A Glympse on Containers

Tools and Infra

Christian Felder

MLZ is a cooperation between
JÜLICH Forschungszentrum
hereon
Technical University of Munich

Containers

Who has heard about containers?

Containers

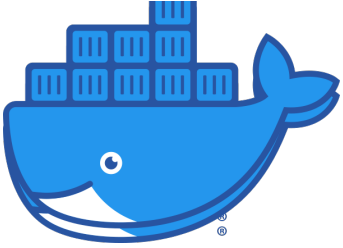
Containers

Properties

- Standardized, well-defined interface
- Easily combinable w/ other containers
- Extendable in the future
- Easy to move somewhere else

Containers

Who has heard about at least one of these tools?

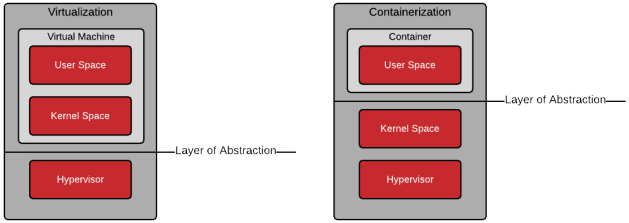


Developers perspective

containers

- Reproducible development environment
  - Build using e.g. `Container-/Dockerfile`
    ⇒ Resulting image captures all dependencies
  - No conflicts w/ installed libraries (namespace)
- Execution *mostly* independent of the underlying operating system
- Easy to share w/ other developers
- Large registry of existing tools available
- Hassle-free deployment from local machine to production

Containers vs. Virtual Machines



- VMs running their own kernel and user space
- Containers are sharing the host's kernel, but may run on their own/shared kernel namespaces
- VMs evolve over time and are often not configured/build in a reproducible way

redhat.com/en/blog/architecting-containers-part-2-why-user-space-matters

Dockerfile

- Description of how to build a container image
- Format: `COMMAND arguments`
- Images are made of Layers
- 1 Cmd == 1 Layer (almost)
- Layers are cached between builds
- Lower layers should be the more stable ones

Dockerfile Keywords: FROM

- Base for the image
- Most distributions/languages available:
  - Ubuntu, Debian, Fedora, etc.
  - Python, Node.js, gcc, etc.
- (advanced) Multi-stage builds

```
FROM ubuntu:24.04
FROM python:3.11
FROM scratch
```

Dockerfile Keywords: RUN

- Runs a command
- Adds a new layer

```
RUN dnf -y install [...]
RUN dnf -y clean all
```

```
RUN dnf -y install [...] && \
    dnf -y clean all
```

Dockerfile Keywords: ADD & COPY

- Copy files from host or earlier build stages
- Adds a new layer
- ADD more powerful (can fetch remote resources)

```
COPY myfile.txt /path/in/container
ADD https://example.com/ex.zip \
    /usr/src/example/
```

Dockerfile Keywords: EXPOSE

- Documentation for which ports are opened
- Adds (empty) layer

```
EXPOSE 80
EXPOSE 12131/udp
```

Dockerfile Keywords: CMD & ENTRYPOINT

- `CMD` sets the default command (overridden by shell args)
- `ENTRYPOINT` specifies executable to run on container start (default is `/bin/sh -c`)
- should have at least one of the two

```
ENTRYPOINT ["mycmd"]
CMD ["--help"]
```

```
docker run imgname \
    https://example.com
```

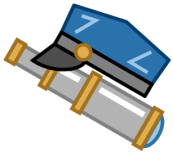
Dockerfile Keywords

- many more (ENV, LABEL, VOLUME, WORKDIR)
- some of them will be in the demo
- have a look at docs.docker.com/reference/dockerfile

Container Tools

podman

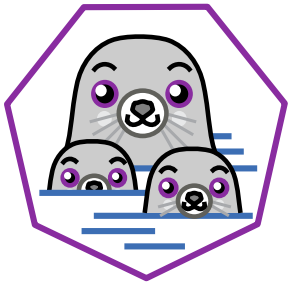
buildah



skopeo

podman

- Container engine, similar to Docker, CRI-O, containerd
- **Daemonless**
- Non-privileged (**rootless**) containers
- privileged (root) containers
- **O**pen **C**ontainer **I**nitiative compliant
  - Container Images
  - Containers
- Relies on OCI compliant container runtime
  - runc (Go), crun (C)
  - Kata Containers
- CLI drop-in replacement for docker (`alias docker=podman`)



buildah

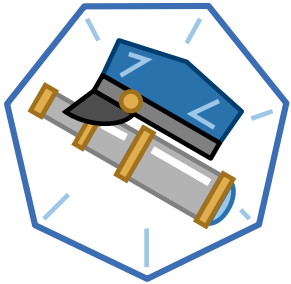
- Building OCI container images
- **Daemonless**
- Replicates all of the commands found in a `Dockerfile`
  - Allows building images w/ and **w/o** `Dockerfiles`
  - Flexibility of integrating other scripting languages in the build process
  - Does **not** require root privileges
- e.g. `buildah unshare ./build.sh`

buildah (build.sh)

```
#!/bin/sh
set -eu
container=$(buildah from registry.jcns.frm2.tum.de/kustomize-box)
scratchmnt=$(buildah mount $container)

dnf -y install --installroot $scratchmnt --releasever 8 --setopt=... \
    git

curl -sf https://.../install_kustomize.sh  sh -s $scratchmnt

buildah config --cmd /usr/bin/bash $container
buildah commit --rm --squash $container builder
```

skopeo

- CLI to perform various operations on container images and repositories
- **Daemonless**
- Supports OCI images as well as Docker v2 images
- Inspect remote image
- Copy image from and to various storage mechanisms
  - e.g. copy image from registry A to registry B
- Delete image from image repository
- Syncing image repositories (e.g. for air-gapped deployments)



Large Scale Deployments

Kubernetes

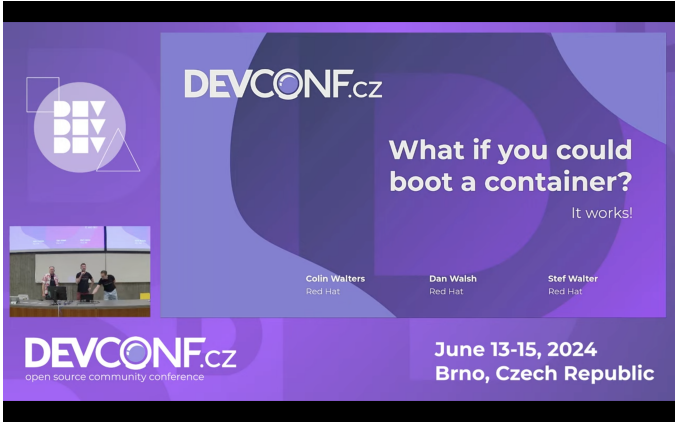
- Container orchestration platform, automating:
  - deployment
  - vertical- and horizontal scaling
  - management of containerized applications
- Declarative configuration using Manifests described in `yaml`
- self-healing: reconcile current state vs. configuration
- Platform for building platforms
  - Extendable API using **C**ustom **R**essource **D**efinitions
  - operator-sdk

Red Hat OpenShift / Okd

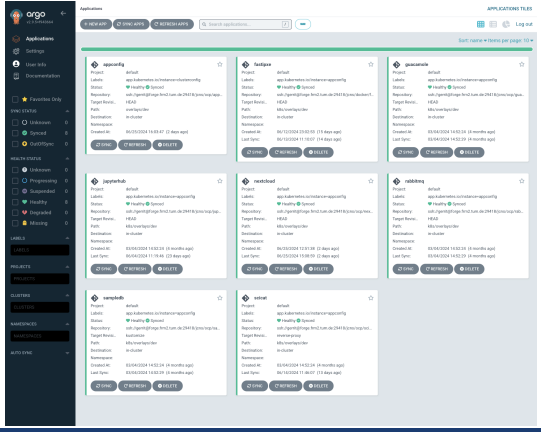
- Enterprise Distribution from Red Hat based on Kubernetes (K8s)
- Upstream open source project: Okd
- `cri-o`, container runtime interface, OCI compliant
- OpenShift/Okd, uses arbitrary user ids in containers (by default)
- `oc` and `kubectl`, command line client

Demo

Bootable containers (bootc)



Thanks for your attention



Get invovled! We're hiring!

Reaching out
Christian Felder M. Sc.
c.felder@fz-juelich.de
+49 - 89 158 860 773