**FH JOANNEUM - University of Applied Sciences**

**Entscheidung Cloud-Computing Plattform**

**Dokumentation**

**Eingereicht für die Lehrveranstaltung “E-Business Anwendungen”**

**Autoren:**

**Florian Reinprecht, BSc.**

**Elisabeth Fellner, BSc.**

**Labinot Jashanica, BSc.**

**Kristian Ndou, BSc.**

**Yannick Collasius, BSc.**

**Supervisor:**

**FH-Prof. Dipl.-Ing. Dr. techn. Erwin Zinser**

**FH-Prof. Mag. Dr. Robert Singer**

**Graz, 2019**



Inhaltsverzeichnis

[1 Microsoft Azure 3](#_Toc10538420)

[1.1 Allgemeines 3](#_Toc10538421)

[1.2 Struktur 3](#_Toc10538422)

[1.3 Resourcenverwaltung 6](#_Toc10538423)

[1.4 Studentenangebote – Azure for Students 7](#_Toc10538424)

[1.4.1 Voraussetzungen: 7](#_Toc10538425)

[1.4.2 Ablauf des Azure-Students Angebots: 7](#_Toc10538426)

[1.5 Kostenmanagement 7](#_Toc10538427)

[1.5.1 Verschiedene Services um Kosten zu überwachen: 10](#_Toc10538428)

[1.6 Docker in Azure 11](#_Toc10538429)

[1.6.1 Azure for Docker 11](#_Toc10538430)

[1.6.2 Azure Container Instances 12](#_Toc10538431)

[1.6.3 Azure Service Fabric 14](#_Toc10538432)

[1.6.4 Azure App Service 15](#_Toc10538433)

[1.6.5 Azure Virtual Machines 16](#_Toc10538434)

[2 Amazon Web Services 17](#_Toc10538435)

[2.1 Allgemeines 17](#_Toc10538436)

[2.2 Struktur 17](#_Toc10538437)

[2.3 Resourcenverwaltung 18](#_Toc10538438)

[2.4 Studentenangebote 19](#_Toc10538439)

[2.5 Kostenmanagement 19](#_Toc10538440)

[2.6 Docker 21](#_Toc10538441)

[2.6.1 Amazon ECS 22](#_Toc10538442)

[2.6.2 Amazon Fargate 23](#_Toc10538443)

[2.6.3 Amazon EKS 23](#_Toc10538444)

[2.6.4 AWS Batch 24](#_Toc10538445)

[2.6.5 Amazon ECS CLI 24](#_Toc10538446)

[2.6.6 Docker Swarm 24](#_Toc10538447)

[2.7 Pricing 25](#_Toc10538448)

[3 Google Cloud Platform 26](#_Toc10538449)

[3.1 Allgemeines 26](#_Toc10538450)

[3.2 Struktur 26](#_Toc10538451)

[3.3 Resourcenverwaltung 28](#_Toc10538452)

[3.4 Studentenangebote 29](#_Toc10538453)

[3.5 Kostenmanagement 29](#_Toc10538454)

[3.6 Docker 31](#_Toc10538455)

[3.6.1 Google App Engine 31](#_Toc10538456)

[3.6.2 Google Compute Engine (Linux) 32](#_Toc10538457)

[3.6.3 Google Compute Engine (Container-Optimized OS) 33](#_Toc10538458)

[3.6.4 Google Kubernetes Engine 33](#_Toc10538459)

[4 Entscheidung Plattform 35](#_Toc10538460)

Abbildungsverzeichnis

[Abbildung 1 - Azure Struktur 4](#_Toc10538461)

[Abbildung 2 - Role Based Access Control (RBAC) Konzept 5](#_Toc10538462)

[Abbildung 3 - Azure Management Group 6](#_Toc10538463)

[Abbildung 4 - Azure Subscription Struktur 6](#_Toc10538464)

[Abbildung 5 - Pricing Calculator Virtuelle Maschine 8](#_Toc10538465)

[Abbildung 6 - Azure Cost Management Übersicht 9](#_Toc10538466)

[Abbildung 7 - Azure Benutzerdefiniertes Dashboard 10](#_Toc10538467)

[Abbildung 8 - Data Storage Preise 11](#_Toc10538468)

[Abbildung 9 - Operations and Data Transfer Preise 12](#_Toc10538469)

[Abbildung 10 – Preisübersicht 13](#_Toc10538470)

[Abbildung 11 - Pricing für IP Adressen 13](#_Toc10538471)

[Abbildung 12 - Service Fabric Kostenübersicht 15](#_Toc10538472)

[Abbildung 13 - App Service Kostenübersicht 16](#_Toc10538473)

[Abbildung 14: AWS Angebote 19](#_Toc10538474)

[Abbildung 15: AWS Kostenmanagement 20](#_Toc10538475)

[Abbildung 16 - AWS Docker 21](#_Toc10538476)

[Abbildung 17: Funktionsweise Amazon ECS 22](#_Toc10538477)

[Abbildung 18 - Amazon ECS Preismodelle 22](#_Toc10538478)

[Abbildung 19 - Amazon EKS 24](#_Toc10538479)

[Abbildung 20 - AWS Preisrechner 25](#_Toc10538480)

[Abbildung 21 - GCP Hierarchie 27](#_Toc10538481)

[Abbildung 22 - GCP Resourcenorganisation 28](#_Toc10538482)

[Abbildung 23 - Beziehung zwischen Rechnungskonto und Zahlungsprofil 29](#_Toc10538483)

[Abbildung 24 - GCP Rechnungskonto und Zahlungsprofil 30](#_Toc10538484)

[Abbildung 25 - GCP Abrechnungsreport 31](#_Toc10538485)

[Abbildung 26 - GAE Kostenübersicht 32](#_Toc10538486)

# Microsoft Azure

In diesem Kapitel werden die wichtigsten Informationen zu Microsoft Azure (Allgemein) und die Möglichkeiten gegeben, die zur Verfügung stehen, um eine Mikro-Service Architektur basierend auf Docker aufzubauen.

## Allgemeines

Cloud-Computing Plattformen bieten eine moderne Alternative und eine höhere Flexibilität zu den herkömmlichen On-Premise Rechenzentren. Bei öffentlichen Clouds, so auch bei Microsofts Azure, ist der Anbieter für den Einkauf, die Wartung und Bereitstellung der Hardware verantwortlich. Es werden jene Dienste und Services gemietet, die man benötigt, nach dem „As-Needed“ Prinzip. Durch dieses Prinzip hat der User die Möglichkeit Dienste zu mieten die sonst zu teuer in der Anschaffung wären. Da bei Microsoft Azure das Prinzip “Pay-As-You-Go” angewandt wird, werden nur die Dienste gezahlt, wenn diese in Einsatz sind.

## Struktur

Die Microsoft Azure Struktur und Hierarchie setzt sich aus mehreren Komponenten zusammen. Diese Komponenten und deren Beziehung zueinander werden in der nachfolgenden Auflistung erläutert, sowie mit einer grafischen Abbildung für das bessere Verständnis abgebildet.

**Tenant:**

In Azure AD werden die Benutzer in Tenants unterteilt. Ein Tenants ist ein logisches Konstrukt, dass eine sichere, dedizierte Instanz von Azure AD darstellt, die typischerweise einem Unternehmen zugeordnet ist. Jede Subscription ist mit einem Azure AD-Tenant verbunden.

Es handelt sich, um eine dedizierte Instanz des Azure AD-Dienstes, den ein Unternehmen erhält und besitzt, wenn es sich für einen Microsoft Cloud Service wie Azure, Microsoft Intune oder Office 365 anmeldet. Jeder Azure AD-Tenant ist anders und von anderen Azure AD-Tenants getrennt.

**Subscription:**

Eine Azure-Subscription ist die Basiseinheit, in der alle Resources enthalten sind. Es definiert auch mehrere Grenzen innerhalb von Azure, wie z.B. die Anzahl der Kerne, virtuelle Netzwerke und andere Resourcen. Azure Resource Groups werden verwendet, um das Subscription-Modell weiter zu verfeinern und eine natürlichere Gruppierung von Resourcen zu ermöglichen.

**Resources:**

In Azure bezieht sich der Begriff Resource auf eine von Azure verwaltete Einheit. Azure Resourcen können virtuelle Maschinen, virtuelle Netzwerke, Web Apps, Machine Learning, Hadoop, Jenkins und Speicherkonten sein.

Liste aller verfügbaren Resourcen: <https://azuremarketplace.microsoft.com/en-us/marketplace/apps>

**Resource Groups:**

Jede Resource in Azure muss einer Resource Group zugeordnet sein. Eine Resource Group ist ein logisches Konstrukt, das mehrere Resources zusammenfasst, damit diese als einzige Einheit verwaltet werden kann. Beispielsweise können Resources, die einen ähnlichen Lebenszyklus haben, wie z.B. die Resources für eine n-tier Anwendung, als Gruppe angelegt oder gelöscht werden.

Machine generated alternative text:
Powershell 
Azure Portal 
REST API endpoints 
Azure Resource Manager 
Azure Resource Providers 
Azure CLI 
Trust 
Tenant 
Resource 
Policies 
Tenant 
Azure Active 
Directory 
Tenant 
• Role-Based 
Access Control 
Azure 
• Azure subscription 
Resource Group 
Resource 
Resource 
Li mits 
Roles 
Financial 
Commitment 
Limits 

Abbildung 1 - Azure Struktur

**Role Based Access Control:**

Das Role-Based-Acces-Control (RBAC) Schema für Rechtevergabe nach Best-Practices ist nativ in Azure eingebunden. Mithilfe von RBAC ist es möglich auf jeder Ebene (ab Subscription-Level) die Rechte auf Resourcen nach dem Least-Privileges Prinzip zu vergeben. Zusätzlich ist es möglich ab Tenant-Ebene (mithilfe von Azure-AD) den Benutzern spezifische Administratoren-Rollen zu vergeben.

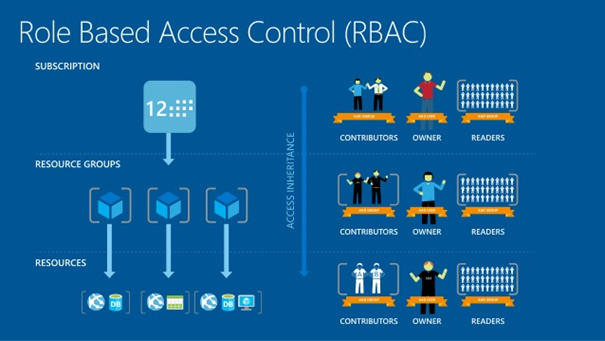


Abbildung 2 - Role Based Access Control (RBAC) Konzept

In Azure ist es möglich mehrere Subscriptions in einem Tenant anzulegen. Mit der Hilfe von Azure Management Groups ist es möglich mehrere Subscriptions in einen logischen Container zu packen und diese mittels Policies zu verwalten. Hier könnte man, wie in der Abbildung zu sehen ist, mehrere Management Groups anlegen und diese, gemeinsam mit den Subscriptions, in eine Root Management Group packen.

Auf dieser Root Management Group kann man eine Policy definieren, die die Auswahl der Regionen beim Erstellen eine virtuelle Maschine begrenzt (z.B.: nur EU-West und EU-Nord werden zugelassen). Die darunterliegenden Objekte im logischen Container bekommen diese Policy vererbt. Policies können auch für einzelne Sub-Management Groups definiert werden.

**Wichtige Eigenschaften von Azure Management Groups:**

* 10.000 Management Groups max. werden in einem einzigen Directory unterstützt.
* Ein Management Groups Tree kann bis zu sechs "levels of depth" unterstützen.
* Dieses Limit beinhaltet weder das Root-Level noch die Subscription-Ebene.
* Jede Management Group und jede Subscription kann nur eine Parent unterstützen.
* Jede Management Group kann mehrere Child-Items (Management Groups, Subscription) haben.
* Alle Abonnements und Management Groups befinden sich innerhalb einer einzigen Hierarchie (Root) in jedem Verzeichnis.
* Root Management Group kann nicht verschoben werden im Gegensatz zu alle darunter liegenden Subscriptions oder (Sub-) Management Groups.

Machine generated alternative text:
Human Resources 
Root Management Group 
Marketing 
EA Subscription 
EA Subscription 
Dev/Test 
Subscription 
Apps 
Production 
Free Trial 
Subscription 
Free Trial 
Subscription 
EA Subscription 
Geo Region 1 
EA Subscription 
Geo Region 2 
EA Subscription 
EA Subscription 
EA Subscription 

Abbildung 3 - Azure Management Group

## Resourcenverwaltung

Der Azure Resource Manager (ARM) ist die neueste Methode um Resourcen in Azure be-reitzustellen und zu gruppieren. Zusätzlich können die erstellten Resourcen nach der Bereitstellung mithilfe von Sicherheits-, Überwachungs- und Kennzeichnungsfunktionen verwaltet werden.

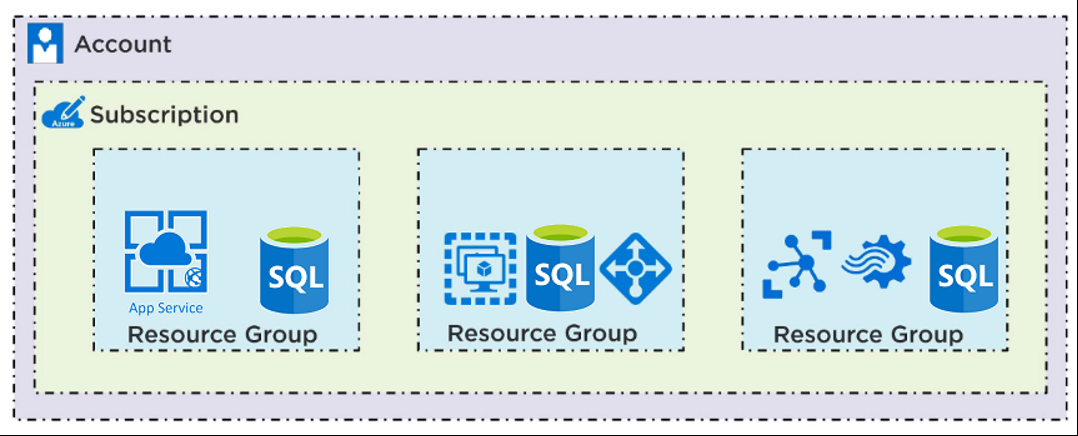


Abbildung 4 - Azure Subscription Struktur

Üblicherweise kommen mehrere Resourcen und Services bei Azure zum Einsatz. Einige Resourcen sind abhängig voneinander und werden somit als logische Gruppe betrachtet. Um diese logischen Gruppierungen in Azure ersichtlich zu machen, werden Resourcen-gruppen verwendet. Ein typisches Szenario wäre die Erstellung einer virtuellen Maschine die aus mehreren Resourcen besteht, genauer gesagt aus: virtuellen Netzwerken, Speicherkonten, Netzwerkschnittstellen, Netzwerksicherheitsgruppen, öffentliche oder statischen IP-Adressen, Datenträger und der virtuelle Computer. Mit ARM werden diese Res-sourcen in einer Resourcengruppe zusammengefasst und können dadurch leichter verwaltet und geändert werden.

## Studentenangebote – Azure for Students

Bei Azure for Student wird man für die ersten 12 Monate mit $100 Azure-Guthaben ausgestattet und erhält kostenlosen Zugriff auf ausgewählte Dienste (Anmerkung: Änderungen vorbehalten. Eine aktuelle Liste aller Dienste findet man in den **FAQ**: <https://azure.microsoft.com/de-de/free/free-account-students-faq/>).

### Voraussetzungen:

* Mitglied einer zu einem akademischen Abschluss führenden Bildungseinrichtung (im Bereich MINT)
* 18 Jahre alt
* Angebot nicht gültig für einen Massenonlinekurs
* Sofern nicht von Microsoft anders formuliert kann dieses Azure for Students nicht mit anderen Angeboten verknüpft werden

### Ablauf des Azure-Students Angebots:

Wenn das verfügbare Guthaben aufgebraucht oder der Zeitraum von 12 Monaten abgelaufen ist, wird das Azure-Abonnement deaktiviert. Wenn die 12 Monate abgelaufen sind und man weiterhin studiert, kann man das Azure for Students-Angebot erneuern.

Wichtig: Wenn man sich am Ende der 12-Monats-Frist oder nach Aufbrauchen des Guthabens in Höhe von $100 (je nachdem, welches Ereignis eher eintritt) gegen ein Upgrade entscheidet, werden alle von bereitgestellten Produkte eingestellt, und man kann nicht mehr darauf zugreifen. Man hat jedoch 90 Tage Zeit um ein Upgrade für das Abonnement durchführen damit man auf die zuvor erstellten Resourcen zugreifen kann.

## Kostenmanagement

Basierend auf nutzungsbasierenden Zahlungsmodellen werden die einzelnen Resourcen in Azure unterschiedlich verrechnet. Alle Kosten werden nach dem “Pay-As-You-Go” Prinzip errechnet, damit man als Benutzer nur das zahlt was man auch verwendet. Grundsätzlich erfolgt die Zahlung über eine Kredit-Karte oder über ein Guthaben welches man durch Angebote (sowie Azure for Students) bekommt.

Das Zahlungsmodell kann sich aber von Resource zu Resource ändern. Ein Beispiel wäre, dass bei einer virtuellen Maschine die Kosten durch die Laufzeit entstehen.

Hingegen bei einem Storage bestimmen unter anderem die Faktoren Speicherkapazität, Kontotyp, Replikationsschema, Speichertransaktionen und Datenausgang die Kosten. Hinzu kommt, dass bei allen Resourcen die Region einen Einfluss auf den Preis nimmt.

Microsoft bietet für Resourcen in ihrer Cloud einen Preisrechner an, mit dem die geschätzten Kosten einer Resource berechnet werden können. Der Preisrechner ist unter der URL <https://azure.microsoft.com/de-de/pricing/calculator/> aufrufbar.

Computergenerierter Alternativtext:
Virtual Machines 
Eu Westen 
Sparen Sie mit dem Azwe•Hybridvorteil bis zu 40 % für Windows Server. 
Standard 
(Nur Betriebssystem) 
B2MS: 2 RAM. 16 «stunde 
Abrechnungsoption 
Sparen Sie bis zu 72 % bei den die rutzungsbasierte Bezahlung mit für die Reservien'ng für 1 Oder 3 
Erfahren Sie mehr zu den Preisen reservierte '„'M-Instanzen. 
@ NutmgstmQte 
0 rævi«t 
75,56 € 

Abbildung 5 - Pricing Calculator Virtuelle Maschine

Der im Azure Hub zu findende Dienst Kostenverwaltung und Abrechnung ist das zentrale Blade in denen Kosten, Prognose, Zugriffssteuerung und vieles mehr, für ein Abonnement angezeigt wird. Über diesen Dienst können Sie sowohl auf zur Kostenübersicht als auch zur Ausgabequote und Prognose gelangen. Des Weiteren können Sie über die Zugriffssteuerung die Entscheidung treffen, welche Benutzer Zugriff auf die Kostenverwaltung für Ihr Abonnement haben. Neben den Zahlungsmethoden, können auch alle Resourcengruppen und Resourcen anzeigt werden.

Computergenerierter Alternativtext:
FH JOANNEUM Abonnement 
FH JOANNEuM 
mbh 
Kosten nach Ressource 
'0.05EUR 
lö.ö3EUR 
löö1EUR 

Abbildung 6 - Azure Cost Management Übersicht

Die Kostenverwaltung bietet eine Übersicht über vergangene und laufende Kosten. Zusätzlich werden eine Ausgabenquote und Prognose für den aktuellen Monat erstellt.

Alle Resourcen in einem Azure Abonnement werden gesammelt und ihre Kosten werden berechnet. Die Nachverfolgung der entstandenen Kosten und der Ausgaben wird ermöglicht. Die Kosten werden in Dashboardberichte angezeigt.

Computergenerierter Alternativtext:
ABO-Administrator 
'5.95EUR 
12.74WR 
li.73 
li2.81EUR 
125.84FUR 
RUN 
79.18EOR 
JOANNEUM 
25.84WR 
20 

Abbildung 7 - Azure Benutzerdefiniertes Dashboard

Azure Link: <https://docs.microsoft.com/de-de/azure/billing/billing-getting-started>

### Verschiedene Services um Kosten zu überwachen:

**Subscription Resource Cost:** Die Ansicht Azure Cost Analysis im Portal bietet einen schnellen Überblick über Kosten und Informationen über die täglichen Ausgaben nach Resource oder Resourcengruppe. Azure Cost Management: Dieses Produkt ist das Ergebnis des Erwerbs von Cloudyn durch Microsoft und ermöglicht es Azure-Ausgaben zu verwalten und zu analysieren, ebenso wie die Ausgaben für andere Public Cloud-Anbieter. Es gibt sowohl kostenlose als auch bezahlte Stufen.

**Azure Budgets and Action Groups:** Mit der Einführung von Azure Budgets und seinen APIs ist es nun möglich, Aktionen zu erstellen, wenn die Kosten einen Schwellenwert erreichen. Zum Beispiel das Herunterfahren einer "Test"-Resourcengruppe, wenn sie 100% ihres Budgets erreicht.

**Azure Advisor:** Azure Advisor gibt Empfehlungen zu Maßnahmen, um Geld zu sparen, die Zuverlässigkeit zu verbessern oder sogar die Sicherheit zu erhöhen.

<https://docs.microsoft.com/en-us/azure/architecture/cloud-adoption/appendix/azure-scaffold#cost-management>

## Docker in Azure

In diesem Abschnitt wird aufgezeigt welche Möglichkeiten es in Azure gibt Docker einzusetzen. Zusätzlich werden etwaige Einschränkungen aufgelistet und eine grobe Übersicht gegeben wie sich die Kosten zusammensetzen.

### Azure for Docker

Azure for Docker ist ein relativ neues Konzept und eines der jüngsten Services die es erlaubt Docker in Azure zu verwenden. Laut Azure und Docker Dokumentation handelt es sich hierbei um eine native-Docker Lösung. Im Endeffekt ist es aber nichts anderes als eine VM mit einem speziellen Kernel basierend auf einem Azure Template welches Docker im Swarm-Mode installiert. Diese Variante stellt alle Docker-Services zur Verfügung.

Unterstützte Docker Services:

 Docker Compose (*Support ohne Einschränkungen*)

 Docker Swarm

**Kosten**

Da Azure for Docker auf VMs aufbaut, müsste man den Azure Pricing Calculator (<https://azure.microsoft.com/en-us/pricing/calculator/>) verwenden, eine VM auszusuchen und in weitere Folge die Kosten ermitteln.

Zusätzlich wird als Storage bei Azure for Docker das Service Azure Files Storage verwendet welches bei Azure for Docker als Cloudstor bezeichnet wird. Cloudstor ist ein modernes Volume Plugin von Docker. Es wird vorinstalliert und vorkonfiguriert in Docker Swarms geliefert, die auf Docker for Azure bereitgestellt werden. Aufgaben im Docker-Schwarmmodus und normale Docker-Container können ein mit Cloudstore erstelltes Volume verwenden, um ein dauerhaftes Datenvolumen einzubinden.

Machine generated alternative text:
Data Storage prices 
Below are prices for storing data in Azure File share, shown as monthly charges per 
Gig of data stored. These prices vary based on the redundancy option that you 
select. 
GiB/month 
Snapshots GiB/month 
STANDARD 
€0.051 
€0.051 
per used Gig 
per used Gig 
PREMIUM (PREVIEW)" 
€0.112 per provisioned 
GiB 
€0.093 per used Gig 

Abbildung 8 - Data Storage Preise

Machine generated alternative text:
Operations and data transfer prices 
For th SMB and REST operations, transaction costs are incurred against your Azure 
File share, covering operations such as enumerating a directory or reading a file. 
These prices vary based on the redundancy option you select. 
Put, Create Container 
Operations (per 10,000) 
List Operations (per 10,000) 
All other operations except 
STANDARD 
€0.01 27 
€0.01 27 
€0.0013 
PREMIUM (PREVIEW) 
Delete, which is free (per 10,000) 
Geo -Replication Data Transfer N/A 
(per GiB) 

Abbildung 9 - Operations and Data Transfer Preise

### Azure Container Instances

Azure Container Instances ist eine gute Variante um einen Container in Azure zu deployen. Jedoch zielt es sehr stark auf isolierte Container ab und stellt somit keine Funktionen wie Orchestrierung, Service Discovery über mehrere Container oder automatisches Skalieren. Die einzige Möglichkeit mehrere Container zu instanziieren, ist die Verwendung einer Azure Container Group. Die Container werden in einer Azure Container Group zusammengefasst und sind über einen zentralen FQDN bzw. eine zentrale IP erreichbar. Container Group werden in einer VM gehostet.

Azure Container Instances dient dazu Docker Container Images zu deployen und zu hosten. Man muss das Image zuerst lokal builden, dann das Image in Azure Container Registry hochladen und dann den Container in Azure Container Instances deployen.

 Docker Compose

 Docker Swarm

Die einzige Möglichkeit mehrere Docker Container zu deployen ist über ein Resource Manager Template oder ein YAML File.

[https://docs.microsoft.com/en-us/azure/container-instances/container-instances-container-groups](https://docs.microsoft.com/en-us/azure/container-instances/container-instances-container-groups%20/) / <https://docs.microsoft.com/en-us/azure/container-instances/container-instances-multi-container-yaml>

**Kosten**

Machine generated alternative text:
Preisübersicht 
Azure Container Instances bill at the "'container group" level which are 
assignments of vCPU/Memory resources that can be used by a single 
container or split by multiple containers. Container groups are co- 
scheduled containers that share the same network and node lifecycle. 
The price depends on the number of vCPU and GBs of memory 
allocated to the container group. You are charged for each Ga and 
vCPlJ second your container group consumes. There is an additional 
charge of €0,0000102 per vCPLJ second for Windows software duration 
on Windows container groups. 
Read more about container groups > 
TAKTUNG 
Containergruppendauer 
GPI_J Container group 
duration 
PRE'S 
Arbeitsspeicher: €0,0000012 pro Gd- 
Sekunde 
vCPU: €0,00001 10 pro vCPU-Sekunde 
Arbeitsspeicher: €0,00000130 pro Ga- 
Sekunde 
vCPU: €0,0000097 pro vCPU-Sekunde 
1<80 vGPU: €0,0001097 pro vGPU-s 
PIOO vGPU: €0,0002674 pro vGPU-s 
VIOO vGPU: €0,0008686 pro vGPU-s 

Abbildung 10 – Preisübersicht

Machine generated alternative text:
TYPE 
Dynamic IP address 
Static IP address 
(reservation + usage) 
Public IP prefix: 
BASIC (ARM) 
€0.0034/hour 
First 5: €0.0034/hour2 
Additional: €0.0068/hour 
N/A 

Abbildung 11 - Pricing für IP Adressen

**Preisbeispiel**: Sie erstellen eine Linux-Containergruppe mit 1 vCPU und 1 GB, die während eines Monats (30 Tage) einmal pro Tag verwendet wird. Die Ausführungsdauer jeder Containergruppe beträgt 5 Minuten (300 Sekunden).

**Speicherdauer**: Anzahl der Containergruppen x Speicherdauer (Sekunden) x GB x Preis pro GB-Sekunde x Anzahl der Tage

|  |
| --- |
| 1 Containergruppe x 300 Sekunden x 1 GB x €0,0000012 pro GB-s x 30 Tage = €0,011 |

**vCPU-Dauer**: Anzahl der Containergruppen x vCPU-Dauer (Sekunden) x vCPU(s) x Preis pro vCPU-Sekunde x Anzahl der Tage

|  |
| --- |
| 1 Containergruppen x 300 Sekunden x 1 vCPU x €0,0000110 pro vCPU-Sekunde x 30 Tage = €0,099 |

**Gesamtabrechnung**: Speicherdauer (Sekunden) + vCPU-Dauer (Sekunden) = Gesamtkosten

|  |
| --- |
| €0,011 + €0,099 = €0,110 |

### Azure Service Fabric

Platform-based in einem Cluster. Nodes im Cluster sind entweder Linux oder Windows VMs. Standardmäßig stellt Service Fabric Dienste als Prozesse bereit und aktiviert sie. Service Fabric kann auch Dienste in Container-Images bereitstellen. Service Fabric unterstützt Docker-Container auf Linux- und Windows-Server-Containern.

 Docker Compose (Limited Support)

 Docker Swarm

Docker-Compose (Version 3 or later) limited Support: Application upgrades sind nicht möglich. Man kann aber Applications beenden, löschen und neu deployen.

Basic Service-Functions die unterstützt werden:

* Services > Deploy > Replicas
* Services > Deploy > Placement > Constraints
* Services > Deploy > Resources > Limits
* -cpu-shares
* -memory
* -memory-swap
* Services > Commands
* Services > Environment
* Services > Ports
* Services > Image
* Services > Isolation (only for Windows)
* Services > Logging > Driver
* Services > Logging > Driver > Options
* Volume & Deploy > Volume

**Kosten:**

Machine generated alternative text:
UNIT 
Container Compute Duration 
Software Duration 
PRICE 
€o.oonoo€/vcpu-s 
€0.000002/Memory Ga-s 
€0.000006/vCPU-S Machine generated alternative text:
Reliable Volumes 
UNIT 
Disk Size 
Price per unit per month 
lops per disk 
Throughput per disk 
PRICE 
32 Gid 
€2.771 
240 
50 MB,'Second Machine generated alternative text:
Reliable Collections 
UNIT 
Service Fabric reliable dictionary and 
queue 
Provisioned lops (per 100 lops, 400 
IOPS minimum) 
PRICE 
€0.127/GB/month 
€0.005/hour 

Abbildung 12 - Service Fabric Kostenübersicht

### Azure App Service

Platform-based Service auf Linux um Application Stacks zu hosten. Azure App Service bietet integrierte Docker-Images unter Linux mit Unterstützung für bestimmte Versionen wie PHP 7.0 und Node.js 4.5. Azure App Service verwendet die Docker-Container-Technologie, um sowohl Built-in Images als auch Custom Images as a Platform als Service bereitzustellen. Docker Images werden in Azure Container Registry (private container) zentral abgelegt. ARC ist ein verwalteter Docker-Registry-Dienst, der auf der Open-Source-Plattform Docker Registry 2.0 basiert.

Die Verwendung von ACR für Containereinsätze ermöglicht es, benutzerdefinierte Container-Images privat in der Azure-Cloud zu speichern. (<https://mikepfeiffer.io/azure-docker-containers.html>)

 Docker Compose

 Docker Swarm

**Kosten:**

Machine generated alternative text:
BASIC 
Dedizierte 
Umgebung 
für das 
STANDARD 
PREMIUM 
Produktionsworkb*terte 
ausführen 
Entwickeln/Testen 
Web-, mobile- Oder 
Apl-Apps 
Festplattenspeicher 
Maximale Instanzen 
Benutzerdefinierte 
Domäne 
Automatisch skalieren 
VPN 
Hybridkonnektivität 
Netzwerkisolation 
Preis pro Stunde 
Unbegrenzt 
10 
Bis zu 3 
Unterstützt 
€0,045 
Unbegrenzt 
50 
Bis zu 10 
Unterstützt 
Unterstützt 
Unterstützt 
€0,081 
Leistung und 
Skalierung 
Unbegrenzt 
250 Ga 
Bis zu 20 
Unterstützt 
Unterstützt 
Unterstützt 
€0, 150 Machine generated alternative text:
Serviceplan Basic 
Der Serviceplan Basic wurde für Apps mit geringeren Anforderungen an den 
Datenverkehr entwickelt, die zudem keine erweiterten Features für automatische 
Skalierung und Datenverkehrsverwaltung benötigen. Die Preise basieren auf der 
Größe und Anzahl der ausgeführten Instanzen. Durch die integrierte 
Unterstützung für Netzwerklastenausgleich wird der Datenverkehr automatisch 
auf Instanzen verteilt. Der Sewiceplan Basic unterstützt in Linux- 
Runtimeumgebungen Web-App für Container. 
INSTANZ 
83 
KERNE 
2 
4 
RAM 
1,75 
3,50 
STORAGE 
10 
10 
10 

Abbildung 13 - App Service Kostenübersicht

### Azure Virtual Machines

Docker EE direkt auf einer Azure Virtual Machine (Linux) installieren. (<https://docs.microsoft.com/en-us/azure/virtual-machines/linux/docker-compose-quickstart>)

 Docker Compose

 Docker Swarm

**Kosten:**

Da Azure for Docker auf VMs aufbaut, müsste man den Azure Pricing Calculator (<https://azure.microsoft.com/en-us/pricing/calculator/>) verwenden, eine VM auszusuchen und in weitere Folge die Kosten ermitteln.

# Amazon Web Services

## Allgemeines

In diesem Kapitel finden sie die allgemeinen Informationen über AWS. Der Inhalt bezieht sich auf das Hosting, Resourcen und wie komme ich zu meiner privaten AWS.

## Struktur

AWS stellt zuverlässige, skalierbare und kosteneffektive RechenResourcen für das Hosten von Anwendungen zur Verfügung. Es gibt verschiedene AWS-Komponenten, welche eigenständig oder auch in einer Kombination von Lösungen verwendet werden können:

**Server**

Amazon Elastic Compute Cloud (EC2) bietet skalierbare Rechenkapazität in der Cloud. Dabei wird eine virtuelle EC2 Umgebung, mit dem für die zu hostende Anwendung erforderlichen Betriebssystem, Services, Datenbanken und Anwendungsplattform-Stack definiert. Dabei wird auch eine Management-Konsole sowie API’s zur Verwaltung der RechenResourcen bereitgestellt. Detailliere Informationen sind dabei zu finden unter (<https://aws.amazon.com/de/ec2/>)

**Speicher**

Amazon Simple Storage Service (Amazon S3) ist stellt eine Web-Service-Schnittstelle zum Abrufen und Speichern einer beliebigen Datenmenge zu jeder Zeit und von jedem Ort im Internet zur Verfügung. Dabei wird entsprechend auf Beständigkeit, Hochverfügbarkeit und Sicherheit geachtet. Dabei speichert Amazon S3 unter anderem auch mehrere redundante Kopien der Daten. Weiterführende Informationen unter (<https://aws.amazon.com/de/s3/>).

**Datenbank**

Amazon Relational Database Service (Amazon RDS) vereinfacht die Einrichtung, Verwaltung und Skalierung von relationalen Datenbanken in der Cloud. Die Datenbankkapazität kann hier stetig und kosteneffizient angepasst werden und es werden Administrationsaufgaben vereinfacht. Amazon RDS ist für verschiedene Datenbank-Interfaces-Typen – optimiert für Arbeitsspeicher, Commons oder E/A - verfügbar und bietet sechs Datenbank-Engines. Dazu zählen Amazon Aurora, PostgreSQL, MySQL, MariaDB, Oracle Database und Microsoft SQL Server. Weiter gibt es einen AWS Database Migration Service, welcher genutzt werden kann um bestehende Datenbanken zu migireren oder zu replizieren. Weitere Informationen zu Amazon RDS sind zu finden unter (<https://aws.amazon.com/de/rds/>).

**Netzwerk**

Amazon CloudFront stellt ein global verteiltes, extrem leistungsfähiges System für die Bereitstellung von Inhalten bereit. Über Amazon CloudFront kann die Anwendung Inhalte an verschiedene Benutzer verteilen oder streamen – mit Integrationsmöglichkeiten zu Amazon S3. Weitere Informationen siehe (<https://aws.amazon.com/de/cloudfront/>).

**Entwicklung**

Amazon Simple Queue Service (Amazon SQS) ist ein Nachrichtenwarteschlangen-Service, der das Entkoppeln und Skalieren von Microservices, verteilten Systemen und serverlosen Anwendungen ermöglicht. Amazon SQS fokussiert sich dabei auf die Middleware, um Nachrichten zwischen Softwarekomponenten in jeder Größe und Menge ohne Verlust von Nachrichten und ohne weitere Services senden, speichern und empfangen. Weiter gibt es zwei Arten von Nachrichtenwarteschlangen: „Standardwarteschlangen“ mit einem maximalen Durchsatz, Sortierung und mindestens eine Zustellung der Nachricht. Die zweite Art ist die „SQS-FIFO-Warteschlange“, diese gewährleistet, dass die Nachrichten genau ein Mail in genau der Reihenfolge, in der sie ankommen verarbeitet werden. Weitere Informationen unter (<https://aws.amazon.com/de/sqs/>)

**Kontoverwaltung**

Amazon DevPay ist einfach Online-Rechnungs- und Kontoverwaltungsservice, welcher es ermöglicht Anwendungen zu verkaufen, die in AWS integriert sind oder auf diesen laufen. Dabei können beispielsweise auch die AWS-Dienste automatisch gemessen werden und dies anschließend in Rechnung zu stellen. Für die Zahlungsabwicklung wird Amazon Payments verwendet, damit kann mit bestehenden Amazon-Konten bezahlt werden. Weitere Details: (<https://aws.amazon.com/de/devpay/>)

**Verzeichnis**

Mit AWS Identity and Access Management (IAM) kann der Zugriff auf AWS Services und Resourcen sicher verwaltet werden. Mithilfe von IAM können Sie AWS-Benutzer und -Gruppen anlegen und verwalten und mittels Berechtigungen ihren Zugriff auf AWS-Resourcen zulassen oder verweigern.

IAM kann genutzt werden, um Ihren Mitarbeitern und Anwendungen Verbundzugriff auf die AWS Management Console und AWS-Service-APIs zu ermöglichen, wobei Ihre bestehenden Identitätssysteme wie Microsoft Active Directory genutzt werden. Sie können jede Lösung für das Identitätsmanagement verwenden, die SAML 2.0 unterstützt.

## Resourcenverwaltung

In AWS können die einzelnen Services in Resource Groups organisiert werden. Damit ist es möglich mehrere Resourcen zu gruppieren und gemeinsam zu managen. Somit können Updates, Upgrades, Port Management und Monitoring auf ganze Resource Groups angewandt werden. Außerdem kann man auf einzelne Resource Groups Permissions setzen und so den Zugriff steuern.

<https://docs.aws.amazon.com/ARG/latest/userguide/welcome.html>

## Studentenangebote

Für AWS gibt es ein eigenes Studentenportal: AWS Educate. Wobei der AWS Educate Zugang kann sehr missverständlich interpretiert werden. AWS Educate ist ein AWS Zugang, der vorwiegend genutzt wird, um einzelne Module zu erlernen. Es gibt ein System mit Abzeichen, die man für jedes bestandene Modul bekommt. Daher hat man mit AWS Educate keinen direkten Cloud Zugang. Jedoch ist es eine gute Möglichkeit, um den Umgang mit der AWS Cloud zu erlernen.

Für AWS selber kann man sich zwar kostenlos registrieren, es müssen aber sehr viele Daten angegeben werden, unter anderem auch die Kreditkartendaten. Aus diesem Grund wurde auf eine sehr detaillierte Untersuchung verzichtet und nur öffentlich zugängliche Informationen für die Bewertung verwendet.

Für den Testversuche mit AWS, kann ein 12-monatige kostenlose Trail verwendet werden. Der Testbereich ist nur für neue AWS-Kunden verfügbar und ist für 12 Monate ab dem Datum Ihrer AWS-Anmeldung verfügbar. Die anderen Angebote stehen sowohl bestehenden als auch neuen AWS-Kunden zur Verfügung und können in ihrer Dauer (z.B. für Testversionen) oder in ihrer kostenlosen Nutzung (z.B. die Höhe des kostenlosen Speicherplatzes für ein Datenbankangebot) begrenzt sein. Sie haben keinen Anspruch auf Angebote, wenn Sie oder Ihre Organisation mehr als ein Konto erstellen, um zusätzliche Leistungen im Rahmen der Angebote zu erhalten. Eine Organisation (unter AWS-Organisationen) kann nur von Angeboten von einem Konto in der Organisation profitieren, und um die Nutzung der AWS-Dienste durch die Organisation unter allen Angeboten zu berechnen, werden wir die Nutzung über alle Konten in der Organisation aggregieren. Für die Nutzung der AWS Services werden Ihnen Standardtarife berechnet, wenn wir feststellen, dass Sie keinen Anspruch auf ein Angebot haben.

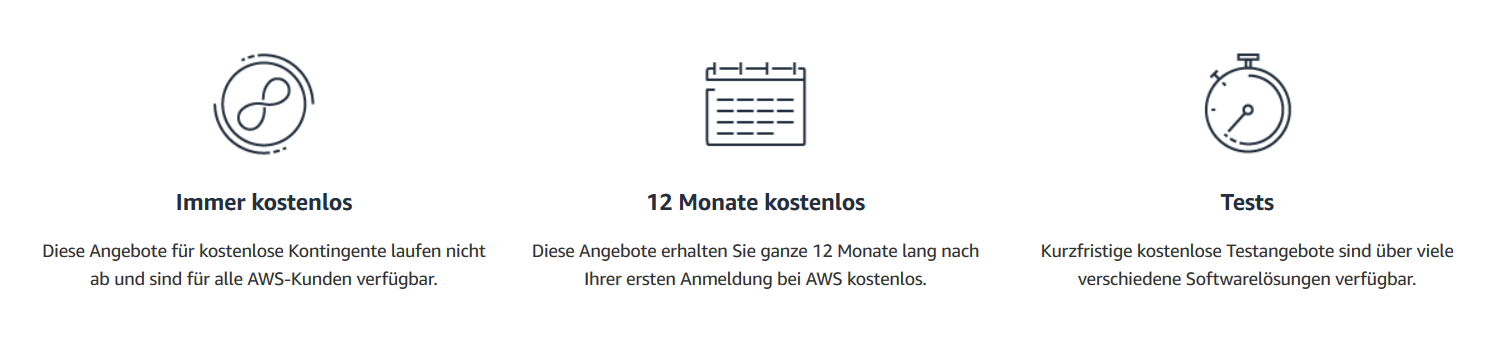


Abbildung 14: AWS Angebote

## Kostenmanagement

Durch AWS-Budgets können Sie benutzerdefinierte Budgets festlegen und sich benachrichtigen lassen, sobald Ihre Kosten oder Ihre Nutzung den veranschlagten Betrag überschreiten. ie AWS-Budgets können auch zum Festlegen von RI-Auslastungs- oder Nutzungszielen einsetzen und Benachrichtigungen erhalten, wenn Ihre Auslastung unter den von Ihnen definierten Schwellenwert fällt. Die Benachrichtigungen werden in Amazon EC2, Amazon RDS, Amazon Redshift und Amazon ElastiCache unterstützt.

Die Budgets können monatlich, vierteljährlich oder jährlich im Dashboard nachverfolgt und es können Start- und Enddatum selbst festgelegt werden. Sie können Ihr Budget noch genauer definieren und die Kosten mit mehreren Dimensionen wie AWS Service, verknüpften Konten, Tags und mehr nachverfolgen. Der Versand von Budgetalarmen ist per E-Mail und/oder Amazon Simple Notification Service-Thema (SNS) möglich.

<https://aws.amazon.com/de/aws-cost-management/aws-budgets/>

Detaillierte Informationen über die aktuellen Kosten der Nutzung können auf der AWS Billing Console bzw. Dashboard eingesehen werden. Dabei können die Kosten monatlich verfolgt werden, dabei werden die kosten pro AWS-Dienst aufgelistet. Zusätzlich kann noch ein Report in CSV oder Excel Format heruntergeladen werden. Auch kann der Kosten- und Nutzungsbericht auch in Amazon Athena, Amazon QuickSight oder andere Tools geladen werden. Mittels der Aktivierung der Ressoruce-Tags in der AWS Billing Console kann die Verrechnung auch entlang der Organisationsstruktur erfolgen. Beispielsweise kann entsprechende die Test- und Entwicklungsumgebung getaggt werden. Mit der Aktivierung der Tags erscheint dieser auch im sogenannten Cost-Explorer und man kann danach gruppieren und filtern. Des Weiteren erscheinen diese auch im bereits zuvor erwähnten AWS Cost & Usage Report.

Die oben erwähnten monatlichen Rechnungen listen dann die Kosten auf, wobei weiter aufgeschlüsselt werden kann nach Regionen (laut AWS) und Konto. Weitergehende Analysen bezüglich Kosten und Trends wie ad-hoc Analysen oder weiter Filterungen und Sortierungen können mit dem Cost Explorer vorgenommen werden, vergleiche auch nachfolgende Abbildung.

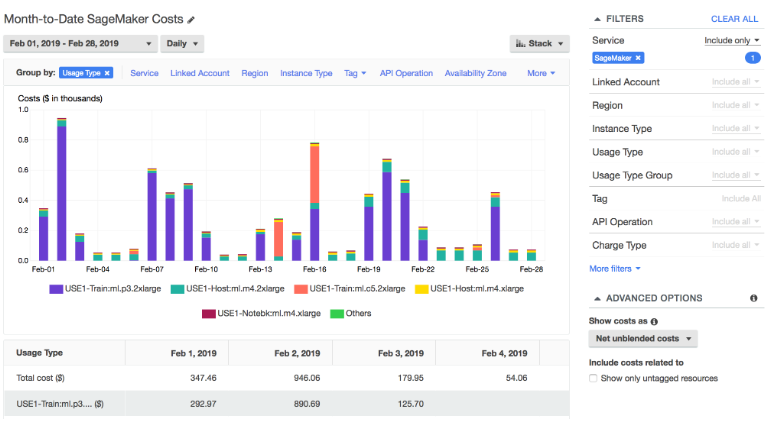


Abbildung 15: AWS Kostenmanagement

Als weiterer Service für das Kostenmanagement gibt es den bereits kurz angesprochenen Service AWS Budget. Mittels dieses Service können pro Service definierte Kosten- und Nutzungsbudget festgelegt werden. Werden die gesetzten Grenzen überschritten können Alerts verschickt werden. Jedes Budget kann 5 Nachrichten definieren, welche an bis zu 10 Email-Adressen versendet werden. Es gibt zudem auch noch die Möglichkeit die Überschreitung als eine Amazon Simple Notification Service (SNS) Thema zu veröffentlichen.

In AWS kann mittels eines Kostenkalkulators (<https://awstcocalculator.com/>) die genauen Kosten für die entsprechenden Resourcen ermittelt werden.

## Docker

Um Docker Anwendungen in AWS nutzen zu können wird der Amazon Elastic Container Service (Amazon ECS) verwendet. Weiters erleichtern Services wie AWS Fargate, Amazon EKS und AWS Batch das Ausführen und Verwalten von Docker-Containern in großem Umfang.

Docker CE kann sowohl auf einem Computer als auch auf einer Instanz einer virtuellen Amazon EC2 Maschine installiert werden oder sofort mit der Amazon ECS-optimierten AMI verwendet werden.

Docker for AWS bietet eine native Docker-Lösung, die eine zusätzliche Komplexität des Betriebs vermeidet und dem Docker-Stack nicht benötigte APIs nicht hinzufügt. Es ermöglicht eine direkte Interaktion mit Docker (einschließlich nativer Docker-Orchestrierung).

Die zusätzliche Konsistenz zwischen den Clouds trägt auch dazu bei, dass eine Migration oder eine Strategie mit mehreren Clouds in Zukunft leichter durchgeführt werden kann, wenn dies gewünscht wird.

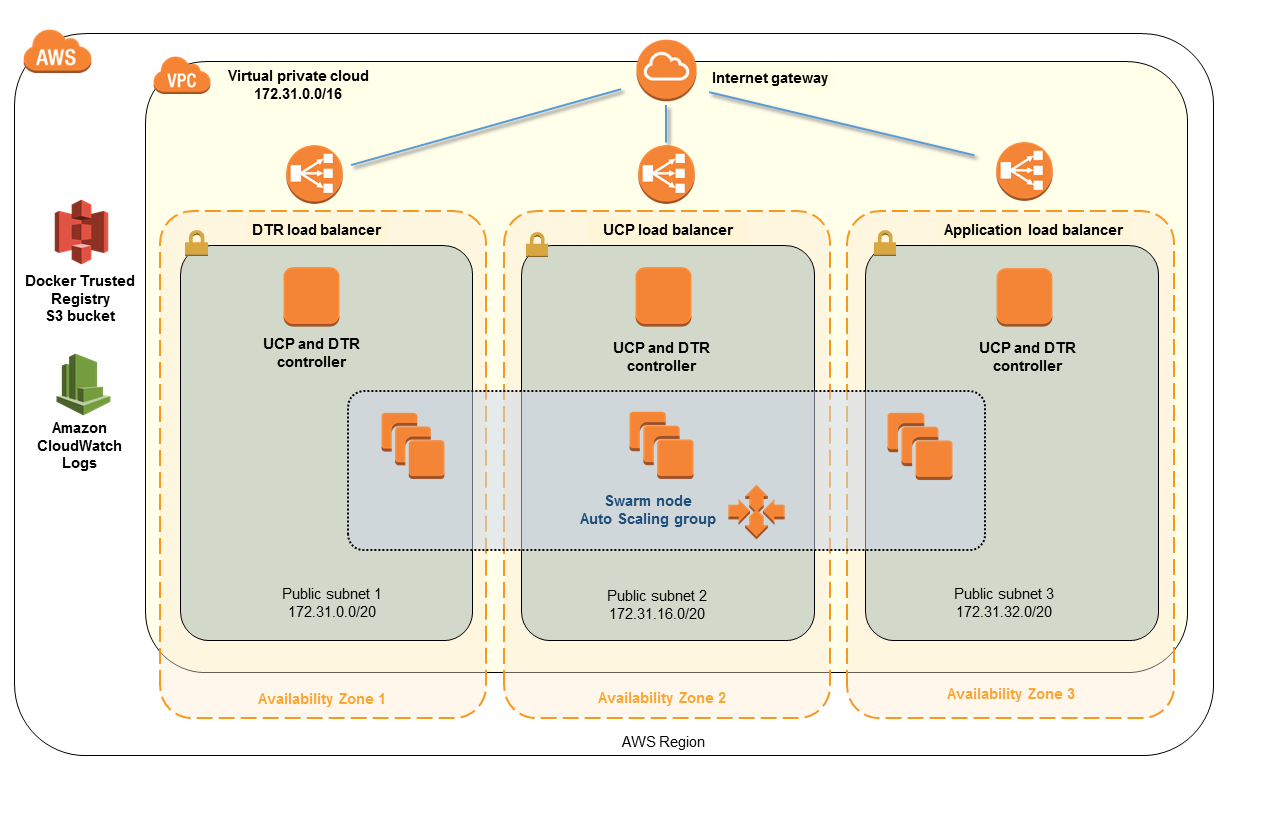


Abbildung 16 - AWS Docker

### Amazon ECS

Amazon ECS ist ein Service zur Container Orchestrierung, der Docker Container unterstützt. ECS ermöglicht einfache Ausführen und Skalieren von Anwendungen in Containern. Durch ECS wird keine andere eigene Orchestrierungssoftware benötigt. Außerdem erleichtert ECS das Verwalten von VM Clustern und das Planen von VMs. Mit Amazon ECS werden die Container in eigenen Amazon VPC gestartet, sodass Sie Ihre VPC-Sicherheitsgruppen und Netzwerk-ACLs nutzen können.

Die Abbildung unten zeigt die Funktionsweise an wie Amazon ECS ins System eingebunden wird.

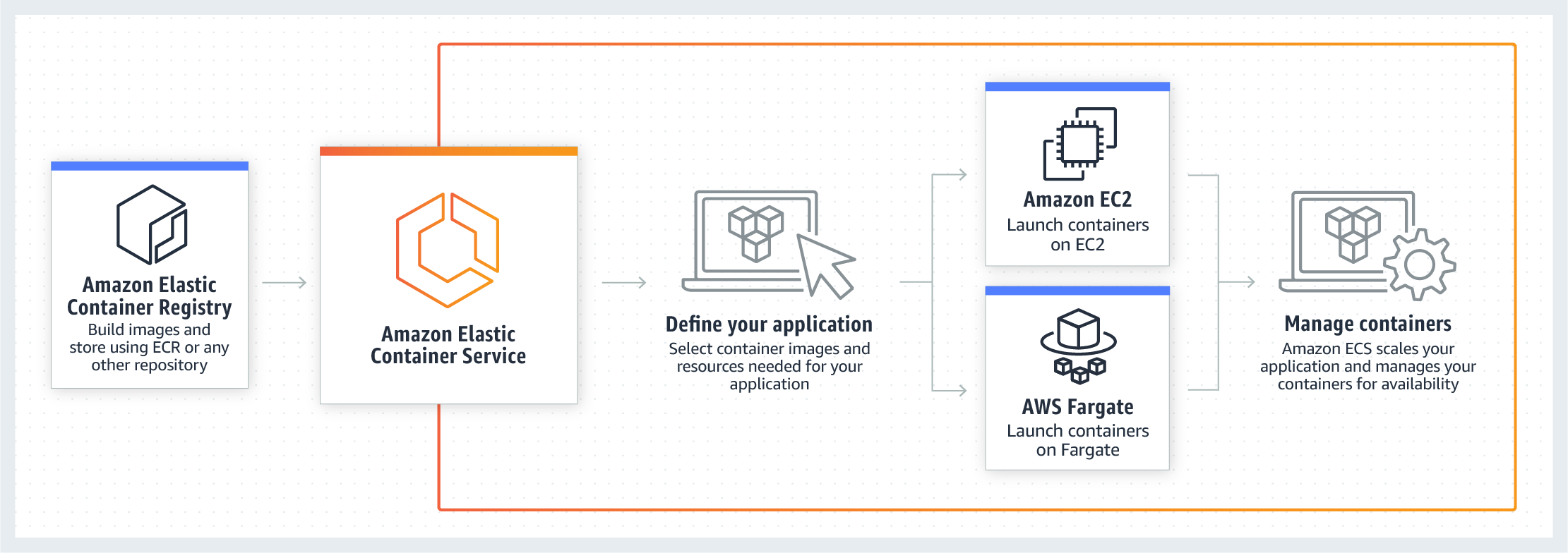


Abbildung 17: Funktionsweise Amazon ECS

Amazon ECS unterstützt Features wie IAM Rollen, Security Groups, Loadbalancer, Amazon CloudWatch Events, AWS CloudFormation templates und AWS CloudTrail logs.

#### Pricing

Für Amazon ECS gibt es zwei verschiedene Verrechnungsmodelle

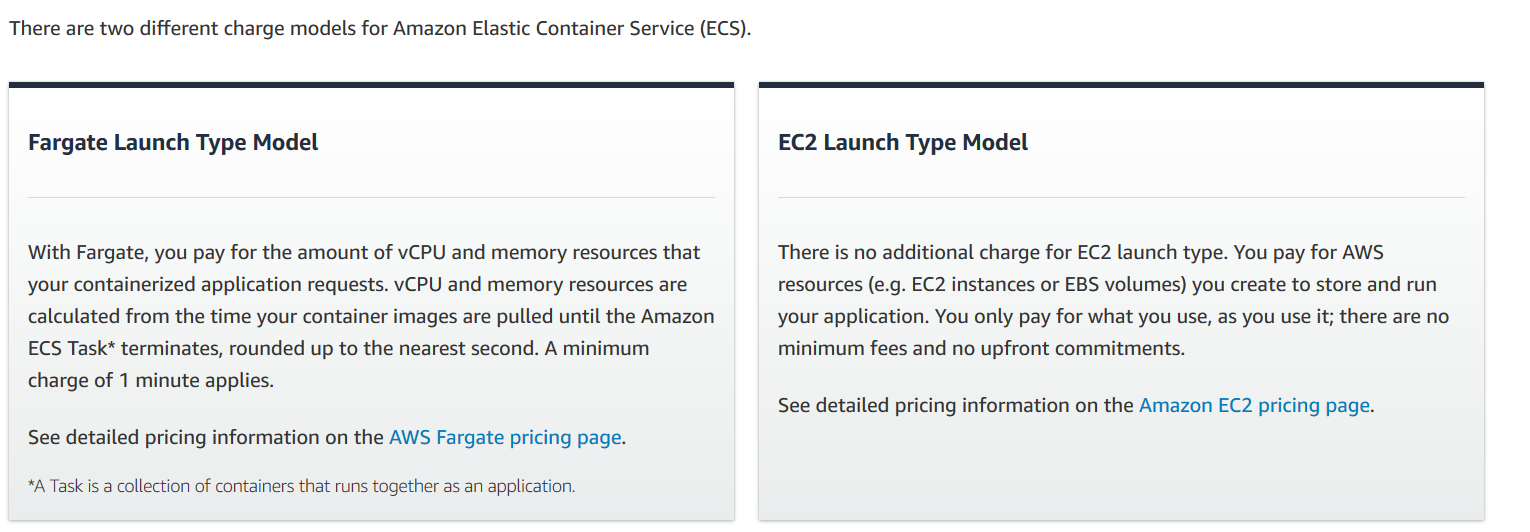


Abbildung 18 - Amazon ECS Preismodelle

### Amazon Fargate

AWS Fargate ist eine Datenverarbeitungs-Engine die es ermöglicht Container ohne Server oder Cluster verwalten zu können. Dadurch muss nicht mehr über die Planung und Interaktion mit Servern nachgedacht werden und es bleibt mehr Raum, um den Fokus auf das Design und die Implementierung von Anwendungen zu legen.

#### Pricing

Für AWS Fargate werden die Kosten aus der vCPU und den SpeicherResourcen, ab Start des Donwloads des Container Images, ermittelt und bis der ECS Task terminiert wird.

### Amazon EKS

Amazon Elastic Container Service for Kubernetes (Amazon EKS) ist es einfach, containerisierte Anwendungen mit Kubernetes unter AWS bereitzustellen, zu verwalten und zu skalieren.

Kubernetes ist eine Open Source-Software, die eine maßgeschneiderte Bereitstellung und Verwaltung von auf Containern ausgeführten Anwendungen ermöglicht. Kubernetes verwaltet Cluster, die aus Amazon EC2-Instances für die Datenverarbeitung bestehen, und führt für diese Instances [Container](https://aws.amazon.com/what-are-containers/) mit Prozessen zur Bereitstellung, Wartung und Skalierung aus. Mithilfe von Kubernetes können jede Art einer auf Containern ausgeführten Anwendung ausführen und dabei sowohl lokal als auch in der Cloud dasselbe Toolkit verwenden. Mit AWS ist die Ausführung von Kubernetes in der Cloud ganz einfach. Hierfür werden eine skalierbare und hochverfügbare Infrastruktur virtueller Maschinen, Integrationen von Services aus der Community sowie der als Kubernetes-konform zertifizierte verwaltete Kubernetes-Service [Amazon Elastic Container Service for Kubernetes (EKS)](https://aws.amazon.com/eks/) eingesetzt. Kubernetes verwaltet einen Cluster aus Instances zur Datenverarbeitung und übernimmt unter Berücksichtigung der verfügbaren RechenResourcen und des Resourcenbedarfs der einzelnen Container die entsprechende Planung der Container. Container werden in logischen Gruppierungen ausgeführt, die als "Pods" bezeichnet werden. Sie können einen, aber auch viele Container gemeinsam als Pod ausführen und skalieren. Die Software der Kubernetes-Steuerebene entscheidet, wann und wo Ihre Pods ausgeführt werden. Außerdem übernimmt sie das Routing des Datenverkehrs und skaliert Ihre Pods nutzungsabhängig oder auf der Grundlage von sonstigen Metriken, die Sie definieren. Kubernetes startet auf Basis des jeweiligen Resourcenbedarfs automatisch Pods in Ihrem Cluster und führt auch automatisch Neustarts von Pods aus, falls diese oder die Instances, in denen sie ausgeführt werden, ausfallen. Jeder Pod erhält eine IP-Adresse und einen individuellen DNS-Namen. Diese Angaben werden von Kubernetes für die Verbindung Ihrer Services untereinander und mit externem Datenverkehr verwendet.

#### Pricing

Für jeden erstellten Amazon EKS Cluster fallen $0,20 pro Stunde an. Es kann ein Amazon EKS-Cluster zum Ausführen mehrerer Anwendungen verwendet werden, indem Kubernetes-Namespaces und IAM-Sicherheitsrichtlinien genutzt werden.

Es fallen Kosten für die AWS Resourcen (z.B. EC2 Instanzen oder EBS Volumes) an, die zum Ausführen der Worker Knoten von Kubernetes erstellt werden. Es wird nur das bezahlt, was verwendet wird, während der Verwendung. Es gibt keine Mindestgebühren und keine Vorabverpflichtungen.

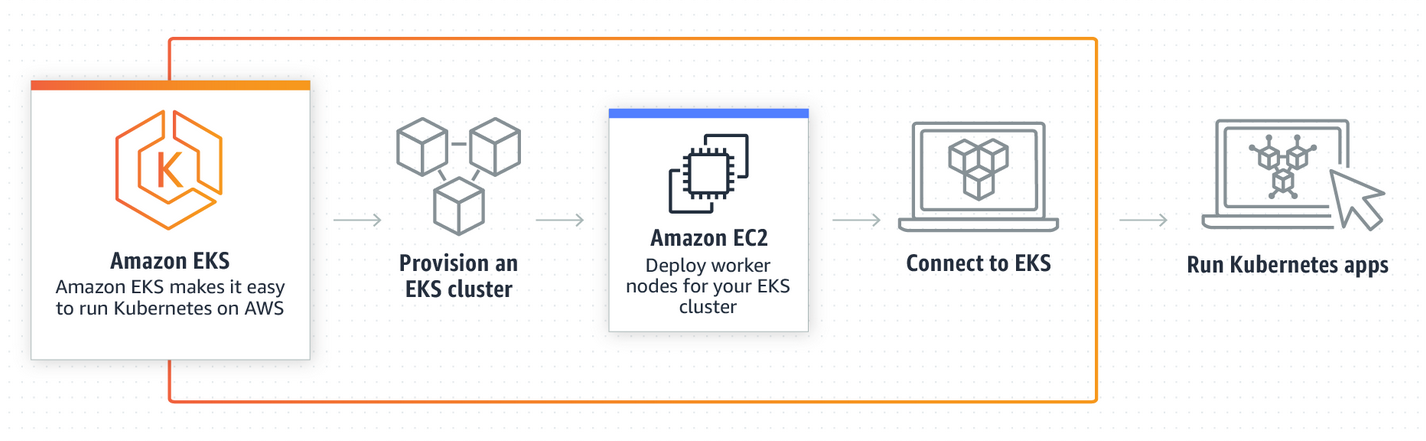


Abbildung 19 - Amazon EKS

### AWS Batch

AWS Batch ermöglicht es effizient unzählige Batchdatenverarbeitungsaufträge durchzuführen. Es stellt dynamisch die optimale Menge geeigneter RechenResourcen bereit (z. B. CPU- oder RAM-optimierte Instanzen).

Die Bereitstellung basiert auf den Volumen- und Resourcenanforderungen der eingereichten Batch-Aufträge. AWS Batch erfordert zur Ausführung von Aufträgen keine Installation oder Verwaltung einer Batchverarbeitungssoftware und keine Servercluster. Zur Vorbereitung, zeitlichen Planung und Ausführung der Batch-Verarbeitungslasten bedient sich AWS Batch des gesamten Spektrums an AWS-Rechenservices und -funktionen, unter anderem auch Amazon EC2 und Spot-Instances.

#### Pricing

Für die Nutzung von AWS Batch fallen keine zusätzlichen Gebühren an. Es muss nur für die AWS-Resourcen (z. B. EC2-Instances), die Sie zum Speichern und Ausführen der Batch-Aufträge erstellt werden, gezahlt werden.

### Amazon ECS CLI

[Amazon Elastic Container-Service](https://aws.amazon.com/de/ecs/) Command Line Interface (Amazon ECS CLI) unterstützt jetzt das Dateiformat Docker Compose Version 3 für die Bereitstellung von Docker-Containern in Amazon ECS. Die Amazon Elastic Container Service (Amazon ECS)-Befehlszeilenschnittstelle (Command Line Interface (CLI)) bietet allgemeine Befehle zum einfacheren Erstellen, Aktualisieren und Überwachen von Clustern und Aufgaben von einer lokalen Entwicklungsumgebung aus. Die Amazon ECS-CLI unterstützt [Docker Compose](https://docs.docker.com/compose/)-Dateien ([Version 1](https://docs.docker.com/compose/compose-file/compose-file-v1/), [Version 2](https://docs.docker.com/compose/compose-file/compose-file-v2/) und [Version 3](https://docs.docker.com/compose/compose-file/)), eine beliebte Open-Source-Spezifikation zur Definition und Ausführung von Multi-Container-Anwendungen. Verwenden Sie die CLI als Teil Ihres täglichen Entwicklungs- und Testzyklus als Alternative zur AWS Management Console.

### Docker Swarm

Für Docker Swarm wird kein eigener AWS Service benötigt. Der Swarm Mode wird unterstützt und kann in AWS konfiguriert werden. Eine detaillierte Anleitung dazu findet man unter <https://docs.docker.com/v17.12/docker-cloud/cloud-swarm/create-cloud-swarm-aws/>

## Pricing

Es gibt in der Linksammlung ein eigenes Tool, um die Software- und Infrastrukturkosten basierend auf Ihren Konfigurationsoptionen zu schätzen. Der Verbrauch und die Kosten können von dieser Schätzung abweichen. Es werden auch in Ihren monatlichen AWS-Abrechnungsberichten berücksichtigt. <https://aws.amazon.com/marketplace/pp/B06XCPWF94#pdp-pricing>

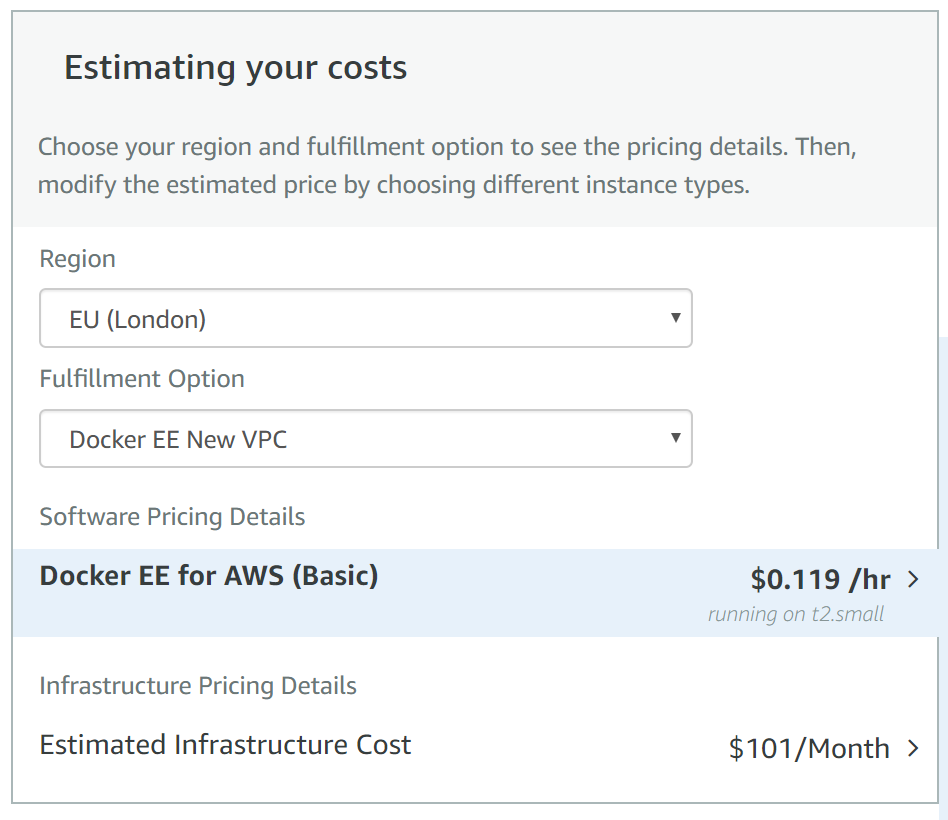


Abbildung 20 - AWS Preisrechner

# Google Cloud Platform

## Allgemeines

Google Cloud Platform (GCP) ist das Cloud-Computing-Dienst von Google, welches auf der gleichen Infrastruktur läuft, die auch intern für Services wie Google Search und YouTube eingesetzt wird. Neben einer Reihe von Management-Tools bietet sie eine Reihe von modularen Cloud-Diensten wie Computing, Datenspeicherung, Datenanalyse und Machine Learning. Die Registrierung erfordert eine Kreditkarte oder Bankverbindung.

Die Google Cloud Platform bietet Infrastruktur als Service (IaaS), Plattform als Service (PaaS) und serverlose Computerumgebungen. Letzteres steht im großen Fokus bei Google.

## Struktur

Grundsätzlich werden alle verrechneten Resourcen in Projekten gruppiert. Zusätzlich ist es auch wichtig zu wissen, dass GCP ihre Resourcen wie folgt strukturiert: **Global**, **Regional** und in **Zonen**. Ähnlich wie bei MS Azure sind Regionen geografisch auf der Ebene eines Subkontinents. Zonen sind im Wesentlichen Rechenzentren innerhalb von Regionen. Diese Unterscheidung ist wichtig, da regionale und zonale Resourcen von der Plattform oft unterschiedlich abgerechnet und behandelt werden.

Die GCP Architektur und ihr Modell unterteilt sich in unterschiedliche Layer. Dieser Aufbau lässt sich gut mit der Struktur von Azure vergleichen. Das Prinzip ist dasselbe nur unter anderen Namen anzufinden. Die unterschiedlichen Ebenen der GCP Hierarchie sind: Domain, Organisation, Ordner, Projekte und Resourcen. Des Weiteren unterscheidet man zwischen Rechnungskonto und Zahlungsprofil. Der Unterschied wird im Abschnitt 3.5 erläutert.

**Domain:**

Eine Domain in GCP ist die primäre Identität der Organisation und bestimmt die Identität des Unternehmens bei allen Google Diensten. Die Domainebene dient der Verwaltung von erstellten Organisationen in der Google Cloud. Auf dieser Ebene wird festgelegt welche Nutzer des Unternehmens auf Googles Cloud Platform zugriff haben.

**Organisation:**

Die OrganisationsResource stellt eine Organisation wie etwa ein Unternehmen dar und ist der Stammknoten in der GCP-Resourcenhierarchie. Die OrganisationsResource ist der hierarchische Ancestor von ProjektResourcen und Ordnern. Die auf die OrganisationsResource angewendeten IAM-Zugriffssteuerungsrichtlinien gelten innerhalb der gesamten Hierarchie für alle Resourcen der Organisation.

Die Projekte, die innerhalb einer Organisation erstellt werden, stehen im Besitz der Organisation und nicht des Mitarbeiters, auch wenn dieser das Projekt angelegt hat. Zusätzlich erlaubt diese Hierarchie Administratoren zentral alle ProjektResourcen zu steuern und zu verwalten. Vergleichbar mit einem Global Administrator bei Azure. Ein gängiges Konzept bei einer Organisation in GCP ist es Rollen auf Organisationebene zu vergeben.

Hier kann man zum Beispiel die Rolle „Netzwerkadministrator“ vergeben. Eine Identity mit dieser Rolle ist dann in der Lage alle netzwerkspezifischen Eigenschaften für alle Projekte zu steuern und zu verwalten. Dadurch ist eine Delegation von administrativen Tätigkeiten stark vereinfacht möglich.

**Ordner:**

Ordner ermöglichen es eine granulare und detailliertere Strukturierung von Projekten innerhalb der Organisation. Ordner können auch als Unterorganisation betrachtet werden. Ordner können verschiedene Rechtspersönlichkeiten, Abteilungen und Teams innerhalb eines Unternehmens darstellen. So lassen sich beispielsweise in einer ersten Ordnerebene die Hauptabteilungen der Organisation darstellen. Da Ordner Projekte und andere Ordner enthalten können, könnten in jedem Ordner wiederum weitere Unterordner für die verschiedenen Teams vorhanden sein.

Ordner werden stark für das Delegieren von Administrationsrechten eingesetzt um zum Beispiel einem Teamleiter volle Inhaberschaft aller Resourcen seiner Abteilung (Ordner) zu geben.

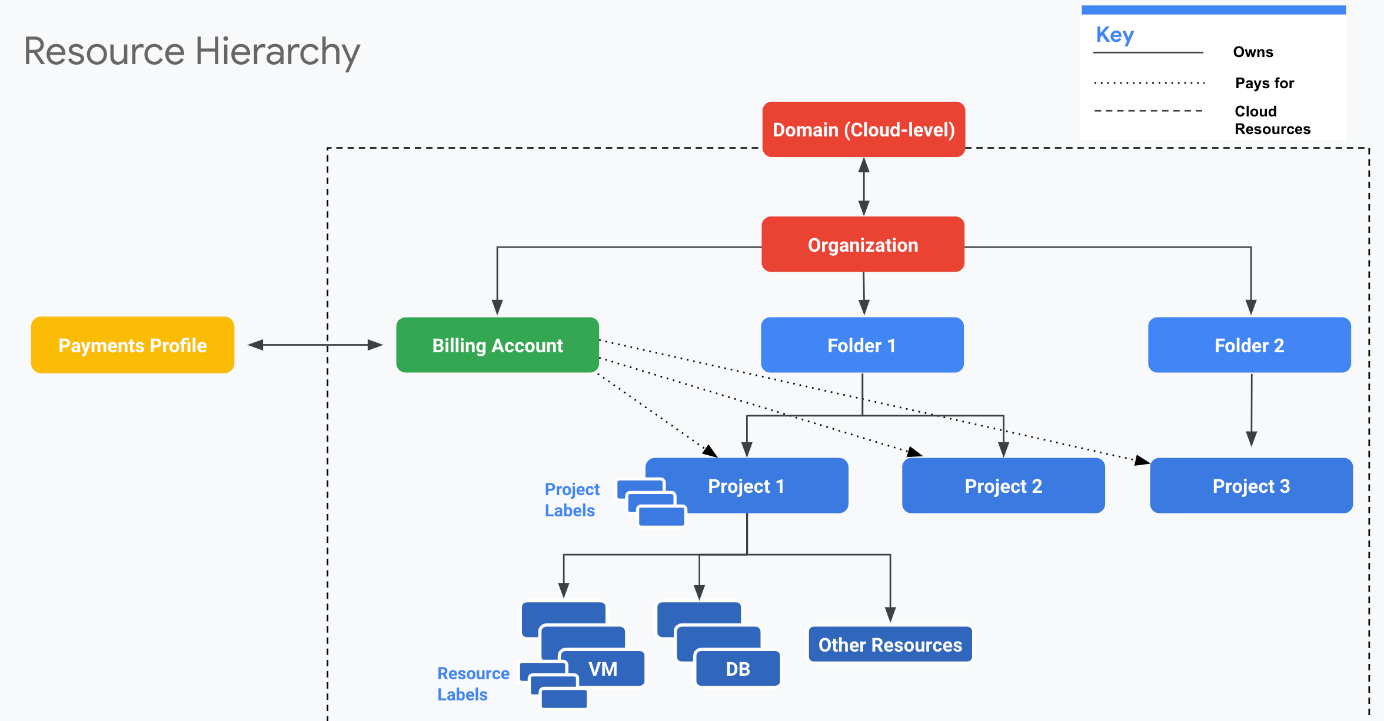


Abbildung 21 - GCP Hierarchie

**Projekte:**

Projekte können als logische Container gesehen werden, die mehrere Resourcen gruppieren und zusammenfassen. Gut zu vergleichen mit einer Resourcengruppe in Azure. Es können mehrere Projekte in einem Ordner und somit in einer Organisation angelegt werden. Projekte sind zwingend Notwendig da sie die den benötigten Container abbilden in denen Resourcen erstellt werden. Wenn kein Projekt existiert und versucht wird eine Resource anzulegen, muss zwingend auch ein Projekt erstellt werden, genauso wie bei Azure.

**Resourcen:**

Gleich wie bei Azure und anderen Cloud Anbietern stellt die Resource ein von Google angebotenes Dienst dar. Dies kann eine virtuelle Maschine, ein Kubernetes Cluster oder ein andere Dienst sein. Eine Liste aller zur Verfügung stehen Dienste: <https://cloud.google.com/products/>

## Resourcenverwaltung

Projekte sind logische Einheiten, die mehrere Resourcen innehat und diese verbraucht. Projekte werden innerhalb von Organisationen eingerichtet. Projekte sind mit Konten und Konten mit Organisationen verknüpft. Die Abrechnung erfolgt jedoch tatsächlich projektbezogen. Jedes Projekt kann als Resourcen + Einstellungen + Metadaten betrachtet werden. Im Kern ist die Abrechnung somit mit Projekten verbunden.

Die Resourcen auf der untersten Ebene stellen die grundlegenden Komponenten aller GCP-Dienste dar. Beispiele für Resourcen sind unter anderem virtuelle Compute Engine-Maschinen (VMs), Cloud Pub/Sub-Themen, Cloud Storage-Buckets und App Engine-Instanzen. Allen Resourcen auf der untersten Ebene können nur Projekte übergeordnet sein. Sie stellen den ersten Gruppierungsmechanismus der GCP-Resourcenhierarchie dar.

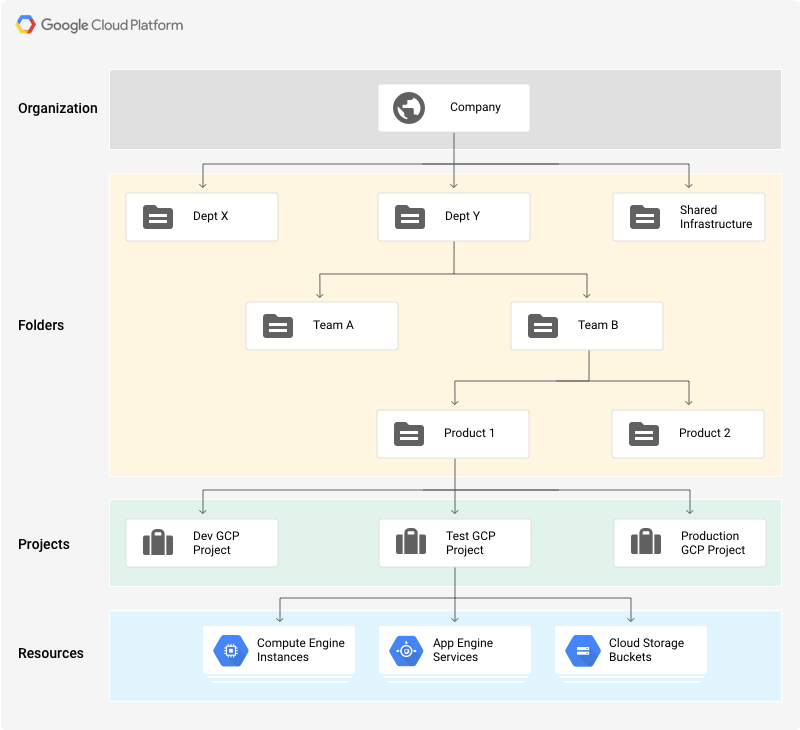


Abbildung 22 - GCP Resourcenorganisation

Als weiterer Gruppierungsmechanismus sind Projekten Ordner übergeordnet. Um Ordner verwenden zu können, benötigen Sie eine OrganisationsResource. Alle Ordner und Projekte sind der OrganisationsResource untergeordnet.

Die OrganisationsResource ist der Stammknoten der GCP-Resourcenhierarchie. Alle zu einer Organisation gehörenden Resourcen sind unter dem Organisationsknoten gruppiert. Dies ermöglicht die zentralisierte Sichtbarkeit und Steuerung aller Resourcen einer Organisation.

## Studentenangebote

Die Informationen, die von Google bezüglich eines Angebots für Studenten bereitgestellt werden, sind im Vergleich zu Azure wenig bis gar nicht vorhanden.

Grundsätzlich gibt es die Möglichkeit GCP Credits einzulösen. Diese können nur durch einen Lehrenden eingefordert werden. Hier bekommt man 100$ pro Kurs für Lehrende und 50$ für jeden Studenten. Dann gibt es noch Training Credits die genutzt werden können um Qwiklab Kurse für GCP durchzumachen. 5,000 Qwiklab Credits werden der Universität und 200 Credits dem Studenten zugewiesen. Diese Kurse sind vergleichbar mit Kursen bei Pluralsight mit dem Hands-On Zusatz.

Eine Liste aller Qwiklab Kurse: <https://google.qwiklabs.com/>

## Kostenmanagement

Das Verrechnungskonzept in der GCP basiert auf ein Rechnungskonto und ein Zahlungsprofil. Basierend auf diese zwei Komponenten werden anfallende Kosten verrechnet. Einen Überblick über die verrechneten Kosten und zukünftige Prognosen erhält man über Built-In Reports in der Google Console (vergleichbar mit dem Azure Portal) oder durch benutzerdefinierte Dashboards im Data Studio Service.

Ein Rechnungskonto definiert wer einen bestimmten Satz von GCP-Resourcen bezahlt. Das Rechnungskonto muss einem Google Zahlungsprofil zugewiesen sein. Das Zahlungsprofil enthält Zahlungsmittel (z.B.: Kreditkarte) zum Abbuchen von Gebühren. Die Unterschiede zwischen Rechnungskonto und Zahlungsmittel und wie diese in Beziehung stehen sind den nachfolgenden Abbildungen zu entnehmen.

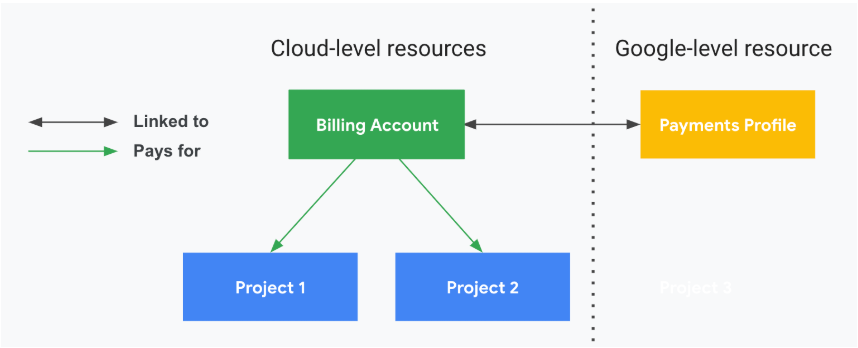


Abbildung 23 - Beziehung zwischen Rechnungskonto und Zahlungsprofil

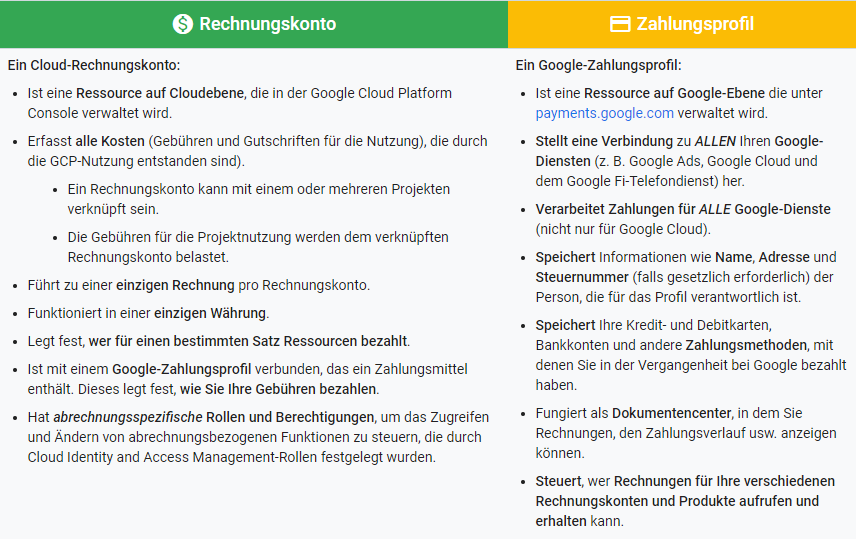


Abbildung 24 - GCP Rechnungskonto und Zahlungsprofil

Es gibt, wie auch bei Azure, zwei unterschiedliche Arten für Services zu zahlen:

**Self-Service**

* Zahlungen erfolgen per Kredit- oder Debitkarte oder per Automated Clearing House (ACH)-Lastschrift, je nach Verfügbarkeit im jeweiligen Land bzw. in der jeweiligen Region.
* Gebühren werden automatisch abgebucht.

**Per Rechnungsstellung**

* Zahlungen erfolgen per Scheck oder Überweisung.
* Rechnungen werden per Post oder elektronisch gesendet.

Ein möglicher GCP Abrechnungsreport könnte wie folgt aussehen:

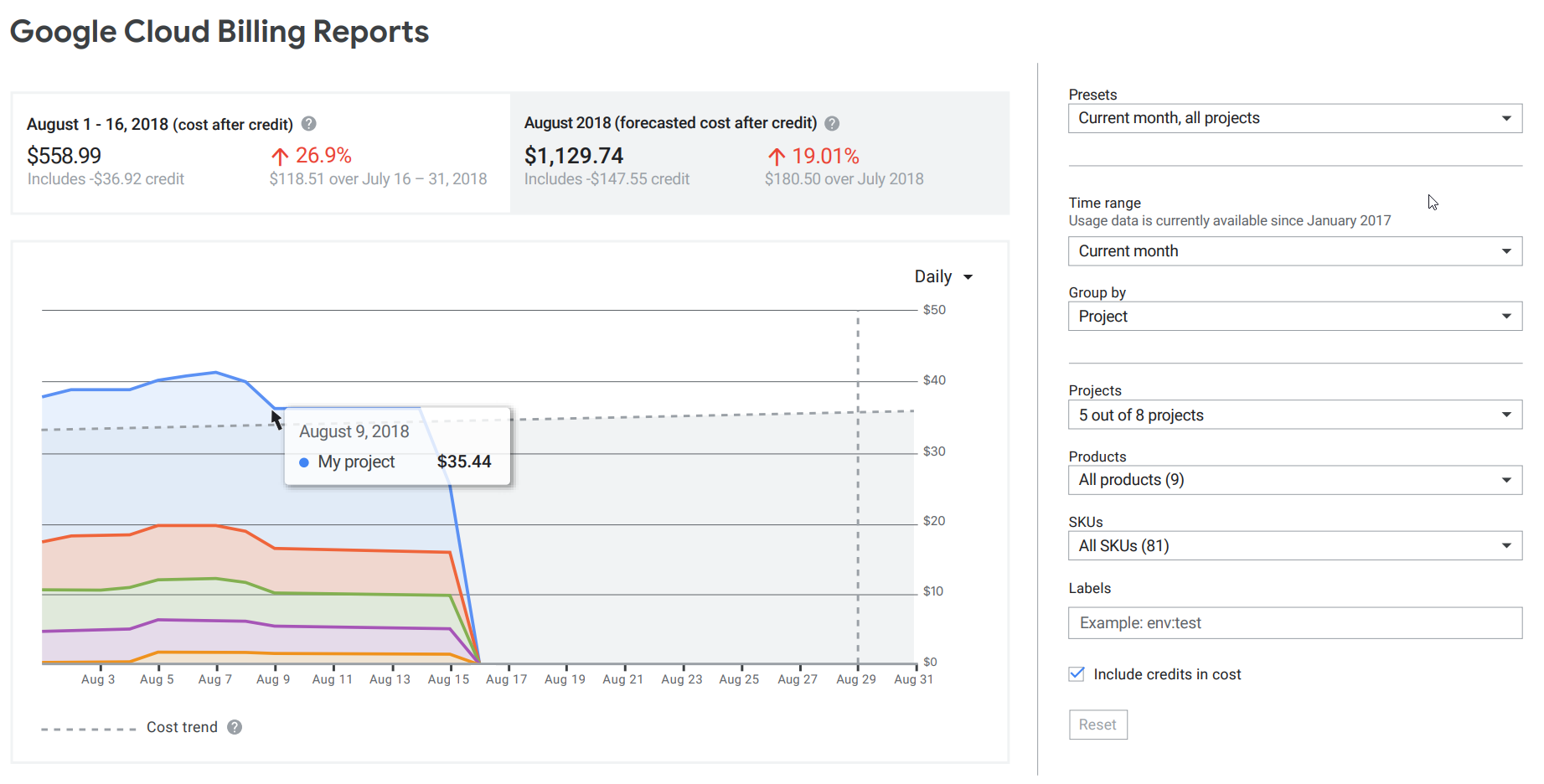


Abbildung 25 - GCP Abrechnungsreport

## Docker

In diesem Abschnitt wird aufgezeigt welche Möglichkeiten es in der GCP gibt Docker einzusetzen. Zusätzlich werden etwaige Einschränkungen aufgelistet und eine grobe Übersicht gegeben wie sich die Kosten zusammensetzen.

### Google App Engine

Managed-Service in dem Docker-Container auf einer VM bereitgestellt werden können.

GAE ist eine Plattform für serverlose Anwendungen, die Entwicklern Entwicklungs- und Hosting-Funktionen bietet. Programme können in den gängigsten Programmiersprachen wie Python, Go und Java entwickelt werden. GAE liefert die Infrastruktur für den Betrieb von Webanwendungen im Cloud-Bereich.

 Docker Compose

 Docker Swarm

**Kosten:**

Instanzen in der flexiblen Umgebung 
In der flexiblen App Engine-Umgebung ausgeführte Anwendungen werden für VM-Typen bereitgestellt, die Sie angeben. 
Diese virtuellen Maschinenressourcen werden pro Sekunde mit einer minimalen Nutzungsgebühr von 1 Minute 
abgerechnet. 
In dieser Tabelle werden die Stundensätze der verschiedenen Computing-Ressourcen zusammengefasst: 
Frankfurt 
Ressource 
VCPU 
Speicher 
Persistente Festplatte 
Einheit 
pro Kernstunde 
pro G8-Stunde 
pro GB pro Monat 
Kosten pro Einheit 
$0.063 
$0.oog 
$0.048 

Abbildung 26 - GAE Kostenübersicht

### Google Compute Engine (Linux)

Bei GCE (Linux) handelt es sich um keinen Managed-Service. Man ist für die instanziierten VMs selbst verantwortlich. Man hat die Möglichkeit Docker EE direkt auf einer Google Cloud Virtual Machine (Linux) zu installieren.

**Einschränkungen:**

* Es ist nicht möglich, die Ports einer VM-Instanz den Ports des Containers zuzuordnen (Docker-Option -p).
* Sie können Container nur aus einem öffentlichen Repository oder aus einem privaten Repository in der Google Container Registry bereitstellen. Andere private Repositories werden derzeit nicht unterstützt.

 Docker Compose

 Docker Swarm

**Kosten:**

Um die Kosten für die Bereitstellung der Mikroservices basierend auf Docker-Images zu berechnen, muss eine VM ausgewählt und konfiguriert werden. Mithilfe des Pricing Calculators können die Kosten ca. vorberechnet werden: https://cloud.google.com/products/calculator/

### Google Compute Engine (Container-Optimized OS)

Bei GCE (Linux) handelt es sich um keinen Managed-Service. Man ist für die instanziierten VMs selbst verantwortlich.

VMs werden auf ein für Docker-Container optimiertes Betriebssystem aufgesetzt. Auf Instanzen mit Container-Optimized OS sind die Docker-Laufzeit und cloud-init vorinstalliert. Wenn Container-Optimized OS verwendet werden, können Docker-Container gleichzeitig mit dem Erstellen der VM online gesetzt werden, ohne dass eine Einrichtung auf dem Host erforderlich ist.

 Docker Compose

 Docker Swarm

**Einschränkungen:**

* Container-Optimized OS enthält keinen Paketmanager, weswegen Softwarepakete nicht direkt auf einer Instanz installieren werden können.
* Container-Optimized OS kann nicht außerhalb der Google Cloud Platform-Umgebung eingesetzt werden.

Anleitung um Docker-Compose auf einer Container-Optimized OS Instanz auszuführen: <https://cloud.google.com/community/tutorials/docker-compose-on-container-optimized-os>

**Kosten:**

Um die Kosten für die Bereitstellung der Mikroservices basierend auf Docker-Images zu berechnen, muss eine VM ausgewählt und konfiguriert werden. Mithilfe des Pricing Calculators können die Kosten ca. vorberechnet werden: <https://cloud.google.com/products/calculator/>

### Google Kubernetes Engine

Die Kubernetes Engine verwaltet alle instanziierten VMs und verwaltet das Scaling.

Das Ausführen jedes Mikrodienstes auf einer separaten VM in Compute Engine kann durch die Beanspruchung des Betriebssystems erhebliche Kosten verursachen. Mit der Kubernetes Engine können mehrere Container und Gruppen von Containern für jede VM-Instanz bereitstellen. Dadurch können VM-Resourcen des Hosts Mikroservices mit geringerem Platzbedarf effizienter zugewiesen werden.

Anleitung um Docker-Containerimage in Kubernetes bereitzustellen: <https://cloud.google.com/kubernetes-engine/docs/tutorials/hello-app>

 Docker Compose

 Docker Swarm

**Kosten:**

GKE verwendet Google Compute Engine-Instanzen für Knoten im Cluster. Diese Instanzen werden Ihnen entsprechend den Compute Engine-Preisen in Rechnung gestellt, bis die Knoten gelöscht werden. Compute Engine-Resourcen werden pro Sekunde mit Mindestnutzungskosten von einer Minute abgerechnet.

# Entscheidung Plattform

Nachdem die drei Cloud Plattformen Azure, GCP und AWS analysiert wurden, fiel die Entscheidung auf Microsoft Azure.

Hauptgründe für diese Entscheidung waren das Studentenangebot bei dem man 100$ Guthaben bekommt, die schon für einige von uns im Jahrgang bekannte Umgebung im Azure Portal und die zahlreichen Services die eine Bereitstellung einer Microservice-Architektur Umgebung in Docker erlaubt. Zusätzlich verfügen wir über ein separates Abonnement, welches vom Studiengang und dem ENTA Bereich bereitgestellt wird. Das Abonnement Entology erlaubt es mehrere Resourcen und Projekte in Azure abzubilden weil das Budget nicht auf 100$ limitiert ist.

In der nachfolgenden Tabelle wird nochmals ein kurzer Vergleich gezogen um die Entscheidung nachvollziehbar zu machen.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Azure** | **GCP** | **AWS** |
| **Studentenangebot** | Azure for Students:   * 12 Monate Subscription * 100$ Guthaben für kostenpflichtige Services   Es stehen 27 Services jederzeit kostenlos zur Verfügung.  Die, auf 12 Monate begränzte, Subscription kann erneuert werden, solange man noch Student ist.  <https://azure.microsoft.com/de-de/free/free-account-students-faq/> | Ähnlich wie bei AWS werden bei GCP Modulle in einer Lernplattform (Qwiklab) in Form von Credits angeboten. Hier stehen 5,000 Qwiklab Credits der Universität zur Verfügung und 200 Credits den Studenten.  Das von GCP angebotene Guthaben ist im Vergleich zu Azure kleiner. Studenten bekommen 50$ auf ihr Konto zugeschrieben. Jedoch können Studenten dieses Guthaben nicht selbst einlösen. Dieses muss von einem Lehrenden bei GCP angefordert werden.  <https://google.qwiklabs.com/> | AWS for Students bietet lediglich eine Lernplattform an.  Ein Free Tier Account bietet kostenlose Services an. Einige wenige Services stehen jederzeit kostenlos zur Verfügung. Andere werden nur in den ersten 12 Monaten kostenlos angeboten.  Für die Registrierung eines Free Tier Accounts müssen sehr viele Daten angegeben und eine Kreditkarte hinterlegt werden.  <https://amzn.to/2MzevN1> |
| **Docker Support** | Azure for Docker:   * Unterstützung für Docker Compose und Docker Swarm * Bereitstellung auf einer VM mit speziell-angepassten Kernel basierend auf einem Azure Template * Azure File Storage wird als Storage verwendet (hohe Effizienz und Leistung)   Azure Container Instances:   * Fokus auf isolierte Container * Keine Unterstützung für Docker Compose oder Docker Swarm * Deployment von mehreren Containern nur durch YAML File oder Resource Manager Template   Azure Service Fabric:   * Docker Compose wird nur eingeschränkt unterstützt. Application Upgrade steht nicht zur Verfügung. Für ein Upgrade muss die Application beendet, gelöscht und neu deployed werden. * Keiner Unterstützung für Docker Swarm * Container werden in einem Cluster bereitgestellt. Dieser besteht aus mehreren Nodes die entweder als Linux oder Windows VM abgebildet werden.   Azure App Service:   * Docker Compose wird unterstützt * Docker Swarm steht nicht zur Verfügung * Managed-Services welches integrierte Docker Images anbietet * Alle Docker Images werden in Azure Container Registry abgelegt   Azure Virtual Machine:   * Unterstützung für Docker Compose und Docker Swarm * Docker Images werden auf Linux oder Windows VM deployed * Kosten werden durch ausgewählte VM Tier bestimmt | Google App Engine:   * Unterstützt Docker Compose * Managed-Service welches Container auf VMs deployed   Google Compute Engine (Linux):   * Wie bei Azure VMs, ist man für die instanziierten VMs selbst verantwortlich * Docker werden in der Linux VM deployed * Unterstützung für Docker Compose und Docker Swarm   Google Compute Engine (Container-Optimized OS):   * Unterstützt Docker Compose * Docker Swarm steht nicht zur Verfügung * Docker werden auf für Docker-Container optimiertes Betriebssystem in einer VM deployed * Container-Optimized OS steht nur innerhalb von GCP zur Verfügung und kann nicht außerhalb der Cloud-Umgebung eingesetzt werden   Google Kubernetes Engine:   * Keine Unterstützung für Docker Compose oder Docker Swarm * Einzelne Docker Container können deployed werden * Bereitgestellte Container setzen auf VMs auf die von der GKE verwaltet werden * Gut geeignet für Microservice Architekturen jedoch mit der Einschränkung --> Compose und Swarm nicht verfügbar | AWS EC2:   * Unterstützung für Docker Compose und Docker Swarm * Docker Images werden auf Linux oder Windows VM deployed * Kosten werden durch ausgewählte VM Tier bestimmt   AWS ECS CLI:   * Unterstützt Docker Compose * Docker Swarm steht nicht nur Verfügung * Wird für die Erstellung, Aktualisierung und Überwachung von Clustern in einer Entwicklungsumgebung eingesetzt   AWS EKS:   * Unterstützt Docker Compose und Docker Swarm * Kubernetes verwaltet Cluster, die aus Amazon EC2-Instances für die Datenverarbeitung bestehen, und führt für diese Instances Container mit Prozessen zur Bereitstellung, Wartung und Skalierung aus   AWS Fargate:   * Als Datenverarbeitungs-Engine wird es ermöglicht Container ohne Server oder Cluster zu verwalten * Docker Compose und Docker Swarm werden nicht unterstütz |
| **Kosten** | Azure bietet eine gebührenfreie Kostenanalyse für alle deployed Services an. Zusätzlich zu den schon angefallenen Kosten werden zukünftig anfallende Kosten (basierend auf den bereitgestellten Services) prognostiziert. Es wird eine detaillierte Kostenübersicht nach Resourcen ermöglicht und eine Ausgabequote sowie Prognose für jede Resourcengruppe bzw. für das Abonnement zur Verfügung gestellt.  Des Weiteren kann der Azure Advisor Service genutzt werden. Dort werden Empfehlungen sowie Maßnahmen gegeben um Geld zu sparen und die Zuverlässigkeit zu verbessern. | Das Verrechnungskonzept bei GCP ist etwas kompliziert auf den ersten Blick da man zwischen Rechnungskonto und Zahlungsprofil unterscheiden muss.  Grundsätzlich stehen alle Services und Funktionen wie sie auch Azure anbietet zur Verfügung mithilfe des Google Cloud Billing Reports. | AWS ermöglicht es mithilfe des AWS Budget Services benutzerdefinierte Budget festzulegen und bei einem selbstdefiniertem Threshold benachrichtigt zu werden.  Budget kann monatlich, vierteljährlich oder jährlich im Dashboard nachverfolgt werden.  Die AWS Billing Console ermöglicht es, genauso wie bei Azure die Kostenanalyse, die angefallenen Kosten zu überblicken. Zusätzlich können Reports im CSV oder Excel Format erstellt werden. |