



// PLUNGING INTO KUBERNETES — AN INTRODUCTION

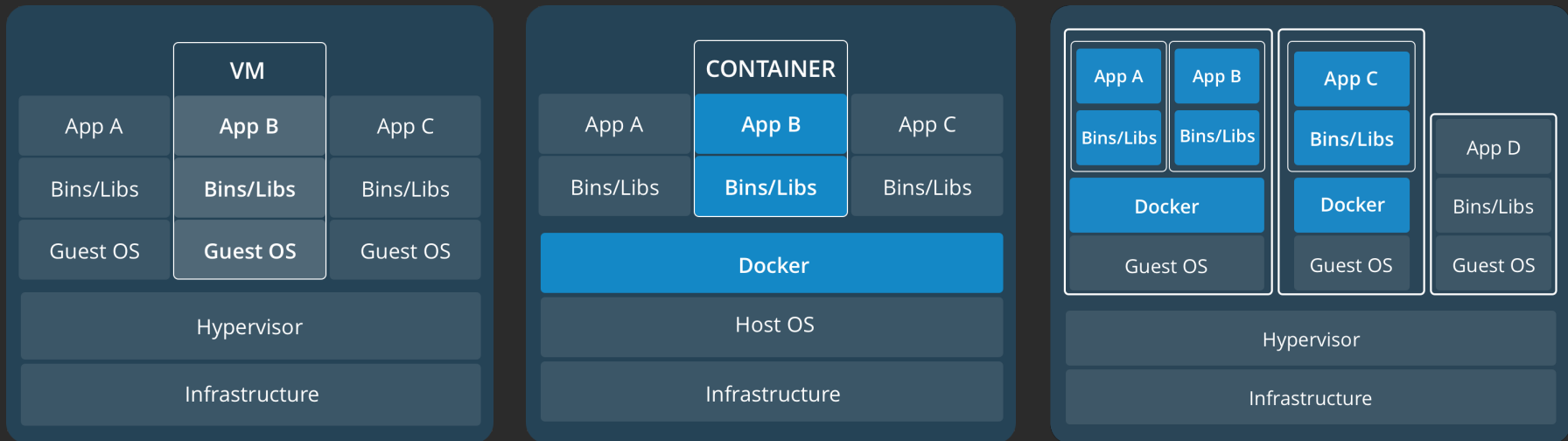
cloudogu

JOHANNES SCHNATTERER
CLOUDOGU GMBH

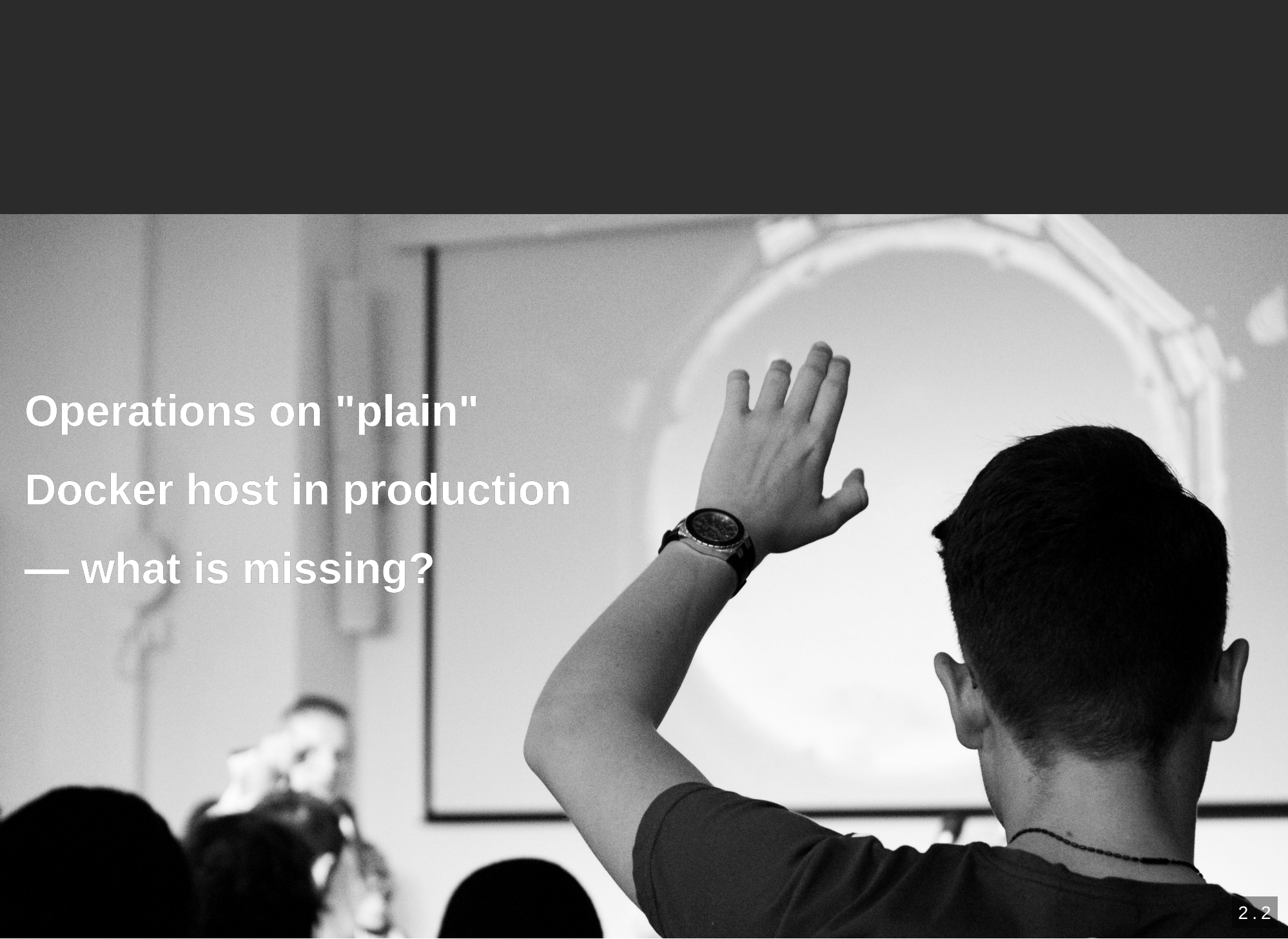
VERSION: 202007061906-EA91F6A



Container Recap



<https://web.archive.org/web/20180701005535/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 clouddogu/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=clouddogu/hello-k8s  
  
# Success?  
$ k get deployment $NAME
```

Access via the internet

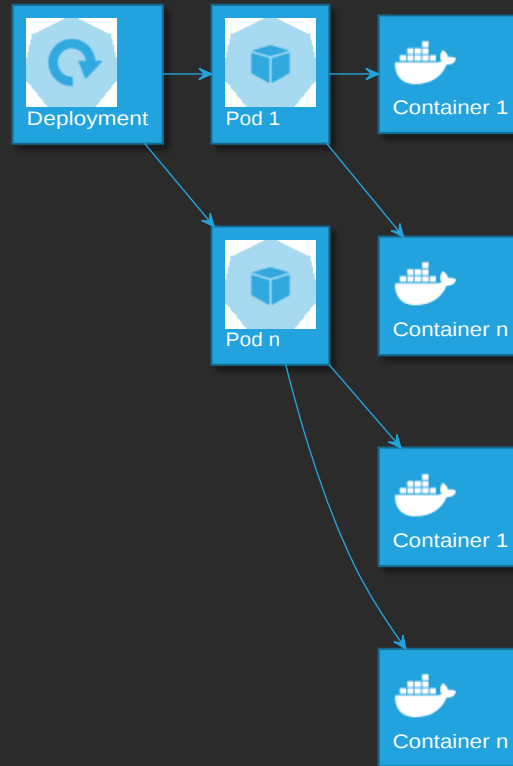
```
k expose deployment $NAME --port=80 --target-port 8080 --type=LoadBalancer  
  
# Query EXTERNAL-IP, then open in browser  
k get service $NAME
```



<https://media.giphy.com/media/z9sFrQMfEME5a/giphy.gif>

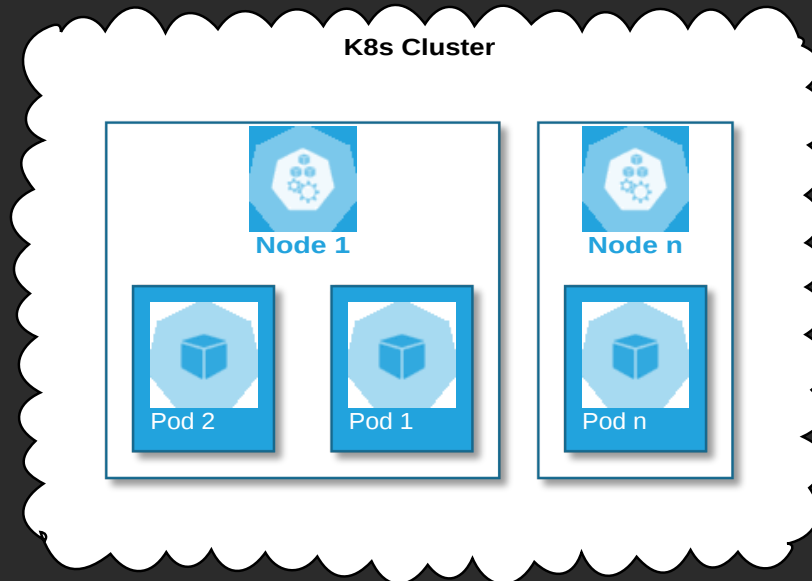


Deployment → Pod → Container



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

Pod → Node



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


The image shows a vast quantity of identical yellow rubber ducks. They are arranged in neat, diagonal rows across multiple shelves or tiers. The perspective is from a low angle, looking down the length of the rows, which creates a strong sense of depth and repetition. The ducks are all the same size and color, with a bright yellow body, a red beak, and a small blue eye. The lighting is warm and even, highlighting the smooth texture of the rubber. The overall composition emphasizes the sheer volume and uniformity of the objects.

High availability?

Scaling out

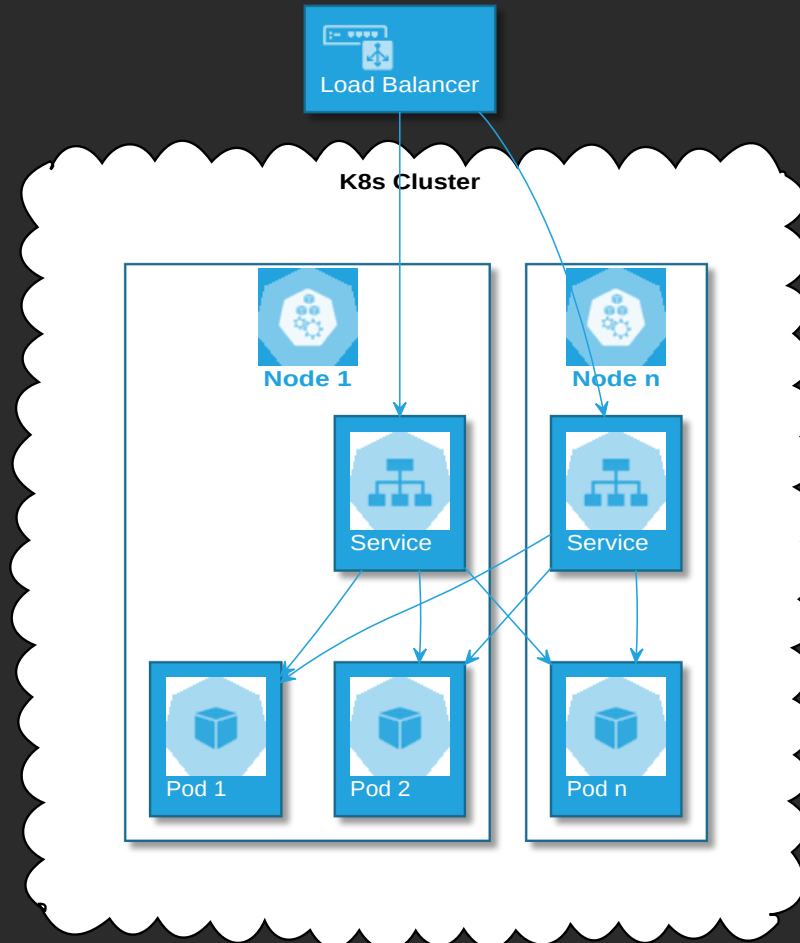
```
$ k scale deployment $NAME --replicas=3  
$ k get deployment $NAME  
$ k get pod | grep $NAME
```

Load Balancing

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

```
# k get service $NAME  
$ EXTERNAL_IP=w.x.y.z  
$ while [ 1 ]; do echo $(curl -s http://$EXTERNAL_IP/api/hostname); done
```

Services



Self healing

```
# New terminal (or use tmux)
$ docker run -it cloudogu/k8s-training
$ k8s-training-auth fdt 2020

# k get service $NAME
$ EXTERNAL_IP=w.x.y.z
$ while [ 1 ]; do echo $(curl -s http://$EXTERNAL_IP/api/hostname); done

$ k scale deployment $NAME --replicas=2
$ k get pod | grep $NAME

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

$ k get pod | grep $NAME
```

Rolling update

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

```
# Other terminal
$ k scale deployment $NAME --replicas=5
$ k get pod | grep $NAME
$ k set image deploy $NAME hello-k8s=cloudogu/hello-k8s:1.9.1
$ k get pod | grep $NAME
```




Key takeaways

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

Johannes Schnatterer

Cloudogu GmbH

 my.cloudogu.com

 cloudogu.com/schulungen

 **Questions or feedback?**

 <https://forum.cloudogu.com/topic/65>

 [@cloudogu](https://twitter.com/cloudogu)

 [@jschnatterer](https://twitter.com/jschnatterer)




```
#!/usr/bin/env bash BASEDIR=$(dirname $0)  
sed -n -e '/bash/, // p' ${BASEDIR}/03-k8s.md | sed '/bash/d;  
s/^\n---\n/' | tee listings.txt
```