



**TECH
STARTER**

Compute Services

Container -
Compute -
Services



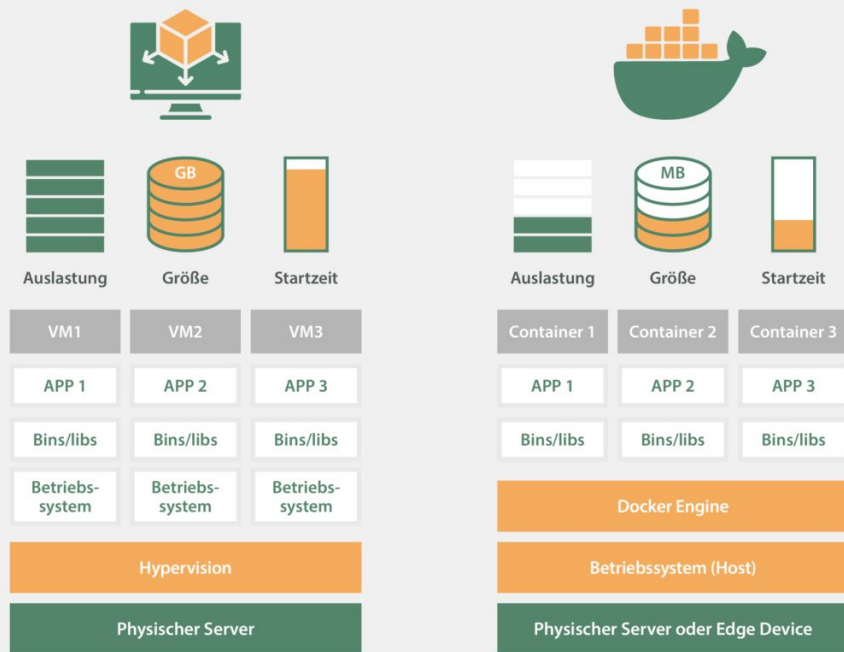
**TECH
STARTER**

Was ist Docker

- Apps werden in Container verpackt
- Können in der Theorie auf jedem OS laufen
- Einfacher skalieren, da Apps schon “vorgepackt” sind
- Apps werden in Docker Images “verpackt”
- Apps laufen überall, wo docker software hinterlegt ist
- Ressourcen werden mit einem Host geteilt → Viele Docker Images auf einem Server

Was ist Docker

Virtuelle Maschine vs. Docker



Quelle: <https://www.opc-router.de/>

Was ist Docker

<https://www.opc-router.de/was-ist-docker/>

Speicherung von Docker Images

- Docker Images werden in Docker Repositories gespeichert
- Öffentlich unter <https://hub.docker.com/>
- AWS bietet eine private Möglichkeit an, Docker Images zu speichern
→ AWS ECR (Elastic Container Registry)

Elastic Container Service (ECS)

- Der Service Docker Container auf AWS laufen zu lassen
- Shared Responsibility Model → Ihr müsst euch um die EC2 Instanzen kümmern
- Shared Responsibility Model → AWS startet/stoppt die Container für euch
- Kann mit dem Application Load Balancer vereint werden

Fargate

- Ein weiterer Service Docker Container auf AWS laufen zu lassen
- Shared Responsibility Model → Ihr müsst euch NICHT um die EC2 Instanzen kümmern
- Shared Responsibility Model → AWS startet/stoppt die Container für euch basierend auf den Ressourcen, welche ihr benötigt (CPU/RAM)
- Einfache Integration mit ECR
 - speichern auf ECR
 - Container starten mit Fargate oder ECS

Serverless Begriff

→ Es gibt Server, ihr müsst euch aber nicht um die Verwaltung der Server kümmern

AWS Lambda



EC2

- VM in der Cloud
- Leistung ist auf die Größe der VM begrenzt
- Abrechnung nach Stunde → Laufen durchgehend
- Best practice: Horizontale Skalierung
- IaaS



Lambda

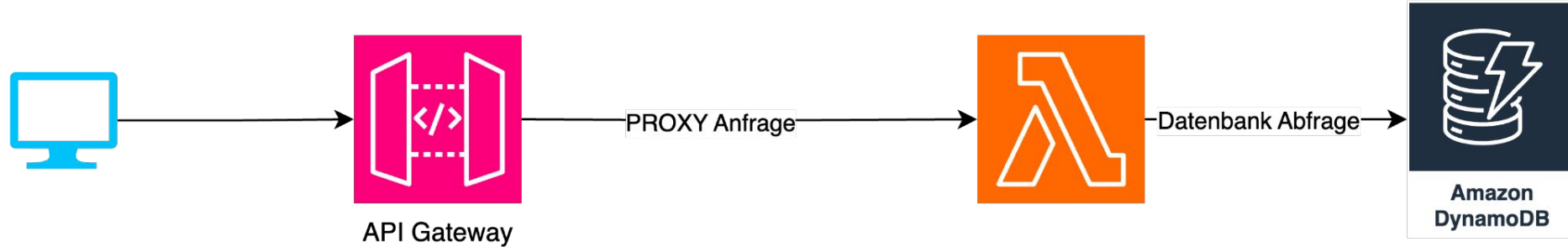
- PaaS
- Kümmt sich um die App an sich
- Zeitlich begrenzt (max.15 Minuten)
- Abrechnung nach Ausführung/ Nutzung des Ressourcen in der Zeit
- Ihr müsst euch nicht um die Skalierung kümmern :)



AWS Lambda - Unterstützte Runtimes

- Node.js (JavaScript)
- Python
- Java (Java 8 compatible)
- C# (.NET Core)
- Golang
- C# / Powershell
- Ruby

API Gateway



- APIs Erstellen mit API Gateway
- Serverless und skalierbar
- Sicheres API Management

AWS Batch

- Ein Batch Job ist eine Gruppe von Aufgaben, die gleichzeitig abgearbeitet werden
- AWS Batch übernimmt die Verwaltung von dem Batch Job, indem AWS die Skalierung und Verwaltung der Ressourcen übernimmt
- Batch Jobs werden über Docker Images definiert und laufen im Hintergrund mit dem ECS Service

AWS Lightsail

- Einfache Alternative für die Nutzung von EC2, RDS, ELB, EBS, Route53,...
 - Weniger Freiheiten in den Einstellungen
- Gut für Leute mit wenig Erfahrung → Also nicht für euch zukünftig :)
- Anwendungen:
 - Einfache Web Anwendungen
 - Dev/Test Umgebungen
- Hat eine hohe Verfügbarkeit, aber kein Auto-Scaling, limitierte AWS Integration