Kubernetes @ SWISS TXT

Theory and Practice - Daniel Menet

Agenda

Intro to Docker/Kubernetes

Kubernetes @ SWISS TXT?

Kubernetes Resources & Applications

[Q&A]

About me

Daniel Menet

- System Engineer at SWISS TXT, focus on tooling and automation.
- Contributor to Docker...

```
hostname, term, err := ResolveRepositoryName(term)
if err != nil { return job.Error(err) }
hostname, err = ExpandAndVerifyRegistryUrl(hostname)
if err != nil { return job.Error(err) }
```

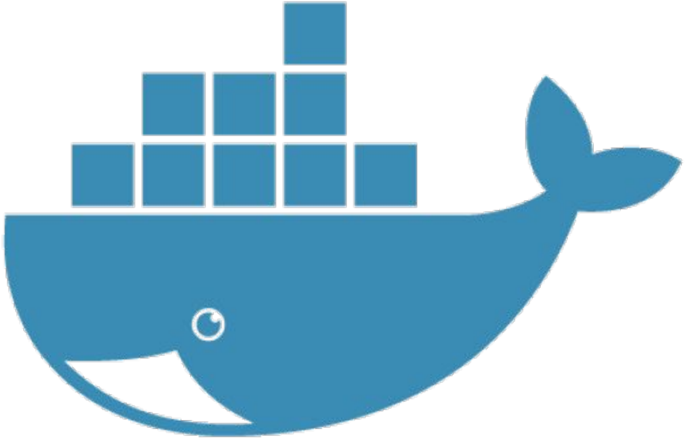
- … and Ansible (1 line changed, 1 line added).
- No books nor awards.

Intro to Docker/Kubernetes

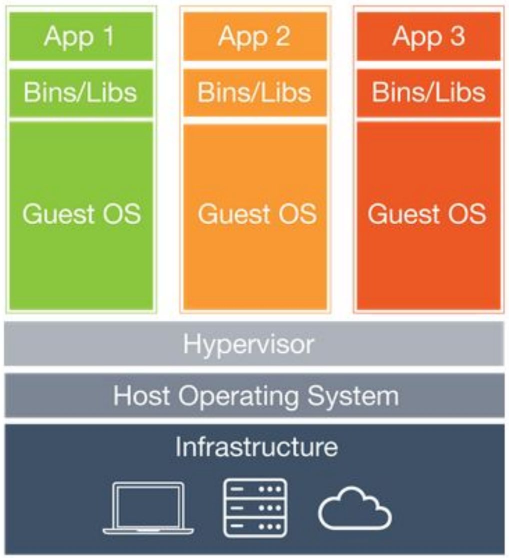
What is Docker?

Software Packaging and Tooling

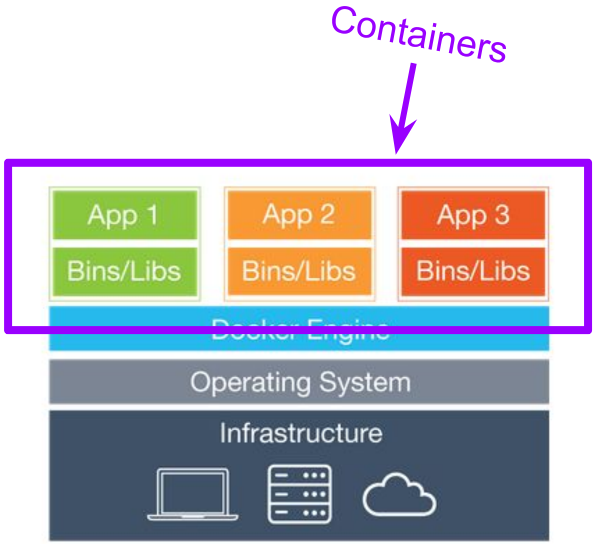
Docker containers ***wrap a piece of software in a complete filesystem that contains everything needed to run****: code, runtime, system tools, system libraries – anything that can be installed on a server. This guarantees that the software will always run the same, regardless of its environment.*



What is Docker? (cont.)

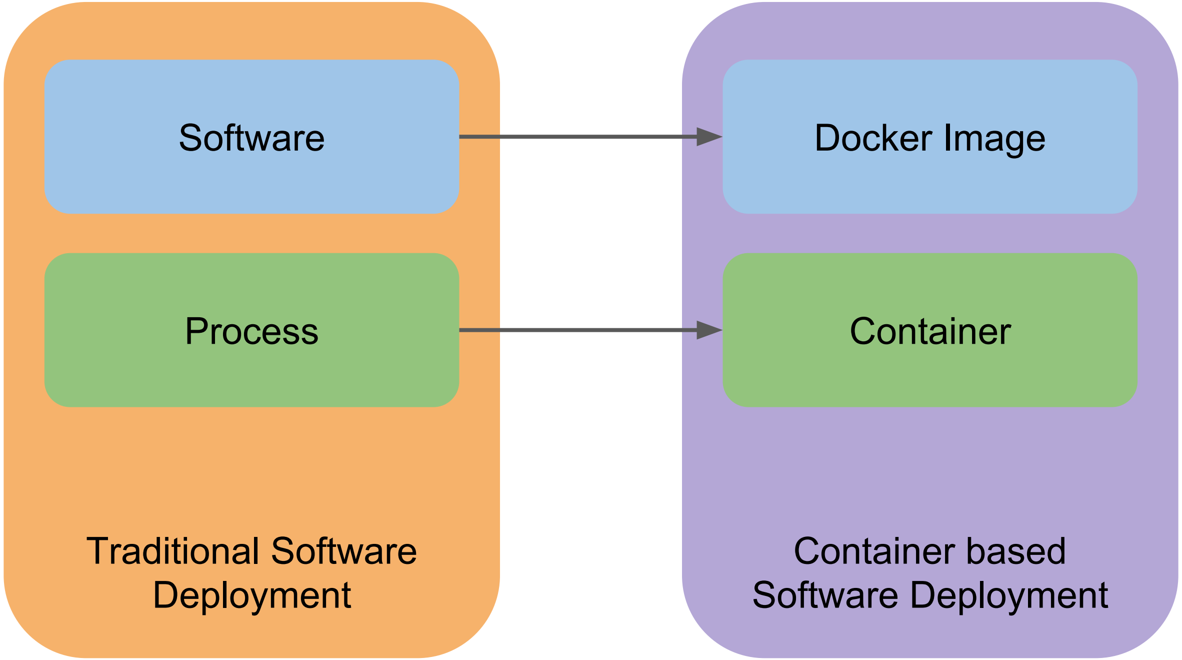


Traditional Virtualization



Containers

Docker Image ≠ Container



Software

Process

Traditional Software Deployment

Docker Image

Container

Container based Software Deployment

The Dockerfile

```
FROM alpine:3.4
MAINTAINER Daniel Menet <daniel.menet@swisstxt.ch>

RUN apk add --no-cache openssh-client tar curl
RUN curl --silent --show-error --fail --location \
      --header "Accept: application/tar+gzip, application/x-gzip, \
      application/octet-stream" -o - \
       "https://caddyserver.com/download/build?os=linux&arch=amd64" \
      | tar --no-same-owner -C /usr/bin/ -xz caddy \
      && chmod 0755 /usr/bin/caddy

COPY Caddyfile /etc/Caddyfile
COPY index.html /srv/index.html

EXPOSE 8008
WORKDIR /srv

ENTRYPOINT ["/usr/bin/caddy", "--conf", "/etc/Caddyfile", "--log", "stdout"]
```

```
docker build -t mycaddy .
```

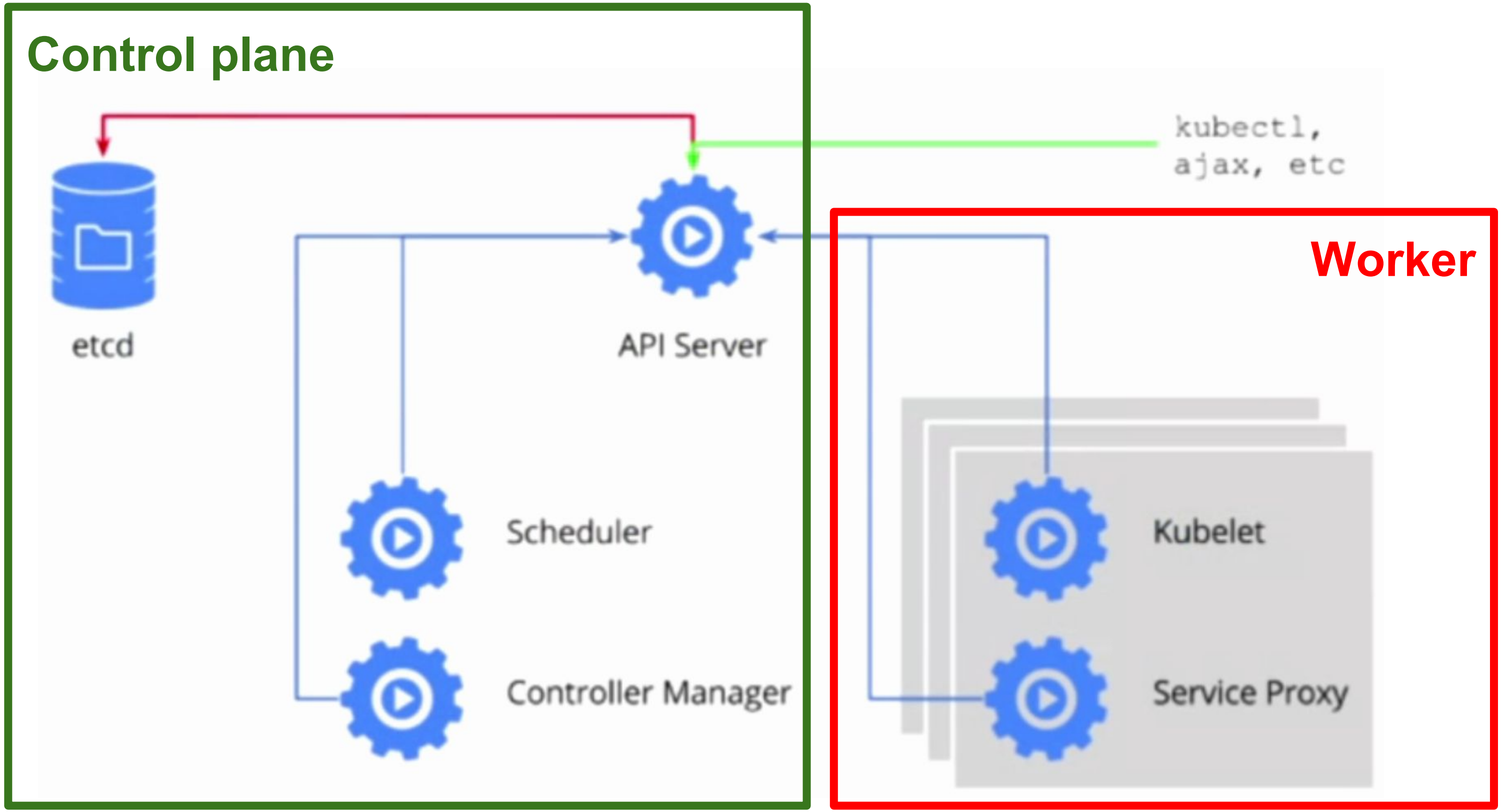
```
docker run -d -p 8008:8008 mycaddy
```

What is Kubernetes?

A container **orchestration** platform:

You describe the state of your system, **Kubernetes establishes** the state you described.

The data center as a computing resource.



Control plane

etcd

API Server

kubectl,
ajax, etc

Worker

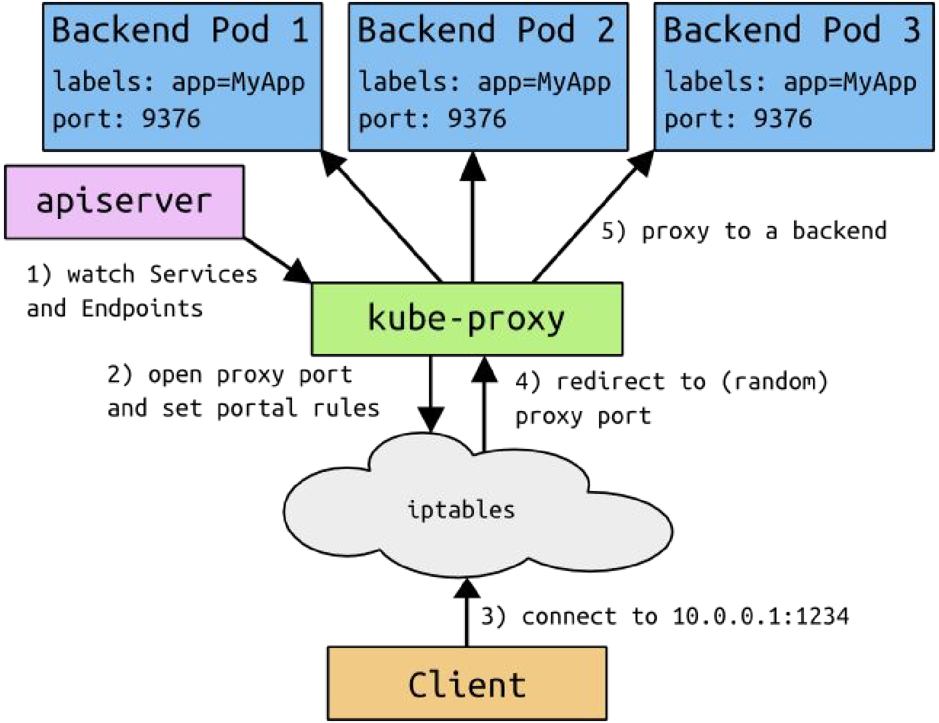
Scheduler

Controller Manager

Kubelet

Service Proxy

Kubernetes Services



Kubernetes Value Added to Docker

➡ Addresses ***Distributed System Problems***

- **Networking** when using multiple docker hosts
- **Service Discovery** is included
- **Availability** can be monitored and managed automatically
- **Deployment** is facilitated
- (Shared) **Persistent Storage** functions are provided

Kubernetes @ SWISS TXT

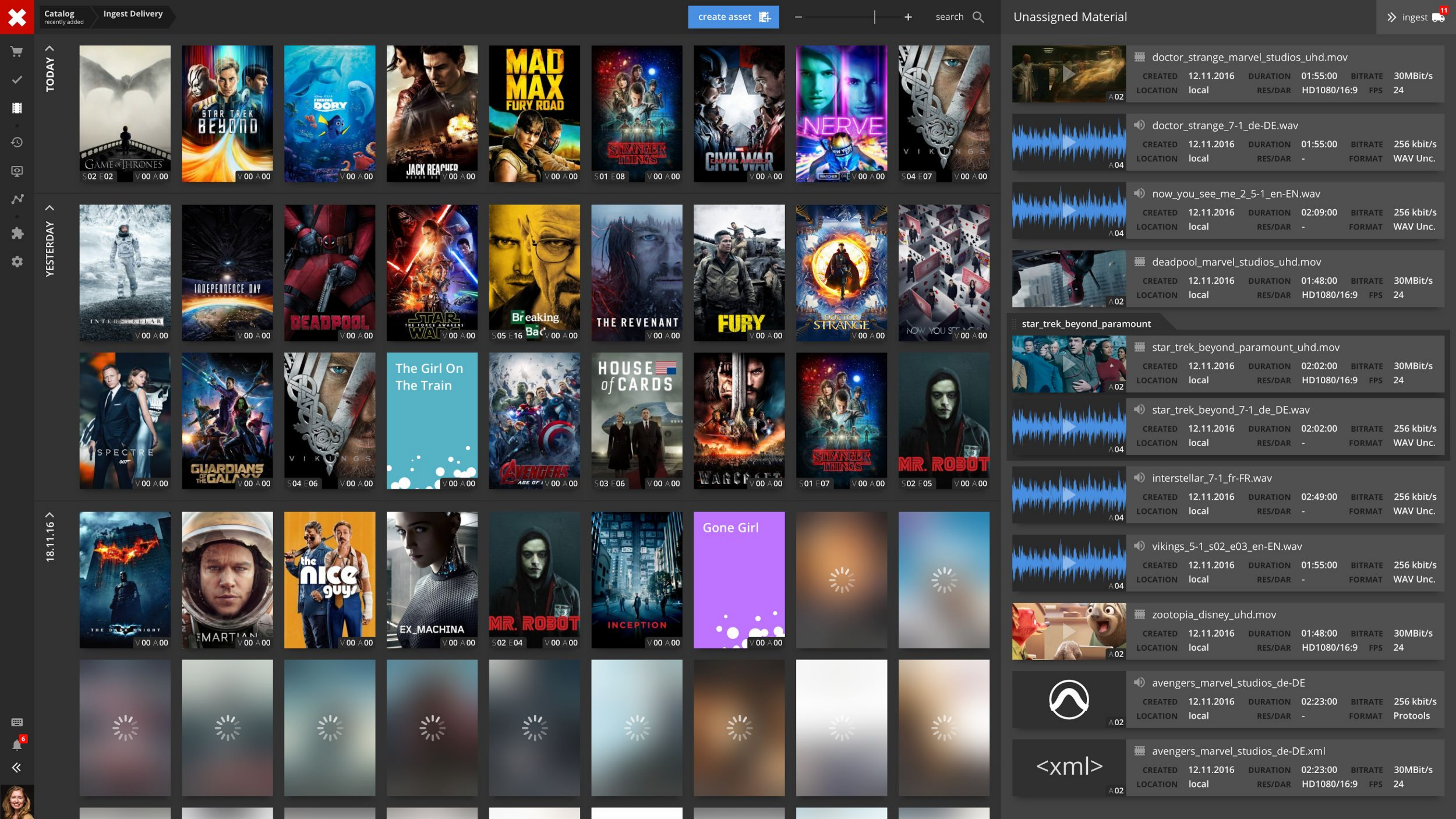
Mediahub

A **workflow driven B2B platform** that allows our customer to **exchange, deliver and master movies** and TV-series. All processes are **automated** to a high degree.

Core functions are:

- Ingest
- QC / Transcoding / etc.
- Delivery
- Preview
- Purchase Order / Work Order / Tasks
- Catalog

[LOGO TO COME]



Unassigned Material

doctor_strange_marvel_studios_uhd.mov
CREATED 12.11.2016 DURATION 01:55:00 BITRATE 30MBit/s
LOCATION local RES/DAR HD1080/16:9 FPS 24
A 02

doctor_strange_7-1_de-DE.wav
CREATED 12.11.2016 DURATION 01:55:00 BITRATE 256 kbit/s
LOCATION local RES/DAR - FORMAT WAV Unc.
A 04

now_you_see_me_2_5-1_en-EN.wav
CREATED 12.11.2016 DURATION 02:09:00 BITRATE 256 kbit/s
LOCATION local RES/DAR - FORMAT WAV Unc.
A 04

deadpool_marvel_studios_uhd.mov
CREATED 12.11.2016 DURATION 01:48:00 BITRATE 30MBit/s
LOCATION local RES/DAR HD1080/16:9 FPS 24
A 02

star_trek_beyond_paramount

star_trek_beyond_paramount_uhd.mov
CREATED 12.11.2016 DURATION 02:02:00 BITRATE 30MBit/s
LOCATION local RES/DAR HD1080/16:9 FPS 24
A 02

star_trek_beyond_7-1_de_DE.wav
CREATED 12.11.2016 DURATION 02:02:00 BITRATE 256 kbit/s
LOCATION local RES/DAR - FORMAT WAV Unc.
A 04

interstellar_7-1_fr-FR.wav
CREATED 12.11.2016 DURATION 02:49:00 BITRATE 256 kbit/s
LOCATION local RES/DAR - FORMAT WAV Unc.
A 04

vikings_5-1_s02_e03_en-EN.wav
CREATED 12.11.2016 DURATION 01:55:00 BITRATE 256 kbit/s
LOCATION local RES/DAR - FORMAT WAV Unc.
A 04

zootopia_disney_uhd.mov
CREATED 12.11.2016 DURATION 01:48:00 BITRATE 30MBit/s
LOCATION local RES/DAR HD1080/16:9 FPS 24
A 02

avengers_marvel_studios_de-DE
CREATED 12.11.2016 DURATION 02:23:00 BITRATE 256 kbit/s
LOCATION local RES/DAR - FORMAT Protools
A 02

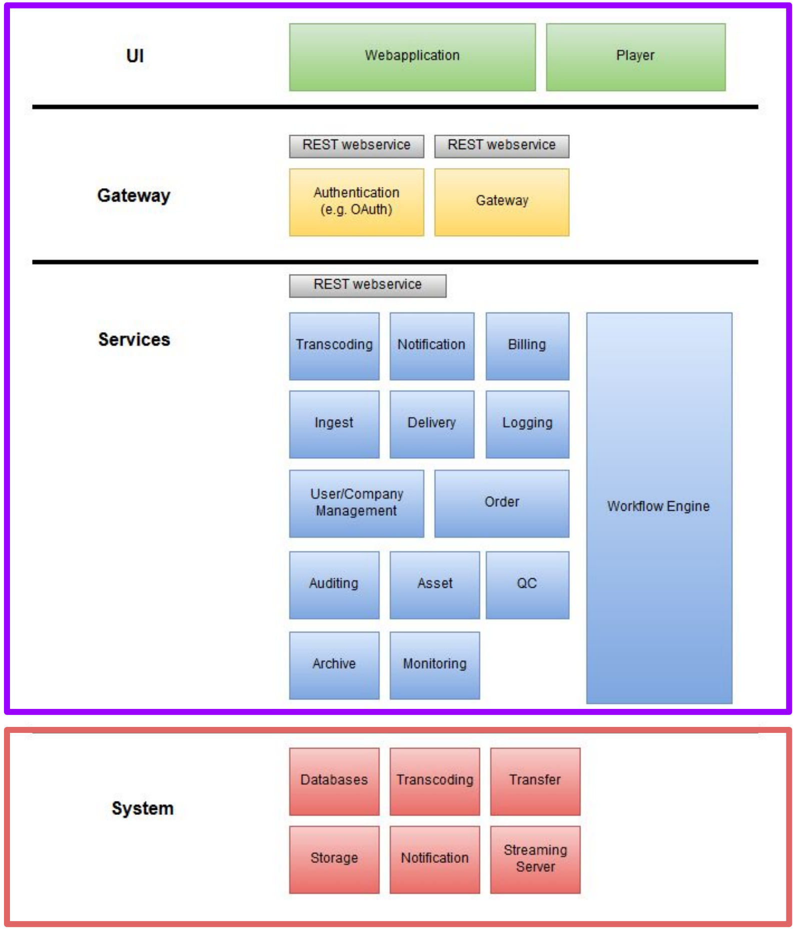
avengers_marvel_studios_de-DE.xml
CREATED 12.11.2016 DURATION 02:23:00 BITRATE 30MBit/s
LOCATION local RES/DAR HD1080/16:9 FPS 24
<xml>
A 02

Architecture

Micro services because ...

- Isolation of errors
- Elimination of long term tech debt
- Less dependencies in development
- Better testability
- Improved scalability

… and some **traditional infrastructure**



Why Containers?

Footprint: No OS overhead

Development/Testing: Running locally or in test is easy and close to production

Deployment: Tiny “contract” between Dev and Ops

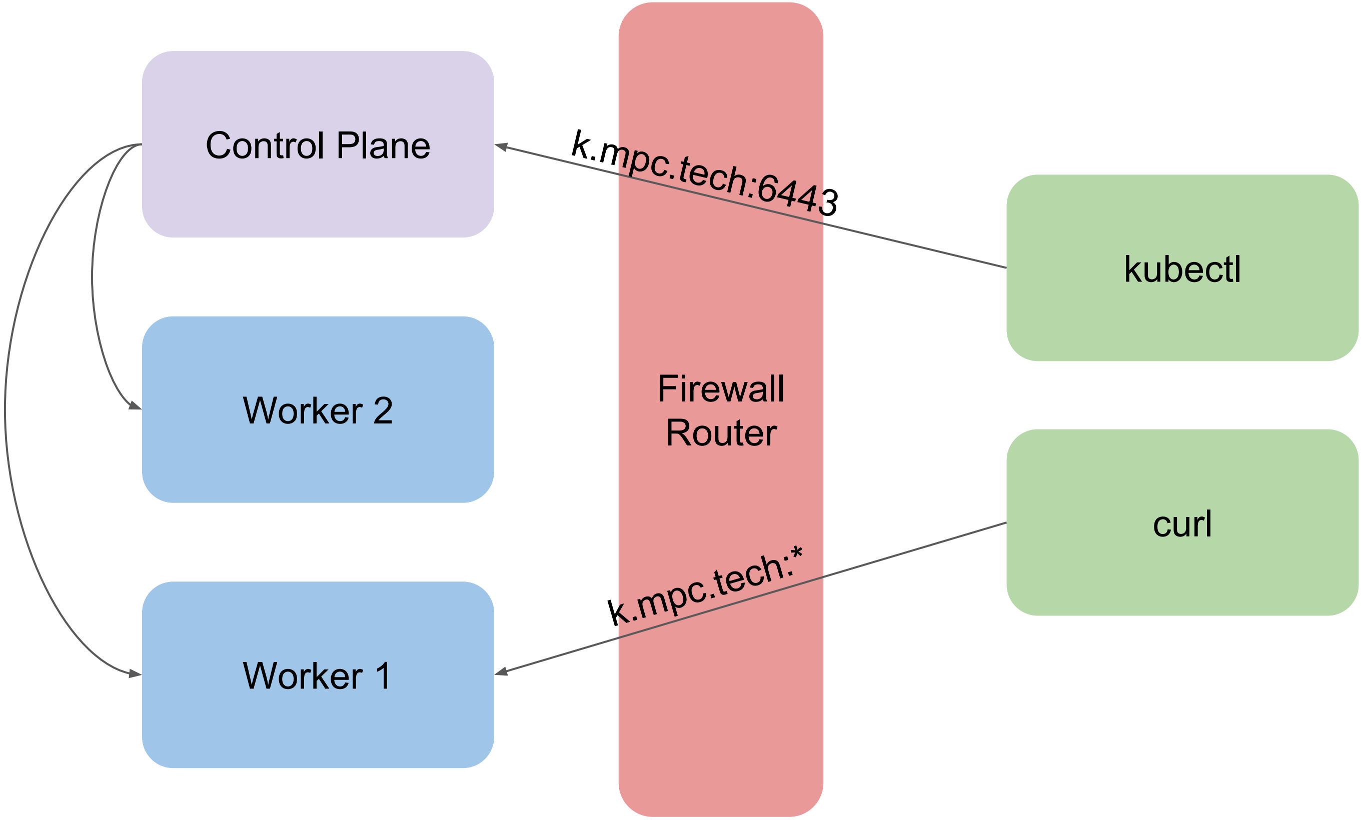
Why Kubernetes?

Because of micro services

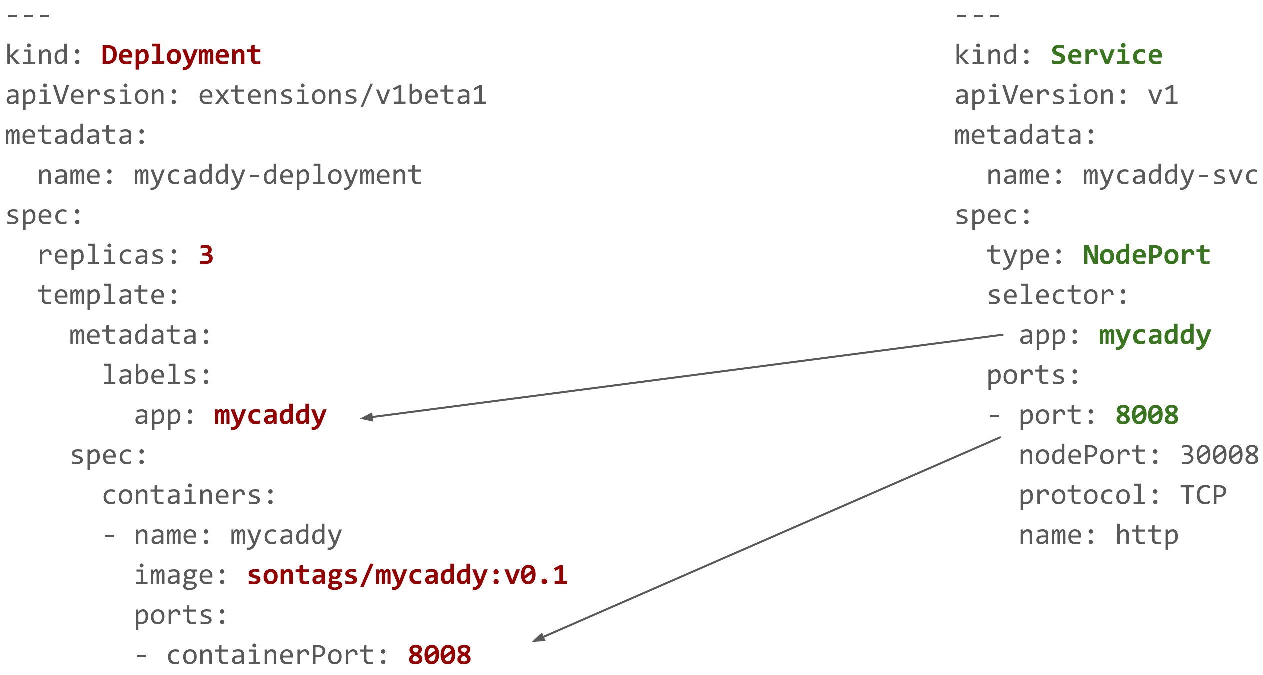
Tooling for CI/CD

Kubernetes Ressources & Applications

Demo Setup


Control Plane
Worker 2
Worker 1
Firewall
Router
k.mpc.tech:6443
k.mpc.tech:*
kubectl
curl

Deployments and Services



```
---
kind: Deployment
apiVersion: extensions/v1beta1
metadata:
  name: mycaddy-deployment
spec:
  replicas: 3
  template:
    metadata:
      labels:
        app: mycaddy
    spec:
      containers:
      - name: mycaddy
        image: sontags/mycaddy:v0.1
        ports:
        - containerPort: 8008
```

```
---
kind: Service
apiVersion: v1
metadata:
  name: mycaddy-svc
spec:
  type: NodePort
  selector:
    app: mycaddy
  ports:
  - port: 8008
    nodePort: 30008
    protocol: TCP
    name: http
```

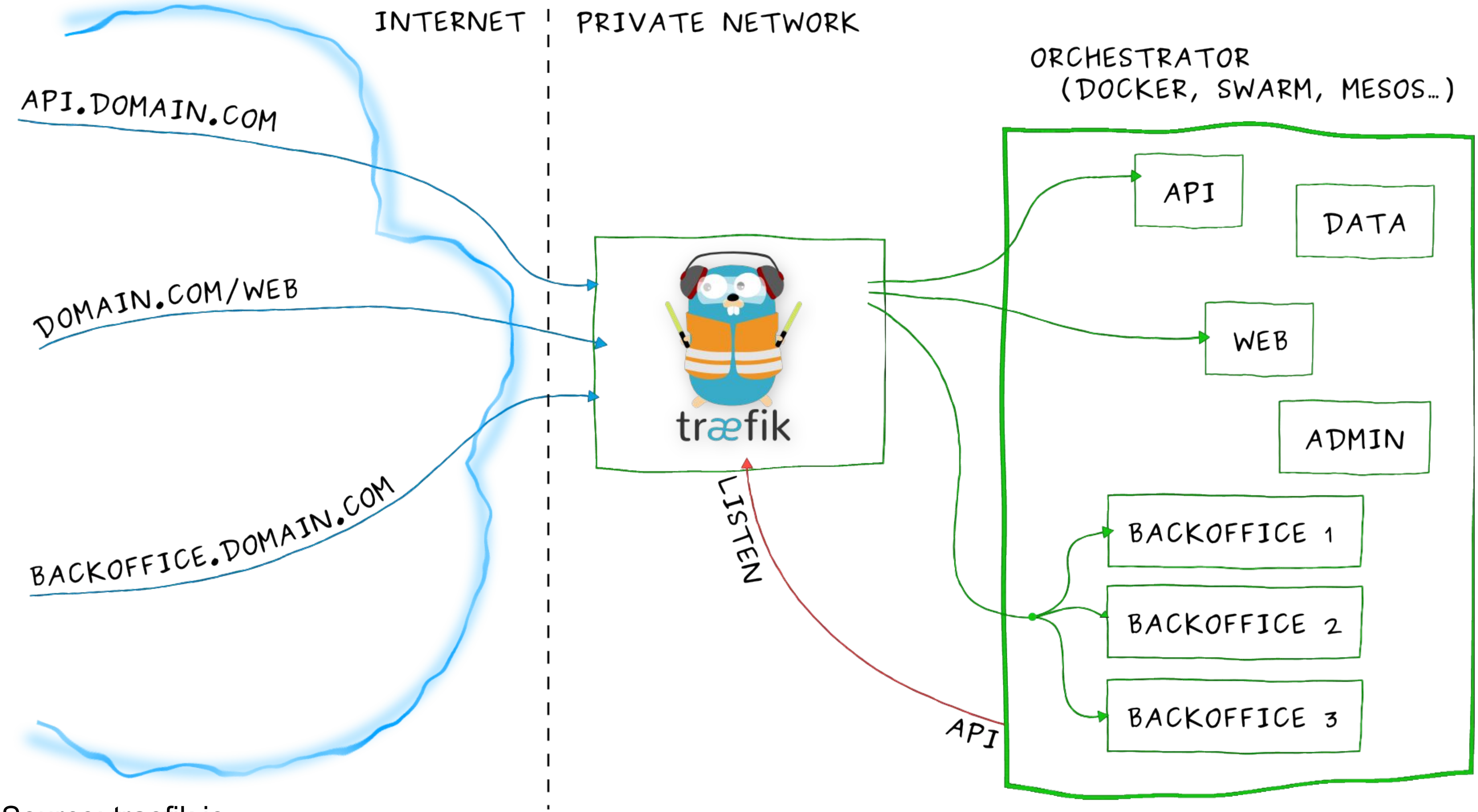
ConfigMap

```
---
kind: ConfigMap
apiVersion: v1
metadata:
  name: traefik-cfg
  labels:
    app: traefik-ingress
data:
  traefik.toml: |
    defaultEntryPoints = ["http"]
    [entryPoints]
      [entryPoints.http]
      address = ":80"
    [web]
    address = ":2727"
    ReadOnly = true
```

DaemonSets

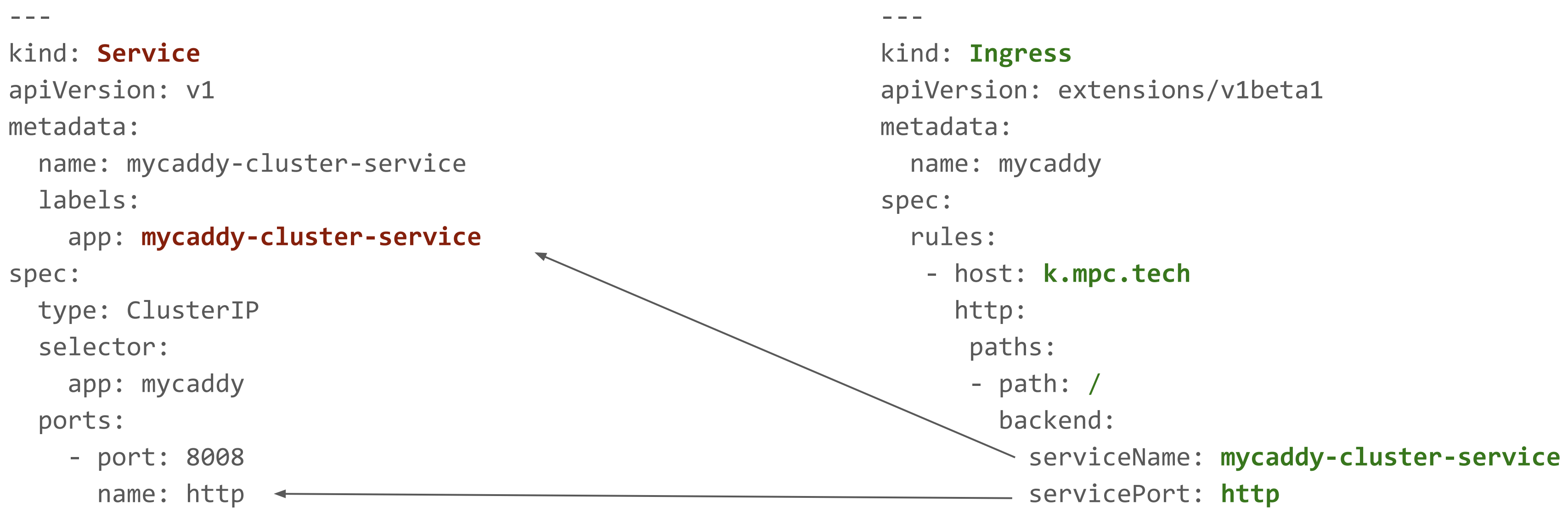
```
---
apiVersion: extensions/v1beta1
kind: DaemonSet
metadata: ...
spec:
  Template: ...
    spec:
      ...
      volumes:
      - name: config
        configMap:
          name: traefik-cfg
      containers:
      - image: traefik
        ...
```

```
volumeMounts:
 - mountPath: "/config"
   name: "config"
 ports:
 - containerPort: 80
   hostPort: 80
 - containerPort: 2727
   hostPort: 2727
 args:
 - --configfile=/config/traefik.toml
 - --kubernetes
```



Source: traefik.io

Ingress



```
---
kind: Service
apiVersion: v1
metadata:
  name: mycaddy-cluster-service
  labels:
    app: mycaddy-cluster-service
spec:
  type: ClusterIP
  selector:
    app: mycaddy
  ports:
    - port: 8008
      name: http
```

```
---
kind: Ingress
apiVersion: extensions/v1beta1
metadata:
  name: mycaddy
spec:
  rules:
   - host: k.mpc.tech
     http:
      paths:
      - path: /
        backend:
          serviceName: mycaddy-cluster-service
          servicePort: http
```

We have seen...

… what **Docker and Kubernetes** are.

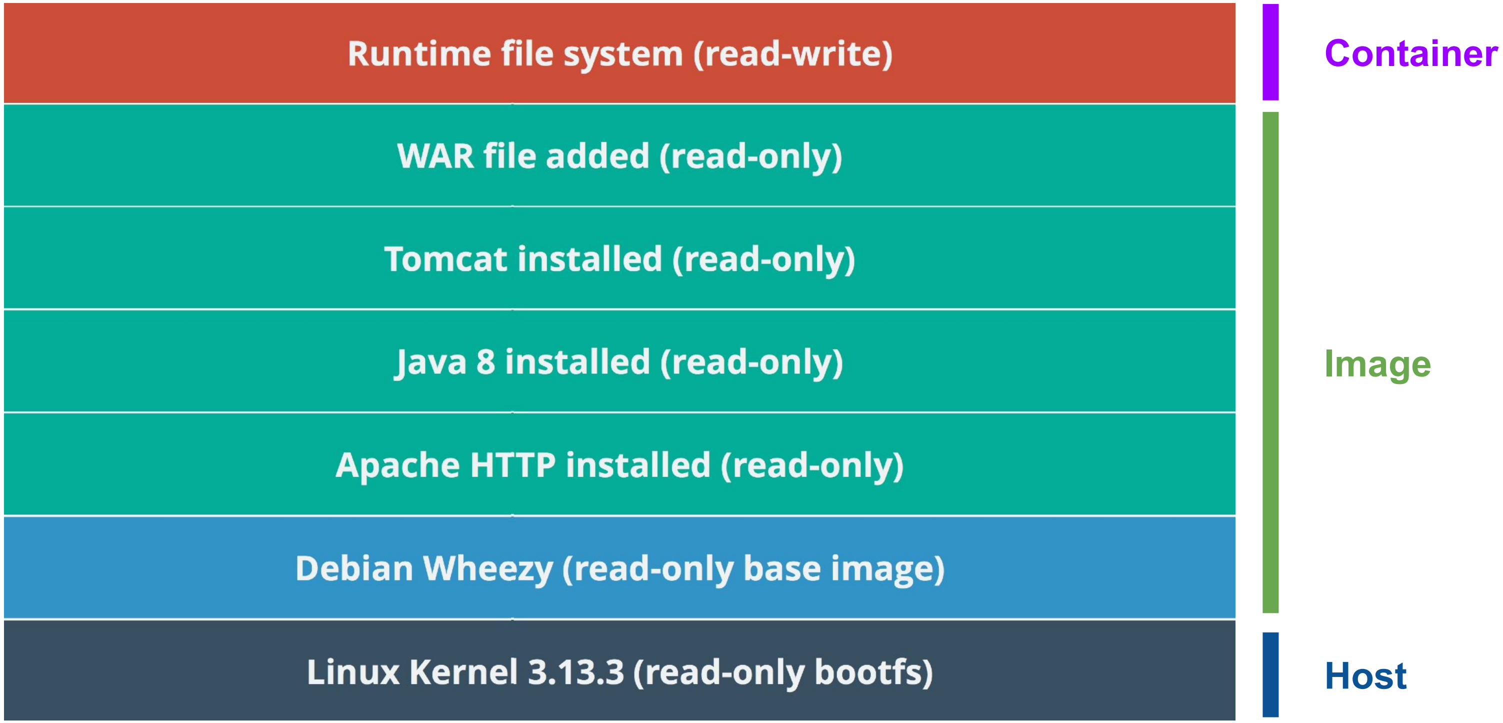
… where they can **help**

… a **use case** at SWISS TXT

… some Kubernetes **resources** and their application

Backup Slides

Docker File System



Runtime file system (read-write)

WAR file added (read-only)

Tomcat installed (read-only)

Java 8 installed (read-only)

Apache HTTP installed (read-only)

Debian Wheezy (read-only base image)

Linux Kernel 3.13.3 (read-only bootfs)

Container

Image

Host