



# // PLUNGING INTO KUBERNETES cloudogu

## – AN INTRODUCTION

JOHANNES SCHNATTERER  
CLOUDOGU GMBH

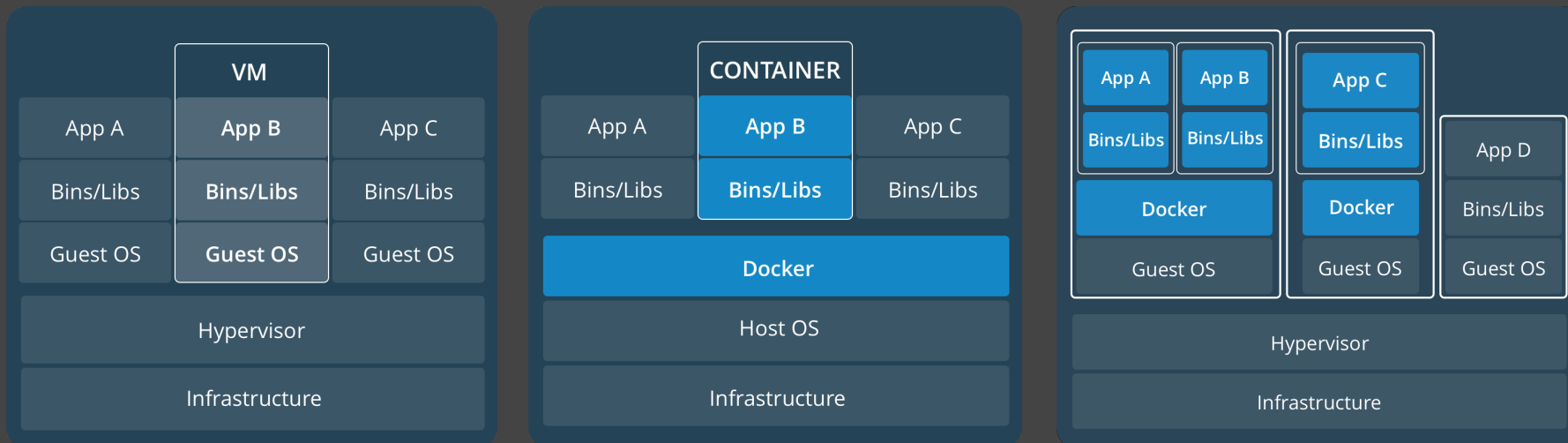
VERSION: 202007171443-97235C5



## Pull image for workshop

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

# 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

```
# In clouddogu/k8s-training container - 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

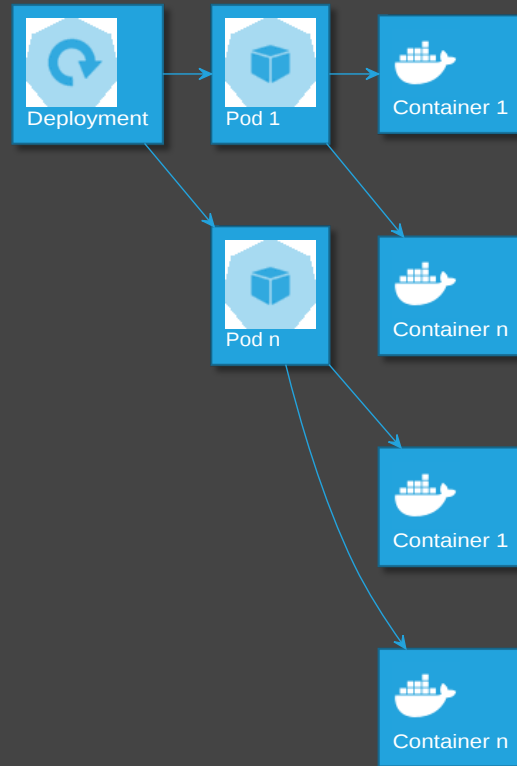
```
k expose deployment $NAME --port=80 --target-port 8080 --type=LoadBalancer  
  
# Query EXTERNAL-IP, then open in browser  
k get service $NAME
```



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

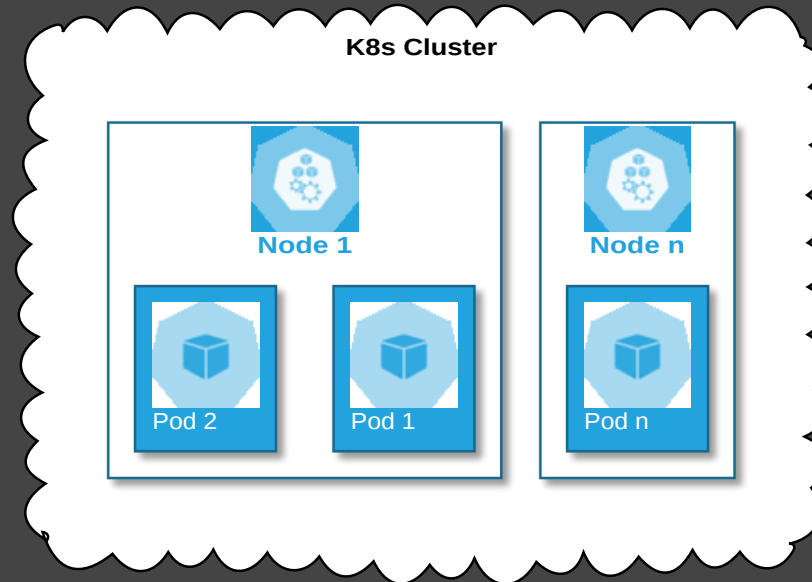


# Deployment → Pod → Container



```
$ 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, parallel rows on multiple levels of yellow shelving. The perspective is from a low angle, looking down the length of the shelves, which creates a strong sense of depth and repetition. The ducks are all facing the same direction, with their red beaks and blue eyes clearly visible. The lighting is bright and even, highlighting the smooth texture of the rubber. The overall effect is one of overwhelming abundance and uniformity.

**High availability?**



# Scaling out

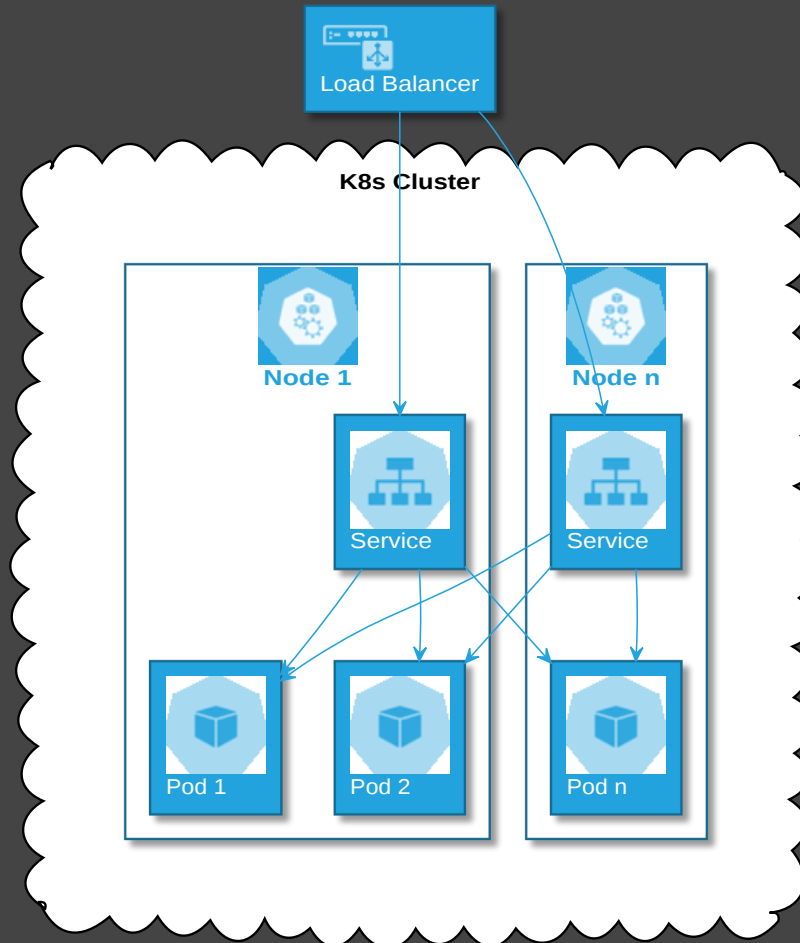
```
$ k scale deployment $NAME --replicas=2  
$ k get deployment $NAME  
$ k get pod | grep $NAME
```

# 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 clouddogu/k8s-training
$ k8s-training-auth fdt 2020

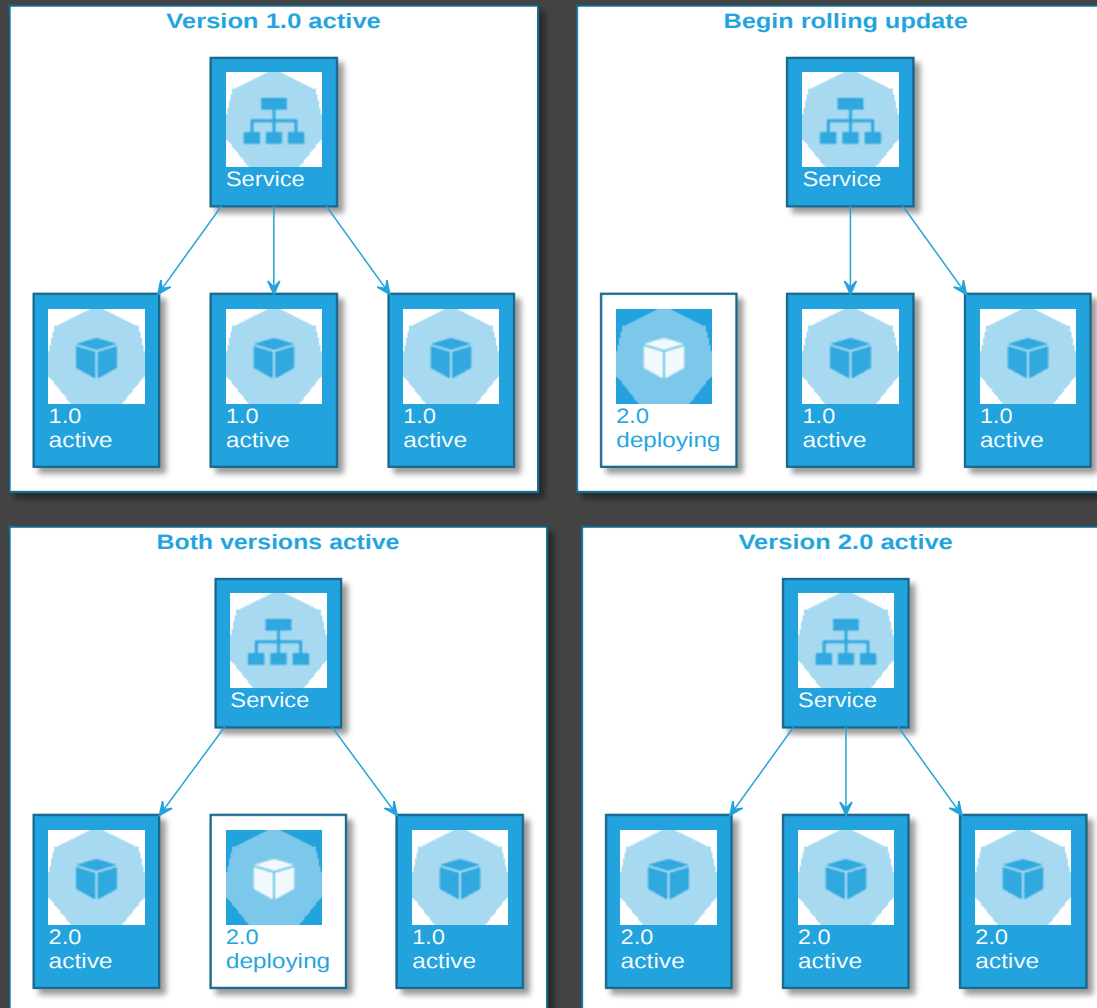
# k get service $NAME
$ EXTERNAL_IP=w.x.y.z
$ while [ 1 ]; do echo $(curl -s http://$EXTERNAL_IP/api/hostname); done
```

```
$ k 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 $(curl -s --connect-timeout 1 -m 1 http://$EXTERNAL_IP/api/appVersion);
done
# -m → max-time
```

```
# Other terminal
$ k get pod | grep $NAME

$ k set image deploy $NAME hello-k8s=cloudogu/hello-k8s:1.9.1

# Multiple times
$ 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](https://my.cloudogu.com)

 [cloudogu.com/schulungen](https://cloudogu.com/schulungen)

 **Questions or feedback?**

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

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

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

