

Übungsblatt 10

Microservices mit Docker

Vorbereitung: Docker aufsetzen in U521

Docker (docker.com) ist eine Virtualisierungsumgebung auf Basis des Linux-Kernels (vgl. Vorlesungsfolien). Entsprechend ist Docker auf Windows- und Mac-Umgebungen nicht nativ lauffähig. Auf privaten Rechnern, auf denen Sie lokale Adminrechte haben, können Sie Docker mit entsprechenden Hilfstools (Kitematic und Quickstart) installieren.

In den CIP-Pools ist dies nicht möglich, hier müssen Sie mit einer virtuellen Linux-Umgebung mit Hilfe von VirtualBox arbeiten. Die Einzelschritte sind ausführlich in einer eigenen Anleitung beschrieben (siehe GRIPS!)

Aufgabe 1: Bestehenden Veranstaltungs-Microservice starten

Führen Sie nachfolgende Anweisungen aus und notieren Sie sich die Docker-Befehle:

- Laden Sie über Ihren lokalen Docker-Client das Image `im-vm-011.oth-regensburg.de:5000/vs10`
- Lassen Sie sich dieses und ggf. noch weitere vorhandene Images anzeigen
- Starten Sie einen Container mit obigem Image; berücksichtigen Sie dabei folgende Optionen:
 - o Die Ausgabe soll interaktiv sein, d. h. es soll STDOUT ausgegeben werden
 - o Die Container-Ports 8080 und 5701 sollen nach Außen veröffentlicht werden
 - o Der Container soll einen eigenen Namen bekommen
 - o Im Container soll eine Umgebungsvariable `PUBLIC_ADDRESS` gesetzt werden (der Wert der Umgebungsvariable soll Ihre öffentliche IP-Adresse sein – startet im CIP-Pool mit 172...)
- Welche Option müssten Sie wählen, um den Container im Daemon-Modus zu starten?
- Wie kann eine Liste von Containern (laufende oder bereits beendete) angezeigt werden?
- Wie werden Container gelöscht?
- Wie werden Images gelöscht?

Über den REST-Client können Sie auf die RESTful-Webservices zugreifen. Beenden können Sie ihren Knoten mit: `http://localhost:8080/webresources/admin/shutdown`

Aufgabe 2: Eigenen Microservice als Docker-Image erstellen und ausführen

In GRIPS finden Sie die Datei `uebung10.jar`. Diese enthält die Ergebnisse des Übungsblatts 9 als ausführbares Java-Archiv mit kleinen Änderungen. (Eine Anleitung, wie man das aus dem eigenen Projekt erstellt, finden Sie am Ende dieses Übungsblatts.)

Laden Sie diese Datei und speichern Sie diese in Ihrem Benutzerverzeichnis in einem neuen Unterordner.

Erstellen Sie im neuen Unterordner eine Textdatei namens `Dockerfile` (ohne Dateierweiterung), in der die Schritte zum Bau eines Docker-Images definiert werden. Folgen Sie hierzu nachfolgenden Hinweisen:

- Ihr Docker-Image soll auf einem bereits bestehenden Docker-Image `java:8` aufbauen (Repository `java` mit Image-Tag `8`; dies entspricht JDK8)
- Geben Sie Ihren Namen/Email als „Maintainer“ an
- Es soll die Anweisung `mkdir /usr/src/vs` im neuen Container ausgeführt werden, um dieses Verzeichnis im Container/Image zu erstellen
- Das Arbeitsverzeichnis im Container soll auf das neu erstellte Verzeichnis gesetzt werden
- Die intern belegten Ports `8080` und `5701` sollen vom Container nach Außen „exposed“ werden
- Kopieren Sie die Datei `./uebung10.jar` in den Container in den Pfad `/usr/src/vs/`
- Die letzte Anweisung nach Hochfahren eines Containers soll sein:
`java -jar uebung10.jar`
diese wird die Webservices und einen DataGrid-Knoten starten

Nach Erstellen des `Dockerfile` können Sie mit folgender Anweisung Ihr eigenes Docker-Image erstellen lassen:

```
$ docker build --tag=<image-name> .
```

Als Image-Name wählen Sie bitte Ihren Nachnamen oder Ihren NDS-Account oder einen sonstigen eindeutigen Namen!

Nach Erstellen des Images können Sie einen neuen Container mit oben erstelltem Image starten:

```
$ docker run -it -e PUBLIC_ADDRESS=<ihre-ip> -p 8080:8080 -p 5701:5701  
--name=<container-name> <image-name>
```

Sie können Ihr erstelltes Image auch über eine Vorlesungs-interne Docker-Registry anderen zur Verfügung stellen¹:

(dazu müssen Sie Ihrem Image zunächst ein „tag“ zuordnen, d. h. es umbenennen)
`$ docker tag <image-name> im-vm-011.oth-regensburg.de:5000/<image-name>`

(anschließend können Sie das Image „pushen“)
`$ docker push im-vm-011.oth-regensburg.de:5000/<image-name>`

¹ push und pull in das private Repository `im-vm-011.oth-regensburg.de:5000` sind nur dann möglich, wenn dem Image-Namen Domain und Port des privaten Repository durch „tagging“ vorangestellt werden.

Zum Herunterladen von Images anderer Studierender können Sie entsprechend deren Images „ziehen“:

```
$ docker pull im-vm-011.oth-regensburg.de:5000/<other-stud-image-name>
```

(Als Imagenname ist nun immer der vollständige Name inkl. im-vm-011... zu verwenden!)