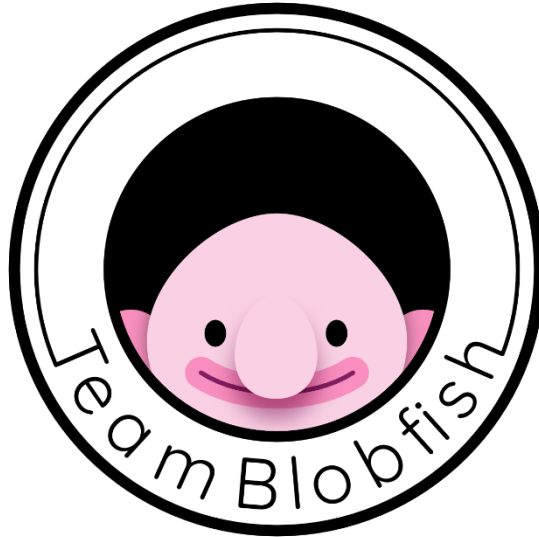
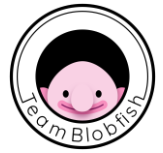


# Deployment Anleitung

Voraussetzung und Step-by-Step Anleitung des Deployments



Autor:  
Florian Braasch, 1824652



## Voraussetzung zum Lokalen Deployment

### Virtualisierungssoftware

Für unseres Deployment habe wir uns für die Virtualisierungsumgebung „Docker Desktop“ entschieden diese ermöglicht nicht nur die Verwendung Docker-Containern, sondern ebenso die Möglichkeit Kubernetes Cluster aufzusetzen.

Um Docker Desktop verwenden zu können benötigt man einen Linux Kernel da wir das Deployment über einen Windows PC realisiert haben, haben wir uns für eine empfohlenen Ubuntu (LTS 18.02) entschieden. Um Linux jedoch verwenden zu können benötigt man ebenfalls die Windows-Subsysteme für Linux Version 2(kurz: WSL2).

Nach dem Download der oben genannten Software folgt ihre Installation dies erfolgt im Fall von Docker Desktop und Ubuntu über eine Bereitgestellte ausführbare Datei. Beginnen wir mit der Installation der Windows-Subsysteme, solange nicht schon zu einem Früheren Zeitpunkt geschehen müssen die WSL zunächst aktiviert werden. Dies erfolgt über Administratoren Rechte und in unserem Fall Windows PowerShell über folgenden Befehl

```
dism.exe /online /enable-feature /featurename:Microsoft-Windows-Subsystem-Linux /all /norestart
```

Anschließend werden die Funktionen eines virtuellen Computers aktiviert über den Folgenden Befehl in der PowerShell

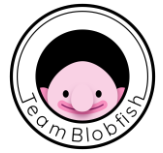
```
dism.exe /online /enable-feature /featurename:VirtualMachinePlatform /all /norestart
```

Unter der Verwendung des „Updatepaket für den WSL2-Kernel“ das durch Microsoft gestellt wird kann man nun die WSL Version Updaten und zur Standard Version des Systems erklären.

```
wsl --set-default-version 2
```

Nun die Installation einer Linux-Verteilung, wie bereits ober geschildert Verwenden wir die Version Ubuntu LTS 18.02 die über den Microsoft Store erhältlich ist. Bei der Installation der Verteilung müssen nun eine Benutzerkonto und Kennwort erstellt werden.

Docker Desktop wird über die von Docker gestellte Installationsdatei durchgeführt nach dem Installationsprozess muss nun in den Einstellungen von Docker Desktop unter dem Menü Kubernetes die Virtualisierung eines Clusters ermöglicht werden.



## Minikube

Die Installation von Minikube auf Windows beginnt mit dem [Download der Installationsdatei](#) „minikube-installer.exe“, nach erfolgreichem Download führt man diesen aus und Installiert Minikube.

## kubectl

Die Installation von kubectl erfolgt über den folgenden Kommandozeilen Befehl

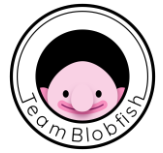
```
curl -LO https://dl.k8s.io/release/v1.21.0/bin/windows/amd64/kubectl.exe
```

der curl Befehl ist nicht nativ durch Windows unterstützt jedoch wird er in PowerShell zu „InvokeWebRequest“ forwarded, wenn diese Vorgehensweise nicht gewünscht ist, kann der Befehl auch als nativer bash Befehl über die Git Bash für Windows aufgeführt werden.

## kompose

Das letzte Tool das Benötigt wird ist ein Übersetzter Tool das die vorhanden docker-compose Datei in ein von Kubernetes verwendbares Manifest umwandelt. Die Installation von Kompose folgt der gleichen Prozedur wie kubectl und erfolgt über den folgenden Kommandozeilen Befehl

```
curl -L https://github.com/kubernetes/kompose/releases/download/v1.22.0/kompose-windows-amd64.exe -o kompose.exe
```



## Step by Step Anleitung des Deployment

Nach Installation der oben genannten Software, werden nun alle nachfolgenden Befehle in der Windows PowerShell ausgeführt. Kommandobefehle werden in blau und kursiv hervorgehoben.

Nr.	Schritt	Intention
1	Docker Desktop starten	Virtualisierungsumgebung für Minikube starten
2	<i>minikube start</i>	Initialisierung des Kubernetes Standard Clusters
3	<i>minikube docker-env</i>	Docker Images für Minikube „sichtbar“ machen
4	In dem Projekt Workspace, <i>docker-compose up</i>	Docker Container Build ausführen
5	<i>minikube ssh</i>	Login in die Minikube Umgebung zum Debuggen
6	<i>docker images</i>	Docker Images, die für Minikube sichtbar sind, anzeigen
7	<i>kompse convert -f docker-compose.yml -o kubernetes-kompse.yaml</i>	Kompose manifest erstellen
8	kubernets-kompse.yaml in einem Editor der Wahl öffnen	Das Manifest muss auf Vollständigkeit überprüft werden
9	In den 3 vorhanden Deployment Objekte die „ImagePullPolicy“ auf „IfNotPresent“ ändern	Sollte dem Deployment ein Image fehlen ist es so in der Lage dieses Nachträglich selbstständig zu pullen.
10	Die Services für Java und Angular hinzufügen.	Jedes Deployment unseres Projektes benötigt einen Service der im Anschluss durch den „reverse-proxy“ Service aufgefasst werden kann.
11	Abschließende Änderung am Manifest ist das Aktualisieren der NetworkPolicy apiVersionen auf „networking.k8s.io/v1“	Ohne diese Änderung ist es nicht möglich das die Services miteinander kommunizieren
12	Nach allen Anpassungen hat die Manifest Datei eine ungefähre Länge von 210 Zeilen.	
13	<i>kubectl apply -f kubernetes-kompse.yaml</i>	Das Manifest wird nun auf das Kubernetes-Cluster angewendet
14	<i>minikube dashboard</i>	Überprüfen des aktuellen Status des Clusters
15	<i>minikube service reverse-proxy</i>	Startet ein Forwarding Tunnel auf die Startseite des Projektes „My-Thai-Star“