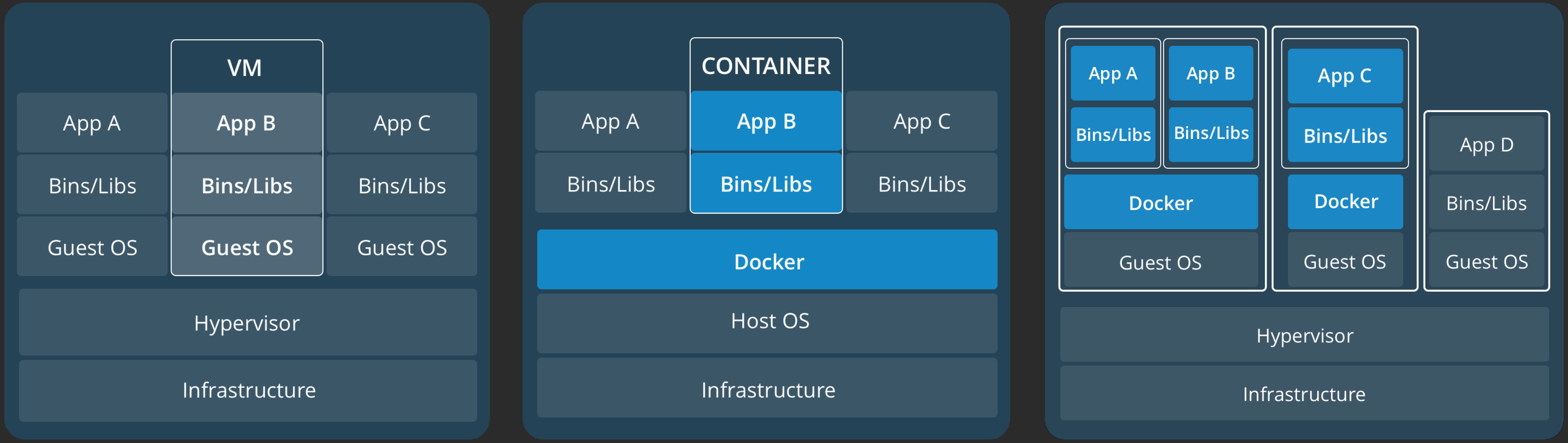
// PLUNGING INTO KUBERNETES — AN INTRODUCTION

JOHANNES SCHNATTERER

CLOUDOGU GMBH

VERSION: 202007021036-474E1F3

Container Recap



🌐 https://www.docker.com/what-container

Operations on "plain" Docker host in production — what is missing?

- For some use cases: Nothing!
- For others:
  - High availability
  - Load Balancing
  - Solutions for challenges of distributed systems, e.g networking, storage
  - Scaling out containers
  - Rolling updates

Solution:

Container orchestrator

Kubernetes (k8s)

You will see the following features hands-on:

- scheduling containers on multiple nodes
- scaling out (scaling horizontally)
- load balancing
- self healing
- rolling updates

🏋️ Cluster access

```
# Start container with all tools necessary
$ docker run -it cloudogu/k8s-training

# Create cluster config
$ k8s-training-auth fdt 2020

# Test connection: no error means success
$ kubectl version
```

First deployment

```
$ NAME=think-of-something-unique

$ kubectl create deployment $NAME --image=cloudogu/hello-k8s

# Success?
$ k get deployment $NAME
```

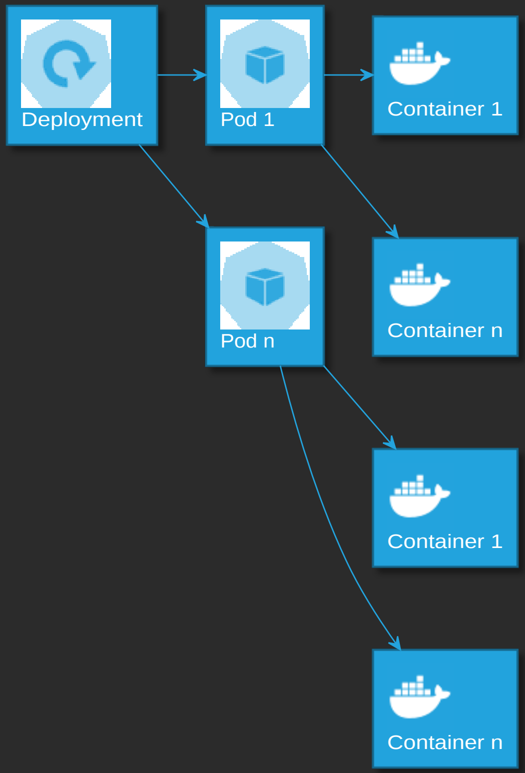
Access via the internet

```
k expose deployment $NAME --port=80 --target-port 8080 --type=LoadBalancer

# Query EXTERNAL-IP, then open in browser
k get service $NAME
```

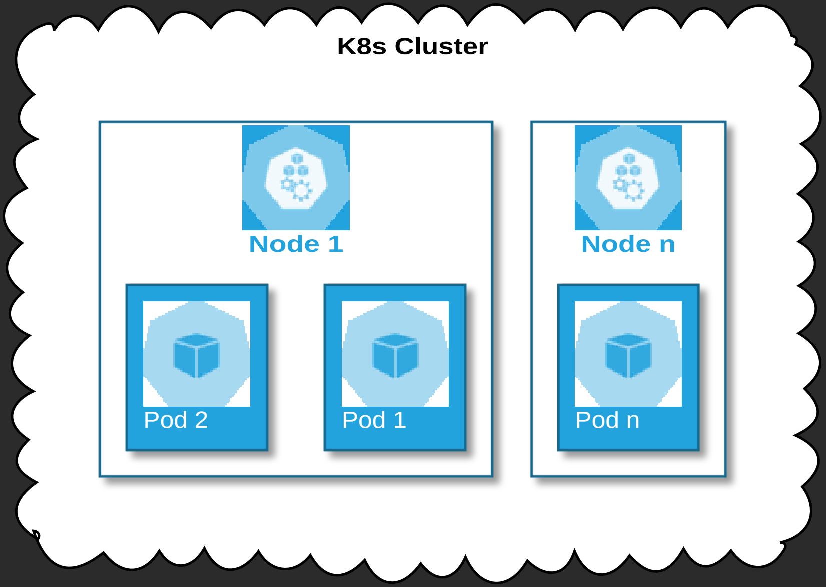
🌐 https://i.giphy.com/media/z9sFrQMfEME5a/giphy.webp

Deployment ➜ Pod ➜ Container



```
$ k get pod | grep $NAME
```

Pod ➜ Node



```
$ k get pod -owide
$ k get node
```

High availability?

Scaling out

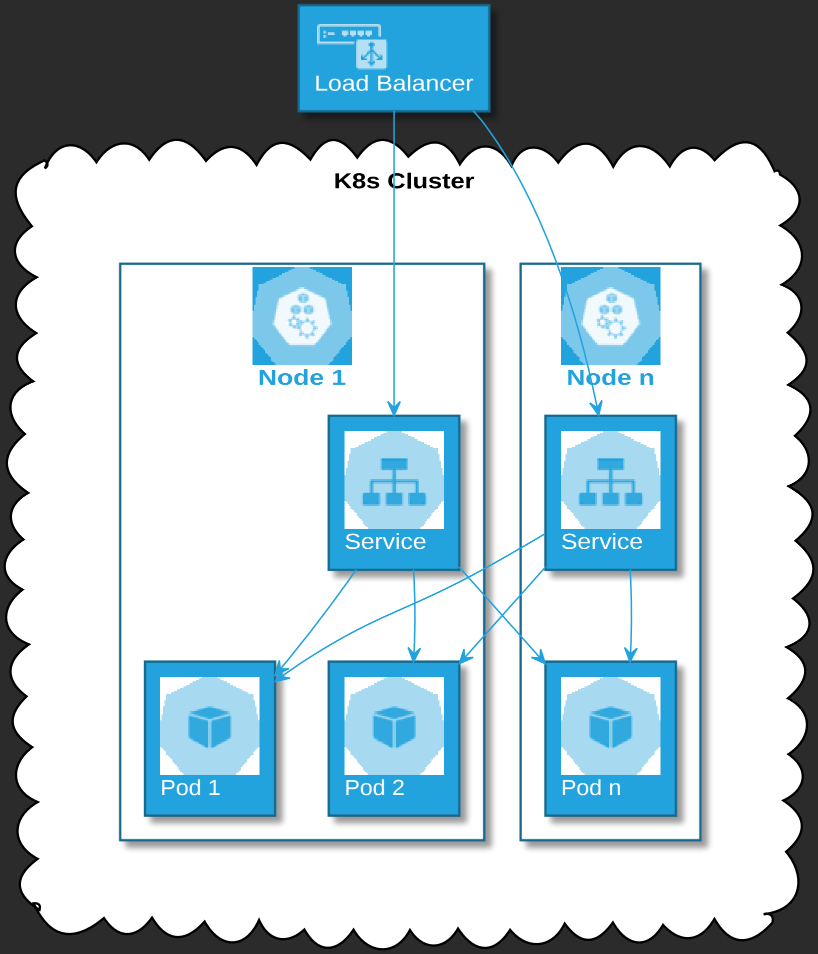
```
$ k scale deployment $NAME --replicas=3
$ k get deployment $NAME
$ k get pod | grep $NAME
```

Load Balancing

- Reload app in browser multiple times (look at "`pod:`")
- or run script:

```
$ EXTERNAL_IP=w.x.y.z
$ while [ 1 ]; do echo $(curl -s http://$EXTERNAL_IP/api/hostName); done
```

Services


Load Balancer
K8s Cluster
Node 1
Node n
Service
Service
Pod 1
Pod 2
Pod n

Self healing

```
$ EXTERNAL_IP=w.x.y.z
$ while [ 1 ]; do echo $(curl -s http://$EXTERNAL_IP/api/hostName); done
```

```
$ k get pod | grep $NAME

$ PODNAME=one-of-your-pods
$ k delete pod $PODNAME

$ k get deployment $NAME
$ k get pod
```

Rolling update

```
$ EXTERNAL_IP=w.x.y.z
$ while [ 1 ]; do
    echo $(curl -s --connect-timeout 1 http://$EXTERNAL_IP/api/appVersion);
  done
```

```
$ k get pod | grep $NAME
$ kubectl set image deploy $NAME hello-k8s=cloudogu/hello-k8s:1.9.1
$ watch "kubectl get pod | grep $NAME"
```

END

Key takeaways

- **k8s** - run containers in a cluster (on multiple nodes)
- **pod** - smallest resource in k8s (comprising containers)
- **deployment**
  - scaling out containers
  - rolling updates
  - self-healing
- **service**
  - cluster access
  - load balancing

Johannes Schnatterer

Cloudogu GmbH

🌐 my.cloudogu.com
🌐 cloudogu.com/schulungen

💬 Questions or feedback?

🌐 https://forum.cloudogu.com/topic/42

🐦 @cloudogu
🐦 @jschnatterer