflichtiger Abonnementbasis anbietet. Sie gilt als der älteste und erfahrenste Cloud Service Anbieter im Cloud-Markt.

2006 wurde AWS mit Service Featuren, wie Elastic Compute Cloud (EC2), Simple Storage Service (Amazon S3) und vielen weiteren Angeboten im Markt veröffentlicht. Die Veröffentlichung weiterer Feature folgt seit 2009. Im Jahr 2009 fand die Veröffentlichung von Elastic Block Store (EBS) statt. Weitere Features wie, Amazon CloudFront, Content Delivery Network (CDN) und mehr wurden im Nachhinein adaptiert.

Google Cloud:

Der Cloud-Computing Dienst, Google Cloud Platform (GCP) wurde im Jahr 2011 veröffentlicht. Sie wurde von Google eingeführt. Die ursprüngliche Absicht von Google ist es gewesen, Google eigene Produkte, wie Google Search Engine und weiters Produkte wie YouTube zu stärken. In nur wenigen Jahren hat es Google geschafft, ein gutes Fundament in der Cloud Branche zu positionieren. Auch GCP stellt ihre Clouddienste für Unternehmen bereit, so wie die anderen Cloud Service Provider.

(Intellipaat 2023)

Verfügbarkeitszonen der Rechenzentren von Azure, AWS und GCP

Da AWS, als der Älteste im Cloud-Markt gilt, konnte konstatiert werden, dass sie an mehreren Standorten weltweit hosten. Auch Azure und GCP hosten weltweit an mehreren Standorten, doch unterscheiden sie sich an der Anzahl der Verfügbarkeitszonen weltweit.

- AWS ist in Besitz von über 66 Verfügbarkeitszonen, 12 weitere sind auf dem Weg.
- Azure ist in 54 Regionen weltweit verfügbar und hat in 140 Ländern der ganzen Welt seine Rechenzentren aufgestellt.
- GCP ist nur auf 20 Regionen weltweit unterwegs und hat drei weitere Rechenzentren im Aufbau.

2.3.5 Container Technologien

Containerlösungen sind sehr lange schon im Markt und wurden jahrzehntelang immer wieder erneut eingeführt. Es wurden von vielen Akteuren Lösungsansätze bezüglich der Containertechnologie entwickelt.

Im Markt verbreitete Implementierungsmodelle sind:

Docker:

Der Open Source Verwaltungstool für Container, wurde zum ersten Mal im Jahr 2013 in den Markt integriert. Sie dient zur Automatisierung der Bereitstellung von Anwendungen.

Die Docker Architektur besteht aus drei Kernkomponenten, diese sind:

- Docker-Client,
- Docker-Host,
- Docker Registry.

Der Host wird in diesem Fall als der Gastgeber Computer gesehen, auf welchem der Docker Daemon (zentrales Steuerungselement der Container) und Docker Container (verkapselte Anwendung) ausgeführt wird. Der Docker-Client hingegen, ist die Benutzerschnittstelle zu Docker. (Docker 2022)

Rocket (rkt):
Im Vergleich zur Docker Containertechnologie, soll Rocket die neue, aufstrebende Containerlösung darstellen. Obwohl das CoreOS Betriebssystem sowohl die Rocket-Container als auch die des Dockers unterstützt, konkurrieren beide Technologien miteinander. Gründe für die Konkurrenz könnten es sein, weil Rocket als eine sichere, offene und konvergente Lösung entwickelt worden ist. (Rocket 2023)

sonstige Technologien:
Weitere Open Source Technologien für Virtualisierungen sind beispielsweise Podman; Buildah; BSDJail; OpenVZ; LinuX Containers (LXC), LXD und viele Weitere.

Trotz der Vielfalt an Container Lösungsansätzen sind Docker und Rocket die aktuell in häufig verwendeten Container Technologien, aufgrund ihrer vielfältigen Obliegenheit.

- Die Vor- und Nachteile von Containerisierung:

Vorteile:

- Mit Hilfe der Containerlösung können Startzeiten, Verarbeitungsaufwand und Lageraufwände im Vergleich zu herkömmlichen VMs stark reduziert werden.
- Beschriftung der Container per Namespaces bieten eine Isolierung der Container pro Prozess an. Aus diesem Grund sind Container isoliert bereitgestellt und steuern Prozesse und Ressourcen.
- Die Cgroup, eine Linux Kernel Funktion, sammelt alle Ressourcennutzungen von Prozessen und ist damit beschäftigt diese zu verwalten. Unter Ressourcennutzung von Prozessen können Speicher, CPU, Block In & Output verstanden werden.

Nachteile:

- In Containerlösungen verbirgt sich noch die kleine Sicherheitslücke, da Prozesse der Benutzerinnen und Benutzer auf dem gemeinsam genutzten Betriebssystem isoliert sind. Bis Dato ist die Methodik, aus technischer Sicht erschwert, das gleiche Maß an Isolation wie in herkömmlichen VMs bereitzustellen.
- Im Markt sich auf enthaltende Containerstandards unterstützen nur 64-Bit-Systeme.

(Guyton 2019)

2.3.6 Tools zur Container Orchestrierung:

Docker Swarm:

Für Tätigkeiten, wie das Laufen lassen von Docker und eventuell für die Auswahl des Hosts zum Durchführen von Containern benutzt Swarm, die API Schnittstelle. Das Tool besteht aus zwei Teilen. Der Swarm Agent durchläuft in jedem Host und hat die Kontrolle über den Swarm Manager. Dessen weitere Aufgabe ist es, für die lückenlose Ausführung der Tätigkeit zu sorgen. Der Swarm Manager hingegen orchestriert und plant Container auf den Hosts. Im Docker Swarm befindet sich ein Erkennungsprozess, wobei dieser Prozess beschäftigt wird, um dem Cluster Hosts hinzuzufügen. Das Container Orchestrierungstool verwendet Docker-Compose, um die horizontale Elastizität zu unterstützen und harmoniert sowohl mit Rocket als auch mit einfachen Docker Containern. (Ward 2016)

Linux CoreOS Fleet:

Ein ähnliches Tool wie Docker Swarm, wäre CoreOS Fleet. Sie ist ein Low-Level-Cluster-Management-Tool und stellt den gesamten Cluster, als ein einziges Init-System dar. Dies führt dazu, dass das System den ersten Prozess in sich selbst startet und aus einem Datei, die weiteren Schritte und Prozesse, die sie befolgen muss herausliest und diese dann ausarbeitet. Fleet besitzt ein sehr gesondertes Management für die Container und kann diese starten, stoppen Informationen über die aktiven Dienste oder Container in verschiedenen Cluster erhalten. Außerdem ist zu erwähnen, dass sie wie Docker Swarm beide Container Modelle, Rocket und Docker, unterstützt. Weiters ist sie sehr fehlertolerant und kann Container von einem Host in ein anderes ausnahmslos migrieren. (Ward 2016)

Apache Mesos:

Apache Mesos ist zwar ebenso ein Open-Source-Cluster-Manager Tool, welches zur Verwaltung und Bereitstellung von Container Anwendungen bedient wird. Wie auch CoreOS Feet, unterstützt auch Apache Mesos die horizontale Elastizität. Im Gegensatz zum Bisherigen ist hier der Unterschied, dass Apache für Riesen Cluster Landschaften entwickelt wurde. Marathon ist ein Jobsystem, die mit Kombination von Mesos, sich um die Durchführung von Jobs und Aufgaben in Clusterlandschaften beschäftigt. (Chandrakant 2019)

Kubernetes (K8s):

Ein weiterer leistungsstarker Container Orchestrierungstool ist die Kubernetes. In Rahmen dieser Bachelorarbeit werde ich mich sehr stark auf die Einzelheiten dieser Technologie fokussieren. In weiterer Folge dieser Arbeit, dient sie für die Nachvollziehbarkeit der verwalteten Kubernetes Services von Hyperscalern.

Kubernetes ist von Google gegründet und sehr bekannt für ihr neues Konzept, Container zu vernetzen und zu organisieren. In Kubernetes wird eine Cluster Technologie verwendet, welche containerbasierte Systeme über auf eine API aufweist und verwaltet. Das Open- Source Orchestrierungstool wird von der Cloud Native Computing Foundation (CNCF) verwaltet und wirkt bei allen großen Cloud Providern und auch bei Softwareanbieter mit. Die Basis Architektur von Kubernetes erfordert einige Komponente, um die Container orchestrieren zu können.

- **Basis Kubernetes Architektur:**
  In Abbildung 2 wird dargestellt, wie die Basis Architektur von Kubernetes bereitgestellt wird. Um die Architektur nachvollziehen zu können, müssen vorerst gewisse allgemeingültige Fachbegriffe wie Node; Cluster und Pod erklärt werden.

  In K8s wird die Rechenhardware, somit entweder die physische Maschine in einem Rechenzentrum oder eine VM, als ein Node bezeichnet. Dessen Kapazität sehr flexibel vordefiniert werden kann in der Konfigurationsphase einer Kubernetes-Architektur.

Mit der Vorstellung, dass Maschinen in Kubernetes als Node bezeichnet werden, ist es umso leichter zu versinnbildlichen, dass Cluster in diesem Sinne, als ein Ganzes zu betrachten sind. Das heißt, unter einem Cluster ist die Kombination von einer Anwendung und eine physische oder virtuelle Maschine gemeint.

Ein weiterer Begriff, der auch in vorherigen Kapiteln angedeutet worden ist, aber noch nicht endgültig klargestellt gewesen ist, ist ein Container. Wie im Kapitel „Docker“ kurz erläutert, ist unter einem Linux Container oder regulär nur Container, die Bereitstellung eines Programms mit all ihren Abhängigkeiten zu verstehen. Unter Abhängigkeiten sind hier die Bibliotheken von Anwendungen gemeint. So wird in einem Container eine Linux Ausführungsumgebung parat gehalten und kann weiterverarbeitet werden. Natürlich besteht die Möglichkeit, mehrere Programme gleichzeitig in einem einzigen Container zu verkapseln, doch aus Gründen wie, Programm Sourcecodes diagnostizieren und Durchführung der Updates von Programmen, ist es sehr empfehlenswert viele, aber kleine Container für Anwendungen bereitzustellen, als einen großen Container, wo alles verkapselt wird.

Weiters verhilft die Definition von Pods ebenso, die Basis-Architektur von Kubernetes leichter erschließen zu lassen. In K8s werden ein oder mehrere Container in eine übergeordnete Struktur eingewickelt, die als Pod bezeichnet werden. Alle Container die sich im selben Pod befinden, teilen sich dieselben Ressourcen und den lokalen Netzwerk.
In einem Pod, können Container problemlos mit anderen Containern kommunizieren, wie als ob sie sich auf derselben Maschine befinden würden, obwohl eine gewisse Isolierung im Pod vorhanden ist.

Pods werden als Replikationseinheit in Kubernetes gesehen. Das heißt nichts anderes wie, wenn eine Anwendung zu beliebt wird und eine einzelne Pod-Instanz, die Last nicht tragen kann, kann dies in Kubernetes so konfiguriert werden, dass bei Bedarf neue Replikate des Pods im Cluster bereitgestellt werden. Dies kann verwendet werden, auch, wenn ein Pod nicht stark ausgelastet ist. Die Verwendung von Pods als Replikationseinheit hat sich so dermaßen etabliert, dass mehrere Kopien eines Pods jederzeit in einem Produktionssystem einer Umgebung ausgeführt werden, um Lastausgleich und Ausfallsicherheit zu gewähren.

Wie in einer Container Bereitstellung, sollte auch in der Pod Bereitstellung darauf geachtet werden, dass ein Pod bei Möglichkeit sehr eingeschränkt und klein gehalten wird. Da Skalierungen innerhalb eines Pods, nach oben und unten stattfinden, werden alle Container in einem Pod zusammen skaliert. In Worstcase endet die Szenario mit Ressourcenverschwendung. (Goel 2023)

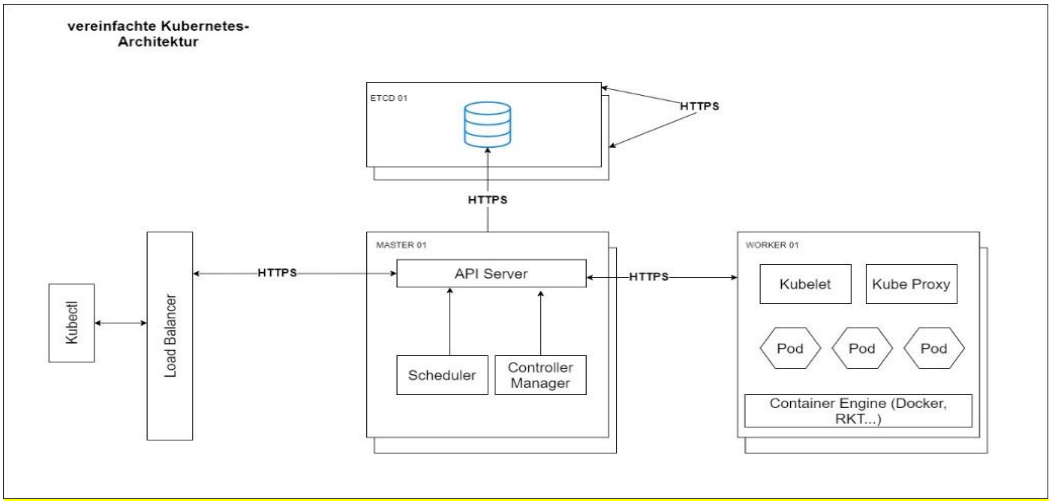


Abbildung 2 Basisarchitektur von Kubernetes

- API-Server:

API-Server verarbeitet alle Anforderungen, die in einem Cluster erscheinen. Über den HTTPS – Port (443) senden sowohl Benutzerinnen und Benutzer als auch die weiteren mitintegrierten Kubernetes-Komponente, Ereignisse an den Server. Der API-Server ist dann für die Verarbeitung und Aktualisierung dieser Ereignisse zuständig, welche im fortlaufendem Datenspeicher (etcd.) fungieren. Außerdem führt der Server auch die Authentifizierungs- und Autorisierungsdienste im Datenspeicher des Kubernetes. Wobei die Dienste sich nach den Grundprinzipien der Clusterkonfigurationen orientieren.

- Datenspeicher auf Steuerungsebene:

Wie schon im Punkt „API-Server“ bekannt gegeben, werden alle Kubernetes relevante Informationen im persistenten Datenspeicher (etcd) gespeichert. Sie gilt als der Standartdatenspeicher für alle relevanten

Clusterdaten. Das heißt, in diesem Speicher können mehrere etcd-Instanzen vorhanden sein, die die hohe Verfügbarkeit der Cluster bereitstellen. Weil etcd die einzige Ressource ist, über die Clusterbildung in Kubernetes, ist es unumgänglich eine Notfallwiederherstellungsfile des fortlaufendenden Standarddatenspeichers sicherzustellen.

- Controller-Manager

Der Controller-Manager ist die Steuerungsebene in der Basis-Architektur. Sie kann als ein nicht abgeschlossenes Regelkreis gesehen werden, welches mehrere Variationen von Controllern ständig im Hintergrund durchführt. Hier werden im Grunde genommen die einzelnen Kubernetes-Objekte beobachten und mit dem vordefinierten Soll-Werten verglichen. Ständig im Hintergrund durchlaufende Funktionen können sein, wie dass Aufrufen von Pod-Zugangskontrollen, die Überprüfung der Pod-Standardwerte eventuell auch das Hinzufügen von weiteren Container in Pods, über die Multicontainer-Sidecar Facette.

- Kubelet (Agent)

Jede Node in einem Cluster von Kubernetes, verfügt wie der Name schon verrät einen Agenten, namens Kubelet. Dieser ist wie ein Wächter zu sehen, welche die Definitionen eines Workloads über den API-Server abliest und sicherstellt, dass die Anforderungen je nach Definition fehlerfrei ausgeführt werden und meldet die positive Ausführung des Workloads und des Nodes, als Status beim API-Server zurück.

- Scheduler

Die Planung der Objekte in Kubernetes basiert auf ein raffiniertes Algorithmus. Der Scheduler berücksichtigt hierbei die Attribute der Ressourcenanforderung und verwendet eine individuelle Priorisierung an, um die Pods, je nach Kubernetes Worknodes bereitzustellen. Das Algorithmus im Scheduler, entscheidet, welcher Pod zu welchem Worknode vorbereitet und bereitgestellt werden muss, damit eine gewisse Anwendung zu bestimmter Zeit durchgeführt werden kann.

- Kube-Proxy

  Mithilfe eines Netzwerkproxys gelingt es jedem Cluster Worknode, eine Verbindung zu den Anwendung im Cluster zu stellen. Der Kube-Proxy wiederspiegelt die im Cluster vordefinierten Dienste. Anhand der Wiederspiegelung im Proxy, gelingt die Verteilung und Weiterleitung der Clientanforderungen, welche zu einzelnen Diensten vordefiniert wurde. (Goel 2023)

Google Kubernetes Engine (GKE):

Wie der Name schon sagt, ist GKE eine verwaltete Containeranwendungsumgebung, welches für die Bereitstellung, Verwaltung und die Skalierung zuständig ist. Die GKE Umgebung besteht aus mehreren Maschinen wie Compute Instanzen, welche zusammen einen GKE-Cluster bilden auf Basis der Google-Infrastruktur. Außerdem befinden sich im GKE, vorgefertigte Bereitstellungstemplates, die für spezifische Unternehmen zur Verfügung gestellt sind. Nicht zu vergessen ist, dass die GKE-Clusterbildung, auf die Basisarchitektur von Kubernetes beruhen. Auf der Abbildung 3 kann die Ähnlichkeit zu der Basis Architektur am blauen Clusterbildung erkannt werden. (Google Cloud 2023)

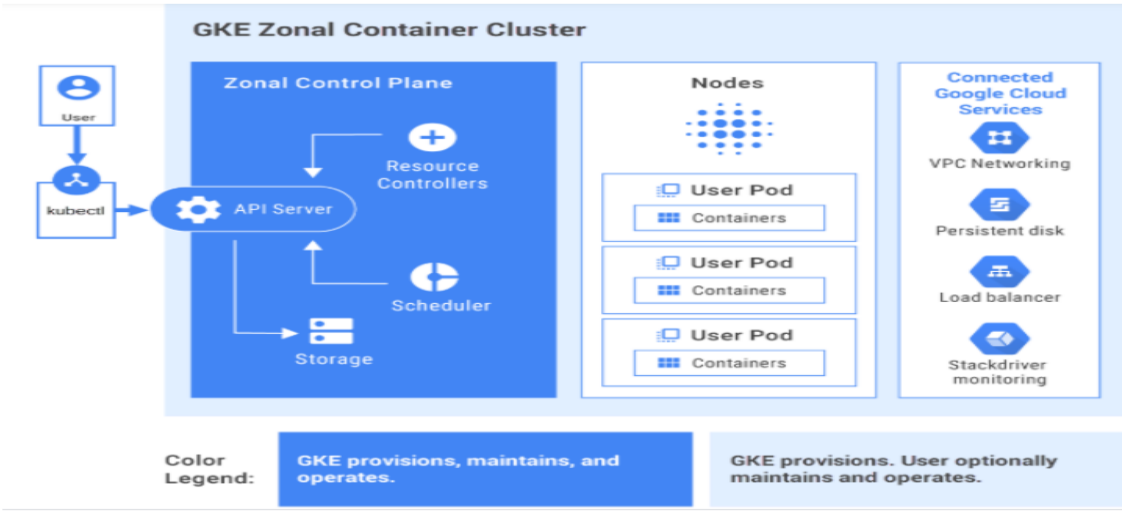


Abbildung 3 GKE-Clusterbildung (Google Cloud 2023)

Azure Kubernetes Service (AKS):

AKS ist ein vollständig verwalteter Kubernetes-Dienst, der von Microsoft Azure zurzeit im Markt angeboten wird. Weiters bietet sie serverlose Kubernetes, Sicherheit und Governance Dienste. Die einfache Bereitstellung von Containeranwendungen wird nach der Kubernetes Cluster Verwaltung von AKS, ermöglicht. Azure Kubernetes Service bereitet automatisch alle Kubernetes Masternodes vor. Im Grunde genommen müssen schließlich nur die Workernodes verwaltet und gepflegt werden.

In Abbildung 4 kann auch gesehen werden, dass Azure zusätzliche Funktionen wie Netzwerke, Azure Active Directory-Integrierung und Überwachung über Azure Monitoren unterstützt. (Microsoft Learn 2023)

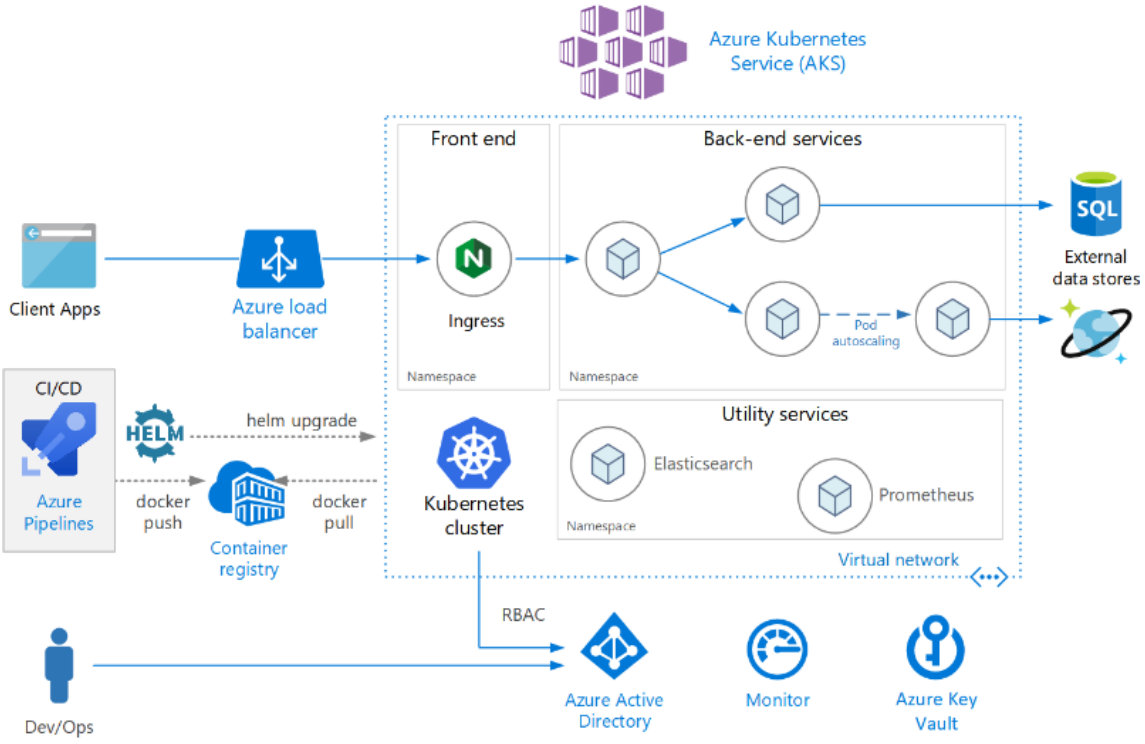


Abbildung 4 AKS-Clusterbildung (Microsoft Learn 2023)

Amazon Elastic Kubernetes Service (EKS):

So wie AKS ist auch EKS, ein komplett verwalteter Dienst von Kubernetes. Wobei in EKS der Aufbau eines Cluster mithilfe des AWS Fargate ausgeführt werden kann. AWS Fargate ist eine Maschine für Datenverarbeitung. Wie in vorherigen Kapiteln bekannt gegeben, basiert die serverlose Funktion nachdem „Pay-as-you-go“ Prinzip. Sie ist sowohl mit Amazon Elastic Container Service (ECS) als auch mit EKS verwendbar. In Abbildung 5 sind die Schrittweise Implementierung bei der Modelle dargestellt, welche in EKS ermöglicht sind. (Amazon.com AWS 2023)

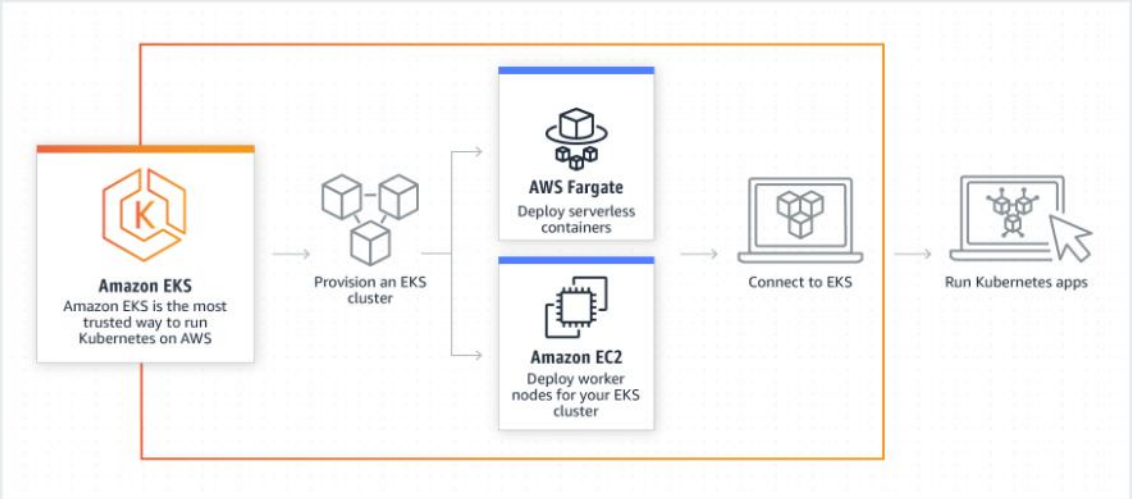


Abbildung 5 EKS-Clusterbildung mit AWS-Fargate (Amazon.com AWS 2023)

2.4 Aktueller Stand der Wissenschaft

In diesem Kapitel wird versucht mithilfe bereits vorhandenen Literaturen, die Forschungsfrage zu beantworten. Zu Erst wird auf die aktuellen Cloud-Computing Trends, sowie auch auf ihre aktuellen Herausforderungen eingegangen. Anschließend folgt der aktuellen Stand der Wissenschaft, der verwalteten Kubernetes Service-Diensten (AKS, EKS & GKE), welche aus Literaturen erforscht wird.

2.4.1 Die Trends im Cloud Computing:

Computing über die Cloud spielt für viele Unternehmen eine wichtige Rolle. Unternehmen versuchen dadurch, die Bedürfnisse ihrer Kundinnen und Kunden in Erfüllung zu bringen und ihren Wettbewerbsstatus zu optimieren. Die effiziente

und effektive Datenspeicherung die dadurch verursacht worden ist, hat den Bedarf an mehr Speicherkapazität befürwortet.

Ein weiterer Trend, welcher sich durch die hohe Nachfrage eruieren ließ ist, dass Dienstanbieter wie Cloud Service Provider, daran arbeiten müssen, die Kapazität von ihren Rechenzentren zu erhöhen. Erfahrungen bewiesen, dass Cloud ein wichtiger Bestandteil für die Aufrechterhaltung solch überlegenen Leistungen sind, um im Cloud-Markt weiters standzuhalten.

Cisco habe ästimiert, dass die Cloud im Jahr 2018 zirka um die 547 Exabytes enthielt. Erhöhung der Speicherkapazität in der Cloud, beeinflussen die Unternehmen sehr positiv. Dadurch gelingt es ihnen größere Mengen an Daten zu speichern. Die großen Datenmengen formieren sich lediglich wiederum in hilfreiche Daten über Kundeninformationen und Kundenwünsche. Weiters können durch diese Daten, dass Verhalten von Kundinnen und Kunden gespeichert und analysiert werden, um weitere zu gewinnen.

Nicht zu vergessen ist auch, dass mit dieser wachsenden Innovation in Cloud Computing, auch kleine Unternehmen profitieren. Aufgrund der Gebührensenkung in Cloud Lösungen, wagen sich auch kleinere Unternehmen diese Daten zu speichern.

Die Ansammlung von riesigen Datenmengen sind in den letzten Jahren, sind auch sehr attraktiv für Hackerinnen und Hacker geworden. Diese fanden Wege, die Sicherheit von Cloudlösungen in Gefahr zu setzen, indem sie über Wanncary und Ransomware (schädliche Software) Angriffe durchführten. Hackerangriffe dieser Art, führten dazu, dass Expertinnen und Experten die Sicherheit und Reaktionszeiten von Cloud Lösungen erhöhten.

Des Weiteren führten Bemühungen von Hackerinnen und Hacker, Datenansammlungen anzugreifen dafür, dass viele Unternehmen viel Zeit in die Erkennung von Malware und jeglichen weiteren Hackerangriffen zu investieren. Viele CSPs, die diese heikle Angelegenheit ebenso wahrnehmen unterstützen die Unternehmen, die dagegen ankämpfen. Die Situationen in der Cloud-Welt sind sehr entgegengestellt. Kundeninformationen zu sichern und aufrechtzuerhalten schöpfen immense Ressourcen aus. Um die Cybersicherheit zu gewähren, müssen Unternehmen Expertinnen und Experten einstellen, die eine Lösung auf allmähliche Hackerangriffe bereitstellen. (Nasser and Elfadil 2020)

2.4.2 Cloud Computing Trends der Zukunft

Sowohl die in der Bachelorarbeit angegebenen Hyperscaler, als auch weitere Cloud Service Provider, die Clouddienste anbieten, werden weiters von vielen Organisationen vergütet. Weltweite Großkonzerne haben begonnen herstellerspezifische Cloud-Netzwerke zu erstellen, die ihren spezifischen Anforderungen entsprechen. Die Kerngedanke hierbei ist, dass Unternehmen es lukrativ finden, eigene private Cloud-Netzwerke bereitzustellen, anstatt die Dienstleistungen von Drittanbieter weiters zu benutzen.

Coca-Cola, beispielsweise ist im Besitz, einer riesigen Datenmengen und kann aus diesem Grund, ein privates Netzwerk mit hohen Sicherheitsmaßnahmen aufbauen, welches auf ihre Bedürfnisse vordefiniert sind. Ein weiteres Beispiel, die hier einfällt ist IBM. Sie sind ein weltweitbekanntes Computerunternehmen und stellen privaten Cloud-Speicherplatz bereit. Es lässt sich sehr leicht vorhersehen, dass verschiedene multinationale Organisationen, sehr wahrscheinlich auch ihre eigene Cloud-Lösungen entwickeln werden. Massenweise Unternehmen die nicht aus diesem Sektor sind, verfügen über eine IT-Abteilung, welche sich um dieselbe Strategie fokussiert haben.

Da aber auch CSPs ihren Fundament darauf aufbauen, entwickeln sie immer wieder komplexere Angebote und bieten immer Cloud-Lösungen, in welchen die aktuellen Bedürfnisse je Unternehmen abgedeckt werden. Die Lösungen von CSPs sorgen dafür, dass weltweit Unternehmen ihre IT-Abteilungen in diese Lösungen auslagern. Diese Auslagerung führt weiters dazu, dass viele Organisation, welche diese Lösung befolgen, kein Geld mehr in die IT-Landschaft und Infrastruktur innerhalb ihrer Organisation mehr investieren müssen.

Aufgrund des rasanten Wachstums, dieser Innovation, sehen sich auch kleinere Betriebe und sogar auch Privatpersonen verpflichtet, dieser Logik von Großkonzernen anzuschließen und entscheiden sich für die Benutzung einer Cloud.

Ein weiteres Merkmal, welches zum Trend geführt hat ist, dass Unternehmen innerhalb eines Jahres, mehrmals Analysen durchführen, um Gewinne exakt ausschöpfen zu können. Um dies durchzuführen widmen sich Unternehmen ebenso an Cloud Dienstleister, da sie für die Durchführung der Analysen leistungsstarke Computer benötigen. Mit Laufe der Zeit wurde bewiesen, dass CSP Dienstleistungen mit dem „Pay-as-you-go“ Prinzip sich für Organisationen sehr gut eignen, weil sie dadurch die Dienstleistungen nur in Anspruch nehmen, wenn sie es für nötig halten.

So konnte schlussfolgert werden, dass die Durchführung von Analysen innerhalb der Cloud, Kosten und Risiken reduzieren. Daher habe sich manifestiert, dass Organisationen durch die Verwendung dieser Dienstleistung ihre Gewinne steigern und Kosten und Risiken minimieren können. (Nasser and Elfadil 2020)

- Weitere Trends im Cloud Computing sind:

    - Mobiles Cloud Computing
      Aufgrund der aktuellen Fortschritte in der Smartphone-Welt, wurde definiert, dass Mobiles Computing die Integration von Cloud Computing mit mobilen Geräten seien könnte. Die Entwicklung soll dazu beitragen, die Rechenleistung und die Speicherkapazität in mobilen Geräten zu erweitern. Weiter wichtige Aspekte, die festgestellt worden sind, Themen wie Anwendungen, Sicherheit und einheitliche Standards im Mobile Cloud Computing.

      Anhand dieser Innovation, soll es mobilen Geräten ermöglicht werden, die Hardware- und Akkulaufzeit zu verlängern. Nichtsdestotrotz sind auch einige Herausforderungen klargestellt. Wie zum Beipsiel in Leistung, Ressourcen und Technik Themen.

      Mittlerweile hat sich mobiles Cloud Computing sehr gut in sozialen Online-Netzwerkdiensten wie Games, Bildverarbeitungen und Videoverarbeitungen und allmählich in E-Business Segmenten etabliert. Dennoch durfte manifestiert werden, viele Merkmalsausprägungen der mobilen Cloud-Anwendungen noch diskutiert werden. (Nasser and Elfadil 2020)

    - Quantum Computing
      Quantum ist und bleibt das heißeste Thema im Cloud-Sektor. Sie stellt immer wieder den aktuellen Stand des Cloud Computing in Frage und dessen Durchdringung in den Sektor würde mit einer hohen Wahrscheinlichkeit zu einer vollständigen Veränderung einführen. Dienstanbieter von Quantum Computing versuchen, den Wettbewerb zurzeit zu unterbinden. Es ist aber durchaus vorhersehbar, dass diese Technik, in naher Zukunft, dass Cloud Computing komplett übernehmen wird. (Nasser and Elfadil 2020)

- hybride Cloud-Lösungen
  Ein weiteres Phänomen, dass ihre Aktualität in den letzten Jahren kaum verloren hat, ist die hybride Lösung von Cloudumgebungen. Sie sind sehr wohl bekannt, weil sie dynamisch und kostengünstig im Markt auftreten. Darüber hinaus, sind sie sehr wohl geeignet dafür, sich an die dynamischen Markanforderungen vertraut machen. (Nasser and Elfadil 2020)

- Automationen über die Cloud
  Das Ausführen von Automationen hat den Vorteil, dass sich immer wieder wiederholende Befehle, beziehungsweise Jobs, in der Cloud reduzieren. Da die Codierung zur Automatisierung fehlerfrei funktionieren muss, reduzieren sich auch die Fehler in der Cloud. Weiters führt diese Funktion zur Produktivitätssteigerung. Für Unternehmen die sich mit Clouds beschäftigen sind Automatisierungen sehr Trend geworden. Sie tragen dazu bei, die Arbeit von Cloud Admins zu erleichtern und viel mehr tragen sie dazu bei, Kosten und Zeit zu ersparen. (Nasser and Elfadil 2020)

2.4.3 Herausforderungen in Cloud Computing

Obwohl die Entwicklung sehr rasant wächst und der Cloudmarkt exponentiell sich ausweitet befinden sich dennoch gewisse Herausforderungen, die das Cloud Computing stagnieren. Diese Herausforderungen sind:

<u>Cloud-Computing Governance</u>

Aufgrund der Technologie, welche die Unternehmensleistung von Tag zu Tag verbessert, gewinnt ihre Einführung, immer mehr Wert, als Entscheidungsträger. Ein weiteres Aspekt der IT-Governance, welches den Entscheidungsträger weiters befestigt, ist die Weiterentwicklung der IT-Ziele, welche die Unternehmensstrategien beeinflussen.

In einem kürzlich neu vorgestelltem Cloud Computing Governance Framework, welches als Ziel definiert hat, Bewertungs- und Integrationsansätze mit IT-Modellen zu entwickeln, hat sich festgestellt, dass dieses Framework, die Einhaltung allgemeiner Vorschriften und Gesetze entgegensprechen könnte.

Dabei sollen auch Gesetze für Kundinnen und Kunden und auch die von Cloud-Anbietern gefährdet sein.

Ein weitere Herausforderung, welcher im Zusammenhang mit der Cloud-Computing Governance sich herausstellt ist, dass Mangel an Fachwissen in der Umgang mit Cloud Computing und dessen IT-Steuerung herrscht. Um die Komplexität der Hardwaresysteme in der Cloud zu entkommen, stellen Expertinnen und Experten eine Vision vor, welche die Komplexität abweichen können soll.

Um die Hardwareentwicklung und die Verifizierung auszuweichen, möchten sie von einem Blackbox gesteuerten Modell zu Whitebox-Modell übergehen. Dieser soll Vertrauen, Zuverlässigkeit und Effizienz auf dynamischer-, kollaborativer- und verantwortlicherweise die Entwicklung verbessern. (Nasser and Elfadil 2020)

Sicherheit in Cloud Computing

Die größte Hürde in Cloud Computing seit Tag eins, ist der Sicherheit der Daten in der Cloudwelt gewesen und anscheinend wird sie dies auch für eine weitere Weile bleiben müssen. Die Daten vor Manipulation und für widerrechtliches Verwenden zu schützen zu können ist das Kerngeschäft von Cloud Providern. Insbesondere vor allem, wenn Hackerinnen und Hacker mit Cloudanbieter zusammenarbeiten. (Nasser and Elfadil 2020)

Um diese gemeine Ausbeutung der Daten durch Hackerinnen und Hacker zu vermeiden, resultierten sich in der Vergangenheit viele Verfahren und Ansätze, wie zum Beispiel das kryptografische Verfahren, ein ellipsenkurviges Kryptographie Verfahren(ECC) und ein Ciphertext Algorithmus basiertes Schlüsselwortsuche Verfahren (LCKS) und viele mehr. Nichtsdestotrotz müssen Userinnen und User von Cloud Anwendungen dennoch mit der Gefahr leben, dass dem Cloud Computing immer eine derart Schwachstelle, als Lücke offen bleibt und solche Angriffe auftreten können. (Zhao et al. 2019)

2.4.4 Vergleich der Hyperscaler (Azure, AWS und Google Cloud)

Cloud Computing ist eine rasante IT-Technologie, mit einem sehr schnellen Bereitstellungsprinzip. Sie ermöglicht Fähigkeiten nach Bedarf und dem Unternehmensprofilen und Anforderungen angepasst einzusetzen. Im Grunde genommen bieten alle Hyperscaler dieselben Fähigkeiten, dennoch kann festgestellt werden, dass jeder einzelner Cloud Service Provider ihren Fokus auf diverse Stärken ausprägt.

Service Vergleich der Cloud Service Provider

Bei einem Vergleich von Service Dienstleistungen von CSPs, durfte eruiert werden, dass Amazon, als ein bedeutender Anbieter und im Cloud-Markt als der Pionier gilt. Sie dominieren in Bereichen des Designs und Monitorings. Sie wird hauptsächlich aufgrund ihres facettenreichen Communitys, risikoloser Verwaltung und anpassungsfähigen Funktionen.

In Anbetracht auf Azure, ist zu ebenfalls zu erwähnen, dass auch die Community von Azure sehr stark verbreitet ist, angesichts in Verwaltungen in Bezug auf Grundlagenarbeit, Prozesskapazität, Speicherung, Organisation und viele Themen mehr. Dennoch lässt sich herauskristallisieren, dass die Qualität von Azure in ihrer Registrierungsleistung liegt. Sie ermöglicht das Erstellen und Verwalten von VMs in einer anderen Dimensionsgrößenordnung, welche innerhalb einiger Minuten erreicht werden kann.

Des Weiteren bietet Microsoft Azure ein weiteres Feature, welches ermöglicht eine Integration des Clouds, mit Microsoft-Instrumenten zu verknüpfen. Das sogenannte „Open-sourcebacking“ und auch viele weitere Funktionen des Hybrid Cloud Modells.

Nichtsdestotrotz sind auch die Fähigkeiten von Google von großer Bedeutung. Vor allem in Bezug auf Automationen und Künstliche Intelligenz. Außerdem ist Google Cloud auch für ihre Unterstützung von Open Source , für Mobilität und ihren anpassbaren Cloud Verwaltungen und Verträgen sehr bekannt. Weiters konnte konstatiert werden, dass die Technologie von Google eher für cloudbasierte Organisationen gedacht ist und DevOps Talente beziehungsweise Gruppen sehr unterstützt. (Kamal et al. 2020)

Preisvergleich der Cloud Service Provider

Aus den Vergleichen durfte herausgestellt werden, dass Amazon sehr offen für Neukunden und Neulingen ist. Die Preise von AWS haben hauptsächlich den Fokus auf die Kosten der anderen Cloud Anbieter. Sie bieten ihren Userinnen und Usern sogar eine Bezahlungsoption auf Stundenbasis, für High-level Benutzungen, welche zum Beispiel beim Kennenlernen der Plattform auftauchen können. Außerdem bieten sie auch kostenlose Vereinbarungen, mit sehr vielen Restriktionen auf Kapazität und Registrierungsmöglichkeiten. Weiters ist noch zu erwähnen, dass AWS auch eine schätzweise Kostenübersicht generieren kann in ihrem Plattform, welche die aktuellen Kosten auf ungefähr dem Endbenutzer

bekannt geben, damit der Kunde einen Überblick über seine Kosten haben kann. Im Gegensatz zu AWS hängt die Kostenabdeckung bei Microsoft Azure, sehr davon ab, welche Elemente benutzt werden, während der Bereitstellung. In Vergleich mit den anderen CSPs bietet Azure eine Vorauszahlung der Elemente an, welches auch in Ratenzahlung umgewandelt werden kann bei Bedarf. Während sich Microsoft wieder Mal durch ihre Vielseitigkeit an Optionen wieder sehr bekannt macht, ermöglicht Google Cloud wieder eine sehr einfache Lösung an. Welches unter den Hyperscalern die beste Bewertung in Bereich der Bezahlung erhalten hat. GCP beruht auf ihre Einfachheit und bietet die Kosten der Bereitstellung mit einem Sekunden- oder eventuell mit einem Minutentakt an. (Kamal et al. 2020)

2.4.5 Wissenschaftlicher Stand von verwalteten Kubernetes Servicediensten

Viele Organisationen können einsehen, welche lukrativen Vorteile, sich hinter dieser Technologie verbergen, aus diesem Grund sind Container Orchestrierungstools im Markt mehr nachgefragt denn je. In einer Umfrage von CNCF aus 2020, wurde festgestellt, dass 23 Prozent der Umfrageteilnehmer mehr als 5000 Container in ihrem Unternehmen betreiben. Verglichen mit den Resultaten aus 2016, weist dieser Wert einen Aufstieg von 106 Prozent. Zwar steigt immer mehr die Nachfrage nach einer Containerorchestrierungslösung im Markt, aber dennoch bleibt die Verwaltung von Kubernetes eine große Hürde. Die Technologie versucht immer aktuell zu bleiben und bringt ständig neue Release raus. Die Veröffentlichung des neue Release fand am 09.12.2022 statt. Die neue Version von Kubernetes 1.26.0 hat eine Lebensdauer bis Q2 2024. (Cloud Native Computing Foundation 2023)

<u>Aktueller Forschungsstand der Wissenschaft in AKS</u>

Azure Kubernetes Service versucht immer den aktuellen Stand des Orchestrierungstools beizubehalten und unterstützt prompt immer die neueren Versionen, inklusive die Container Laufzeiten, welche direkt von Kubernetes eingeführt werden. Dadurch drängt es ihre Endbenutzer diese auch zu aktualisieren.

Weiters ist noch zu erwähnen, dass Azure die Entwicklungsebene verbessert hat. Es wurden automatische Upgrades für Node und geplante Wartungsfenster in AKS involviert. Des Weiteren wurde festgestellt, dass bis dato Azure in AKS, der einzige CSP ist, welcher einen kostenlosen verwalteten Kontrollplan Service

anbietet, während die anderen verwalteten Kubernetes Dienstleister wie Amazon Web Services und Google Cloud Plattform, eine ähnliche Dienstleistung um 0,10 USD die Stunde anbieten. Ein fehlendes Feature bei AKS ist die Containeroptimierung. AKS Userinnen und User sind dagegen auf Ubuntu und Windows Servern verwiesen, um die Containeroptimierung aus zu manövrieren.

Gemäß den Entwicklungserfahrungen der Expertinnen und Experten ist hier kein Package Manager vorhanden. Userinnen und User müssen wiederum auf Alternativen, wie Docker, Ansible, Helm und viele mehr zugreifen. Weiters hat Microsoft im VS-Code, dem Microsoft entwickelten Code-Editor, die Erweiterung „Bridge to Kubernetes“ zur Verfügung gestellt. In welcher die Überbrückung der Anwendungen bereitgestellt werden können. Zwar funktioniert „Bridge to Kubernetes“ mit etlichen Kubernetes Releases, dennoch benötigt es auch hier die Unterstützung von Azure CLI und eventuell einem beliebigen Package Manager für Kubernetes, wie Helm. (Hwang 2021)

Aktueller Forschungsstand der Wissenschaft in EKS

Da Amazon sowieso unter den Hyperscalern als der Pionier am Cloudmarkt gilt, ist laut dem Cloud Bericht 2021 von Flexera auch hier Amazons Orchestrierungstool, AWS ECS/ EKS, wieder der Marktführer unter den Hyperscalern, welcher laut dem Bericht aus 2021 als das Orchestrierungstool gilt, welches die meisten Container bereitstellte.

Ein Umfrageergebnis von Datadog 2021 stellt fest, dass AWS unter den Hyperscalern wie Azure und GCP, am wenigstens mit EKS die Anwendung bereitstellt. Der Grund ist eigentlich sehr offensichtlich. AWS setzt seinen Fokus eher auf seine eigene Techniken, wie zum Beispiel ECS, ECR und Fargate im Bereich des Containerangebots und tendiert zu den Funktionen des Kubernetes in EKS, bei automatische Verwaltungsfeatures, wie automatische Node Reparatur oder automatische Upgrades und für Monitoring. Des Weiteren noch bei vertikaler Pod-Autoskalierungen, holt sich AWS die EKS Feature. Der praktischer Ansatz von EKS, den Userinnen und Usern mehr Flexibilität und Verantwortung zu gewähren, kann in gewissen Ansätzen sehr komplex zum Bereitstellen sein. Beispielsweise müssen bei der Clusterbereitstellung in EKS zusätzliche Workloads bereitgestellt werden, welche wiederum in anderen CSPs, wie Azure und GCP, standardgemäß automatisch eingeführt sind. Eines dieser Workloads ist zum Beispiel das Anwendungslastenausgleich, der Application Loadbalancer von AWS oder der Node Autoscaler. Im Vergleich mit GKE oder

AKS, sind diese zwei Komponente stets als verwaltetes Kubernetes Dienstservice eingebettet, während der EKS Userinnen und User, diese Feature bei Bedarf zusätzlich zu modifizieren hat. Grundsätzlich beschweren sich auch sehr viele Expertinnen und Experten bei der Aktualisierung von Nodes. Allgemein kann zu Nodes gesagt werden, dass anstatt ein Upgrade der Node-Gruppen durchzuführen, dass Erstellen und Entleeren des Alten einfacher funktioniert, als eine direkte Aktualisierung zu implementieren.

(Hwang 2021)

Aktueller Forschungsstand der Wissenschaft in GKE

Google Kubernetes Engine hat durch die Einführung des GKE Autopilot, den Bezug zu verwalteten Kubernetes Services vervielfacht. Dadurch sind die sehr nachgefragten Funktionen wie vertikale Pod-Autoskalierung, Verwaltung der Nodes und DNS-Cache und vieles mehr, vereinfacht.

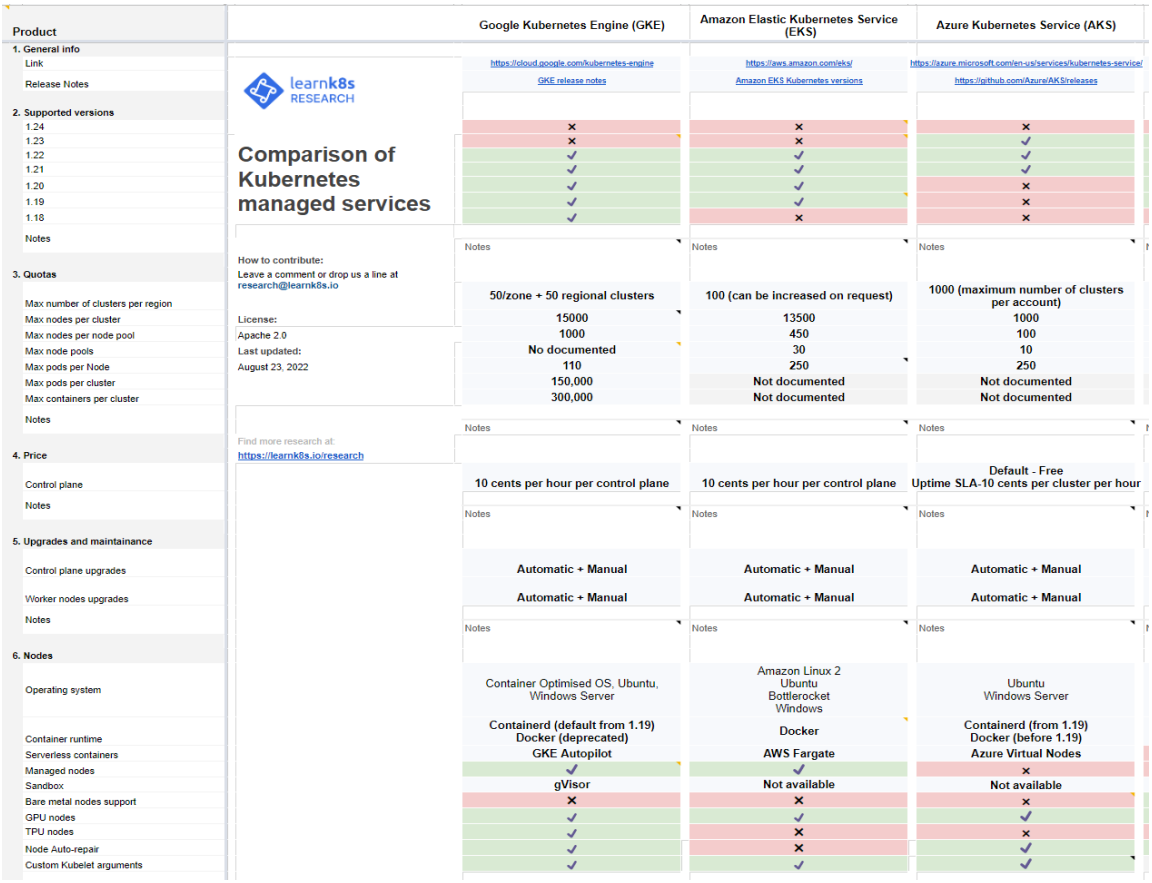
Der Autopilot von GKE ist im Grunde genommen ein Zwischenweg, zwischen Cloud Run, dem serverlosen Container Angebot von Google und dem Standard GKE-IaaS. Laut Entwicklerinnen und Entwicklern verhilft dieser Ansatz, die Grundverwaltungsvorgänge stark zu reduzieren. Das heißt, dadurch werden die Dauer der Vorgänge, wie automatische Skalierung, Ressourcenoptimierung, Node Updates und Bereitstellung von containerisierten Anwendungen, verkürzt.

Ein weiterer Punkt, dass erwähnt werden muss ist, dass durch den Einsatz des Autopilot auch anerkannte Sicherheitspraktiken wie Node-Identität und -Integrität, sowie auch die Restriktionen von Containerberechtigungen eingeführt werden. Außerdem werden durch GKE Autopilot, nur die verwendeten Ressourcen kalkuliert, anstatt alle Node die implementiert worden sind. Nichtsdestotrotz können die Sicherheitsvorkehrrungen vom Autopilot, auch gewisse Barrieren darstellen, wenn bei der Erstellung von Containern gewisse privilegierte Zugriffe aufgefordert werden.

(Hwang 2021)

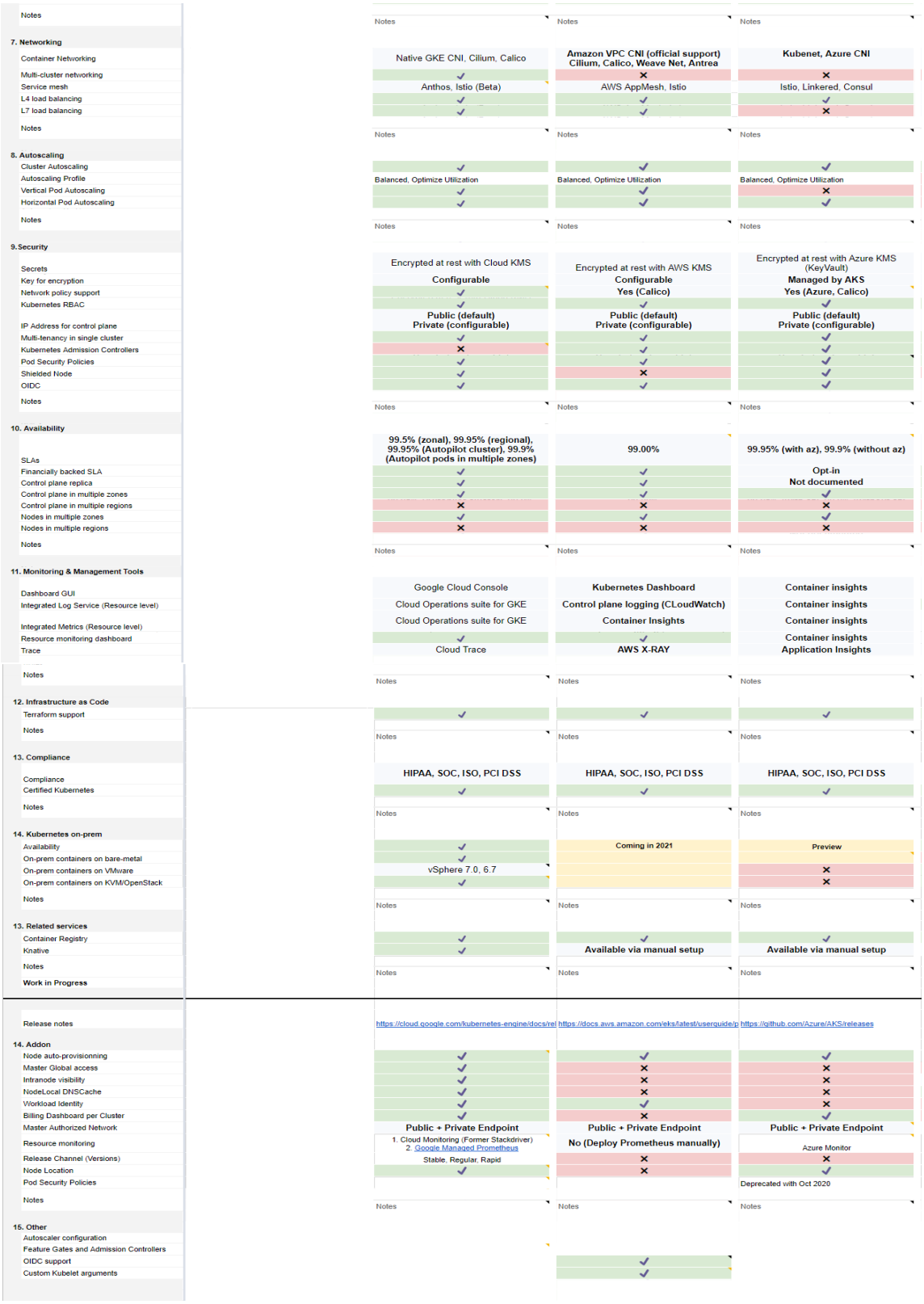
Vergleich der verwalteten Kubernetes Servicediensten

In Abbildung 6 und 7 geteilt befindet sich die Vergleichstabelle der verwalteten Kubernetes Servicediensten, von Hyperscaler, welche auch im Rahmen dieser Arbeit opportun sind.



Product		Google Kubernetes Engine (GKE)	Amazon Elastic Kubernetes Service (EKS)	Azure Kubernetes Service (AKS)
1. General info				
Link	learnk8s RESEARCH	https://cloud.google.com/kubernetes-engine	https://aws.amazon.com/eks/	https://azure.microsoft.com/en-us/services/kubernetes-service/
Release Notes	**Comparison of Kubernetes managed services**	GKE release notes	Amazon EKS Kubernetes versions	https://github.com/Azure/AKS/releases
2. Supported versions				
1.24		✗	✗	✗
1.23		✗	✗	✓
1.22		✓	✓	✓
1.21		✓	✓	✓
1.20		✓	✓	✗
1.19		✓	✓	✗
1.18		✓	✗	✗
Notes	How to contribute: Leave a comment or drop us a line at research@learnk8s.io	Notes	Notes	Notes
3. Quotas				
Max number of clusters per region		50/zone + 50 regional clusters	100 (can be increased on request)	1000 (maximum number of clusters per account)
Max nodes per cluster	License: Apache 2.0	15000	13500	1000
Max nodes per node pool		1000	450	100
Max node pools	Last updated: August 23, 2022	No documented	30	10
Max pods per Node		110	250	250
Max pods per cluster		150,000	Not documented	Not documented
Max containers per cluster		300,000	Not documented	Not documented
Notes	Find more research at: https://learnk8s.io/research	Notes	Notes	Notes
4. Price				
Control plane		10 cents per hour per control plane	10 cents per hour per control plane	Default - Free Uptime SLA-10 cents per cluster per hour
Notes		Notes	Notes	Notes
5. Upgrades and maintainance				
Control plane upgrades		Automatic + Manual	Automatic + Manual	Automatic + Manual
Worker nodes upgrades		Automatic + Manual	Automatic + Manual	Automatic + Manual
Notes		Notes	Notes	Notes
6. Nodes				
Operating system		Container Optimised OS, Ubuntu, Windows Server	Amazon Linux 2 Ubuntu Bottlerocket Windows	Ubuntu Windows Server
Container runtime		Containerd (default from 1.19) Docker (deprecated)	Docker	Containerd (from 1.19) Docker (before 1.19)
Serverless containers		GKE Autopilot	AWS Fargate	Azure Virtual Nodes
Managed nodes		✓	✓	✗
Sandbox		gVisor	Not available	Not available
Bare metal nodes support		✗	✗	✗
GPU nodes		✓	✓	✓
TPU nodes		✓	✗	✗
Node Auto-repair		✓	✗	✓
Custom Kubelet arguments		✓	✓	✓

Abbildung 6 Vergleichstabelle der managed Kubernetes Services Teil 1, (learnk8s 2023)



Notes	Notes	Notes	Notes
7. Networking			
Container Networking	Native GKE CNI, Cilium, Calico	**Amazon VPC CNI (official support) Cilium, Calico, Weave Net, Antrea**	**Kubenet, Azure CNI**
Multi-cluster networking	✓	✗	✗
Service mesh	Anthos, Istio (Beta)	AWS AppMesh, Istio	Istio, Linkered, Consul
L4 load balancing	✓	✓	✓
L7 load balancing	✓	✓	✗
Notes	Notes	Notes	Notes
8. Autoscaling			
Cluster Autoscaling	✓	✓	✓
Autoscaling Profile	Balanced, Optimize Utilization	Balanced, Optimize Utilization	Balanced, Optimize Utilization
Vertical Pod Autoscaling	✓	✓	✗
Horizontal Pod Autoscaling	✓	✓	✓
Notes	Notes	Notes	Notes
9. Security			
Secrets	Encrypted at rest with Cloud KMS	Encrypted at rest with AWS KMS	Encrypted at rest with Azure KMS (KeyVault)
Key for encryption	**Configurable**	**Configurable**	**Managed by AKS**
Network policy support	✓	✓	✓
Kubernetes RBAC	✓	**Yes (Calico)**	**Yes (Azure, Calico)**
IP Address for control plane	**Public (default) Private (configurable)**	✓	✓
Multi-tenancy in single cluster	✓	**Public (default) Private (configurable)**	**Public (default) Private (configurable)**
Kubernetes Admission Controllers	✗	✓	✓
Pod Security Policies	✓	✓	✓
Shielded Node	✓	✗	✓
OIDC	✓	✓	✓
Notes	Notes	Notes	Notes
10. Availability			
SLAs	**99.5% (zonal), 99.95% (regional), 99.95% (Autopilot cluster), 99.9% (Autopilot pods in multiple zones)**	**99.00%**	**99.95% (with az), 99.9% (without az)**
Financially backed SLA	✓	✓	**Opt-in**
Control plane replica	✓	✓	**Not documented**
Control plane in multiple zones	✓	✓	✓
Control plane in multiple regions	✗	✗	✗
Nodes in multiple zones	✓	✓	✓
Nodes in multiple regions	✗	✗	✗
Notes	Notes	Notes	Notes
11. Monitoring & Management Tools			
Dashboard GUI	Google Cloud Console	**Kubernetes Dashboard**	**Container insights**
Integrated Log Service (Resource level)	Cloud Operations suite for GKE	**Control plane logging (CLoudWatch)**	**Container insights**
Integrated Metrics (Resource level)	Cloud Operations suite for GKE	**Container Insights**	**Container insights**
Resource monitoring dashboard	✓	✓	**Container insights**
Trace	Cloud Trace	**AWS X-RAY**	**Application Insights**
Notes	Notes	Notes	Notes
12. Infrastructure as Code			
Terraform support	✓	✓	✓
Notes	Notes	Notes	Notes
13. Compliance			
Compliance	**HIPAA, SOC, ISO, PCI DSS**	**HIPAA, SOC, ISO, PCI DSS**	**HIPAA, SOC, ISO, PCI DSS**
Certified Kubernetes	✓	✓	✓
Notes	Notes	Notes	Notes
14. Kubernetes on-prem			
Availability	✓	**Coming in 2021**	**Preview**
On-prem containers on bare-metal	✓		
On-prem containers on VMware	vSphere 7.0, 6.7		✗
On-prem containers on KVM/OpenStack	✓		✗
Notes	Notes	Notes	Notes
13. Related services			
Container Registry	✓	✓	✓
Knative	✓	**Available via manual setup**	**Available via manual setup**
Notes	Notes	Notes	Notes
Work in Progress			
Release notes	https://cloud.google.com/kubernetes-engine/docs/rel	https://docs.aws.amazon.com/eks/latest/userguide/p	https://github.com/Azure/AKS/releases
14. Addon			
Node auto-provisionning	✓	✓	✓
Master Global access	✓	✗	✗
Intranode visibility	✓	✗	✗
NodeLocal DNSCache	✓	✗	✗
Workload Identity	✓	✓	✗
Billing Dashboard per Cluster	✓	✗	✓
Master Authorized Network	**Public + Private Endpoint**	**Public + Private Endpoint**	**Public + Private Endpoint**
Resource monitoring	1. Cloud Monitoring (Former Stackdriver) 2. Google Managed Prometheus	**No (Deploy Prometheus manually)**	Azure Monitor
Release Channel (Versions)	Stable, Regular, Rapid	✗	✗
Node Location	✓	✗	✓
Pod Security Policies			Deprecated with Oct 2020
Notes	Notes	Notes	Notes
15. Other			
Autoscaler configuration		✓	
Feature Gates and Admission Controllers		✓	
OIDC support			
Custom Kubelet arguments			

Abbildung 7 Vergleichstabelle der managed Kubernetes Services Teil 2, (learnk8s 2023)

3. Die Nutzwertanalyse (NWA)

Im Kapitel „Die Nutzwertanalyse (NWA)“ widme ich mich auf die Definition der Nutzwertanalyse und werde weiters auf dessen Anwendungsbereiche begeben. Daraufhin folgt, auf welche Grundvorkehrungen es bei der Erstellung einer Nutzwertanalyse zu achten ist und eine allgemeine Definition des generellen Ablaufs einer Nutzwertanalyse. Der detaillierte Ablauf einer Nutzwertanalyse wird erst im nächsten Kapitel im Zusammenhang mit den verwalteten Kubernetes Services durchgeführt.

3.1 Die Definition der Nutzwertanalyse

Die Nutzwertanalyse ist eine Variation der Analyse, in welcher mehrere Alternativen, mit dem Zweck, die Elemente der Alternativen dazugehörend mit den Tendenzen des Entscheidungsträgers, im Hinblick der Ausgangsbasis zu kategorisieren. Das heißt, es ist ein Bewertungsverfahren, in welcher die Beurteilung der Alternativen stattfindet, wobei nicht vergessen werden darf, dass für die Durchführung der Analyse nur mehrdimensionale Zielsysteme in Betracht gezogen werden dürfen. Mehrdimensionale Zielsysteme befürworten auch mehrere Maßstäbe, wobei eindimensionales Bewertungsmaßstab nicht für die Tätigung einer Nutzwertanalyse ausreichend ist. Unter eindimensional darf ein Produkt oder ein Objekt verstanden werden, welches nur eine einzige Eigenschaft besitzt. In Worstcase führt dies zu einem Problem der Wertsynthese und ist erfüllt nicht die Bedingungen des Verfahren der Nutzwertanalyse. (Rinza and Schmitz 1992)

Des Weiteren, dient das Verfahren als ein Entscheidungsmodell, welches mit einer Entscheidung hilfreichen Ansatz Auswahlprobleme löst. Außerdem unterscheidet sich die NWA von weiteren Ermittlungsmodellen, wie zum Beispiel von einer Investitionsrechnung. Bei einer Entscheidungssituation, wird in die Investitionsrechnung, die Rolle der Entscheidungsvorbereitung zugewiesen. (Zangemeister 2014)

3.1.1 Anwendungsbereiche der NWA

Die Einsatzmöglichkeit der NWA hat sich heutzutage nahezu in allen Bereichen der Technologie, der Ökonomie und ebenso in allen Bereichen des öffentlichen Lebens etabliert. Ihre Hilfe, Entscheidungsträgern durch die überzeugenden Argumentationen die Vor- und Nachteile der unterschiedlichen Alternativen aufzuzählen hat sich inzwischen allgemeingültig qualifiziert. Die Tabelle 1

visualisiert die Anwendungsbereiche der Nutzwertanalyse und hat keinen Anspruch auf Vollständigkeit und stellt die Vielfältigkeit des Verfahrens dar. Da die angegebene Literatur etwas älter ist, kann durchaus angenommen werden, dass sich die NWA auf mehrere Bereichen ausgeweitet hat. (Rezanka 2013)

Tabelle 1 Anwendungsbereiche der Nutzwertanalyse (Rezanka 2013)

Entscheidung über Forschung- und Entwicklungsvorhaben	Strebel, Scoring Modelle, 1975; Dean/Nishry, Soring, 1965; Moore/Baker , Scoring, 1969; Schweizer, Forschungsplanung 1971
Auswahl neuer Produkte	Hirsch, Bewertungsprofil, 1968; Schermerhorn/ Taft, Measuring, 1968
Beurteilung der Funktionsfähigkeit technischer Systeme	Kiener, Nutzwertanalyse, 1974; Schoenfelder, Nutzwertanalyse, 1976
Auswahl von EDV-Anlagen	Lehner, Microcomputer- Auswahl 1982; Hansen/Bauer, Computerauswahl, 1980
Investitionsplanung im privaten und öffentlichen Bereich	Swart/ Rubenstein/ Burgess, Quantifying, 1973; Weiss, Grants management, 1973
Regionalplanung	Strassert/ Turkowski, Nutzwertanaylse, 1971; Affeld/Strassert/ Turkowski, Kommunalstruktur, 1974
Freizeit und Fremdenverkehrsplanung	Zaus, Bewertungsgrundlage, 1975
Stadtentwicklungsplanung	Iblher/Jansen, Bewertung, 1972
Standortplanungen	Calvo/ Marks, Location, 1973
Umweltschutzplanung	Ellis/ Keeney, Air pollution, 1972
Planungen im Kommunalbereich	Kunze/ Blanek/ Simons, Nutzwertanalyse 1969
Arbeitsplatzzuordnung und Berufswahl	Kunze/ Blanek / Simons, Nutzwertanalyse 1969; Huber/Daneshgar/ Ford, Utility models, 1971
Beurteilung unterschiedlicher Verkehrssysteme	Weigelt, Verkehrswert, 1973
Beurteilung einer Organisationsstruktur	Knight/ Wegenstein, Effektivität, 1971

3.2 Voraussetzungen und die Verfahrensschritte

Im Grunde genommen sind bei der Nutzwertanalyse auf drei rationale Prinzipien zu achten. Auf diese sind erst nach der Teilung in die Teilnutzen der Lösung zu richten.

Die Prinzipien sind:

- Alternativen sind durch betrachtende Bewertung des Zielerreichungsgrades direkt zu begutachten
- Teilbereiche der Alternativen werden separat begutachtet. Zielkriterien werden zuerst aufgespalten und schrittweise zu jedes Kriterium der Alternativen vergleichend bewertet und geordnet.
- Teilbewertungen sind zu einem Gesamtwert zu summieren.

<u>Weitere Voraussetzungen zur Lösung der Nutzwertanalyse:</u>

- Kriterien sind durch eindimensionale Funktionen zu bewerten
- Der Nutzen der Alternative ist dimensionslos, keine physisch definierte Dimension, wie km/h, kg und vieles mehr.
- Zielerfüllungsgrade sind eindeutig und konstant gewichtet. Die Nutzenskalierung ist kardinal durchzuführen.
- Bestehung der Nutzungsunabhängigkeit zwischen den Zielkriterien
- Für die Berechnung der Alternative wird Gesamtnutzen geteilt durch die Teilbewertung.

(Zangemeister 2014)

In der NWA müssen die Nutzenbeiträge der Alternativen, welche aus ihren abgestimmten Eigenschaften ergeben, erfasst werden. Um diese erfassen zu können ist es sehr opportun, die für die Entscheidung wichtige Größen zu kennen und zu ordnen. Dadurch ist es möglich, die Alternativen, entsprechend den Paradigmen des Entscheidungsträgers miteinander zu vergleichen. Durch diese Vorbereitung wird der gesamte Entscheidungsraum in kleine Teile zerlegt und überschaubar. So wird das Gesamtziel, in Teilziele und weiters in Unterziele und diese wiederum in Kriterien ihrer Erfüllung segmentiert.

Das heißt:

$$N_{ges} = f(N_1, N_2, N_3, \dots N_n) \qquad (1)$$

(N_{ges}) stellt den Gesamtnutzwert einer Alternative, (N_i) dagegen, den Nutzwertbeitrag des Kriterium. Die Verknüpfung der Nutzwertbeiträge hat aufgrund der Zuverlässigkeit des Beurteilungsergebnisses einen großen Einfluss und ist aus diesem Grund auch wichtig. Am häufigsten wird die Verknüpfung mit einer Addition der einzelnen Nutzwertbeiträge definiert. Die Formel dafür sieht wie gefolgt aus:

$$N_{ges} = \sum_{i=1}^{n} N_i \qquad (2)$$

Anhand dieser Formel, lassen sich die einzelnen Nutzwertbeiträge sehr leicht gegeneinander aufsummieren, so dass zwischen einem geringen und einen hohen Nutzwertbeitrag ein Balance beim Bewertungskriterium entsteht. Die Formel ist auch unter dem Namen „Aufrechenbarkeit des Nutzens oder Substitution zwischen den Zielen“ bekannt.

Im Grunde genommen kann gesagt werden, dass der Beitrag in einer NWA, sich aus dem Grad ergibt und bekannt gibt, wie lückenlos ein gewisses Ziel erreicht ist. Des Weiteren kann aus dem Nutzwertbeitrag noch die Bedeutung des Entscheidungsträgers bemessen werden, denn es gilt:

$$N_i = w_i * E_i \qquad (3)$$

In dieser Gleichung steht (w_i) für die Bedeutung oder gar für die Gewichtung des Kriteriums im Gesamtnutzen. Der Erfüllungsgrad des Kriteriums, beschreibt die Erfüllung der Eigenschaft des Kriteriums und ist in der Gleichung mit (E_i) gekennzeichnet. Durch die Gleichung (3) kann manifestiert werden, dass sowohl bei geringe Erfüllungsgradänderungen als auch bei große Erfüllungs-gradänderungen, die Nutzwertänderungen konstant bleiben. Erfüllungsgrade sind in der NWA sehr von den bewertenden Leistungen abhängig und deshalb gilt:

$$E_i = f(L_i) \qquad (4)$$

Daraus folgt, dass (N_i) auch folgendermaßen definiert werden kann:

$$N_i = W_i \,.\, E_i = W_i \,.\, f(L_i) \qquad (5)$$

somit kann der Gesamtnutzwert durch diese Formel ermittelt werden:

$$N_{ges} = \sum_{i=1}^{n} w_i \cdot E_i = \sum_{i=1}^{n} w_i \cdot f(L_i) \qquad (6)$$

(Rinza and Schmitz 1992)

3.3 Ablauf der Nutzwertanalyse

Allgemein wird eine Nutzwertanalyse in sieben Teilschritten durchgeführt. Die Schritte müssen für eine allgemeine Gültigkeit schrittweise durchgeführt werden damit es lückenlos ausgeführt werden kann.

Diese sind:

1.Schritt Aufstellung des Zielsystems:
Als ersten werden die Bewertungsziele hierarchisch gegliedert aufgelistet. Durch diese Auflistung wird das Zielsystem aufgestellt. Wichtig ist hierbei, dass bei der Auflistung, die Ziele logisch verknüpft aufgeführt werden.

2.Schritt Gewichtung:
Der nächste Schritt für die Durchführung der NWA ist die Definierung der Gewichte der Ziele. Zielen werden relative Gewichte zugewiesen, so dass dessen Gesamtgewicht 100 Prozent ergeben.

3.Schritt Aufstellen der Wertefunktionen und Wertetabellen:
Als nächstes folgt die Aufstellung der Wertetabellen oder Wertefunktionen. Diese dienen zur objektiver Ermittlung der für die Bewertung aufgestellten Alternativen. Durch ihre Verwendung sollen Manipulationen an der Ermittlung weggeschaffen werden. Manipulationen könnten sein, Kenntnisse der Eigenschaft der Alternativen, die den Zusammenhang zwischen Erfüllungsgrad der Alternative und Eigenschaft der Alternative ausdrücken können.

4.Schritt Bestimmung und Bewertung der Alternativen:
Erst im Schritt vier der Nutzwertanalyse, findet die Bewertung der Alternativen statt. Die vorgeschlagenen Alternativen werden zum ersten Mal im Schritt vier vorgelegt. Anschließend folgt eine Zusammenstellung ihrer Eigenschaften anhand des vordefinierten Zielsystems. Im letzten Schritt des vierten Schritts, werden die Eigenschaften der Alternativen mithilfe der im Schritt drei erstellten Wertetabellen oder Wertefunktionen in Erfüllungsgrade der Eigenschaft transformiert.

5.Schritt Berechnung der Nutzwerte und Ermittlung der Rangfolge:
Nach der Umsetzung der Erfüllungsgrade folgt die Berechnung der Nutzwerte. Mithilfe der Kombinationen im Schritt zwei definierter Gewichtung und im Schritt vier umgesetzte Erfüllungsgrade, können die Nutzwertbeiträge kalkuliert und zum (N_{ges}) aufsummiert werden.

6.Schritt Empfindlichkeitsanalyse:
Lediglich wird für die Dauerhaftigkeit der Ergebnisse der Nutzwerte eine Empfindlichkeitsanalyse vorgenommen. Da vor allem bei der Gewichtung und bei der Aufstellung der Wertetabelle oder Wertefunktionen, die nicht quantifizierbare Eigenschaften der Kriterien, in subjektiven Momenten eine große Rolle spielen, führt diese Analyse zu Stabilität der Ergebnisse.

7.Schritt Beurteilung und Darstellung des Ergebnisses der Nutzwertanalyse:
Im letzten Schritt der NWA, findet die Evaluierung und Darstellung der Ergebnisse statt. Die Resultate aus Schritt fünf und Schritt sechs ermöglichen eine Benotung der Alternativen. Die Ergebnisse können in Form eines Diagramms wiedergeben werden.

(Rinza and Schmitz 1992)

4. Das methodische Vorgehen in AKS, EKS und GKE

Am Anfang des zweiten Abschnitts der Bachelorarbeit wird auf die Begriffsdefinition der Funktionen von verwalteten Kubernetes Services eingegangen und geschildert. Der Hauptfokus liegt auf bestimmte Funktionen, der verwalteten Kubernetes Services, die im Rahmen dieser Arbeit opportun sind und die Beantwortung der Forschungsfrage befürworten. Daraufhin folgt die Erklärung der Plausibilitätstheorie, in welcher Logik die sieben Schritte der Nutzwertanalyse aufgebaut werden. Abschließend folgt die Beschreibung der Vorgehensweise, wie das Prototyping stattfinden wird und zum Schluss die Screenshots zu den Prototyp-Implementierungen von verwalteten Kubernetes Services, je Cloud Provider (Azure, Amazon Web Service und Google Cloud Platform).

4.1 Beschreibung der Funktionen

Wie oben im Kapitel 2.4.5 auf Abbildung sechs und sieben dargestellt, beinhalten die verwalteten Kubernetes Services eine Vielzahl an Funktionen, die als Dienstleistungen repräsentiert werden, doch um die Bachelorarbeit erfolgreich zu vollenden wird der Fokus stark auf folgende sieben Funktionen reduziert. Diese sind:

- Command Line Interface (CLI),
- Spawn Cluster Zeit,
- Kubernetes Versionsunterstützung,
- Monitoring,
- rollenbasierte Zugriffssteuerung (RBAC),
- Überwachung der Knotenintegrität und
- Preisgestaltung.

Vorab muss erwähnt werden, dass, um den Rahmen der Bachelorarbeit nicht mit Begriffsschilderungen zu überziehen, gewisse Begriffe, teilweise bis high Level behandelt werden, so dass die Leserschaft, in der Endphase der Bachelorarbeit , die Analyse der Nutzwertanalyse und dessen Kontingent nachvollziehen kann.

- **Begriffsdefinition und Vorhebung der Funktionen**

  Command Line Interface (CLI)

  Wird als eine Befehlszeilenschnittstelle bezeichnet und ist eine textbasierte Benutzeroberfläche für den Gebrauch des Anwenders. Die Schnittstelle wird zum Ausführen von Programmen, zum Verwalten von Dateien und zum Interagieren mit dem Computer oder der Computerressourcen verwendet. Ihre Verwendung wird sehr einfach gehalten. Mithilfe von eingegebenen Befehlen, können dem Endgerät, Befehle zur Anwendung und oder zur Steuerung, kommandiert werden.

  Diese werden durch den Computer durch- & ausgeführt. Im Prinzip sind die CLIs unerlässlich, doch mittlerweile bieten alle Cloud Provider zusätzlich auch eine grafische Benutzeroberfläche, um Anwendungen für die User, augenfreundlicher zu gestalten. Eine solche Benutzeroberfläche wird auch als graphical User Interface (GUI) bezeichnet.

  (TechTarget 2023)

  Dienste, welche bei Hyperscaler, nicht auf dem ersten Blick gefunden werden und neue Dienste, dessen Icon noch nicht auf der Benutzeroberfläche zu finden sind, können via Befehlszeilenschnittstelle aufgefunden und ausgeübt werden. Die CLIs der Hyperscaler müssen stets aktuell sein, um Dienste schnell wie möglich ihren Usern gewähren zu können.

  (Amazon.com AWS 2023)

  Spawn-Cluster Zeit

  Wie oben im Kapitel **2.3.6 Tools zur Container Orchestrierung**, in der Basis Kubernetes Architektur, beschrieben, ist ein Cluster als ein Ganzes zu betrachten. Das heißt, ein Cluster enthält somit die Kombination von Anwendungen und eine physische oder eine virtuelle Maschine.

  Doch das Innenleben, eines sogenannten Kubernetes Clusters, besteht mindestens aus zwei Arten von Nodes. Die Komponente dieser Nodes sind, bereits im Kapitel 2.3.6 unter der Basis Kubernetes Architektur beschrieben.

Die zwei Arten der Nodes, je Komponente sind wie gefolgt unterteilt:

- Master-Node
    - API-Server
    - Scheduler
    - Controller
    - ETCD
- Worker-Node
    - Pods und Container
    - Kubelet
    - Kube- Proxy

Etwas genauer beschrieben ist der Master Node in einem Kubernetes Cluster als die Steuerungseinheit zu sehen. Diese enthält Konfigurations- und Zustandsdaten, die zur Aufrechterhaltung des gewünschten Zustands verwendet werden. Mithilfe des Master Nodes, wird die Kommunikation zu den Worker Nodes aufrechterhalten. äAnhand dieser können Container innerhalb des Worker Nodes besser geplant werden.

Wie der Name schon verrät, sind Worker Nodes, die sogenannten Nodes, die tatsächlich die Arbeit leisten. Sie führen die Pods aus. In den Pods sind in der Regel, die einzelne Instanzen einer Anwendung. Wie zum Beispiel die Applikation selbst, mit ihrer Bibliothek zusammen. Die Applikation einer Anwendung mit dessen geeigneter Bibliothek wird in einem Container bereitgestellt. Dieser Container wird dann in den Pods, die zur Verfügung stehen, ausgeführt.

Ein Pod, dass mit Containern belagert ist, wird in einem Worker Node ausgeführt. Üppig ist, dass bei großen Applikationen aufgrund der Redundanzgründen, ein Kubernetes Cluster aus mindestens drei Worker Nodes besteht.

(Roper 2022)

Zusätzlich zu der Standardprozedur einer Cluster Erstellung, kommen noch weitere Merkmale Zustande, welche definiert werden müssen, damit in den Cloud Service Providern generell ein verwaltetes Kubernetes Service erstellt werden kann.

Diese sind bei minimaler Einhaltung der Merkmale, dass Lokalisierungstyp der zu erstellenden verwalteten Service; die Release Version von Kubernetes; das Betriebssystem der virtuellen Maschine als Image-Typ innerhalb eines Clusters und die Konfiguration der VM.

Das heißt, die Spawn Cluster Zeit, gibt die Dauer an, wie lange ein Hyperscaler benötigen würde, eine vom User definierte Konfiguration bereitzustellen.

Kubernetes Versionsunterstützung

Die Versionen in Kubernetes werden als x.y.z Parameter benannt. Hierbei steht die X, für die Hauptversion, Y für die Nebenversion und Z gilt als die Patchversion. Aktualisierung der Versionen, sind hauptsächlich für Problemlösungen, Sicherheitsumkehrungen und für Erneuerung in Kubernetes gedacht. Viele Clusterbereitstellungstools, wie AKS, EKS und GKE unterstützen zwar die aktuellen Versionen von Kubernetes, dennoch sind unter ihnen, bedingte Versionsverzerrungen vorhanden.

(kubernetes.io 2023)

Monitoring

Das Monitoring via Dashboards in der Cloud ist als ein Prozess der Überprüfung, Steuerung und sowie die Verwaltung der betrieblichen und aktiven Arbeitsläufe, als auch die Prozesse innerhalb der Cloud-Infrastruktur zu sehen. Durch dessen Einsatz per manueller und oder automatisierter Überwachungs- und Verwaltungstechnik, hilft es den Benutzer: innen, bei der Sicherstellung, dass die bereitgestellte Infrastruktur oder die Plattform in optimierter Leistung reibungslos betrieben werden kann.

Die reibungslose Cloud Operationen, welche durch Dashboards gewährleistet werden können, sind, Operationen wie gefolgt:

- Buchhaltung,
- Abrechnung,
- SLA-Verwaltung (Service Level Agreement),
- Dienst- & Ressourcenbereitstellungen,
- Kapazitätsplanung,
- Management der Konfiguration,

- Sicherheit, Datenschutz und
- Fehlermanagement.

(Bulla 2019)

Rollenbasierte Zugriffssteuerung (RBAC)

Der role-based access Control (RBAC), in Deutsch übersetzt, die rollenbasierte Zugriffssteuerung, ist als eine Identitäts- und Zugriffsverwaltung zu verstehen. Mithilfe der RBAC, werden Benutzer: innen in der Cloud, eine Reihe von Berechtigungen und Restriktionen festgelegt.

Restriktionen wie, welche Benutzer: in, als Subjekt gesehen, was für eine Tätigkeit als Verb gesehen, wo in Form von welcher Ressourcengruppe als Namenspace verstanden, ausführen darf.

Überwachung der Knotenintegrität

Die Knotenintegrität kann auch mit der Integrität der Nodes in einem Cluster assoziiert werden. Von einer Knotenintegrität wird gesprochen, wenn gewisse Vorkehrungen mit der Konfiguration, übereinstimmen.

Das heißt, um zu überprüfen, ob ein bereitgestellter Node, reibungslos funktioniert, wird überprüft:

- Ob jedes Node, welches bereitgestellt ist, auch eine virtuelle Maschine ist, welches sich in einem Cluster befindet;
- Ob jedes Node in einem erstellen Projekt in der Cloud, zu einer bestimmten Ressourcengruppe beziehungsweise zu einer verwalteten Instanz-Gruppe angehörig ist;
- Ob jedes Node, im Kubelet des jeweiligen verwalteten Kubernetes Services, ein eigenes Zertifikat zugewiesen worden ist.

(Google Cloud 2023)

Preisgestaltung

Hyperscaler bieten in der Cloud verschiedene Möglichkeiten zur Bezahlung der bereitgestellten Modelle. Hauptsächlich bestehen die Preise aus Nutzungspreise für verwendete Ressourcen. Die Preise für das Ressourcen, modellieren sich durch die Laufwerksgröße, der Arbeitsspeicher (RAM) des Maschinentyps und die Netzwerknutzung in Gigabyte. Jegliche Feature und oder Funktionen die erweiterbar sind, können Hyperscaler abhängig sein, ob diese in die Preise inkludiert werden.

(Google Cloud 2023)

4.2 Beschreibung der Methode

Wie auch im **Kapitel 3.1 die Definition der Nutzwertanalyse**, schon beschrieben, kann die Ermittlung der Nutzwertanalyse sehr vielfältig sein. In der Analyse selbst, werden mehrere Alternativen, mit dem Zweck, die Elemente der Alternativen und zusätzlich mit den dazugehörenden Tendenzen des Entscheidungsträgers, im Hinblick auf die Ausgangslage analysiert.

Anschließend werden die erlangten Zwischenergebnisse summiert und somit die Nutzwerte realisiert. Daraufhin folgt die Erstellung der Rangfolge. Anhand der Rangfolge kann definiert werden, welche Alternative den höchsten Nutzen erzielt.

Im nächsten Schritt ist es zu bedenken, dass der allgemeingültige Aufbau der Nutzwertanalyse, durch die sieben Teilschritte definiert ist. Jedes dieser einzelnen Schritte bietet mehrere Verfahren an, die es Benutzer: innen ermöglicht, durch die Anwendung verschiedener möglichen Methoden, die Ziele der Teilschritte zu erreichen. Die Ziele der Teilschritte sind klar nachvollziehbar. Sie dienen der Bedingungen der einzelnen Schritte der Nutzwertanalyse und führen Benutzer: innen zum nächsten Zwischenschritt.

Ein wichtiger Punkt, welcher noch angerissen gehört ist, dass die Nutzwertanalyse und die Kosten-Nutzen-Analyse sehr ähnlich aufgebaut sind, aber sich dennoch in einem Punkt leicht unterscheiden lassen. In der Nutzwertanalyse, wie schon der Name verrät, wird der Nutzen analysiert, während in der Kosten-Nutzen-Analyse, sowohl der Nutzen als auch die Kosten ermittelt werden. Bei Kriterien, in welchen die Kosten oder Beträge der Alternativen ermittelt werden, trennen sich die Ermittlungswege in zwei verschiedene Richtungen. Kurzgefasst werden hierbei die Rahmenbedingungen

der zu erstellenden Analyse erweitert, und es finden mathematische Berechnungen statt, in denen diverse Formeln zur Anwendung kommen.

(Rinza and Schmitz 1992)

Für die sieben Schritte, die wie gefolgt nochmals aufgezählt werden:

- Aufstellen des Zielsystems,
- Festlegung der Gewichte,
- Aufstellen der Wertetabelle oder Wertefunktionen,
- Bewertung der Alternativen,
- Berechnung der Nutzwerte,
- Empfindlichkeitsanalyse (optional) und
- Beurteilung und Darstellung der Ergebnisse,

gibt es ebenso Rahmenbedingungen, auf welche bei der Erstellung einer Nutzwertanalyse zu achten sind, da sonst die Analyse als, manipuliert gelten würde oder ihre Gültigkeit verliert.

(Rinza and Schmitz 1992)

4.2.1 Rahmenbedingungen für die sieben Schritte der Nutzwertanalyse

Aufstellen des Zielsystems:

Das Aufstellen des Zielsystems ist ein kreativer Vorgang und besitzt aus diesem Grund keine allgemeingültige Methode, doch es herrschen Präferenzen, die bevorzugt werden dürfen, um eine Strukturierung und Ordnung in das Zielsystem zu vermitteln. Die Ziele des Zielsystems können am günstigsten erstellt werden, wenn diese in hierarchischer Form als ein Strukturplan dargestellt werden. Hierarchisch bedeutet, dass Kategorie des Oberziels, Sub-Ziele aufweist und die Sub-Ziele, weitere Sub-Ziele beinhalten.

Die Anzahl der Kriterien, die das Zielsystem und die Zielebenen, hierarchisch befüllen sollen, sind von der Wichtigkeit des Bewertungssystems und von der Investition selbst abhängig. Je höher die Anzahl der Alternativen, die hier verglichen werden, desto höher muss die Anzahl der Kriterien sein, die hier überprüft werden müssen, damit die Zielhierarchie weder zu grob noch zu fein ist und die Analyse zu einer Grobabschätzung führen kann.

Weiters sind noch auf folgende Forderungen zu achten:

- Vollständigkeit:
    - Ziele im Zielsystem müssen sinnvoll geordnet sein
- Angemessenheit:

    - Anzahl der Ziele im Zielsystem muss dem Entscheidungsumfang gemessen sein.
- Systematik:
    - Kostenkriterien dürfen nicht im Zielsystem sein, falls doch, müssen diese bei Bedarf in nicht monetäre Kriterien umgewandelt werden.
- Unabhängigkeit:
    - Zwei Zielkriterien dürfen nicht dieselbe Eigenschaft beschreiben.

Festlegung der Gewichte:

Nach der Erstellung des Zielsystems mit je sinnvoll gefundenen Zielebenen, müssen nun die Oberziele und die Sub-Ziele der Oberziele gewichtet werden. Damit die Gewichtung und die Berechnung der Nutzwerte gemäß der Gewichtung sinnvoll und nachvollziehbar werden, sollen nicht mehr als vier Zielebenen erstellt werden, außer es ist aus opportunistischen Gründen bedeutsam.

Weiteres muss bei der Gewichtung geachtet werden, dass bei Definierung der Knotengewichte, also Gewichte der Oberziele und Stufengewichte, Gewichte der Sub-Ziele, in Summe je 100% ergeben. Die Summe ergibt entweder in Prozent angegeben 100% oder die 1,0. Bei Abweichung der Gewichtung, meistens ist hierbei die Stufengewichtung betroffen, müssen diese entweder korrigiert oder bei der Ermittlung der Nutzwerte bedacht kalkuliert und miteinbezogen werden.

Die Gewichtung kann durch mehrere Methoden erfolgen. Diese sind:

- Gewichtung mit einem absoluten Maßstab,
- Einfache singuläre Vergleich,
- Verfeinerte singuläre Vergleich,
- Sukzessive Vergleich und
- Matrix-Verfahren.

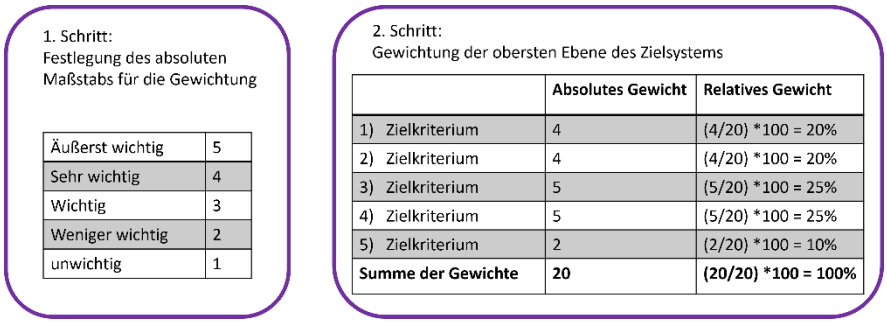
Die oben angeführte Auflistung weist der Reihenfolge nach, wie einfach und genau die Methode bei der Durchführung der Gewichtung ist.

Sowohl im Fallbeispiel Nutzwertanalyse - Kauf eines Personenkraftwagens aus dem Buch, welches zitiert wird und auch im Rahmen der Bachelorarbeit, der absoluter Maßstab verwendet.

(Rinza and Schmitz 1992)

Im ersten Schritt der Abbildung 8, wird die allgemeingültige Gewichtung festgelegt. Im nächsten Schritt findet die Berechnung der relativen Gewichte der Oberziele statt. Je nach Anzahl der Sub-Ziele unter einem Hauptziel, muss kalkulatorisch wieder die 100% nachgewiesen werden.

Angenommen ein Oberziel besteht in zweiter Zielebene nur aus zwei Subzielen, so besteht das Knotengewicht je Subziel aus 50% und ergibt in Summe die 100% wieder.



1. Schritt:
Festlegung des absoluten Maßstabs für die Gewichtung

Äußerst wichtig	5
Sehr wichtig	4
Wichtig	3
Weniger wichtig	2
unwichtig	1

2. Schritt:
Gewichtung der obersten Ebene des Zielsystems

	Absolutes Gewicht	Relatives Gewicht
1) Zielkriterium	4	(4/20) *100 = 20%
2) Zielkriterium	4	(4/20) *100 = 20%
3) Zielkriterium	5	(5/20) *100 = 25%
4) Zielkriterium	5	(5/20) *100 = 25%
5) Zielkriterium	2	(2/20) *100 = 10%
Summe der Gewichte	**20**	**(20/20) *100 = 100%**

Abbildung 8 absoluter Maßstab (Rinza and Schmitz 1992)

Aufstellen der Wertetabelle oder Wertefunktionen:

Wie im **Kapitel 3.2 Voraussetzungen und Verfahrensschritte** (3) erwähnt, erfolgt die Berechnung der Nutzwert (N_i) aus dem Gewichtfaktor (w_i) und dem Erfüllungsgrad (E_i).

$$N_i = w_i * E_i \qquad (3)$$

Doch um den Erfüllungsgrad zu errechnen, wird eine Wertetabelle oder eine Wertefunktion benötigt. Mithilfe der Wertetabellen oder Wertefunktionen, werden die unterschiedlichen Dimensionen der zu bewertenden Kriteriums, in eine einheitliche Dimension umgewandelt und wird ein Wert zugeteilt. Anhand des Wertes gelingt es Benutzer: innen einer Nutzwertanalyse, die Zielerfüllung darzustellen.

Bevor die Wertetabellen und Wertefunktionen erstellt und generiert werden, muss überlegt werden, welche Zielkriterien wie ermittelt werden können. Aus diesem Grund muss unterschieden werden, was genau eine Wertefunktion und eine Wertetabelle ist.

In einer Wertefunktion lassen sich physikalische Zielkriterien messen. Sie wird angewendet, wenn von quantifizierbaren Kriterien gesprochen wird.

Quantifizierbare Kriterien sind, Kriterien, die abgemessen werden können wie, Kriterien, welche die Geschwindigkeit ansprechen und in km/h bemessen werden können, eventuell auch Kriterien, die anhand eines Gewichtes bemessen werden können wie in Kilogramm (Kg).

Wertefunktionen werden bei einer einfachen Nutzwertanalyse wenn möglich, per lineare Funktionen dargestellt. Sowie in Abbildung 9 dargestellt, gibt es drei verschiedene Arten, die zur Anwendung kommen können.

(Rinza and Schmitz 1992)

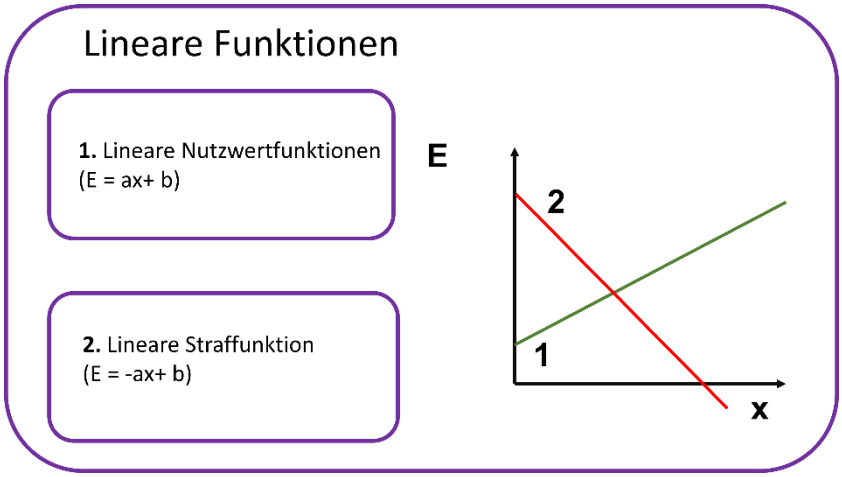


Abbildung 9 Nutzwertfunktionen (Rinza and Schmitz 1992)

> Quantifizierbare Kriterien sind in Rahmen dieser Bachelorarbeit, für die Beantwortung der Forschungsfrage, beispielsweise die Anzahl der Nodes in einem Cluster, je verwaltete Kubernetes Service.

Wertetabellen werden hingegen zu Wertefunktionen verwendet, wenn Zielkriterien beziehungsweise deren Qualität, nur durch eine Beschreibung bestimmt werden kann. Wertetabellen werden hauptsächlich bei nicht quantifizierbaren Kriterien verwendet.

(Rinza and Schmitz 1992)

Eine weitere Rahmenbedingung, bei Erstellung von Wertetabellen und Wertefunktionen ist, dass Erfüllungsgrade wie in Abbildung 10 dargestellt werden.

Note	Erfüllt die Anforderungen
6	Sehr gut
5	Gut
4	Befriedigend
3	Ausreichend
2	Mangelhaft
1	Ungenügend

Mittelwert: 3,5

Note	Erfüllt die Anforderungen
6	Sehr gut
5	Gut
4	Befriedigend
3	Ausreichend
2	Mangelhaft
1	Ungenügend
0	Nicht erfüllt

Mittelwert: 3,0

Note	Erfüllt die Anforderungen
6	Sehr gut
3	Ausreichend

Mittelwert: 4,5

Abbildung 10 Erfüllungsgrade (Rinza and Schmitz 1992)

Üblich ist bei einer Nutzwertanalyse, die Verwendung von Kardinalskalierung, wie in der Abbildung 10, links abgebildet.

Weitere Bedingungen sind, dass bei einer Ja-Nein Entscheidung eines Zielkriteriums aus dem Zielsystem, die Erfüllungsgrade mit drei und sechs bewertet werden sollen. Wobei drei, den niedrigsten und sechs den höchsten Erfüllungsgrad bei der Bewertung repräsentieren soll.

(Rinza and Schmitz 1992)

<u>Bewertung der Alternativen:</u>

Im vierten Schritt der Nutzwertanalyse befinden sich keine aufregenden Rahmenbedingungen bei der Abarbeitung des Teilschritts. Hier sollen lediglich die Bewertungen der Alternativen stattfinden. Bevor es zu einer Bewertung kommt, muss offengelegt werden, welches Zielkriterium per Wertetabelle und per Wertefunktion ermittelt werden soll.

(Rinza and Schmitz 1992)

<u>Berechnung der Nutzwerte:</u>

In diesem Teilschritt werden die Nutzwertbeiträge (N_i) für jedes Zielkriterium kalkuliert. Für die Berechnung werden die Knotengewichte (w_i) und die Erfüllungsgrade (E_i) je Kriterium nach der Gleichung miteinander multipliziert und zusätzlich werden anschließend einzelne Nutzwertbeiträge zum Gesamtnutzwert addiert. Anschließend findet die Reihung nach der Rangfolge statt.

Empfindlichkeitsanalyse (optional):

Der optionaler Schritt der Nutzwertanalyse ist hauptsächlich für Beruhigung von Meinungsunterschieden empfohlen. Hier sollen jegliche Schritte, wie die Zielsystem Zusammensetzung, Gewichtung, Wertetabellen und Wertefunktionen inklusive ihre Definierung, Berechnung der Nutzwerte neudurchdacht werden. Die Empfindlichkeitsanalyse bietet keine genauen Angaben und bedient sich nur durch Abschätzung in drei Stufen. Die Vorgehensweise hierbei lässt sich definieren durch, optimistische, wahrscheinliche und pessimistische Werte.

(Rinza and Schmitz 1992)

Beurteilung und Darstellung der Ergebnisse:

Abschließend einer Nutzwertanalyse sollen die Arbeitsergebnisse, die für die Auswahlentscheidung von Bedeutung sind, dargestellt werden. Bei einer Nutzwertanalyse sind Darstellungsformen wie ein Balkendiagramm, sternförmiger Matrix, kreisförmige Darstellung oder auch eine tabellarische Form der Darstellung, die wohlbekanntesten Darstellungsarten.

(Rinza and Schmitz 1992)

Die Methoden und einzelne Schritte der Nutzwertanalyse, welche für die Beantwortung der Forschungsfrage in dieser Bachelorarbeit relevant sind, werden wie gefolgt aus dem obigen Kapitel 4.3.1 Rahmenbedingungen für die sieben Schritte der Nutzwertanalyse, adaptiert und im **Kapitel 5** per einer Nutzwertanalyse dargestellt.

4.3 Beschreibung der Vorgehensweise

Die Definition von Prototyping laut IT-Service Network ist, das Erstellen und Testen von IT-Modellen, bevor es noch zu der finalen Endversion gelangt und veröffentlich wird. Die Idee dahinter ist es, am Ende zu überprüfen, inwiefern beispielsweise eine Software, auf die Bedürfnisse der Nutzer: innen nahe liegt. Anhand des Prototyps einer Software oder einer Applikation, kann das Produkt bereits in seiner Entwicklungsphase überprüft und somit in Zukunft bestmöglich für den Markt zur Vermarktung vorbereitet werden.

(IT-Service Network 2023)

Wie im vorherigen Absatz angerissen, ist die Vorgehensweise in dieser Bachelorarbeit, dass schrittweise annähern per Prototyping an das fertige Endprodukt. Das heißt, es wird in jeweiligen Hyperscaler ein Prototyp der Cloud Umgebung erstellt, in dieser werden die verwalteten Kubernetes Service implementiert. In die verwalteten Kubernetes Service, wird ein GitHub Repository bereitgestellt.

> „Ein Repository in GitHub, beinhaltet jegliche Dateien eines Projekts, welches bei Bedarf für öffentliche Nutzung bereitgestellt ist. Dort kann der Quellcode oder auch der Revisionsverlauf jeder Datei nachgeschaut, und falls sie veröffentlich sind, auch verwendet werden.“
>
> (GitHub 2023)

Das Github Repository beinhaltet hauptsächlich einen Dockerfile, in welchem die Java-Quelldatei, mit der Information „Hello World“, vorbereitet ist. Zusätzlich ist diese Java-Quelldatei mit einer Web Service Applikation in Java – Programmiersprache geschrieben, welche bei jeder Aktualisierung der Webapplikation, hinter dem Wort „Hello World“, weitere signifikante Buchstaben adaptiert, die sich willkürlich immer auf drei verschiedene Versionen ständig abwechseln und ändern.

Sobald das GitHub Repository implementiert wird, in den jeweiligen Kubernetes Cluster, sind die verwalteten Kubernertes Services auch aktiv und können auf die, für die Bachelorarbeit definierten Funktionen, überprüft und bemessen werden.

Angefangen von der Cloud Hauptmenü, bis hin zu den einzelnen Bereichen der ausgewählten sieben Funktionen, werden die jeglichen Funktionen auf ihren einbringenden Nutzen überprüft und per Abbildung dokumentiert. In der Vorgehensweise per Prototyp, wird weiters noch ein Timer bereitgestellt, um abzumessen wie lange eine Cluster Bereitstellung gedauert hat.

Außerdem werden auch jegliche Informationen, bei Bedarf, aus der Dokumentation des jeweiligen Cloud Providers, übernommen und hinzugefügt. Diese Informationen dienen zur Ergänzung der Abbildungen, aufgrund ihrer Wichtigkeit, für die Bewertung einer qualitativen Nutzwertanalyse im nächsten Kapitel.

Anhand der Rahmenbedingungen einer Nutzwertanalyse aus dem **Kapitel 4.2 Beschreibung der Methode,** konnten folgende Zielkriterien, wie in der Tabelle 2, eruiert werden:

Tabelle 2 Zielkriterien für die Nutzwertanalyse

1. CLI Unterstützung	
1.1 Zusatzinstallation	Wertetabelle
1.2 Yaml-Unterstützung	Wertetabelle
1.3 Lokale Installation	Wertetabelle
1.4 Einfache Bedienung	Wertetabelle
2. RBAC	
2.1 Manuelle Einstellung	Wertetabelle
2.2 Verfügbarkeit von Templates	Wertetabelle
2.3 Master Global Access	Wertetabelle
3. Monitoring	
3.1 Thirdparty Möglichkeit	Wertetabelle
3.2 Dashboard Planung	Wertetabelle
3.3 Cluster Monitoring	Wertetabelle
3.4 Pod Monitoring	Wertetabelle
4. Preisgestaltung	
4.1 Kosten bei Zusatzfeatures	Wertetabelle
4.2 differenzierte Preismodellierung	Wertetabelle
5. K8s Versionsunterstützung	
5.1 Downgrade Option	Wertetabelle
5.2 Upgrade Option	Wertetabelle
5.3 automatisches Update	Wertetabelle
6. Knotenintegrität u. Überwachung	
6.1 Automatisierte Überwachung	Wertetabelle
6.2 Automatische Knoten Reparatur	Wertetabelle
6.3 Vertikale Pod Skalierung	Wertetabelle
6.4 Pod Limit per Node	**Wertefunktion**

7. Spawn Cluster Zeit	
7.1 Flexibilität in der Clusterbildung	Wertetabelle
7.2 Node Limit je Cluster	**Wertefunktion**
7.3 Betriebssysteme der Nodes	Wertetabelle
7.4 TPU Nodes	Wertetabelle
7.5 Node Location	Wertetabelle

Da drei Alternativen (AKS, GKE und EKS), miteinander verglichen werden, müssen mindestens 25 Zielkriterien definiert werden, damit die Nutzwertanalyse, welche erstellt wird, überhaupt als eine mittelmäßige Entscheidungstreffung gelten darf, die qualitativ ist.

In ihrer Gesamtheit ergeben sich dadurch zwei Zielebenen, die sieben Hauptkriterien und die 25 Subkriterien. Die Ergebnisse der Zielkriterien, werden wie in der Tabelle 2 dargestellt, entweder per Wertetabelle oder per Wertefunktion wiedergeben. Grundsätzlich werden die jeweiligen Sub-Ziele per Ja-Nein Entscheidung bewertet, da dies für eine grobe Bewertung ausreichend ist.

Somit wird im ersten Zielkriterium, CLI Unterstützung, untersucht, ob eine Zusatzinstallation notwendig ist; eine YAML – Datei Bearbeitung und eine lokale Installation auf den PC möglich ist. Außerdem wird anhand des Prototyps bewertet, ob diese auch eine einfache Bedienung gewährleistet.

Im zweiten Zielkriterium, RBAC, wird in der Cloud Umgebung nachgesehen, ob eine manuelle Restriktion eingestellt werden darf und ob vordefinierte Restriktionen vorhanden sind. Im Anschluss die Überprüfung auf Master Global Access. Im Allgemeinen ist die Master Global Access nichts anderes, wie ein Accountuser, dass wie eine Admin-Rolle, Zugang auf alle Ressourcen hat und bei Wunsch, auch bearbeiten/ erstellen und löschen darf, ohne die Erlaubnis oder Genehmigung der Cloud Inhaber: innen.

Zielkriterium drei ist definiert mit Monitoring. Hier wird konstatiert, ob eine Anbindung zu Drittanbieter erstellt werden darf, ob das Dashboard, eigenständig planbar ist und ob in Monitoring, sowohl die Kubernetes Cluster und Pods separat, per jeweilige Metriken überwacht werden können.

Das vierte Zielkriterium, die Preisgestaltung, musste den Rahmenbedingungen der Nutzwertanalyse angepasst werden. Da bei einer Nutzwertanalyse keine

Kostenkriterien überprüft werden dürfen, wird bei der Ja-Nein Entscheidung analysiert, ob es Kostenfallen gäbe, bei zusätzlich, angebotenen Features vom jeweiligen verwalteten Kubernetes Services und ob differenzierte Preismodellierungen vorhanden sind.

In der K8s Versionsunterstützung, beziehungsweise in der Kubernetes Versionsunterstützung, wird ausfindig gemacht, ob Downgrades (=Installation von älteren Release Versionen) oder Upgrades (=Aktualisierung der Release Versionen) unterstützt werden und ob Updates im jeweiligen Service automatisch durchführbar sind.

Im vorletzten Zielkriterium, Knotenintegrität und Überwachung, wird angemerkt, ob die Überwachung der Knoten (Nodes) automatisch erfolgt; ob eine automatische Node-Reparatur ermöglicht wird; ob die Pods vertikal skaliert werden können und wie viele Pods in einer Node maximal bereitgestellt werden können.

Im letzten Zielkriterium, Spawn Cluster Zeit, wird dargelegt, wie flexibel die verwalteten Kubernetes Services in der Clusterbildung ist. Des Weiteren wird überprüft, wie viele Nodes in einem Cluster maximal bereitgestellt werden können. Außerdem wird nachgesehen, welche Betriebssysteme angeboten sind für die Nodes. Daraufhin folgt die Überprüfung auf die TPU Nodes und die Durchführbarkeit einer Node Lokalisierung.

TPU Nodes, auch bekannt als Tensor Processing Unit. TPU Nodes sind hauptsächlich Hardwarebeschleuniger, die von Google für Machine Learning Workloads entwickelt wurden.

(Google Cloud 2023)

Themen die sich mit Machine Learning beschäftigen sind im Grunde genommen nicht das eigentliche Augenmerk im Prototyping oder dieser Bachelorarbeit, aus diesem Grund, wird im Rahmen der Bachelorarbeit nur überprüft, ob die angeführten Hyperscaler, ein TPU Node für die User zur Verfügung stellen, die des Google Cloud Plattform gleichen.

Vorab soll erwähnt werden, dass grundsätzlich in allen drei Hyperscaler, eine Trail- Cloud Umgebung erstellt werden kann mit einem bestimmten Guthaben, dass für frei zur Verfügung steht, um die Cloud und Cloud Funktionen kennenzulernen, doch im Zuge der Bachelorarbeit, wird die Erstellung der Cloud Umgebung übersprungen. Bei der Erstellung der Cloud Umgebung, muss der Inhaber: in, relevante Informationen über sich selbst eintragen, um auch identifiziert werden zu können. Diese sind allgemein gefasst, Name; Anschrift und Zahlungsmethode.

Der verwaltete Kubernetes Service kann sowohl per Icons aus der GUI, aber auch per CLI Befehle erstellt werden. Für die Bachelorarbeit wurde die Verwendung der CLI Kommandobefehle vorgesehen.

4.4 Prototyping in Azure Kubernetes Service (AKS)

Aus den Anhaltspunkten des Vorkapitels **4.3 Beschreibung der Vorgehensweise,** wird nun der AKS Cluster bereitgestellt. Die Anmeldung in die Azure Landschaft folgt unter https://www.portal.azure.com/#home

- Command Line Interface (CLI)

Der CLI ist auf der Azure Cloud Landschaft, wie auf Abbildung 11, rechts oben dargestellt zu finden. Der eigentliche CLI, den ich hierbei überprüfen möchte, ist der Azure CLI, welche während der Beschäftigung in der Azure Landschaft, Usern, die Ressourcen gewährt.

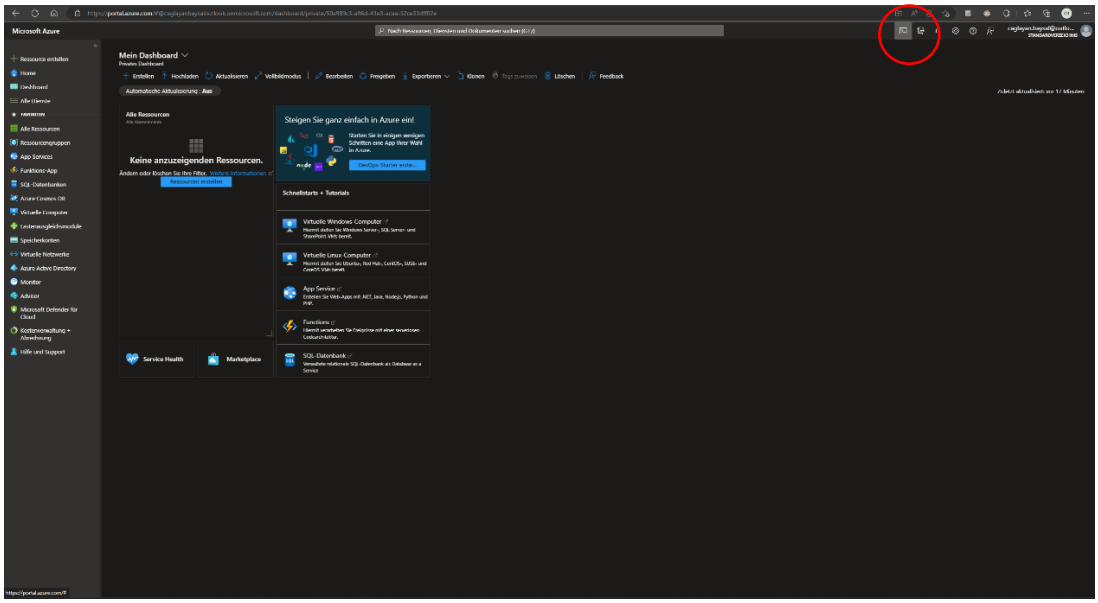


Abbildung 11 Cloud Landschaft Azure

Mit dem Befehl **az version**, kann festgestellt werden, welche Version der Azure CLI die Azure Landschaft besitzt. Für die Erstellung und Bearbeitung einer AKS, wird der kubectl benötigt. Auch dieser lässt sich einfach per Versionsabfrage feststellen. Der Befehl dafür ist **kubectl version.** Sollte der kubectl schon vordefiniert sein per Azure Cloud Umgebung und erreichbar in Azure CLI sein,

wird wie in Abbildung 12, die derzeitige Version vorgestellt. Im rot markierten Bereich der Abbildung 12, kann der Azure Cloud Shell gesehen, sobald dieser gestartet wird. In Abbildung 13, wird das Ergebnis der Abfrage etwas größer visualisiert.

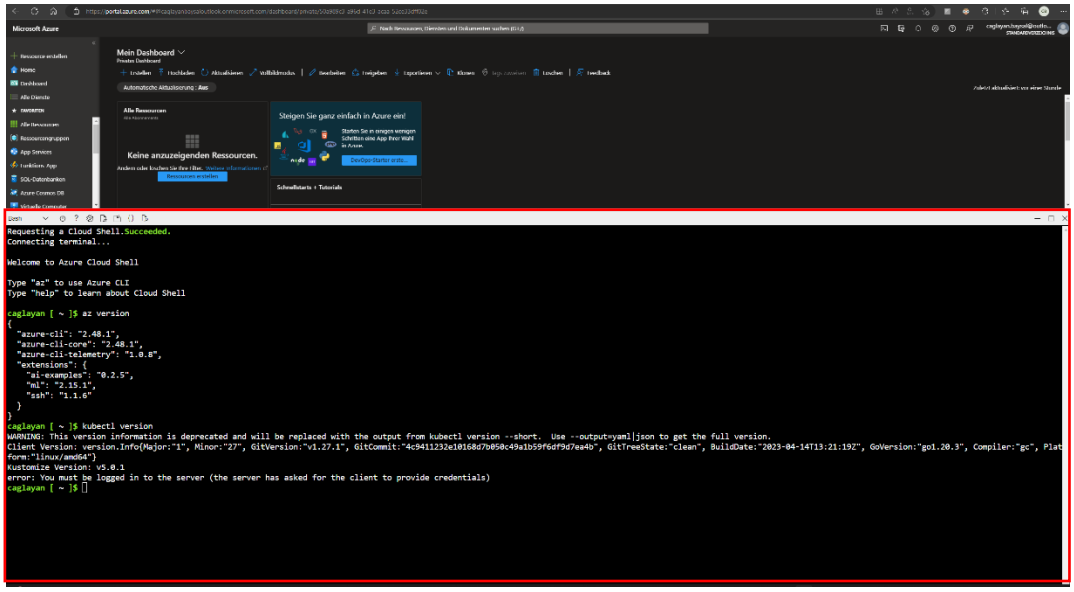


Abbildung 12 Azure Cloud Shell

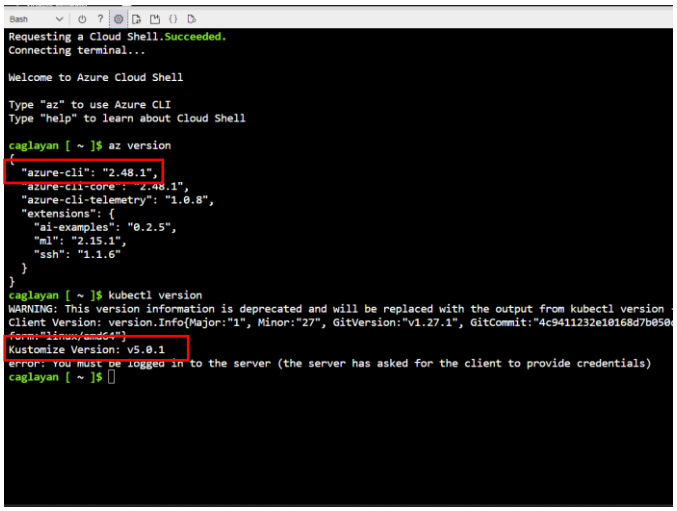


Abbildung 13 CLI Abfragen (az version und kubectl version)

Für die Erstellung jeglicher Funktionen in Cloud, wie zum Beispiel bei der Erstellung einer Azure Kubernetes Service, muss eine Ressourcengruppe verfasst werden. Für die Aufrechterhaltung der SLAs, wie beispielsweise die Hoheverfügbarkeit der Cloud Applikationen, muss auch beim Verfassen einer Ressourcengruppe auch ein Lokal festgelegt werden. In meinem Fall ist dies

Westeuropa, weil ich wohnhaft in Österreich bin. Weiters müssen je User: innen Zugriffe gewährt werden, damit diese auch die Erlaubnis haben, AKS zu bearbeiten. Die RBACs in Azure sind soweit ausgedehnt, dass nicht nur auf User Restriktionen eingestellt werden kann, sondern auch auf die Service.
Die CLI Befehle lauten:

- az group create --name kubernetes-resource-group --location westeurope
- az ad sp create-for-rbac --skip-assignment --name kubernetes-cluster-service-principal

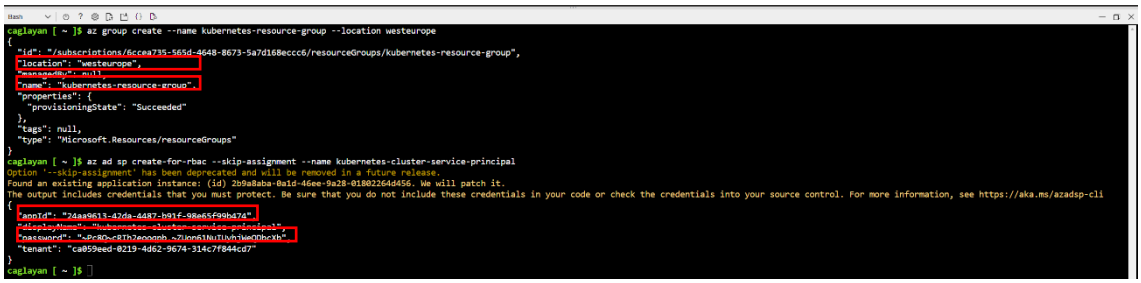


Abbildung 14 CLI Befehle für die Erstellung der Ressourcengruppe, Lokalisierung, RBAC für Serviceprinzipien

Durch die Eingabe des zweiten Befehls, die Erstellung einer Restriktion je Serviceprinzip, habe ich eine appID und ein Passwort erhalten. Mithilfe der automatisch generierten Informationen, kann per weiteres CLI Befehl, der AKS Cluster mit jeweiligen Parametern erstellt werden.

Der nächste CLI Befehl lautet daher:

- az aks create --name bachelor-cluster --node-count 4 --enable-addons monitoring --resource-group kubernetes-resource-group --vm-set-type VirtualMachineScaleSets --load-balancer-sku standard --enable-cluster-autoscaler --min-count 1 --max-count 7 --generate-ssh-keys --service-principal 24aa9613-42da-4487-b91f-98e65f99b474 --client-secret Qjk8Q~PynmDST5nwfUHJdQXdQiU_XMkLR1kpoc8H

In den Parametern wird definiert, wie der AKS Cluster heißen soll, im Prototyp entschied ich mich für den Cluster Namen „Bachelor-Cluster“. Weiters sollen vier Nodes in diesem Cluster erstellt werden. Anhand des – enable-addons, habe ich dem Cluster das Monitoring gewährt. Im weiteren Parametern wurde definiert, zu welcher Ressourcegruppe und sowie auch welche Einstellungen die virtuelle Maschine haben darf.

In diesem Befehl wird die VM nicht konfiguriert, sie soll sich automatisch anpassen und bei Bedarf skalieren. Außerdem ist noch zu erwähnen, dass anhand des erstellen Passworts und appID, die jeweiligen Parameter sich eindeutig zuweisen lassen, durch die Eingabe dieser Informationen im Parameter.

Nun kann der GitHub Repository heruntergeladen werden. Der Link dafür lautet: https://github.com/mrbysl1412/testk8s . In Abbildung 15, kann per rot markierten Rechteck gesehen werden, welche Dateien sich in diesem Repository befinden und wie die vorbereitete Datei, als ZIP-Datei heruntergeladen werden kann.

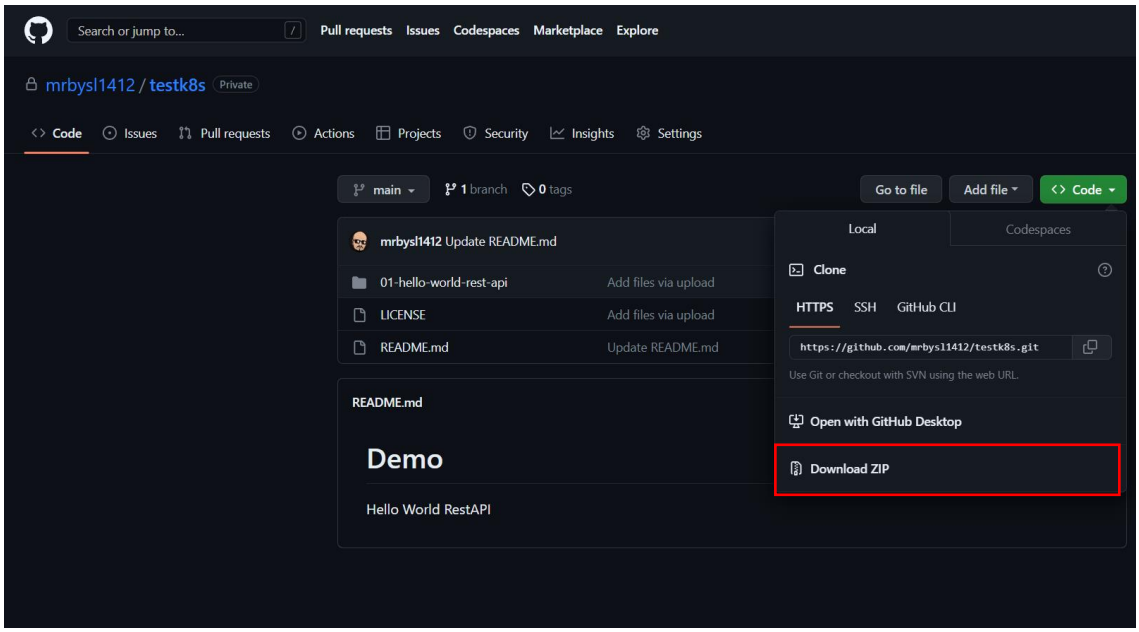


Abbildung 15 Inhalte der GitHub Repository

Dieser Abschnitt der Bereitstellung wird kurzgefasst, da dieser Vorgang, weitere Erklärungen beinhaltet, welche nicht im Rahmen der Bachelorarbeit vorgesehen sind.

In der heruntergeladenen ZIP-Datei, welche zum lokalen Computer zu extrahieren ist, befindet sich ein Ordner namens „01 hello-world rest api“. In dieser befindet sich ein Docker Container Datei namens „Dockerfile“, wo die eigentliche Webapplikation verschachtelt sich befindet, mit jeglichen Bibliotheken, die relevant für die Web- Applikation sind. Der Dockerfile ist verifiziert mit einer bestimmten Docker - Image- Name. Der Name des Docker-Images ist „in28min“.

Mit dem ersten Befehl in Abbildung 16, stellen wir fest, wo der Konfigurationsdatei des Bachelor-Clusters zusammengeführt ist. Mit dem nächsten Befehl in der Azure CLI, wie in der Abbildung 16 dargestellt, wird die Webapplikation für den AKS Cluster für die Bereitstellung vorbereitet. Mit dem dritten Befehl in Abbildung 16, wird diese Bereitstellung vorgeführt. Der Webapplikation -Service „hello-world-rest-api“ ist nun mithilfe der externen IP-Adresse und dem Portzugang 8080 erreichbar.

Die verwendeten CLI Befehle in Abbildung 16, sind:

- az aks get-credentials --resource-group kubernetes-resource-group --name bachelor-cluster
- kubectl create deployment hello-world-rest-api --image=in28min/hello-world-rest-api:0.0.1.RELEASE
- kubectl expose deployment hello-world-rest-api --type=LoadBalancer --port=8080

```
caglayan [ ~ ]$ az aks get-credentials --resource-group kubernetes-resource-group --name bachelor-cluster
Merged "bachelor-cluster" as current context in /home/caglayan/.kube/config
caglayan [ ~ ]$ kubectl create deployment hello-world-rest-api --image=in28min/hello-world-rest-api:0.0.1.RELEASE
deployment.apps/hello-world-rest-api created
caglayan [ ~ ]$ kubectl expose deployment hello-world-rest-api --type=LoadBalancer --port=8080
service/hello-world-rest-api exposed
caglayan [ ~ ]$
```

Abbildung 16 Bereitstellung der hello-world-rest-api

Die externe IP Adresse der Service kann mit dem Befehl überprüft werden:

- Kubectl get service –watch

Wie in Abbildung 17 nachgewiesen, lautet die externe IP-Adresse meiner hello-world-rest-api Web Services „20.23.201.195“

```
caglayan [ ~ ]$ kubectl get service --watch
NAME                 TYPE           CLUSTER-IP     EXTERNAL-IP     PORT(S)          AGE
hello-world-rest-api LoadBalancer   10.0.144.189   20.23.201.195   8080:30775/TCP   13m
kubernetes           ClusterIP      10.0.0.1       <none>          443/TCP          60m
```

Abbildung 17 externe IP-Adresse des Web-Service

Durch die Eingabe von diesem Zugriffslink 20.23.201.195:8080/hello-world, in einem leeren Webbrowser, kann der Web Service aufgerufen werden, wie in Abbildung 18, wiedergeben.

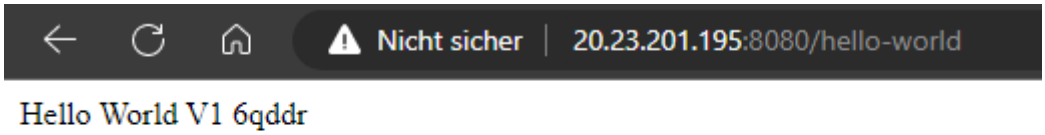


Abbildung 18 Web Service Aufruf

Die Duplizierung der Bereitstellung erfolgt unter dem Befehl:

- kubectl scale deployment hello-world-rest-api –replicas=3

Mit dem Befehl:

- kubectl get pods

Kann überprüft werden, ob die Duplizierung erfolgreich gewesen ist. Es sollte drei Pods auflisten, die einen aktiven Status haben.

Zu den weiteren Sub-Zielen, des ersten Zielkriteriums, durfte in erster Linie konstatiert werden, dass eine Zusatzinstallation für Azure CLI nicht benötigt wurde. Außerdem ist die Yaml–Unterstützung gewährt. Mit dem Kommandobefehl **vim pod.yaml,** konnte ein Yaml-File erstellt werden. Weiters ist auch eine lokale Installation des Azure CLIs unterstützt. Der Azure CLI kann per einer Installer-Datei heruntergeladen und einfach installiert werden.

(Microsoft Learn 2023)

Während der Vorbereitung des Prototyps, bin ich häufig auf Fehler zu gestoßen, wo der Azure CLI einfach nicht mit den Befehlen umgehen konnte, aus diesem Grund finde ich, dass die Verwendung von Azure CLI nicht einfach gewesen ist.

- rollenbasierte Zugriffssteuerung (RBAC)

Anhand der Erfahrung, welche durch die Erstellung des Prototyps gesammelt werden konnten, können RBACs User spezifisch generiert werden. Doch wie in Abbildung 14 reflektiert, ist es in Azure ebenso möglich Restriktionen per Azure Service Prinzipien zu generieren. Das heißt, eine manuelle Einstellung des RBACs wird in AKS unterstützt. Außerdem sind auch vordefinierte Restriktionsrollen vorhanden, doch ein Master Global Access wird hier nicht unterstützt, diese müsste durch jegliche Anbindungen von Genehmigungen und Rollen selbst kreiert werden.

(Microsoft Azure 2023)

- Monitoring

In Azure wird der Monitoring auch über andere Drittanbieter Tools und Applikationen gewährt, wie zum Beispiel Prometheus und Grafana. Auch kann der Monitoring Dashboard selbst kreiert werden. Anhand Azure Monitor; Application Insights und Container Insights ist es in AKS möglich, sowohl Pods als auch Cluster-Metriken anhand des Monitoring Tools zu überwachen.

(Microsoft Learn 2023)

- Preisgestaltung

Bezüglich der Preisgestaltung in AKS, konnte festgestellt werden, dass Kosten bei Zusatzfeatures anfallen können, wie zum Beispiel, dass für das Service AKS keine Kosten anfallen, aber für die dafür bereitgestellten virtuellen Maschinen. Eine differenzierte Preismodellierung ist je nach Tarif und SLA Vereinbarung nicht ausgeschlossen.

(Microsoft Azure 2023)

- Kubernetes Versionsunterstützung

Die Versionsunterstützung in AKS, ist in der Abbildung 19 veranschaulicht.

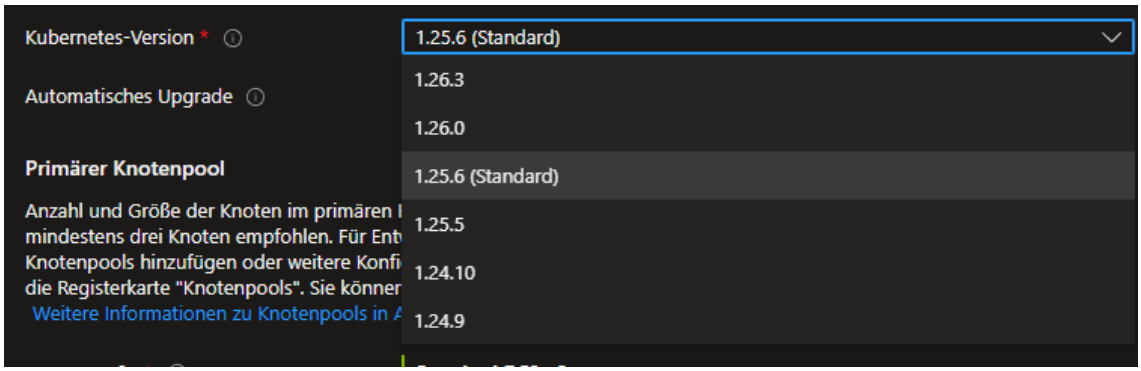


Abbildung 19 Versionsunterstützung K8s

Ein automatisches Updaten der Kubernetes Version wird in diversen Möglichkeiten angeboten. Dadurch kann auch festgestellt werden, dass ein Upgrade der Kubernetes Version möglich ist. Ein direkter Downgrade ist nicht möglich. Dafür müsste eine Widerherstellungsdatei des jeweiligen Clusters erstellt werden und so könnte bei der neuen Erstellung des Clusters, eine ältere Version ausgewählt werden.

(Microsoft Azure 2023)

- Überwachung der Knotenintegrität

Zum Zielkriterium sechs, konnte anhand der Dokumentation herausgefunden, dass eine Überwachung des Nodes per dezidierte Monitoring Metriken möglich ist und eine automatische Node-Reparatur eingestellt werden kann. Mit Stand Mai 2023 konnte aber nicht überprüft werden, wie viele Pods in einer AKS Cluster bereitgestellt werden und eine vertikale Pod-Skalierung, welche automatisch funktionieren soll, ist nicht unterstützt.

(Microsoft Learn 2023)

- Spawn Cluster Zeit

Wie im oben angeführt, bei der Erstellung des Clusters per Parameter, hat die Bereitstellung des Clusters wie in Abbildung 20 dargelegt,

ungefähr acht Minuten gedauert.

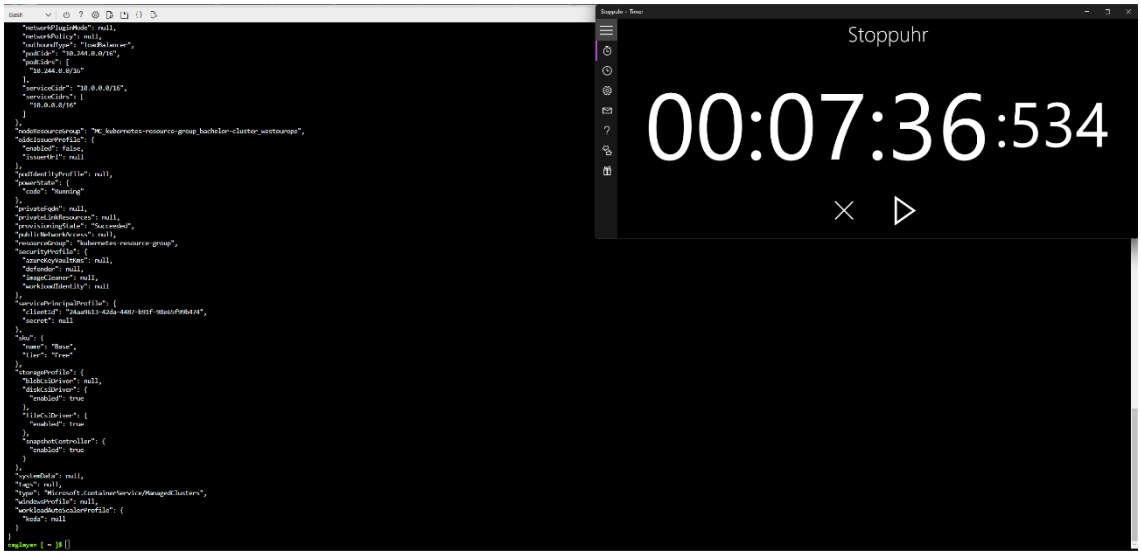


Abbildung 20 Spawn Cluster Zeit

Die im Hintergrund während des Bereitstellungsprozesses erstellte Ressourcen sind auf Abbildung 21, abgebildet.

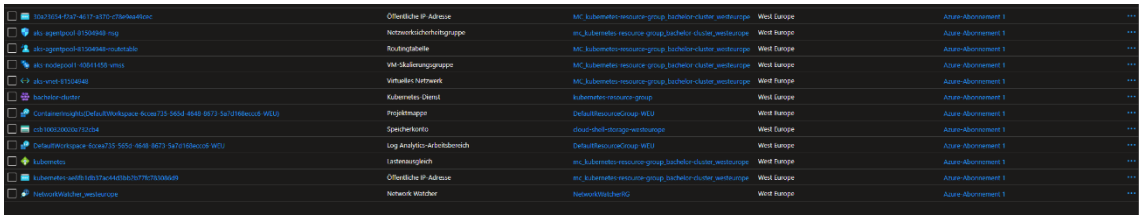


Abbildung 21 erstelle Ressourcen während der Bereitstellung des AKS Clusters

Zu den weiteren Sub-Zielkriterien des Zielkriteriums konnte fündig gemacht werden, dass AKS sowohl die Clustererstellung der GUI als auch die Clusterbildung per Cloud Shell unterstützt. Die maximale Anzahl der Nodes per Cluster beträgt 1000 und die maximale Anzahl der Pods in einer Node sind 250. Ein weiteres Sub-Ziele, in diesem Zielkriterium ist die Auswahl der Betriebssysteme der Nodes gewesen. Nodes in AKS könnten per Ubuntu oder Windows Servern definiert werden.

Die Recherche in der Dokumentation zur TPU Nodes Verfügbarkeit in AKS ist zwar nicht erfolgreich gewesen, doch Microsoft bietet eine Alternative, die sich stets weiterentwickelt. Das sogenannte FBGA (Fine-Pitch Ball Grid Array). Ein System on a Chip designte Lösung statt TPU Node. Zum Schluss noch ein weiterer erwähnenswerter Punkt dass in AKS, mithilfe der Node Pool Konfiguration, die Nodes lokalisierbar konfiguriert werden können.
(Microsoft Azure 2023)

4.5 Prototyping in Elastic Kubernetes Service (EKS)

Der zweite Prototyp, welcher erstellt wird ist der Elastic Kubernetes Service. Auch hier wird die Erstellung der AWS Cloud Umgebung übersprungen und fortgesetzt mit dem CLI. Doch es muss davor erwähnt werden, dass bei der Erstellung der Cloud Landschaft in Amazon Web Service ein Root-Account für das AWS Konto und ein Identity Access Management User erstellt werden muss. Der IAM – User: in hat die Erlaubnis, ihrer Rolle eingetragene und erlaubte Ressourcen zu bearbeiten.

Die Cloud Umgebung in AWS ist unter https://aws.amazon.com/de/ zu erreichen.

Auf der Cloud Konsole von AWS kann im Suchfeld „IAM“ eingegeben, um das Service aufzurufen und dann die Benutzergruppe ausgewählt erstellt werden. Für das weitere Vorgehen wurde nun eine Benutzergruppe namens „Devs“ erstellt und diesem die Adminrechte gewährt. Der User „devcrb“ wurde dann dieser Benutzergruppe hinzugefügt. Nun muss als IAM – User angemeldet werden, um Bearbeitungen in der Cloud Landschaft durchführen zu können.

- Command Line Interface (CLI)

Zum starten des AWS CloudShells kann rechts oben markierte Icon, wie in Abbildung 22, angeklickt werden, so eröffnet sich im unteren Bereich des Webbrowsers der AWS CloudShell. Der eksctl ist der eigentliche AWS CLI, dass uns die Erstellung des EKS Clusters gewährt. Nach einer kurzen Überprüfung, durfte festgestellt werden, dass kein eksctl vorhanden ist.

Mit einem Linux Befehl im AWS CloudShell, wird der eksctl heruntergeladen.

Der Linux Befehl:

- Curl--silent –location "https://github.com/weaveworks/eksctl/releases/latest/download/eksctl_$(uname -s)_amd64.tar.gz" | tar xz -C /tmp

Mit dem zweiten Linux Befehl, wird die heruntergeladene Tar-Datei in den Bin-Folder des CloudShells verschoben und automatisch installiert.

- sudo mv /tmp/eksctl /usr/local/bin

Wie in Abbildung 22 dargestellt, kann per **eksctl version** Abfrage, die Installation überprüft und die Version abgefragt werden.

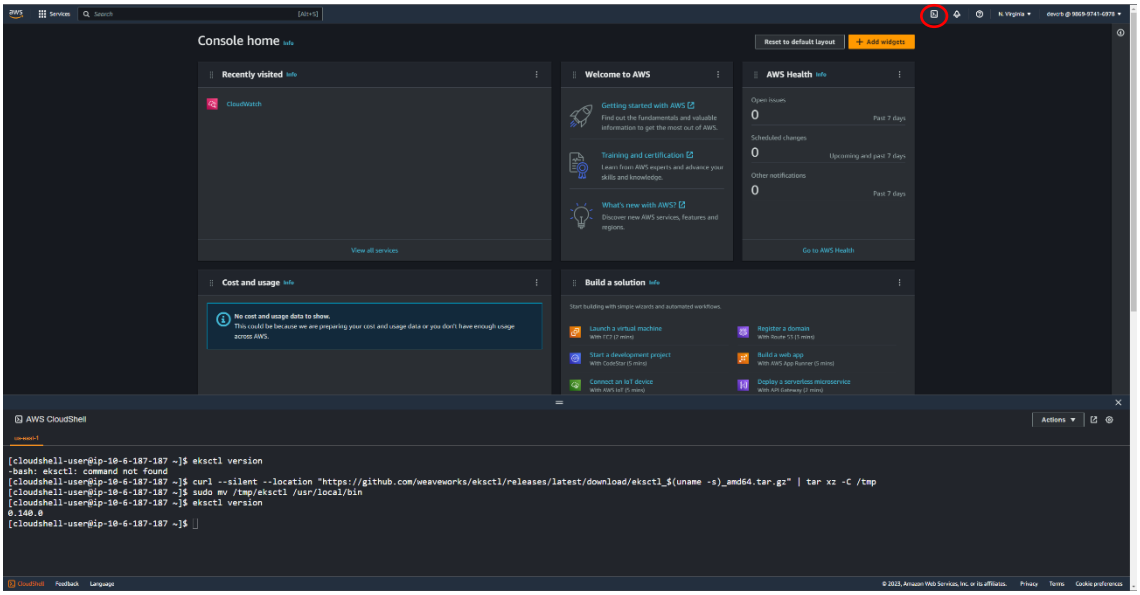


Abbildung 22 AWS CloudShell und Installation des Eksctl

Der nächste Schritt für die Erstellung eines EKS Clusters ist der eksctl Befehl.

- eksctl create cluster --name bachelor-cluster --nodegroup-name bachelor-cluster-node-group --node-type t2.medium --nodes 3 --nodes-min 3 --nodes-max 7 --managed --asg-access

Mithilfe der Parametern wurde definiert, dass unser EKS Cluster „bachelor-cluster“ und unsere Ressourcengruppe „bachelor-cluster-node-group“ benannt wird. Aufgrund des erstellen Free Tier-AWS Accounts, ist die virtuelle Maschine in der Node, der sogenannte in AWS EC2 Instanz Typ, t2.medium. Dieser bietet zwei vCPUs und vier RAM. In dieser wurden drei Nodes erstellt, für unsere Webapplikation und eine automatische Node – Skalierung von Minimum Nodes an drei und mit einer maximalen Node-Anzahl von sieben. In Abbildung 23 und 24, wird dargestellt wie lange die Erstellung eines EKS Clusters gedauert hat. Die Bereitstellung des Clusters beträgt 17 Minuten.

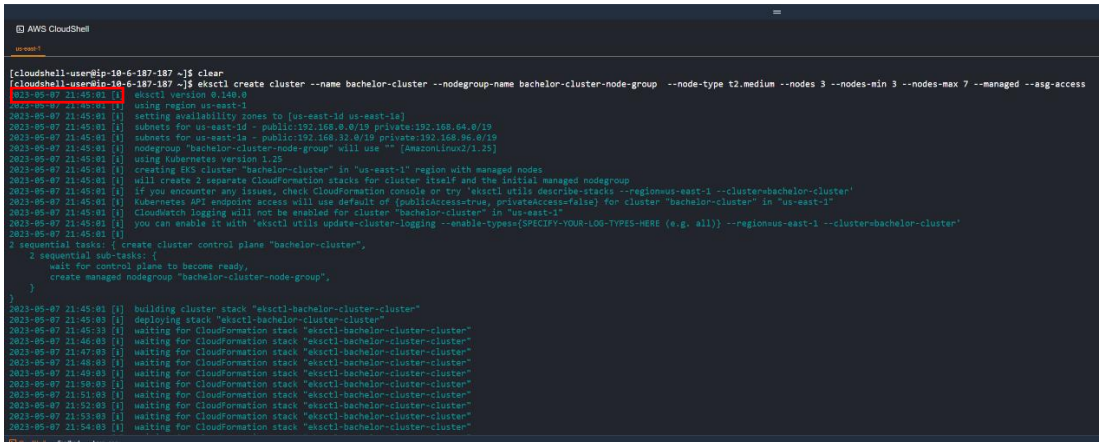


Abbildung 23 Spawn Cluster Startzeit

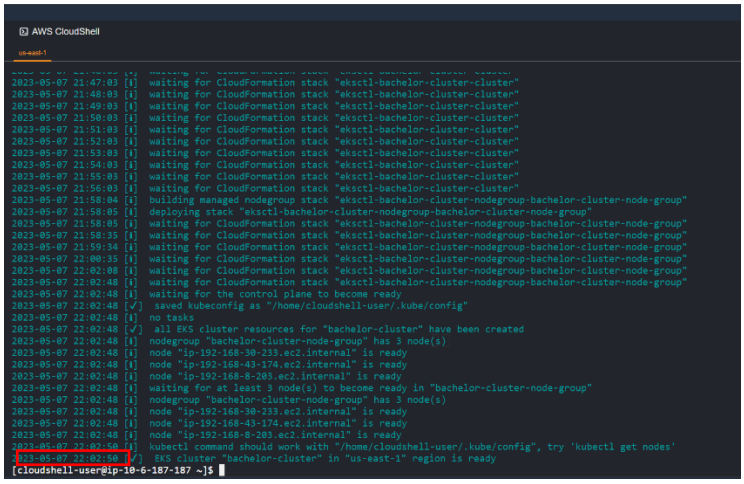


Abbildung 24 Spawn Cluster Endzeit

Um nachzusehen, ob die drei Nodes auch bereitgestellt worden sind, kann in AWS CloudShell folgender Befehl ausgelöst werden:

- kubectl get nodes

Dieser zeigt uns, wie in Abbildung 25 ersichtlich, Informationen über die implementierten Nodes bekannt.

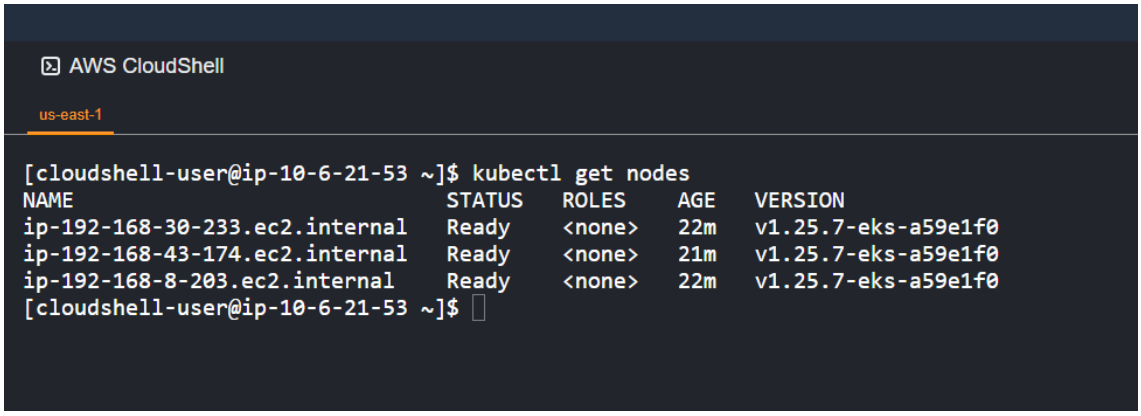


Abbildung 25 Node Informationen EKS Cluster

Wie im Prototyping von AKS können nun aus der GitHub Repository, heruntergeladene Dateien vorbereitet werden auf die Bereitstellung und danach ausgeführt werden zum Exponieren. Die Befehle sind wieder dieselben:

- kubectl create deployment hello-world-rest-api --image=in28min/hello-world-rest-api:0.0.1.RELEASE
- kubectl expose deployment hello-world-rest-api --type=LoadBalancer --port=8080

Zusätzlich kann als Zwischenschritt das Deployment aus dem ersten Befehl überprüft werden per:

- kubectl get deployment

In Abbildung 26 können die Schritte nachgefolgt werden. Zuerst wird vorbereitet, dann überprüft dann bereitgestellt. Im rot markierten Rechteck, kann gesehen werden, dass der Web Service zwar für die Bereitstellung vorbereitet worden ist, aber noch nicht ausgeführt wurde.

Abbildung 26 Deployment und Expose vom Web Service

Der Web Service ist wieder unter der externen IP Adresse mit der Ergänzung des Ports erreichbar. Die Informationen zu der externen IP-Adresse kann, wie in Abbildung 27, anhand des Befehls:

- Kubectl get svc

erhalten werden. (SVC) ist die Abkürzung in der Kubernetes-Sprache für Services.

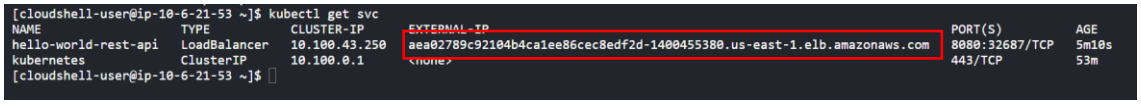


Abbildung 27 Service Informationen

Der rot markierte Rechteck in Abbildung 27, zeigt die externe IP-Adresse des Web Services an. In Abbildung 28, wird visualisiert, wie in den neuen Browser die externe IP-Adresse einzugeben ist, damit der Web Service aufgerufen werden kann.

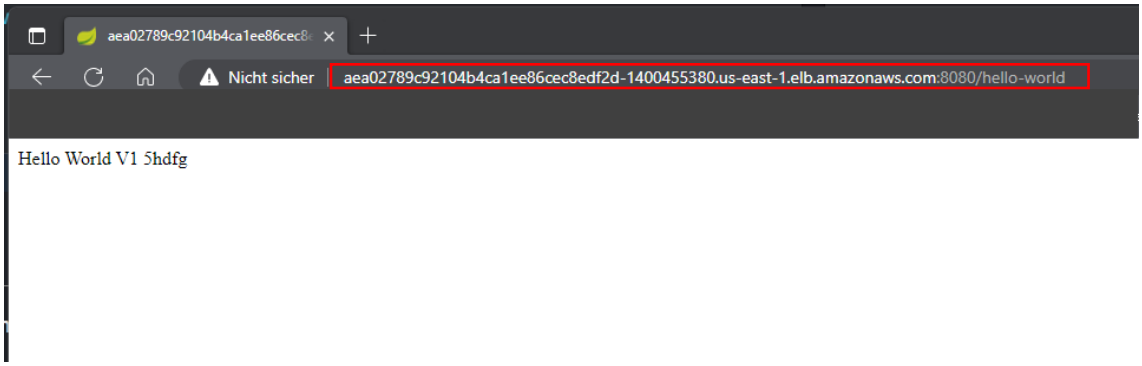


Abbildung 28 Aufruf des Webservice

Um die Bereitstellungen zu duplizieren wird wie gefolgt, weitergegangen:

- kubectl scale deployment hello-world-rest-api –replicas=3

Um Informationen über die duplizierten Pods zu erhalten wird:

- kubectl get pods

ausgeführt. In Abbildung 29, wird das Resultat des zweiten Befehls veranschaulicht.

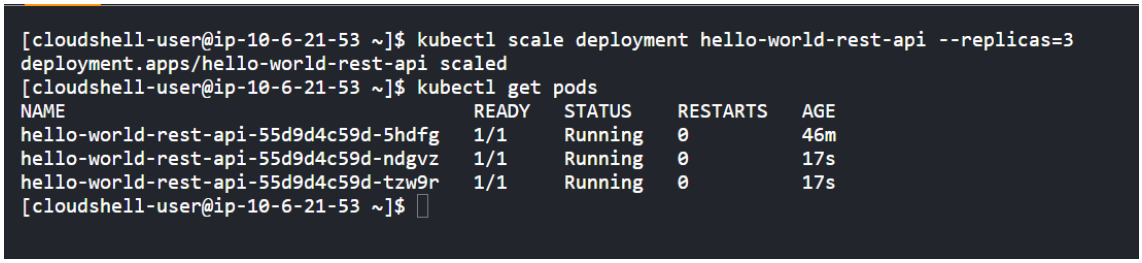


Abbildung 29 Duplizierung der Pods

Zum ersten Zubziel **CLI Unterstützung** in AWS, muss wie auch festgestellt per Prototyp, eine Zusatzinstallation des EKS CLIs, dem sogenannten eksctl ausgeführt werden. Außerdem muss bekannt gegeben werden, dass der eksctl die Yaml-Files unterstützt. In der Dokumentation von eksctl konnte festgestellt werden, dass auch hier eine lokale Installation ermöglicht wird.

(Weaveworks 2023)

Doch den Hyperscalern äquivalente Voraussetzungen bereitzustellen, wurde auch in AWS, bei der Erstellung des EKS Clusters der AWS CloudShell, über den Webbrowser benutzt.

Im Grunde genommen kann allgemein gesagt werden, dass die Bedienung des eksctls einfach gewesen ist und während des Prototypings, keine Schwierigkeiten vorbereitet hat.

- rollenbasierte Zugriffssteuerung (RBAC)

Bei der Definierung des Root-Accounts und IAM-Users konnte begegnet werden, dass auch für EKS manuell eine RBAC Einstellung ausgelöst werden kann. Des Weiteren sind auch hier Templates vordefiniert gewesen, die einfach übernommen werden können. Doch auch in AWS ist eine Master Global Access nicht vorgesehen.

(Amazon.com AWS 2023)

- Monitoring

Das eigentliche Monitoring Tool ist der Amazon CloudWatch, doch auch weitere Tools wie AWS CloudTrail, AWS Certificate Manager und Amazon EC2 Dashboard gewähren ähnliche Funktionen. Anhand der Dokumentation von EKS, konnte ebenso festgestellt werden, dass Monitoring per Drittanbieter Tools durchgeführt werden können, wie zum Beispiel das Grafana Monitoring Tool und viele mehr. Durch die Vielzahl an Möglichkeiten in AWS Monitoring zu bereiten, können auch einzelne Metriken selektiert werden. Aus diesem Grund können sowohl Cluster, als auch Pods Metriken überwacht werden. Für Clusterüberwachung ist Amazon EMR (Elastic MapReduce) und für die Überwachung von Pods ist der Container Insights sehr gut geeignet.

(Amazon.com AWS 2023)

- Preisgestaltung

Die genauere Betrachtung der Preisgestaltung in EKS, ist sehr empfohlen. Zu je Ressourcengruppe sind die ersten 40 Überwachung des Clusters kostenlos zur Verfügung gestellt, danach fallen Kosten für die Überwachungen. Doch auch in AWS, sind für die Bereitstellung des EKS Clusters, differenzierte Preismodellierungen auswählbar.

(Amazon.com AWS 2023)

- Kubernetes Versionsunterstützung

Die Kubernetes Versionsunterstützung ist in der EKS Dokumentation sehr kompliziert dargestellt, doch wie auf Abbildung 30 präsentiert, können die unterstützten Version nachgelesen werden. Diese sind von 1.22 bis 1.26. Das Update einer vorhandenen EKS Cluster funktioniert nur inkrementell (eks. <n+1>). Das heißt, nichts anderes wie, dass vorhandene Cluster nur auf eine höhere Version aktualisiert werden können, sinngemäß, um den Zusammenhang der Hauptversion nicht zu entfalten. Jedoch sind Downgrades auch in Elastic Kubernetes Service nicht direkt durchführbar. Auch hier ist eine Wiederherstellungsdatei zu entwerfen, damit der neue Cluster mit der gewünschten Altversion erstellt werden kann. Anschließend sind die Wiederherstellungsdateien zu transferieren. Grundsätzlich ist ein automatisches Update der Kubernetes Version nicht ausgeschlossen, doch es müssen, dafür Voraussetzungen erfüllt werden wie, die AMI (Amazon Machine Images) und Einplanung einer Strategie für das Node Pool. So kann auch eine automatische Aktualisierung der K8s Versionen miteingebunden werden.
(Amazon.com AWS 2023)

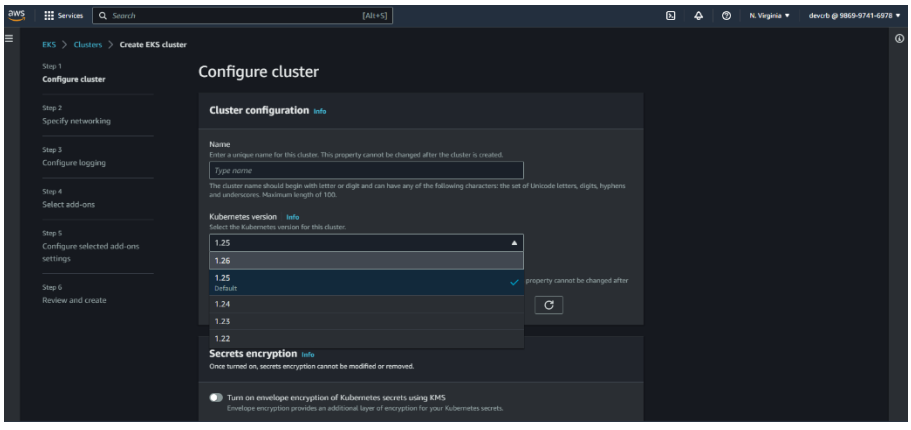


Abbildung 30 Kubernetes Versionen

- Überwachung der Knotenintegrität

Über das sechste Zielkriterium, konnte per Dokumentation des EKS fündig werden, dass eine automatische, vertikale Pod Skalierung und eine automatische Überwachung der Nodes per verwaltete Nodepools möglich sind. Aber eine automatische Node- Reparatur bietet das EKS nicht an. Wie viele Pods in einem Cluster bereitgestellt werden können, konnte nicht herausgefunden werden über die Dokumentation, doch dass 250 Pods in einer Node bereitgestellt werden können, ist eindeutig gekennzeichnet.

(Amazon.com AWS 2023)

- Spawn Cluster Zeit

Die Bereitstellung des Cluster wurde per Prototyp ausgetestet. In Abbildung 23 und 24 ist die Anfangszeit und Endzeit der Clusterbereitstellung, anhand eines rechten Rechtecks markiert. Die Bereitstellung hat ungefähr 17 Minuten gedauert. Ein sehr wichtiges Augenmerk, ist bei der EKS Clusterbereitstellung gewesen, dass EKS durch die Anbindung mit AWS Fargate und AWS ECS sehr flexibel ist. AWS Fargate ist ein Serverless-Computing-Engine, welches anstatt einer virtuellen Maschine verwendet werden. AWS ECS ist Amazons elastischer Container Service, dass voll verwaltet Container bereitstellen und orchestrieren kann. Weiters konnte eruiert werden, dass in einem EKS Cluster maximal 13500 Nodes installiert werden können. Außerdem bietet Amazon sehr viele Möglichkeiten an Betriebssystem für die Nodes. Diese sind an erster Stelle Amazon Linux 2; Ubuntu; Bottlerocket und Windows. Des Weiteren ist noch zu erwähnen, dass AWS zwar keine TPU Nodes bereitstellt und keine Node Lokalisierung anbietet, doch stattdessen EC2 P3 Instanzen hat und diese für Machine Learning und High Performance Computing Anwendungen zur Verfügung stellt.

(Amazon.com AWS 2023)

4.6 Prototyping in Google Kubernetes Engine (GKE)

Der letzte verwaltete Kubernetes Service, welcher im Rahmen der Bachelorarbeit, anhand eines Prototyps erstellt wird, ist der GKE. Ähnlich wie bei den anderen Prototypen wird auch hier der Schritt der Account Erstellung übersprungen. Die Google Cloud Plattform kann über den Link https://cloud.google.com/?hl=de erreicht werden. Für eine bessere Nachvollziehbarkeit wird der GKE mithilfe des GUI Dashboard erstellt. Dieser Vorgang könnte auch bei anderen Cloud Providern verwendet werden, wie beispielsweise in AWS und Microsoft Azure. Sobald der Account erstellt und die Cloud Landschaft, des Google Cloud Platform betreten worden ist, muss dem Account Inhaber: in klar sein, dass dieser Account einen Master Global Access besitzt. Das heißt, dass der Account des Cloudinhaber: in, stets auch alle Rechte besitzt und in der Cloud Bearbeitungen durchführen kann, ohne spezifische User: innen anlegen zu müssen.

Bevor es mit der Erstellung des GKE Cluster weitergeht, muss vorab innerhalb des Google Clouds ein Projekt erzeugt werden, in welchen der zu erstellende GKE Cluster und weitere Ressourcen zugeordnet werden können. In Abbildung 31, wird angezeigt, wo der Google Cloud Shell und die Suchleiste zu finden wäre und weiters der Bereich, um ein neues Projekt anzulegen.

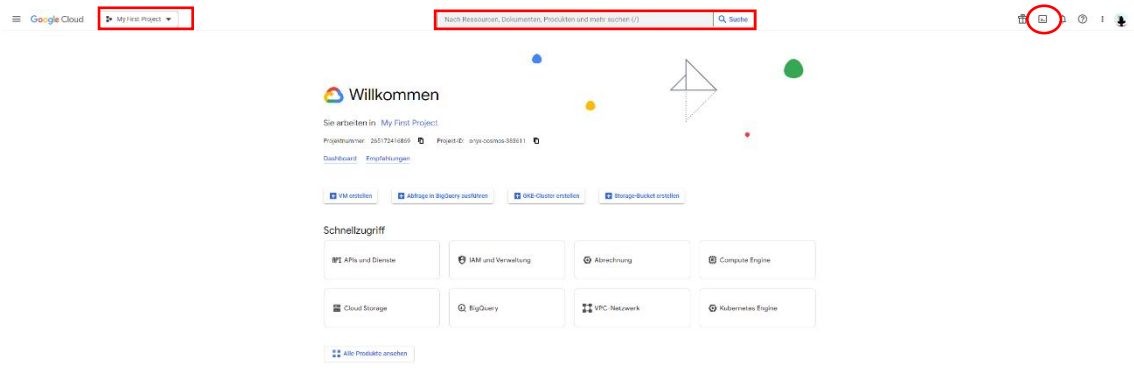


Abbildung 31 Google Cloud Konsole

Die mittlere Markierung ist eigentlich sehr selbst erklärend, das sogenannte Suchfeld in der Cloud. Die runde Markierung rechts in Abbildung 31, zeigt an, wo der Google Cloud Shell sich befindet, während die rechteckige Markierung links veranschaulicht, wie ein Projekt erstellt wird. Sobald darauf angeklickt wird, öffnet sich ein neues Fenster, wo ein neues Projekt generiert werden kann. Wie auf

Abbildung 32 zu sehen, habe ich für den Prototyp den Projektnamen „Prototyp1“ ausgewählt.

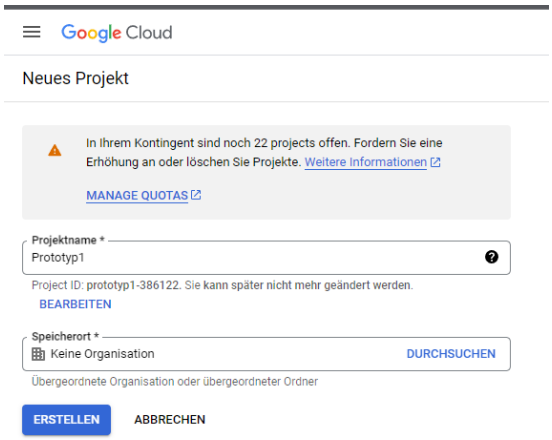


Abbildung 32 Erstellung eines neuen Projekts

Sobald das Projekt erstellt ist, muss zu nächst der Kubernetes Engine aktiviert werden. Mithilfe der Suchfunktion wird nach „Kubernetes Engine“ gesucht und geöffnet. Die Ergebnisse der Suchfunktion, können anfangs zu Irritation führen. Der zu aktivierende Kubernetes Engine ist in Abbildung 33 abgebildet.

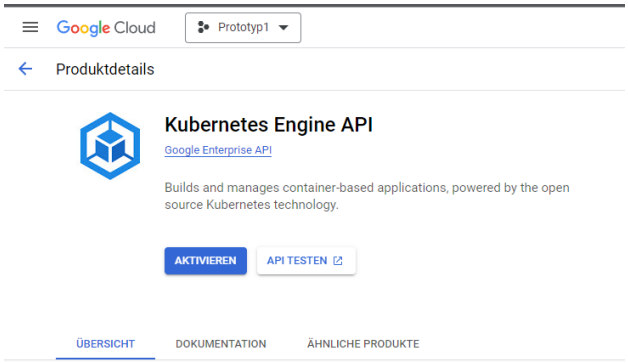


Abbildung 33 Kubernetes Engine API

Anhand dieser Einbindung des Kubernetes Engine API, ist nun die Erstellung eines Google Kubernetes Engine Clusters, in jeweiligen Projekt gewährt.

Es eröffnet sich automatisch der Kubernetes Engine, welcher auch in Abbildung 34 dargestellt wird. Sobald auf die Markierung angeklickt wird, muss der Cluster konfiguriert werden, bevor dieser erstellt wird. Google Cloud bietet zwei Optionen für die Erstellung eines Clusters an, entweder den Autopilot Cluster oder der

Standard Cluster. Der Unterschied hierbei ist, dass beim Autopiloten, der Node; der Netzwerk, die Sicherheit und die Systemlogs und weitere Monitoring Elemente, automatisch erstellt werden. Bei der Standardkonfiguration ist alles manuell zu definieren, wie der GKE Cluster harmonisieren soll. Der Prototyp wird per Standardkonfiguration bereitgestellt.

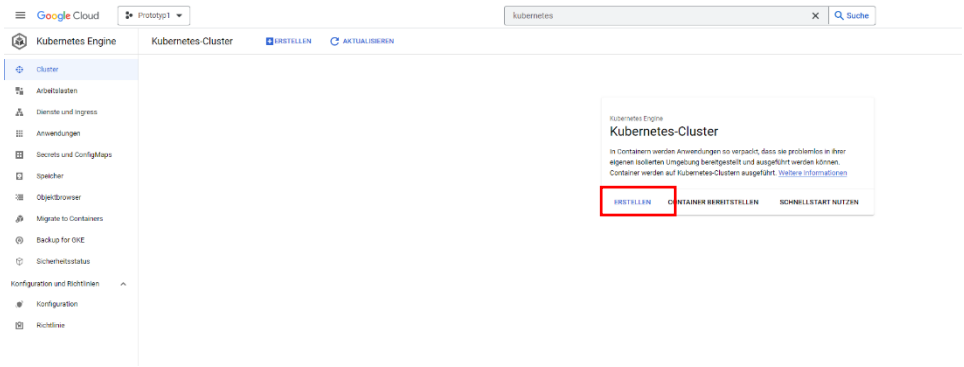


Abbildung 34 Cluster-Erstellung

Bevor der Cluster in Detail konfiguriert werden kann, müssen vorerst die Grundlagen abgedeckt werden. Der Name des Clusters, wie in anderen Prototypen, die erstellt worden sind, wird erneut „bachelor-cluster“ benannt. Der Standorttyp des Clusters, wird aufgrund der fallenden Kosten auf den günstigsten Standort zugewiesen. In diesem Fall ist es die „us-central1-a“. Die Kubernetes Version gehört ebenso zu den Grundlagen. Für die Erstellung des Clusters, wurde die aktuelle Release Version ausgewählt, die bei der Standardeinstellung dabei ist. In Abbildung 35 wird mithilfe des roter Markierung visualisiert, welche weiteren Unterpunkte es gibt, die bei Bedarf konfiguriert werden können.

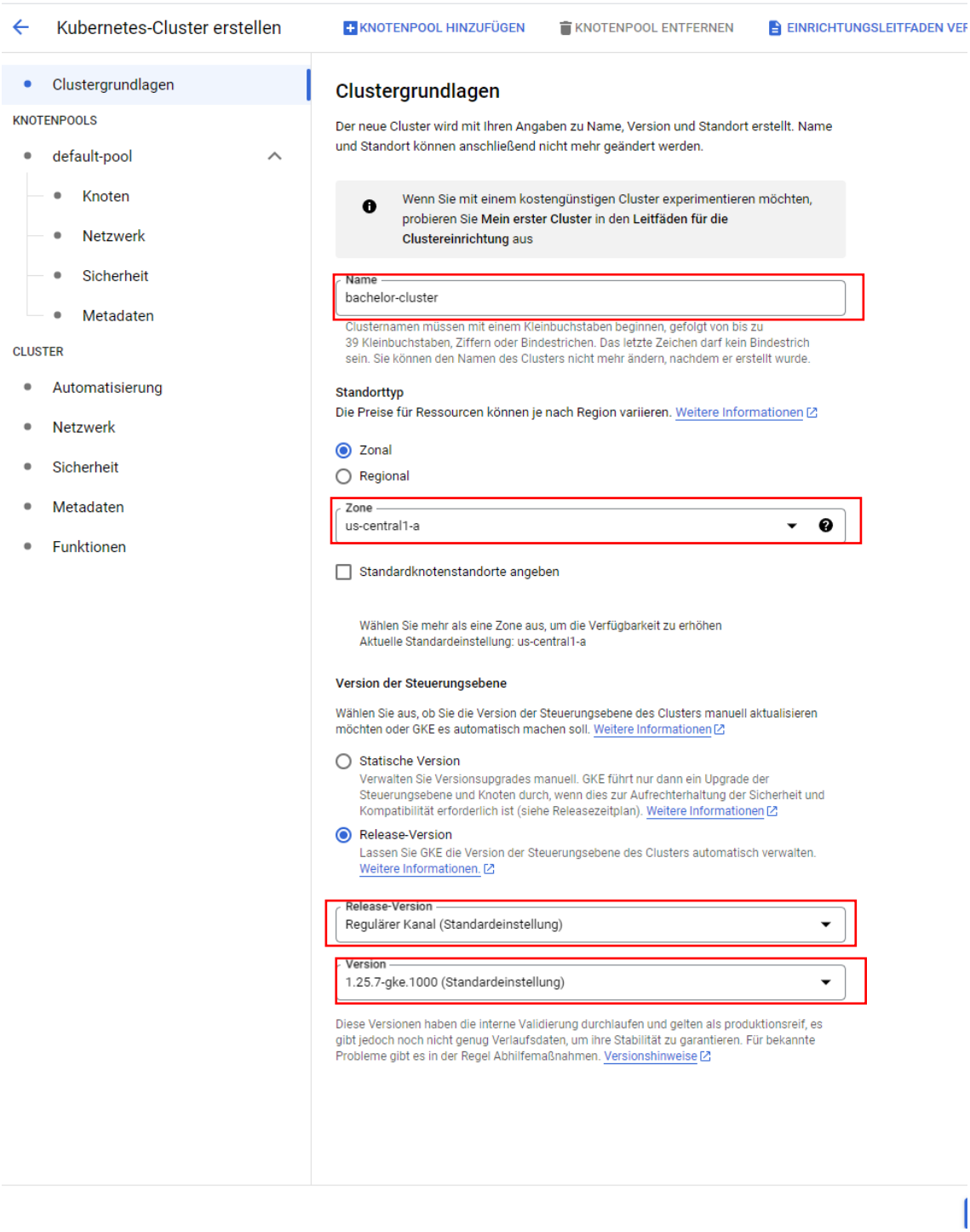


Abbildung 35 Clustergrundlagen GKE

Nach der Definierung der Clustergrundlagen ist der nächste Schritt, die Node Konfiguration, doch davor ist im Zwischenschritt die Anzahl der Nodes zu bestimmten. Dieser ist, wie in Abbildung 35 ersichtlich unter dem Untermenü „default pool“ auszuwählen. Anzahl der zu erstellenden Nodes in diesem

Prototypen beträgt drei Nodes. Nun kann in den nächsten Schritt, auf die Node Konfiguration gewechselt werden. Das Image-Typ des Nodes, ist per Standard definiert. Das heißt, es ist das von Linux definierte Container optimierte Betriebssystem „cos_containerd“. Allgemein bieten Cloud Provider sehr viele Variationen von virtuellen Maschinen. Für das Prototyping, ist der kleinste Maschinentyp ausreichend. Daher wurde aus der Reihe „N1“, der Maschinentyp „n1-standard-1“ ausgewählt mit 1vCPU und 3,75GB Arbeitsspeicher. Die Konfiguration für die Zwecke des Prototyps sind vollendet. Der GKE Cluster kann nun erstellt werden.

In Abbildung 36, ist zu sehen, dass die Erstellung eines, wie oben definierten GKE Clusters ungefähr sechs Minuten beträgt.

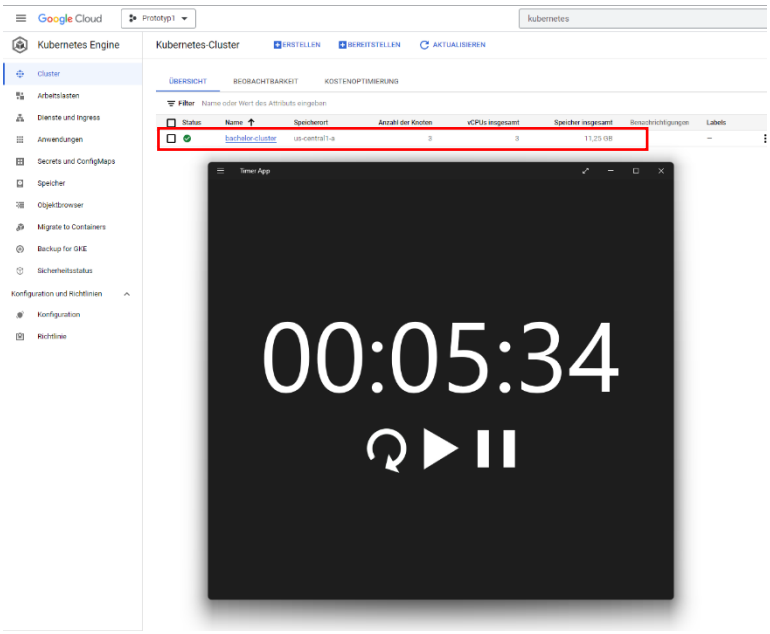


Abbildung 36 Spawn Cluster Zeit GKE

Die Markierung in Abbildung 36, zeigt weiters noch den Status des Clusters. Daran kann erkannt werden, dass der Cluster bereitgestellt worden ist, sobald da ein grünes Häkchen ist.

Darauffolgend kann nun der GitHub Repository in den GKE Cluster bereitgestellt werden. Dafür ist man aufgefordert den Google Cloud Shell auf der Google Cloud Konsole zu aktivieren, wie rechts oben in Abbildung 31 markiert. Weiters wird ein Cloud Shell Befehl benötigt, damit der CLI direkt sich mit dem Cluster verbindet. Der CLI Befehl ist direkt im „bachelor-cluster“ zu entnehmen, wie in Abbildung 37 vorgezeigt.

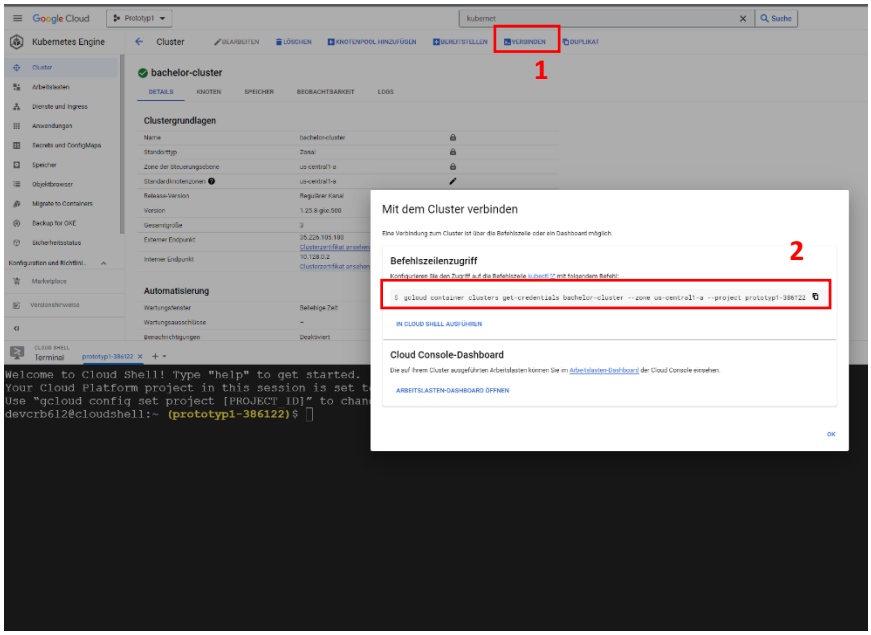


Abbildung 37 CLI Befehl für den GKE Cluster

Durch das Anklicken der ersten Markierung, erscheint das Fenster, wo die zweite Markierung ersichtlich ist. Hier kann der Kommando Befehl kopiert und in den Google Cloud Shell hinzugefügt werden, um den Shell mit dem GKE Cluster zu verbinden.

Der Befehl lautet:

- gcloud container clusters get-credentials bachelor-cluster --zone us-central1-a --project prototyp1-386122

Anschließend ist der CLI zu autorisieren.

Nun kann wie in anderen Prototypen, auch hier die Bereitstellung vorbereitet und danach exponiert werden, wie in Abbildung 38, per:

- kubectl create deployment hello-world-rest-api --image=in28min/hello-world-rest-api:0.0.1.RELEASE
- kubectl expose deployment hello-world-rest-api --type=LoadBalancer --port=8080

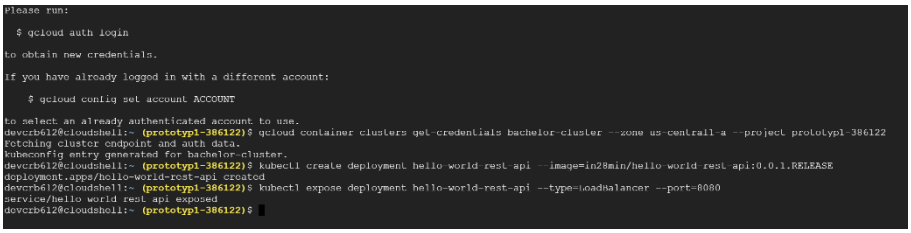


Abbildung 38 Bereitstellung des GKE Clusters

Im weiteren Feld, kann auch über die Google Cloud Konsole, per Dienste und Ingress, überprüft werden, ob der „Hello-World“ Web Service fertiggestellt ist. Weiters kann auch wie in Abbildung 39 markiert, abgelesen werden wie die externe IP Adresse lautet, um auf das Web Service zuzugreifen.

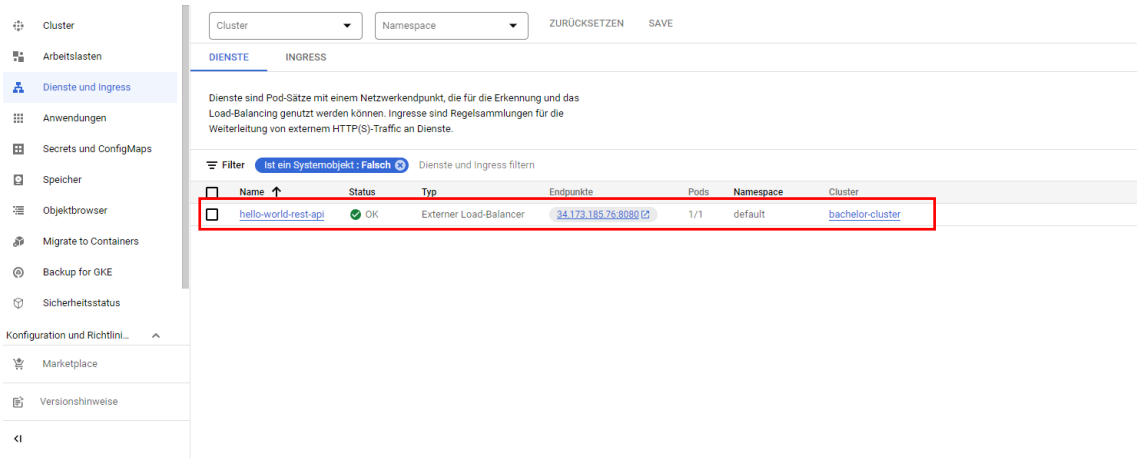


Abbildung 39 Bereitstellung des GKE Clusters

In einem neuen Webbrowser kann die externe IP-Adresse des Clusters, mit dem definierten Portzugang besucht und somit das Web Service aufgerufen werden. In Abbildung 40, wird der die externe IP-Adresse angezeigt.

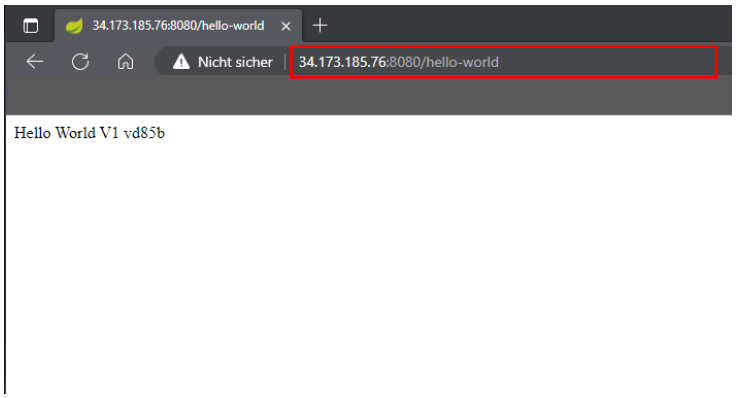


Abbildung 40 Aufruf der externe IP-Adresse des GKE-Clusters

Durch die Befehle:

- Kubectl get svc
- Kubectl scale deployment hello-world-rest-api –replicas=3

kann erstmals im Google CLI nachgesehen werden, welche Arten von Services vorhanden sind. Weiters wird wie in der Beschreibung der Prototyp Vorgehensweise, dass Web Service verdreifacht. Dies setzt sich aus dem zweiten Befehl, welcher in der CLI eingegeben wird.

Mit dem Befehl:

- Kubectl get replicaset

Kann sichergestellt werden, dass zwei weitere Web Service Duplikate erstellt worden sind. Die zu eingebenden Befehle, sind in Abbildung 41 wiedergeben.

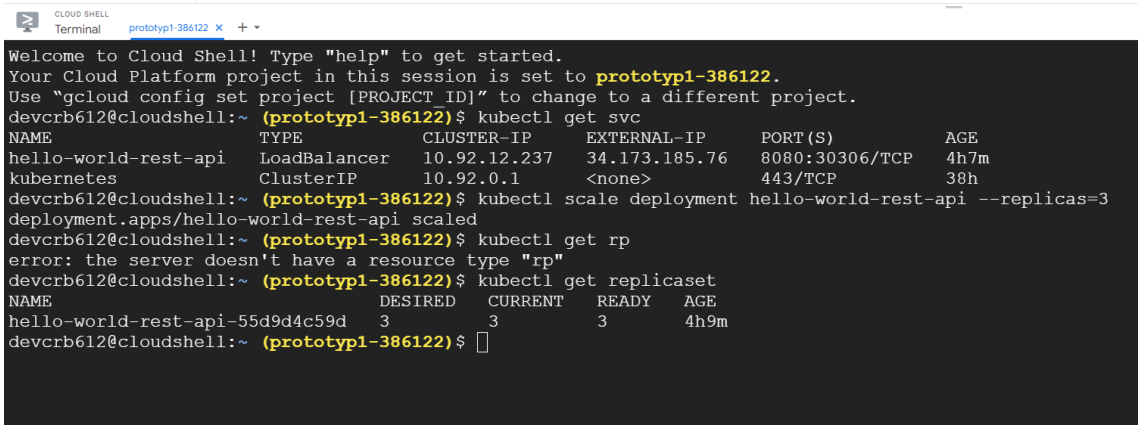


Abbildung 41 Erstellung von Duplikaten des Webservices

Nach der Einstellung der Anzahl der Duplikate, des vorhandenen Web-Services, erhielt ich im Webbrowser folgende Hello World V1 Versionen:

- Hello World V1 gp84n
- Hello World V1 fk8gv
- Hello World V1 vd85b

- Command Line Interface (CLI)

Zu den Subkriterien des ersten Zielkriterium ist mithilfe der Dokumentation und des Prototyps eruiert werden können, dass in Google Cloud Shell, keine Zusatzinstallationen notwendig gewesen sind, bis auf die Autorisierung der Bearbeitung. Laut der Dokumentation ist die Verwendung von Yaml-Files in Google nicht ausgeschlossen. Während des Prototyps ist auf kein Verweis beziehungsweise Hinweis auf eine lokale Installation zugestoßen worden, doch auch eine lokale Installation ist in Google gewährt.

(Google Cloud 2023)

- rollenbasierte Zugriffssteuerung (RBAC)

Aufgrund der Master Global Access Funktion von GCP ist bei der Bereitstellung des GKE Clusters keine weiteren Restriktionen notwendig gewesen, doch auch

GCP bietet RBAC vordefinierte Profile, welche für je Projekt und Rolle zugewiesen werden können.

(Google Cloud 2023)

- Monitoring

Wie im zweiten Kapitel bekannt gegeben, ist Kubernetes ein Produkt von Google, dass grundsätzlich für Microservices gedacht ist. Aus diesem Grund bietet Google Cloud zusätzlich zum Cluster Monitoring und Pod-Monitoring, noch die API-Monitoring. Services sind per APIs miteinander verbunden und bilden somit eine Cloud Umgebung. Weiterer erwähnenswerter Punkt ist, dass GCP auch Drittanbieter Monitoring Tools, wie Grafana und Prometheus nicht ausgelassen hat aus dem Sortiment.

(Google Cloud 2023)

- Preisgestaltung

Die preisliche Gestaltung von GKE ist bei Standard Instanzen kostenlos zur Verfügung gestellt. Bei größeren Reichweiten von Instanzen fallen wiederum Kosten an für die virtuelle Maschinen. Jegliche Features sind weiters kostenlos. Auch der GKE bietet für größere Clusterbereitstellungen differenzierte Preismodellierungen an.

(Google Cloud 2023)

- Kubernetes Versionsunterstützung

Ein Downgrade der Kubernetes Version ist auch in Google nicht vorgesehen, doch das alternative Manövrieren des Downgrade Vorgangs ist nicht ausgelassen. Automatische Updates der K8s Versionen sind per Nodepool zugänglich. Zusätzlich sind in Tabelle 3 die aktuellen Kubernetes Versionen je Patch Version reflektiert.

Tabelle 3 Übersicht der Kubernetes Versionen in GKE

1.26.3	gke.1000
1.26.2	gke.1000
1.25.8	gke.1000
1.25.8	gke.500
1.25.7	gke.1000 (Standard)
1.24.12	gke.1000

1.24.12	gke.500
1.24.11	gke.1000
1.24.10	gke.2300
1.24.10	gke.1200
1.24.9	gke.3200
1.23.17	gke.2000
1.23.17	gke.1700
1.23.17	gke.300
1.23.16	gke.2500
1.23.16	gke.1400
1.22.17	gke.8000
1.22.17	gke.7500
1.22.17	gke.6100
1.22.17	gke.5400
1.21.14	gke.18800
1.21.14	gke.18100
1.21.14	gke.15800
1.21.14	gke.8500

- Überwachung der Knotenintegrität

In der Dokumentation von Google Cloud Plattform ist die maximale Anzahl der Pods per Nodes bekannt gegeben mit 110 Pods. Anhand der automatischen Node Reparatur in GKE, ist weiters auch die automatische Überwachung der Nodes und die automatisch, vertikale Skalierung der Pods ausführbar.

(Google Cloud 2023)

- Spawn Cluster Zeit

Zum letzten Zielkriterium konnte klargestellt werden, dass in GCP die Clusterbildung sowohl per Dashboard über die GUI und durch die Google CLI implementierbar sind. Betriebssysteme für die Nodes, sind die gängigsten OS-Modelle wie Container Optimised OS und Ubuntu von Linux und Windows Server.

Weiters bietet Google Kubernetes Engine die Möglichkeit, 15000 Nodes per Cluster bereitzustellen. Während der Node Konfiguration kann in GKE vordefiniert werden, die Nodes lokalisieren zu wollen und für Machine Learning und Artificial intelligence (Künstliche Intelligenz) – Anwendungen bietet GKE die TPU Nodes an.

(Google Cloud 2023)

5. Die Nutzwertanalyse

In diesem Kapitel der Arbeit wird, wie im methodischen Vorgehen definiert worden, Schritt für Schritt die Nutzwertanalyse dargestellt. Angefangen wird mit der Darstellung, der im Vorkapitel definierten Ziel- und Subzielkriterien. Anschließend folgt die Beschreibung der ausgewählten Methoden und die Kalkulation der Nutzenermittlung. Abschließend wird das Ergebnis der Analyse mit einer sternförmigen Matrix dargestellt und evaluiert.

5.1 Konzeptionierung des Bewertungsmodells (7 Schritte der Nutzwertanalyse)

Die sieben Schritte dieser Nutzwertanalyse basieren grundsätzlich auf die Rahmenbedingungen der Nutzwertanalyse, wie aus dem **Kapitel 4.2.1 Rahmenbedingungen für die sieben Schritte der Nutzwertanalyse.** Das Zielsystem, welches anhand der Prototypen und der Dokumentation überprüft worden ist, sieht wie in Abbildung 42 dargestellt, folgenderweise aus:

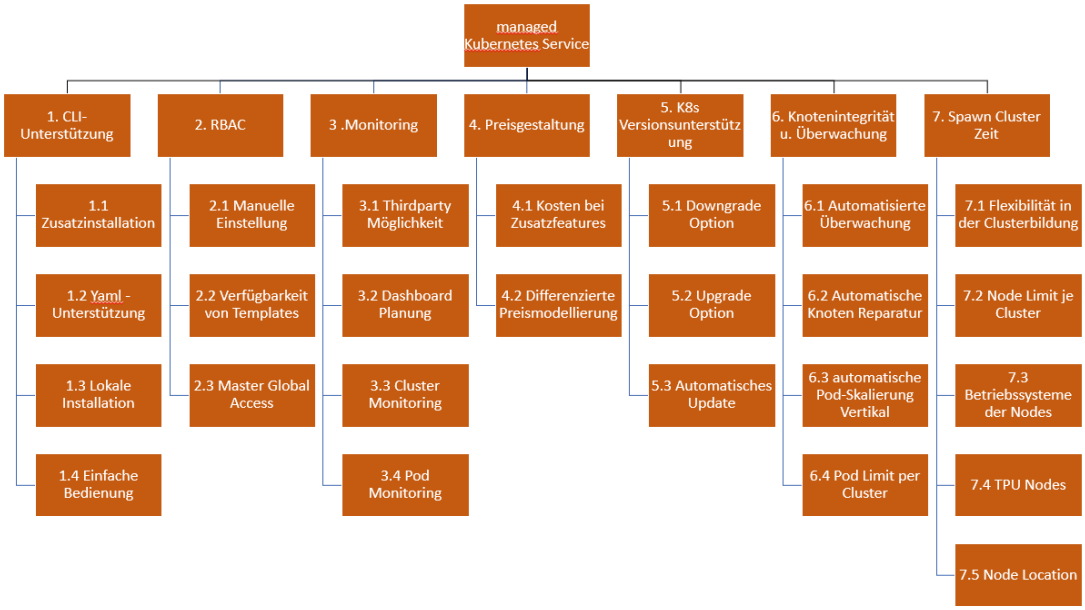


Abbildung 42 Zielsystem als Projektstrukturplan

Die sieben Zielkriterien inklusive der Subzielkriterien, bilden somit eine zwei schichte Zielebene. Insgesamt werden 25 Sub-Ziele untersucht, welches wiederum für eine Vorabbewertung mit einer Nutzwertanalyse ausreichend ist.

Die Gewichtung der Oberziele und Sub-Ziele folgt wie in Abbildung 8 vorgestellt, über die Methode des absoluten Maßstabes. Mithilfe dieser Methode sieht die Gewichtung, wie in Tabelle 4 aus:

Tabelle 4 Gewichtung der Oberziele

Zielkriterium	Oberziele	
	Wertung	rl. Gew (%)
1. CLI Unterstützung	5	18%
2. RBAC	4	14%
3. Monitoring	3	11%
4. Preisgestaltung	3	11%
5. K8s Versionsunterstützung	3	11%
6. Knotenintegrität u. Überwachung	5	18%
7. Spawn Cluster Zeit	5	18%
Summe	28	100%

Aus der Gewichtung der Oberziele, ergeben sich weitere Gewichte, wie die Knoten- und Stufengewichtung aus. Diese sind abgebildet in Tabelle 5. Wie schon in vorherigen Kapiteln erwähnt, dürfen in einer Nutzwertanalyse, keine Kosteneinflüsse vorhanden sein. Dies zu überprüfen wurde bei der Gewichtung ein weiteres Mal das Feld, wie in Tabelle 5 ersichtlich erstellt, um bei Zielkriterien, wo Kosten bemessen werden anhand einer „X“ Markierung, die Zeile zu markieren und zu eliminieren. Zielkriterium vier, ist ein solcher Kandidat gewesen, welcher doch aufgrund der JA-Nein Bewertung der Erfüllungsgrade, weiters noch erhalten bleiben durfte. Würden in dieser Nutzwertanalyse Kosten bemessen, so wären dies monetäre Kriterien, welche wiederum ein Kriterium für eine Kosten- Nutzen Analyse und nicht für die Nutzwertanalyse.

Anhand Abbildung 43, ist auch möglich herauszulesen, welche dieser Zielkriterien, wie bewertet werden. In der Spalte „Wertetabelle oder.

Wertefunktion“ wurde ermittelt, dass zwei Sub-Ziele per Wertefunktion dargestellt werden.

Zielkriterien	Knotengew.	Stufengew.	Kosteneinfluß	Wertetabelle od. Wertefunktion
1. CLI Unterstützung	**18**			
1.1 Zusatzinstallation	25	4,5		Wertetabelle
1.2 Yaml-Unterstützung	25	4,5		Wertetabelle
1.3 Lokale Installation	25	4,5		Wertetabelle
1.4 Einfache Bedienung	25	4,5		Wertetabelle
2. RBAC	**14**			
2.1 Manuelle Einstellung	33	4,66		Wertetabelle
2.2 Verfügbarkeit von Templates	33	4,66		Wertetabelle
2.3 Master Global Access	33	4,66		Wertetabelle
3. Monitoring	**11**			
3.1 Thirdparty Möglichkeit	25	2,75		Wertetabelle
3.2 Dashboard Planung	25	2,75		Wertetabelle
3.3 Cluster Monitoring	25	2,75		Wertetabelle
3.4 Pod Monitoring	25	2,75		Wertetabelle
4. Preisgestaltung	**11**			
4.1 Kosten bei Zusatzfeatures	50	5,5		Wertetabelle
4.2 differenzierte Preismodellierung	50	5,5		Wertetabelle
5. K8s Versionsunterstützung	**11**			
5.1 Downgrade Option	33	3,66		Wertetabelle
5.2 Upgrade Option	33	3,66		Wertetabelle
5.3 automatisches Update	33	3,66		Wertetabelle
6. Knotenintegrität u. Überwachung	**18**			
6.1 Automatisierte Überwachung	25	4,5		Wertetabelle
6.2 Automatische Knoten Reparatur	25	4,5		Wertetabelle
6.3 Vertikale Pod Skalierung	25	4,5		Wertetabelle
6.4 Pod Limit per Node	25	4,5		**Wertefunktion**
7. Spawn Cluster Zeit	**18**			
7.1 Flexibilität in der Clusterbildung	20	3,6		Wertetabelle
7.2 Node Limit je Cluster	20	3,6		**Wertefunktion**
7.3 Betriebssysteme der Nodes	20	3,6		Wertetabelle
7.4 TPU Nodes	20	3.6		Wertetabelle
7.5 Node Location	20	3,6		Wertetabelle

Abbildung 43 Knoten- und Stufengewichte des Zielkriteriums

Grundsätzlich gilt, dass diese 25 Zielkriterien für eine Nutzwertanalyse über die Hyperscaler, als die Spitze des Eisbergs gelten. Wenn die Zielhierarchie, sehr fein abgestimmt werden sollte, würde eine ausführliche Wertetabelle für die heraus Kristallisierung der einen Zielebenen aus opportunistischen Gründe mehr Sinn ergeben, falls die dafür notwendige Ersterfahrung vorhanden ist. Die Abfrage der Zielkriterien in diesem Fall werden per Ja-Nein Entscheidungen getroffen. Bei einer Ja-Nein Entscheidung gibt es insgesamt drei Noten für die Erfüllung der Anforderungen. Auch wenn das Zielkriterium in der Alternative gar nicht vorhanden ist, muss diese mit der Note drei bewertet werden, als

ausreichend. Falls das Zielkriterium die gefragte Anforderung sehr gut erfüllt erhält das Zielkriterium eine sechs. Bei Ergebnissen, die zwischen einer Ja und einer Nein liegen mit dem Erfüllungsgrad, wird der Mittelwert mit viereinhalb verwendet. Die Erfüllungsgrade sind im Vorkapitel, Abbildung 10 dargestellt. Die Werte und die Informationen für die Wertetabelle werden aus dem Kapitel des Prototyps entnommen und in nächsten Kapitel direkt in der Nutzenermittlung der Alternativen wiedergeben. Die Wertefunktionen, im Zielkriterium 6.4 und 7.2, bilden eine lineare Nutzwertfunktion, die grafisch visualisiert wird.

5.2 Nutzenermittlung

Mit den Informationen, aus Kapitel 3 die Nutzwertanalyse, wie ein Nutzen ermittelt werden kann, ist das Ergebnis der Nutzenermittlung in den Abbildungen, 45, 46 und 47 dargestellt. In Abbildung 44, sind die Wertefunktionen ersichtlich, die durch die Ermittlung der Nutzen bewertet werden konnten.

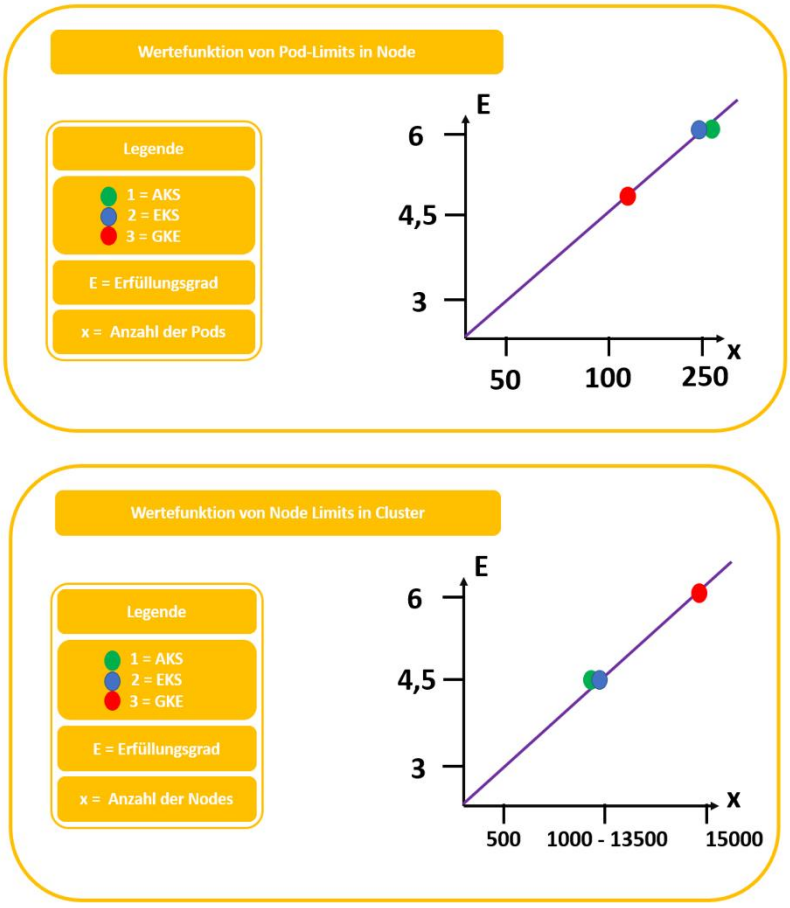


Abbildung 44 Wertefunktionen von Node Limits in Cluster und Pod Limits in Node

			Alternative 1 : AKS	
Nr.	Zielkriterien	Stufengew. %	Erfüllungsgrad	Nutzwert
1.1	1.1 Zusatzinstallation	4,5	6	0,270
1.2	1.2 Yaml-Unterstützung	4,5	6	0,270
1.3	1.3 Lokale Installation	4,5	6	0,270
1.4	1.4 Einfache Bedienung	4,5	4,5	0,203
1	**1. CLI Unterstützung**	**18**	**5,625**	**1,013**
2.1	2.1 Manuelle Einstellung	4,66	6	0,280
2.2	2.2 Verfügbarkeit von Templates	4,66	6	0,280
2.3	2.3 Master Global Access	4,66	3	0,140
2	**2. RBAC**	**14**	**4,99**	**0,699**
3.1	3.1 Thirdparty Möglichkeit	2,75	6	0,165
3.2	3.2 Dashboard Planung	2,75	6	0,165
3.3	3.3 Cluster Monitoring	2,75	4,5	0,124
3.4	3.4 Pod Monitoring	2,75	6	0,165
3	**3. Monitoring**	**11**	**5,63**	**0,619**
4.1	4.1 Kosten bei Zusatzfeatures	5,5	3	0,165
4.2	4.2 differenzierte Preismodellierung	5,5	6	0,330
4	**4. Preisgestaltung**	**11**	**4,5**	**0,495**
5.1	5.1 Downgrade Option	3,66	4,5	0,165
5.2	5.2 Upgrade Option	3,66	6	0,220
5.3	5.3 automatisches Update	3,66	4,5	0,165
5	**5. K8s Versionsunterstützung**	**11**	**4,99**	**0,549**
6.1	6.1 Automatisierte Überwachung	4,5	6	0,270
6.2	6.2 Automatische Knoten Reparatur	4,5	6	0,270
6.3	6.3 auto. Vertikale Pod Skalierung	4,5	3	0,135
6.4	6.4 Pod Limit per Node	4,5	6	0,270
6	**6. Knotenintegrität u. Überwachung**	**18**	**5,25**	**0,945**
7.1	7.1 Flexibilität in der Clusterbildung	3,6	4,5	0,162
7.2	7.2 Node Limit je Cluster	3,6	4,5	0,162
7.3	7.3 Betriebssysteme der Nodes	3,6	4,5	0,162
7.4	7.4 TPU Nodes	3,6	4,5	0,162
7.5	7.5 Node Location	3,6	6	0,216
7	**7. Spawn Cluster Zeit**	**18**	**4,8**	**0,864**
	Nutzwert	**100%**	**5,18**	**5,183**
	RANGFOLGE		**2**	

Abbildung 45 Nutzenermittlung AKS

			Alternative 2 : EKS	
Nr.	Zielkriterien	Stufengew. %	Erfüllungsgrad	Nutzwert
1.1	1.1 Zusatzinstallation	4,5	3	0,135
1.2	1.2 Yaml-Unterstützung	4,5	6	0,270
1.3	1.3 Lokale Installation	4,5	6	0,270
1.4	1.4 Einfache Bedienung	4,5	6	0,270
1	**1. CLI Unterstützung**	**18**	**5,25**	**0,945**
2.1	2.1 Manuelle Einstellung	4,66	6	0,280
2.2	2.2 Verfügbarkeit von Templates	4,66	6	0,280
2.3	2.3 Master Global Access	4,66	3	0,140
2	**2. RBAC**	**14**	**4,99**	**0,699**
3.1	3.1 Thirdparty Möglichkeit	2,75	6	0,165
3.2	3.2 Dashboard Planung	2,75	6	0,165
3.3	3.3 Cluster Monitoring	2,75	4,5	0,124
3.4	3.4 Pod Monitoring	2,75	6	0,165
3	**3. Monitoring**	**11**	**5,63**	**0,619**
4.1	4.1 Kosten bei Zusatzfeatures	5,5	3	0,165
4.2	4.2 differenzierte Preismodellierung	5,5	6	0,330
4	**4. Preisgestaltung**	**11**	**4,5**	**0,495**
5.1	5.1 Downgrade Option	3,66	4,5	0,165
5.2	5.2 Upgrade Option	3,66	6	0,220
5.3	5.3 automatisches Update	3,66	4,5	0,165
5	**5. K8s Versionsunterstützung**	**11**	**4,99**	**0,549**
6.1	6.1 Automatisierte Überwachung	4,5	6	0,270
6.2	6.2 Automatische Knoten Reparatur	4,5	3	0,135
6.3	6.3 auto. Vertikale Pod Skalierung	4,5	6	0,270
6.4	6.4 Pod Limit per Node	4,5	6	0,270
6	**6. Knotenintegrität u. Überwachung**	**18**	**5,25**	**0,945**
7.1	7.1 Flexibilität in der Clusterbildung	3,6	4,5	0,162
7.2	7.2 Node Limit je Cluster	3,6	4,5	0,162
7.3	7.3 Betriebssysteme der Nodes	3,6	4,5	0,162
7.4	7.4 TPU Nodes	3,6	4,5	0,162
7.5	7.5 Node Location	3,6	3	0,108
7	**7. Spawn Cluster Zeit**	**18**	**4,2**	**0,756**
	Nutzwert	**100%**	**5,01**	**5,008**
	RANGFOLGE		**3**	

Abbildung 46 Nutzenermittlung EKS

			Alternative 3 : GKE	
Nr.	Zielkriterien	Stufengew. %	Erfüllungsgrad	Nutzwert
1.1	1.1 Zusatzinstallation	4,5	6	0,270
1.2	1.2 Yaml-Unterstützung	4,5	6	0,270
1.3	1.3 Lokale Installation	4,5	6	0,270
1.4	1.4 Einfache Bedienung	4,5	6	0,270
1	**1. CLI Unterstützung**	**18**	**6**	**1,080**
2.1	2.1 Manuelle Einstellung	4,66	6	0,280
2.2	2.2 Verfügbarkeit von Templates	4,66	6	0,280
2.3	2.3 Master Global Access	4,66	6	0,280
2	**2. RBAC**	**14**	**5,99**	**0,839**
3.1	3.1 Thirdparty Möglichkeit	2,75	6	0,165
3.2	3.2 Dashboard Planung	2,75	6	0,165
3.3	3.3 Cluster Monitoring	2,75	6	0,165
3.4	3.4 Pod Monitoring	2,75	6	0,165
3	**3. Monitoring**	**11**	**6**	**0,660**
4.1	4.1 Kosten bei Zusatzfeatures	5,5	6	0,330
4.2	4.2 differenzierte Preismodellierung	5,5	6	0,330
4	**4. Preisgestaltung**	**11**	**6**	**0,660**
5.1	5.1 Downgrade Option	3,66	4,5	0,165
5.2	5.2 Upgrade Option	3,66	6	0,220
5.3	5.3 automatisches Update	3,66	6	0,220
5	**5. K8s Versionsunterstützung**	**11**	**5,49**	**0,604**
6.1	6.1 Automatisierte Überwachung	4,5	6	0,270
6.2	6.2 Automatische Knoten Reparatur	4,5	6	0,270
6.3	6.3 auto. Vertikale Pod Skalierung	4,5	6	0,270
6.4	6.4 Pod Limit per Node	4,5	4,5	0,203
6	**6. Knotenintegrität u. Überwachung**	**18**	**5,63**	**1,013**
7.1	7.1 Flexibilität in der Clusterbildung	3,6	4,5	0,162
7.2	7.2 Node Limit je Cluster	3,6	6	0,216
7.3	7.3 Betriebssysteme der Nodes	3,6	4,5	0,162
7.4	7.4 TPU Nodes	3,6	6	0,216
7.5	7.5 Node Location	3,6	6	0,216
7	**7. Spawn Cluster Zeit**	**18**	**5,4**	**0,972**
	Nutzwert	**100%**	**5,83**	**5,827**
	RANGFOLGE		**1**	

Abbildung 47 Nutzenermittlung GKE

5.3 Darstellung der Nutzwertanalyse per Matrix

Die Darstellung der Nutzwertanalyse wird anhand einer sternförmigen Matrix mit je sieben Zielkriterien dargestellt. In jedem Zielkriterium können maximal sechs Punkte erhalten werden, je Erfüllungsgrad. In Abbildung 48 sind die einzelnen Matrizen, je verwalteter Kubernetes Service der Hyperscaler präsentiert.

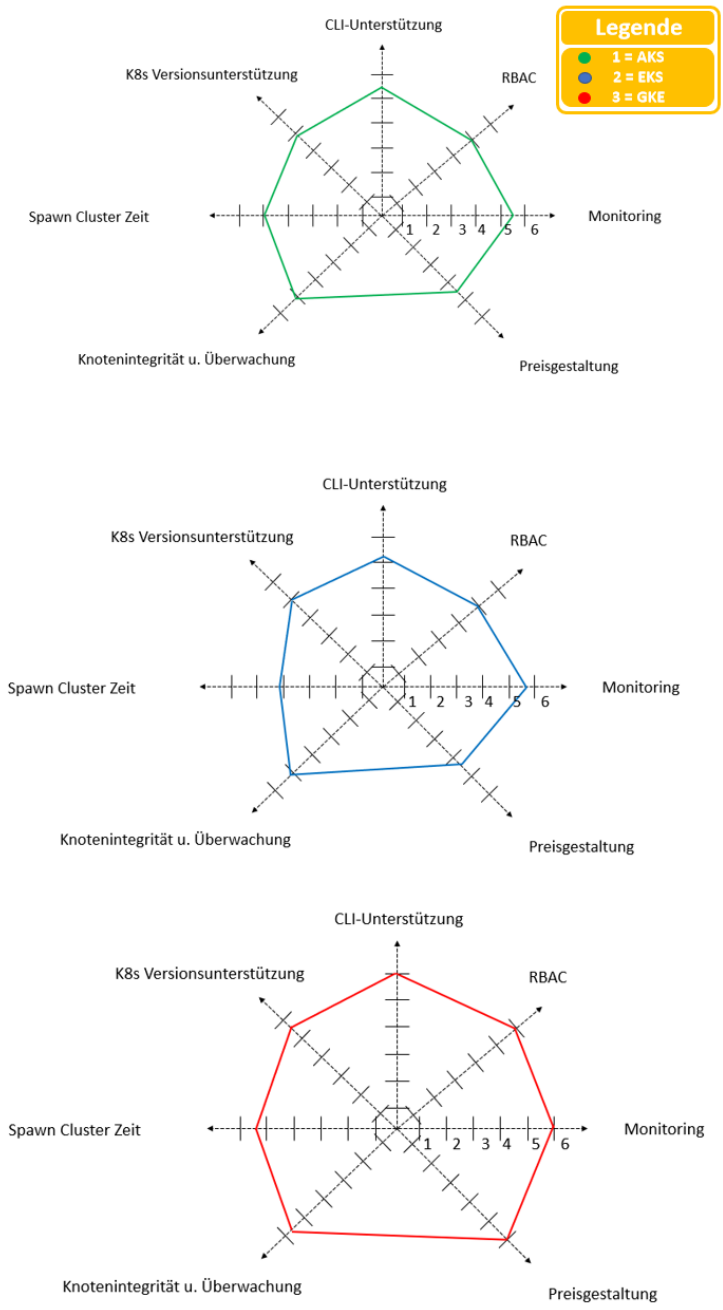


Abbildung 48 Matrizen verwaltete Kubernetes Services (AKS, EKS und GKE)

5.4 Evaluierung der Matrix

In einem Vergleich der Matrizen miteinander, lässt sich sofort herauskristallisieren, dass die Matrix von Google Kubernetes Engine, die meisten Punkte sammeln durfte, bei der Bewertung. Wird genauer in die einzelne Zielkriterien betrachtet, angefangen von der Verwendung der Command Line Interface, so darf festgestellt werden, dass alle Hyperscaler fast dasselbe anbieten bis auf, dass bei AWS CLI eine Zusatzinstallation des eksctl aufgefordert wird, um generell eine Bearbeitung in EKS durchführen zu können. In der rollenbasierten Zugangskontrolle (RBAC), ist Google ausnahmslos, der verwaltete Kubernetes Service, dass alle Punkte erreicht, aufgrund des vorhandenen Features, der sogenannte Master Global Access. Im dritten Zielkriterium, dem Monitoring, sind wiederum alle Hyperscaler fast im gleichen Stand, mit der einzigen Ausnahme, dass GKE aufgrund ihrer Haus aus angebotener Funktion, die bereitgestellten Cluster automatisch überwacht. Auch in Preisgestaltung ist der Cloud Provider Google an der Front, da das Produkt „Kubernetes“ ein Produkt von Google ist, ist der Cloud Provider hier sehr großzügig, bei Anbindung von weiteren Funktionen in GKE. Im sechsten Zielkriterium bietet keiner der Hyperscaler, ein direktes Downgraden der Kubernetes Version, doch mit den vielen Möglichkeiten der K8s Versionen ist wieder GKE, der Sieger dieses Rennens. In Funktionen, wie die Knotenintegrität und Knotenüberwachung, schlagen sich AKS und EKS tapfer miteinander, doch mit den Funktionalitäten des Google Kubernetes Engine‘s, wie automatische Knotenreparatur und automatisch, vertikale Pod- Skalierung, können die beiden Konkurrenten nicht mithalten. Das letzte Zielkriterium in der Nutzwertanalyse ist die Spawn-Cluster Zeit der verwalteten Kubernetes Services gewesen. Die Spawn-Cluster Zeit, gibt die Zeit an, wie lange ein Hyperscaler benötigt in der Cloud Umgebung einen verwalteten Kubernetes Cluster zu erstellen. Mit der Unterstützung der Prototypen, konnten die Zeit bemessen werden, wie lange eine Bereitstellung des Cluster gedauert hat. Mit zwei Minuten Differenz ist die Bereitstellung des GKE Clusters der eindeutiger Sieger dieser Bewertung gewesen, während die Bereitstellung eines EKS-Clusters ungefähr 17 Minuten gedauert hat.

6. Fazit der Resultate

Im letzten Kapitel der Bachelorarbeit wird erneut auf die Forschungsfrage eingegangen, indem der erste und der zweite Teilbereich der Bachelorarbeit prompt erwähnt wird. Anschließend folgt die Bewertung der Hypothese, um herauszufinden, ob diese der Wahrheit entspricht. Abschließend folgen die Ausblicke der verwalteten Kubernetes Services.

In dieser wissenschaftliche Arbeit wurde an erster Stelle veranschaulicht, wie die Innovation in der IT sich entwickelt hat, von der IT- Infrastruktur, der alten Generation bis hin in die neue Ära, die Cloud Technologie. Fortgesetzt wurde mit Cloud Computing und welche Cloud Provider zurzeit im Markt, die wohlbekanntesten sind.

Anschließend wurde versucht auf die Forschungsfrage einzugehen, indem der Wert der Containerisierung und Orchestrierung untersucht wurde. Die eigentliche Bedeutung von verwalteter Kubernetes Service, ließ sich ableiten durch die Orchestrierungstechnologien. Um die Forschungsfrage beantworten zu können, wurde eine Kombination aus zwei Forschungsmethoden verwendet.

Während im ersten Teil der Bachelorarbeit grundsätzlich auf die Theorie der Technologie und der Nutzwertanalyse eingegangen ist, wurde im zweiten Teil der Bachelorarbeit, die Frage anhand der Bereitstellung der Prototypen, jeweiliger verwaltete Kubernetes Services, auf die ausgewählten sieben Zielkriterien untersucht und bemessen.

Um ein aussagekräftiges Resultat mit der Nutzwertanalyse zu erzielen, mussten die sieben Oberziele, auf eine weitere Zielebene vertieft werden. Damit die Zielhierarchie weder zu grob noch zu fein wird, sind 25 Zielkriterien ausreichend gewesen, um die sieben Hauptkriterien bewerten zu können. Anschließend wurde die Nutzwertanalyse durchgeführt und die jeweiligen Nutzen der verwalteten Kubernetes Service einkalkuliert und in Rangfolgen visualisiert.

6.1 Assessment der Hypothese

Angesichts der sieben Zielkriterien, die definiert worden sind für die Bewertung der Nutzwertanalyse, konnte die Hypothese bestätigt werden, dass Google Kubernetes Engine die höchsten Nutzen erzielt, doch die Ergebnisse könnten andere Resultate ergeben, wenn zusätzlich zu den verwalteten Kubernetes Services der Hyperscaler auch dessen Cloud Umgebung miteinbezogen wäre.

Dies würde jedoch das Themengebiet der Forschungsfrage überschreiten und die Beantwortung der Forschungsfrage und generell die Erstellung einer spezifischen Forschungsfrage erschweren, weil eine Cloud Umgebung sehr tief ausgeweitet ist. Des Weiteren könnten auch andere Zielkriterien, wie Cluster Zonen auf der Welt, beziehungsweise Regionen und SLAs zu unterschiedlichen Ergebnissen führen.

6.2 Ausblick der verwalteten Kubernetes Services

Die Erkenntnisse der Nutzwertanalyse über die verwalteten Kubernetes Services wie AKS, EKS und GKE können von IT-Expert: innen; IT-Architekt: innen; Cloud Benutzer: innen und sowie auch IT- Berater: innen angewendet werden, um die Entscheidungstreffung zu beschleunigen, zu welchem Hyperscaler sie sich widmen sollen mit ihrer IT-Landschaft und welchen verwalteten Kubernetes Service sie sich dafür entscheiden sollen.

Außerdem könnten auch Personen, die in Berührung mit ERP-Systemen sind, von diesen Erkenntnissen profitieren und somit zu einer schnellen Entscheidungstreffung kommen, da auch ERP-Systeme sehr beliebter geworden sind in der Cloud.

Eine mögliche Weiterentwicklung dieser Technologie ist die Verbindung der Cluster mit dem aktuell sehr nachgefragten ChatGPT. Zwar befindet sich dieser in der Entwicklungsphase, doch es soll versucht werden, sowohl Workloads innerhalb des Systems, mit Automationen zu adaptieren und weiters Entwickler: innen, Supports und hilfreichere Unterstützungen bei der Bereitstellung anbieten.

7. Key Words

IT	Informationstechnologie
AWS	Amazon Web Services
CLI	Command Line Interface
K8s	Kubernetes
AKS	Azure Kubernetes Service
EKS	Elastic Kubernetes Service
GKE	Google Kubernetes Engine
NIST	National Institute of Standards and Technology
SOX	Sarbanes Oxley
LAN	Local Area Network
iOS	iPhone Operating System
Mac	Macintosh
VM	Virtual Machine
VPN	Virtual Private Network
WAN	Wide Area Network
QoS	Quality-of-Services
IaaS	Infrastructure as a Service
PaaS	Platform as a Service
SaaS	Software as a Service
Gbit/s	Gigabit pro Sekunde
SPOF	Single Point of Failure
OSI	Open Systems Interconnection
IP	Internet Protocol
FaaS	Function as a Service
IBM	International Business Machines
CSP	Cloud Service Provider
GCP	Google Cloud Platform

EC2	Elastic Compute Cloud
S3	Simple Storage Service
EBS	Elastic Block Store
CDN	Content Delivery Network
rkt	Rocket
CoreOS	Core Operating System
OpenVZ	Open Virtualization
LXC	LinuX Containers
LXD	LinuX Container Daemon
Cgroup	Control Groups
CPU	Central Processing Unit
API	Application Programming Interface
CNCF	Cloud Native Computing Foundation
HTTPS	Hyper Text Transfer Protocol Secure
ETCD	konsistenter Open-Source-Schlüssel-Wert-Speicher
ECS	Elastic Container Service
ECC	Error Correction Code/ Error Checking and Correcting
LCKS	Schlüsselwortsuche Verfahren Ciphertext
USD	US – Dollar
VS	Visual Studio
ECR	Efficient Consumer Response
DNS	Domain Name System
NWA	Nutzwertanalyse
Nges	Gesamtnutzwert
Ni	Nutzwertbeitrag
Ei	Erfüllungsgrad
wi	Gewichtung
Li	Leistung

RBAC	role – based access Control
GUI	graphical User Interface
SLA	Service Level Agreement
RAM	Random Access Memory

Literaturverzeichnis

Amazon.com AWS. 2023a. “What Is Amazon EKS? - Amazon EKS.” January 5, 2023. https://docs.aws.amazon.com/eks/latest/userguide/what-is-eks.html.

———. 2023b. “New Features and Changes in AWS CLI Version 2 - AWS Command Line Interface.” March 28, 2023. https://docs.aws.amazon.com/cli/latest/userguide/cliv2-migration-changes.html.

———. 2023c. “Verwalteter Kubernetes-Service – Amazon EKS – Amazon Web Services.” Amazon Web Services, Inc. May 8, 2023. https://aws.amazon.com/de/eks/.

Bulla, Chetan. 2019. “Cloud Monitoring System: A Review.” *International Journal of Engineering Sciencesand Management-A Multidisciplinary Publicationof VTU*, January. https://www.academia.edu/43711745/Cloud_Monitoring_System_A_Review.

Chandrakant, Kumar. 2019. “Mesos vs. Kubernetes | Baeldung.” August 27, 2019. https://www.baeldung.com/ops/mesos-kubernetes-comparison.

Cloud Native Computing Foundation. 2023. “CNCF Survey 2020.” Survey of Orchestriationtools. January 9, 2023. https://www.cncf.io/wp-content/uploads/2020/11/CNCF_Survey_Report_2020.pdf.

Course Hero. 2023. “Introduction to Kubernetes.” https://www.coursehero.com/file/80900554/Kubernetes-Made-Easypdf/.

Docker. 2022. “Docker Documentation.” Docker Documentation. December 29, 2022. https://docs.docker.com/.

GitHub. 2023. “Informationen Zu Repositorys - GitHub-Dokumentation.” April 27, 2023. https://docs.github.com/de/repositories/creating-and-managing-repositories/about-repositories.

Goel, Deepack. 2023. “Erweitertes Kubernetes - DZone Refcardz.” January 1, 2023. https://dzone.com/refcardz/advanced-kubernetes.

Google Cloud. 2023a. “Übersicht über GKE | Google Kubernetes Engine (GKE).” Google Cloud. January 5, 2023. https://cloud.google.com/kubernetes-engine/docs/concepts/kubernetes-engine-overview?hl=de.

———. 2023b. “Alle Preise | Compute Engine-Dokumentation.” Google Cloud. April 26, 2023. https://cloud.google.com/compute/all-pricing?hl=de.

———. 2023c. “Shielded GKE-Knoten verwenden | Google Kubernetes Engine (GKE).” Google Cloud. April 26, 2023. https://cloud.google.com/kubernetes-engine/docs/how-to/shielded-gke-nodes?hl=de.

Guyton, Stephen. 2019. “Containers & Containerization - Pros and Cons.” Atomic Spin. May 24, 2019. https://spin.atomicobject.com/2019/05/24/containerization-pros-cons/.

Hurwitz, Judith, Robin Bloor, Marcia Kaufmann, and Fern Halper, eds. 2010. *Cloud Computing for Dummies*. For Dummies. Hoboken, NJ: Wiley Pub.

Hwang, Yitaek. 2021. “State of Managed Kubernetes 2021.” *Geek Culture* (blog). June 8, 2021. https://medium.com/geekculture/state-of-managed-kubernetes-2021-43e8a4ca0207.

Intellipaat. 2023. “AWS vs Azure vs Google Cloud - Detailed Cloud Comparison.” Intellipaat Blog. January 5, 2023. https://intellipaat.com/blog/aws-vs-azure-vs-google-cloud/.

IT-Service Network. 2023. “Prototyping | Definition & Erklärung.” April 27, 2023. https://it-service.network/it-lexikon/prototyping.

Kamal, Muhammad Ayoub, Hafiz Raza, Muhammad Alam, and M. Mazliham. 2020. “Highlight the Features of AWS, GCP and Microsoft Azure That Have an Impact When Choosing a Cloud Service Provider.” *International Journal of Recent Technology and Engineering (IJRTE)* 8 (January). https://doi.org/10.35940/ijrte.D8573.018520.

kubernetes.io. 2023. “Version Skew Policy.” Kubernetes. April 25, 2023. https://kubernetes.io/releases/version-skew-policy/.

Microsoft Azure. 2023a. “Managed Kubernetes Service (AKS) | Microsoft Azure.” May 7, 2023. https://azure.microsoft.com/en-us/products/kubernetes-service.

———. 2023b. “Preise – Container Service | Microsoft Azure.” May 7, 2023. https://azure.microsoft.com/de-de/pricing/details/kubernetes-service/.

Microsoft Learn. 2023a. “Introduction to Azure Kubernetes Service - Azure Kubernetes Service.” January 5, 2023. https://learn.microsoft.com/en-us/azure/aks/intro-kubernetes.

———. 2023b. “Monitor Azure Kubernetes Service (AKS) with Azure Monitor - Azure Kubernetes Service.” March 8, 2023. https://learn.microsoft.com/en-us/azure/aks/monitor-aks.

———. 2023c. “Automatisches Durchführen eines Upgrades für Azure Kubernetes Service (AKS) Clusterknoten-Betriebssystemimages - Azure Kubernetes Service.” May 7, 2023. https://learn.microsoft.com/de-de/azure/aks/auto-upgrade-node-image.

———. 2023d. “Install the Azure CLI for Windows.” May 7, 2023. https://learn.microsoft.com/en-us/cli/azure/install-azure-cli-windows.

Nasser, Tableb, and Mohamed Elfadil. 2020. “Cloud Computing Trends: A Literature Review.” *Academic Journal of Interdisciplinary Studies* 9 (January): 91. https://doi.org/10.36941/ajis-2020-0008.

Reciprocity. 2021. “NIST’s Definition of Cloud Computing.” Reciprocity. 2021. https://reciprocity.com/blog/nists-definition-of-cloud-computing/.

Red Hat. 2020. “Was ist FaaS?” January 3, 2020. https://www.redhat.com/de/topics/cloud-native-apps/what-is-faas.

Rezanka, Robert. 2013. “Die Nutzwertanalyse in der Investitionsrechnung.” Wirtschaftsuniversität.

Rinza, Peter, and Heiner Schmitz. 1992. *Nutzwert-Kosten-Analyse*. 2nd ed. Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-51504-0.

Rocket. 2023. “Trying out Rkt - Rocket.” January 1, 2023. https://rocket.readthedocs.io/en/latest/Documentation/trying-out-rkt/.

Roper, Jack. 2022. “What Is a Kubernetes Cluster? Key Components Explained.” Spacelift. July 21, 2022. https://spacelift.io/blog/[slug].

Rountree, Derrick, and Ileana Castrillo. 2014. “The Basics of Cloud Computing.” *Elsevier Inc.* 1: 218.

TechTarget. 2023. “What Is a Command-Line Interface (CLI)?” SearchWindowsServer. March 28, 2023. https://www.techtarget.com/searchwindowsserver/definition/command-line-interface-CLI.

Ward, Chris. 2016. “An Introduction to CoreOS.” CloudBees. September 22, 2016. https://www.cloudbees.com/blog/an-introduction-to-coreos.

Weaveworks. 2023. “Eksctl.” May 8, 2023. https://eksctl.io/.

Zangemeister, Christof. 2014. *Nutzwertanalyse in der Systemtechnik: Eine Methodik zur multidimensionalen Bewertung und Auswahl von Projektalternativen*. BoD – Books on Demand.

Zhang, Qi, Lu Cheng, and Raouf Boutaba. 2010. “Cloud Computing: State-of-the-Art and Research Challenges.” *Journal of Internet Services and Applications* 1 (1): 7–18. https://doi.org/10.1007/s13174-010-0007-6.

Zhao, Minghao, Chengyu Hu, Xiangfu Song, and Chuan Zhao. 2019. “Towards Dependable and Trustworthy Outsourced Computing: A Comprehensive Survey and Tutorial.” *Journal of Network and Computer Applications* 131 (April): 55–65. https://doi.org/10.1016/j.jnca.2019.01.021.