

# **Docker & Kubernetes**

## **Teil 4**

# Teil 4 – Docker Compose & K8s

- Docker Compose
  - CLI Kommandos
  - Konfigurations-Files
- Kubernetes
  - Grundlagen
  - Konfiguration von Anwendungen

# Docker Compose

- Installation
  - In Docker Desktop (automatisch)
  - In Docker Engine als Plugin
    - In CMD-Docker-auf-Linux-Server.txt enthalten

```
$ sudo apt-get update  
$ sudo apt-get install docker-compose-plugin
```



- Vorbereitetes VirtualBox Image

# Docker Compose - CLI

- Allgemeiner Aufruf

```
$ docker compose [OPTIONS] COMMAND [ARGS]
```

- Auch docker-compose bei Nutzung des Binaries

- Wesentliche Options

- f, --file <file-name>

- Die Konfigurationsdatei
    - Kann mehrfach vorkommen ... Inhalte werden kombiniert
    - Ohne -f wird immer docker-compose.yml genutzt

- profile <profile>

- Auswahl der Services des angegebenen Profils
    - Kann mehrfach vorkommen

- help

```
$ docker compose --help
```



# Docker Compose - CLI

- Steuerung der Konfiguration
  - YAML-Dateien
  - Umgebungsvariablen
  - Parameter der Kommandos

=> Beschreibung von Services mit allen ihren Ressourcen
- Ein Beispiel: Wordpress
  - Wordpress-WEB-Server und MariaDB-Server
  - Vorbereitete Datei `docker-compose-wordpress.yml`



# Docker Compose - CLI

- Wordpress erzeugen und starten

```
$ docker compose -f docker-compose-wordpress.yml up
```

- Aufruf mit <http://localhost:8001>
- Inspektion der Docker Objekte in Portainer
- Wordpress stoppen und entfernen

```
$ docker compose -f docker-compose-wordpress.yml down
```

- Inspektion in Portainer



# Docker Compose - CLI

- Wesentliche Kommandos
  - create ... Docker Objekte erzeugen
  - start ... Service starten
  - stop ... Service anhalten
  - up ... create und start kombiniert
  - down ... stop und Entfernen aller Objekte
  - pull ... images aktualisieren
  - logs ... logs zeigen

# Docker Compose - CLI

- Laufende Compose Projekte auflisten

```
$ docker compose ls
```



```
Terminal - mb@mb-GL502VMZ ~/Desktop/Schulung-Docker
Datei Bearbeiten Ansicht Terminal Reiter Hilfe
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $ docker compose ls
NAME      STATUS        CONFIG FILES
schulung-docker    running(2)   /home/mb/Desktop/Schulung-Docker/docker-compose-wordpress.yml
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $
```

# Docker Compose - CLI

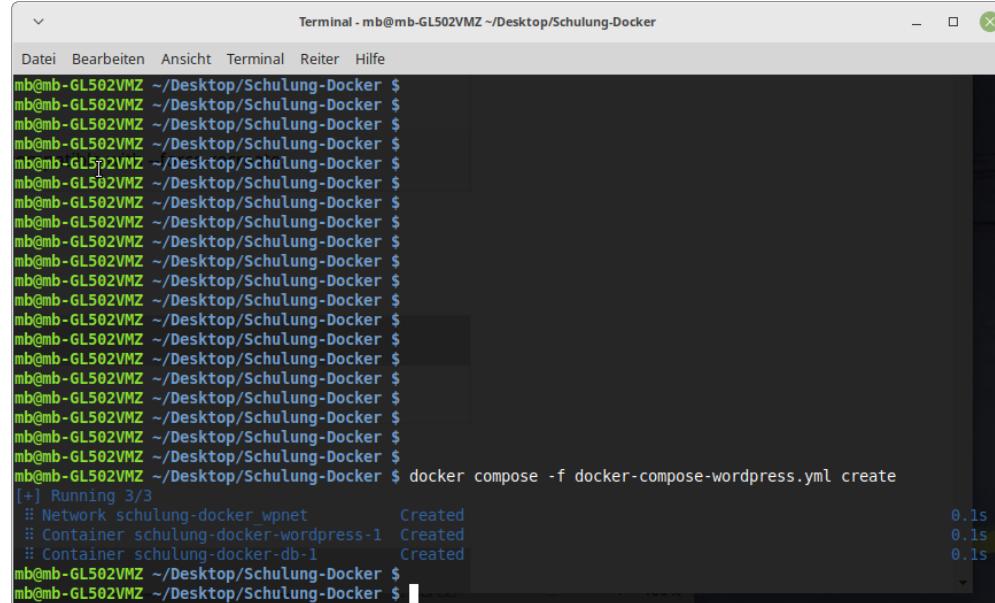
- Docker Objekte erzeugen

```
$ docker compose [OPTIONS] create [SERVICES]
```

```
$ docker compose -f docker-compose-wordpress.yml create  
$ docker compose -f docker-compose-wordpress.yml create db
```



Erzeugt die Docker Objekte der ganzen Projekte oder nur die am Ende aufgelisteten Services.



```
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $ docker compose -f docker-compose-wordpress.yml create  
[+] Running 3/3  
:: Network schulung-docker wpnet      Created  
:: Container schulung-docker-wordpress-1 Created  
:: Container schulung-docker-db-1     Created  
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $
```

# Docker Compose - CLI

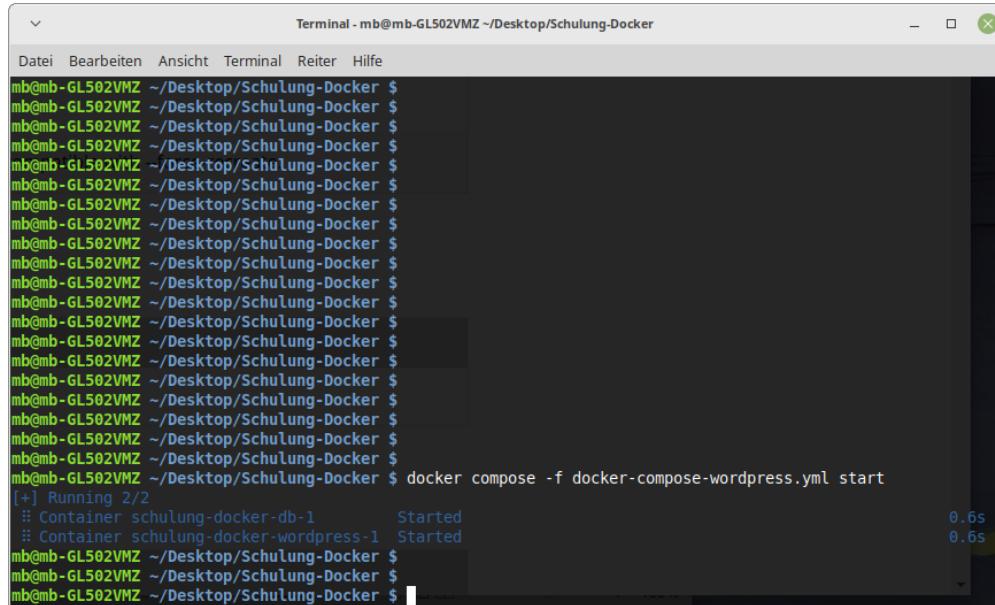
- Compose Projekt starten

```
$ docker compose [OPTIONS] start [SERVICES]
```

```
$ docker compose -f docker-compose-wordpress.yml start  
$ docker compose -f docker-compose-wordpress.yml start db
```



Startet die angegebenen Projekte oder nur die am Ende aufgelisteten Services.



```
Terminal - mb@mb-GL502VMZ ~/Desktop/Schulung-Docker
Datei Bearbeiten Ansicht Terminal Reiter Hilfe
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $ docker compose -f docker-compose-wordpress.yml start
[+] Running 2/2
  :: Container schulung-docker-db-1      Started          0.6s
  :: Container schulung-docker-wordpress-1 Started          0.6s
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $
```

# Docker Compose - CLI

- Compose Projekt stoppen

```
$ docker compose [OPTIONS] stop [SERVICES]
```

```
$ docker compose -f docker-compose-wordpress.yml stop  
$ docker compose -f docker-compose-wordpress.yml stop db
```

Stoppt die ganzen  
Projekte oder nur die am  
Ende aufgelisteten  
Services.



The terminal window shows the command `docker compose -f docker-compose-wordpress.yml stop` being run. The output lists all services in the compose file as stopped, with their previous status shown in brackets. The services listed are schulung-docker-wordpress-1 and schulung-docker-db-1. The terminal is titled "Terminal - mb@mb-GL502VMZ ~/Desktop/Schulung-Docker".

```
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $ docker compose -f docker-compose-wordpress.yml stop
[+] Stopped 2/2
:: Container schulung-docker-wordpress-1 Stopped
:: Container schulung-docker-db-1 Stopped
```

# Docker Compose - CLI

- Images im Projekt aktualisieren

```
$ docker compose [OPTIONS] pull [SERVICES]
```

```
$ docker compose -f docker-compose-wordpress.yml pull  
$ docker compose -f docker-compose-wordpress.yml pull db
```



Lädt für ganze Projekte  
oder nur die am Ende  
aufgelisteten Services  
die Images neu.

The screenshot shows a terminal window titled "Terminal - mb@mb-GL502VMZ ~/Desktop/Schulung-Docker". The terminal output is as follows:

```
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $ docker compose -f docker-compose-wordpress.yml pull  
[+] Running 2/2  
  :: wordpress Pulled  
  :: db Pulled  
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $
```

# Docker Compose - CLI

- Projekte erzeugen und starten

```
$ docker compose [OPTIONS] up [OPTIONS-2] [SERVICES]
```

```
$ docker compose -f docker-compose-wordpress.yml up -d  
$ docker compose -f docker-compose-wordpress.yml up db
```



Erzeugt und startet ganze Projekte oder nur die am Ende aufgelisteten Services.

Optionen-2:

--detach, -d ... starten die Projekte im Hintergrund (detached)

```
Terminal - mb@mb-GL502VMZ ~/Desktop/Schulung-Docker
Datei Bearbeiten Ansicht Terminal Reiter Hilfe
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $ docker compose -f docker-compose-wordpress.yml up -d
[+] Running 3/3
  ● Network schulung-docker-wpnet    Created
  ● Container schulung-docker-wordpress-l Started
  ● Container schulung-docker-db-1   Started
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $ 0.1s
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $ 0.8s
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $ 0.7s
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $ Matthias Boldt
```

# Docker Compose - CLI

- Projekte stoppen und entfernen

```
$ docker compose [OPTIONS] down [SERVICES]
```

```
$ docker compose -f docker-compose-wordpress.yml down  
$ docker compose -f docker-compose-wordpress.yml down db
```

Stoppt und löscht ganze Projekte oder nur die am Ende aufgelisteten Services.



# Docker Compose - CLI

- Logs der Projekte anzeigen

```
$ docker compose [OPTIONS] down [SERVICES]
```

```
$ docker compose -f docker-compose-wordpress.yml down  
$ docker compose -f docker-compose-wordpress.yml down db
```

Stoppt und löscht ganze Projekte oder nur die am Ende aufgelisteten Services.



```
Terminal - mb@mb-GL502VMZ ~/Desktop/Schulung-Docker
Datei Bearbeiten Ansicht Terminal Reiter Hilfe
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $ docker compose -f docker-compose-wordpress.yml down
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $ [+] Running 3/3
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $ :: Container schulung-docker-db-1 Removed
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $ :: Container schulung-docker-wordpress-1 Removed
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $ :: Network schulung-docker_wpsnet Removed
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $ mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $ mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $ mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $
```

# Docker Compose - CLI

- Logs der Projekte anzeigen

```
$ docker compose [OPTIONS] logs [OPTIONS-2]
```

```
$ docker compose -f docker-compose-wordpress.yml logs  
$ docker compose -f docker-compose-wordpress.yml logs -f
```



Zeigt die Logs aus den Projekten an.

-f ... fortlaufende Ausgabe der Log-Ausschriften

```
Terminal - mb@mb-GL502VMZ ~/Desktop/Schulung-Docker
Datei Bearbeiten Ansicht Terminal Reiter Hilfe
schulung-docker-db-1 | 2022-09-01 18:36:42 140070445070016 [Note] InnoDB: 128 out of 128 rollback segments are active.
schulung-docker-db-1 | 2022-09-01 18:36:42 140070445070016 [Note] InnoDB: Creating shared tablespace for temporary tables
schulung-docker-db-1 | 2022-09-01 18:36:42 140070445070016 [Note] InnoDB: Setting file './ibtmp1' size to 12 MB. Physically writing the file full; Please wait ...
schulung-docker-db-1 | 2022-09-01 18:36:42 140070445070016 [Note] InnoDB: File './ibtmp1' size is now 12 MB.
schulung-docker-db-1 | 2022-09-01 18:36:42 140070445070016 [Note] InnoDB: Waiting for purge to start
schulung-docker-db-1 | 2022-09-01 18:36:42 140070445070016 [Note] InnoDB: 5.7.38 started; log sequence number 2707264
schulung-docker-db-1 | 2022-09-01 18:36:42 140069344110336 [Note] InnoDB: Loading buffer pool(s) from /var/lib/mysql/ib_buffer_pool
schulung-docker-db-1 | 2022-09-01 18:36:42 140070445070016 [Note] Plugin 'FEEDBACK' is disabled.
schulung-docker-db-1 | 2022-09-01 18:36:42 140070445070016 [Note] Server socket created on IP: '::'.
schulung-docker-db-1 | 2022-09-01 18:36:42 140070445070016 [Note] Reading of all Master_info entries succeeded
schulung-docker-db-1 | 2022-09-01 18:36:42 140070445070016 [Note] Added new Master_info '' to hash table
schulung-docker-db-1 | 2022-09-01 18:36:42 140070445070016 [Note] mysqld: ready for connections.
schulung-docker-db-1 | Version: '10.2.44-MariaDB-1:10.2.44+maria-bionic' socket: '/var/run/mysqld/mysqld.sock' port: 3306 mariadb.org binary distribution
schulung-docker-db-1 | 2022-09-01 18:36:42 140069344110336 [Note] InnoDB: Buffer pool(s) load completed at 220901 18:36:42
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $
```

# Docker Compose

- Umgebungsvariablen zur detaillierten Steuerung

- .env Datei

```
PORTAINER_HOST_PORT_1 = 8000  
PORTAINER_HOST_PORT_2 = 9443  
DOCKER_SOCK = '/var/run/docker.sock'
```

- Variablenersetzung in der YAML-Steuerdatei

```
portainer:  
  image: portainer/portainer-ce:latest  
  ports:  
    - ${PORTAINER_HOST_PORT_1}:8000  
    - ${PORTAINER_HOST_PORT_2}:9443  
  volumes:  
    - ${DOCKER_SOCK}:/var/run/docker.sock  
    - portainer-data:/data
```

- .env Datei automatisch aus Projektverzeichnis
  - Option --env-file zur Nutzung alternativer Dateien
  - Option -e zum Setzen einzelner Variablen, z.B. -e DEBUG=1

# Docker Compose - Environment

- Beispiel mit Umgebungsvariablen

- .env Datei ... .env

```
WORDPRESS_PORT = 8001  
WORDPRESS_DB = wordpress
```

- YAML-Steuerdatei ... docker-compose-wordpress-mit-env.yml

```
wordpress:  
  image: wordpress:latest  
  ports:  
    - ${WORDPRESS_PORT}:80  
  restart: always  
  environment:  
    - WORDPRESS_DB_PASSWORD=wordpress  
    - WORDPRESS_DB_NAME=${WORDPRESS_DB}
```

```
$ docker compose -f docker-compose-wordpress-mit-env.yml up -d
```



# Docker Compose - Yaml

- YAML-Dateien
  - YAML Ain't Markup Language™
  - Strukturierte Textdatei für Datenstrukturen
  - <https://yaml.org/spec/1.2.2/>
  - compose.yml oder docker-compose.yml

```
services:  
  mongodb:  
    image: mongo:4.2-bionic  
    restart: unless-stopped  
    ports:  
      - ${MONGODB_HOST_PORT}:27017  
    volumes:  
      - mongo_data:/db/data  
    networks:  
      - default
```

# Docker Compose - Yaml

- Wird auch genutzt bei ... Kubernetes, Swagger/OpenAPI, ...
- YAML Syntax
  - Strukturelemente werden durch Einrückung definiert
  - Wesentliche Elemente:
    - Strukturen
    - Skalare
    - Listen
    - Arrays

# Docker Compose - Yaml

- YAML-Strukturen
  - Nach Einrückung folgen die Elemente
  - Elemente können Skalare, Strukturen, Listen, Arrays sein
  - Die Elemente von Strukturen bilden eine (assoziative) Liste, wobei jedes mit <name>: <value> angegeben wird

```
services:  
  mongodb:  
    image: mongo:4.2-bionic  
    restart: unless-stopped  
    admin:  
      name: admin  
      password: asdf0815
```

# Docker Compose - Yaml

- YAML-Skalare
  - Einzelwerte (Zeichkette, Zahl)
  - <value> entspricht dem Einzelwert

```
services:  
  mongodb:  
    image: mongo:4.2-bionic  
    restart: unless-stopped  
    count: 3  
    admin:  
      name: admin  
      password: asdf0815
```

# Docker Compose - Yaml

- YAML-Arrays
  - Ungeordnete Menge von Elementen
  - Werden mit einem „-“ als Präfix definiert
  - <value> können wieder Strukturen, Skalare, ... sein

```
- mongo:4.2-bionic
- unless-stopped
- 3
- admin:
  name: admin
  password: asdf0815
```

# Docker Compose - Yaml

- YAML-Listen
  - Menge von Elementen mit <name>: <value>
  - <value> können wieder Strukturen, Skalare, ... sein

```
image: mongo:4.2-bionic
restart: unless-stopped
Ports: 3
Volumes: 5
networks:
  - default
  - mongo-net
  - logging
```

# Docker Compose - Yaml

- Vorstellung wesentlicher Statements
- Vollständige Dokumentation und Spezifikation
  - <https://docs.docker.com/compose/compose-file/>
  - <https://github.com/compose-spec/compose-spec/blob/master/spec.md>
- Wichtige Bereiche im YAML-File
  - services ... muss verpflichtend enthalten sein
  - networks, volumes, configs, secrets ... sind optional

# Docker Compose - Yaml

- Bereich `services`
  - Muss im YAML-File enthalten sein
  - Ein Service definiert Container mittels einer Images und einer Menge von Parametern
  - Alle Container eines Services werden gleichartig erzeugt
  - Kann optional eine „build“-Sektion enthalten

# Docker Compose - Yaml

- Bereich `services` wichtigste Parameter je Service:
  - `image`
  - `restart`
  - `ports`
  - `volumes`
  - `networks`
  - `environment`
  - `command`
  - `deploy`
  - `depends_on`

# Docker Compose - Yaml

- Bereich `services` – Parameter `image`:
  - Definition des Docker Image als Grundlage für den Bau der Container
  - Format: [<registry>/][<project>/]<image>[:<tag>]

Services:

```
sm-flowcontroller:  
  image: ${SYS_REGISTRY}/sm-flowcontroller:${SYS_VERSION}  
  
mongodb:  
  image: mongo:4.2-bionic  
  
caddy:  
  image: abiosoft/caddy:latest
```

# Docker Compose - Yaml

- Bereich `services` – Parameter `restart`:
  - Definiert das Neustart-Verhalten bei Termination
  - Optionen:
    - `no` ... niemals
    - `always` ... immer, bis zum Löschen
    - `unless-stopped` ... immer, bis zum Stop oder Löschen
    - `on-failure` ... bei einem Abbruch auf Grundlage eines Fehlers

`services:`

```
caddy:  
  image: "abiosoft/caddy:latest"  
  restart: unless-stopped
```

# Docker Compose - Yaml

- Bereich `services` – Parameter `ports`:
  - Freigabe von Ports des Containers
  - Format: [HOST:]CONTAINER[/PROTOCOL]
    - HOST ... Port des Hosts ... [IP:](port | range)
    - CONTAINER ... Port des Containers ... port | range
    - PROTOCOL ... „tcp“ oder „udp“

```
services:  
  portainer:  
    image: portainer/portainer-ce:latest  
    ports:  
      - ${PORTAINER_HOST_PORT_1}:8000  
      - ${PORTAINER_HOST_PORT_2}:9443
```

# Docker Compose - Yaml

- Bereich `services` – Parameter `volumes`:
  - Einbindung von Volumes in den Containern
  - Format: `VOLUME:CONTAINER_PATH[:ACCESS_MODE]`
    - `VOLUME` ... Pfad auf dem Host oder Name des Volumes
    - `CONTAINER_PATH` ... Mount-Pfad im Container
    - `ACCESS_MODE` ... „rw“ oder „ro“

```
services:
```

```
  portainer:  
    image: portainer/portainer-ce:latest  
    volumes:  
      - ${DOCKER_SOCK}:/var/run/docker.sock  
      - portainer-data:/data
```

# Docker Compose - Yaml

- Bereich `services` – Parameter `networks`:
  - Netzwerke, zu denen die Container verbunden sind
  - Auflistung der Netzwerke

```
services:  
  redis:  
    image: redis  
    networks:  
      - default  
      - wpnet
```

# Docker Compose - Yaml

- Bereich `services` – Parameter `environment`:
  - Umgebungsvariablen, die im Container gesetzt werden
  - Auflistung der Umgebungsvariablen

```
services:  
  
  minio:  
    image: minio/minio  
    restart: unless-stopped  
    volumes:  
      - ./minio:/data  
    environment:  
      - MINIO_ACCESS_KEY=supadmin  
      - MINIO_SECRET_KEY=password01  
      - MINIO_ROOT_USER=supadmin  
      - MINIO_ROOT_PASSWORD=password01
```

# Docker Compose - Yaml

- Bereich `services` – Parameter `command`:
  - Überschreibt den CMD-Parameter des Containers

```
services:  
  
  minio:  
    image: minio/minio  
    restart: unless-stopped  
    volumes:  
      - ./minio:/data  
    command: server /data --console-address ":9001"
```

# Docker Compose - Yaml

- Bereich `services` – Parameter `deploy`:
  - Laufzeitanforderungen eines Containers def.
  - Wesentliche Sub-Parameter:
    - mode
      - global ... ein Container pro Docker-Node
      - replicated ... eine festzulegende Anzahl an Containern (default)
    - replicas ... Anzahl der Container im replicated-Mode
    - resources ... Ressources je Container

# Docker Compose - Yaml

- Bereich `services` – Parameter `deploy:`
  - `resources`
    - `limits` ... obere Grenzen
    - `reservations` ... Mindestgrößen
  - Werte der Resource-Definitionen
    - `cpus` ... Anzahl der CPU's [0.0 ... n] (z.B. 0.5)
    - `memory` ... Speicher in Bytes (z.B. 512k, 20M, 2G)

# Docker Compose - Yaml

- Bereich services – Parameter deploy:

```
services:  
  sm-profilecontroller:  
    image: ${SYS_REGISTRY}/sm-profilecontroller:${SYS_VERSION}  
    restart: unless-stopped  
    volumes:  
      - ${TMP_DIRECTORY}:/tmp  
      - ${CONF_DIRECTORY}:/config  
    ports:  
      - ${PROFILECONTROLLER_HOST_PORT}:4242  
    Deploy:  
      mode: replicated  
      replicas: 5  
      resources:  
        limits:  
          cpus: "0.2"  
          memory: 512m  
        reservations:  
          cpus: "0.1"  
          memory: 50m
```

# Docker Compose - Yaml

- Bereich `services` – Parameter `depends_on`:
  - Liste von anderen Services, auf die bei Start und Shutdown gewartet wird

Services:

```
ai-preprocessing:  
  image: ${SYS_REGISTRY}/ai-preprocessing:${SYS_VERSION}  
  restart: unless-stopped  
  deploy:  
    replicas: 1  
    resources:  
      limits:  
        cpus: "0.5"  
        memory: 4g  
      reservations:  
        cpus: "0.1"  
        memory: 500m  
  depends_on:  
    - "redis"  
    - "queue-redis"  
    - "mongodb"
```

# Docker Compose - Yaml

- Bereich networks
  - Definition von Networks
  - Wichtigste Parameter
    - driver ... die vorhandenen Docker-Driver (host, bridge, ...)
    - attachable ... bei „true“ können standalone-Container (neben den Services der Compose-Projekte) hinzugefügt werden

```
networks:
```

```
  wpnet:  
    driver: bridge  
    attachable: true
```

# Docker Compose - Yaml

- Bereich `volumes`
  - Definition von Volumes, die in den Services von Containern genutzt werden können
  - Wichtigster Parameter: `external ...` bei „true“ wird das Volume nicht von Compose verwaltet, sondern als existent vorausgesetzt

```
services:  
  db:  
    image: mariadb:10.2  
    volumes:  
      - db_data:/var/lib/mysql  
  
volumes:  
  db_data:  
    external: true
```

# Docker Compose - Yaml

- Profile in Compose-Projekten
- Parameter `profiles` als Liste im Service
- Start einer Auswahl von Services

```
services:  
  db:  
    image: mariadb:10.2  
    volumes:  
      - db_data:/var/lib/mysql  
  profiles:  
    - database  
    - all
```

# Docker Compose - Yaml

- Auswahl des Profiles über den --profile Parameter des docker compose Kommandos
- Beispiel in docker-compose-wordpress-mit-profile.yml

```
$ docker compose -f docker-compose-wordpress-mit-profile.yml --profile database up -d
```



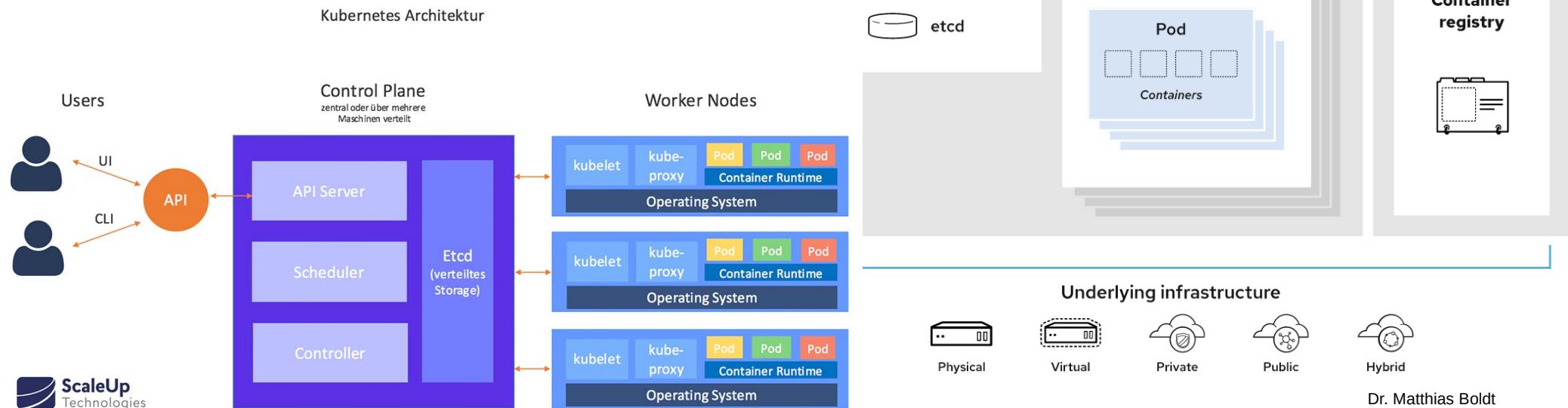
```
Terminal - mb@mb-GL502VMZ ~/Desktop/Schulung-Docker
Datei Bearbeiten Ansicht Terminal Reiter Hilfe
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $ mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $ docker compose -f docker-compose-wordpress-mit-profile.yml --profile database up -d
[+] Running 2/2
  ● Network schulung-docker-wpnet Created
  ● Container schulung-docker-db-1 Started
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $ docker container list
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
c3dacec13284 mariadb:10.2 "docker-entrypoint.s..." 10 seconds ago Up 9 seconds 330
6/tcp schulung-docker-db-1
f3df9e99be57 portainer/portainer-ce:latest "/portainer --admin..." 3 months ago Up 5 hours 0.0
.0.0:8000->8000/tcp, 0.0.0.0:9443->9443/tcp, 9000/tcp senseaition-on-premis_portainer_1
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $
```

# Kubernetes = k8s

- <https://kubernetes.io/>
- System zur Orchestrierung von Linux-Containern
- Betrieb und Skalierung über verschiedene Nodes
- Windows-Container auf spez. Windows-Nodes
- Schnelles Skalieren
- Infrastructure-as-a-Service
- Automatisierung durch Scripts
- „Selbstheilungskräfte“  
=> „erweitertes Docker System“

# Kubernetes - Architektur

- Master-Node
- Worker-Node(s)
- Nutzt u.a. containerd
- In go/golang implementiert



# Kubernetes – Master-Node (Linux)

- Kube-API-Server
  - Zentrales REST-API für Steuerung/Konfiguration
  - Nutzung z.B. über kubectl, Kubernetes-Dashboard ... oder eigene Programme
- Etcd
  - Datenbank-Backend
  - Verteilter Key-Value-Storage für Konfigurationen, Zuständen, etc.
- Kube-Scheduler
  - Zuweisung von Cluster-Ressourcen zu Pods
- Kube-Controller-Manager
  - Allgemeine Ablaufsteuerung des K8s-Clusters
  - Überwachung des Cluster-Zustandes

# Kubernetes – Worker-Nodes

- Kubelet
  - Daemon der Pods startet und überwacht
- Kube-Proxy
  - Netzwerkverbindung steuern
- Container Runtime
  - Meist containerd (von Docker) ... auch cri-o, rktlet, ...

# Kubernetes – Objekte/Terminologie

- Basisobjekte
  - Pods
  - Services
  - Volumes
  - Namespaces
- Zusätzliche Objekte
  - ReplicaSets
  - Deployments
  - StatefulSets
  - DaemonSets
  - Jobs

# Kubernetes – Pods

- Beherbergen die aufenden Container
- Können prinzipiell mehrere Container enthalten
- Alle Container eines Pods teilen gleiche Ressourcen
- Pods sind die Grundlage der Replikation/Skalierung
- Alle Pods eines ReplicaSets haben gleichen DNS-Namen und eigene IP-Nummern => K8s kann intern automatisch loadbalancing
- Replicas von Pods werden z.B. durch Deployments organisiert
- Pods sind von der äußeren Welt isoliert => loadbalancer / ingress-Controller regelt den Zugriff von außen

# Kubernetes – Pods

- Lebenszyklus & Status eines Pods
  - *Pending* ... Pod-Definition ist akzeptiert und K8s bereitet den Start vor
  - *Waiting* ... z.B. Image wird aus Registry geladen
  - *Running* ... alle Container im Pod laufen
  - *Succeeded* ... Container im Pod wurden beendet und es gibt keinen Neustart
  - *Failed* ... alle Container wurden unterbrochen, es gab Fehler
  - *Terminated* ... Failed oder Succeeded
  - *Unknown* ... interner Kommunikationsfehler mit Pod

# Kubernetes – Pods

- K8s besitzt „Selbstheilungskräfte“
  - RestartPolicy: Always, OnFailure, Never
  - Probes für den Health-Check ... REST-Calls werden regelmäßig von **kubelet** ausgeführt ... sollten im Container implementiert sein
    - LivenessProbe ... Container läuft
    - ReadinessProbe ... Container kann Request beantworten
    - StartupProbe ... App im Container läuft/alle Ressourcen geladen
  - Automatischer Austausch von „kranken“ Pods

# Kubernetes – Services

- Anwendung aus mehreren Pods wird als Service im Netzwerk angeboten / freigegeben
- Zu einem Service gehörige Pods werden über Selektoren gewählt
- Definieren einen Microservice / ein REST-Interface für andere Teile der App

# Kubernetes – Volumes

- Container & Pods sind stateless
- Persistente Daten werden in Volumes gespeichert
- Verzeichnis, das in den Container gebunden (mounted) wird
- Viele verschiedene Typen (abhängig von K8s-Installation)
  - awsElasticBlockStore
  - hostPath
  - local
  - nfs
  - **persistentVolumeClaim**
  - ...

# Kubernetes – Volumes

- persistentVolumes
  - Nutzung von Volumes möglich, unabhängig von der konkreten Implementation
  - In der K8s-Installation ist festgelegt, welche Storage-Typen verfügbar sind
  - Lebenszyklus
    - Provisioning
    - Binding
    - Using

# Kubernetes – Namespaces

- Isolierung und Gruppierung von Objekten
- Namespace ist Teil des DNS-Namen eines Services

`<service-name>. <namespace-name>. svc.cluster.local`

- Einige Objekte sind clusterweit verfügbar und nicht „namespaced“ ... Nodes, PersistentVolumes, ...

# Kubernetes – ReplicaSets

- Garantiert die Verfügbarkeit einer festen Anzahl von Pod eines Typs
- Deployments tun das auch ... ReplicaSets müssen nicht explizit angelegt werden

# Kubernetes – Deployments

- Verwaltet *Pods* und *ReplicaSets*
- Pods und ReplicaSets sollten nicht separat angelegt/verwaltet werden

# Kubernetes – Jobs/CronJobs

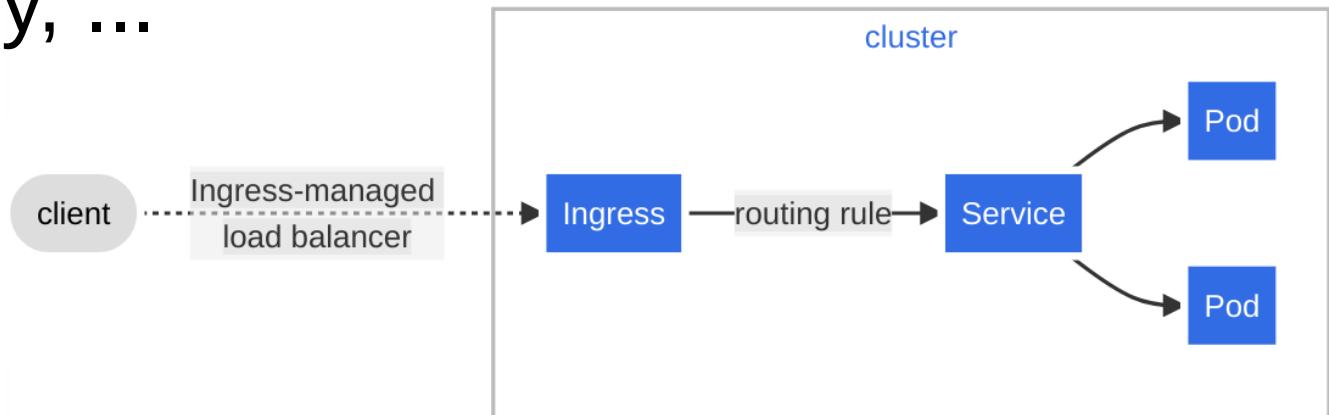
- Job
  - Erzeugt spezifizierte Pods so lange, bis ein erfolgreicher Abschluss erreicht ist
  - Einer oder mehrere Pods und eine festgelegte Anzahl von erfolgreichen Abschlüssen
- CronJob
  - Erzeugt Jobs zu festgelegten Zeitpunkten
  - Funktioniert wie *crontab*

# Kubernetes – Netzwerke

- Pods
  - Bekommen eindeutige IP's im Cluster
  - Können mit jedem anderen Pod auf jedem Node im Cluster kommunizieren
  - Bekommen eindeutige DNS-Namen im Cluster
- Services
  - Können Pods von außen erreichbar machen
  - Können auch nur von innerhalb des Clusters erreichbar gemacht werden
  - Bekommen eindeutige DNS-Namen im Cluster

# Kubernetes – Netzwerke

- Ingress-Controller (optional)
  - Verbindet Außenwelt mit Services im Cluster
  - WEB-Server für REST-API
  - nginx, caddy, ...



# Kubernetes – Installation

- Entwicklung
  - Docker-Desktop ... von Docker (bereits ausgeführt :-)
  - Rancher-Desktop ... von Suse
  - minikube ... <https://minikube.sigs.k8s.io/docs/> ... ein Node
  - kind ... <https://kind.sigs.k8s.io/> ... mehrere Nodes
  - MicroK8s ... von Ubuntu
- Produktion
  - Kubernetes, kubeadm / kops / kubespray
  - Rancher, RKE, K3s ... von Suse
  - OpenShift ... von RedHat
  - Managed-K8s von Cloud-Service-Providern ... AWS, GCP, Azure, DigitalOcean

# Kubernetes – managed vs. own

- Managed
  - + Einfacher Einstieg, da K8s bereits installiert und konfiguriert ist
  - + Keine feste Bindung von „bare metal“ Ressourcen
  - lock-in Effekte durch Speziallösungen
  - oft komplizierte Management-UI's
- Eigene Installation
  - + Mit Systemen wie Rancher ist der „overhead“ gering
  - + Kein lock-in-Effekt
  - „bare metal“ Ressourcen sind fest gebunden

# Kubernetes – managed vs. own

- Nutzung beider Welten
  - Plain K8s nutzen (keine Speziallösungen)
  - Einfacher Start mit einer managed-K8s-Lösung
  - Wechsel zwischen Cloud-Anbietern und/oder zu eigener Lösung jederzeit möglich

# Rancher

- <https://rancher.com/>
- Management aller eigenen K8s-Cluster
  - Ein gemeinsames Tool für alle Cluster
  - Ausführliches Super-Dashboard
  - Integrierter Marketplace für einfache Installation zusätzlicher Tools
  - Lokal & remote
  - Managed K8s & „bare metal“
  - 100% open source & kein „lock-in“
  - „Treiber“ für viele Cloud-Provider

# Rancher



## Shared Tools & Services



**LONGHORN**



**Prometheus**



**Istio**



**Open Policy Agent**

 **Grafana**

 **fluentd**

## Security & Authentication



**Active Directory**

**GitHub**

**SAML**

 **Ping Identity**

 **okta**

## Simplified Cluster Operations & Infrastructure Management

Kubernetes Version Management

Visibility & Diagnostics

Monitoring & Alerting

Centralized Audit

Node Pool Management

Cluster Provisioning

Container Storage

Continuous Deployment



**RKE**



Datacenter



Amazon EKS



Azure AKS



Google GKE



**K3S**



Cloud



Dev



Branch



5G / Edge

Docker & Kubernetes

# Rancher – Cluster Dashboard

The screenshot shows the Rancher Cluster Dashboard for the 'sense-prod' cluster. The dashboard provides an overview of cluster resources, capacity usage, and recent events.

**Cluster Overview:**

- Provider: RKE1
- Kubernetes Version: v1.21.6
- Created: 300 days ago
- Install Monitoring button

**Total Resources:** 197

**Nodes:** 7

**Deployments:** 61

**Capacity Metrics:**

- Pods:** Used 212 / 440 (48.18%)
- Cores:** Reserved 2.22 / 32 (6.94%)
- Memory:** Reserved 0.76 / 61 GiB (1.25%)

**Events:**

Reason	Resource	Date
Completed	Job sm-billingcontroller-job-receipt-27716753	24 secs ago
Created	Pod sm-billingcontroller-job-receipt-27716753-l55tr	25 secs ago
Pulled	Pod sm-billingcontroller-job-receipt-27716753-l55tr	25 secs ago
Started	Pod sm-billingcontroller-job-receipt-27716753-l55tr	25 secs ago

**Cluster Tools:** v2.6.2

# Rancher – Nodes Liste

The screenshot shows the Rancher interface for the 'sense-prod' cluster. The left sidebar contains navigation links for Cluster, Workload, Apps & Marketplace, Service Discovery, Storage, RBAC, and More Resources. Under Cluster, 'Nodes' is selected, showing 7 nodes. The main area displays a table of nodes with columns: State, Name, Roles, Version, External/Internal IP, OS, CPU, RAM, Pods, and Age. The table lists seven nodes: fsn-worker-1, fsn-worker-4, fsn-worker-6, fsn-worker-7, master-1, master-2, and master-3. Master nodes have the 'Control Plane, Etcd' role, while worker nodes have the 'Worker' role. The table includes a 'Taints' section for master nodes. A 'Cluster Tools' button is at the bottom left, and the version 'v2.6.2' is at the bottom center.

State	Name	Roles	Version	External/Internal IP	OS	CPU	RAM	Pods	Age	
Active	fsn-worker-1	Worker	v1.21.6	192.168.1.28	None	Linux	7.4%	44%	86%	180 days
Active	fsn-worker-4	Worker	v1.21.6	192.168.1.33	None	Linux	12%	49%	51%	97 days
Active	fsn-worker-6	Worker	v1.21.6	192.168.1.14	None	Linux	25%	90%	49%	12 days
Active	fsn-worker-7	Worker	v1.21.6	192.168.1.12	None	Linux	2.4%	13%	6.4%	10 days
Active	master-1	Control Plane, Etcd	v1.21.6	192.168.1.57	None	Linux	16%	35%	4.5%	300 days
		Taints:	node-role.kubernetes.io/controlplane=true:NoSchedule node-role.kubernetes.io/etcd=true:NoExecute							
Active	master-2	Control Plane, Etcd	v1.21.6	192.168.1.53	None	Linux	20%	41%	5.5%	300 days
		Taints:	node-role.kubernetes.io/controlplane=true:NoSchedule node-role.kubernetes.io/etcd=true:NoExecute							
Active	master-3	Control Plane, Etcd	v1.21.6	192.168.1.57	None	Linux	23%	49%	6.4%	300 days
		Taints:	node-role.kubernetes.io/controlplane=true:NoSchedule node-role.kubernetes.io/etcd=true:NoExecute							

# Rancher – Deployments Liste

The screenshot shows the Rancher web interface for managing Kubernetes resources. The left sidebar navigation includes 'Cluster' (selected), 'Workload' (CronJobs: 8, DaemonSets: 8, Deployments: 61, Jobs: 32, StatefulSets: 6, Pods: 261), 'Apps & Marketplace', 'Service Discovery', 'Storage', 'RBAC', and 'More Resources'. The top bar shows the cluster name 'sense-prod' and a dropdown for 'Only User Namespaces'. The main area is titled 'Deployments' with a star icon. It features buttons for 'Redeploy', 'Download YAML', and 'Delete'. A 'Create' button is in the top right. The table lists deployments across three namespaces: 'cattle-fleet-system', 'cattle-system', and 'kube-system'. Each row includes columns for State (Active), Name, Image, Endpoints, Ready, Up-to-date, Available, and Age.

Namespace	Name	Image	Endpoints	Ready	Up-to-date	Available	Age
cattle-fleet-system	fleet-agent	rancher/fleet-agent:v0.3.7	1/1	1	1	300 days	...
cattle-system	cattle-cluster-agent	rancher/rancher-agent:v2.6.2	2/2	2	2	300 days	...
kube-system	calico-kube-controllers	rancher/mirrored-calico-kube-controllers:v3.19.2	1/1	1	1	300 days	...
	coredns	rancher/mirrored-coredns:coredns:1.8.4	2/2	2	2	300 days	...
	coredns-autoscaler	rancher/mirrored-cluster-proportional-autoscaler:1.8.3	1/1	1	1	300 days	...
	hcloud-cloud-controller-manager	hetznercloud/hcloud-cloud-controller-manager:v1.9.1	1/1	1	1	300 days	...
	hcloud-csi-controller	k8s.gcr.io/sig-storage/csi-attacher:v3.2.1 + 4 more	1/1	1	1	180 days	...
	kube-state-metrics	k8s.gcr.io/kube-state-metrics/kube-state-metrics:v2.2.4	1/1	1	1	300 days	...

# Rancher – Marketplace

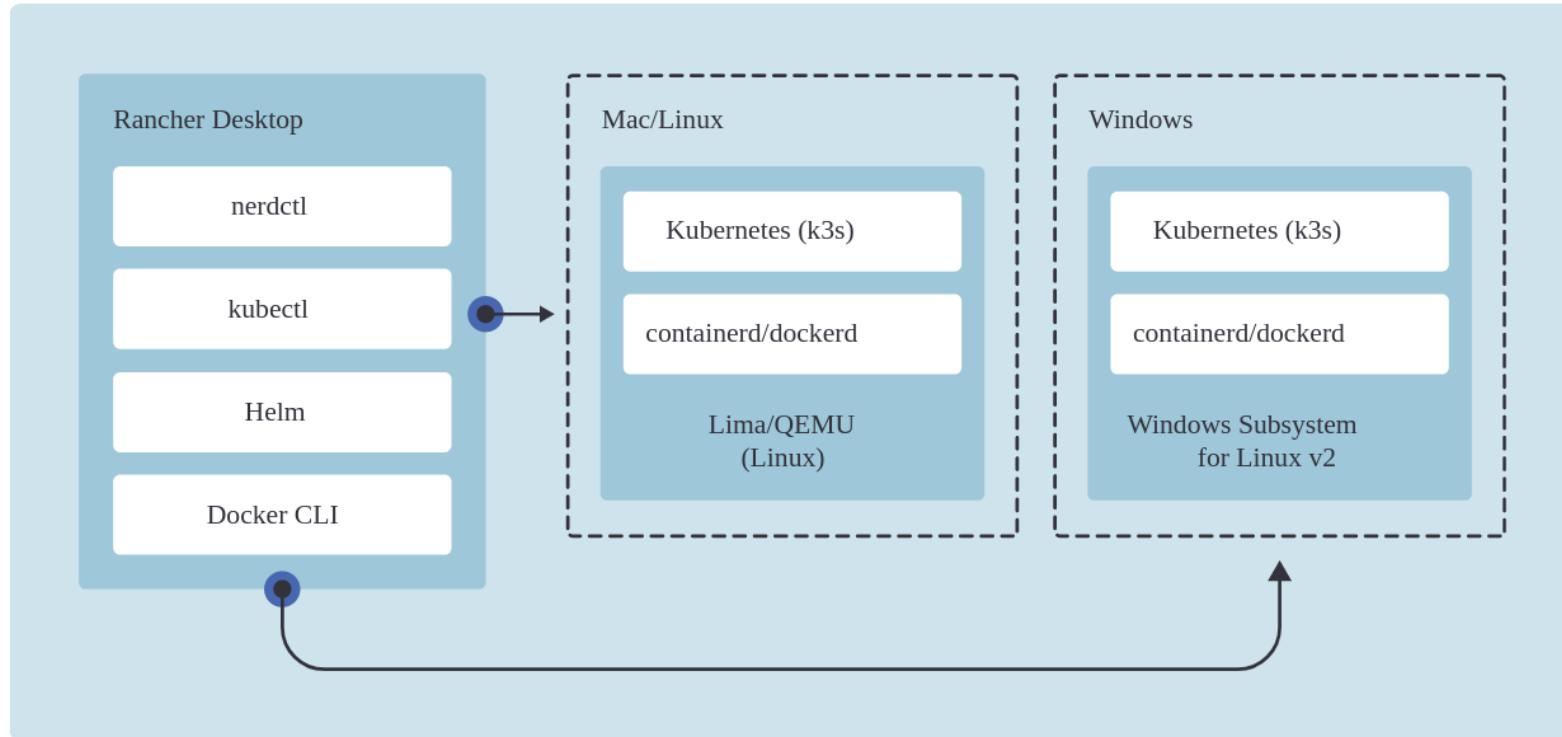
The screenshot shows the Rancher Marketplace interface. At the top, there's a navigation bar with a sidebar menu on the left containing items like Cluster, Workload, Apps & Marketplace (with Charts selected), Service Discovery, Storage, RBAC, and More Resources. The main area is titled "Charts" and displays a grid of 20 different charts, each with a title, icon, and a brief description.

Category	Chart Name	Description
All	Alerting Drivers	The manager for third-party webhook receivers used in Prometheus Alertmanager
All	CIS Benchmark	The cis-operator enables running CIS benchmark security scans on a...
All	External IP Webhook	Deploy the external-ip-webhook to mitigate k8s CVE-2020-8554
All	Harvester Cloud Provider	A Helm chart for Harvester Cloud Provider
All	Harvester CSI Driver	A Helm chart for Harvester CSI driver
All	Istio	A basic Istio setup that installs with the istioctl. Refer to...
All	Logging	Collects and filter logs using highly configurable CRDs. Powered by Banzai Cloud...
All	Longhorn	Longhorn is a distributed block storage system for Kubernetes.
All	Monitoring	Collects several related Helm charts, Grafana dashboards, and...
All	OPA Gatekeeper	Modifies Open Policy Agent's upstream gatekeeper chart that...
Experimental	sriov	SR-IOV network operator configures and manages SR-IOV networks in the...
All	vSphere CPI	vSphere Cloud Provider Interface (CPI)
All	vSphere CSI	vSphere Cloud Storage Interface (CSI)
All	Ambassador Edge Stack	A Helm chart for Datawire Ambassador
All	artifactory-ha	Universal Repository Manager supporting all major packaging formats...
All	artifactory-jcr	JFrog Container Registry
Citrix	Citrix Adc Istio Ingress ...	A Helm chart for Citrix ADC as Ingress Gateway installation in Istio Service...
Citrix	Citrix Cpx Istio Sidecar I...	A Helm chart to deploy resources which install Citrix ADC CPX in Istio...
Citrix	Citrix Ingress Controller	A Helm chart for Citrix Ingress Controller configuring MPX/VPX...
Citrix	citrix-cpx-with-ingress-c...	A Helm chart for Citrix ADC CPX with Citrix ingress Controller running as...

At the bottom of the interface, there are buttons for "Cluster Tools" and "v2.6.2", and a footer with the text "Docker & Kubernetes" and "Dr. Matthias Boldt".

# Rancher-Desktop

- Hat nur den Namen mit Rancher gemeinsam
- <https://rancherdesktop.io/>



# Rancher-Desktop

Rancher Desktop

File Edit View Help

RANCHER DESKTOP

- General
- Port Forwarding
- Images
- Troubleshooting

## Welcome to Rancher Desktop

Rancher Desktop provides Kubernetes and image management through the use of a desktop application.

- Project Discussions: #rancher-desktop in [Rancher Users](#) Slack
- Project Links: [Homepage](#) [Issues](#)

**Version: 1.5.1**

Allow collection of anonymous statistics to help us improve Rancher Desktop

Send anonymized usage info, error reports, etc. to help improve Rancher Desktop. Your data will not be shared with anyone else, and no information about what specific resources or endpoints you are deploying is included.

Rancher Desktop - Preferences

Preferences	
Application	Memory (GB)
Virtual Machine	8 <input type="range" value="8"/>
Container Runtime	# CPUs
Kubernetes	6 <input type="range" value="6"/>

Cancel  Apply

Rancher Desktop - Preferences

Preferences	
Application	Kubernetes
Virtual Machine	<input checked="" type="checkbox"/> Enable Kubernetes
Container Runtime	Kubernetes version
Kubernetes	v1.24.3 <input type="button"/>
	Kubernetes Port
	6443 <input type="button"/>
	Traefik
	<input checked="" type="checkbox"/> Enable Traefik

Cancel  Apply

Dr. Matthias Boldt

# Kubernetes - kubectl

- <https://kubernetes.io/docs/reference/kubectl/>
- <https://kubernetes.io/docs/reference/kubectl/cheatsheet/>
- DAS Tool zur Verwaltung von K8s-Clustern
- Greift auf den Kube-Apiserver zu
- Kommandozeilenbasiert
- Nutzt eine Konfigurationsdatei
  - Je K8s-Cluster unterschiedliche Konfiguration `kubectl config get-contexts`
  - Umschaltung des Kontexts mit Kommando `kubectl config use-context <name>`
  - `~/.kube/config`

# Kubernetes - kubectl

- kubectl installieren
  - Windows ... binary laden

```
> curl -LO "https://dl.k8s.io/release/v1.25.0/bin/windows/amd64/kubectl.exe"
```

- Linux (und in der WSL unter Windows)

```
$ curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"  
$ sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl  
$ kubectl version --short  
$ kubectl cluster-info
```



# Kubernetes - kubectl

- kubectl konfigurieren

```
$ kubectl config get-contexts  
$ kubectl config use-context docker-desktop  
$ kubectl config use-context rancher-desktop  
$ kubectl get nodes  
$ kubectl get pods --all-namespaces
```



Terminal - mb@mb-GL502VMZ ~

```
mb@mb-GL502VMZ ~ $ kubectl config get-contexts  
mb@mb-GL502VMZ ~ $  
mb@mb-GL502VMZ ~ $ Using 28 items  
mb@mb-GL502VMZ ~ $  
mb@mb-GL502VMZ ~ $ kubectl config get-contexts  
CURRENT NAME CLUSTER AUTHINFO NAMESPACE  
* docker-desktop docker-desktop docker-desktop alpine:latest  
rancher-desktop rancher-desktop rancher-desktop  
mb@mb-GL502VMZ ~ $ kubectl config use-context docker-desktop  
Switched to context "docker-desktop".  
mb@mb-GL502VMZ ~ $ kubectl get nodes 288bb96  
NAME STATUS ROLES AGE VERSION  
docker-desktop Ready control-plane 51m v1.25.0  
mb@mb-GL502VMZ ~ $  
mb@mb-GL502VMZ ~ $ 6aebe2e59488  
klausmeyer/docker-registry-b  
mb@mb-GL502VMZ ~ $  
mb@mb-GL502VMZ ~ $  
mb@mb-GL502VMZ ~ $ registry  
mb@mb-GL502VMZ ~ $ registry:2
```

# Kubernetes - kubectl

kubectl [command] [TYPE] [NAME] [flags]

- Wesentliche Kommandos
  - config ... kubectl Konfiguration anpassen
  - apply ... eine Konfigurationsänderung anwenden/Ressourcen erzeugen
  - create ... eine Ressource erzeugen
  - get ... Ressourcen auflisten
  - describe ... Details zum Status von Ressourcen zeigen
  - explain ... Dokumentation zu Ressourcen zeigen
  - delete ... Ressourcen entfernen
  - logs ... Logs eines Containers im Pod anzeigen
  - run ... ein einzelnes Image im Cluster starten
  - exec ... einen Befehl in einem Container ausführen
  - edit ... eine Ressource interaktiv in VI anpassen

# Kubernetes - kubectl

- Organisation vieler Ressourcen in Namespaces
- Wesentliche Typen (Ressourcen-Typen und Abkürzungen)
  - nodes ... no
  - pods ... po
  - namespaces ... ns
  - deployments ... deploy
  - ingresses ... ing
  - services ... svc
  - persistentvolumeclaims ... pvc
  - persistentvolumes ... pv

# Kubernetes - kubectl

- Wesentliche Flags
  - `--namespace <value>`
  - `--kubeconfig <file>`
  - `-f <file>`
  - `-o json ...` Ausgabe als Json-Struktur
  - `-o yaml ...` Ausgabe als Yaml-Struktur
  - `-o wide ...` Ausgabe als Text
  - `--all-namespaces`

# Kubernetes - kubectl

- nginx mit kubectl erzeugen und starten

```
$ kubectl apply -f k8s-nginx.yml  
$ kubectl get deploy  
$ kubectl get po
```



Terminal - mb@mb-GL502VMZ ~/Desktop/Schulung-Docker

```
Datei Bearbeiten Ansicht Terminal Reiter Hilfe  
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $ kubectl apply -f k8s-nginx.yml  
deployment.apps/nginx-deployment created  
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $ kubectl get deploy  
NAME READY UP-TO-DATE AVAILABLE AGE  
nginx-deployment 2/2 2 2 39s  
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $ kubectl get po  
NAME READY STATUS RESTARTS AGE  
nginx-deployment-544dc8b7c4-kng75 1/1 Running 0 68s  
nginx-deployment-544dc8b7c4-qnrnb 1/1 Running 0 68s  
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $ kubectl delete nginx  
error: the server doesn't have a resource type "nginx"  
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $ kubectl delete deploy  
NAME READY UP-TO-DATE AVAILABLE AGE  
nginx-deployment 2/2 2 2 2m27s  
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $ kubectl delete nginx  
error: the server doesn't have a resource type "nginx-deployment"  
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $ kubectl delete deploy  
deployment.apps "nginx-deployment" deleted  
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $ kubectl proxy  
Starting to serve on 127.0.0.1:8001  
^C  
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $ kubectl apply -f k8s-  
serviceaccount/admin-user created
```

# Kubernetes - kubectl

- nginx innerhalb des Clusters testen

```
$ kubectl expose deployment/nginx  
$ kubectl run curl --image=radial/busyboxplus:curl -i --tty  
[ root@curl:/ ] $ nslookup webserv.default.svc.cluster.local  
[ root@curl:/ ] $ curl webserv
```



Terminal - mb@mb-GL502VMZ ~/Desktop/Schulung-Docker

```
Datei Bearbeiten Ansicht Terminal Reiter Hilfe  
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $  
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $  
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $ kubectl expose deployment/nginx-deployment  
service/nginx-deployment exposed  
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $ kubectl run curl --image=radial/busyboxplus:curl -i --t  
ty  
If you don't see a command prompt, try pressing enter.  
[ root@curl:/ ] $ nslookup nginx-deployment  
Server: 10.43.0.10  
Address 1: 10.43.0.10 kube-dns.kube-system.svc.cluster.local  
k8s.io false CSINode  
Name: nginx-deployment  
Address 1: 10.43.248.193 nginx-deployment.default.svc.cluster.local  
[ root@curl:/ ] $ curl nginx-deployment  
<!DOCTYPE html>  
<html>  
<head>  
<title>Welcome to nginx!</title>  
<style> false VolumeAttach  
html { color-scheme: light dark; }  
body { width: 35em; margin: 0 auto;  
font-family: Tahoma, Verdana, Arial, sans-serif; }  
</style>  
</head>  
<body><h1>Not Found</h1>  
<p>The requested URL was not found on this server.</p>  
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $ kub  
ACpcG4jk Sy5Mwxt8TkgaRwcBvw-xQ8Wix1FYvTN1X2E8h  
Out9Vgbu0 Nry2lk0oJ3rJ0xmZwffUfBWvCXai6183RZ1MGzn  
E0913 11:48:07.538886 61816 proxy_server.go:  
E0913 11:57:20.122461 61816 proxy_server.go:  
Example I  
operations: I  
^C  
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $ kub  
eyJhbGciOiJSUzI1NiIsImtpZCI6IiiFcTlvTzgwRjBle  
cHM6Ly9rdWJlc5ldGVzLmRlZmF1bHQuc3ZjLmNsdxN0ZX  
4MTAsImlzcyI6Imh0dHBz0i8va3ViZXJuZXRLcy5kZWhd
```

# Kubernetes - kubectl

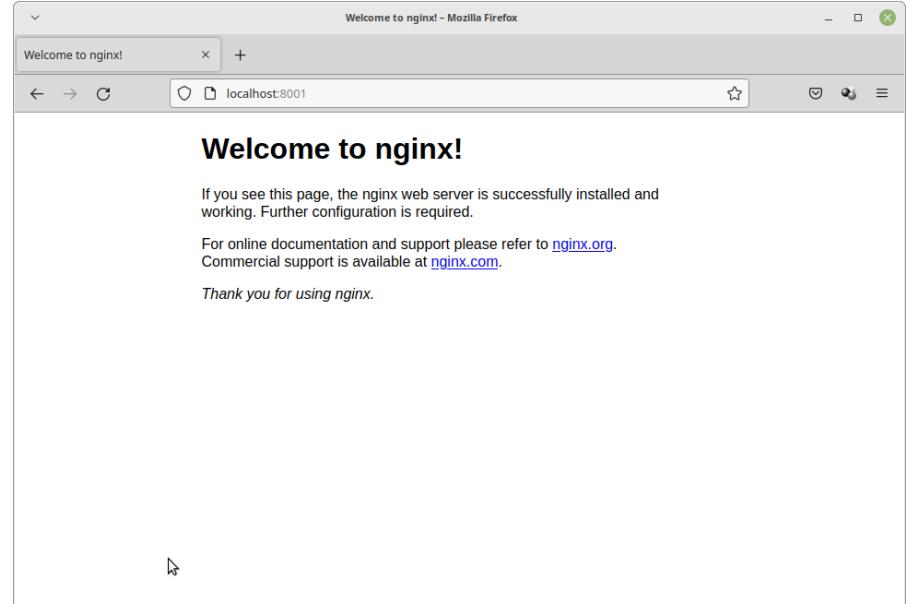
- Port-Forwarding für äußeren Zugriff

```
$ kubectl port-forward service/webserv 8001:80
```



Screenshot of a terminal window showing the command being run:

```
Terminal - mb@mb-GL502VMZ ~/Desktop/Schulung-Docker
Datei Bearbeiten Ansicht Terminal Reiter Hilfe
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $ kubectl port-forward service/webserv 8001:80
Forwarding from 127.0.0.1:8001 -> 80
Forwarding from [::1]:8001 -> 80
Handling connection for 8001
```



# Kubernetes – kubectl ... hint

- Einen einfachen Linux-Container im Cluster als Pod starten
  - Interaktiv
  - Verbindung zur Host-Konsole über stdin und stdout

```
$ kubectl run linux --image=radial/busyboxplus -i --tty  
/ # nslookup linux  
/ # ping linux  
/ # exit
```



- Pod läuft weiter
- Wiederholte Verbindung mit:

```
$ kubectl attach linux -c linux -i --tty
```

- Alternativen:

```
$ kubectl run alpine --image=alpine -i --tty  
$ kubectl run ubuntu --image=ubuntu -i --tty
```

```
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $  
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $  
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $ kubectl run linux --image=radial/busyboxplus -i --tty  
If you don't see a command prompt, try pressing enter.  
/ # nslookup linux  
/bin/sh: nslookup: not found  
/ # nslookup linux  
Server: 10.96.0.10  
Address 1: 10.96.0.10 kube-dns.kube-system.svc.cluster.local  
  
Name: linux  
Address 1: 10.1.0.127 linux  
/ # ping linux  
PING linux (10.1.0.127): 56 data bytes  
64 bytes from 10.1.0.127: seq=0 ttl=64 time=0.510 ms  
64 bytes from 10.1.0.127: seq=1 ttl=64 time=0.136 ms  
64 bytes from 10.1.0.127: seq=2 ttl=64 time=0.196 ms  
^C  
--- linux ping statistics ---  
3 packets transmitted, 3 packets received, 0% packet loss  
round-trip min/avg/max = 0.136/0.280/0.510 ms  
/ # exit  
Session ended, resume using 'kubectl attach linux -c linux -i -t' command when the pod is running  
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $  
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $  
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $  
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $
```

# Kubernetes - Konfiguration

- Über YAML- oder JSON-Dateien
  - Strukturierte Informationen zu Ressourcen
  - Pflichtparameter: apiVersion, kind, metadata (name, id, namespace), spec
  - Spec-Formatierung ist abhängig vom Objekt-Typ
  - <https://kubernetes.io/docs/concepts/workloads/>
  - <https://kubernetes.io/docs/concepts/services-networking/>
- „ausprobieren“ ... und interaktive Anpassung :-(

# Kubernetes - Konfiguration

- Über YAML- oder JSON-Dateien
  - Strukturierte Informationen zu Ressourcen
  - Pflichtparameter: apiVersion, kind, metadata (name, id, namespace), spec
  - Spec-Formatierung ist abhängig vom Objekt-Typ
  - <https://kubernetes.io/docs/concepts/workloads/>
  - <https://kubernetes.io/docs/concepts/services-networking/>
- „ausprobieren“ ... und interaktive Anpassung :-(

# Kubernetes - Konfiguration

- Erzeugung eines Pods und eines ReplicaSets mittels eines Deployments
- Beispiel eines nginx-WEB-Servers

```
apiVersion: v1
kind: Deployment
metadata:
  name: nginx
spec:
  selector:
    matchLabels:
      app: nginx
  replicas: 2
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
          image: nginx:latest
          ports:
            - containerPort: 80
```

# Kubernetes - Konfiguration

- Erzeugung eines Services für das Deployment

```
apiVersion: v1
kind: Service
metadata:
  name: nginx
  labels:
    run: nginx
spec:
  ports:
    - port: 80
      protocol: TCP
  selector:
    app: nginx
  type: LoadBalancer
```

- Deployment & Service in k8s-nginx-komplett.yaml

# Kubernetes - Beispiel

- Zuerst „aufräumen“

```
$ kubectl delete service webserv  
$ kubectl delete deployment webserv  
$ kubectl get all
```



Terminal - mb@mb-GL502VMZ ~/Desktop/Schulung-Docker

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
service/kubernetes	ClusterIP	10.43.0.1	<none>	443/TCP	26d

# Kubernetes - Beispiel

- nginx-Web-Server installieren

```
$ kubectl apply -f k8s-nginx-komplett.yml  
$ kubectl get all
```



Terminal - mb@mb-GL502VMZ ~/Desktop/Schulung-Docker

```
Datei  Bearbeiten  Ansicht  Terminal  Reiter  Hilfe  
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $  
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $ kubectl apply -f k8s-nginx-komplett.yml  
deployment.apps/webserv created  
service/webserv created  
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $ ter  
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $ kubectl get all  
NAME          READY   STATUS    RESTARTS   AGE  
pod/webserv-b798fbfcf-92f8v  1/1     Running   0          5s  
pod/webserv-b798fbfcf-zpw4l  1/1     Running   0          5s  
  
NAME           TYPE      CLUSTER-IP      EXTERNAL-IP   PORT(S)      AGE  
service/kubernetes  ClusterIP  10.96.0.1    <none>        443/TCP    2d1h  
service/webserv   LoadBalancer  10.105.229.131  localhost   80:30952/TCP  5s  
  
NAME          READY   UP-TO-DATE  AVAILABLE   AGE  
deployment.apps/webserv  2/2       2           2           5s  
  
NAME          DESIRED  CURRENT  READY   AGE  
replicaset.apps/webserv-b798fbfcf  2         2        2   5s  
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $  
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $  
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $
```

# Kubernetes - Beispiel

- Service arbeitet als LoadBalancer ... testen

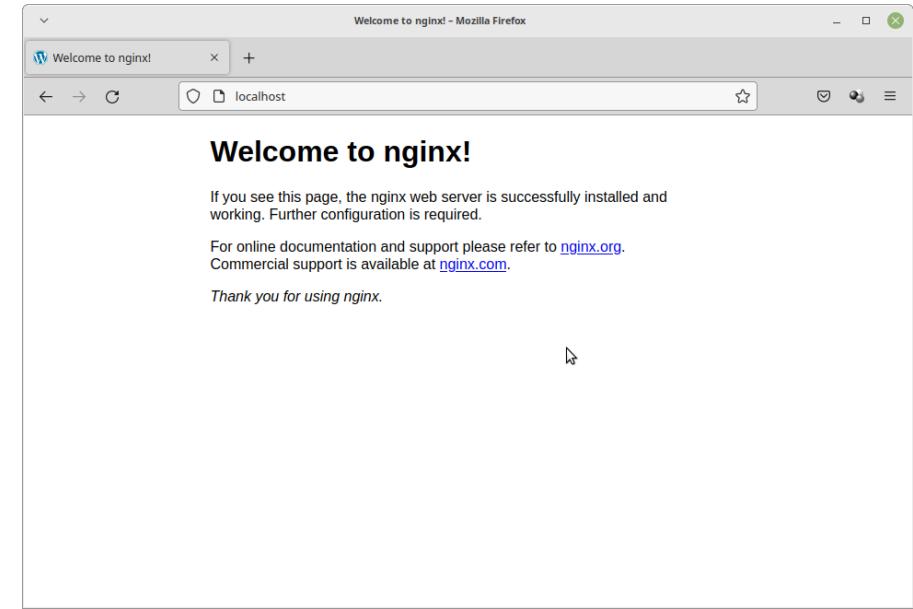
```
$ curl localhost
```

```
http://localhost
```



Terminal - mb@mb-GL502VMZ ~/Desktop/Schulung-Docker

```
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $ 
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $ 
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $ curl localhost
<!DOCTYPE html>
<html> over 6 years ago      955.64 MB
<head> I
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto; font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body> Connect for free
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>
<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>
<p><em>Thank you for using nginx.</em></p>
</body>
</html>
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $
```



# Kubernetes

- Kubernetes-Dashboard installieren

```
$ kubectl apply -f https://raw.githubusercontent.com/kubernetes/dashboard/v2.6.1/aio/deploy/recommended.yaml
```



Rancher Desktop

File Edit View Help

RANCHER DESKTOP

General Port Forwarding Images Troubleshooting

Images

Namespace: k8s.io kube

Image	Tag	Image ID	Size
kubernetesui/dashboard	<none>	290bebc3cd96	245.8 MiB
kubernetesui/dashboard	v2.6.1	290bebc3cd96	245.8 MiB
kubernetesui/metrics-scrapers	<none>	76049887f07a	41.8 MiB
kubernetesui/metrics-scrapers	v1.0.8	76049887f07a	41.8 MiB

Terminal - mb@mb-GL502VMZ ~

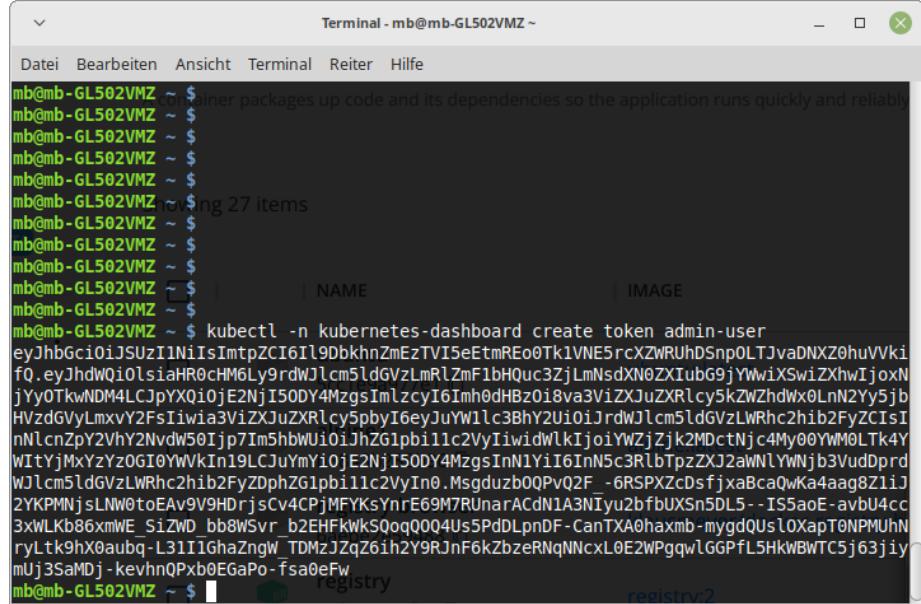
```
Datei Bearbeiten Ansicht Terminal Reiter Hilfe
```

```
mb@mb-GL502VMZ ~ $ container packages up code and its dependencies so the application runs quickly and reliably
mb@mb-GL502VMZ ~ $ namespace/kubernetes-dashboard unchanged
mb@mb-GL502VMZ ~ $ serviceaccount/kubernetes-dashboard unchanged
mb@mb-GL502VMZ ~ $ service/kubernetes-dashboard unchanged
mb@mb-GL502VMZ ~ $ secret/kubernetes-dashboard-certs unchanged
mb@mb-GL502VMZ ~ $ secret/kubernetes-dashboard-csrfs configured IMAGE
Warning: resource secrets/kubernetes-dashboard-key-holder is missing the kubectl.kubernetes.io/last-applied-configuration annotation which is required by kubectl apply. kubectl apply should only be used on resources created declaratively by either kubectl create --save-config or kubectl apply. The missing annotation will be patched automatically.
secret/kubernetes-dashboard-key-holder configured
configmap/kubernetes-dashboard-settings unchanged
role.rbac.authorization.k8s.io/kubernetes-dashboard unchanged Alpine:latest
clusterrole.rbac.authorization.k8s.io/kubernetes-dashboard unchanged
rolebinding.rbac.authorization.k8s.io/kubernetes-dashboard unchanged
clusterrolebinding.rbac.authorization.k8s.io/kubernetes-dashboard unchanged
deployment.apps/kubernetes-dashboard unchanged
service/dashboard-metrics-scraper unchanged
deployment.apps/dashboard-metrics-scraper unchanged
klausmeyer/docker-registry-b
mb@mb-GL502VMZ ~ $ dashboard registry2
```

# Kubernetes

- Kubernetes-Dashboard konfigurieren

```
$ kubectl apply -f k8s-service-account.yml  
$ kubectl apply -f k8s-dashboard-adminuser.yml  
$ kubectl -n kubernetes-dashboard create token admin-user
```



```
mb@mb-GL502VMZ ~ $ container packages up code and its dependencies so the application runs quickly and reliably  
mb@mb-GL502VMZ ~ $  
mb@mb-GL502VMZ ~ $ 1 of 27 items  
mb@mb-GL502VMZ ~ $  
mb@mb-GL502VMZ ~ $  
mb@mb-GL502VMZ ~ $  
mb@mb-GL502VMZ ~ $  | NAME | IMAGE  
mb@mb-GL502VMZ ~ $  
mb@mb-GL502VMZ ~ $  | NAME | IMAGE  
mb@mb-GL502VMZ ~ $ kubectl -n kubernetes-dashboard create token admin-user  
eyJhbGciOiJSUzI1NiIsImtpZC16Il9dbkhnZmEzTVI5eetmREo0Tk1VNE5rcXZRuhDSpn0LTJvaDNXZ0huVVki  
f0.WejhdWQi0lsiaHR0cHM6Ly9rdWJlcm5ldGVzLmRlZmF1bHQuc3ZjLmNsdXN0ZXIubG9jYWwiXswiZXhwIjoxN  
jYyOTkwNDM4LCjpxXQiojE2NjI50DY4MzgsImlzcyI6Imh0dHBz0i8va3ViZXJuZXRLcy5kZWZhdxwOLnN2Yy5jb  
HVzdGvylmxvY2FsIiwi3ViZXJuZXRLcy5pbvyI6eyJuYW1lc3bhY2Ui0iJrdwJlcm5ldGvzlWRhc2hib2FyZCI  
nNlcnPzY2VhY2NvdW50Ijp7Im5hbWUi0iHgZ1pb11c2VyIwidWlkIjoiYWZjZjk2MdctNjc4My00YWM0LTk4Y  
WItYjMxYzYz0GI0YVkkIn19LCJuYm10jE2NjI50DY4MzgsIn1YI16In5c3RltTpzZXJ2awNLyWnjb3VudDprd  
WJlcm5ldGvzLWRhc2hib2FyZDphZG1pb11c2VyIn0.Msgduzb0QPvQ2F -6RSPXZcDsfxaBcaQwKa4aag8Z1j  
2YKPMNjsLNW0toEAyv9V9HDrijsCv4CPjMFYKsYnrE69M7RUnarAcdN1A3N1yu2bfuUXSn5DL5 -IS5aoE -avbU4cc  
3xWLKb86xmWE_SizWD_bb8WSvr_b2EHFKwksQoq0004Us5PdDlpnDF -CanTXA0hxamb -mygdQUslOxapT0NPMuHN  
ryLtk9hXaubq -L3I1lGhaZngW_TDMzJzqZ6ih2Y9RJnf6kZbzErnqNcxL0E2WPqqlGGPfL5HkWBWTC5j63jiy  
muJ3SaMDj -kevhnQPxb0EGaPo -Tsa0EfW  
mb@mb-GL502VMZ ~ $ registry  
registry.v2
```

# Kubernetes

- Kubernetes-Dashboard starten

```
$ kubectl proxy
```

```
http://localhost:8001/api/v1/namespaces/kubernetes-dashboard/services/https:kubernetes-dashboard:/proxy/
```



```
Terminal - mb@mb-GL502VMZ ~
```

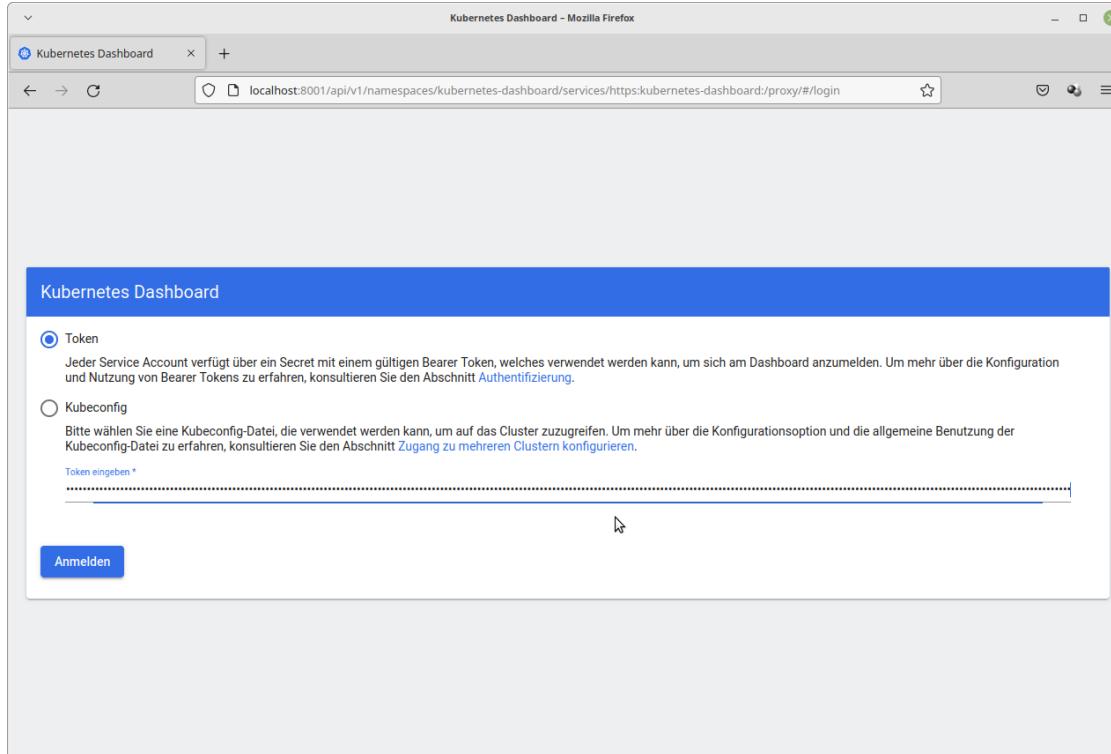
NAME	IMAGE
ubuntu	ubuntu:latest
alpine1	alpine:latest
registry-browser	klausmeyer/docker-registry-b
kubectl proxy	registry

```
Starting to serve on 127.0.0.1:8001
```

```
registry2
```

# Kubernetes

- Kubernetes-Dashboard starten ... zuvor erzeugten Token nutzen



# Kubernetes

- Kubernetes-Dashboard starten

The screenshot shows the Kubernetes Dashboard interface in Mozilla Firefox. The title bar reads "Kubernetes Dashboard - Mozilla Firefox". The address bar shows the URL "localhost:8001/api/v1/namespaces/kubernetes-dashboard/services/https:kubernetes-dashboard:/proxy/#/discovery?namespace=default". The main header has a "kubernetes" logo and dropdown menus for "default" and "Suchen". Below the header is a blue navigation bar with the text "Service". On the left, there is a sidebar titled "Workloads" with links to Cron Jobs, Daemon Sets, Deployments, Jobs, Pods, Replica Sets, Replication Controllers, and Stateful Sets. The "Service" link is highlighted. The main content area is titled "Services" and contains a table with the following data:

Name	Labels	Typ	Cluster-IP	Interne Endpoints	Externe Endpoints
kubernetes	component: apiserver provider: kubernetes	ClusterIP	10.96.0.1	kubernetes:443 TCP kubernetes:0 TCP	-

Below the table, there are sections for "Ingresses", "Ingress Classes", and "Services". At the bottom, there are sections for "Konfiguration und Datenspeicherung" (Config Maps, Persistent Volume Claims, Secrets) and "Dokumentation" (Documentation, API Reference, Glossary).

# Kubernetes

The screenshot shows the Kubernetes Dashboard interface running in Mozilla Firefox. The title bar reads "Kubernetes Dashboard - Mozilla Firefox". The address bar shows the URL `localhost:8001/api/v1/namespaces/kubernetes-dashboard/services/https:kubernetes-dashboard:/proxy/#/workloads`. The dashboard has a header with the Kubernetes logo, a dropdown menu set to "default", a search bar with placeholder "Suchen", and three icons for "+" (Create), a bell (Notifications), and a user profile.

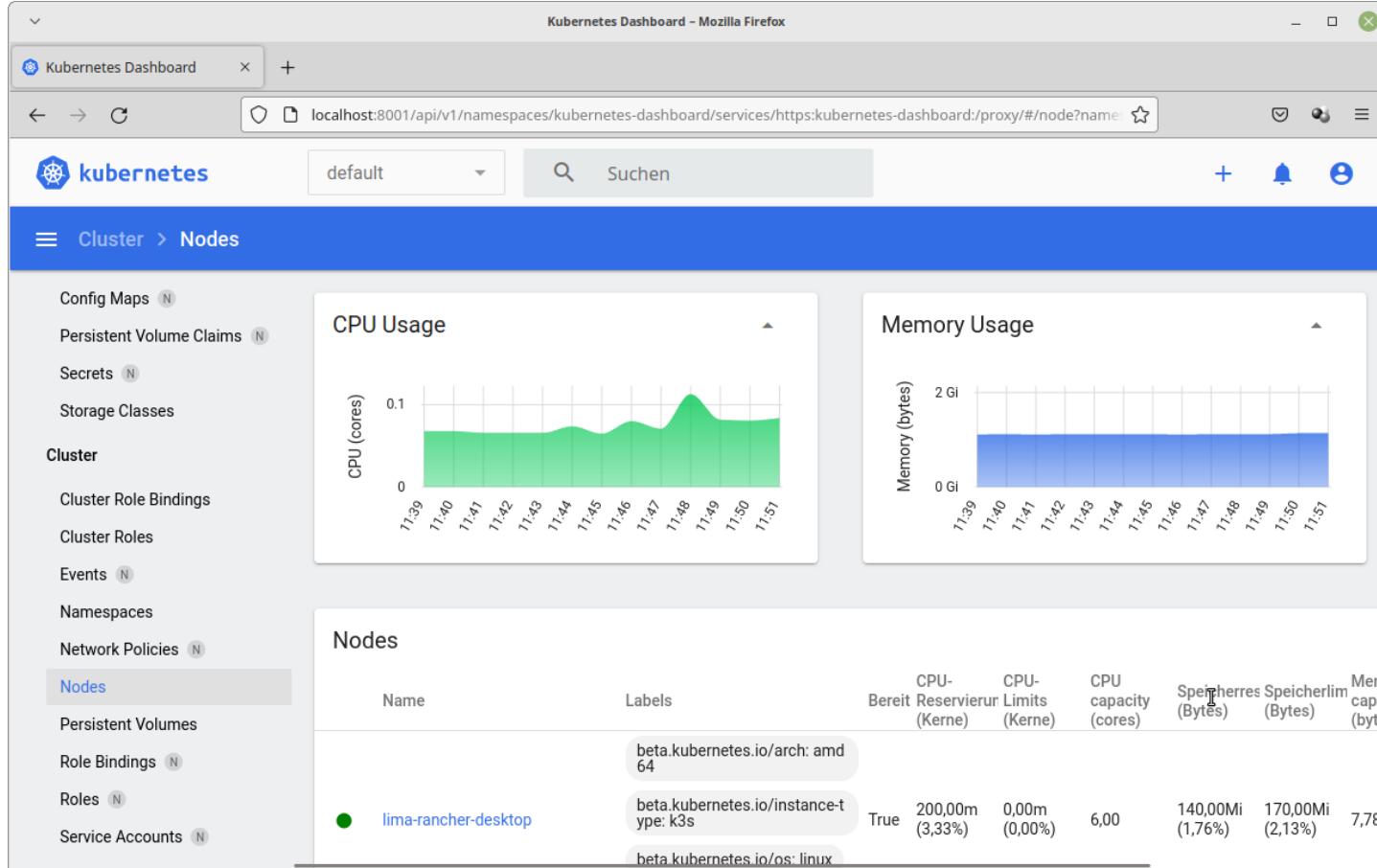
The main area is titled "Workloads" and displays a summary of the status of three resource types:

- Deployments:** Shows one green circle representing "Running: 1".
- Pods:** Shows two green circles representing "Running: 2".
- Replica Sets:** Shows one green circle representing "Running: 1".

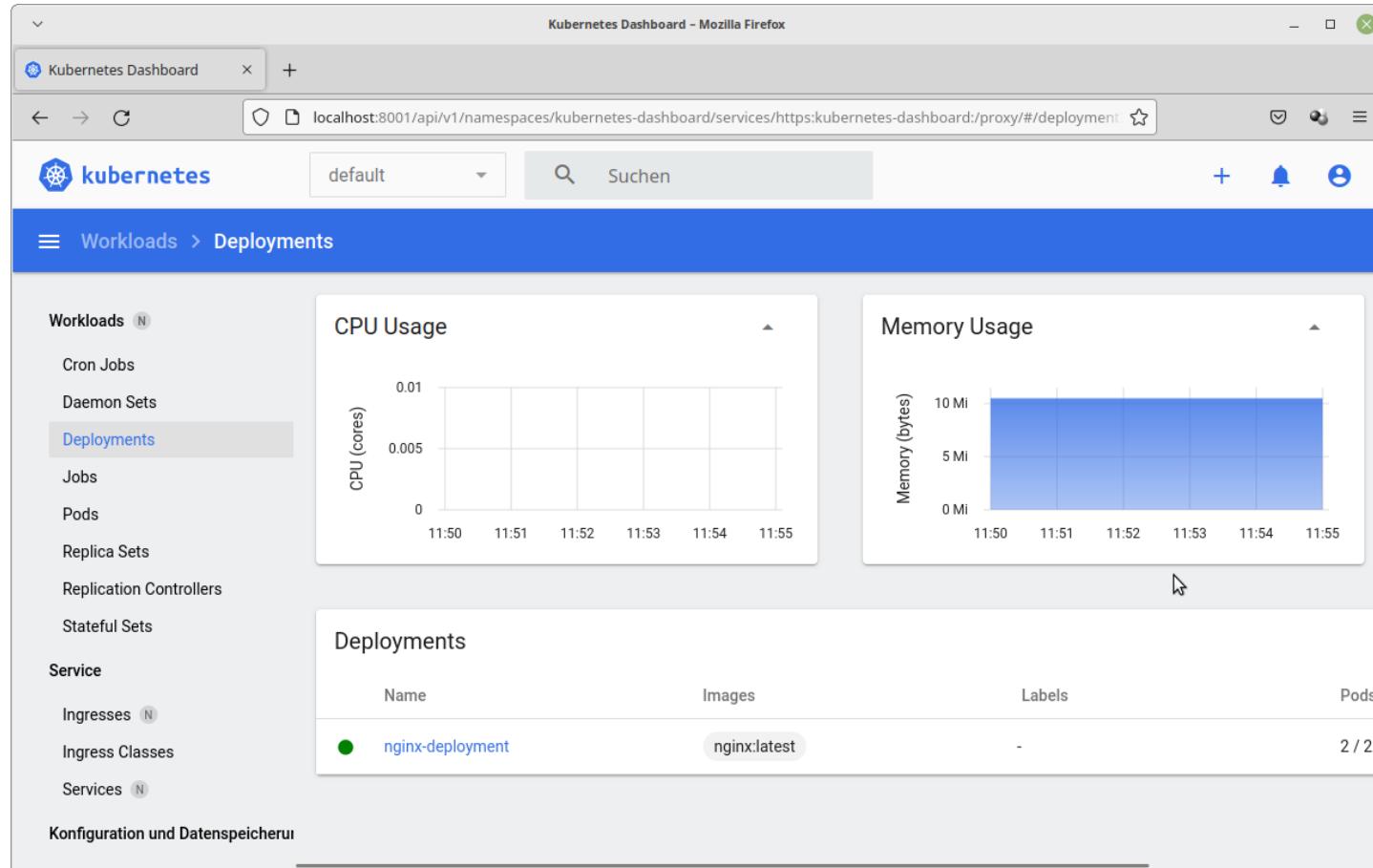
Below this summary, there is a table titled "Deployments". The table has a header row with columns: Name, Images, Labels, Pods, and Erstellungszeitpunkt (CreationTimestamp). There is one data row shown in the table.

Name	Images	Labels	Pods	Erstellungszeitpunkt
DeploymentName	ImageName	Label1, Label2	2	2023-09-12T10:00:00Z

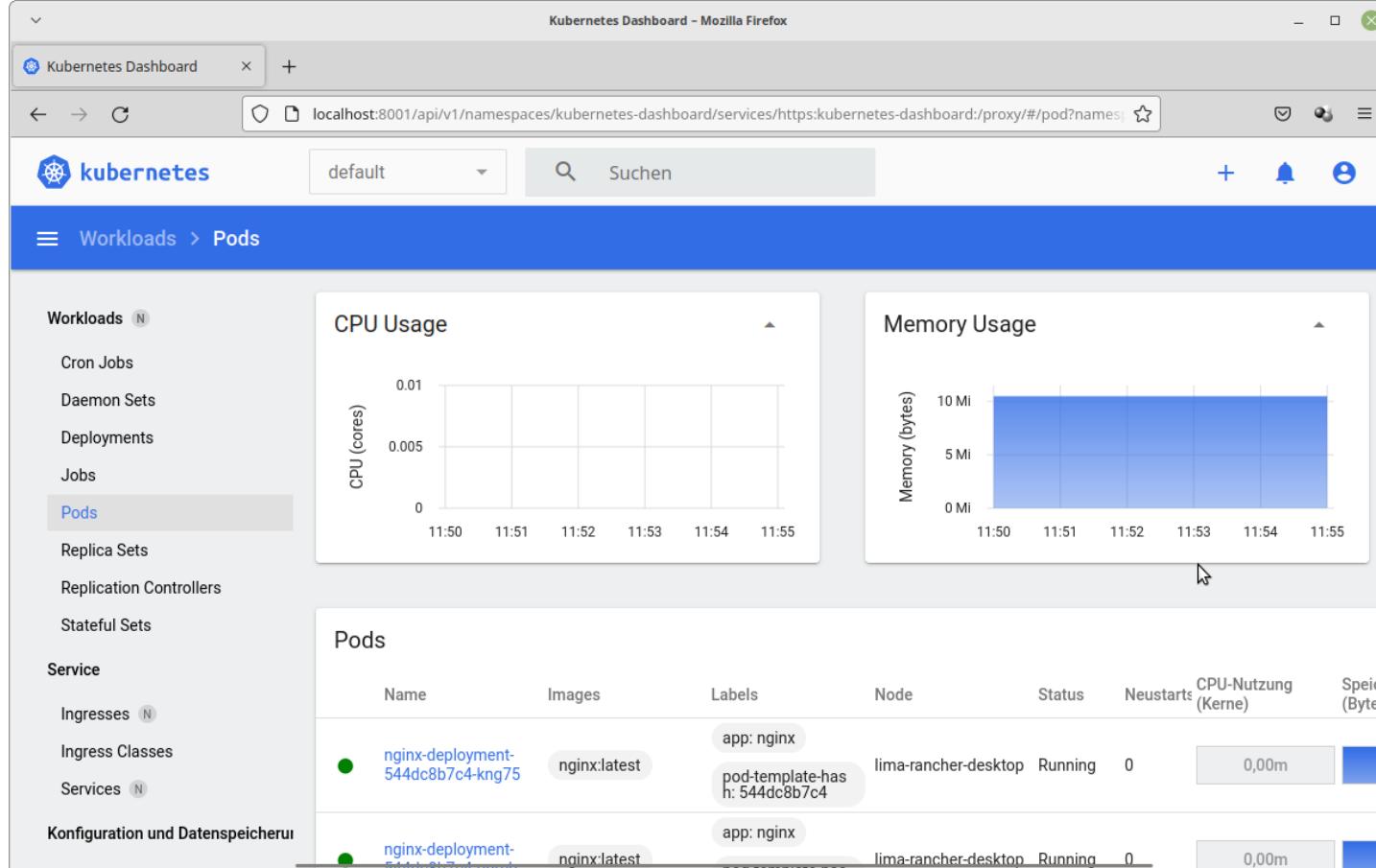
# Kubernetes



# Kubernetes



# Kubernetes



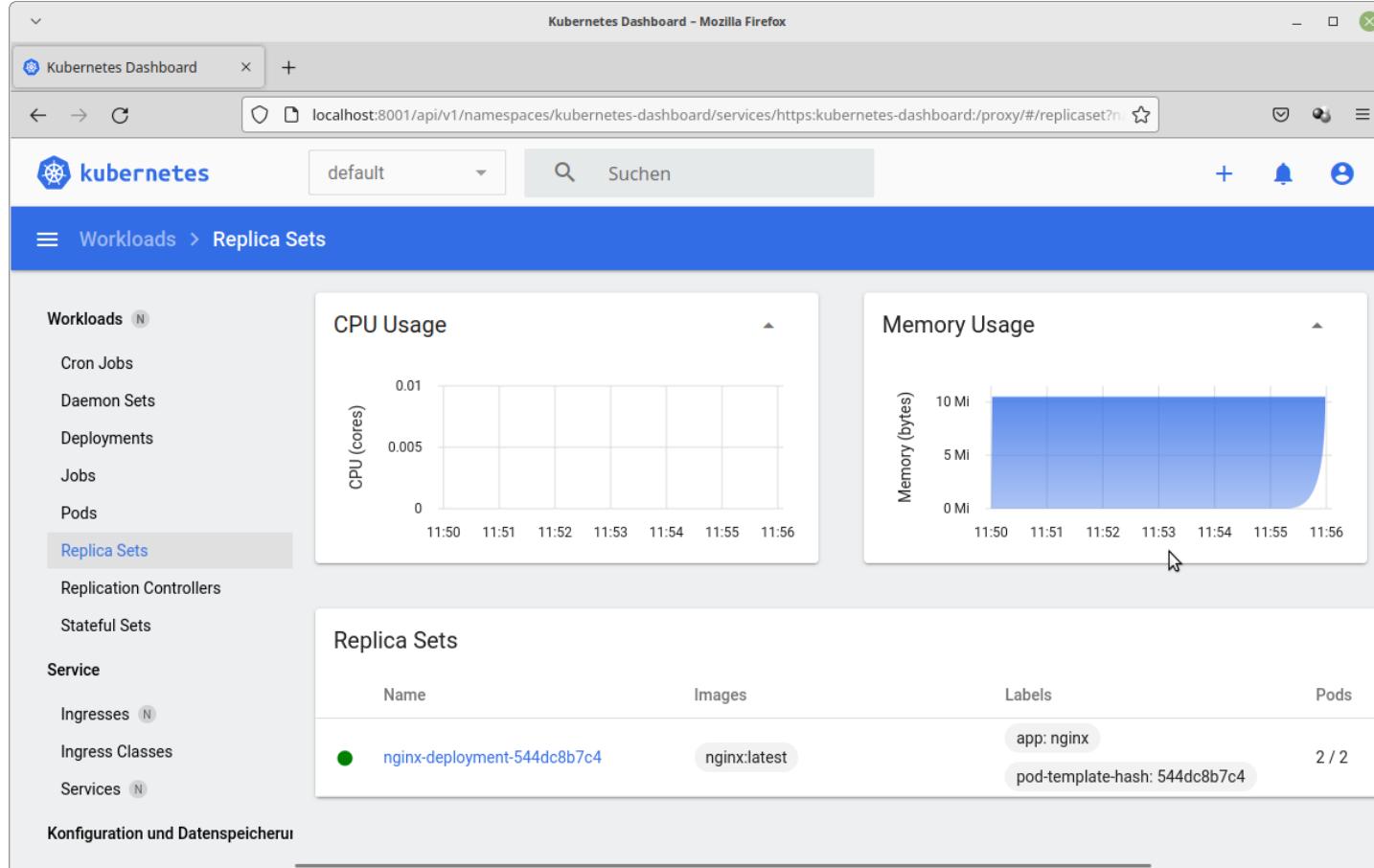
# Kubernetes

The screenshot shows a Mozilla Firefox browser window displaying the Kubernetes Dashboard. The URL in the address bar is `localhost:8001/api/v1/namespaces/kubernetes-dashboard/services/https:kubernetes-dashboard:/proxy/#/log/default/`. The dashboard interface includes a navigation bar with tabs for 'Kubernetes Dashboard' and 'default'. Below the navigation is a search bar with the placeholder 'Suchen'. The main content area is titled 'Workloads > Pods > nginx-deployment-544dc8b7c4-kng75 > Logs'. The log viewer displays the following output:

```
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2022/09/13 09:49:36 [notice] 1#1: using the "epoll" event method
2022/09/13 09:49:36 [notice] 1#1: nginx/1.23.1
2022/09/13 09:49:36 [notice] 1#1: built by gcc 10.2.1 20210110 (Debian 10.2.1-6)
2022/09/13 09:49:36 [notice] 1#1: OS: Linux 5.15.57-0-virt
2022/09/13 09:49:36 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2022/09/13 09:49:36 [notice] 1#1: start worker processes
2022/09/13 09:49:36 [notice] 1#1: start worker process 31
2022/09/13 09:49:36 [notice] 1#1: start worker process 32
2022/09/13 09:49:36 [notice] 1#1: start worker process 33
2022/09/13 09:49:36 [notice] 1#1: start worker process 34
2022/09/13 09:49:36 [notice] 1#1: start worker process 35
2022/09/13 09:49:36 [notice] 1#1: start worker process 36
```

At the bottom of the log viewer, it says 'Logs from 13.09.2022 to 13.09.2022 UTC' and features navigation arrows for the log scroll.

# Kubernetes



# Kubernetes - Helm

- <https://helm.sh/>
- Package-Manager für K8s-Apps
- Packages = Charts ... wieder TAR-Archive
- Nutzung von Repositories
  - Verwaltung von Apps und Versionen
  - Austausch von Apps
- Teil des CNCF-Stacks

# Kubernetes - Helm

- Funktionsweise
  - Sammlung von YAML- und JSON-Files, Zertifikaten, ...
  - Am Ende ... YAML-Dateien für K8s-Konfiguration
  - Parametrisierung, Funktionen
  - Charts sind kaskadierbar & modifizierbar
  - Templates
  - Kommandozeilentool für die Erzeugung, Installation, Verwaltung, Anpassung und das Entfernen kompletter Anwendungen in K8s

# Kubernetes - Helm

- Chart-Verzeichnisstruktur (wesentliche Elemente)

```
<chart-name>/  
  Chart.yaml      # Basisinformationen zum Chart  
  README.md      # OPTIONAL: README-Datei für Dokumentation  
  values.yaml     # Default-Werte für die Installation  
  charts/         # Verzeichnis mit Charts, von denen dieses abhängt  
  files/          # OPTIONAL: spezielle Ressourcen-Definitionen  
  templates/      # Verzeichnis von Templates ... werden mit  
                  # values.yaml kombiniert und bilden die YAML-Dateien  
                  # für dieK8s-Konfiguration
```

# Kubernetes - Helm

- Chart.yaml (wesentliche Elemente)

```
apiVersion: v2
name: webserv
namespace: webserv
version: 0.1.0
appVersion: "0.1.0"
description: A Helm chart for installation of simple webserv
home: https://mydomain.de
maintainers:
- name: Matthias Boldt
  email: Matthias.Boldt@mydomain.de
```

- Erforderlich sind nur:
  - apiVersion
  - name
  - version

# Kubernetes - Helm

- Templates (Ausschnitt aus deployment.yaml)

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: {{ include "webserv.fullname" . }}
  labels:
    {{- include "webserv.labels" . | nindent 4 }}
spec:
  replicas: {{ .Values.replicaCount }}
  selector:
    matchLabels:
      {{- include "webserv.selectorLabels" . | nindent 6 }}
template:
  metadata:
    labels:
      {{- include "webserv.selectorLabels" . | nindent 8 }}
  spec:
    containers:
      - name: {{ .Chart.Name }}
        image: "{{ .Values.image.repository }}:{{ .Values.image.tag }}"
        imagePullPolicy: {{ .Values.image.pullPolicy }}
        ports:
          - name: http
            containerPort: 80
```

# Kubernetes - Helm

- Default-Variablen in values.yaml
  - Können bei Installation überschrieben werden
  - Es sind mehrere, alternative Dateien möglich ... Auswahl bei Installation

```
replicaCount: 2

image:
  repository: nginx
  pullPolicy: IfNotPresent
  tag: "latest"

imagePullSecrets: []
nameOverride: ""
fullnameOverride: ""

service:
  type: LoadBalancer
  port: 80
```

# Kubernetes - Helm

- Ausgesuchte, vordefinierte Variablen
  - Release.Name: Name der Relese
  - Release.Namespace: Name des Namespace
  - Chart: Inhalt von Chart.yaml ... z.B. Chart.Version
  - Values: Inhalt von values.yaml
- Variablenersetzung/im Scripting nutzt
  - vordefinierte Variablen
  - Installationsparameter
  - Variablen aus values.yaml

# Kubernetes - Helm

- Variablenersetzung und Scripting:
  - {{ <variable> }} ... z.B. {{ .Values.image.repository }}
  - Überschreibbar mit --set <variable>=<value> bei Installation/Update
  - Bedingte Verarbeitung mit Vergleichen, Schleifen, ...
  - Funktionen
  - Pipelining: {{ .Values.favorite.drink | quote }}

# Kubernetes - Helm

- Archive
  - \*.tgz ... komprimierte TAR-Archive
  - Können signiert werden
  - Verzeichnisse:
    - <https://artifacthub.io/>
    - <https://bitnami.com/stacks/helm>
    - ...

# Kubernetes - Helm

- Wesentliche Helm-Kommandos
  - create ... Erzeugung eines neuen (leeren) Charts
  - package ... packt ein Tar-Archiv des Charts
  - lint ... prüft ein Chart auf Vollständigkeit, Fehler, ...
  - list ... listet alle installierten Charts auf
  - install ... installiert ein Chart (die entsprechende App)
  - uninstall ... entfernt ein Chart (die entsprechende App)
  - update ... aktualisiert die Installation eines Charts
  - repo, pull, push ... Verwaltung von Reposities und dortigen Charts

# Kubernetes - Helm

- Helm installieren

```
$ curl -fsSL -o get_helm.sh https://raw.githubusercontent.com/helm/helm/main/scripts/get-helm-3
$ chmod 700 get_helm.sh
$ ./get_helm.sh
```



```
$ helm repo add bitnami https://charts.bitnami.com/bitnami
$ helm search repo bitnami
```

Terminal - mb@mb-GL502VMZ ~

```
Datei Bearbeiten Ansicht Terminal Reiter Hilfe
mb@mb-GL502VMZ ~ $ 
mb@mb-GL502VMZ ~ $ 
mb@mb-GL502VMZ ~ $ helm repo add bitnami https://charts.bitnami.com/bitnami
"bitnami" has been added to your repositories
mb@mb-GL502VMZ ~ $ helm search repo bitnami
NAME          CHART VERSION APP VERSION DESCRIPT
ION          Service      13.1.4     2.3.4      Apache A
bitnami/airflow  Service      13.1.4     2.3.4      Apache A
irflow is a tool to express and execute...
bitnami/apache   Service      9.2.3      2.4.54    Apache H
HTTP Server is an open-source HTTP serve...
bitnami/argo-cd  Workloads   4.1.3      2.4.11    Argo CD
is a continuous delivery tool for Kuber...
bitnami/argo-workflows  Jobs      2.4.3      3.3.9      Argo Wor
kflows is meant to orchestrate Kuber...
bitnami/aspnet-core  DaemonSets  3.5.2      6.0.8      ASP.NET
Core is an open-source framework for we...
bitnami/cassandra Deployments  9.4.2      4.0.6      Apache C
assandra is an open source distributed ...
bitnami/cert-manager  Deployments  0.8.2      1.9.1      Cert Man
ager is a Kubernetes add-on to automate...
bitnami/common     Pods       2.0.3      2.0.3      A Librar
y Helm Chart for grouping common logic ...
Do
```

Terminal - mb@mb-GL502VMZ ~

```
Datei Bearbeiten Ansicht Terminal Reiter Hilfe
mb@mb-GL502VMZ ~ $ 
mb@mb-GL502VMZ ~ $ 
mb@mb-GL502VMZ ~ $ helm repo add bitnami https://charts.bitnami.com/bitnami
"bitnami" has been added to your repositories
mb@mb-GL502VMZ ~ $ helm search repo bitnami
NAME          CHART VERSION APP VERSION DESCRIPT
ION          Service      13.1.4     2.3.4      Apache A
bitnami/airflow  Service      13.1.4     2.3.4      Apache A
irflow is a tool to express and execute...
bitnami/apache   Service      9.2.3      2.4.54    Apache H
HTTP Server is an open-source HTTP serve...
bitnami/argo-cd  Workloads   4.1.3      2.4.11    Argo CD
is a continuous delivery tool for Kuber...
bitnami/argo-workflows  Jobs      2.4.3      3.3.9      Argo Wor
kflows is meant to orchestrate Kuber...
bitnami/aspnet-core  DaemonSets  3.5.2      6.0.8      ASP.NET
Core is an open-source framework for we...
bitnami/cassandra Deployments  9.4.2      4.0.6      Apache C
assandra is an open source distributed ...
bitnami/cert-manager  Deployments  0.8.2      1.9.1      Cert Man
ager is a Kubernetes add-on to automate...
bitnami/common     Pods       2.0.3      2.0.3      A Librar
y Helm Chart for grouping common logic ...
Do
```

# Kubernetes - Helm

- Helm-Chart installieren
  - Wieder einmal Wordpress ... aus dem Bitnami-Repository

```
$ helm search repo bitnami/wordpress  
$ helm install bitnami/wordpress --generate-name
```



The terminal window shows the following steps:

```
mb@mb-GL502VMZ ~ $ helm search repo bitnami/wordpress  
mb@mb-GL502VMZ ~ $ helm install bitnami/wordpress --generate-name
```

After the installation, the terminal displays the following message:

Your WordPress site can be accessed through the following DNS name from within your cluster:  
wordpress-1662988610.default.svc.cluster.local (port 80)

To access your WordPress site from outside the cluster follow the steps below:

1. Get the WordPress URL by running these commands:

```
export SERVICE_IP=$(kubectl get svc --namespace default --selector app=wordpress -o yaml | grep ip | awk '{print $2}')  
echo "WordPress URL: http://$SERVICE_IP/"  
echo "WordPress Admin URL: http://$SERVICE_IP/admin"
```
2. Open a browser and access WordPress using the obtained URL.
3. Login with the following credentials below to see your blog:

```
echo Username: user  
echo Password: $(kubectl get secret --namespace default wordpress-1662988610 -o yaml | base64 -d | jq -r '.data.wordpress-password')
```

# Kubernetes - Helm

- Wordpress-Chart installiert ... Test

<http://localhost:80>



Kubernetes Dashboard – Mozilla Firefox

Kubernetes Dashboard default Suchen

kubernetes

Service

Workloads

Cron Jobs

Daemon Sets

Deployments

Jobs

Pods

Replica Sets

Replication Controllers

Stateful Sets

Service

Ingresses

Ingress Classes

Services

Konfiguration und Datenspeicherung

Config Maps

Persistent Volume Claims

Secrets

Services

Name	Labels	Typ	Cluster-IP	Interne Endpoints
wordpress-1662988610	app.kubernetes.io/instance: wordpress-1662988610 app.kubernetes.io/managed-by: Helm app.kubernetes.io/name: wordpress	LoadBalancer	10.103.54.191	wordpress-1662988610:80 TCP wordpress-1662988610:31792 TCP wordpress-1662988610:443 TCP wordpress-1662988610:31404 TCP
wordpress-1662988610-mariadb	app.kubernetes.io/component: primary app.kubernetes.io/instance: Wordpress-1662988610 app.kubernetes.io/managed-by: Helm	ClusterIP	10.105.28.13	wordpress-1662988610-mariadb:3306 TCP wordpress-1662988610-mariadb:0 TCP
kubernetes	component: apiserver provider: kubernetes	ClusterIP	10.96.0.1	kubernetes:443 TCP kubernetes:0 TCP

User's Blog! – Just another WordPress site – Mozilla Firefox

localhost

User's Blog!

Sample Page

Hello world!

Welcome to WordPress. This is your first post. Edit or delete it, then start writing!

The image displays two screenshots of web browsers. The left screenshot shows the 'Kubernetes Dashboard' in Mozilla Firefox, specifically the 'Services' section. It lists three services: 'wordpress-1662988610', 'wordpress-1662988610-mariadb', and 'kubernetes'. Each service entry includes its name, labels, type, cluster IP, and internal endpoints. The 'wordpress-1662988610' entry has labels like 'app.kubernetes.io/instance: wordpress-1662988610', 'app.kubernetes.io/managed-by: Helm', and 'app.kubernetes.io/name: wordpress'. The right screenshot shows a WordPress blog post titled 'User's Blog!' with the content 'Hello world!' and a welcome message: 'Welcome to WordPress. This is your first post. Edit or delete it, then start writing!'. There is also a decorative illustration of a hummingbird in flight.

# Kubernetes – Helm – ein Chart

- Neues Chart erzeugen

```
$ helm create <name>
```

```
$ helm create webserv
$ ls -l webserv
```

- Chart editieren
  - Vorbereitet
  - webserv



```
Terminal - mb@mb-GL502VMZ ~/Schulung-Docker
Datei Bearbeiten Ansicht Terminal Reiter Hilfe
mb@mb-GL502VMZ ~/Schulung-Docker $ 
mb@mb-GL502VMZ ~/Schulung-Docker $ helm create webserv
Creating webserv directory in that directory, conflicting files
mb@mb-GL502VMZ ~/Schulung-Docker $ ls -l webserv/
insgesamt 16
drwxr-xr-x 2 mb mb 4096 Sep 14 20:34 charts
-rw-r--r--g1 mb mb 1143 Sep 14 20:34 Chart.yaml
drwxr-xr-x 3 mb mb 4096 Sep 14 20:34 templates
-lw-r--r-- 1 mb mb 1874 Sep 14 20:34 values.yaml
mb@mb-GL502VMZ ~/Schulung-Docker $ 
mb@mb-GL502VMZ ~/Schulung-Docker $ cat to Helm starter scaffold
mb@mb-GL502VMZ ~/Schulung-Docker $ 
mb@mb-GL502VMZ ~/Schulung-Docker $
```

# Kubernetes – Helm – ein Chart

- Chart packen und validieren

```
$ helm package <directory>
$ helm lint <tar archive>
```

```
$ helm package webserv
$ helm lint webserv-0.1.0.tgz
```



Terminal - mb@mb-GL502VMZ ~/Desktop/Schulung-Docker

```
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $ helm package webserv
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $ helm lint webserv-0.1.0.tgz
==> Linting webserv-0.1.0.tgz
[INFO] Chart.yaml: icon is recommended

1 chart(s) linted, 0 chart(s) failed
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $ mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $ mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $
```

# Kubernetes – Helm – ein Chart

- Chart installieren

```
$ helm install <name> <tar archive>
```

```
$ helm install webserv webserv-0.1.0.tgz
```



A screenshot of a terminal window titled "Terminal - mb@mb-GL502VMZ ~/Desktop/Schulung-Docker". The terminal shows the command \$ helm install webserv webserv-0.1.0.tgz being run, followed by deployment details:

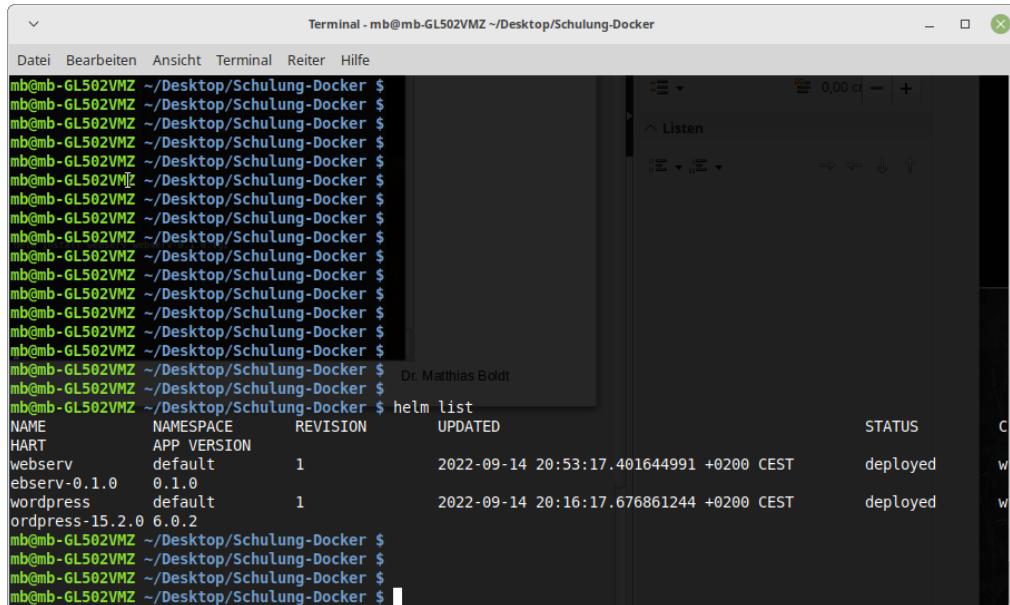
```
Dr. Matthias Boldt
NAME: webserv
LAST DEPLOYED: Wed Sep 14 20:53:17 2022
NAMESPACE: default
STATUS: deployed
REVISION: 1
TEST SUITE: None
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $
```

# Kubernetes – Helm – ein Chart

- Chart auflisten und testen

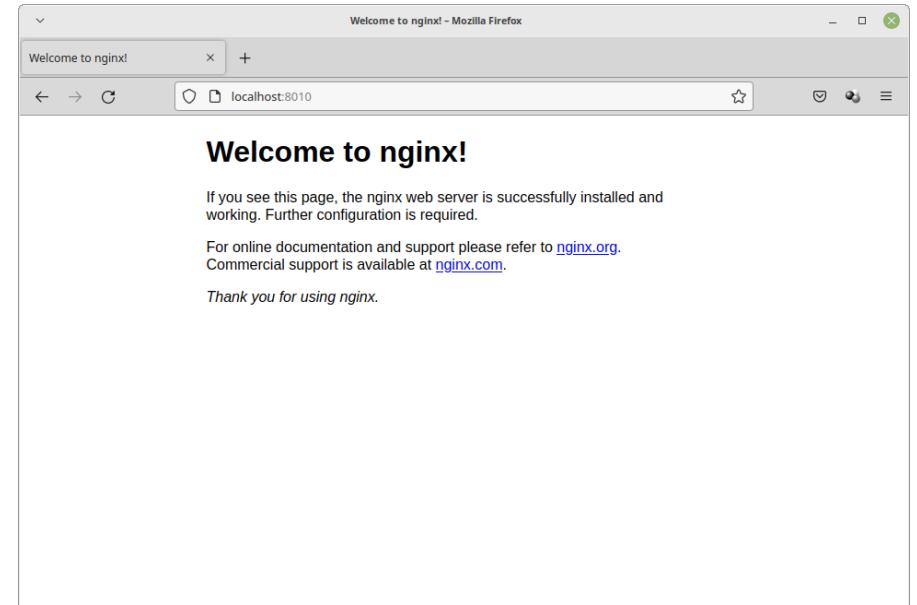
```
$ helm list
```

<http://localhost:8010>



A terminal window titled "Terminal - mb@mb-GL502VMZ ~/Desktop/Schulung-Docker". The user has run the command \$ helm list. The output shows three charts: "webserv" (revision 1), "ebsrv-0.1.0" (revision 0.1.0), and "wordpress" (revision 1). Each entry includes the name, namespace, app version, revision, update time, status, and chart name.

NAME	NAMESPACE	APP VERSION	REVISION	UPDATED	STATUS	CHART
webserv	default	0.1.0	1	2022-09-14 20:53:17.401644991 +0200 CEST	deployed	webserv
ebsrv-0.1.0	default	0.1.0	0	2022-09-14 20:16:17.676861244 +0200 CEST	pending	ebsrv-0.1.0
wordpress	default	6.0.2	1	2022-09-14 20:16:17.676861244 +0200 CEST	deployed	wordpress



# Kubernetes – Helm – ein Chart

- Chart (und App) entfernen

```
$ helm uninstall <name>
```

```
$ kubectl get all  
$ helm uninstall webserv
```



```
Terminal - mb Datei Bearbeiten Ansicht Terminal Reiter Hilfe  
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $ mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $  
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $ mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $  
NAME READY STATUS RESTARTS AGE  
pod/alpine 1/1 Running 1 (4h35m ago) 4h36m  
pod/linux 1/1 Running 2 (4h36m ago) 4h40m  
pod/ubuntu 1/1 Running 1 (4h34m ago) 4h35m  
pod/webserv-7c4bf5c77b-9hcsd 1/1 Running 0 54m  
pod/webserv-7c4bf5c77b-wgslc 1/1 Running 0 54m  
pod/wordpress-68944c8778-49ffl 1/1 Running 0 54m  
pod/wordpress-mariadb-0 1/1 Running 0 54m  
  
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE  
service/kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 2d7h  
service/webserv LoadBalancer 10.102.162.162 localhost 80:31024/TCP,443:32345/TCP 54m  
service/wordpress LoadBalancer 10.99.230.143 <none> 3306/TCP 54m  
service/wordpress-mariadb ClusterIP Dr. Matthias Boldt  
  
NAME READY UP-TO-DATE AVAILABLE AGE  
deployment.apps/webserv 2/2 2 1 54m  
deployment.apps/wordpress 1/1 1 1 54m  
  
NAME DESIRED CURRENT READY AGE  
replicaset.apps/webserv-7c4bf5c77b 2 2 2 54m  
replicaset.apps/wordpress-68944c8778 1 1 1 54m  
  
NAME READY AGE  
statefulset.apps/wordpress-mariadb 1/1 54m  
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $ helm uninstall webserv  
release "webserv" uninstalled  
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $  
  
Terminal - mb@mb-GL502VMZ ~/Desktop/Schulung-Docker Datei Bearbeiten Ansicht Terminal Reiter Hilfe  
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $ mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $  
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $ kubectl get all  
NAME READY STATUS RESTARTS AGE  
pod/alpine 1/1 Running 1 (4h35m ago) 4h36m  
pod/linux 1/1 Running 2 (4h36m ago) 4h40m  
pod/ubuntu 1/1 Running 1 (4h34m ago) 4h35m  
pod/wordpress-68944c8778-49ffl 1/1 Running 0 54m  
pod/wordpress-mariadb-0 1/1 Running 0 54m  
  
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE  
service/kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 2d7h  
service/webserv LoadBalancer 10.102.162.162 localhost 80:31024/TCP,443:32345/TCP 54m  
service/wordpress LoadBalancer 10.99.230.143 <none> 3306/TCP 54m  
  
NAME READY UP-TO-DATE AVAILABLE AGE  
deployment.apps/wordpress 1/1 1 1 54m  
  
NAME DESIRED CURRENT READY AGE  
replicaset.apps/wordpress-68944c8778 1 1 1 54m  
  
NAME READY AGE  
statefulset.apps/wordpress-mariadb 1/1 54m  
mb@mb-GL502VMZ ~/Desktop/Schulung-Docker $
```

# ... Bildquellen

- RedHat
- IDG
- Docker
- Earthly
- Aquasec
- dev.to
- ScaleUp