

Container

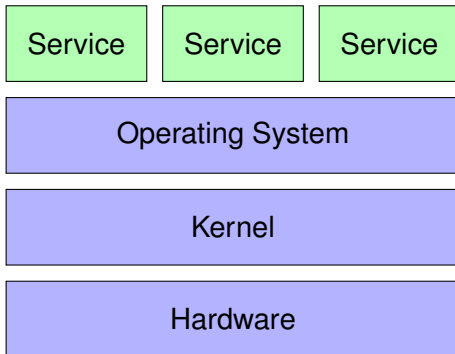
Christoph Handel

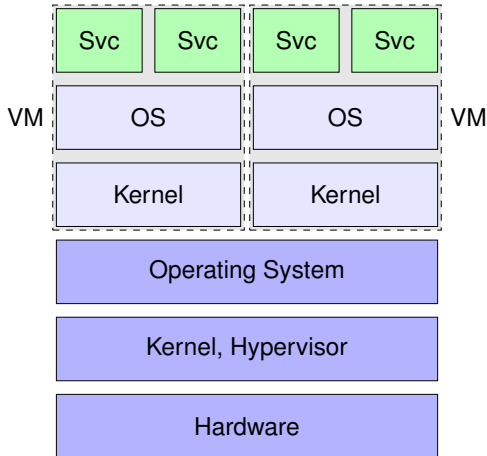
28. Oktober 2019

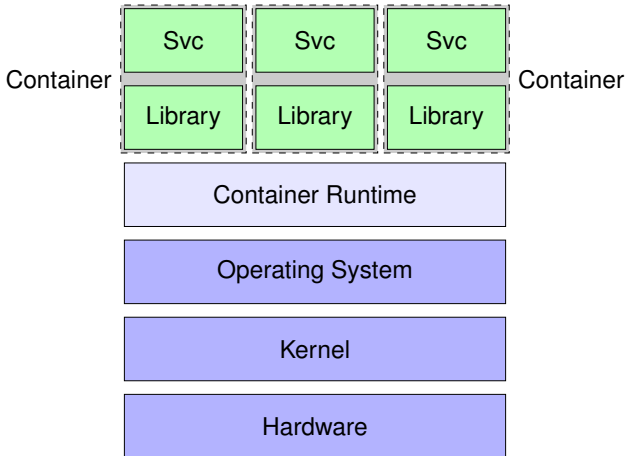
Container What Building

Container
What
Building

@GSI







- ▶ Encapsulate a Service and all it's required libraries.
- ▶ Is isolated from other Processes. Including CPU, Memory, Network and Filesystem.
- ▶ Runs on the same Kernel.
- ▶ Becomes a portable Software Package

Container
What
Building

@GSI

A container is created from an image.
There is more than one way to create an image. Most common is the Dockerfile.

```
FROM aco/jdk11:7.6.1810

RUN curl -f -L -o /opt/app.zip \
    "https://artifacts.acc.gsi.de/.../app-0.1.2.zip"

# unpack
RUN unzip /opt/app.zip -d /opt && rm -f /opt/app.zip

# define executable
CMD /opt/bin/lisa-server-gsi.sh --config=dev
```

- ▶ inherit from another image
- ▶ run build instructions inside image
- ▶ declare command to execute during runtime

```
FROM aco/jdk11:7.6.1810

RUN curl -f -L -o /opt/app.zip \
    "https://artifacts.acc.gsi.de/.../app-0.1.2.zip"

# unpack
RUN unzip /opt/app.zip -d /opt && rm -f /opt/app.zip

# define executable
CMD /opt/bin/lisa-server-gsi.sh --config=dev
```

- ▶ inherit from another image
- ▶ run build instructions inside image
- ▶ declare command to execute during runtime

- ▶ Parameters that are used during the build
version 0.1.2 \Rightarrow ARG
- ▶ Parameters that are used during runtime
config dev \Rightarrow ENV

```
ARG version
FROM aco/jdk11:7.6.1810

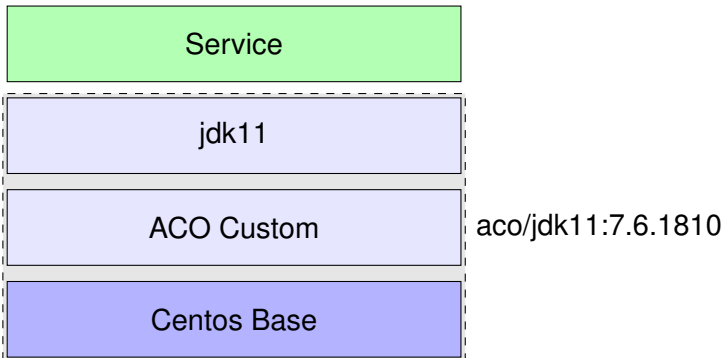
ENV CONFIG dev

RUN curl -f -L -o /opt/app.zip \
    "https://artifacts.acc.gsi.de/.../app-${version}.zip"

RUN unzip /opt/app.zip -d /opt && rm -f /opt/app.zip

CMD /opt/bin/lisa-server-gsi.sh --config=${CONFIG}
```

```
podman build
    --build-arg version=0.1.2
    -t sample/app:0.1.2
```



- ▶ single parent
- ▶ single child
- ▶ identified by hashes including metadata
- ▶ shared

- ▶ the container version of maven artifact repository
- ▶ store layers and metadata
- ▶ proxy, cache, search, ...
- ▶ note: tags are not immutable

- ▶ containers are ephemeral
- ▶ fetch, stack, merge layers
- ▶ start process
- ▶ delete on end of process
- ▶ no data survives restarts

- ▶ not really, there is a storage layer
- ▶ it's complicated
- ▶ ephemeral is the main concept

@GSI

- ▶ kubernetes
- ▶ registry
- ▶ network
- ▶ maybe storage

- ▶ Write Dockerfiles
- ▶ Externalize config
- ▶ Package and Tag Containers
- ▶ Write deployment descriptors
- ▶ Write parameter files
- ▶ Deploy container
- ▶ Update Container, including Security
- ▶ manage resource
- ▶ Figure out why it's not working

- ▶ process/library isolation
- ▶ delivery format
- ▶ rollout concept
- ▶ self-service services

- ▶ running kubernetes is work
- ▶ resource management is developers responsibility
- ▶ run whatever you want
- ▶ updates? what updates?