

169: DIENSTE MIT CONTAINERN BEREITSTELLEN

M169 THEORIETEST

Pascal Gafner

Name

Vladan Marlon
Vranjes

Datum

15.03.2024

Prüfung

M169 Theorietest

Durchführung

M169 LB2
Inf2022c/e Gafner

**Punkte
Total**

13.78/ 20 Punkte

Note

4.4

Fragenpool Docker

Welche Container laufen?

Mit welchem Befehl finde ich heraus, welche Container am laufen sind?

Single-Choice

✖ 0 / 1 Punkte

☐ sudo docker ps -a

Lösung

☐ sudo docker show container -a

☐ sudo docker container show -a

☒ sudo docker container show -running

Falsch

Images anzeigen

Wie können wir alle Images auf dem PC anzeigen?

Single-Choice ✖ 0 / 1 Punkte

- ☐ `sudo docker images -a`
- ☐ `sudo docker images show -all`
- ☒ `sudo docker ls images -a`
- ☐ `sudo docker show images -a`

[Lösung](#)

Falsch

Markup Lang für Docker config File

In welcher Markup Sprache werden die Docker Konfigurationsdateien geschrieben?

Multiple-Choice ✔ 1 / 1 Punkte

- ☐ XML
- ☐ HTML
- ☒ YAML
- ☐ JSON

Richtig

Port Mapping_ck

Wie wird der Host-Port auf den Port des Containers gesetzt?

Single-Choice

✓ 1 / 0 Punkte

- ☐ docker run -i : imagename
- ☒ docker run -p 5000:5000 flask_app:1.0
- ☐ docker run --expose 5000:5000 flask_app:1.0
- ☐ Keine der obigen Antworten trifft zu

Falsch

Kein Feature von VMs_ck

Welche Eigenschaft ist nich durch VMs gegeben?

Single-Choice

✓ 0 / 0 Punkte

- ☒ Skalierbarkeit
- ☐ Bessere zusammenlegen von Ressourcen
- ☐ Garantierte Applikations-Portabilität
- ☐ Alle genannten Punkte sind Eigenschaften von VMs.

Falsch

Single-Choice

✓ 0 / 0 Punkte

Container starten

Mit welchem Befehl können Sie den Container hello-world starten?

Single-Choice

✓ 1 / 1 Punkte

☒ `sudo docker run hello-world`

Richtig

☐ `sudo docker hello-world start`

☐ `sudo docker up hello-world`

☐ `sudo docker hello-world -up`

Fragenpool Image auf Git

Grösse eines Images_ck

Wie viel Speicher wird für ein Image der TODO App v2 in einem Repo benötigt?
Markieren sie die korrekte Grössenordnung des benötigten Speicherplatzes.

Single-Choice

✓ 1 / 0 Punkte

☐ zwischen 1..10kB

☒ zwischen 1..10MB

Falsch

☐ zwischen 1..10GB

registry

Worin liegt der Unterschied zwischen den beiden Befehlen?

```
docker login
docker login registry.gitlab.com
```

✓ 1 / 1 Punkte

Docker login alleine meldet sich bei docker hub an, wobei registry.gitlab.

Lösung

ohne Pfad: Docker Hub, mit Pfad auf andere Ablage (hier gitlab)

Fragenpool Docker

Compose

Mehrere Files angeben

Sie geben beim Befehl `docker compose` mit der `-f` Option mehrere gültige Dateien wie folgt an:

```
docker compose -f docker-compose.yml -f docker-compose.admin.yml run backup_db
```

Single-Choice

✓ 1 / 1 Punkte

- ☒ Die Dateien werden zu einer einzigen Konfiguration zusammengeführt. Richtig
- ☐ Nur die erste Datei wird berücksichtigt.
- ☐ Nur die letzte Datei wird berücksichtigt.
- ☐ Eine Fehlermeldung erscheint und der Befehl wird nicht ausgeführt.

Docker-Compose

Wordpress

In folgendem Docker-Compose File gingen Volumes und Ports vergessen.

1. Ergänzen Sie bei wordpress ein Volumen. Der Hostordner webseite, soll mit dem Ordner /var/www/html verbunden sein. Der Hostordner webseite befindet sich im Ordner /home/vmadmin/wordpress.
2. Die Datenbank erhält ein Docker-Volume. Der Name ist database und ist mit dem Ordner des Containers /var/lib/mysql verbunden.
3. Wordpress soll auf dem Host per 8080 erreichbar sein und an den Container Port 80 weitergeleitet werden. Auch die Datenbank soll über den Port 3306 angesprochen werden können.

Ergänzen sie das Skript und kopieren ihr Resultat in das Abgabefeld unten:

```
version: '3.3'
```

```
services:
```

```
mysql:
```

```
  image: mysql:latest
```

```
  restart: always
```

```
  environment:
```

```
    MYSQL_ROOT_PASSWORD: somewordpress
```

```
    MYSQL_DATABASE: wordpress
```

```
    MYSQL_USER: wordpress
```

```
    MYSQL_PASSWORD: wordpress
```

```
wordpress:
```

```
  depends_on:
```

- mysql

image: wordpress:latest

restart: always

environment:

WORDPRESS_DB_HOST: mysql:3306

WORDPRESS_DB_USER: wordpress

WORDPRESS_DB_PASSWORD: wordpress

WORDPRESS_DB_NAME: wordpress

unter wordpress:

COPY /var/www/html /home/vmadmin/wordpress

WORKDIR /wordpres

Lösung

version: '3.3'

services:

mysql:

image: mysql:latest

volumes:

- database:/var/lib/mysql

restart: always

environment:

MYSQL_ROOT_PASSWORD: somewordpress

MYSQL_DATABASE: wordpress

MYSQL_USER: wordpress

MYSQL_PASSWORD: wordpress

wordpress:

depends_on:

- mysql

image: wordpress:latest

ports:

- '8000:80'

```
restart: always
```

```
volumes:
```

```
- ./website:/var/www/html
```

```
environment:
```

```
WORDPRESS_DB_HOST: mysql:3306
```

```
WORDPRESS_DB_USER: wordpress
```

```
WORDPRESS_DB_PASSWORD: wordpress
```

```
WORDPRESS_DB_NAME: wordpress
```

Dockercompose Next-

Cloud

In folgendem Docker-Compose File gingen die Volume's und Ports vergessen.

1. Ergänzen Sie bei nextcloud ein Volumen. Der Hostorder config, soll mit dem Containerordner /var/www/html/config verbunden werden. Der Hostorder config befindet sich am gleichen Ort wie das Docker-compose File.

2. Die Datenbank erhält ein Docker-Volume. Der Name ist nextcloud-db und ist mit dem Ordner des Containers /var/lib/mysql verbunden.

3. Nextcloud soll auf dem Host per 8080 erreichbar sein und an den Container-Port 80 weitergeleitet werden. Auch die Datenbank soll über den Port 3306 angesprochen werden können.

Ergänzen sie das Skript und kopieren ihr Resultat in das Abgabefeld unten:

version: '3'

services:

nextcloud-db:

image: mariadb

container_name: nextcloud-db

restart: always

environment:

- MYSQL_ROOT_PASSWORD=Muster!

- MYSQL_PASSWORD=Muster!

- MYSQL_DATABASE=nextcloud

- MYSQL_USER=nextcloud

nextcloud:

image: nextcloud

container_name: nextcloud

depends_on:

- nextcloud-db

unter nextcloud:

port: 8080:80

unter db:

COPY /var/www/html/config /var/lib/mysql

WORKDIR: /var/lib/mysql

Lösung

Volum 2 Punkte

Port 2 Punkte

version: '3'

services:

nextcloud-db:

image: mariadb

container_name: nextcloud-db

ports:

-3306:3306

restart: always

volumes:

- ./db:/var/lib/mysql

environment:

- MYSQL_ROOT_PASSWORD=Muster!

- MYSQL_PASSWORD=Muster!

- MYSQL_DATABASE=nextcloud

- MYSQL_USER=nextcloud

nextcloud:

image: nextcloud

container_name: nextcloud

ports:

- 8080:80

depends_on:

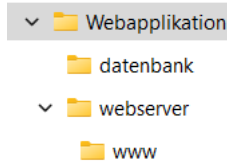
- nextcloud-db

volumes:

- ./config:/var/www/html/config

Docker-compose Webapplikation

Erstellen Sie den Inhalt eines docker-compose File, welches sich im Ordner Webapplikation befindet:



Es soll ein Container hochgefahren werden. Im Ordner webserver befindet sich ein Dockerfile.

Der Webserver soll vom Host auf dem Port 8080 erreichbar und an den Port 80 des Containers weitergeleitet werden.

Das Volume des Webserver usr/share/nginx/html soll mit Ordner des Host Webserver/www verknüpft werden.

Geben Sie den Container sinnvolle Namen.

Code-Aufgabe

3 / 4 Punkte

version: 2 services: service1: image: xxxx port: 8080:80 build: Dockerfile
Volumes: volume: usr/share/nginx/html

Feedback der Lehrperson

Service: Name 1p

image: Korrekter pfad 1p

Ports: 80:8000 1p

Volumes+Environment: Korrekter Pfad 1p

Docker-compose Datenbank

Erstellen Sie den Inhalt eines docker-compose File, welches sich im Ordner Webapplikation befindet:



Es soll ein Container hochgefahren werden. Im Ordner datenbank befindet sich ein Dockerfile.

Die Datenbank soll vom Host auf dem Port 4000 erreichbar und an den Port 3306 des Containers weitergeleitet werden.

Auch die Datenbank legt ihre Daten auf einem Volume ab. Dabei wird das Volume db-mysql mit dem Containerordner var/lib/mysql verknüpft.

Geben Sie den Container sinnvolle Namen.

Ergänzen sie auch die folgenden Umgebungsvariabeln:

MYSQL_ROOT_PASSWORD: somewordpress

MYSQL_DATABASE: wordpress

MYSQL_USER: wordpress

MYSQL_PASSWORD: wordpress

Code-Aufgabe

✖ 0 / 4 Punkte

port: 4000:3306 build: dockerfile volumes: volume: volume1 depends
on: MYSQL_ROOT_PASSWORD: somewordpress MYSQL_DATABASE:
wordpress MYSQL_USER: wordpress MYSQL_PASSWORD: wordpress

Feedback der Lehrperson

build ./datenbank
ports: 4000:3306

Webserver mit Ports+Volumes

depends on DB

Fragepool Kubernetes

Microservices

Worin liegt der Unterschied von synchronen und asynchronen Microservices?

✓ 1 / 0 Punkte

Synchrone microservices laufen bedingt miteinander mit request und antwort wobei asynchrone nicht voneinander abhängig sind und nicht auf eine Antwort von einer anfrage warten

Lösung

Ein Microservice ist synchron, wenn er bei der Bearbeitung von Requests selber einen Request an andere Microservices stellt und auf das Ergebnis wartet. Dies ist bei REST der Fall.

Der asynchrone Microservice schickt einem anderen Microservice (Event Bus, Message Server) einen Request, wartet aber nicht auf eine Antwort.

Cloud Native Apps

Eine cloud-native App muss folgendes unterstützen: (alles was zutrifft ankreuzen)

Multiple-Choice

✓ 0 / 0 Punkte

Multiple-Choice

✗ 0.78 / 1 Punkte

☒ Skalierung nach Bedarf

Richtig

☐ Zerro Down Time

☒ Rolling Updates ohne Ausfallzeit

Richtig

☐ Funktion wo auch immer es Kubernetes gibt

Lösung

☐ self termination

☐ Self-Checking

☒ Self-healing

Richtig

☐ infinite upscaling

☐ Läuft nur in der public Cloud

LoadBalancer

Was ist die Aufgabe eines LoadBalancers? Beschreiben Sie ein Anwendungsszenario.

✓ 0 / 0 Punkte

Ein loadbalancer verteilt die last von aufgaben indem er erkennt wenn ein service sehr oft gebraucht wird und diese auf weitere container verteilt, damit alle gleich belastet sind. Dies geschieht nach zufallsprinzip von einer geteilten Virtual ip oder auch absichtlich.

Scaling

Scale up und scale down beeinflusst:

Single-Choice

✓ 1 / 1 Punkte

- ☐ Anzahl Deployments
- ☐ Anzahl Services
- ☐ Anzahl Loadbalancers
- ☒ Anzahl Pods
- ☐ Anzahl Nodes
- ☐ Anzahl Clusters

Richtig

Konzeptuelle Verständnisfragen

Container

Was ist ein Container? (Die Aussage wählen, die am ehesten zutrifft)

Single-Choice

✓ 1 / 1 Punkte

- ☐ Eine "light" VM, die auch Hardware virtualisiert
- ☐ Ein Vorlage für ein Paket, das eine App beinhaltet sowie alle dazu nötigen Abhängigkeiten.
- ☒ Eine Instanz eines Container Images
- ☐ Eine laufende Instanz eines Container Images

Richtig

Port

Meine containerisierte node Webapp ist so programmiert, dass sie auf Port 80 hört. Port 80 ist aber auf meinem Host bereits besetzt. Wie starte ich die containerisierte App, damit ich vom Host aus via Port 8080 darauf zugreifen kann?

Single-Choice

✔ 1 / 1 Punkte

☒ `docker run -p 8080:80 -d myapp`

Richtig

☐ `docker run -p 80:8080 myapp`☐ `docker run -p 80:80 myapp`☐ `docker run -p 80:80 -d myapp`☐ `docker run -p 8080:8080 myapp`

k8

Was ist ein k8s deployment?

Single-Choice

✘ 0 / 1 Punkte

☐ Ein k8s Objekt ("kind"), das die Erstellung und Verwaltung von ReplicaSets erleichtert.

Lösung

☐ Die Definition eines oder mehrerer Images und deren Abhängigkeiten☒ Eine YAML Definition der Konfiguration des K8S Clusters

Falsch

☐ Eine YAML Definition für die Speicherung von persistenten Daten (z.B. mittels StatefulSets)