

acend

Container Technology

Docker

Benj & Dänu, Bern, 3. September 2021

experience knowledge

Agenda

Container Technology Docker

1. Introduction to Containers

1.1. What is a Container?

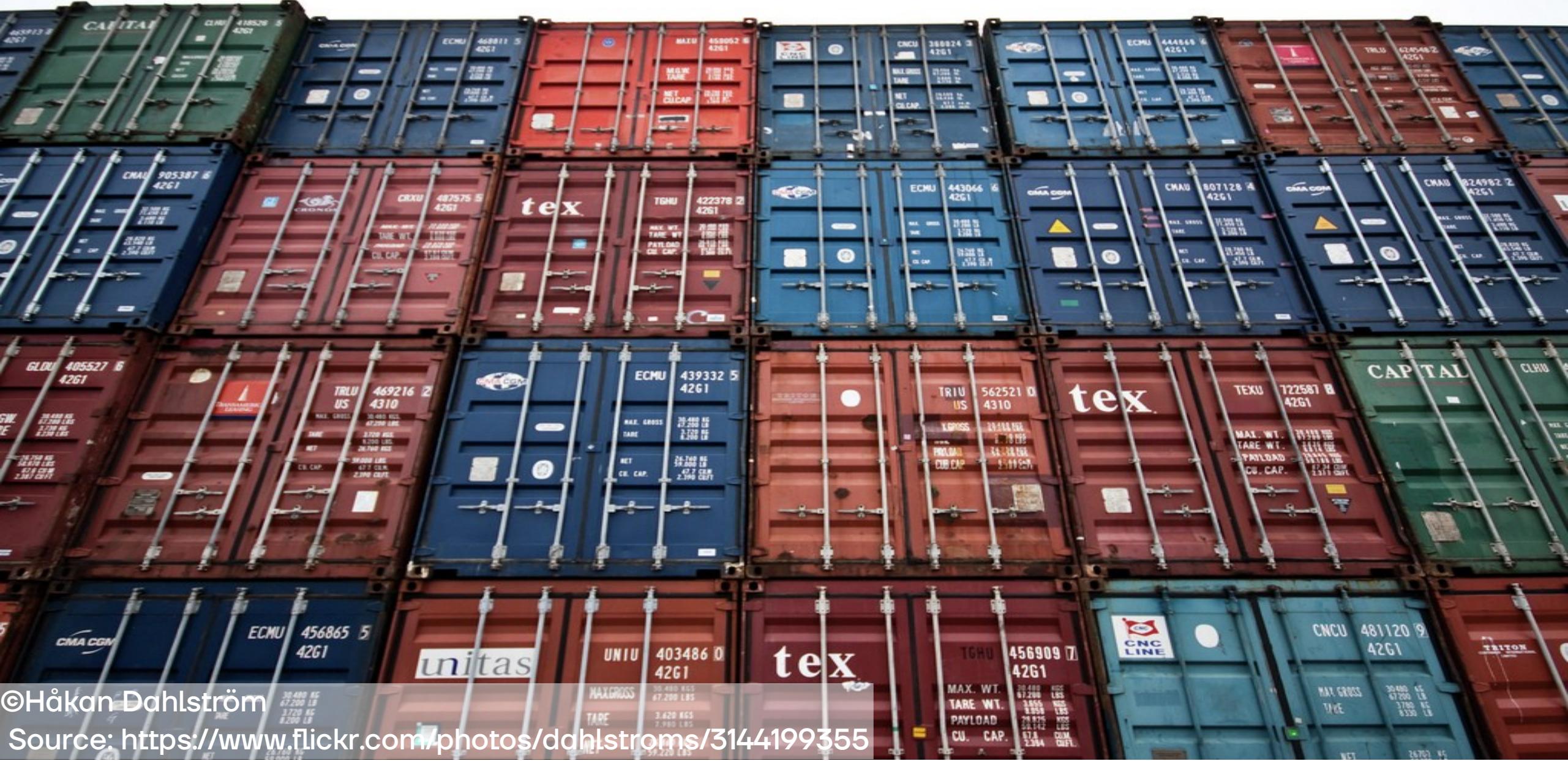
1.2. Examples of Use

1.3. Chances and Challenges

2. TechLab

Introduction to Containers

What is a Container?



©Håkan Dahlström

Source: <https://www.flickr.com/photos/dahlstroms/3144199355>

What is a Container?

Developments in IT

1995



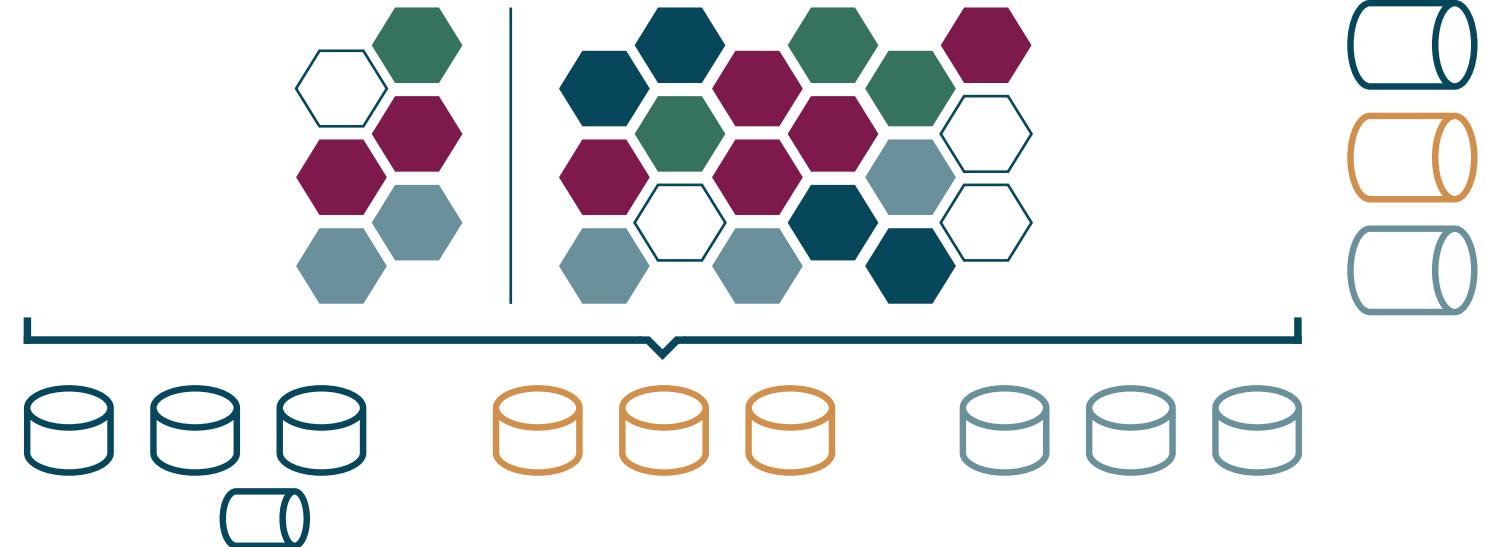
What is a Container?

Developments in IT

1995

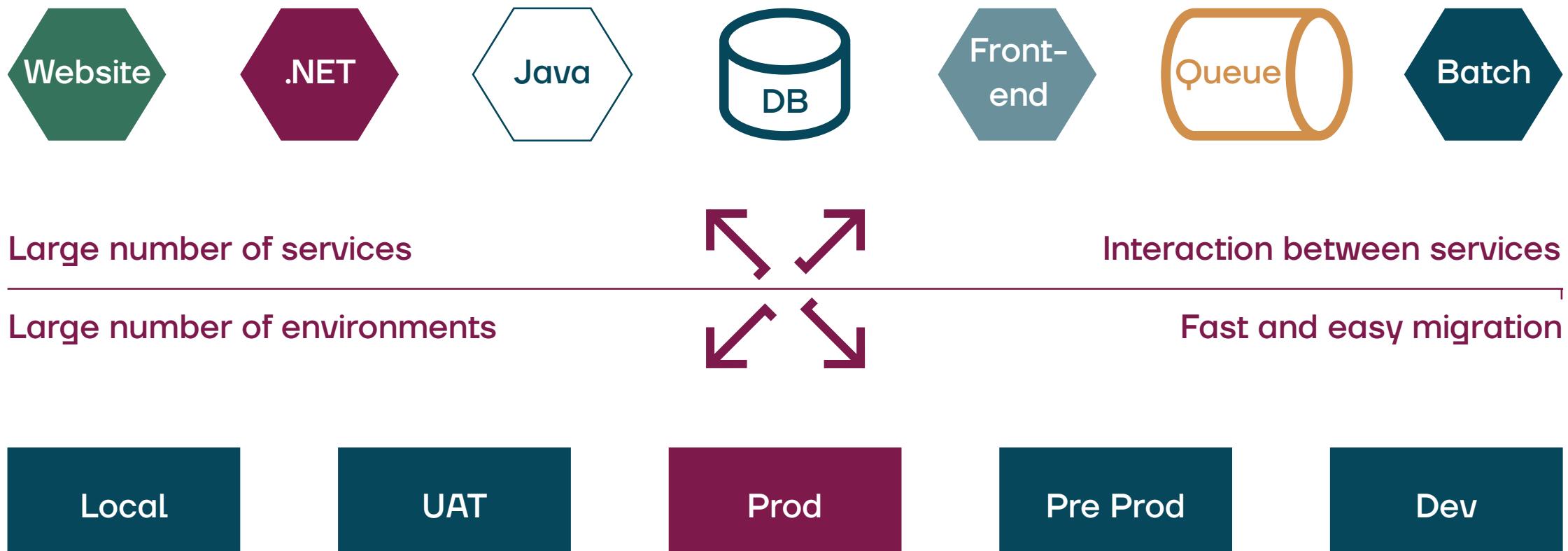


Today



What is a Container?

Plethora of Combinations



What is a Container?

Matrix from Hell

	Dev PC	Dev	Test	Prod	Cloud	Customer server	...
Website	?	?	?	?	?	?	?
Frontend	?	?	?	?	?	?	?
Web-Service	?	?	?	?	?	?	?
Database	?	?	?	?	?	?	?
Queue	?	?	?	?	?	?	?
Application	?	?	?	?	?	?	?

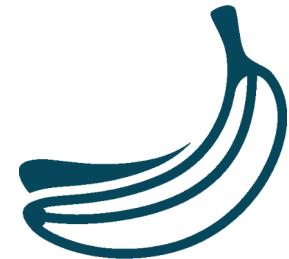
What is a Container?

Goods Traffic before 1960



What is a Container?

Plethora of Combinations

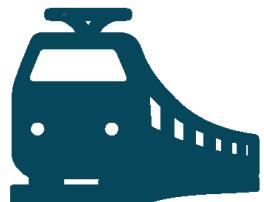
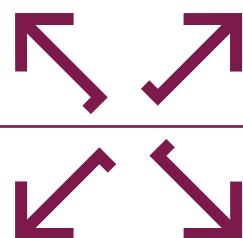


Large number of goods

Large number of routes

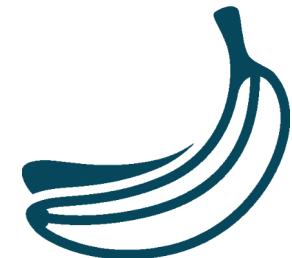
Interaction between goods

Fast and smooth transport



What is a Container?

Solution



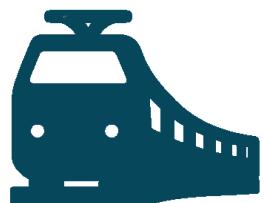
Large number of goods

Large number of routes



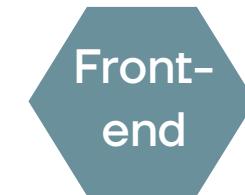
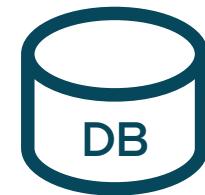
Interaction between goods

Fast and smooth transport



What is a Container?

Application in Containers



Large number of services

Large number of environments



Interaction between services

Fast and easy migration



What is a Container?

Matrix Reloaded

	Dev PC	Dev	Test	Prod	Cloud	Customer server	...
Website							
Frontend							
Web-Service							
Database							
Queue							
Application							

What is a Container?

Purpose

- ¬ Standardized mechanism for building, deploying and operating applications
- ¬ Isolation of applications
- ¬ Clear definition of interfaces between application and platform
- ¬ Potential to unify workflows
- ¬ Dev and test environments analog to production





What is a Container?

Hello World Docker Example

```
docker run fedora-minimal /bin/echo "Hello world"
```


What is a Container?

What happen in the background?

- Container start
- Allocation of filesystem
- Mount of a read/write filesystem layer
- Attachment of networking layer
- Execution of `echo` command
- Output to my console
- Container stop

What is a Container?

<1s

What is a Container?

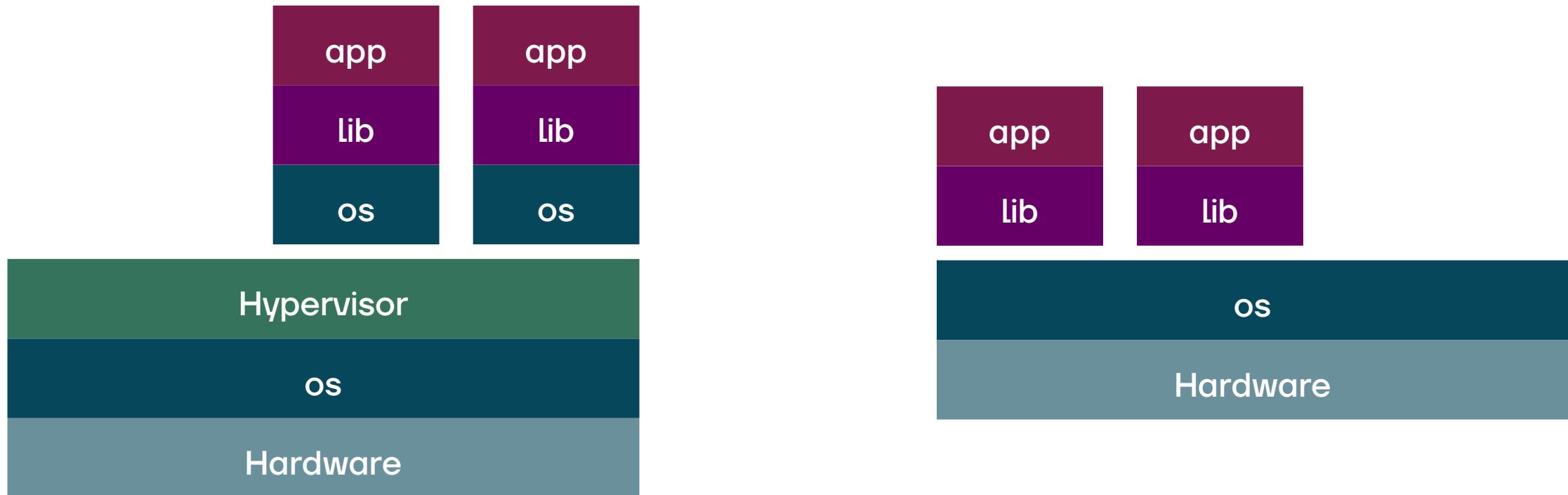
What is a Container?

Nothing new!

LXC, VServer, FreeBSD Jails,
Google...

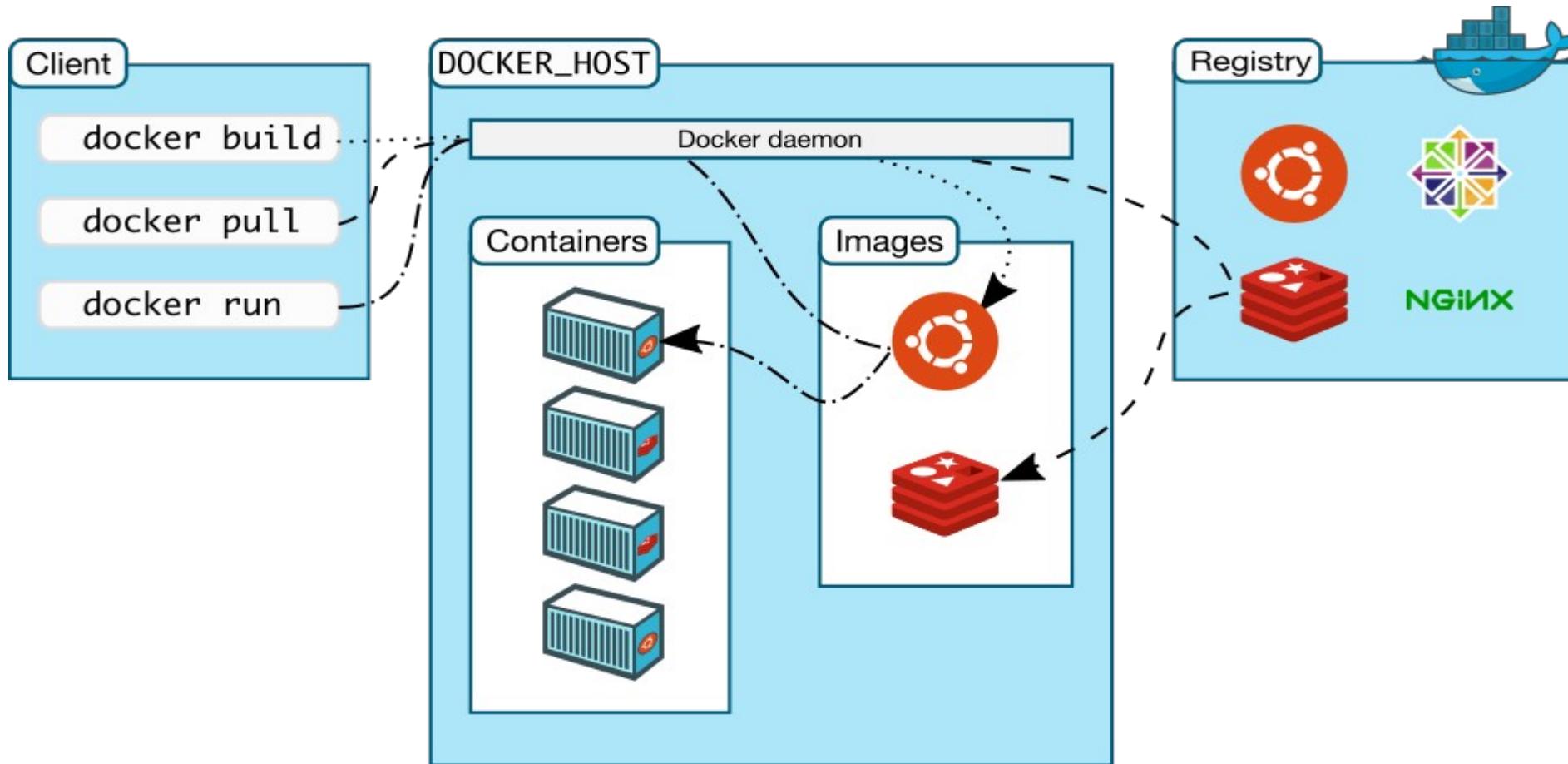
What is a Container?

Virtualization vs. Containerization



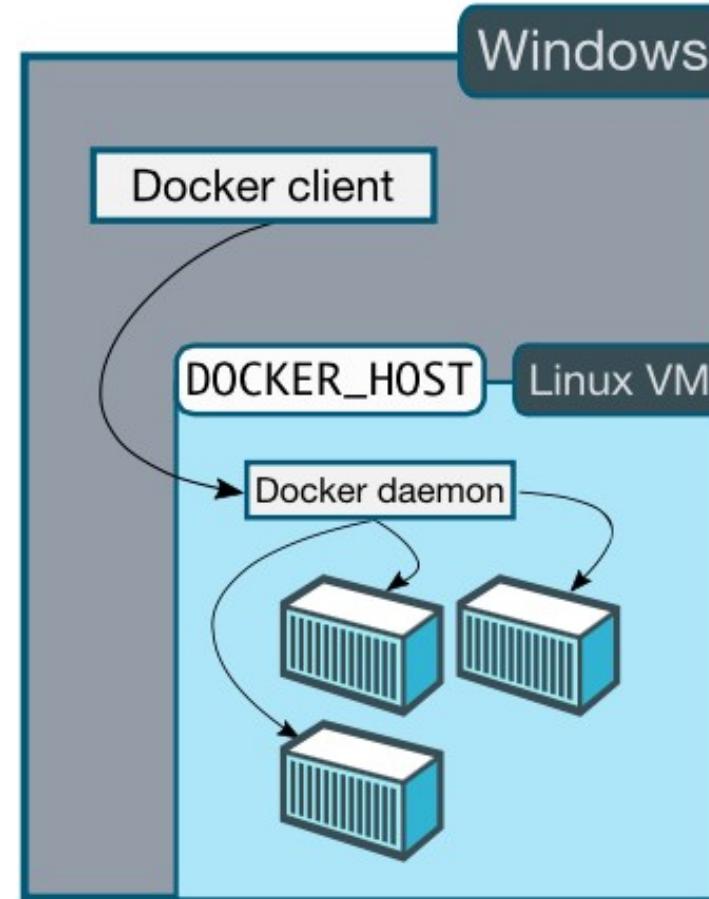
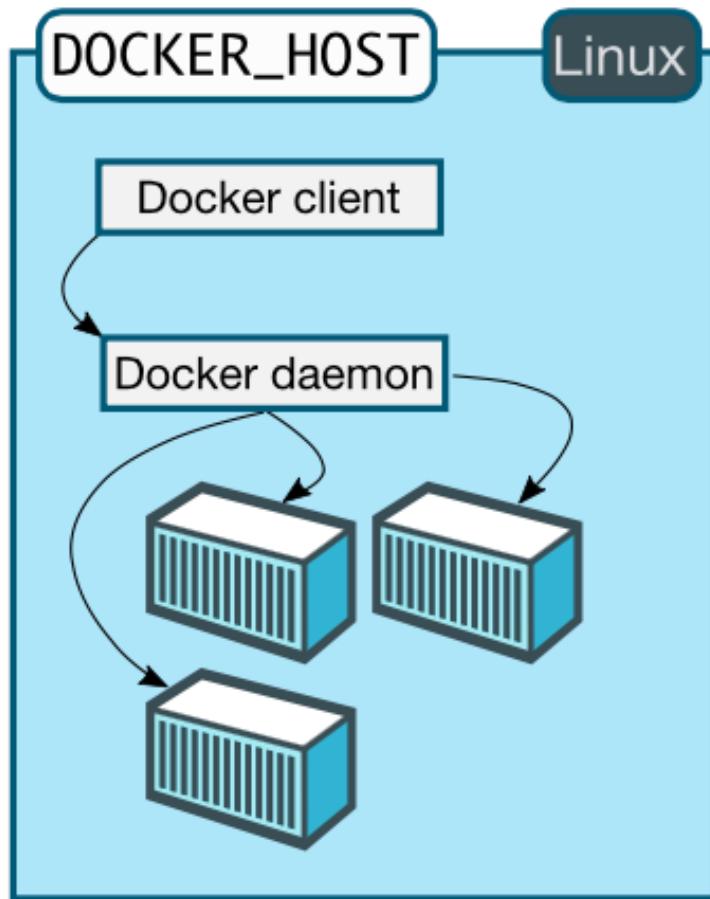
What is a Container?

Docker Architecture



What is a Container?

Docker Architecture on Windows



What is a Container?

Docker Hub

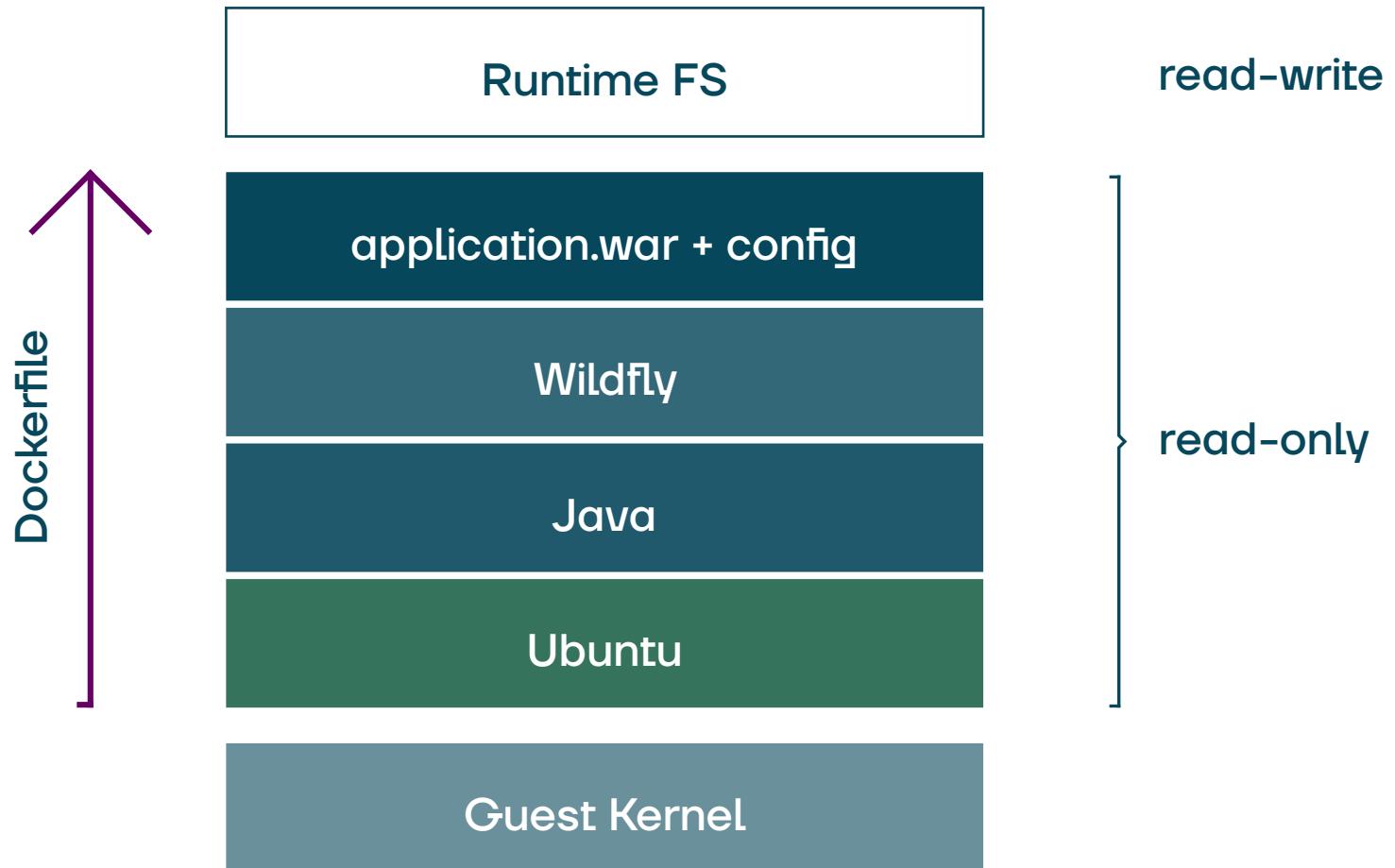
- ¬ Public registry
- ¬ Lots of (official) container images
- ¬ Build platform for your own images
- ¬ Auto build and integration into GitHub
- ¬ ...

What is a Container?

Behind the scenes

- Docker is implemented in GO
- Namespaces (visibility)
- Control Groups (resources)
- Union File System

What is a Container?



What is a Container?

Container Concepts

- ¬ Containers are immutable
- ¬ Update a container by replacing it
 - ¬ applies to applications and system patches
- ¬ No local filesystem for data
- ¬ Persistent storage

What is a Container?

Dockerfile (1)

```
FROM centos7
MAINTAINER Thomas Philipona <philipona@puzzle.ch>
EXPOSE 8080

# Install Java
RUN INSTALL_PKGS="tar unzip bc which lsof java-1.8.0-openjdk java-1.8.0-openjdk-devel" && \
    yum install -y $INSTALL_PKGS && \
    yum clean all -y
```

What is a Container?

Dockerfile (2)

...

```
USER 1001
```

```
# Add application source to image
```

```
ADD . /opt/app-root/src/
```

```
# build application and copy to correct location
```

```
RUN sh /opt/app-root/src/gradlew build && \  
 cp -a /opt/app-root/src/build/libs/springboots2idemo*.jar \  
 /opt/app-root/application.jar
```

What is a Container?

Dockerfile (2)

...

```
CMD ["java", "-Xmx64m", "-Xss1024k", "-jar",
      "/opt/app-root/application.jar"]
```

What is a Container?

Dockerfile Best Practices (1/2)

- Containers should be stateless, state should only exist in mounted volumes
- One process per container
- Keep number of layers small
- Use a `.dockerignore` file
- Log to stdout → let platform handle aggregation and rotation

What is a Container?

Dockerfile Best Practices (2/2)

- Only use trusted base images
- Automatic rebuild → security patches
- Sort multiline arguments:

```
RUN apt-get update && apt-get install -y \
    bazaar \
    cvs \
    git \
    mercurial \
    subversion
```

Introduction to Containers

Examples of Use

Examples of Use

LaTex Rendering

Installation of a LaTex rendering infrastructure can be a pain (depending on OS)...

Put it into a container!

Standardization, traceability, installation documentation, fast

```
docker run pdflatex -output-directory output /input.tex
```

Examples of Use

Build infrastructure

Java, JavaScript, Ruby on Rails, Node, ...

Build tools in different versions for varying projects have to be installed...

Put it in a container (-:

Define the exact development environment for each application.

Reusable, fast, isolated.

Examples of Use

Java EE application with WildFly

Add your .war file and configuration into the image and let's go!

```
FROM jboss/wildfly

ADD app-web.war /opt/jboss/wildfly/standalone/deployments/
ADD standalone.xml /opt/jboss/wildfly/standalone/configuration/
```

Examples of Use

And so on...

Introduction to Containers

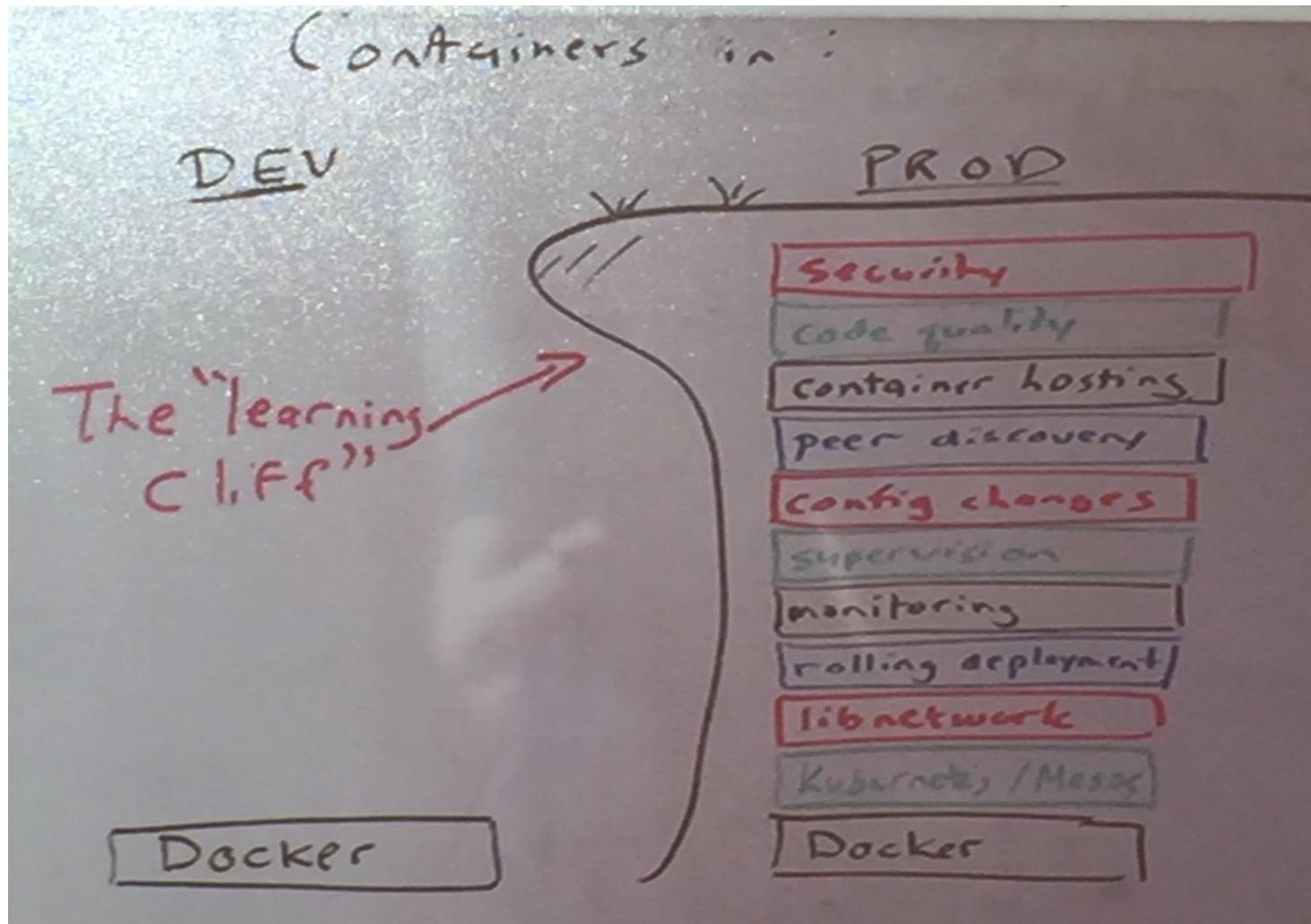
Chances and Challenges

Chances and Challenges

Advantages

- ¬ Lightweight and fast
- ¬ Standardized
- ¬ Easy to use and extend
- ¬ Large community
- ¬ Lots of available images
- ¬ "Works on my machine" was yesterday

Chances and Challenges

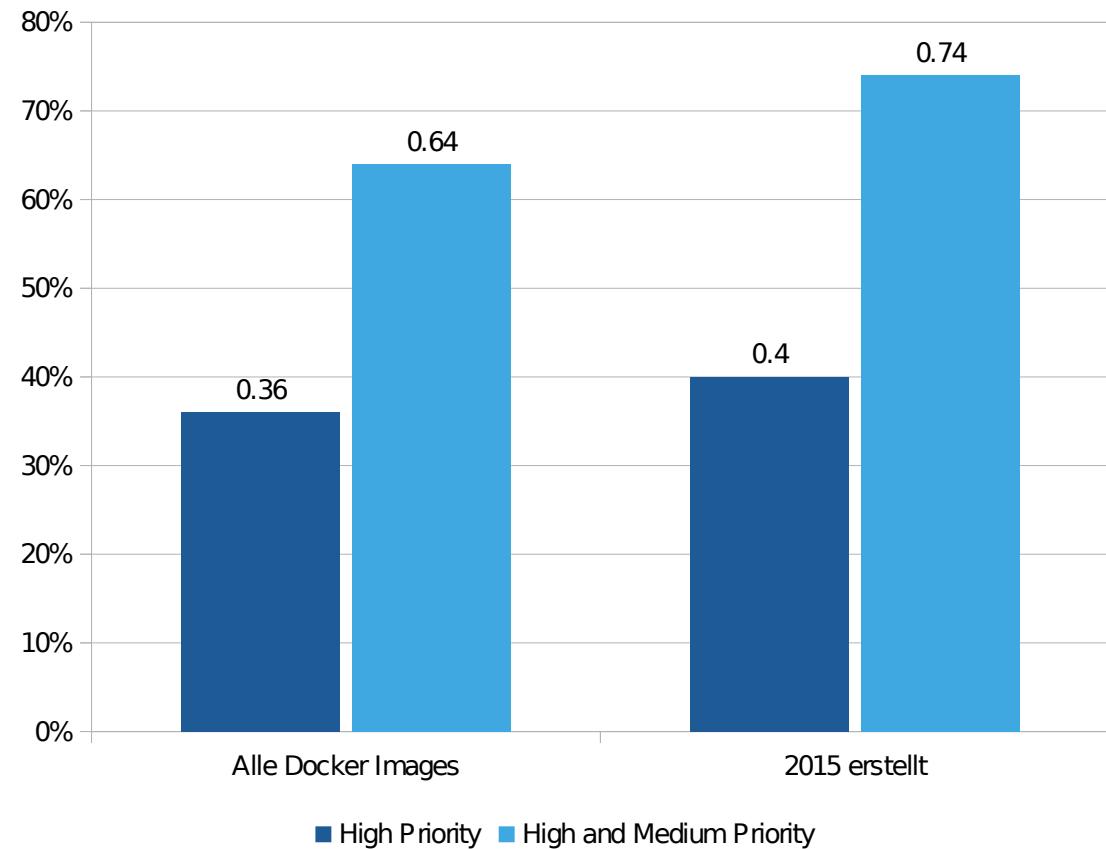


Chances and Challenges

Challenges: Security

"Over 30% of Official Images in Docker Hub Contain High Priority Security Vulnerabilities":

- ShellShock (Bash)
- Heartbleed (OpenSSL)
- Poodle (OpenSSL)
- ...



Source: <http://www.banyanops.com/blog/analyzing-docker-hub/>
May 2015: Jayanth Gummaraju, Tarun Desikan and Yoshio Turner

Chances and Challenges

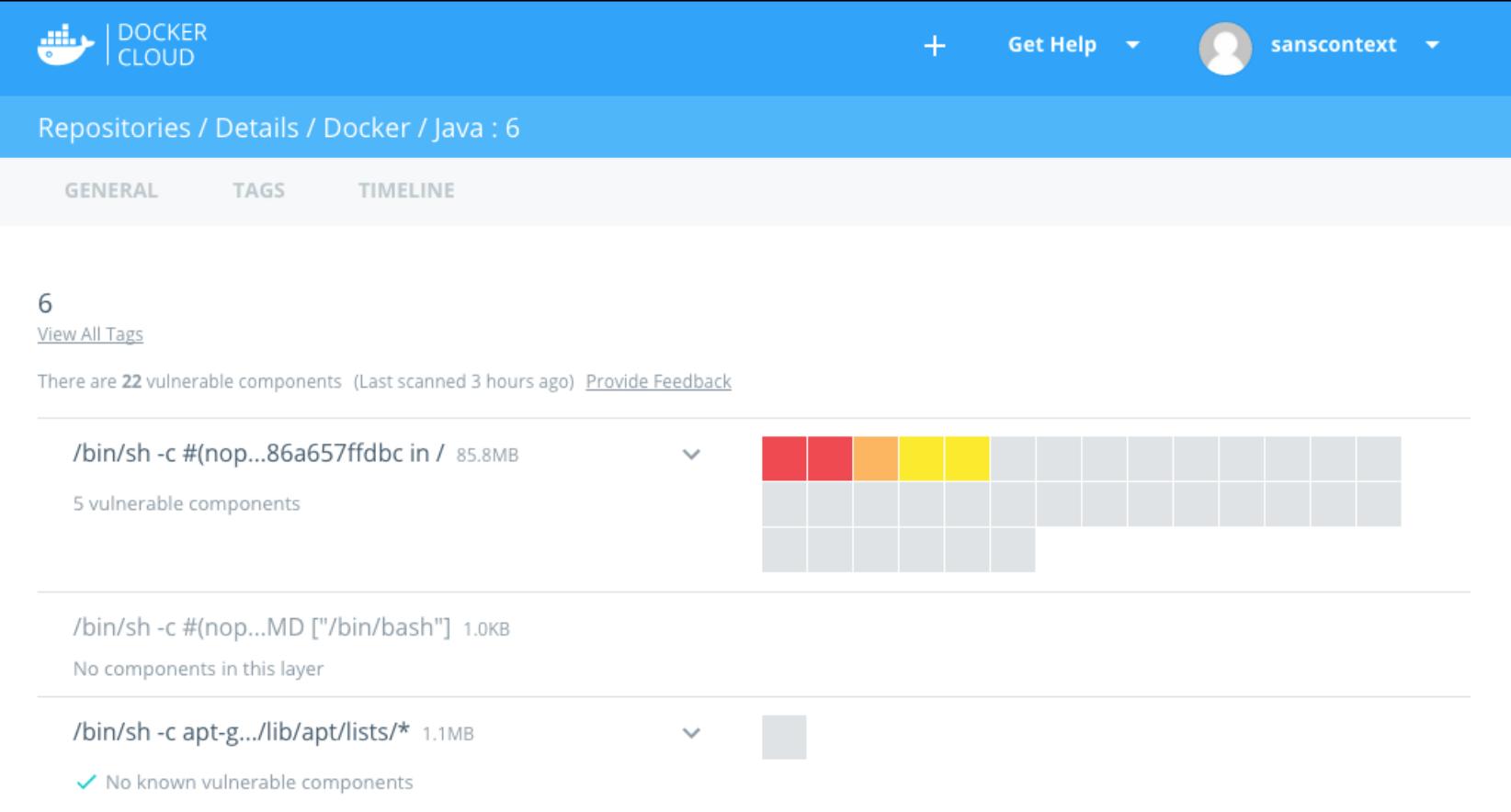
Challenges: Security

- Treat processes as if they were running on the host
- Do not let processes run as root
- Only open necessary ports

Sources: <https://opensource.com/business/14/7/docker-security-selinux>
<http://www.pro-linux.de/news/1/22315/docker-erh%C3%B6ht-die-sicherheit.html>

Chances and Challenges

Docker Security Scanning



The screenshot shows the Docker Cloud interface for security scanning. At the top, there's a blue header bar with the Docker Cloud logo, a '+' button, 'Get Help', and a user profile for 'sanscontext'. Below the header, the URL 'Repositories / Details / Docker / Java : 6' is visible. The main content area has tabs for 'GENERAL', 'TAGS', and 'TIMELINE', with 'GENERAL' selected. It displays a count of 6 tags and a link to 'View All Tags'. A message indicates 22 vulnerable components were found in the last scan (3 hours ago) with a 'Provide Feedback' link. The interface lists three layers: the first layer contains 5 vulnerable components (red, orange, yellow squares), while the second and third layers have no known vulnerabilities (grey squares). A note at the bottom states 'No known vulnerable components'.

GENERAL TAGS TIMELINE

6
[View All Tags](#)

There are 22 vulnerable components (Last scanned 3 hours ago) [Provide Feedback](#)

/bin/sh -c #(nop...86a657ffdbc in / 85.8MB

5 vulnerable components

/bin/sh -c #(nop...MD ["/bin/bash"] 1.0KB

No components in this layer

/bin/sh -c apt-g.../lib/apt/lists/* 1.1MB

✓ No known vulnerable components

Chances and Challenges

Challenges: Operations

Storage

Lifecycle

Containers e Infrastrucuture

Backup

Monitoring

Network

Log Server

Databases

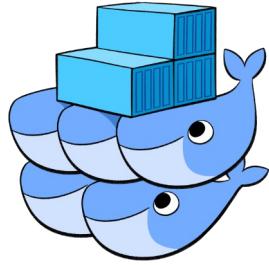
Jenkins

Mail Server



Chances and Challenges

Deployment and Orchestration



MESOS



OPENSIFT



THX]

experience knowledge