

Block 3: Einführung in Container-Technologie

Installation Docker Engine

Dependencies

```
sudo apt update  
sudo apt install -y apt-transport-https ca-certificates curl gnupg-agent software-properties-common
```

Repository

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg  
echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archive-keyring.gpg]  
https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list  
> /dev/null
```

Docker Engine

```
sudo apt-get update  
sudo apt-get install docker-ce docker-ce-cli containerd.io -y
```

Anpassungen User (dieser Schritt ist wichtig, da sonst immer mit sudo gearbeitet werden muss)

```
sudo usermod -aG docker $USER
```

```
newgrp docker
```

Auftrag 3.1: Grundlagen Container-Technologie

1. Was ist Container-Technologie oder Container-Virtualisierung? Container-Technologie oder Container-Virtualisierung ist ein Ansatz zur Softwareentwicklung, bei dem eine Anwendung und ihre Abhängigkeiten in einem Prozess gekapselt werden, der von der zugrunde liegenden Betriebssysteminfrastruktur isoliert ist. Jeder Container teilt sich das Betriebssystem-Host und in der Regel auch die Binärdateien und Bibliotheken, ist aber in seiner Ausführung vom Rest des Systems getrennt.
2. Was sind die Vor- und Nachteile der Container-Technologie zu virtuellen Servern (VM)?
Vorteile von Containern gegenüber VMs: • Geringerer Ressourcenbedarf: Container benötigen weniger Systemressourcen als VMs, da sie weniger Overhead haben, da sie das Betriebssystem nicht duplizieren müssen. • Schnellere Startzeiten: Container können in Sekundenbruchteilen gestartet und gestoppt werden, während VMs Minuten benötigen. • Portabilität: Anwendungen in Containern können einfach zwischen verschiedenen Systemen und Cloud-Plattformen verschoben werden.
Nachteile von Containern gegenüber VMs: • Geringere Isolierung: Da Container das gleiche Betriebssystem teilen, sind sie weniger isoliert als VMs. • Spezifische Betriebssystemabhängigkeiten: Container sind oft an das zugrunde liegende Betriebssystem gebunden, während VMs beliebige Betriebssysteme ausführen können.
3. Welche Produkte kennen Sie im Zusammenhang mit virtuellen Servern und Containern?
Im Zusammenhang mit virtuellen Servern sind Produkte wie VMware vSphere, Microsoft Hyper-V und Oracle VM bekannt. Bezüglich Containern sind Docker, Kubernetes, Red Hat OpenShift und Rancher einige der bekanntesten Technologien.
4. Wie unterscheiden sich die Technologien VM und Container in Bezug auf Bereitstellung, Speicherplatz, Portabilität, Effizienz und Betriebssystem (Kernel)? • Bereitstellung: Container können schneller bereitgestellt werden als VMs, da sie weniger Ressourcen benötigen. • Speicherplatz: Container benötigen weniger Speicherplatz als VMs, da sie das Betriebssystem nicht duplizieren müssen. • Portabilität: Container sind portabler als VMs, da sie die gesamte Anwendung und ihre Abhängigkeiten in einem einzigen Paket kapseln. • Effizienz: Container sind effizienter als VMs in Bezug auf Systemressourcen, da sie weniger Overhead haben. • Betriebssystem (Kernel): VMs können verschiedene Betriebssysteme auf demselben physischen Host ausführen, während Container das Host-Betriebssystem teilen müssen.
5. Können virtuelle Server immer durch Container ersetzt werden? Nein, es gibt Szenarien, in denen VMs eine bessere Wahl sind. Zum Beispiel, wenn vollständige Isolierung oder spezifische Betriebssystemanforderungen erforderlich sind. Außerdem können bestimmte

Anwendungen oder Systeme aufgrund ihrer spezifischen Anforderungen oder Abhängigkeiten besser für VMs geeignet sein.

6. Was ist unterschied zwischen Self-Managed und Fully Managed? Ein Self-Managed-Service bedeutet, dass der Nutzer selbst für die Installation, Wartung, Aktualisierungen und Sicherheit verantwortlich ist. Der Nutzer hat mehr Kontrolle, aber auch mehr Verantwortung. Ein Fully Managed Service wird vollständig vom Anbieter verwaltet. Der Anbieter kümmert sich um die Installation, Wartung, Aktualisierungen und Sicherheit. Dies kann die Belastung für den Nutzer reduzieren, nimmt ihm jedoch auch einen Teil der Kontrolle ab.

Auftrag 3.2: Webserver mit Docker

Lassen Sie ihre HTML Webseite in einem Container laufen. Nutzen Sie dazu Container Images von Docker Hub, z.B. das Image von nginx: https://hub.docker.com/_/nginx

```
docker run -p 8080:80 -v /home/minikube/index.html:/usr/share/nginx/html/index.html index:latest
```

```
version: '3.8'
```

```
services:
```

```
  nginx:
```

```
    image: index:latest
```

```
    ports:
```

```
      - 8080:80
```

```
    volumes:
```

```
      - /home/minikube/index.html:/usr/share/nginx/html/index.html
```

Revision #1

Created 15 December 2023 14:10:55 by Manuel Regli

Updated 15 December 2023 14:11:25 by Manuel Regli