

Kubernetes (Mini)Kube

Service Freigabe mit load balancer & ingress

Lothar Schulz

Jahrestagung der Gesellschaft für Informatik (GI)-Fachgruppe Architekturen 2017
20.06.2017 Leipzig



<https://creativecommons.org/licenses/by-nd/4.0/>

Agenda

— — —

- Container
- Container Management / Orchestrierung
- Kubernetes
- Minikube
- LB & Ingress
- Demo
- Developer Happiness

Container

— — —

- leichtgewichtige Virtualisierung unter Linux
- kein Hypervisor und Gastbetriebssystem nötig
- Container teilen sich Kernel Ressourcen
 - cgroups
 - namespaces
 - etc...

Container

— — —

- Stapel von Schichten, der alle nötigen Ressourcen und Umgebungen zum Betreiben von Applikation beinhaltet
 - Libs
 - Apps
 - Runtime Environments
 - Env Variablen
 - ...

Container Management / Orchestrierung

— — —

- Unterstützung beim Betreiben von sehr vielen Containern
- Beispiel Kubernetes:
 - [Kubernetes is an open-source system for automating deployment, scaling, and management of containerized applications.](#)
 - Support für alle gängigen Cloud Provider und bare metal
- [andere Vertreter](#)

Kubernetes

— — —

- **Pods**

- Gruppe von Containern die sich Ressourcen (z.B. Speicher) teilen
- Pod entspricht einer Applikation mit ggf. mehreren (Micro)Services
- Docker Beispiel: namespaces und volumes werden geteilt

- **Services**

- logische Einheit von Pods welche definiert wie die Pods angesprochen werden können

Kubernetes

— — —

- **Deployment**
 - Deklarative Updates für Pods und ReplicaSets
 - Roll back von Deployments, Skalieren von Deployments
- **Namespaces**
 - virtuelle cluster im gleichen physischen cluster

Minikube

— — —

- Kubernetes auf dem lokalen Rechner

Kubernetes Loadbalancer / Ingress

— — —

- Loadbalancer
 - gibt Kubernetes Services im Internet frei
 - z.B. Kubernetes Service type 'loadbalancer'
 - nutzt den von Kubernetes Provider konfigurierten Load Balancer
- Ingress
 - Sammlung von Regeln um Internet Verkehr an Services zu leiten

```
internet
  | | |
  | ingress |
---|---|---
    services
```

Demo

Voraussetzungen

- <https://github.com/lotharschulz/akkahttp-playground/releases/tag/0.0.6-minikube-lb-ing-presentation-leipzig>
- Setup: scala, sbt, docker, minikube, kubectl, local registry, docker images in der lokalen registry

Demo

— — —

Schritte:

- `kubectl create -f minikube-deployment-config.yaml`
- `kubectl expose deployment akkahttpplayground-deployment --type="LoadBalancer" --port=8181 -target-port=8181`
- `minikube service akkahttpplayground-deployment` (öffnet Browserfenster)
- `minikube addons enable ingress`
- `kubectl create -f minikube-ingress.yaml`
- `echo "$(minikube ip) myminikube.info" | sudo tee -a /etc/hosts`
- check
 - `http://myminikube.info/`
 - `http://myminikube.info/hello`

Erfahrungen

- Loadbalancer Setup & Änderung dauert bei Cloud Providern eher länger (Sekunden/Minuten) – DNS Änderungen brauchen Zeit
- Loadbalancer Beschreibung ist Teil der Service-Beschreibung
- Ingress Setup ist in allen Fällen schneller anzulegen und zu adaptieren
- Ingress hat eine eigene Beschreibung als Kubernetes Manifest
- SSL setup ist mit beiden möglich

Developer Happiness

— — —

- Minikube erlaubt Entwicklern lokal Erfahrungen mit Kubernetes zu sammeln
- Minikube Cluster Anpassung ist sehr ähnlich wie Kubernetes Cluster Anpassung
- Minikube hat kaum Unterschiede zum wahrscheinlichen Produktions Setup (Zertifikate)
- Verifizierung von Anforderungen an Container Orchestrierung kann im ersten Schritt mit Minikube erfolgen ohne ein Datacenter oder Cloud Ressourcen verwenden zu müssen

Vielen Dank

— — —

`mail@lothar-schulz.info`

`@lothar_schulz`

`gh.com/lotharschulz/akkahttp-playground`

ausgewählte Links

— — —

<https://kubernetes.io/docs/tutorials/stateless-application/hello-minikube/#create-a-deployment>

<https://stackoverflow.com/questions/39850819/kubernetes-minikube-external-ip-does-not-work>

<https://medium.com/@0skarr3/setting-up-ingress-on-minikube-6ae825e98f82>

backup slides

— — —

```
minikube start
```

```
minikube docker-env
```

```
eval $(minikube docker-env)
```

```
docker run -d -p 5000:5000 --name registry registry:2
```

```
cd base/docker/java/
```


backup slides

— — —

```
docker build -t localhost:5000/java08:0.0.5 -f Dockerfile . && docker push localhost:5000/java08:0.0.5
```

```
cd ../scala/
```

```
docker build -t localhost:5000/scala:0.0.5 -f Dockerfile . && docker push localhost:5000/scala:0.0.5
```

```
cd ../../..
```

```
sbt -DdockerOrganization=info.lotharschulz -DdockerName=akkahttp-playground  
-DdockerBImage=localhost:5000/scala:0.0.5 -DdockerRepo=localhost:5000  
-DartefactVersion=0.0.5 -DversionInDocker=0.0.5 -DdockerPackageName=akkahttp-playground docker:publishLocal  
&& docker:publish
```

```
docker login localhost:5000
```

```
kubectl create secret docker-registry myregistrykey --docker-server=localhost:5000 --docker-username=admin1  
--docker-password=[xxxxxxx] --docker-email=yyyyy@zzzzz.aa
```

backup slides

— — —

```
(kubectl create -f pod-config.yaml
```

```
kubectl describe pod akkahttpplaygroundname
```

```
kubectl logs akkahttpplaygroundname
```

```
kubectl exec -it akkahttpplaygroundname bash
```

```
    curl http://localhost:8181/hello
```

```
    curl http://localhost:8181/
```

```
)
```

backup slides

— — —

```
kubectl create -f minikube-deployment-config.yaml --record=true
```

```
kubectl get deployments
```

```
kubectl rollout status deployment/akkahttpplayground-deployment
```

```
kubectl describe deployment/akkahttpplayground-deployment
```

```
kubectl get rs
```

```
kubectl describe rs akkahttpplayground-deployment-4161283909
```

```
kubectl get po
```

```
kubectl describe po akkahttpplayground-deployment-4161283909-sqk1q
```

```
kubectl get svc
```

```
kubectl describe svc akkahttpplayground-deployment
```

backup slides

— — —

```
kubectl expose deployment akkahttpplayground-deployment --type="LoadBalancer" --port=8181 --target-port=8181
```

(external-IP pending forever in

```
kubectl describe svc akkahttpplayground-deployment &
```

```
kubectl get svc akkahttpplayground-deployment)
```

```
minikube service akkahttpplayground-deployment
```

(opens a browser with IP e.g. `http://192.168.99.100:31148` `http://192.168.99.100:31148/hello`)

backup slides

— — —

```
kubectl proxy
```

```
minikube addons enable ingress
```

```
> ingress was successfully enabled
```

```
minikube addons list
```

```
kubectl create -f minikube-ingress.yaml
```

```
kubectl describe ing akkahttpplayground-ingress
```

```
kubectl get ing akkahttpplayground-ingress
```

```
echo "$(minikube ip) myminikube.info" | sudo tee -a /etc/hosts
```

```
check
```

```
http://myminikube.info/
```

```
http://myminikube.info/hello
```

```
kubectl get deploy,ev,ing,jobs,po,pv,rs,svc,statefulset
```