Red Hat Summit

Connect

OpenShift zum Anfassen

Der Softwarelebenszyklus mit einer Containerplattform

Hands-On Lab 🧑‍💻

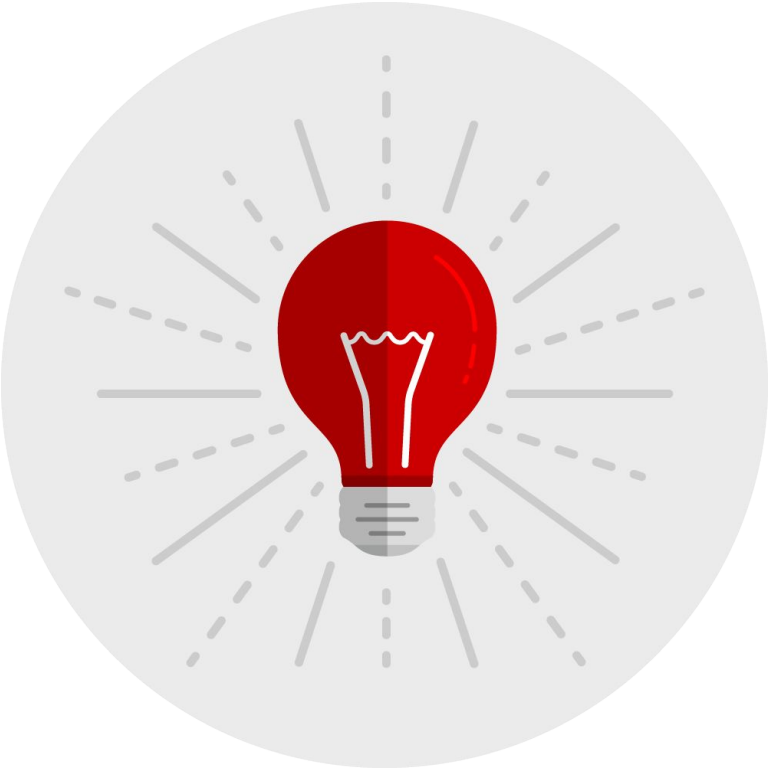
Georg Modzelewski
Specialist Solution Architect

Karsten Gresch
Specialist Solution Architect

AGENDA

11:50 – 13:15

- Why Kubernetes?
- Container Technology
- What is Kubernetes
- Kubernetes Cluster
- Workshop Modules

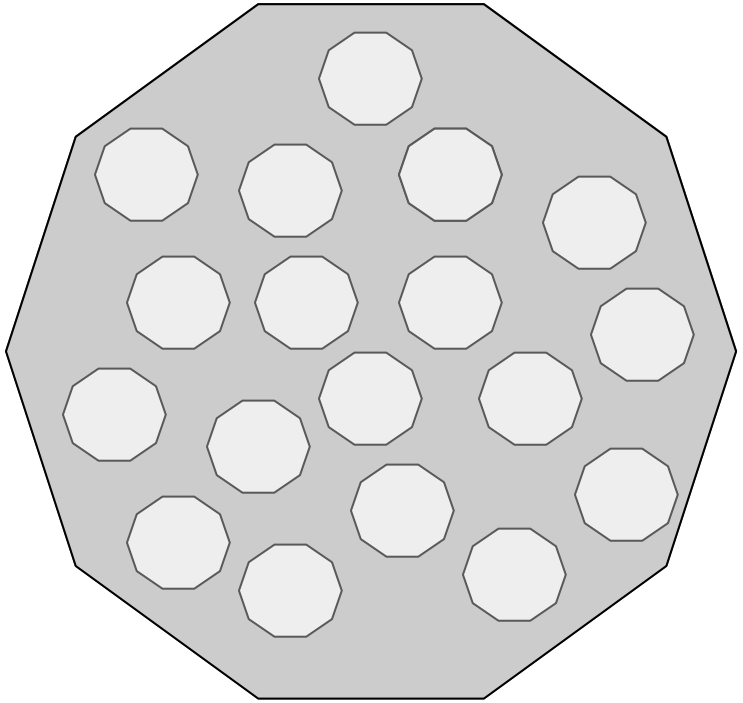


Fragen vorab

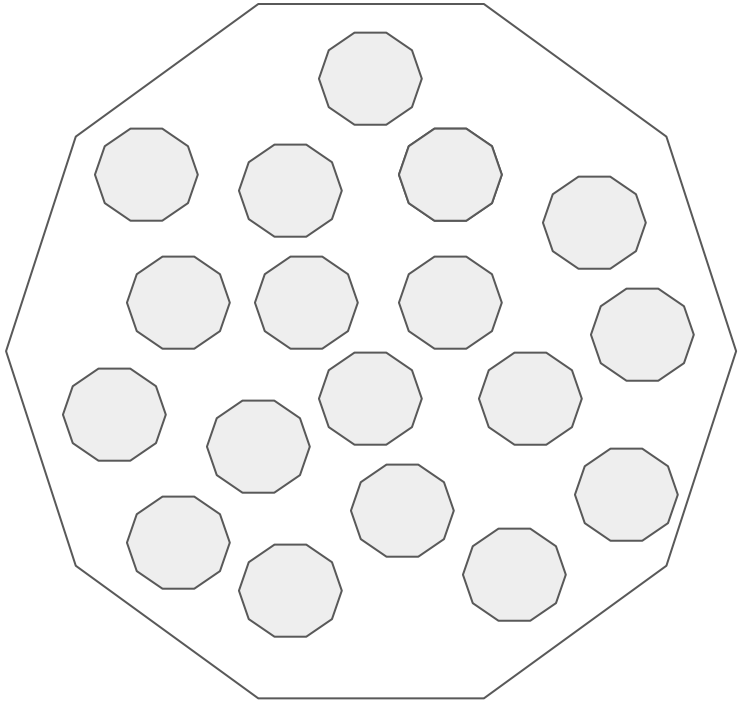
Wer weiß nicht, was ein **Container** ist?

Kubernetes:
was und warum?

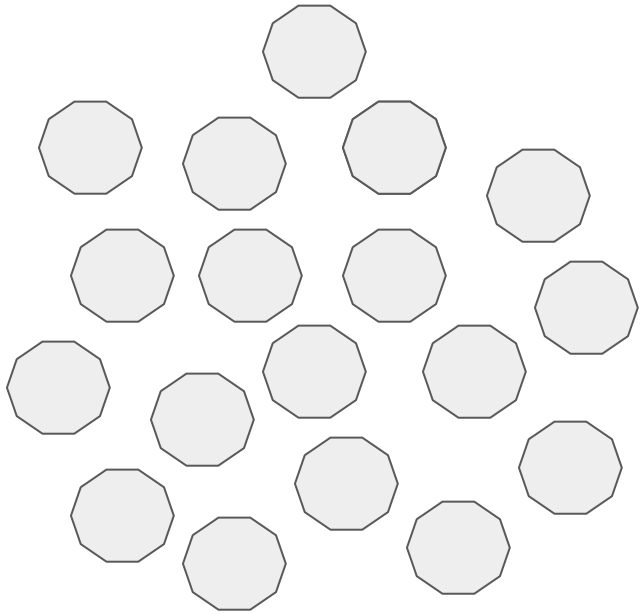
Anwendung



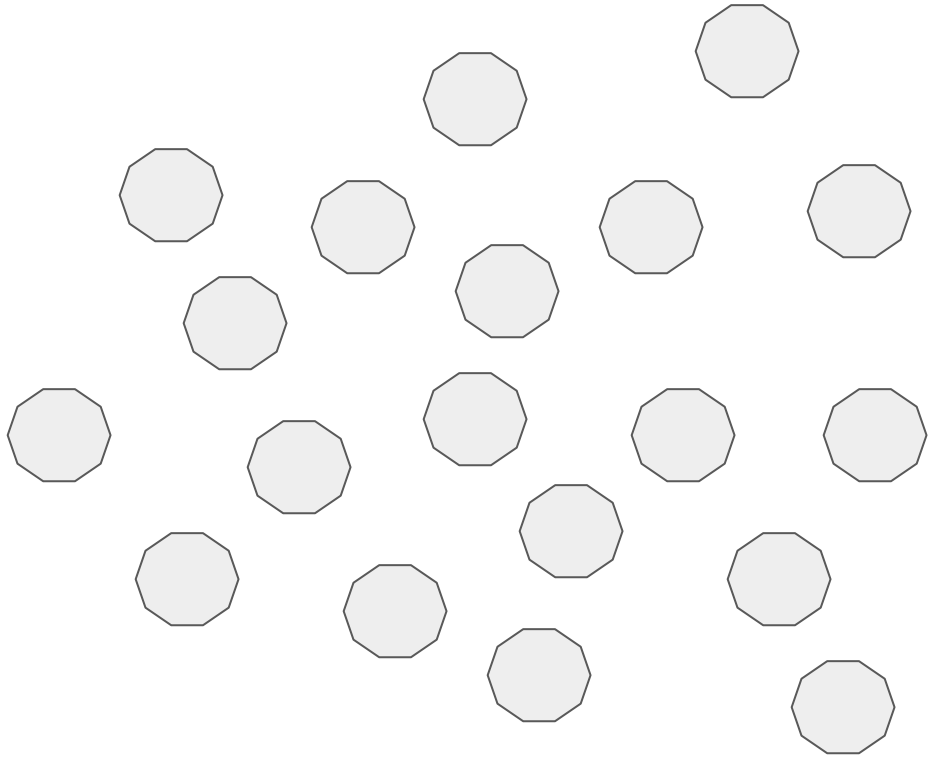
Module



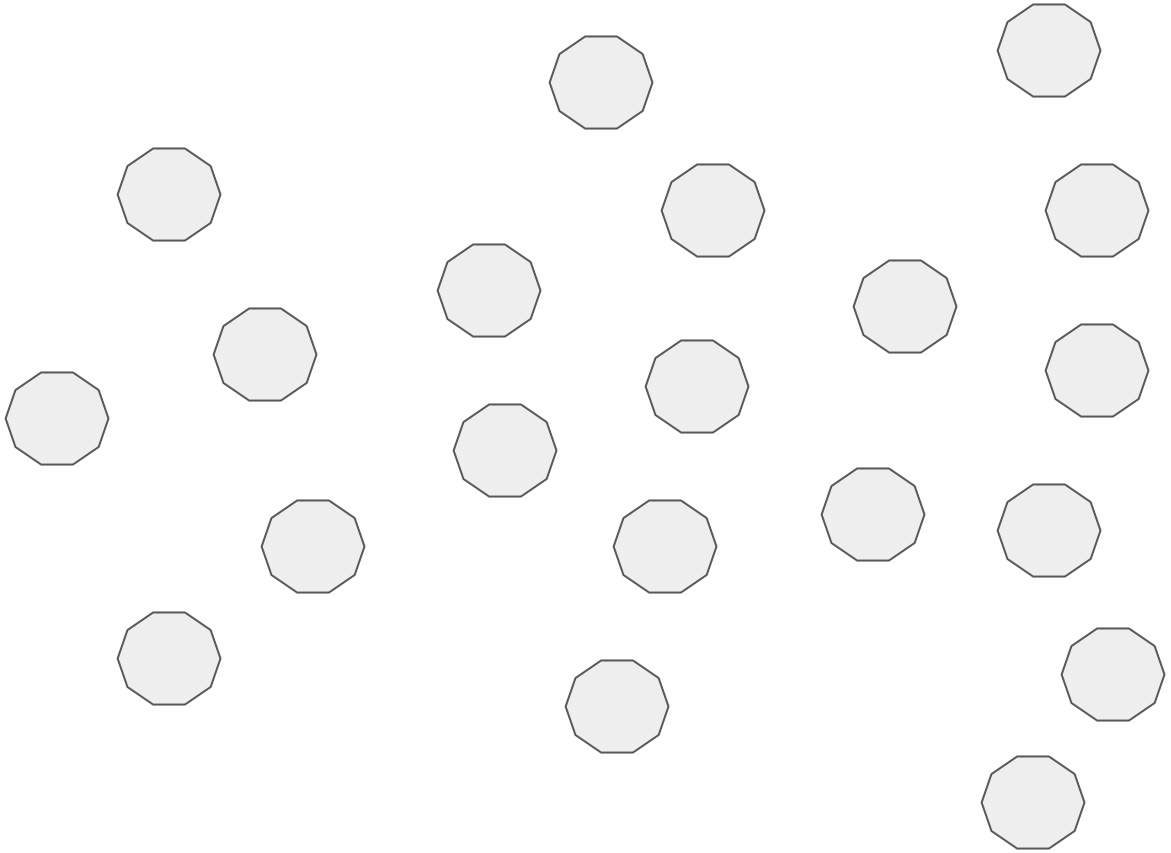
Microservices



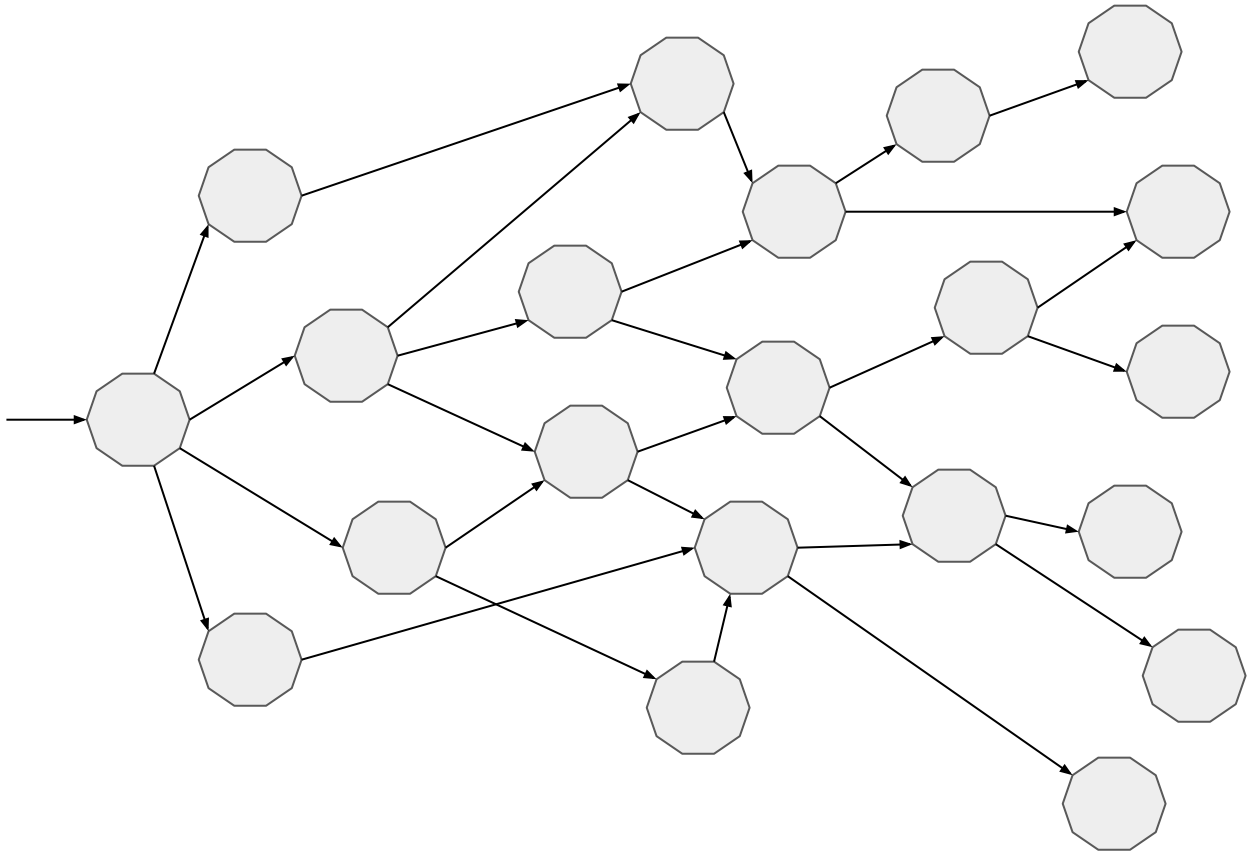
Microservices



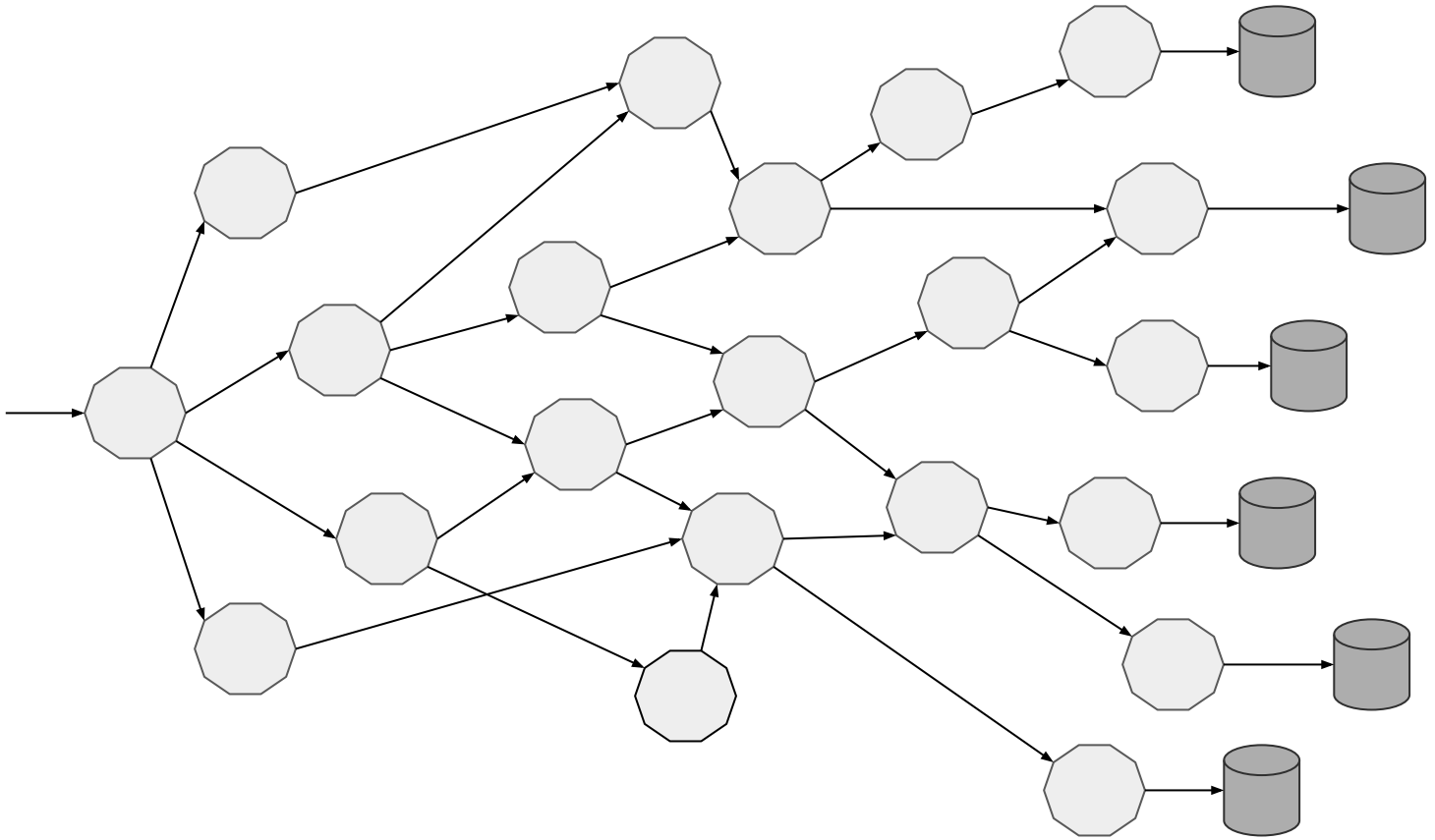
Microservices



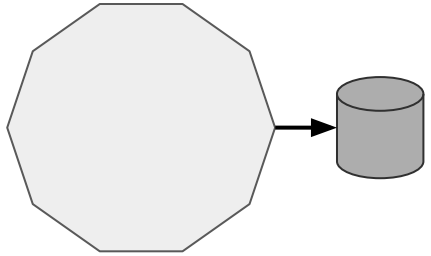
Netzwerk aus Services



Microservices mit eigener Datenhaltung

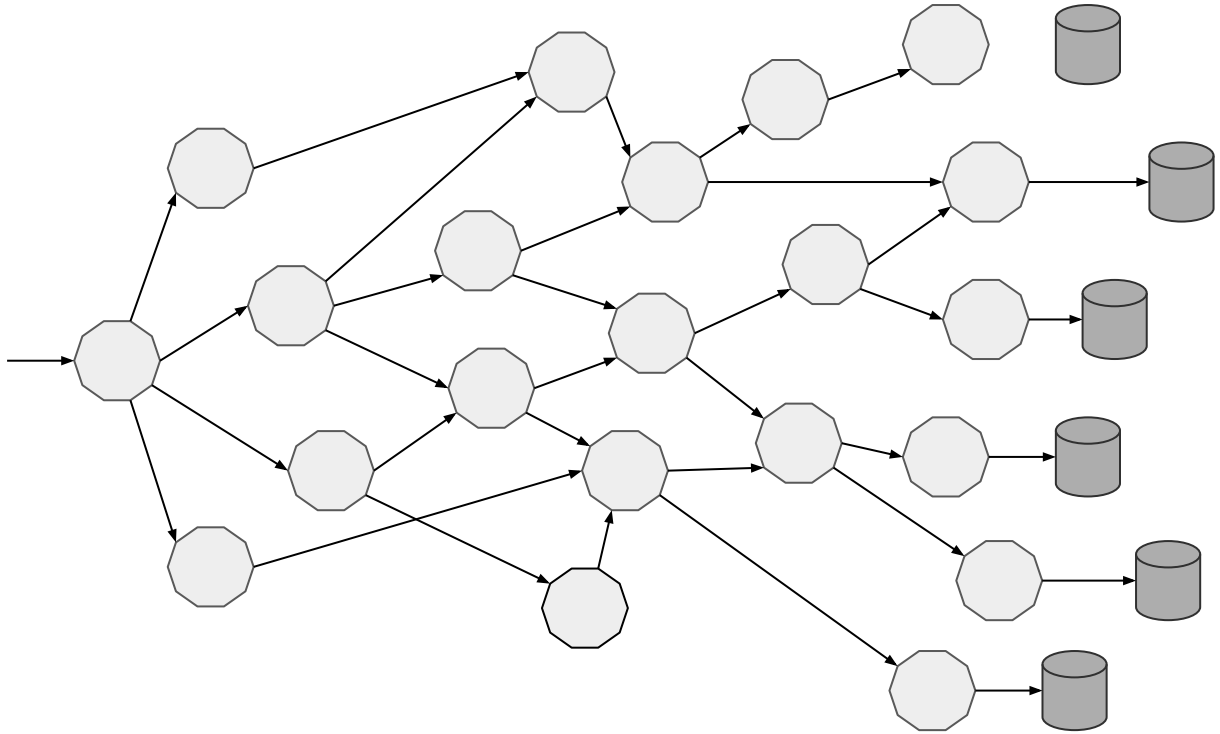


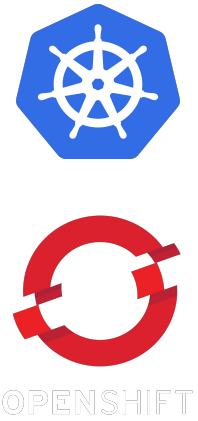
Alte Schule



Wir 😍 Monolithen

Neue Schule



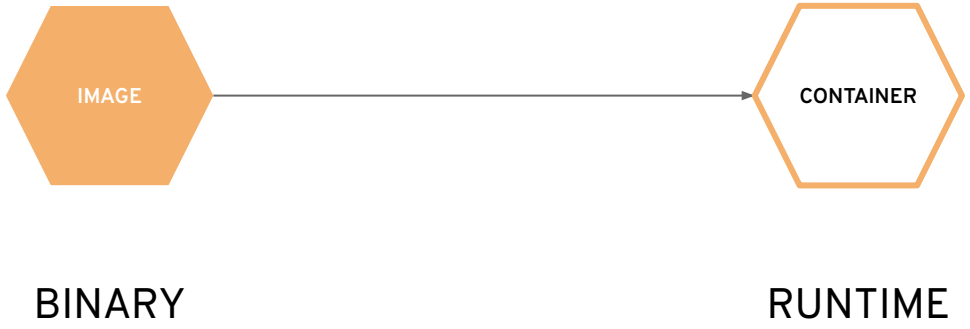


Container-Technologie

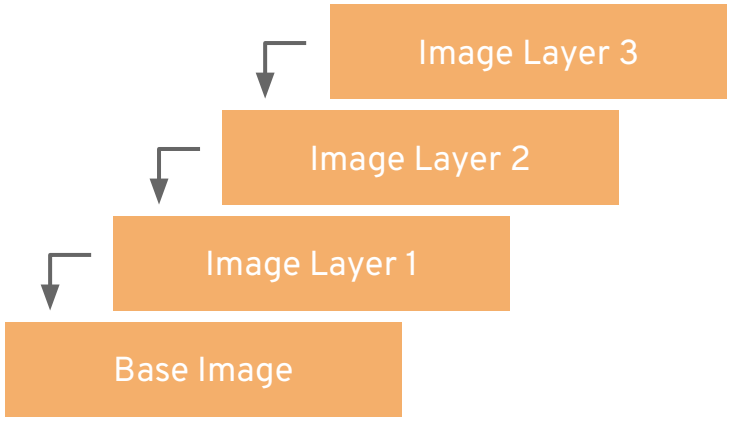
- Einfach hochskalierbar
- Ausgereifte Technologie
- Für Cloud-Native und moderne App-Workloads

Ein **Container** ist die kleinste Recheneinheit

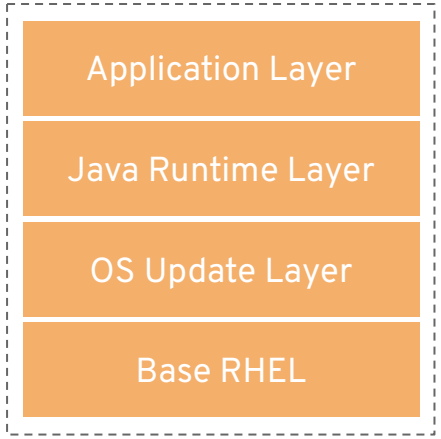
Container werden aus **Container-Images** erstellt



Container-Images sind in Layern aufgebaut



Container Image-Layer



Container Image
(Beispiel)

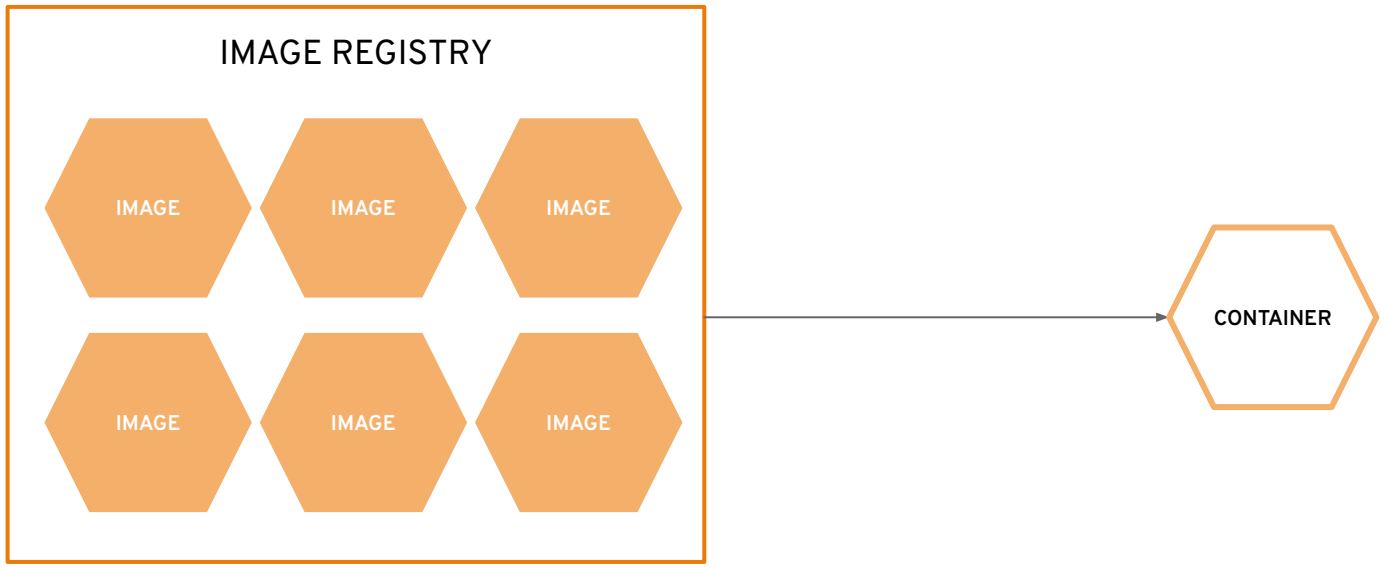
Anatomie eines Containerfiles (Dockerfiles)

```
FROM registry.access.redhat.com/ubi8/ubi
ENV foo=text
RUN dnf install -y java-11-openjdk
ADD my-app.jar /home/my-app.jar
EXPOSE 8080
CMD java -jar /home/my-app.jar
```

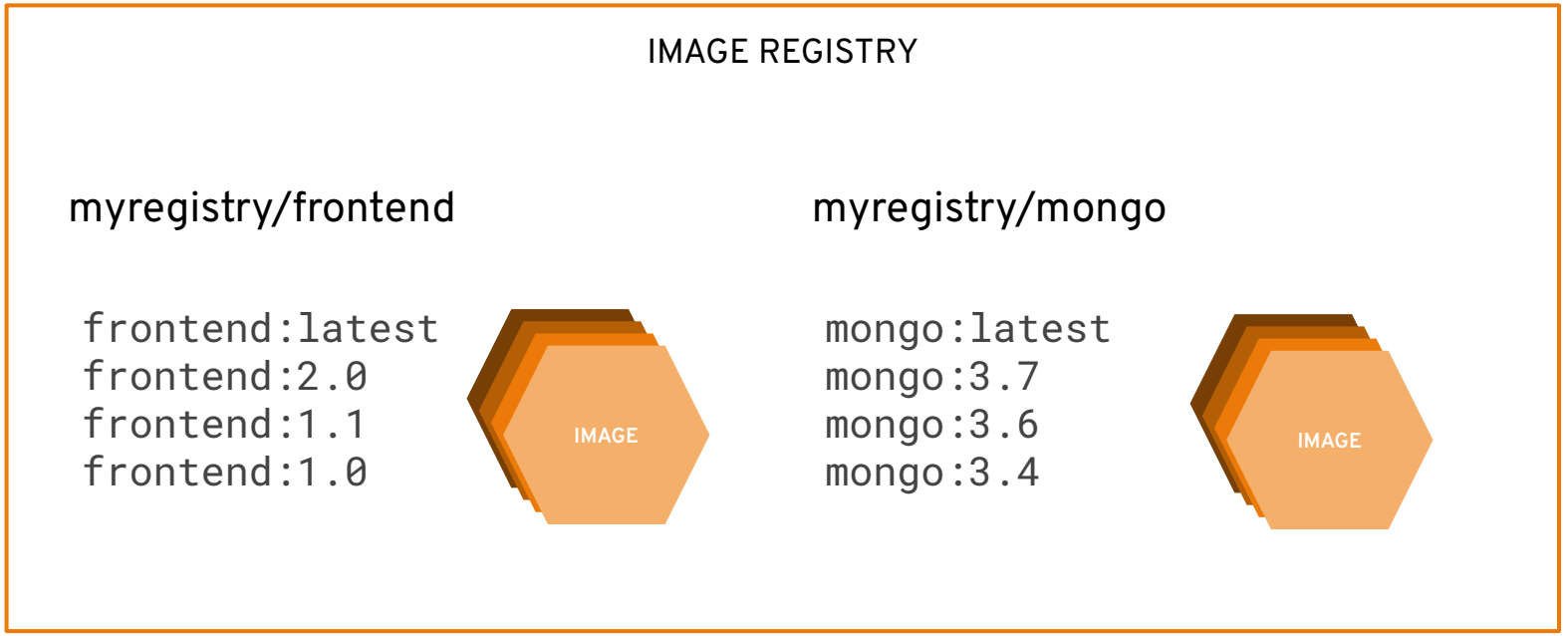
1. Erbt von einem Basis-Image
2. Parameter als Umgebungsvariablen
3. Abhängigkeiten installieren (Tooling vom Basis-Image)
4. Eure App als neuer Layer
5. Port der App freigeben
6. App starten

Beispiel: Java-Anwendung

Container-Images werden im Image-Registry vorgehalten



Ein Image-Repository enthält alle Versionen eines Images in der Image-Registry

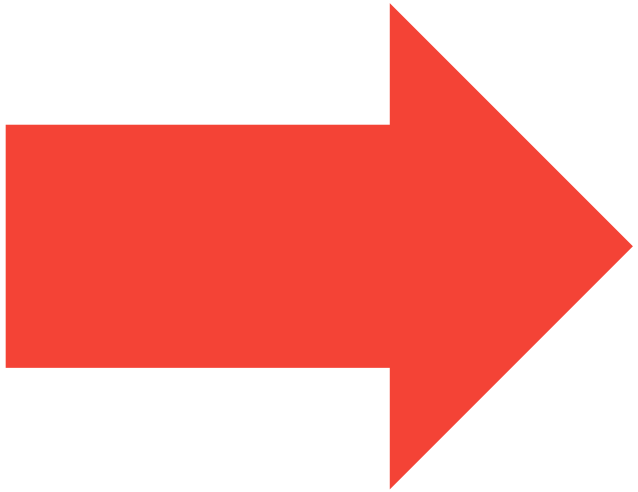


Was ist Kubernetes?

Was ist Kubernetes?

Ein Open-Source-Orchestrierungssystem zum Verwalten von containerisierten Workloads über einen Cluster von Nodes hinweg.

Kubernetes-Objekte verstehen



Kubernetes-Objekte sind persistente Einheiten, die den gewünschten Zustand deines Clusters zeigen und die mit der K8s-API verwalten werden können.

Kubernetes-Objekte verstehen

Pod

ReplicaSet

Deployment

Secret

Namespace

ConfigMap

Service

PersistentVolume

Kubernetes hat ein API

Grundlegende API-Bestandteile:

```
kind
apiVersion
metadata
spec
status
```

Node

- Node: Ein Rechner, auf dem containerisierte Workloads laufen
- Die Node-Aktivität wird von einer oder mehreren Master-Instanzen verwaltet

Pod

- A group of one or more co-located containers
- Minimum unit of scale

```
kind: Pod
apiVersion: v1
metadata:
  creationTimestamp:
  name: hello-k8s
  labels:
    run: hello-k8s
spec:
  containers:
  - name: hello-k8s
    image: jkleinert/nodejsint-workshop
    ports:
    - containerPort: 8080
    resources: {}
```

Pod

```
kubectl create -f
https://raw.githubusercontent.com/jankleinert/hello-workshop/master/pod.jso
```

```
kubectl get pods
```

```
kubectl describe pod/hello-k8s
```

Service

- Funktioniert als alleiniger Endpunkt für eine Gruppe replizierter Pods
- Ähnlich wie ein Load Balancer

```
kind: Service
apiVersion: v1
metadata:
  name: hello-k8s
  creationTimestamp:
  labels:
    run: hello-k8s
spec:
  ports:
  - protocol: TCP
    port: 8080
    targetPort: 8080
  selector:
    run: hello-k8s
  type: NodePort
status:
  loadBalancer: {}
```

Service

```
kubectl expose pod/hello-k8s --port 8080 --type=NodePort
```

```
kubectl get svc/hello-k8s -o yaml
```

```
curl hello-k8s.<userX>:8080
```

Clean up

```
kubectl get pods -l run=hello-k8s
```

```
kubectl delete pods -l run=hello-k8s
```

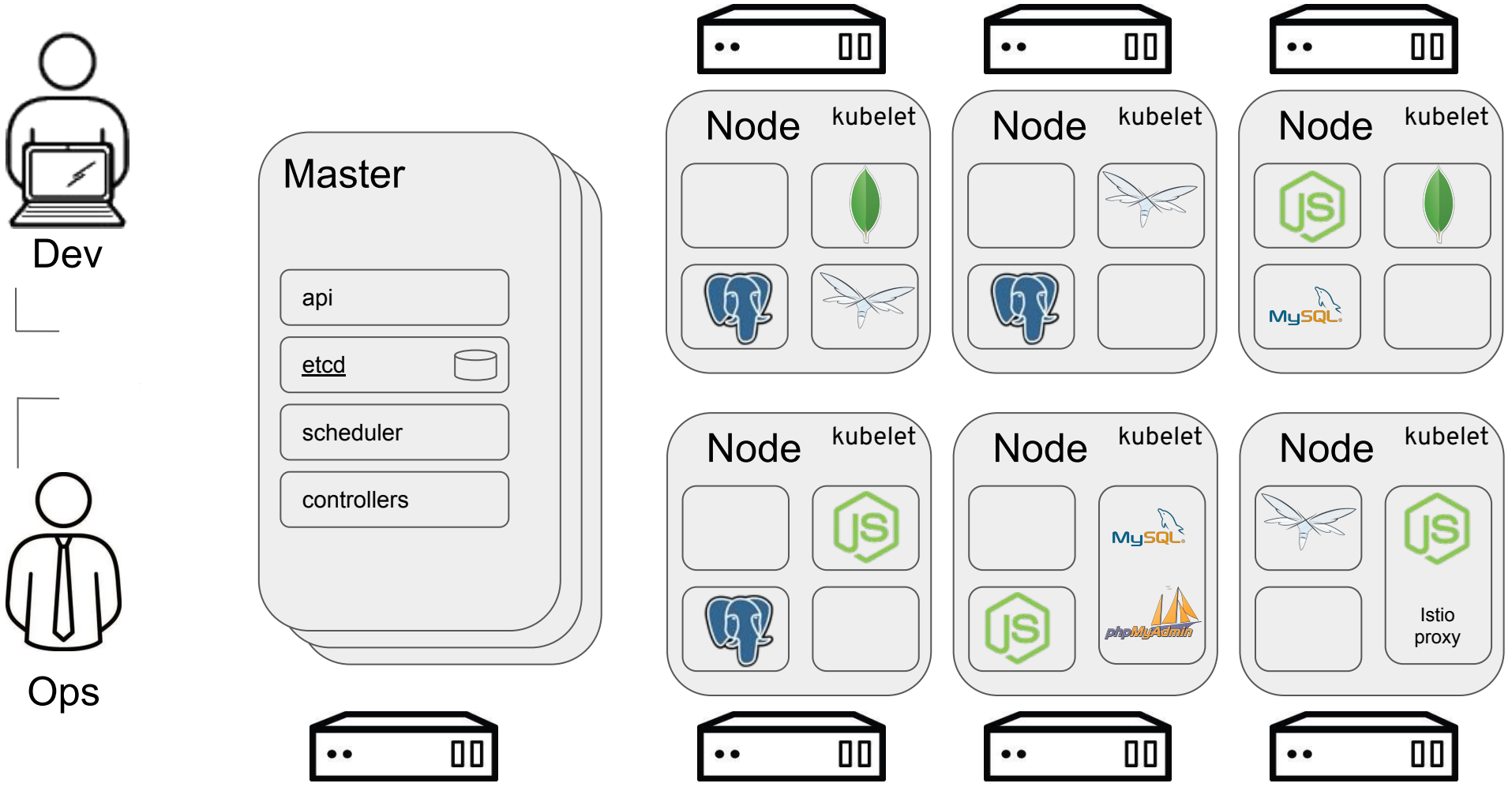
```
kubectl delete service hello-k8s
```

Deployment

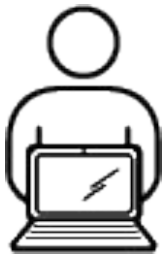
- Hilft dabei, die Container-Runtime aus Pod-Sicht festzulegen


```
kind: Deployment
apiVersion: apps/v1
metadata:
  name: hello-k8s
  creationTimestamp:
  labels:
    run: hello-k8s
spec:
  replicas: 1
  selector:
    matchLabels:
      run: hello-k8s
  template:
    metadata:
      creationTimestamp:
      labels:
        run: hello-k8s
    spec:
      containers:
      - name: hello-k8s
        image: jkleinert/nodejsint-workshop
        resources: {}
  strategy: {}
status: {}
```

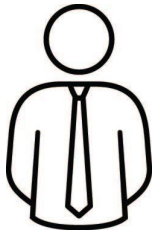
Kubernetes Cluster-Nodes



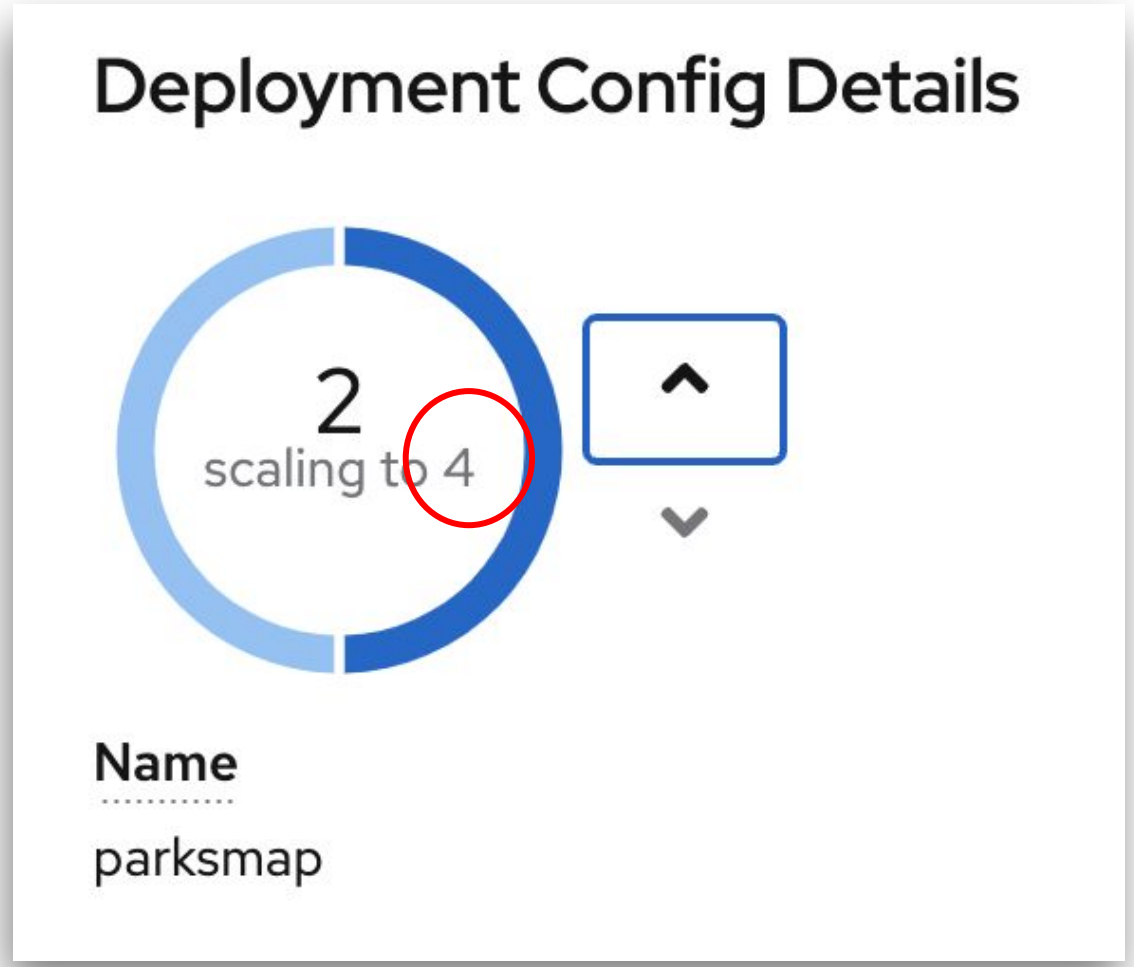
Kubernetes Cluster - Deklarativ



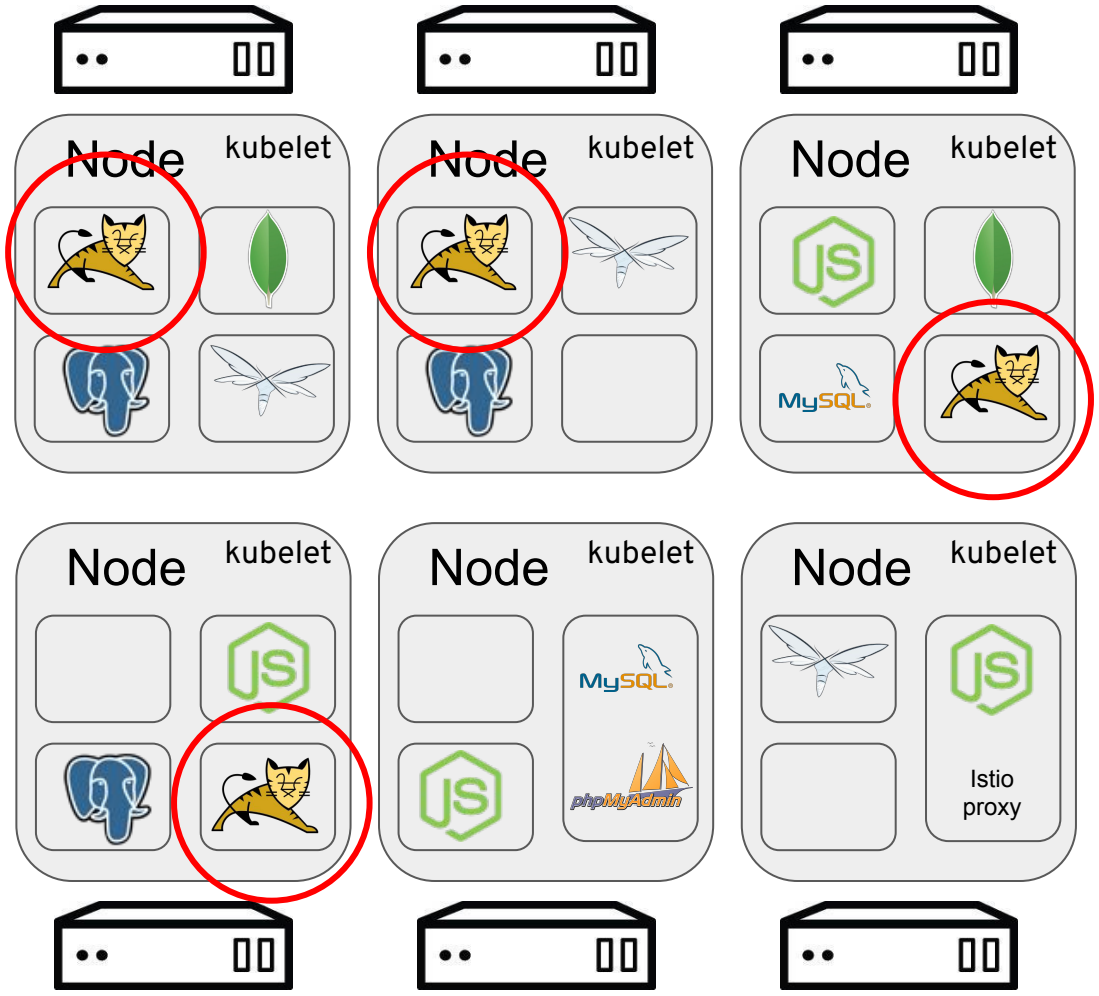
Dev



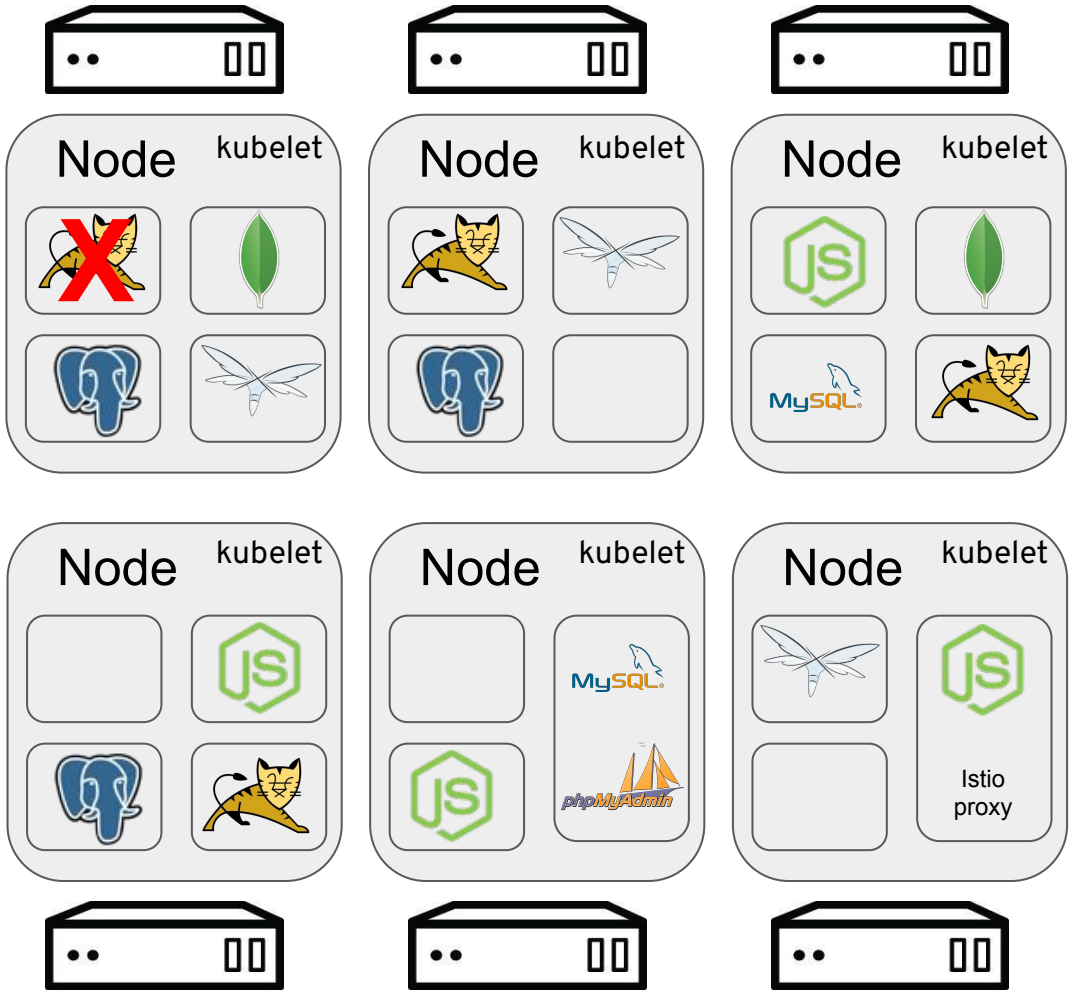
Ops



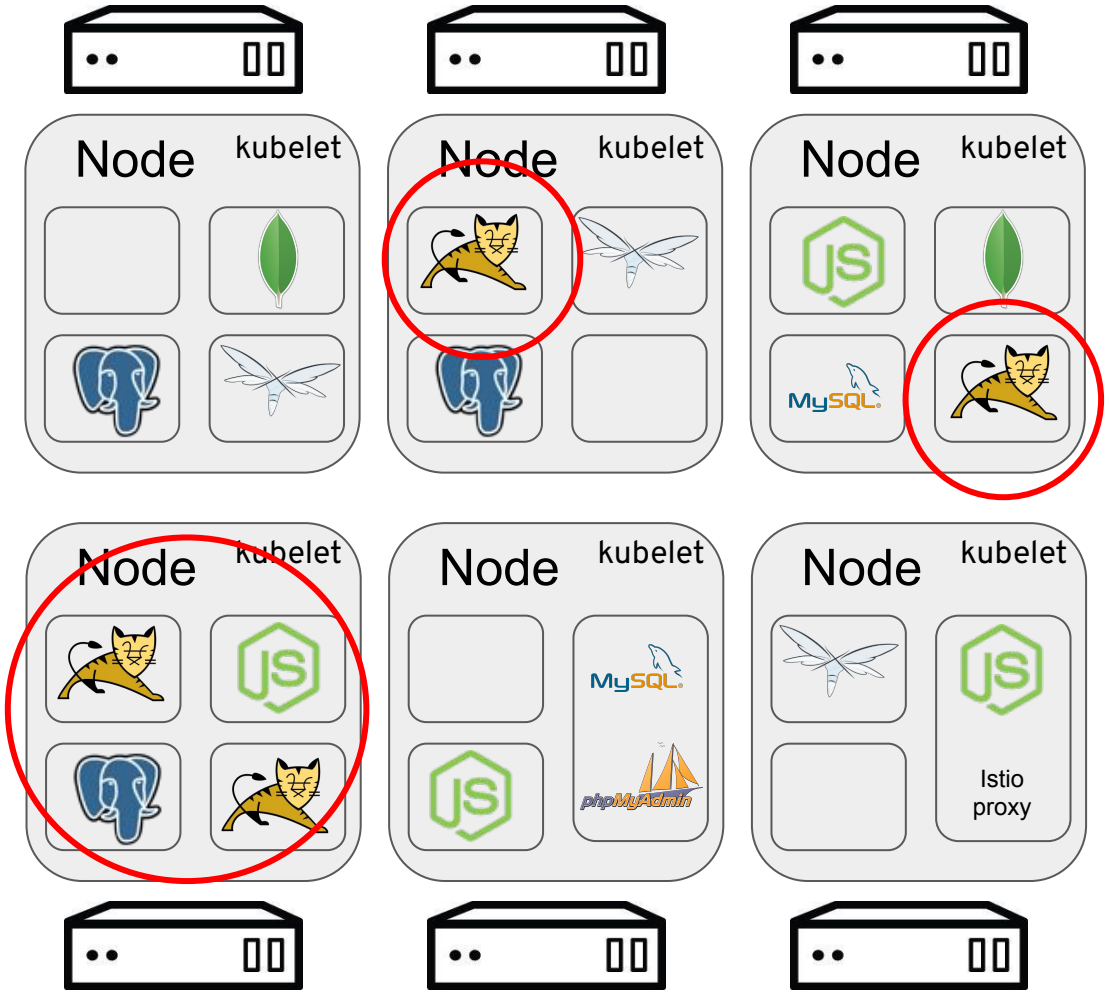
Kubernetes Cluster - 4 Tomcats



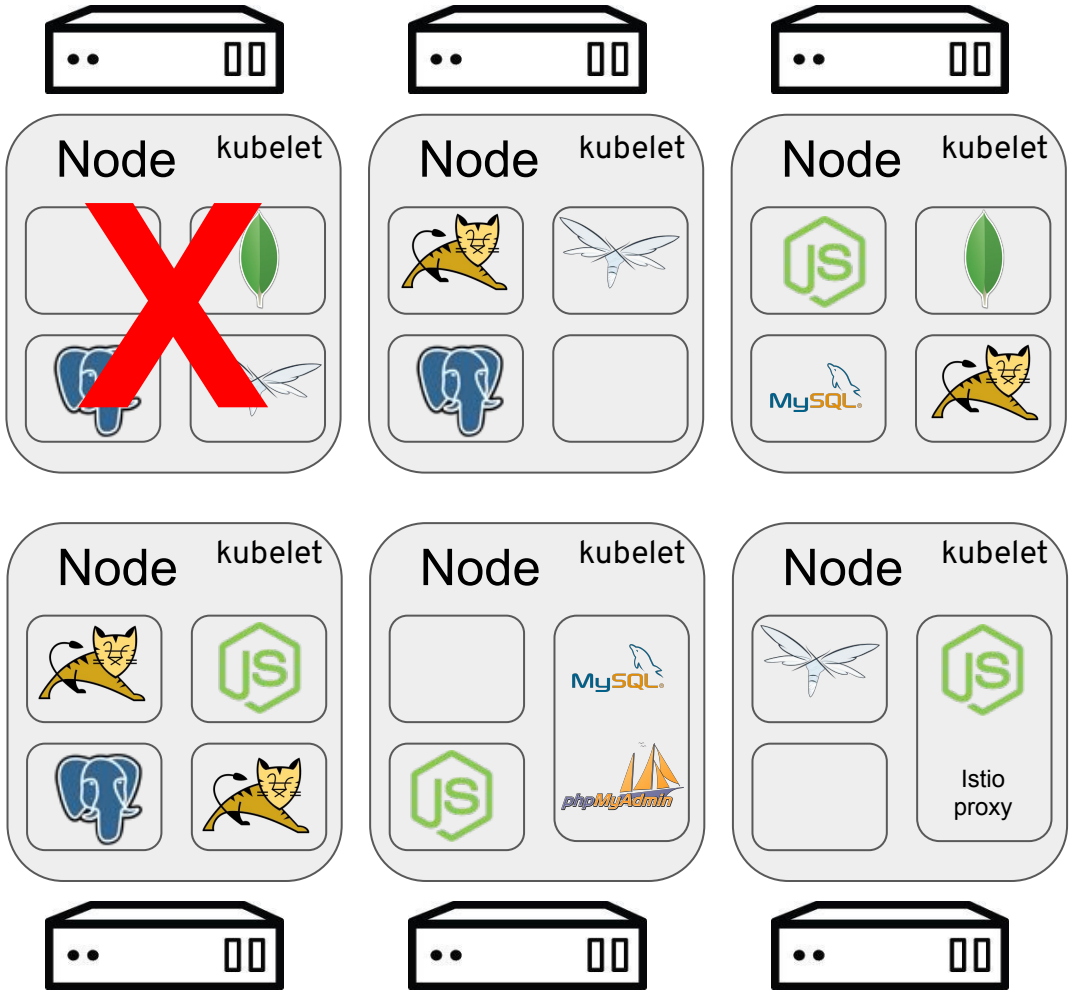
Kubernetes Cluster - Pod kaputt 😲



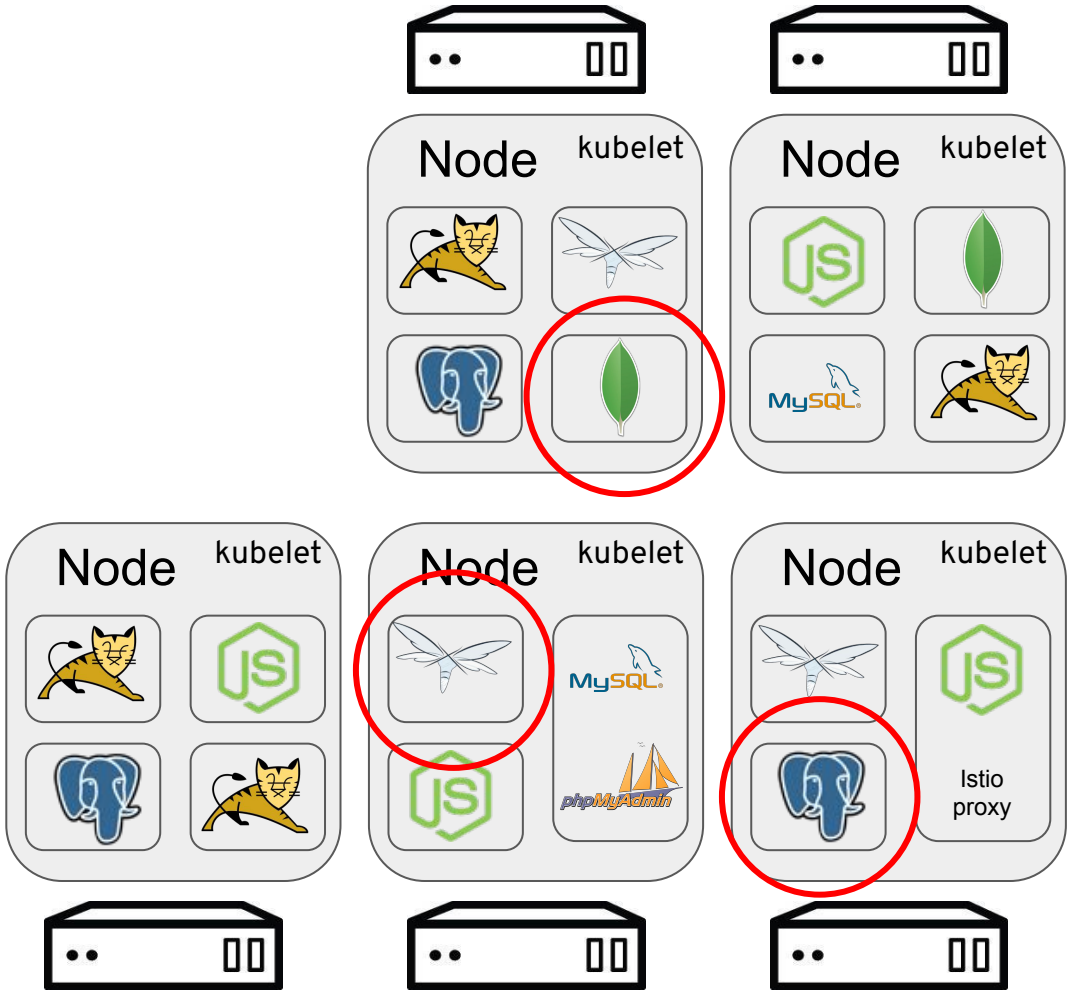
Kubernetes Cluster - korrigiert 🙂



Kubernetes Cluster - Node kaputt 😩

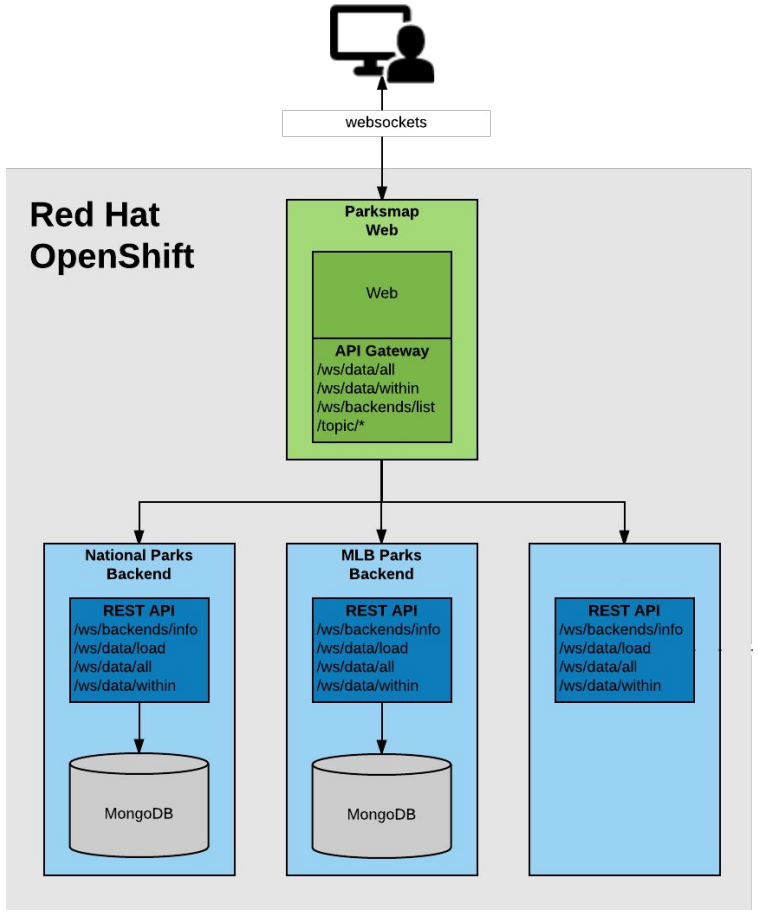


Kubernetes Cluster - Pods replaziert 😎



Lab-Module

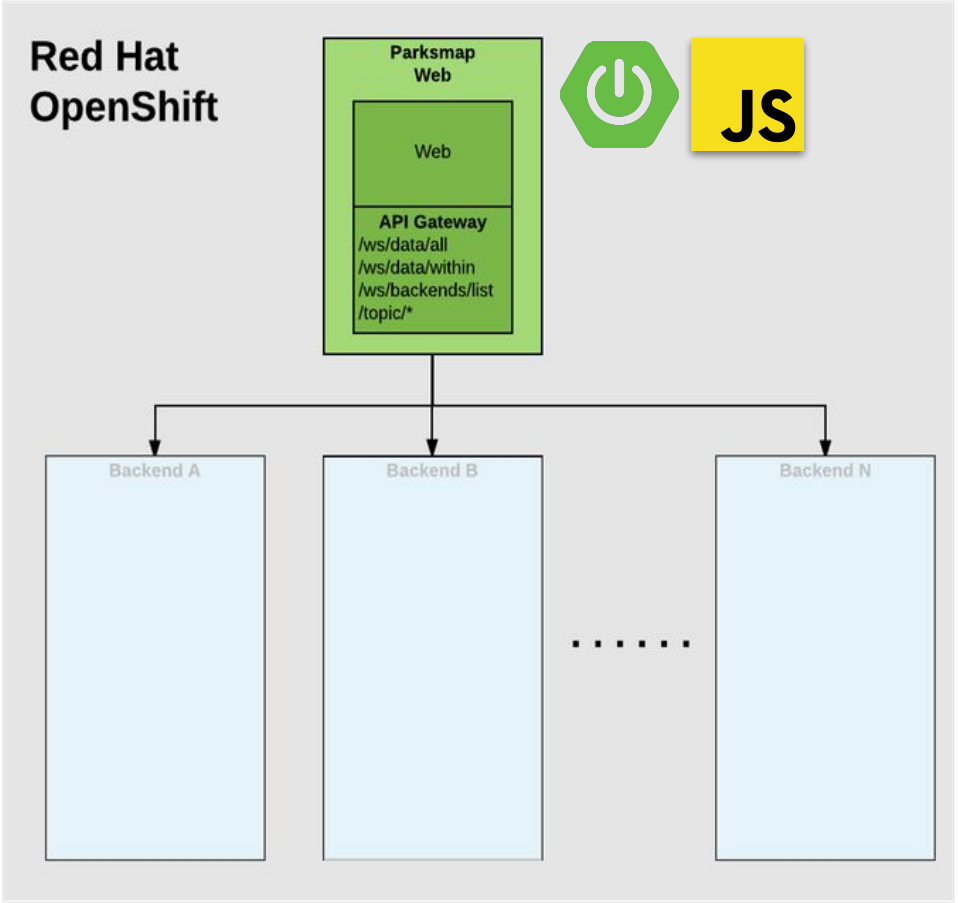
Parksmap: Architektur



Red Hat

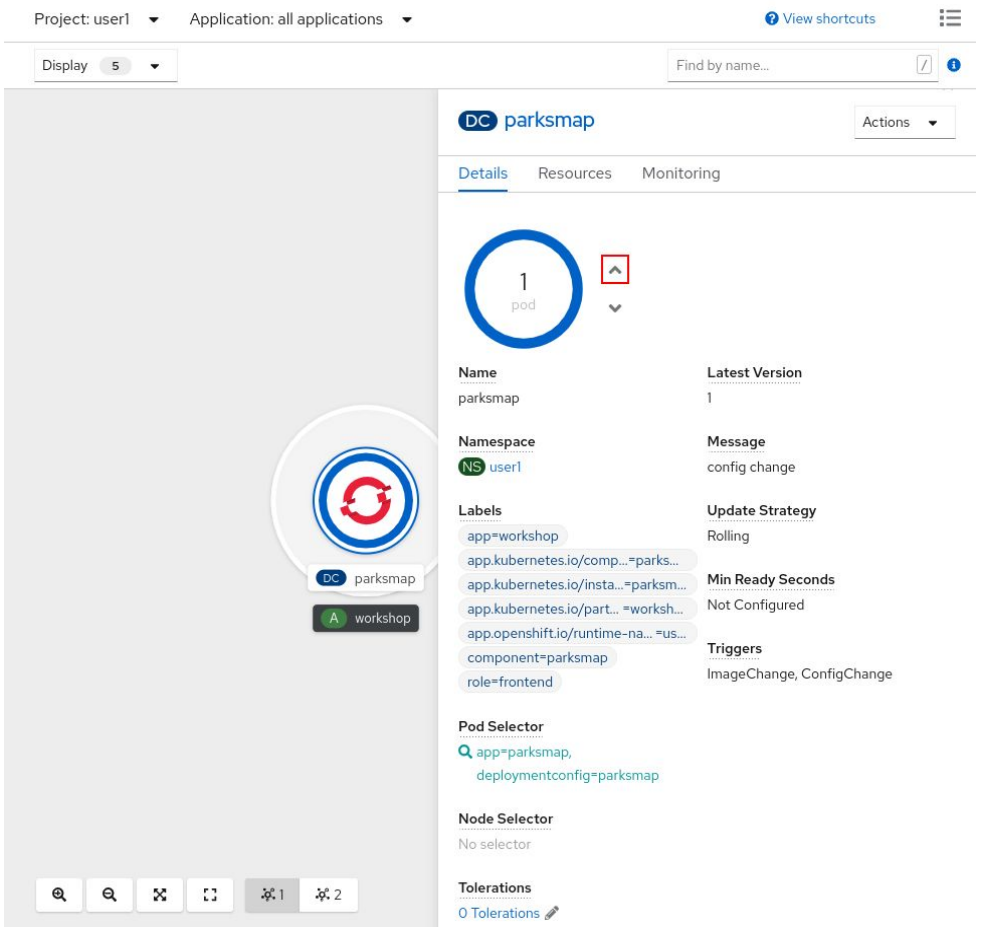
Parksmap Web

- Spring Boot-Frontend, das das Mapbox-JavaScript-API nutzt, um eine Weltkarte mit Datenpunkten anzuzeigen
- Als Container-Image bereitgestellt, das öffentlich auf Quay.io verfügbar ist
- Arbeitet mit verschiedenen Backends zusammen, die dieselben REST-Endpunkte bereitstellen (kann ein API-Gateway integrieren)
- Erste App-Bereitstellung über die OpenShift Developer Console



Red Hat

Parksmap: OpenShift erkunden

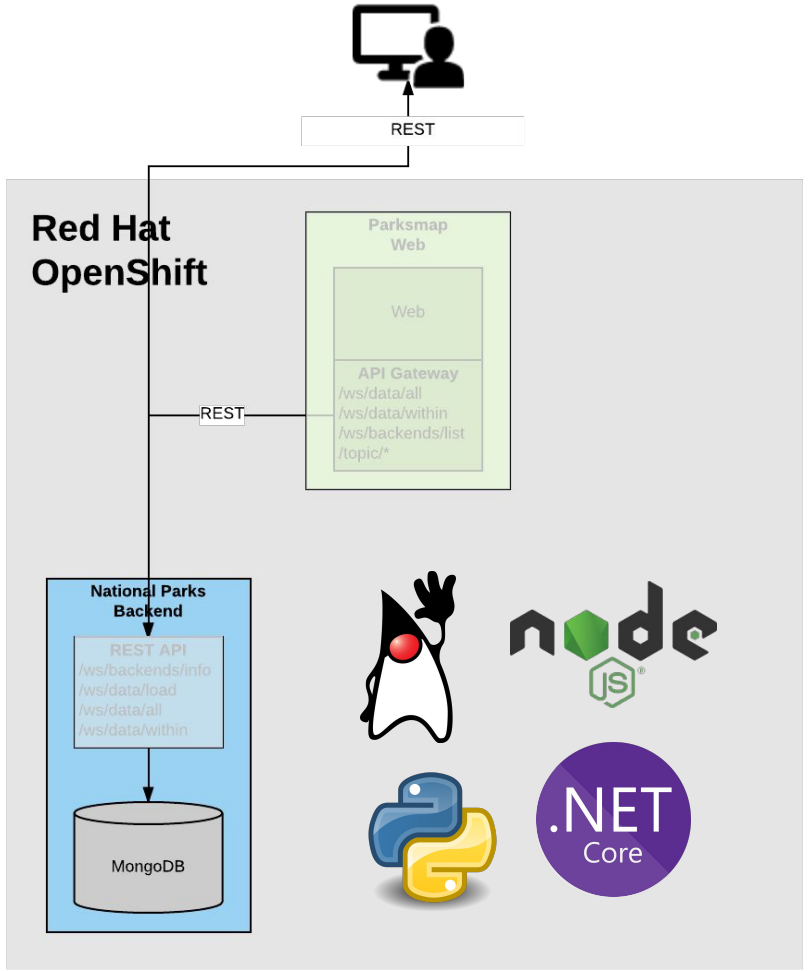


- Apps skalieren
- Logging
- Labels
- Berechtigungen
- Zugriff auf Container und Debugging

Red Hat

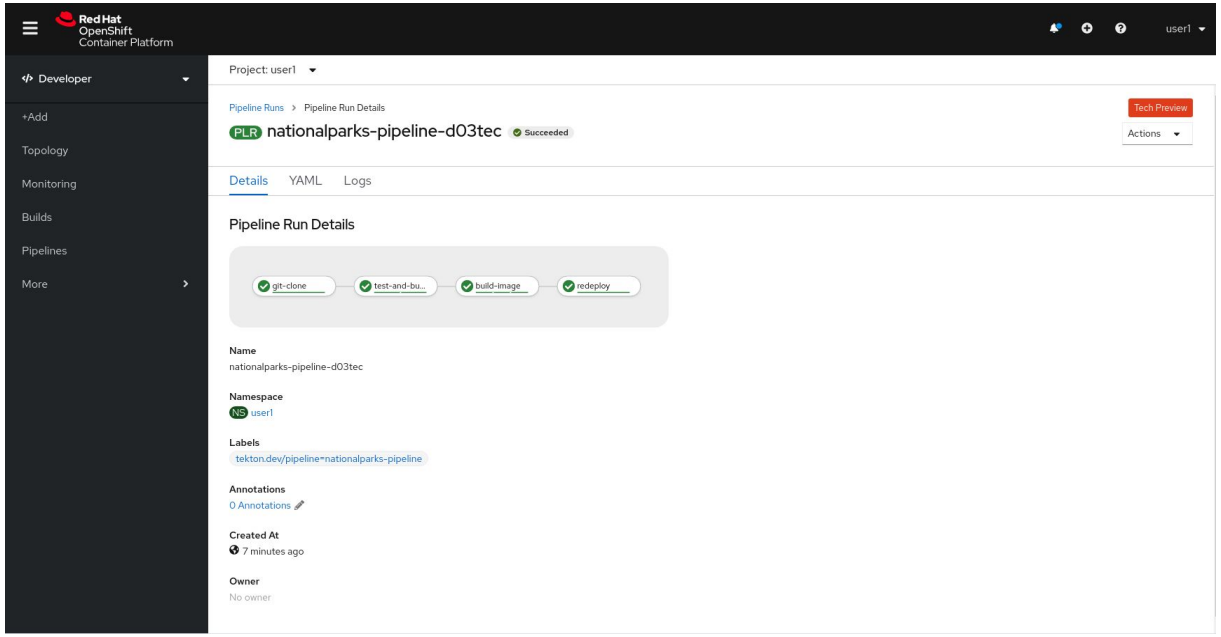
NationalParks Backend

- Backend zur Anzeige weltweiter Nationalparks
- Verwendung einer MongoDB zum Speichern und Abrufen von Daten als Geokoordinaten
- Bereitstellung von REST-APIs für das Parksmap-Frontend
- Automatische Erstellung von Container-Images aus dem Quellcode mit S2I (Source-to-Image)
- Verfügbar für Java, NodeJS, Python und .NET Core

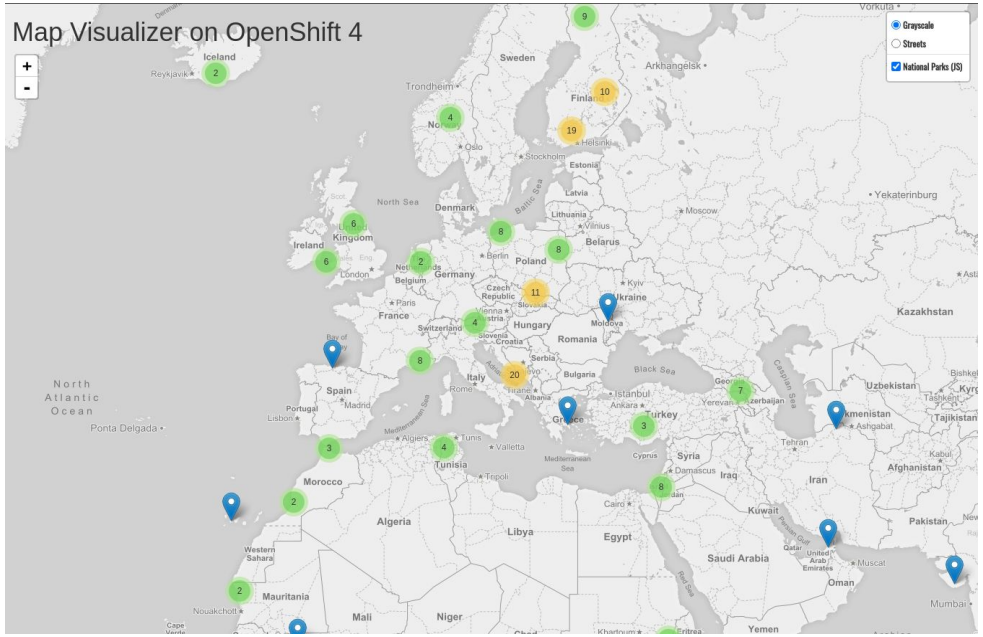


Red Hat

NationalParks: OpenShift erkunden



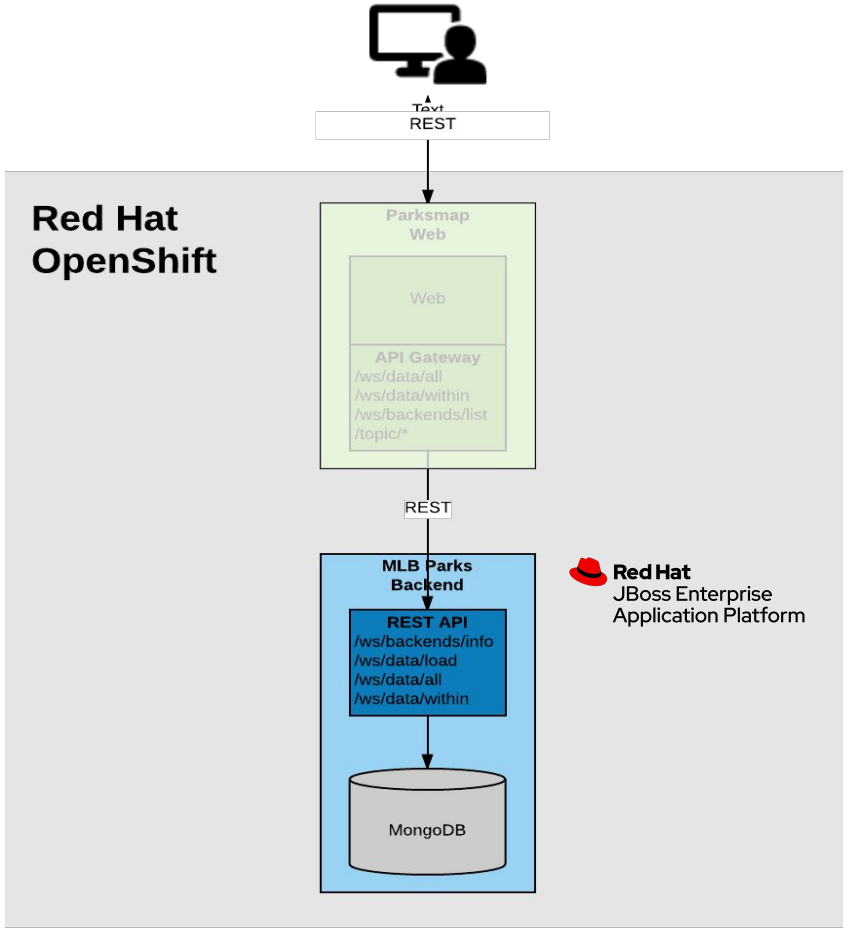
- Health Checks
- Automatisierung mit Pipelines
- Web Hooks zum automatischen Erstellen und Bereitstellen von Code-Änderungen



Red Hat

MLB Parks-Backend (Java)

- OpenShift kann prima "Legacy"-Apps betreiben
- Java EE-Backend zur Anzeige der Major League Baseball-Stadien in Nordamerika
- Artefakte (.war) lokal mit deiner IDE oder Workstation erstellen
- Container-Images mit S2I-Binaries erstellen auf OpenShift betreiben



Red Hat


THIS IS A PROTECTED SPACE
ARE SAFE
YOU MISTAKES HERE
MISTAKES WELCOME
EMBRACE IMPERFECTION
GROW AND LEARN
ALL ARE ALLOWED

Lab-Anleitung

- **Alles läuft im Browser** – Ihr braucht keine lokalen Befehle oder Installationen auf dem Laptop.
- Funktioniert am besten mit Chrome oder Firefox.
- Bitte VPN ausschalten (intensive Websockets-Nutzung) und AdBlocker für die Lab-Domain pausieren (keine Werbung, versprochen).
- Jeder arbeitet mit einem eigenen einmaligen Login, z. B.: user45 / getstarted!

Lab URL: red.ht/ocp-zum-anfassen

Passwort: getstarted!

Wer nicht weiterkommt: Einfach melden 😀

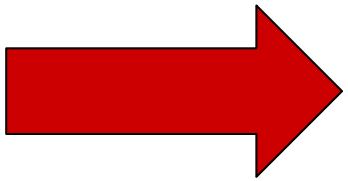
Ressourcen

- Die Lab-Anleitung online:
  https://rhpds.github.io/ocp4-getting-started-showroom/modules/main/index.html
- Lab-Anleitung auf GitHub:
  https://github.com/rhpds/ocp4-getting-started-showroom
- Sourcen für das Lab (Ansible für AWS, nennt sich "agnosticd", hoher Einarbeitungsaufwand):
  https://github.com/redhat-cop/agnosticd/tree/development/ansible/roles_ocp_workloads/ocp4_workload_getting_started

Red Hat
Demo Platform

OCP4 Getting Started Workshop (2025)

Instructions for OCP4 Getting Started Workshop (2025)



Lab User Interface	https://showroom-showroom-jpjdm-user1.apps.cluster-jpjdm.dynamic.redhatworkshops.io/
Messages	Lab instructions: https://showroom-showroom-jpjdm-user1.apps.cluster-jpjdm.dynamic.redhatworkshops.io/

Data

```
console_url: console-openshift-console.apps.cluster-jpjdm.dynamic.redhatworkshops.io
gitea_console_url: https://gitea.apps.cluster-jpjdm.dynamic.redhatworkshops.io
gitea_password: MTIzODU4
gitea_user: user1
login_command: >-
  oc login --insecure-skip-tls-verify=false -u user1 -p MTIzODU4
  https://api.cluster-jpjdm.dynamic.redhatworkshops.io:6443
openshift_cluster_ingress_domain: apps.cluster-jpjdm.dynamic.redhatworkshops.io
openshift_console_url: >-
  https://console-openshift-console.apps.cluster-jpjdm.dynamic.redhatworkshops.io
password: MTIzODU4
showroom_primary_view_url: >-
  https://showroom-showroom-jpjdm-user1.apps.cluster-jpjdm.dynamic.redhatworkshops.io/
user: user1
```

Red Hat Summit

Connect

Danke für's Mitmachen 

linkedin.com/company/red-hat

facebook.com/redhatinc

youtube.com/user/RedHatVideos

twitter.com/RedHat