



One of the most unpolished
Kubernetes introduction
presentations ever given on a
Thursday afternoon from
Freiburg, Germany, ever

Martin Danielsson -- 2017-01-05

Haufe-Lexware GmbH & Co. KG, CTO Office

Twitter: @donmartin76, GitHub: DonMartin76

You might have figured,
this is the Kubernetes logo





Thank you for your attention!

Just kidding.



What is Kubernetes?

“Kubernetes is an [open-source platform for automating deployment, scaling, and operations of application containers](https://kubernetes.io/docs/whatisk8s/) across clusters of hosts, providing container-centric infrastructure.”

<http://kubernetes.io/docs/whatisk8s/>

Holy smokes!

Which means?



What does Kubernetes do?

- Provide a runtime environment for Docker containers
- Scale and load balance docker containers
- Abstract away the infrastructure containers run on
- Monitor/health check containers
- Declarative definition for running containers
- Update containers (also rolling updates)
- Storage mounting (allow abstracting infrastructure)
- Service discovery and exposure
- Labelling and selection of any kind of object (we'll get to this)



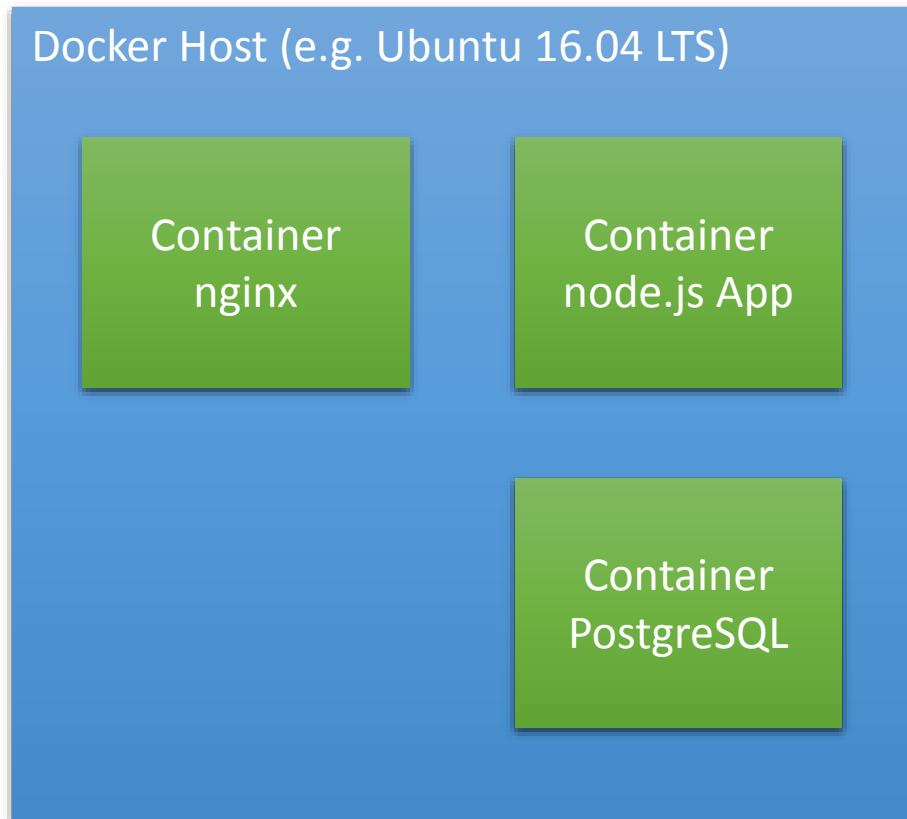
What does Kubernetes fucking not do?

- Provide any additional services than just Docker
- Compile or build your source code (needs images)
- Provide real orchestration, relies on containers working independently
- Mandate or provide any kind of special configuration language
 - You're free to do whatever you want
 - But: You have to find out a way yourself



Kubernetes \cong
General Purpose
Platform-as-a-Service (PaaS)

Recap - Docker?



- Docker containers share Kernel with Host
- Many containers can efficiently run on one Host
- Powerful developer tooling
- Dockerfiles are used to build images
- A Docker Registry can store Docker images
 - Haufe's registry.haufe.io
 - Official hub.docker.com
- Docker Engine used by Kubernetes under the hood
- Kubernetes is alternative to Docker Swarm
 - Older, more mature
 - Used e.g. by GCE (Google Container Engine)



Kubernetes and Docker

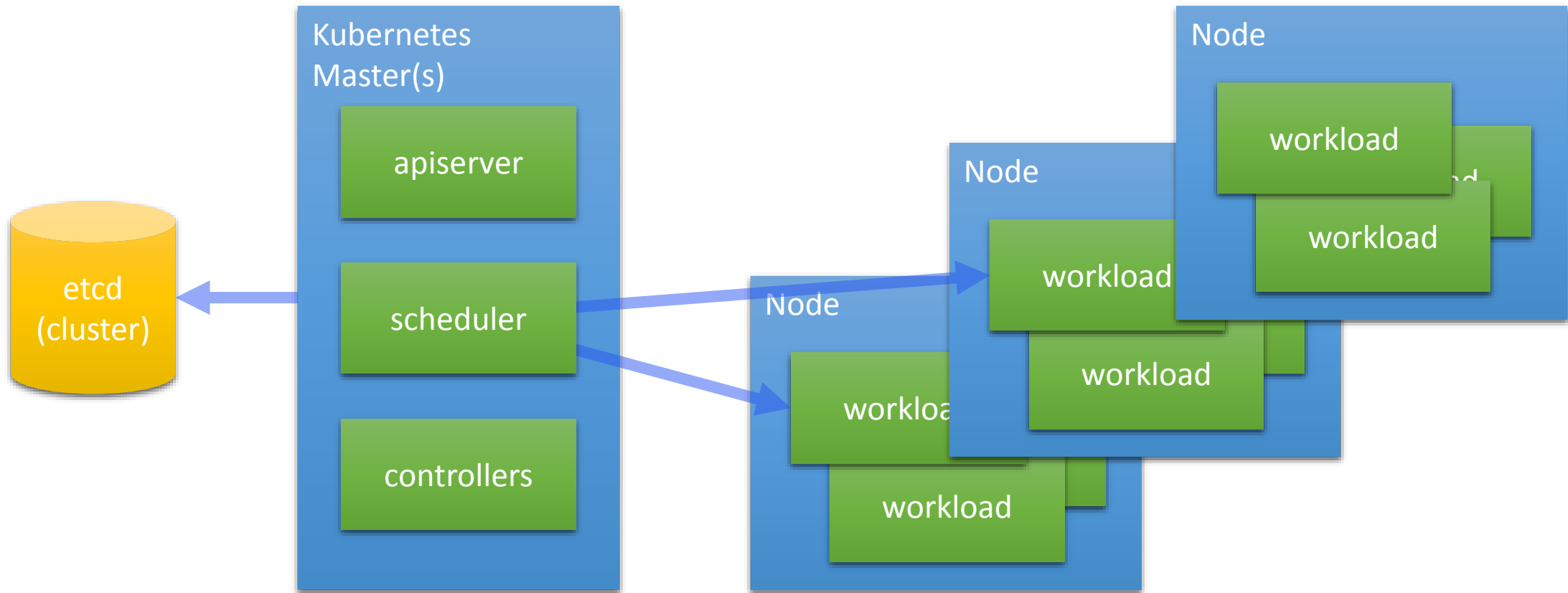
- Kubernetes adds functionality to Docker
- Manages a set of Docker Hosts, forming a Cluster
- Takes care of Container scheduling
- Supervises Docker containers
- Kubernetes is **replacement/alternative** for Docker Swarm



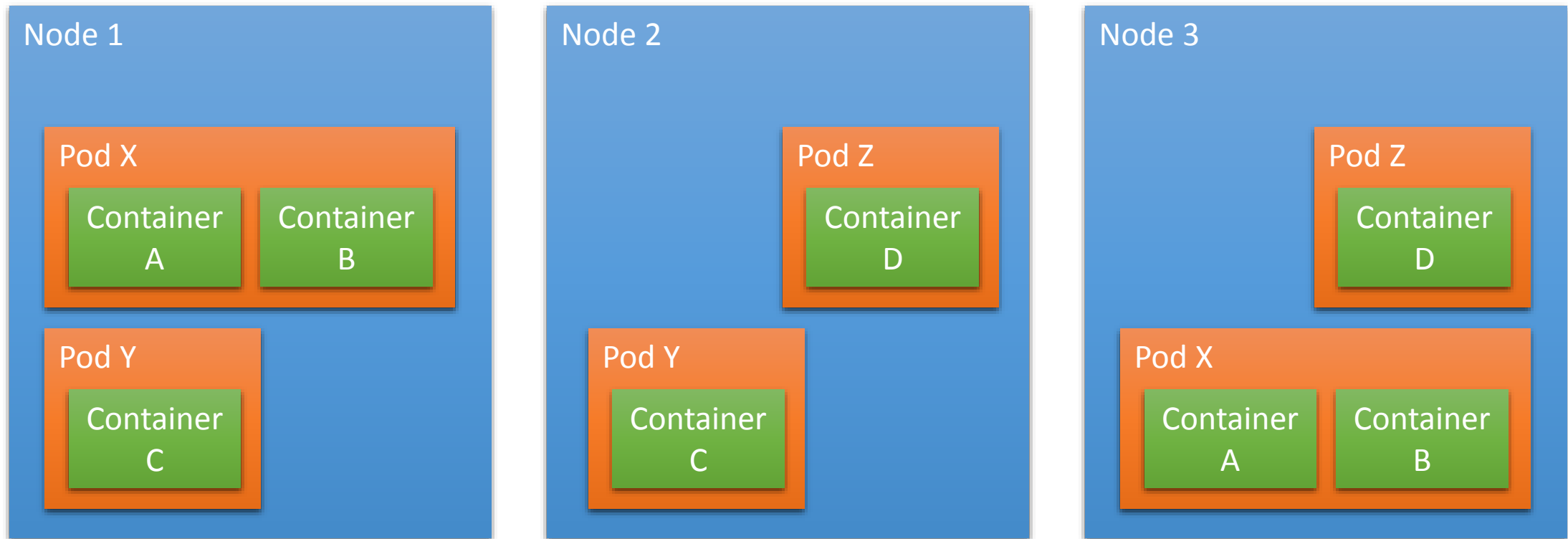
Deployment Architecture

All blue boxes are Docker Hosts (VMs)

Kubernetes Components are also running as stateless containers!

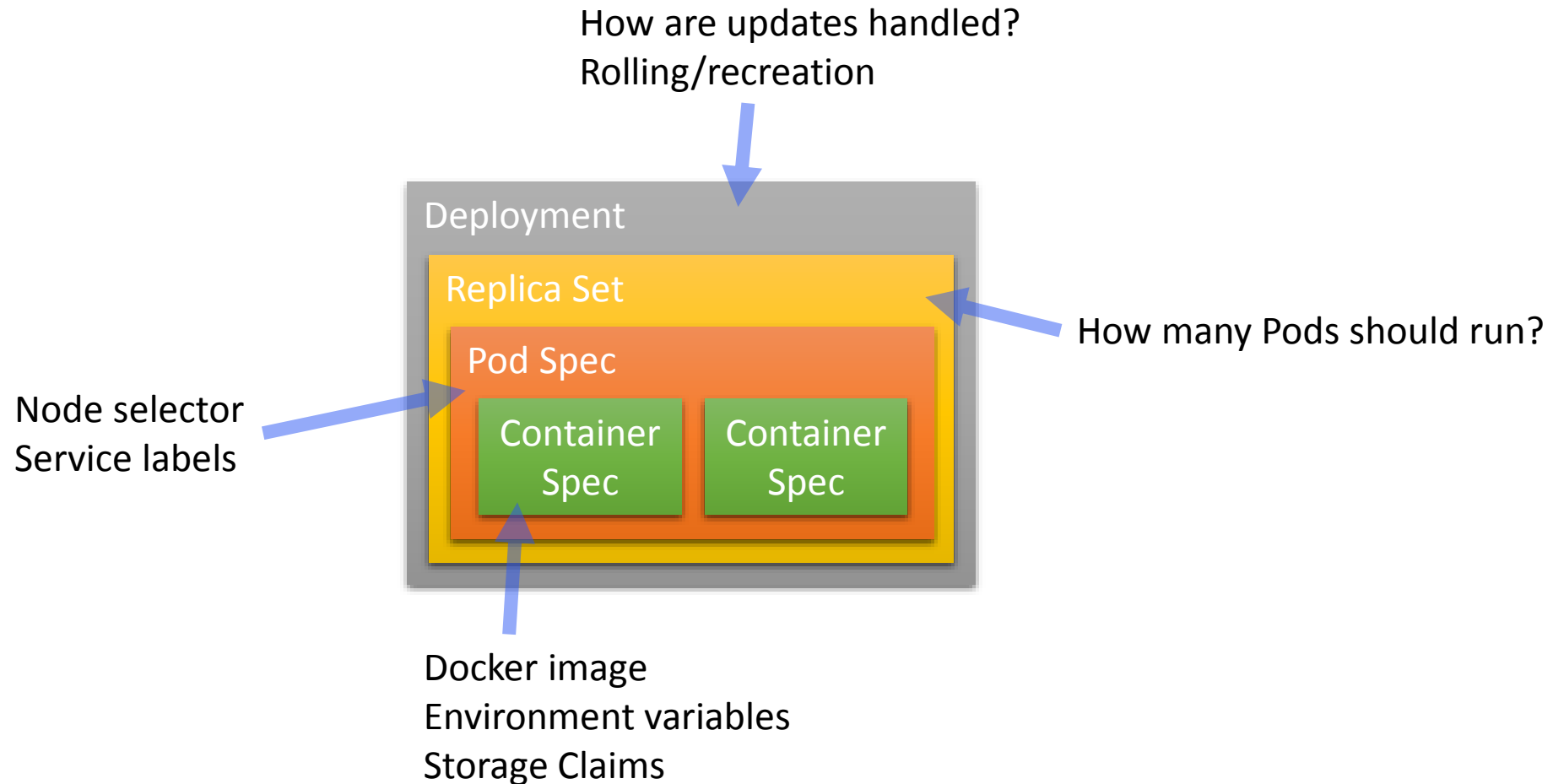


Kubernetes Runtime



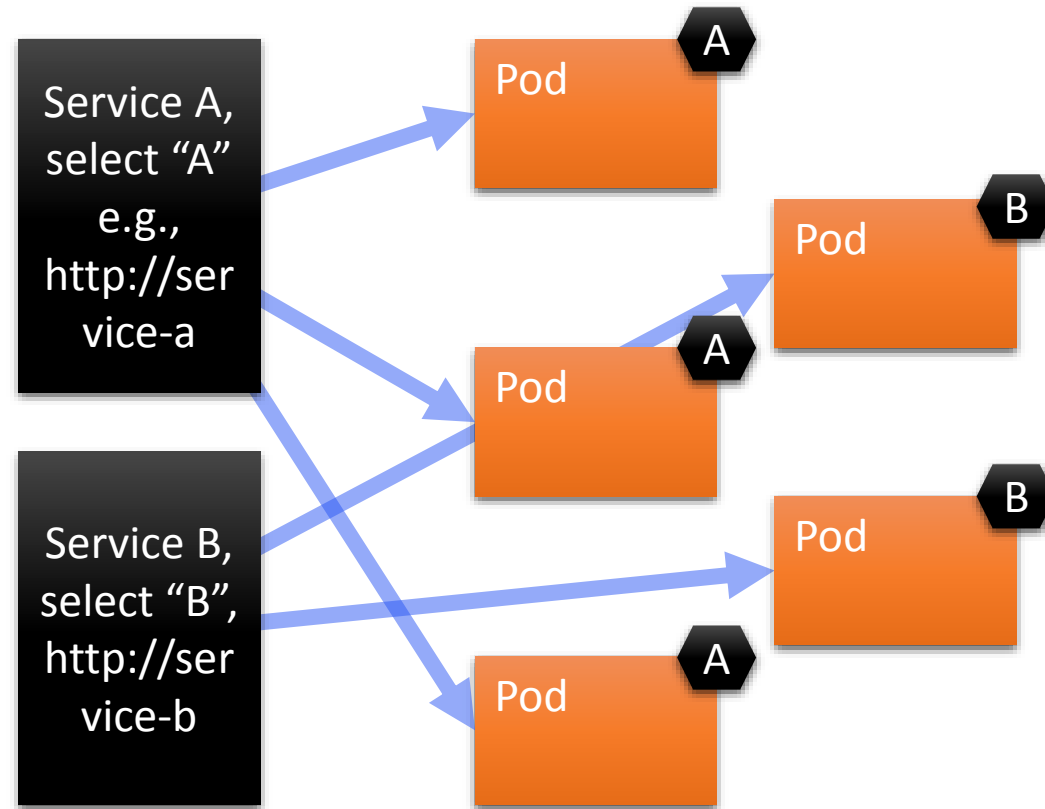


Abstractions (1) - “Boxes in boxes”





Abstractions (2) - Services



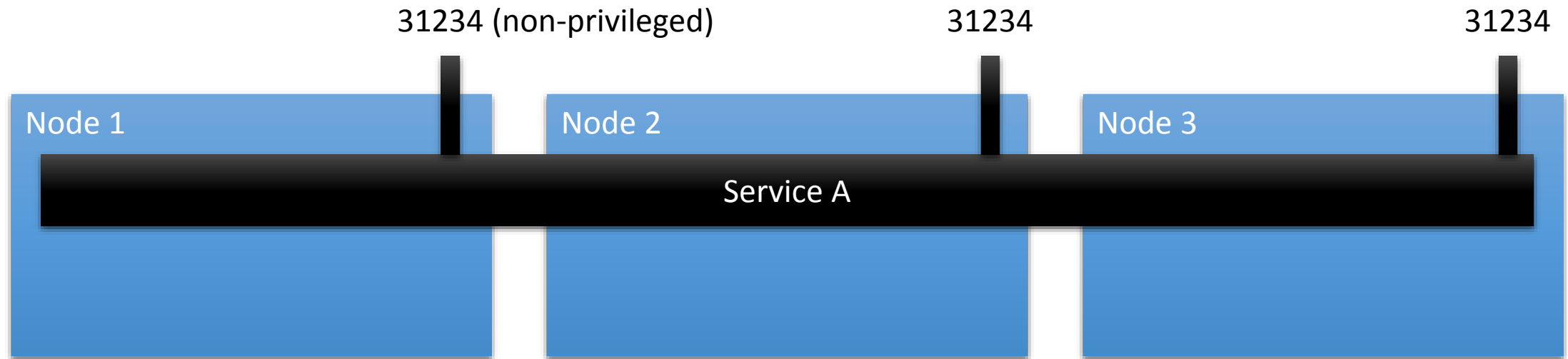


Abstractions (3) - Other things

- Jobs (one-off containers)
- DaemonSets (one container per node)
- StatefulSet (handles lifetime of pods differently)



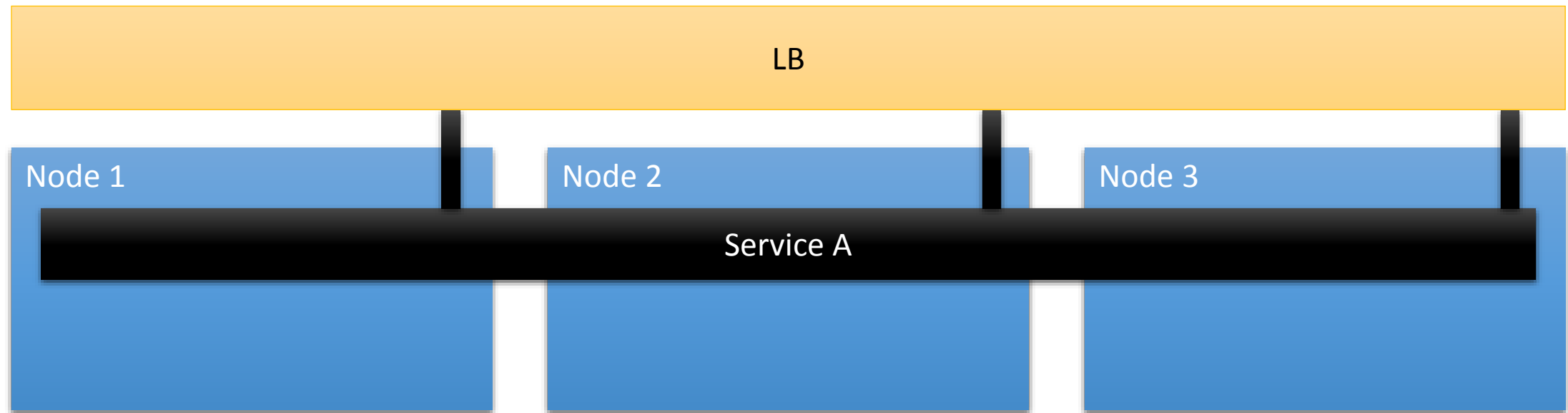
Exposing Services (1) - NodePort



Can be used to put an external Load Balancer in front of a service



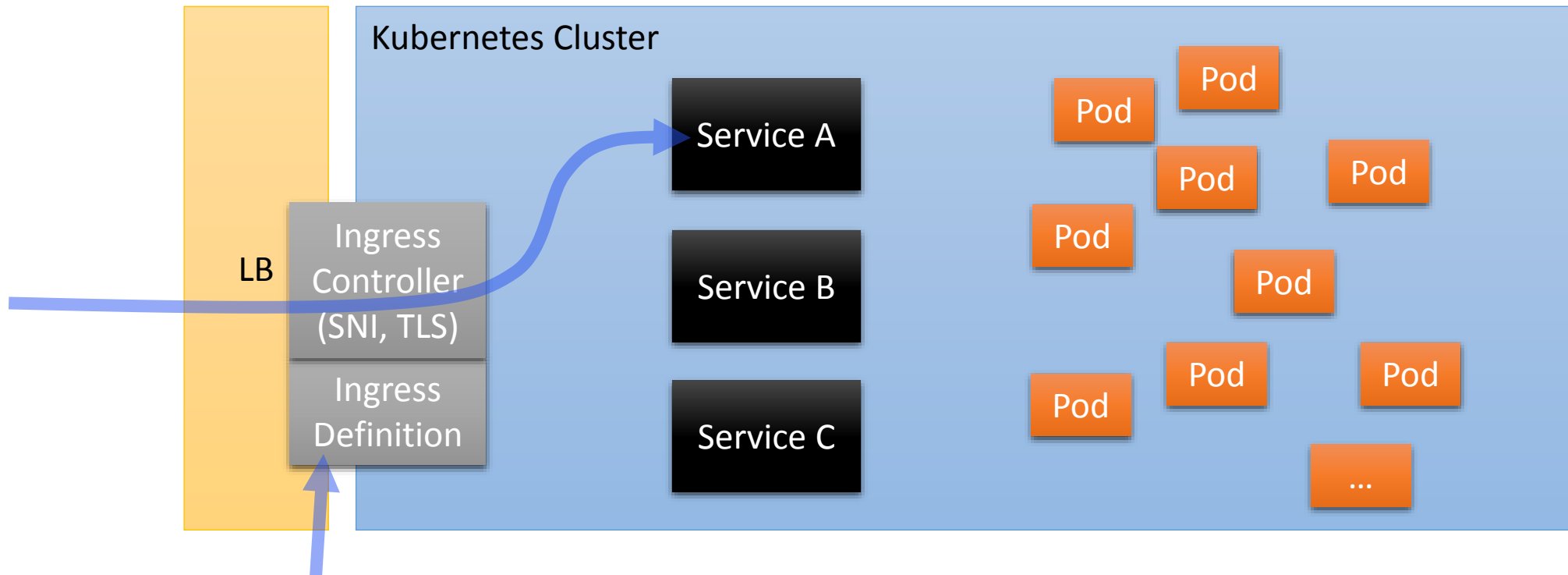
Exposing Services (2) - LoadBalancer



- Depends on Cloud Provider (Azure, AWS,...) how this is done
- Will provision a Load Balancer on the cloud provider's infrastructure
- (e.g. Elastic LB, Azure LB,...)



Exposing Services (3) - Ingress



- E.g., “route Host x.y.z to Service A”, “Use TLS Certificate abc for host x.y.z”
- Abstract definition of rules
- Implemented by Ingress Controller
- Flexible; leverages “LoadBalancer” on cloud provider
- Can provide SNI (Server Name Indication) and TLS termination



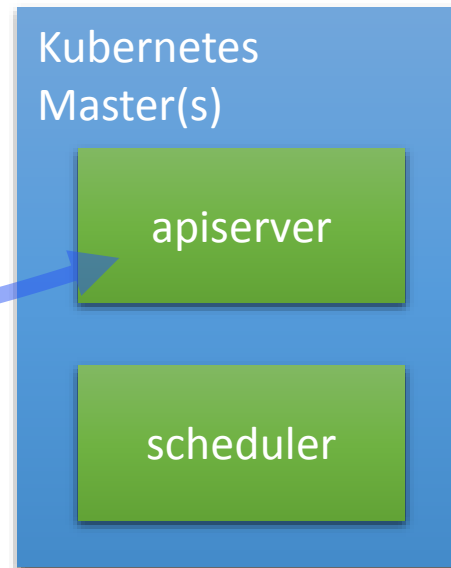
Working with **kubectl**

```
$ kubectl apply -f deployment.yml
Created deployment "nginx"
$
```

- **kubectl** is a convenient way to talk to the Kubernetes API
- Uses **~/.kube/config** for AuthN/Z

~/.kube/config

Authentication/
Authorization





Example **kubectl** YAML file

Deployment

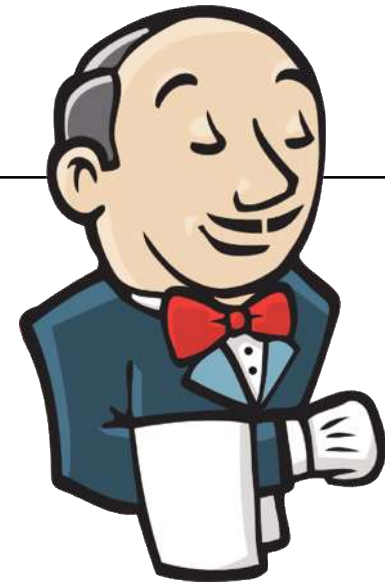
Replica Set

Pod

Container(s)

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: idesk-mobile-2851
spec:
  replicas: 2
  # keep the latest three deployments on k8s
  revisionHistoryLimit: 3
  template:
    metadata:
      labels:
        service: idesk-mobile-2851
        feature: "false"
    spec:
      containers:
      - env:
        - name: API_GATEWAY_HOST
          value: https://api.idesk-apim.haufe-ep.de
        - name: CLIENT_ID
          value: "ad283bd8273bdbe9a72bdef"
        image: "aurora/aurora.mobile.client:v2851"
        name: idesk-mobile
        ports:
        - containerPort: 80
          protocol: TCP
        restartPolicy: Always
        imagePullSecrets:
        - name: aurora-registry-haufe-io
```

```
$ kubectl apply -f idesk-mobile.yml
Created deployment "idesk-mobile-2851"
$ kubectl get po
NAME          READY  STATUS   RESTARTS  AGE
idesk-mobile  1/2    Running  0          10s
$
```



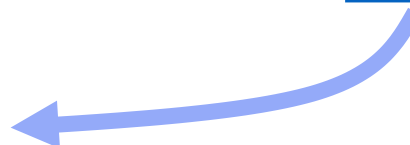


Example Service YML

```
apiVersion: v1
kind: Service
metadata:
  labels:
    service: idesk-mobile-2851
    feature: "true"
  name: idesk-mobile-2851
spec:
  type: NodePort
  ports:
    - name: "https"
      port: 443
      protocol: TCP
      targetPort: 443
      nodePort: 31851
  selector:
    service: idesk-mobile-2851
status:
  loadBalancer: {}
```

Reachable over any Agent IP:

<https://aurora-k8s.haufe-ep.de:31851>





Demo

- Add an nginx Deployment
- Add a service
- Expose service via an Ingress definition
- Kubernetes Dashboard
- Deleting with a selector filter



Example Ingress YAML

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: nginx-demo
spec:
  rules:
  - host: nginx-demo.donmartin76.com
    http:
      paths:
      - path:
          backend:
            serviceName: nginx-demo
            servicePort: 80
```