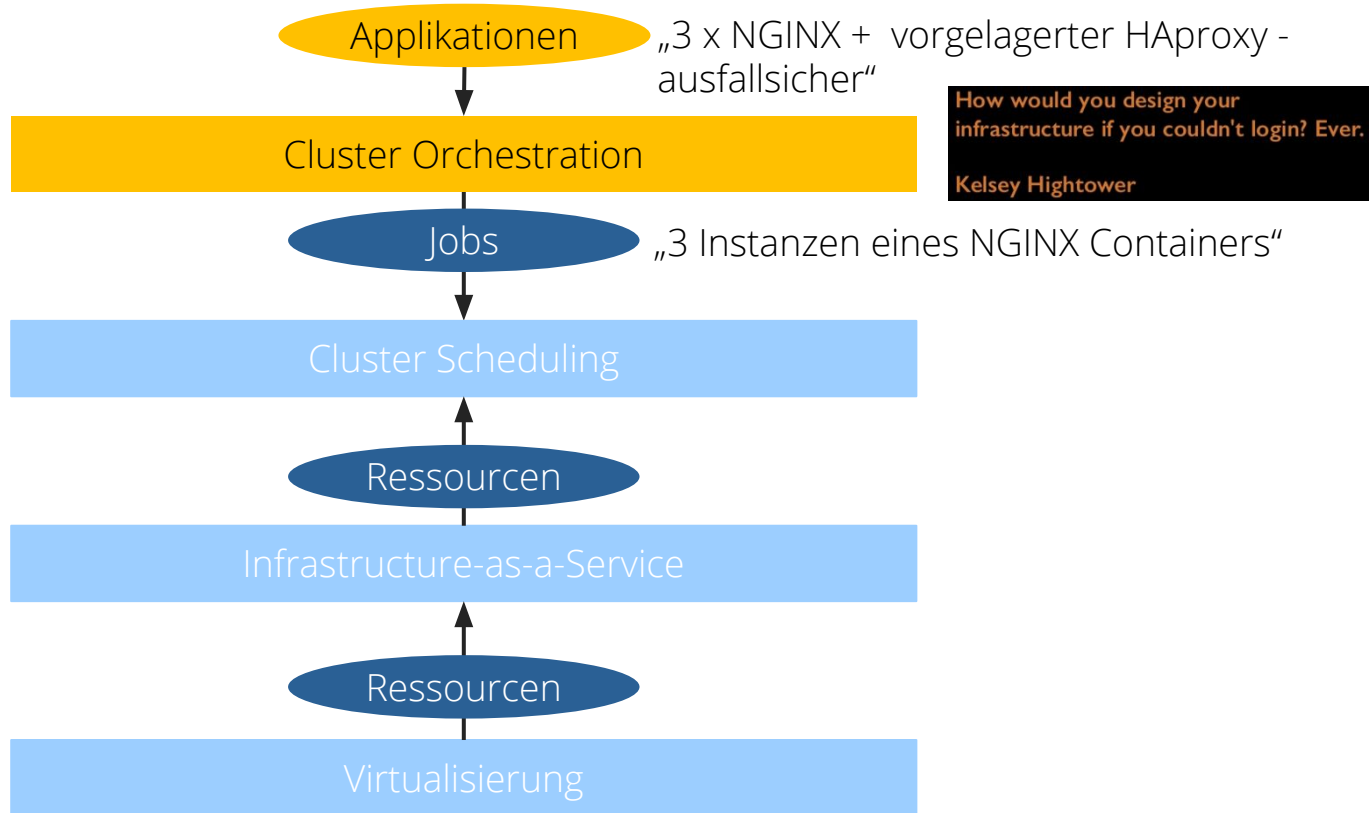


Cloud Computing Orchestrierung

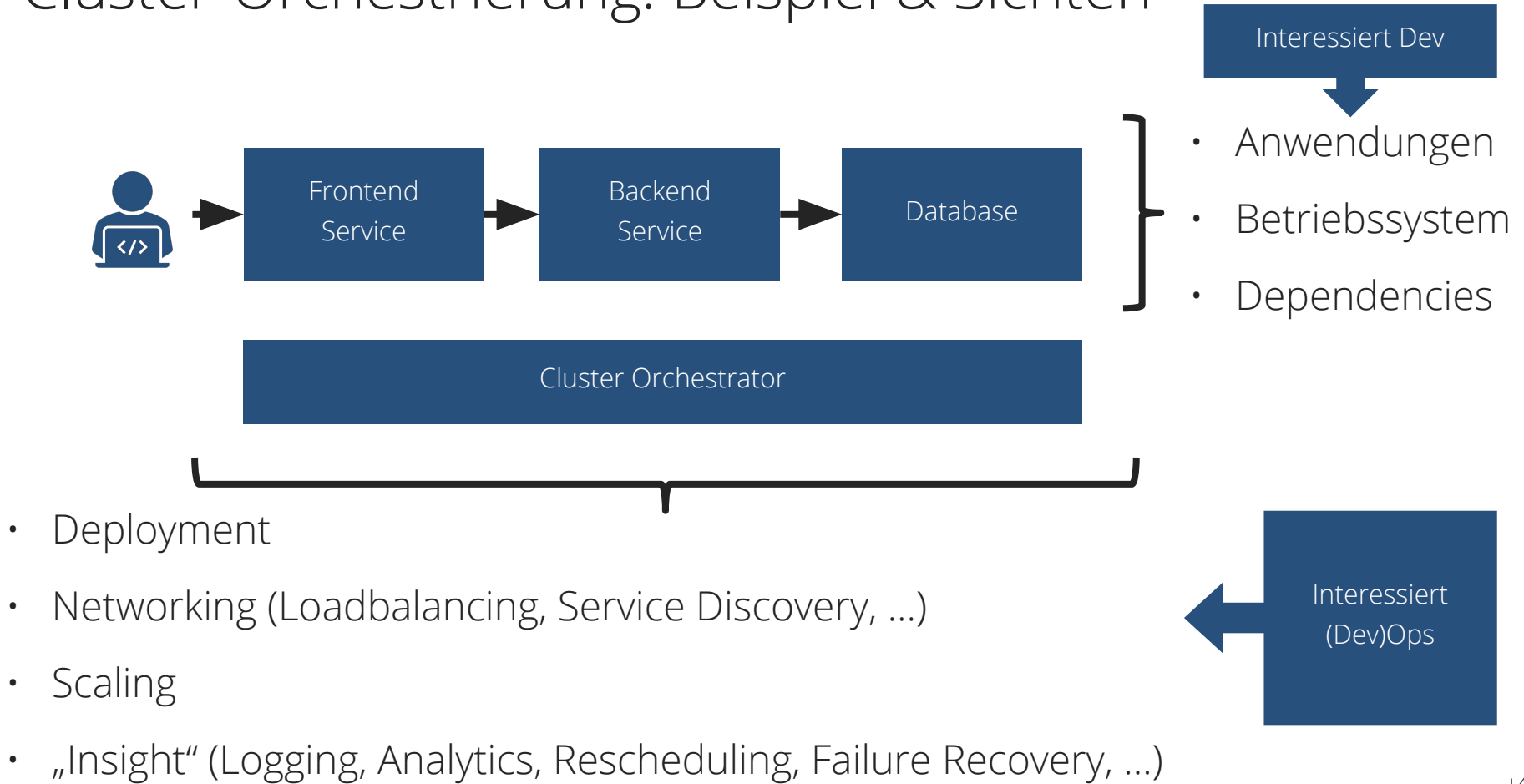


Orchestrierung - Definition, Aufgaben, Konzepte

Big Picture: Wir sind nun auf Applikationsebene



Cluster-Orchestrierung: Beispiel & Sichten



Cluster-Orchestrierung

- Ziel: Eine Anwendung, die in mehrere Betriebskomponenten (Container) aufgeteilt ist, auf mehreren Knoten laufen lassen.
„Running Containers on Multiple Hosts“.
DockerCon SF 2015: Orchestration for Sysadmins
- Führt Abstraktionen zur Ausführung von Anwendungen mit ihren benötigten Schnittstellen und Betriebskomponenten ein.
- Orchestrierung ist keine statische, einmalige Aktivität wie die Provisionierung, sondern eine dynamische, kontinuierliche Aktivität.
- Orchestrierung hat den Anspruch, alle Standard-Betriebsprozeduren einer Anwendung zu automatisieren.

Blaupause der Anwendung, die den gewünschten Betriebszustand der Anwendung beschreibt: Betriebskomponenten (Container), deren Betriebsanforderungen sowie die angebotenen und benötigten Schnittstellen.



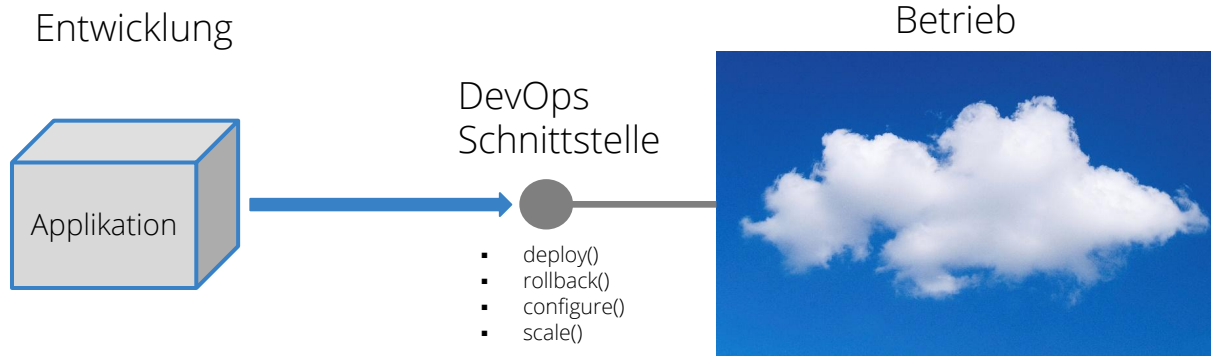
Cluster-Orchestrierer



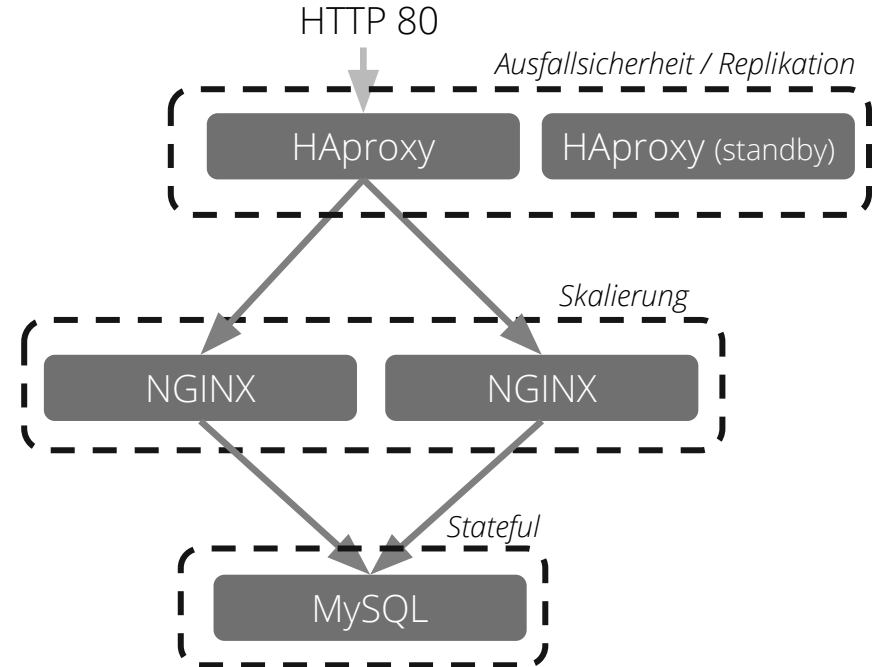
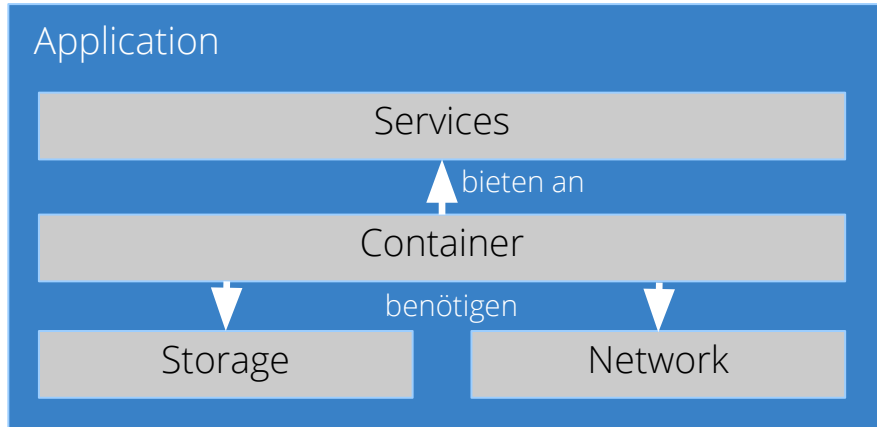
Steuerungsaktivitäten im Cluster:

- Start von Containern auf Knoten (□ Scheduler)
- Verknüpfung von Containern
- ...

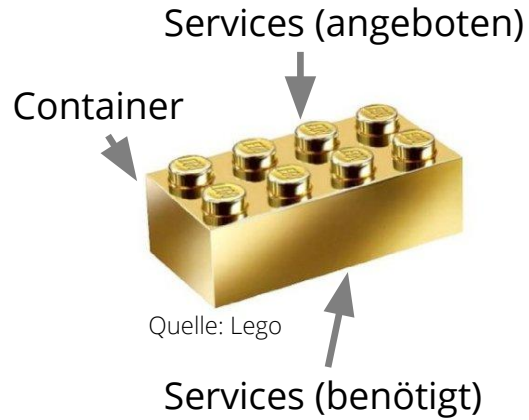
Ein Cluster-Orchestrierer bietet eine Schnittstelle zwischen Betrieb und Entwicklung für ein Cluster an



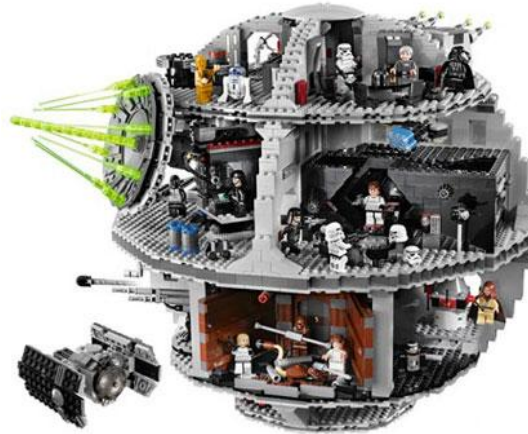
Dafür benötigt er für jede Anwendung eine Blaupause



Analogie 1: Lego Star Wars

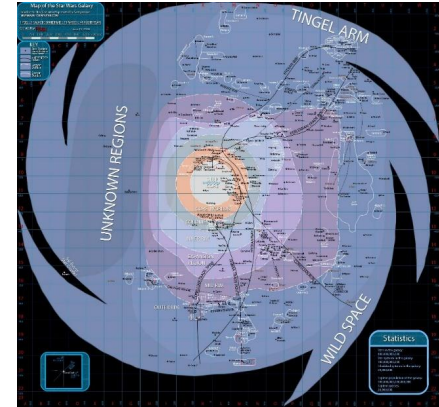


Cluster-Orchestrierer



Quelle: Lego

Cluster-Scheduler

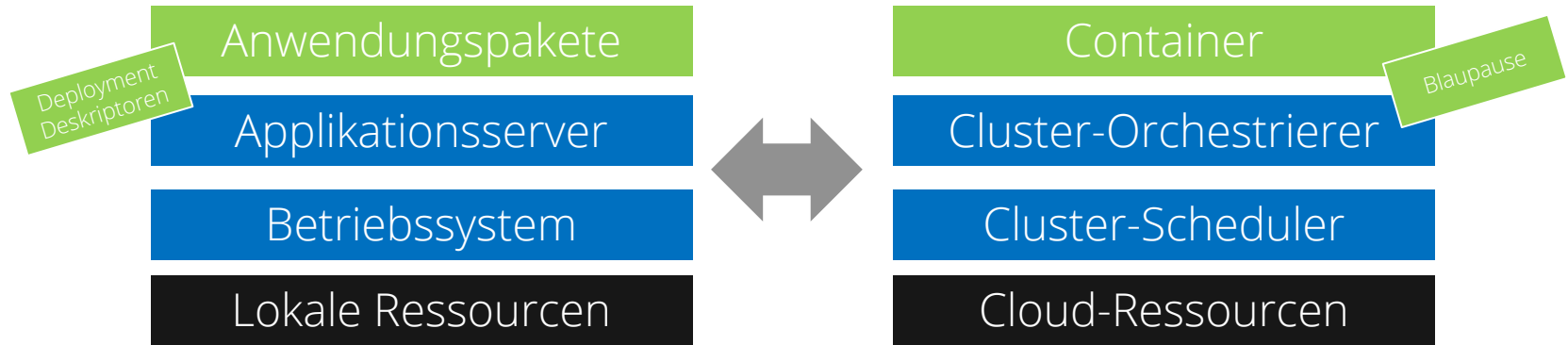


Quelle: wikipedia.de

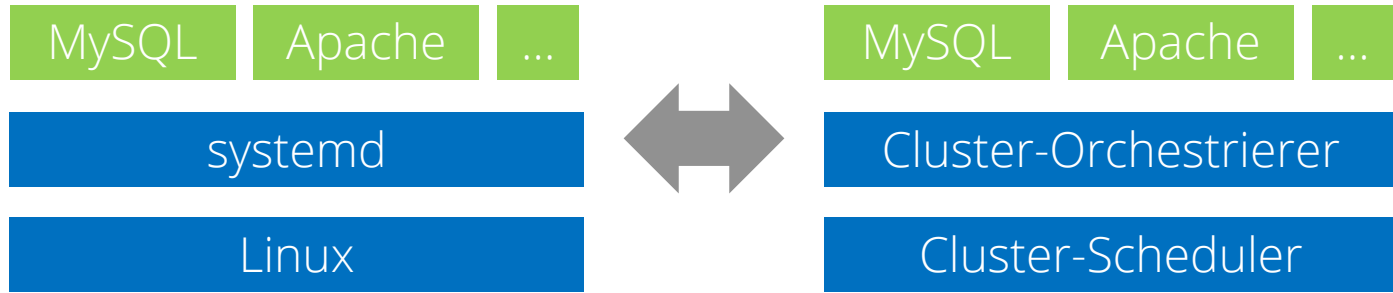


Blaupause

Analogie 2: Applikationsserver



Analogie 3: Betriebssystem



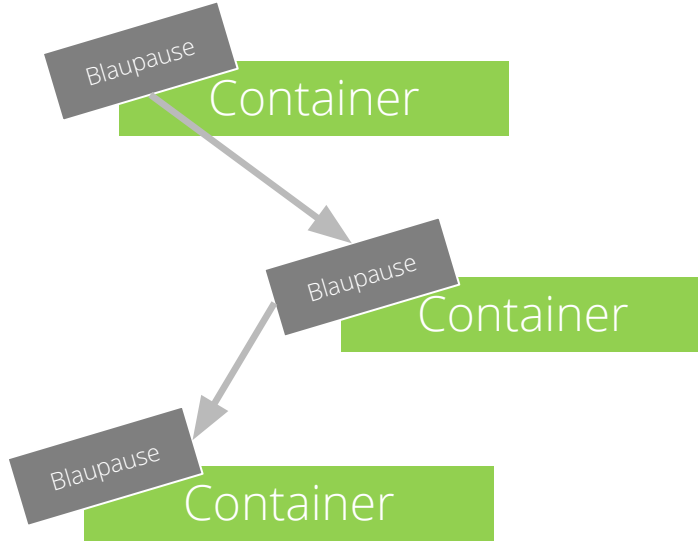
Cluster-Orchestrierer automatisieren verschiedene Betriebsaufgaben für Anwendungen auf einem Cluster (1/2):

- Container-Logistik: Verwaltung und Bereitstellung von Containern.
- Package-Management: Verwaltung und Bereitstellung von Applikationen.
- Bereitstellung von Administrationsschnittstellen (Remote-API, Kommandozeile).
- Management von Services: Service Discovery, Naming, Load Balancing.
- Automatismen für Rollout-Workflows wie z.B. Canary Rollout.
- Monitoring und Diagnose von Containern und Services.

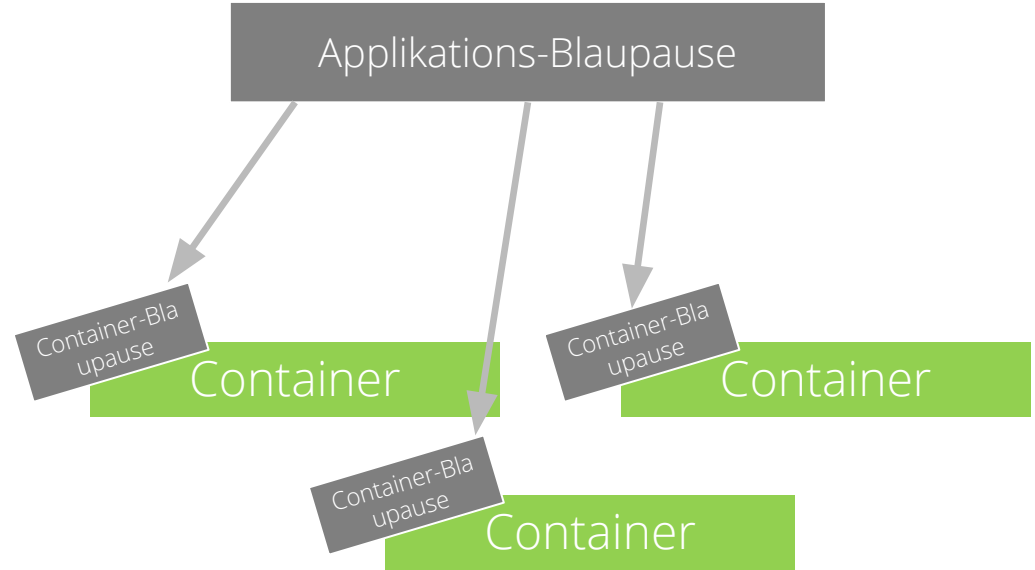
Cluster-Orchestrierer automatisieren verschiedene Betriebsaufgaben für Anwendungen auf einem Cluster (2/2):

- Scheduling von Containern mit applikationsspezifischen Constraints (z.B. Deployment- und Start-Reihenfolgen, Gruppierung, ...)
- Aufbau von notwendigen Netzwerk-Verbindungen zwischen Containern.
- Bereitstellung von persistenten Speichern für zustandsbehaftete Container.
- (Auto-)Skalierung von Containern.
- Re-Scheduling von Containern im Fehlerfall (Auto-Healing) oder zur Performance-Optimierung.

1-Level- vs. 2-Level-Orchestrierung



1-Level-Orchestrierung
(Container-Graph)



2-Level-Orchestrierung
(Container-Repository mit zentraler Bauanleitung)

1-Level- vs. 2-Level-Orchestrierung

Plain Docker

```
FROM ubuntu
ENTRYPOINT nginx
EXPOSE 80
```

```
docker run -d --link
nginx:nginx
```

1-Level-Orchestrierung
(Container-Graph)

<https://docs.docker.com/compose/compose-file>

Docker Compose

weba:

```
image: qaware/nginx
expose:
  - 80
```

webb:

```
image: qaware/nginx
expose:
  - 80
```

haproxy:

```
image: qaware/haproxy
links:
  - weba
  - webb
ports:
  - „80:80“
expose:
  - 80
```

FROM ubuntu
ENTRYPOINT nginx
EXPOSE 80

FROM ubuntu
ENTRYPOINT haproxy
EXPOSE 80

2-Level-Orchestrierung

(Container-Repository mit zentraler Bauanleitung)



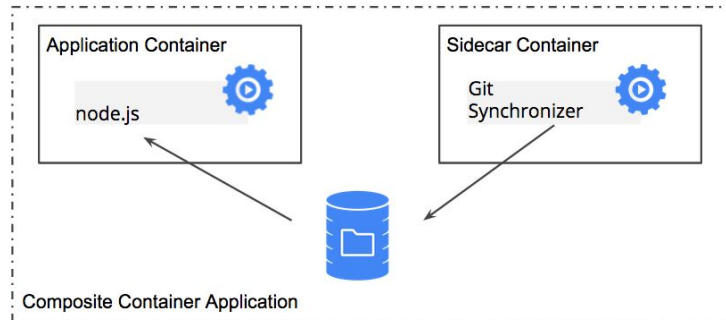
Übung: Orchestrierungsmuster

- Separation of Concerns mit modularen Containern

Sidecar Containers

Sidecar containers extend and enhance the “main” container, they take existing containers and make them better. As an example, consider a container that runs the Nginx web server. Add a different container that syncs the file system with a git repository, share the file system between the containers and you have built Git push-to-deploy. But you’ve done it in a modular manner where the git synchronizer can be built by a different team, and can be reused across many different web servers (Apache, Python, Tomcat, etc). **Because of this modularity, you only have to write and test your git synchronizer once and reuse it across numerous apps.** And if someone else writes it, you don’t even need to do that.

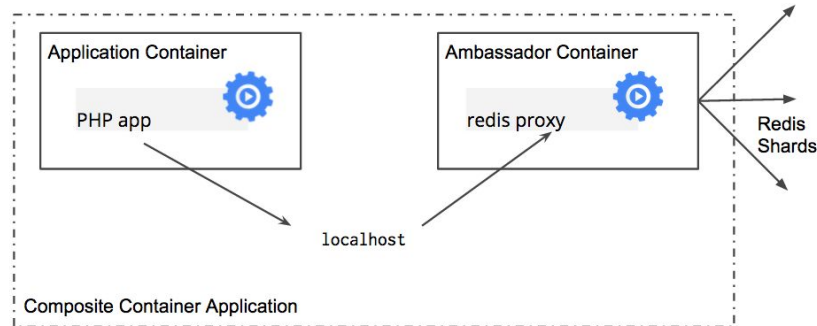
Quelle: <https://kubernetes.io/blog/2015/06/the-distributed-system-toolkit-patterns/>



Ambassador containers

Ambassador containers proxy a local connection to the world. As an example, consider a Redis cluster with read-replicas and a single write master. You can create a Pod that groups your main application with a Redis ambassador container. **The ambassador is a proxy is responsible for splitting reads and writes and sending them on to the appropriate servers.** Because these two containers share a network namespace, they share an IP address and your application can open a connection on “localhost” and find the proxy without any service discovery. **As far as your main application is concerned, it is simply connecting to a Redis server on localhost.** This is powerful, not just because of separation of concerns and the fact that different teams can easily own the components, but also because in the development environment, you can simply skip the proxy and connect directly to a Redis server that is running on localhost.

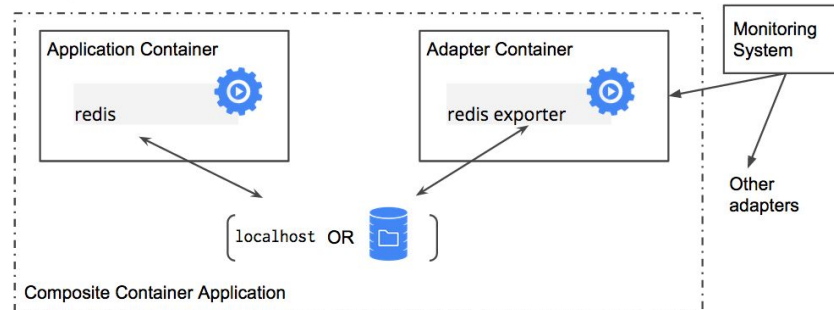
Quelle: <https://kubernetes.io/blog/2015/06/the-distributed-system-toolkit-patterns/>



Adapter containers

Adapter containers standardize and normalize output. Consider the task of monitoring N different applications. Each application may be built with a different way of exporting monitoring data. (e.g. JMX, StatsD, application specific statistics) but every monitoring system expects a consistent and uniform data model for the monitoring data it collects. **By using the adapter pattern of composite containers, you can transform the heterogeneous monitoring data from different systems into a single unified representation by creating Pods** that groups the application containers with adapters that know how to do the transformation. Again because these Pods share namespaces and file systems, the coordination of these two containers is simple and straightforward.

Quelle: <https://kubernetes.io/blog/2015/06/the-distributed-system-toolkit-patterns/>



The background of the image is a dark blue field filled with a complex, abstract pattern of thin white lines and small white dots. These elements form a network of interconnected geometric shapes, primarily triangles and polygons, creating a sense of depth and complexity. The pattern is non-repeating and organic, resembling a molecular structure or a digital network map.

Kubernetes



kubernetes

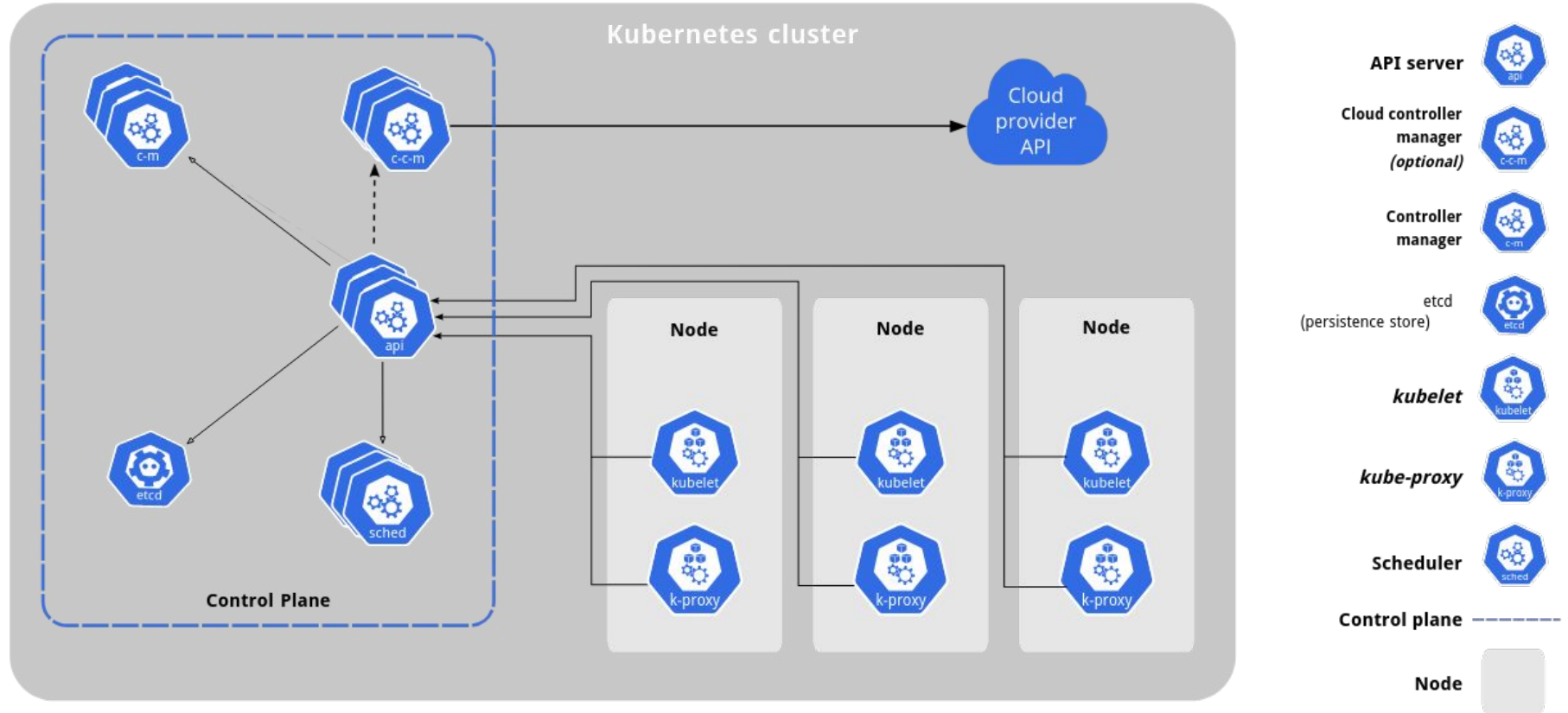
Kubernetes (auch als K8s bezeichnet) ist ein **Open-Source**-System zur **Automatisierung** der **Bereitstellung**, **Skalierung** und **Verwaltung** von **Container-Anwendungen**, das ursprünglich von Google entworfen und an die Cloud Native Computing Foundation (CNCF) gespendet wurde. Es zielt darauf ab, eine „Plattform für ... Anwendungscontainer[n] auf **verteilten Hosts**“ zu liefern. Es unterstützt eine Reihe von Container-Tools, einschließlich **Docker**.

[Quelle: Wikipedia](#)

Wieso?!

- Ungelöst: Effiziente Nutzung der Hardware in einem Rechenzentrum
 - Ein Server pro Team? Was passiert, wenn ein Team keinen ganzen braucht?
- Virtualisierung und Elastizität löst das Problem der Hardware-Beschaffung
- Ungelöst: Development & Deployment der Anwendungen
- Trend: Weg vom Monolithen, hin zu Microservices
 - Verstärkt das Problem des Deployments
 - Wie finden sich die Microservices gegenseitig?
- Trend: Zero-downtime Deployments
- Lösung: Kubernetes als Standard
 - Anwendung in Container verpacken
 - Deployment beschreiben
 - Kubernetes kümmert sich um den Betrieb: startet neu, skaliert, etc.
 - Zero-Downtime und Rollbacks inkl.

Architektur von Kubernetes



Aufgaben der Bausteine auf einem Node

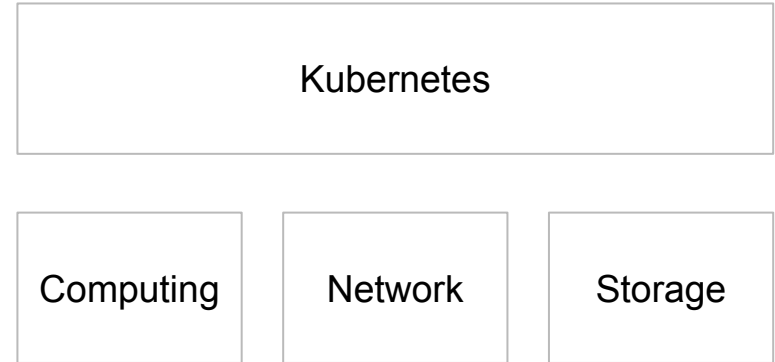
- **Pods** are the smallest deployable units of computing that you can create and manage in Kubernetes.
- Kubernetes runs your workload by placing containers into Pods to run on **Nodes**. A node may be a virtual or physical machine, depending on the cluster.
- **Container runtime**: "The container runtime is the software that is responsible for running containers. Kubernetes supports several container runtimes, one of them is Docker"
- **kubelet**: "An agent that runs on each node in the cluster. It makes sure that containers are running in a Pod."
- A **service** is an abstract way to expose an application running on a set of Pods as a network service.
- **kube-proxy**: "kube-proxy is a network proxy that runs on each node in your cluster, implementing part of the Kubernetes Service concept."

Aufgaben der Bausteine in der Control Plane

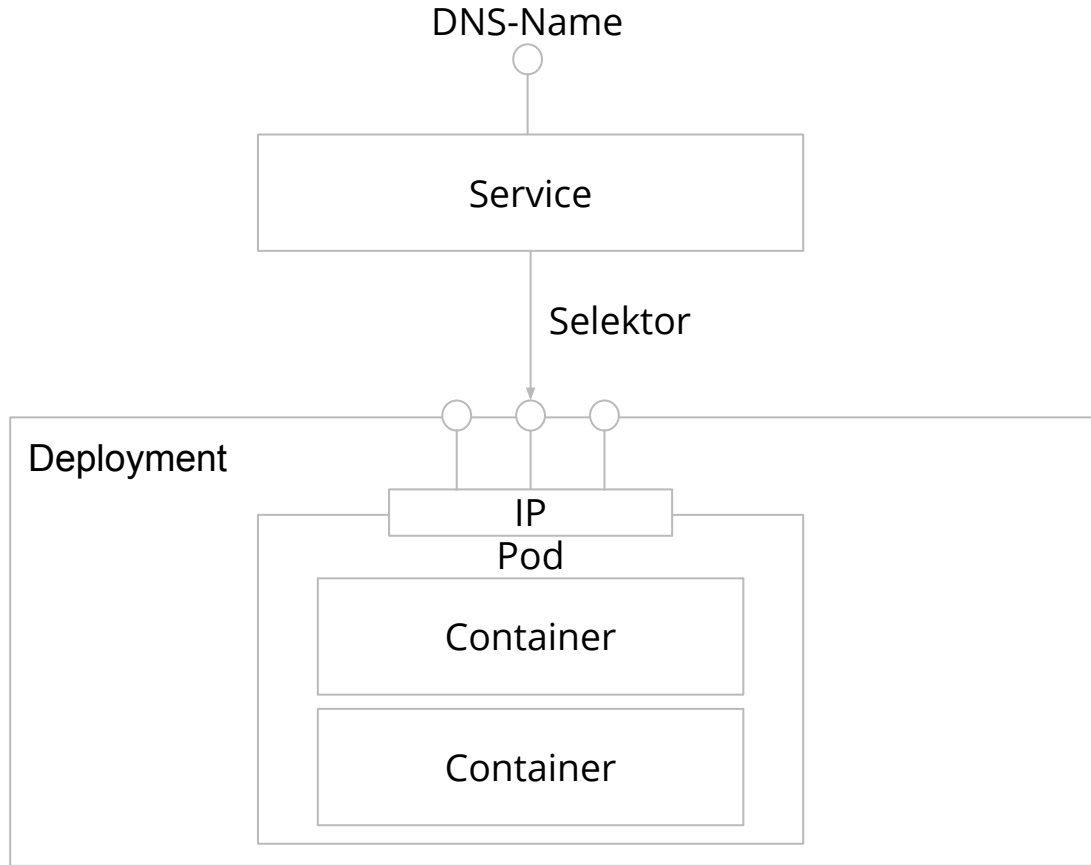
- **API Server:** "The API server is a component of the Kubernetes control plane that exposes the Kubernetes API. The API server is the front end for the Kubernetes control plane."
- **etcd:** "Consistent and highly-available key value store used as Kubernetes' backing store for all cluster data."
- **scheduler:** "Control plane component that watches for newly created Pods with no assigned node, and selects a node for them to run on."
- **Controller Manager:** "Control Plane component that runs controller processes."
 - Node controller: Responsible for noticing and responding when nodes go down.
 - Replication controller: Responsible for maintaining the correct number of pods for every replication controller object in the system.
 - ...
- **Cloud Controller Manager** (optional): "A Kubernetes control plane component that embeds cloud-specific control logic. The cloud controller manager lets you link your cluster into your cloud provider's API, and separates out the components that interact with that cloud platform from components that just interact with your cluster."

Kubernetes virtualisiert:

- Computing durch die Container Runtime
- Network durch virtuelle IPs und Overlay Networks
- Storage durch Persistent Volumes



Die wichtigsten Abstraktionen von Kubernetes



Die wichtigsten Abstraktionen von Kubernetes

Service: Endpunkt unter einem definierten DNS-Namen, der Aufrufe an die Pods im Hintergrund verteilt (Load Balancing, Failover). Die für einen Service relevanten Pods werden über ihre Labels selektiert, z.B.: `role = apache`, `env != test`, `tier in (web, app)`

Pod: Gruppe an Containern, die auf dem selben Knoten laufen und sich eine Netzwerk-Schnittstelle inklusive einer dedizierten IP, persistente Volumes und Umgebungsvariablen teilen. Ein Pod ist die atomare Scheduling-Einheit in K8s. Ein Pod kann über sog. Labels markiert werden, das sind frei definierbare Schlüssel-Wert-Paare.

Deployment: Klammer um einen gewünschten Zielzustand im Cluster in Form eines Pods mit dazugehörigem Replication Controller. Ein Deployment bezieht sich nicht auf Services, da diese in der K8s-Philosophie einen von Pods unabhängigen Lebenszyklus haben.



Live-Demo: Lokales Kubernetes mit minikube

Release v0.23.7

Release v0.23.7

Notes

Thank you to all our contributors who helped out, report bugs, and implemented, for 2023 try to make some of them easier to find. But if you don't mind, please let us know if you're having any issues with the tool. If you're having any issues, please let us know (please help me verify and close the support, and please let us know if you're having any issues with the tool).

If you're having any issues, please let us know (please help me verify and close the support, and please let us know if you're having any issues with the tool).


On 2023 Please join our [Discord](#)

Maintenance Release!

Resolved Issues/Features

Resolved PRs:

© 2023 Endless Software LLC. All rights reserved under Apache 2.0



Minikube

- [Minikube](#) installiert ein lokales Kubernetes-Cluster auf dem eigenen Rechner
- Eignet sich super zum Ausprobieren von Kubernetes
- [kubectl](#) ist ein Client für die Kubernetes API, die verwendet wird, um Kubernetes zu konfigurieren
- [K9s](#) ist eine Terminal UI für Kubernetes



Live-Demo: Pods & Deployment

Install and Set Up kubectl | Kubernetes - Official Website

kubernetes

Documentation | Kubernetes Blog | Pricing | Partners | Community | Case Studies | Videos | Sign In

KubeCon + CloudNativeCon NA 2020 virtual
A day of multiple sponsored sessions, hands, and talks across seven time zones
November 17 - 24 2020

Get started page
Kubernetes page
K8s page

Home
Getting started
Concepts
Tools
Install Tools
Install and Set Up kubectl
Administer a Cluster
Troubleshooting
Community
Manage Kubernetes Objects
Managing Security
Inject Data into Applications
Run Applications
Networks
Access Applications in a Cluster
Monitoring, Logging, and Debugging
Extend Kubernetes
Tutorials
Manage Cluster Settings
Service Catalog
Networking
Kubernetes Architecture
Manage Cloud Native

Install and Set Up kubectl

The Kubernetes command-line tool, **kubectl**, allows you to run commands against Kubernetes clusters. This page describes how to install **kubectl** on desktop, mobile, and server environments, and other topics that are a prerequisite for all **kubectl** operations. See [kubectl overview](#) for details.

Before you begin

You must use a **kubectl** version that is within one minor version difference of your cluster. For example, a v1.2 cluster should work with v1.1, v1.2, and v1.3 binaries. Using the latest version of **kubectl** helps avoid compatibility issues.

Install kubectl on Linux

Install kubectl binary with curl on Linux

1. Download the kubectl release with the command:

```
curl -LO https://kubernetes.io/docs/tutorials/kubernetes-basics/tutorial_01/ | sh
```

To download a specific version, replace the `latest` with the version number. For example, to download version v1.20.0 on Linux, type:


```
curl -LO https://kubernetes.io/docs/tutorials/kubernetes-basics/tutorial_01/ | sh
```

2. Make the **kubectl** binary executable:

<https://youtu.be/a5skHawFcIM>

Pods & Deployments

- [Pods](#) sind die kleinste schedulbare Einheit in Kubernetes
- Ein [Deployment](#) beschreibt, aus welchen Containern der Pod besteht und wie er konfiguriert ist



Übung, Aufgabe 1, 2 & 3
30 Minuten



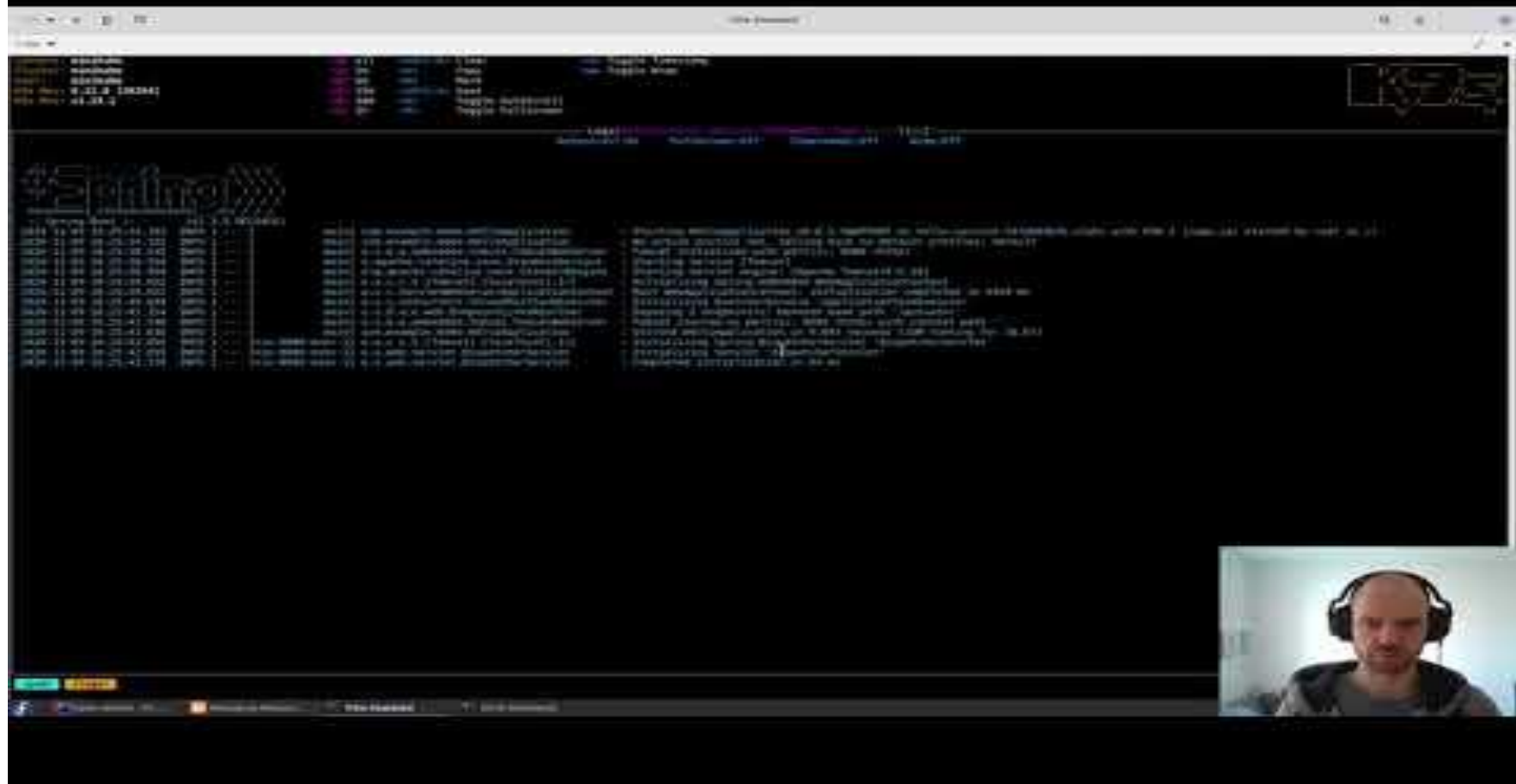
Live-Demo: Probes

Probes

- <https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle/#container-probes>
- Liveness probes prüfen, ob der Container noch am Leben ist. Falls nicht, wird dieser neu gestartet.
- Readiness probes prüfen, ob der Container bereit ist, Traffic zu bekommen. Falls nicht, wird dieser aus dem Load Balancing genommen.
- Mit einer Startup Probe kann das Problem von langsam startenden Containern mitigiert werden



Live-Demo: Resource Constraints



<https://youtu.be/dUyAueNWA1k>

Resource constraints

- <https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>
- Resource Requests helfen dem Scheduler, einen geeigneten Node für den Pod zu finden
- Resource Limits schützen vor amoklaufenden Containern, in dem sie den Container abschießen, sollte dieser die Memory-Limits erreichen
- Limits können auf Speicher und auf CPU gesetzt werden

Resource Constraints: CPU

Limits and requests for CPU resources are measured in cpu units. One cpu, in Kubernetes, is equivalent to 1 vCPU/Core for cloud providers and 1 hyperthread on bare-metal Intel processors.

Fractional requests are allowed. A Container with `spec.containers[].resources.requests.cpu` of 0.5 is guaranteed half as much CPU as one that asks for 1 CPU. The expression 0.1 is equivalent to the expression 100m, which can be read as "one hundred millicpu". Some people say "one hundred millicores", and this is understood to mean the same thing. A request with a decimal point, like 0.1, is converted to 100m by the API, and precision finer than 1m is not allowed. For this reason, the form 100m might be preferred.

CPU is always requested as an absolute quantity, never as a relative quantity; 0.1 is the same amount of CPU on a single-core, dual-core, or 48-core machine.

Quelle: <https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>



Übung, Aufgabe 4

15 Minuten



Live-Demo: Services

Keywords: *depression, mood, mood disorder, mood disorder with anxiety, mood disorder without anxiety, mood disorder with anxiety, mood disorder without anxiety, mood disorder with anxiety, mood disorder without anxiety*



- [Find this page](#)
- [Report a problem with this page](#)
- [Feedback](#)

- Home
- Getting started
- Concepts
- Download
- Cluster Architecture
- Installation
- Workloads
- Security, Load Balancing, and Monitoring**
- Scripts
- Related Readings
- What Our Readers Are Saying
- Customer Comments
- Feedback
- Blog Comments
- Related Comments
- Related Products
- Getting our book right
- Comments and discussions
- © 2011 Pearson Education, Inc.
- Language
- Translations
- Security
- Privacy
- Signaling and Flow

With Kalameter, you don't need to modify your application to use an unfamiliar remote discovery mechanism. Kalameter gets data that you'd otherwise need to write your own code for a set of facts, and we use known values from

Scholarship funds are created and managed by NAFSA for the benefit of your member. Funds are restricted resources. Please add a designated title to your grant use. Full credit will be given to your title.

Each Post paid to mail its address, included in a Department, the set of Posts having it and treated to this could not afford to be the set of Posts having that distribution as a direct set.

[illegible]

© 2004 Blackwell Publishing Ltd
Journal of Internal Medicine 255: 103–110

For Klemm, a *barrier* is an abstract notion without content. It signals a set of facts and a policy by which to decide those facts. Furthermore, this system is called a *barrier* because the set of facts suggested by a barrier is usually determined by a decision. To learn about what we mean by *barriers* here we must understand what *barriers* are and how they are used.

—It is important to note that the data were collected using a 3-point Likert scale (1 = strongly disagree, 2 = neutral, 3 = strongly agree). The data were analyzed using a 3-point Likert scale (1 = strongly disagree, 2 = neutral, 3 = strongly agree).

The Service will publish answers to this document as

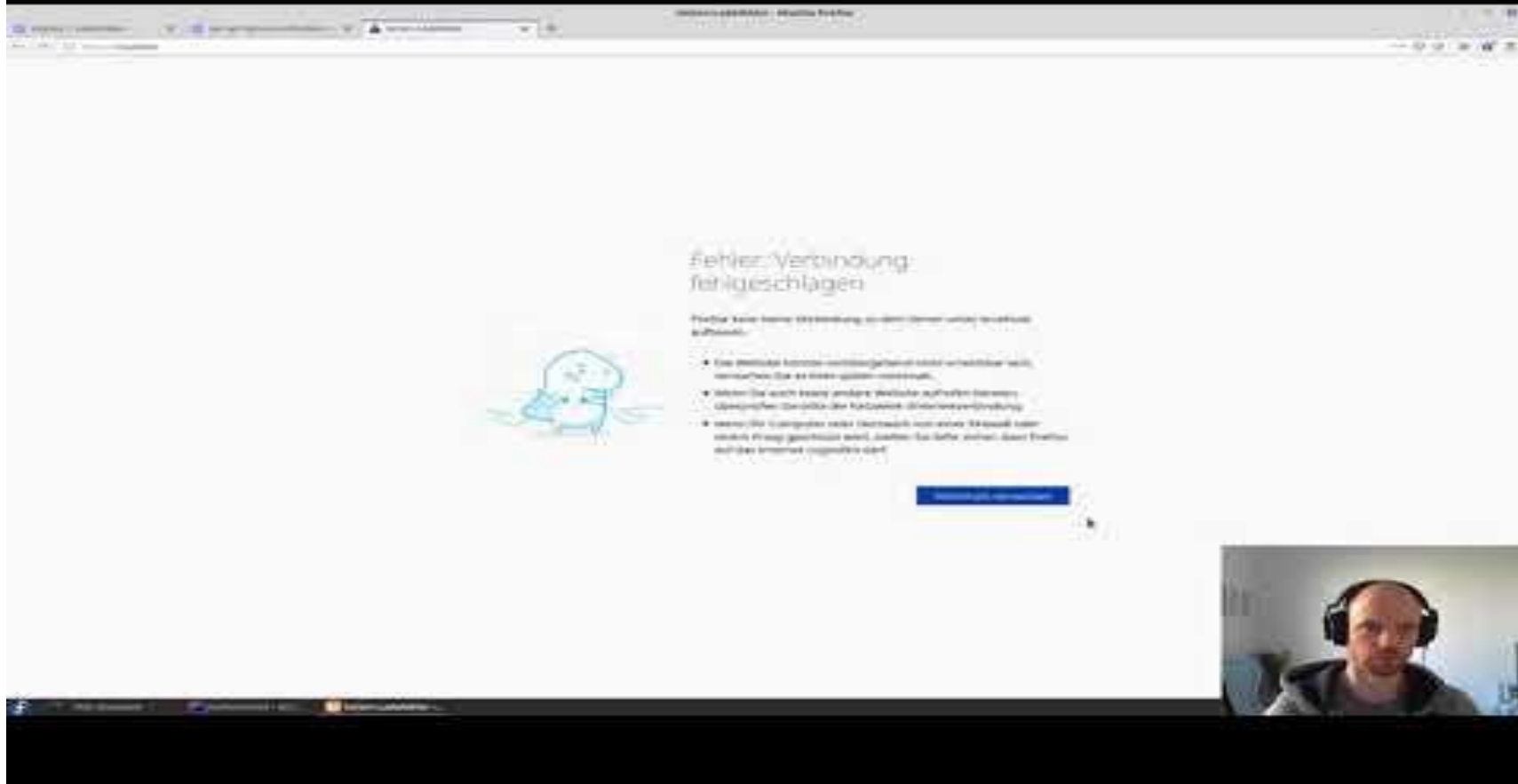


Services

- <https://kubernetes.io/docs/concepts/services-networking/service/>
- Services machen Anwendungen in Pods für andere Pods im Kubernetes-Cluster zugreifbar
- Services verwenden Label selectors, um die Pods zu finden, auf die der Service verweist
- Services load-balancen zwischen mehreren Pods



Live-Demo: Ingresses



<https://youtu.be/lkBag5znVjE>

Ingress

“An API object that manages **external access to the services** in a cluster, typically HTTP. Ingress may provide load balancing, SSL termination and name-based virtual hosting”

<https://kubernetes.io/docs/concepts/services-networking/ingress/>

- Durch einen Ingress kann ein Service von außerhalb des Clusters erreicht werden



Übung, Aufgabe 5

15 Minuten



Live-Demo: Config Maps

Subnetes

Documentation, Kubernetes Blog, Training, Partners, Community, Contributor, Version, English

Home, Getting started, Examples, Tutorials, What's New, Kubernetes & Linux, Configure Pods and Containers

Configure Pods and Containers

Define container environment variables using ConfigMap data

Define a container environment variable with data from a single ConfigMap

1. Define an environment variable as a key-value pair in a ConfigMap

2. Inject the ConfigMap data into the container's environment

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: configmap-env
data:
  MY_ENV_VAR: myenvval
```

```
apiVersion: v1
kind: Pod
metadata:
  name: pod-env
spec:
  containers:
  - name: my-container
    image: nginx
    env:
    - name: MY_ENV_VAR
      valueFrom:
        configMapKeyRef:
          name: configmap-env
          key: MY_ENV_VAR
```

With this code, you can create a ConfigMap and a Pod that uses the ConfigMap data to set an environment variable.

1. Create a ConfigMap with the environment variable data.

2. Create a Pod that uses the ConfigMap data to set an environment variable.

3. Verify that the environment variable is set in the Pod.

Config Maps

- [Config Maps](#) werden verwendet, um eine Anwendung in Kubernetes zu konfigurieren
- Die Werte von Config-Maps können entweder als Umgebungsvariablen oder als Dateien im Container erscheinen

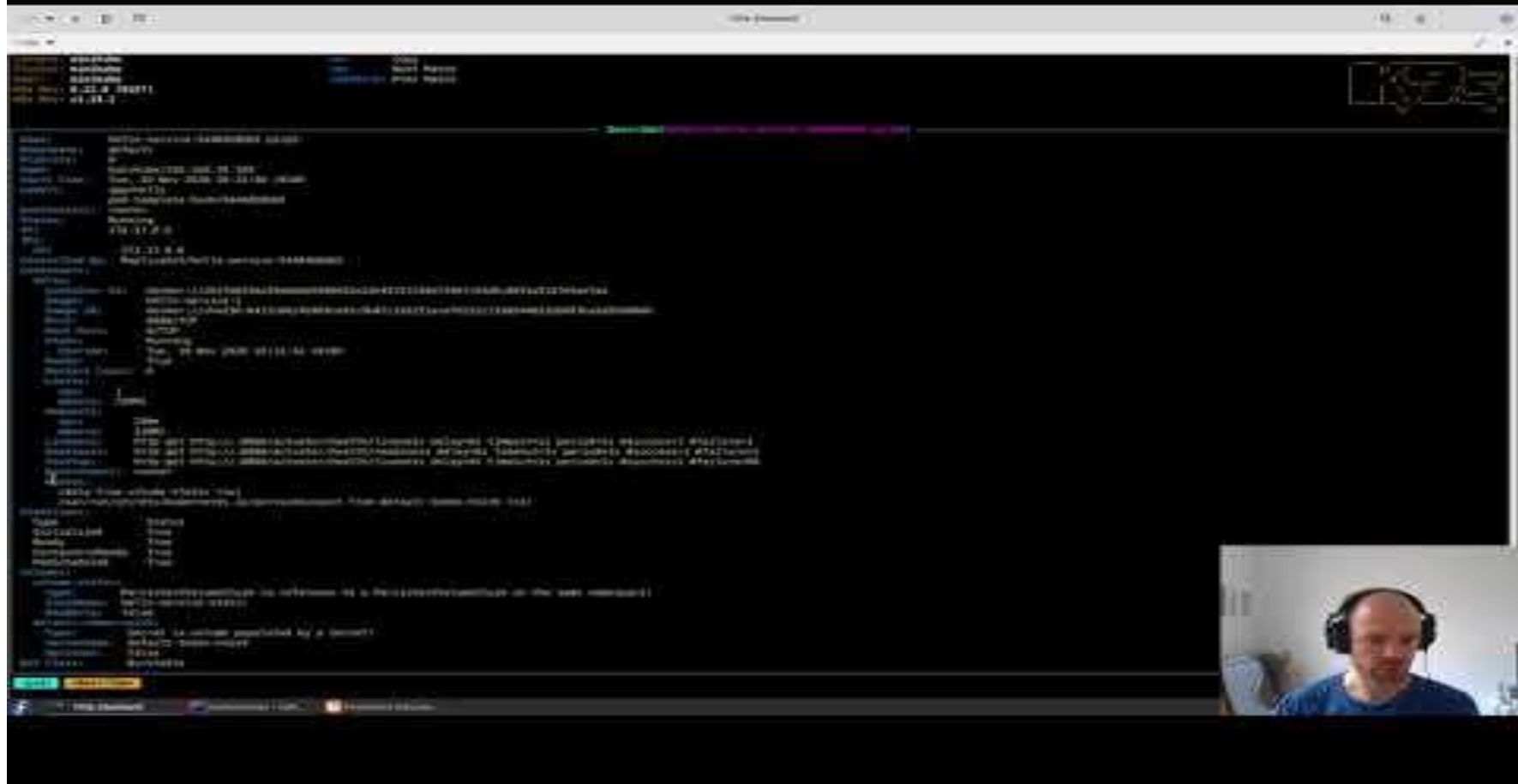


Übung, Aufgabe 6

15 Minuten



Live-Demo: Persistent Volumes



https://youtu.be/P35_Ya2VxME

Volumes

- Persistent Volumes stellen Speicherplatz bereit
 - Diese können entweder statisch oder dynamisch provisioniert werden
 - Dies ist für den Admin des Kubernetes-Clusters interessant
- Persistent Volume Claims fordern ein Persistent Volume an
 - Dies ist für den Entwickler des Kubernetes-Clusters interessant
- Über ein Volume kann ein Pod ein Persistent Volume Claim verwenden
- Volume mounts lassen das Volume in einem Container über einen Linux-Mount erscheinen



Übung, Aufgabe 7

15 Minuten



Live-Demo: Helm, der Package Manager

YouTube video player interface showing a transcript of a presentation. The transcript is displayed in two columns, with the left column containing the main text and the right column containing a smaller, partially visible version of the same text. The text appears to be a list of items, possibly a table of contents or a list of references, with some items highlighted in green. The video player includes standard controls like play/pause, volume, and a progress bar. A small inset window in the bottom right corner shows a person wearing a headset, likely the presenter.

Transcript (Left Column):

- 1.1.1
- 1.1.2
- 1.1.3
- 1.1.4
- 1.1.5
- 1.1.6
- 1.1.7
- 1.1.8
- 1.1.9
- 1.1.10
- 1.1.11
- 1.1.12
- 1.1.13
- 1.1.14
- 1.1.15
- 1.1.16
- 1.1.17
- 1.1.18
- 1.1.19
- 1.1.20
- 1.1.21
- 1.1.22
- 1.1.23
- 1.1.24
- 1.1.25
- 1.1.26
- 1.1.27
- 1.1.28
- 1.1.29
- 1.1.30
- 1.1.31
- 1.1.32
- 1.1.33
- 1.1.34
- 1.1.35
- 1.1.36
- 1.1.37
- 1.1.38
- 1.1.39
- 1.1.40
- 1.1.41
- 1.1.42
- 1.1.43
- 1.1.44
- 1.1.45
- 1.1.46
- 1.1.47
- 1.1.48
- 1.1.49
- 1.1.50
- 1.1.51
- 1.1.52
- 1.1.53
- 1.1.54
- 1.1.55
- 1.1.56
- 1.1.57
- 1.1.58
- 1.1.59
- 1.1.60
- 1.1.61
- 1.1.62
- 1.1.63
- 1.1.64
- 1.1.65
- 1.1.66
- 1.1.67
- 1.1.68
- 1.1.69
- 1.1.70
- 1.1.71
- 1.1.72
- 1.1.73
- 1.1.74
- 1.1.75
- 1.1.76
- 1.1.77
- 1.1.78
- 1.1.79
- 1.1.80
- 1.1.81
- 1.1.82
- 1.1.83
- 1.1.84
- 1.1.85
- 1.1.86
- 1.1.87
- 1.1.88
- 1.1.89
- 1.1.90
- 1.1.91
- 1.1.92
- 1.1.93
- 1.1.94
- 1.1.95
- 1.1.96
- 1.1.97
- 1.1.98
- 1.1.99
- 1.2.00
- 1.2.01
- 1.2.02
- 1.2.03
- 1.2.04
- 1.2.05
- 1.2.06
- 1.2.07
- 1.2.08
- 1.2.09
- 1.2.10
- 1.2.11
- 1.2.12
- 1.2.13
- 1.2.14
- 1.2.15
- 1.2.16
- 1.2.17
- 1.2.18
- 1.2.19
- 1.2.20
- 1.2.21
- 1.2.22
- 1.2.23
- 1.2.24
- 1.2.25
- 1.2.26
- 1.2.27
- 1.2.28
- 1.2.29
- 1.2.30
- 1.2.31
- 1.2.32
- 1.2.33
- 1.2.34
- 1.2.35
- 1.2.36
- 1.2.37
- 1.2.38
- 1.2.39
- 1.2.40
- 1.2.41
- 1.2.42
- 1.2.43
- 1.2.44
- 1.2.45
- 1.2.46
- 1.2.47
- 1.2.48
- 1.2.49
- 1.2.50
- 1.2.51
- 1.2.52
- 1.2.53
- 1.2.54
- 1.2.55
- 1.2.56
- 1.2.57
- 1.2.58
- 1.2.59
- 1.2.60
- 1.2.61
- 1.2.62
- 1.2.63
- 1.2.64
- 1.2.65
- 1.2.66
- 1.2.67
- 1.2.68
- 1.2.69
- 1.2.70
- 1.2.71
- 1.2.72
- 1.2.73
- 1.2.74
- 1.2.75
- 1.2.76
- 1.2.77
- 1.2.78
- 1.2.79
- 1.2.80
- 1.2.81
- 1.2.82
- 1.2.83
- 1.2.84
- 1.2.85
- 1.2.86
- 1.2.87
- 1.2.88
- 1.2.89
- 1.2.90
- 1.2.91
- 1.2.92
- 1.2.93
- 1.2.94
- 1.2.95
- 1.2.96
- 1.2.97
- 1.2.98
- 1.2.99
- 1.3.00
- 1.3.01
- 1.3.02
- 1.3.03
- 1.3.04
- 1.3.05
- 1.3.06
- 1.3.07
- 1.3.08
- 1.3.09
- 1.3.10
- 1.3.11
- 1.3.12
- 1.3.13
- 1.3.14
- 1.3.15
- 1.3.16
- 1.3.17
- 1.3.18
- 1.3.19
- 1.3.20
- 1.3.21
- 1.3.22
- 1.3.23
- 1.3.24
- 1.3.25
- 1.3.26
- 1.3.27
- 1.3.28
- 1.3.29
- 1.3.30
- 1.3.31
- 1.3.32
- 1.3.33
- 1.3.34
- 1.3.35
- 1.3.36
- 1.3.37
- 1.3.38
- 1.3.39
- 1.3.40
- 1.3.41
- 1.3.42
- 1.3.43
- 1.3.44
- 1.3.45
- 1.3.46
- 1.3.47
- 1.3.48
- 1.3.49
- 1.3.50
- 1.3.51
- 1.3.52
- 1.3.53
- 1.3.54
- 1.3.55
- 1.3.56
- 1.3.57
- 1.3.58
- 1.3.59
- 1.3.60
- 1.3.61
- 1.3.62
- 1.3.63
- 1.3.64
- 1.3.65
- 1.3.66
- 1.3.67
- 1.3.68
- 1.3.69
- 1.3.70
- 1.3.71
- 1.3.72
- 1.3.73
- 1.3.74
- 1.3.75
- 1.3.76
- 1.3.77
- 1.3.78
- 1.3.79
- 1.3.80
- 1.3.81
- 1.3.82
- 1.3.83
- 1.3.84
- 1.3.85
- 1.3.86
- 1.3.87
- 1.3.88
- 1.3.89
- 1.3.90
- 1.3.91
- 1.3.92
- 1.3.93
- 1.3.94
- 1.3.95
- 1.3.96
- 1.3.97
- 1.3.98
- 1.3.99
- 1.4.00
- 1.4.01
- 1.4.02
- 1.4.03
- 1.4.04
- 1.4.05
- 1.4.06
- 1.4.07
- 1.4.08
- 1.4.09
- 1.4.10
- 1.4.11
- 1.4.12
- 1.4.13
- 1.4.14
- 1.4.15
- 1.4.16
- 1.4.17
- 1.4.18
- 1.4.19
- 1.4.20
- 1.4.21
- 1.4.22
- 1.4.23
- 1.4.24
- 1.4.25
- 1.4.26
- 1.4.27
- 1.4.28
- 1.4.29
- 1.4.30
- 1.4.31
- 1.4.32
- 1.4.33
- 1.4.34
- 1.4.35
- 1.4.36
- 1.4.37
- 1.4.38
- 1.4.39
- 1.4.40
- 1.4.41
- 1.4.42
- 1.4.43
- 1.4.44
- 1.4.45
- 1.4.46
- 1.4.47
- 1.4.48
- 1.4.49
- 1.4.50
- 1.4.51
- 1.4.52
- 1.4.53
- 1.4.54
- 1.4.55
- 1.4.56
- 1.4.57
- 1.4.58
- 1.4.59
- 1.4.60
- 1.4.61
- 1.4.62
- 1.4.63
- 1.4.64
- 1.4.65
- 1.4.66
- 1.4.67
- 1.4.68
- 1.4.69
- 1.4.70
- 1.4.71
- 1.4.72
- 1.4.73
- 1.4.74
- 1.4.75
- 1.4.76
- 1.4.77
- 1.4.78
- 1.4.79
- 1.4.80
- 1.4.81
- 1.4.82
- 1.4.83
- 1.4.84
- 1.4.85
- 1.4.86
- 1.4.87
- 1.4.88
- 1.4.89
- 1.4.90
- 1.4.91
- 1.4.92
- 1.4.93
- 1.4.94
- 1.4.95
- 1.4.96
- 1.4.97
- 1.4.98
- 1.4.99
- 1.5.00
- 1.5.01
- 1.5.02
- 1.5.03
- 1.5.04
- 1.5.05
- 1.5.06
- 1.5.07
- 1.5.08
- 1.5.09
- 1.5.10
- 1.5.11
- 1.5.12
- 1.5.13
- 1.5.14
- 1.5.15
- 1.5.16
- 1.5.17
- 1.5.18
- 1.5.19
- 1.5.20
- 1.5.21
- 1.5.22
- 1.5.23
- 1.5.24
- 1.5.25
- 1.5.26
- 1.5.27
- 1.5.28
- 1.5.29
- 1.5.30
- 1.5.31
- 1.5.32
- 1.5.33
- 1.5.34
- 1.5.35
- 1.5.36
- 1.5.37
- 1.5.38
- 1.5.39
- 1.5.40
- 1.5.41
- 1.5.42
- 1.5.43
- 1.5.44
- 1.5.45
- 1.5.46
- 1.5.47
- 1.5.48
- 1.5.49
- 1.5.50
- 1.5.51
- 1.5.52
- 1.5.53
- 1.5.54
- 1.5.55
- 1.5.56
- 1.5.57
- 1.5.58
- 1.5.59
- 1.5.60
- 1.5.61
- 1.5.62
- 1.5.63
- 1.5.64
- 1.5.65
- 1.5.66
- 1.5.67
- 1.5.68
- 1.5.69
- 1.5.70
- 1.5.71
- 1.5.72
- 1.5.73
- 1.5.74
- 1.5.75
- 1.5.76
- 1.5.77
- 1.5.78
- 1.5.79
- 1.5.80
- 1.5.81
- 1.5.82
- 1.5.83
- 1.5.84
- 1.5.85
- 1.5.86
- 1.5.87
- 1.5.88
- 1.5.89
- 1.5.90
- 1.5.91
- 1.5.92
- 1.5.93
- 1.5.94
- 1.5.95
- 1.5.96
- 1.5.97
- 1.5.98
- 1.5.99
- 1.6.00
- 1.6.01
- 1.6.02
- 1.6.03
- 1.6.04
- 1.6.05
- 1.6.06
- 1.6.07
- 1.6.08
- 1.6.09
- 1.6.10
- 1.6.11
- 1.6.12
- 1.6.13
- 1.6.14
- 1.6.15
- 1.6.16
- 1.6.17
- 1.6.18
- 1.6.19
- 1.6.20
- 1.6.21
- 1.6.22
- 1.6.23
- 1.6.24
- 1.6.25
- 1.6.26
- 1.6.27
- 1.6.28
- 1.6.29
- 1.6.30
- 1.6.31
- 1.6.32
- 1.6.33
- 1.6.34
- 1.6.35
- 1.6.36
- 1.6.37
- 1.6.38
- 1.6.39
- 1.6.40
- 1.6.41
- 1.6.42
- 1.6.43
- 1.6.44
- 1.6.45
- 1.6.46
- 1.6.47
- 1.6.48
- 1.6.49
- 1.6.50
- 1.6.51
- 1.6.52
- 1.6.53
- 1.6.54
- 1.6.55
- 1.6.56
- 1.6.57
- 1.6.58
- 1.6.59
- 1.6.60
- 1.6.61
- 1.6.62
- 1.6.63
- 1.6.64
- 1.6.65
- 1.6.66
- 1.6.67
- 1.6.68
- 1.6.69
- 1.6.70
- 1.6.71
- 1.6.72
- 1.6.73
- 1.6.74
- 1.6.75
- 1.6.76
- 1.6.77
- 1.6.78
- 1.6.79
- 1.6.80
- 1.6.81
- 1.6.82
- 1.6.83
- 1.6.84
- 1.6.85
- 1.6.86
- 1.6.87
- 1.6.88
- 1.6.89
- 1.6.90
- 1.6.91
- 1.6.92
- 1.6.93
- 1.6.94
- 1.6.95
- 1.6.96
- 1.6.97
- 1.6.98
- 1.6.99
- 1.7.00
- 1.7.01
- 1.7.02
- 1.7.03
- 1.7.04
- 1.7.05
- 1.7.06
- 1.7.07
- 1.7.08
- 1.7.09
- 1.7.10
- 1.7.11
- 1.7.12
- 1.7.13
- 1.7.14
- 1.7.15
- 1.7.16
- 1.7.17
- 1.7.18
- 1.7.19
- 1.7.20
- 1.7.21
- 1.7.22
- 1.7.23
- 1.7.24
- 1.7.25
- 1.7.26
- 1.7.27
- 1.7.28
- 1.7.29
- 1.7.30
- 1.7.31
- 1.7.32
- 1.7.33
- 1.7.34
- 1.7.35
- 1.7.36
- 1.7.37
- 1.7.38
- 1.7.39
- 1.7.40
- 1.7.41
- 1.7.42
- 1.7.43
- 1.7.44
- 1.7.45
- 1.7.46
- 1.7.47
- 1.7.48
- 1.7.49
- 1.7.50
- 1.7.51
- 1.7.52
- 1.7.53
- 1.7.54
- 1.7.55
- 1.7.56
- 1.7.57
- 1.7.58
- 1.7.59
- 1.7.60
- 1.7.61
- 1.7.62
- 1.7.63
- 1.7.64
- 1.7.65
- 1.7.66
- 1.7.67
- 1.7.68
- 1.7.69
- 1.7.70
- 1.7.71
- 1.7.72
- 1.7.73
- 1.7.74
- 1.7.75
- 1.7.76
- 1.7.77
- 1.7.78
- 1.7.79
- 1.7.80
- 1.7.81
- 1.7.82
- 1.7.83
- 1.7.84
- 1.7.85
- 1.7.86
- 1.7.87
- 1.7.88
- 1.7.89
- 1.7.90
- 1.7.91
- 1.7.92
- 1.7.93
- 1.7.94
- 1.7.95
- 1.7.96
- 1.7.97
- 1.7.98
- 1.7.99
- 1.8.00
- 1.8.01
- 1.8.02
- 1.8.03
- 1.8.04
- 1.8.05
- 1.8.06
- 1.8.07
- 1.8.08
- 1.8.09
- 1.8.10
- 1.8.11
- 1.8.12
- 1.8.13
- 1.8.14
- 1.8.15
- 1.8.16
- 1.8.17
- 1.8.18
- 1.8.19
- 1.8.20
- 1.8.21
- 1.8.22
- 1.8.23
- 1.8.24
- 1.8.25
- 1.8.26
- 1.8.27
- 1.8.28
- 1.8.29
- 1.8.30
- 1.8.31
- 1.8.32
- 1.8.33
- 1.8.34
- 1.8.35
- 1.8.36
- 1.8.37
- 1.8.38
- 1.8.39
- 1.8.40
- 1.8.41
- 1.8.42
- 1.8.43
- 1.8.44
- 1.8.45
- 1.8.46
- 1.8.47
- 1.8.48
- 1.8.49
- 1.8.50
- 1.8.51
- 1.8.52
- 1.8.53
- 1.8.54
- 1.8.55
- 1.8.56
- 1.8.57
- 1.8.58
- 1.8.59
- 1.8.60
- 1.8.61
- 1.8.62
- 1.8.63
- 1.8.64
- 1.8.65
- 1.8.66
- 1.8.67
- 1.8.68
- 1.8.69
- 1.8.70
- 1.8.71
- 1.8.72
- 1.8.73
- 1.8.74
- 1.8.75
- 1.8.76
- 1.8.77
- 1.8.78
- 1.8.79
- 1.8.80
- 1.8.81
- 1.8.82
- 1.8.83
- 1.8.84
- 1.8.85
- 1.8.86
- 1.8.87
- 1.8.88
- 1.8.89
- 1.8.90
- 1.8.91
- 1.8.92
- 1.8.93
- 1.8.94
- 1.8.95
- 1.8.96
- 1.8.97
- 1.8.98
- 1.8.99
- 1.9.00
- 1.9.01
- 1.9.02
- 1.9.03
- 1.9.04
- 1.9.05
- 1.9.06
- 1.9.07
- 1.9.08
- 1.9.09
- 1.9.10
- 1.9.11
- 1.9.12
- 1.9.13
- 1.9.14
- 1.9.15
- 1.9.16
- 1.9.17
- 1.9.18
- 1.9.19
- 1.9.20
- 1.9.21
- 1.9.22
- 1.9.23
- 1.9.24
- 1.9.25
- 1.9.26
- 1.9.27
- 1.9.28
- 1.9.29
- 1.9.30
- 1.9.31
- 1.9.32
- 1.9.33
- 1.9.34
- 1.9.35
- 1.9.36
- 1.9.37
- 1.9.38
- 1.9.39
- 1.9.40
- 1.9.41
- 1.9.42
- 1.9.43
- 1.9.44
- 1.9.45
- 1.9.46
- 1.9.47
- 1.9.48
- 1.9.49
- 1.9.50
- 1.9.51
- 1.9.52
- 1.9.53
- 1.9.54
- 1.9.55
- 1.9.56
- 1.9.57
- 1.9.58
- 1.9.59
- 1.9.60
- 1.9.61
- 1.9.62
- 1.9.63
- 1.9.64
- 1.9.65
- 1.9.66
- 1.9.67
- 1.9.68
- 1.9.69
- 1.9.70
- 1.9.71
- 1.9.72
- 1.9.73
- 1.9.74
- 1.9.75
- 1.9.76
- 1.9.77
- 1.9.78
- 1.9.79
- 1.9.80
- 1.9.81
- 1.9.82
- 1.9.83
- 1.9.84
- 1.9.85
- 1.9.86
- 1.9.87
- 1.9.88
- 1.9.89
- 1.9.90
- 1.9.91
- 1.9.92
- 1.9.93
- 1.9.94
- 1.9.95
- 1.9.96
- 1.9.97
- 1.9.98
- 1.9.99
- 2.0.00
- 2.0.01
- 2.0.02
- 2.0.03
- 2.0.04
- 2.0.05
- 2.0.06
- 2.0.07
- 2.0.08
- 2.0.09
- 2.0.10
- 2.0.11
- 2.0.12
- 2.0.13
- 2.0.14
- 2.0.15
- 2.0.16
- 2.0.17
- 2.0.18
- 2.0.19
- 2.0.20
- 2.0.21
- 2.0.22
- 2.0.23
- 2.0.24
- 2.0.25
- 2.0.26
- 2.0.27
- 2.0.28
- 2.0.29
- 2.0.30
- 2.0.31
- 2.0.32
- 2.0.33
- 2.0.34
- 2.0.35
- 2.0.36
- 2.0.37
- 2.0.38
- 2.0.39
- 2.0.40
- 2.0.41
- 2.0.42
- 2.0.43
- 2.0.44
- 2.0.45
- 2.0.46
- 2.0.47
- 2.0.48
- 2.0.49
- 2.0.50
- 2.0.51
- 2.0.52
- 2.0.53
- 2.0.54
- 2.0.55
- 2.0.56
- 2.0.57
- 2.0.58
- 2.0.59
- 2.0.60
- 2.0.61
- 2.0.62
- 2.0.63
- 2.0.64
- 2.0.65
- 2.0.66
- 2.0.67
- 2.0.68
- 2.0.69
- 2.0.70
- 2.0.71
- 2.0.72
- 2.0.73
- 2.0.74
- 2.0.75
- 2.0.76
- 2.0.77
- 2.0.78
- 2.0.79
- 2.0.80
- 2.0.81
- 2.0.82
- 2.0.83
- 2.0.84
- 2.0.85
- 2.0.86
- 2.0.87
- 2.0.88
- 2.0.89
- 2.0.90
- 2.0.91
- 2.0.92
- 2.0.93
- 2.0.94
- 2.0.95
- 2.0.96
- 2.0.97
- 2.0.98
- 2.0.99
- 2.1.00
- 2.1.01
- 2.1.02
- 2.1.03
- 2.1.04
- 2.1.05
- 2.1.06
- 2.1.07
- 2.1.08
- 2.1.09
- 2.1.10
- 2.1.11
- 2.1.12
- 2.1.13
- 2.1.14
- 2.1.15
- 2.1.16
- 2.1.17
- 2.1.18
- 2.1.19
- 2.1.20
- 2.1.21
- 2.1.22
- 2.1.23
- 2.1.24
- 2.1.25
- 2.1.26
- 2.1.27
- 2.1.28
- 2.1.29
- 2.1.30
- 2.1.31
- 2.1.32
- 2.1.33
- 2.1.34
- 2.1.35
- 2.1.36
- 2.1.37
- 2.1.38
- 2.1.39
- 2.1.40
- 2.1.41
- 2.1.42
- 2.1.43
- 2.1.44
- 2.1.45
- 2.1.46
- 2.1.47
- 2.1.48
- 2.1.49
- 2.1.50
- 2.1.51
- 2.1.52
- 2.1.53
- 2.1.54
- 2.1.55
- 2.1.56
- 2.1.57
- 2.1.58
- 2.1.59
- 2.1.60
- 2.1.61
- 2.1.62
- 2.1.63
- 2.1.64
- 2.1.65
- 2.1.66
- 2.1.67
- 2.1.68
- 2.1.69
- 2.1.70
- 2.1.71
- 2.1.72
- 2.1.73
- 2.1.74
- 2.1.75
- 2.1.76
- 2.1.77
- 2.1.78
- 2.1.79
- 2.1.80
- 2.1.81
- 2.1.82
- 2.1.83
- 2.1.84
- 2.1.85
- 2.1.86
- 2.1.87
- 2.1.88
- 2.1.89
- 2.1.90
- 2.1.91
- 2.1.92
- 2.1.93
- 2.1.94
- 2.1.95
- 2.1.96
- 2.1.97
- 2.1.98
- 2.1.99
- 2.2.00
- 2.2.01
- 2.2.02
- 2.2.03
- 2.2.04
- 2.2.05
- 2.2.06
- 2.2.07
- 2.2.08
- 2.2.09
- 2.2.10
- 2.2.11
- 2.2.12
- 2.2.13</

Helm

- <https://helm.sh/>
- Paket-Manager für Kubernetes
- Erlaubt das Installieren von Charts in den Cluster
- Charts gibt es für alles mögliche: Redis, PostgreSQL, etc.
- Es gibt eine [Suchmaschine für Charts](#)
- Charts können über den `--set` oder den `--values` Parameter angepasst werden



(Optional) Übung, Aufgabe 8

20 Minuten