Chair of Computer Architecture and Parallel Systems
TUM School of Computation, Information and Technology
Technical University of Munich

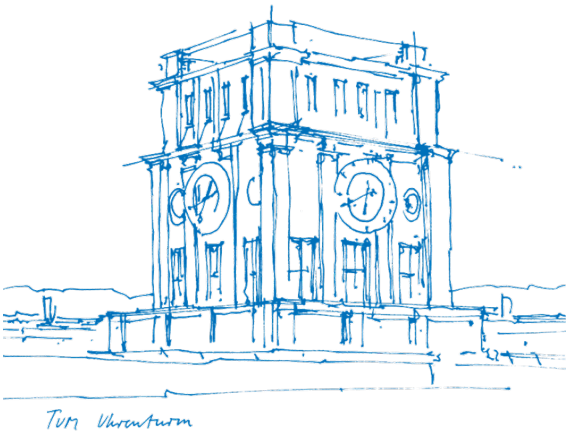
Docker, OCI containers

VT Student Presentation

Roberto Castellotti

TUM School of Computation, Information and Technology
Technical University of Munich

December 6th, 2022



History

- chroot
- freeBSD jails
- lxc/lxd

chroot

- "Jails build upon the `chroot(2)` concept, which is used to change the root directory of a set of processes. This creates a safe environment, separate from the rest of the system. **Processes created in the chrooted environment cannot access files or resources outside of it**. For that reason, compromising a service running in a chrooted environment should not allow the attacker to compromise the entire system. However, a chroot has several limitations. It is suited to easy tasks which do not require much flexibility or complex, advanced features.
- Over time, **many ways have been found to escape from a chrooted environment**, making it a less than ideal solution for securing services."[1]

[1] https://docs.freebsd.org/en/books/handbook/jails/

FreeBSD Jails (2000)

Jails improve on the concept of the traditional chroot environment in several ways.

- system resources
- system users
- running processes
- network stack

A jail has:

- a directory subtree
- a hostname
- an IP address
- a command to run

Root user is limited to the jail, this root account can't do anything on the host system.

LXC

Linux Containers (LXC) is an operating-system-level virtualization method for running multiple isolated Linux systems (containers) on a control host using a single Linux kernel.[2]. Can be used for both System Containers and Application Containers

- Kernel namespaces (ipc, uts, mount, pid, network and user)
- Apparmor and SELinux profiles
- Seccomp policies
- Chroots (using pivot_root)
- Kernel capabilities
- CGroups (control groups)

[2] https://en.wikipedia.org/wiki/LXC

LXC/LXD (lxc:2008)

LXD is a next generation system container and virtual machine manager. It offers a unified user experience around full Linux systems running inside containers or virtual machines.[3], Uses LXC for containers and Qemu for full virtualization.

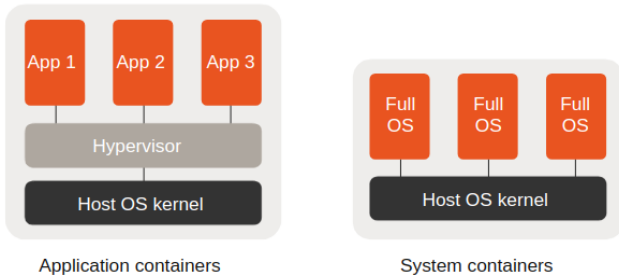


Figure 1 Application Containers (docker) vs System Containers (lxc)

[3] https://linuxcontainers.org/lxd/introduction/

LXD

A daemon, essentially the interface `lxc-*` commands interact with

- Both supports VMS and System Containers.
- Should use System Containers (shared Kernel) when possible
- Should use Virtual VMs if:
  - using a kernel feature not provided by host kernel
  - using a different OS (of course)

Probably the reason Docker had more success is this entire idea of "System Containers" is not really something needed in real world.

"Docker"

The Open Container Initiative is an open governance structure for the express purpose of creating open industry standards around container formats and runtimes.

Containers vs VMs (once again)

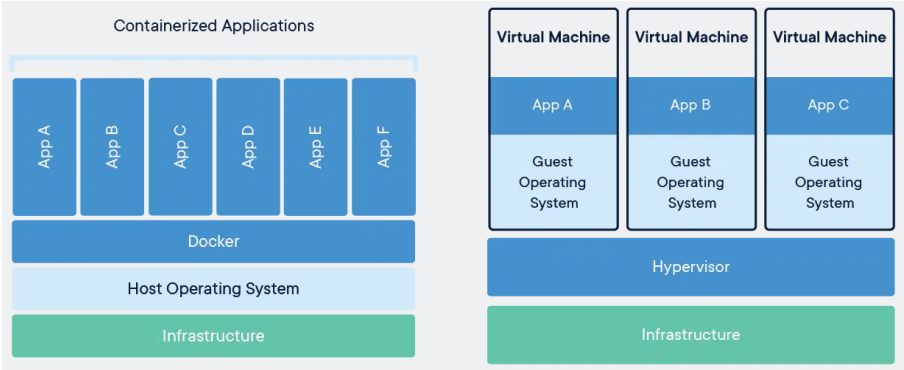


Figure 2 from: https://www.docker.com/resources/what-container/

Open Container Initiative (OCI)

The OCI was launched on June 22nd 2015 by Docker, CoreOS (Container Linux) and other leaders in the container industry. Currently 3 specifications:

- the Runtime Specification (runtime-spec)
- the Image Specification (image-spec)
- the Distribution Specification (distribution-spec).

Docker donated its container format and runtime, runC (tool to spawn and run containers on linux), to the OCI to serve as the cornerstone of this new effort. It is available now at opencontainers/runc.

How do containers provide isolation?

"Applications are safer in containers and Docker provides the strongest default isolation capabilities in the industry" [4]

- linux namespaces (limits what you see)
- cgroups (limits what you can use use)

[4] https://www.docker.com/resources/what-container/

docker: quickstart

- create a Dockerfile:

```
FROM ubuntu:18.04
COPY . /app
RUN make /app
CMD python /app/app.py
```

- `sudo docker build -t test .`
- `sudo docker run test` (ports? volumes?)

"Dockerfile" is supported also by other tools (podman, buildah) for compatibility reasons, while other tools, like HashiCorp Packer decided not to support them in order to not be tied to Docker in any way[5]

[5] https://developer.hashicorp.com/packer/plugins/builders/docker

How do we run containers in production?

- NOT `docker-compose`
- Docker Swarm, Kubernetes, Nomad orchestration tools
- "Kubernetes, also known as K8s, is an open-source system for automating deployment, scaling, and management of containerized applications." [6]
- Usually managing a K8s cluster is not reccomended (huge complexity)
- Amazon EKS clusters, Azure AKS, Google GKE
- A lot of k8s distros: k8s is "only" a set (of huge) API specifications (minikube for local development, GKE for production)
- K8s and Nomad are big and powerful tools, use them only if needed! (HashiCorp Nomad scheduled 2,000,000 Docker containers on 6,100 hosts in 10 AWS regions in 22 minutes.)
- logging? backups? replicas? monitoring?

[6] https://kubernetes.io/

Docker is lighter than a VM, or is it?

- microVMs: provide enhanced security and workload isolation over traditional VMs, while enabling the speed and resource efficiency of containers. [7]
- **Firecracker**: KVM but no Qemu, custom VMM
- **Firecracker**: offers memory overhead of less than 5MB per container, boots to application code in less than 125ms, and allows creation of up to 150 MicroVMs per second per host.
- Why do we need this?
- **LightVM**: able to boot a (unikernel) VM in as little as 2.3ms, reach same-host VM densities of up to 8000 VMs
- **Unikernels**: tiny virtual machines where a minimalistic operating system (Unikraft)
- In some cases, a VM is lighter (and safer) than a container

[7] https://firecracker-microvm.github.io/

further readings/talks

- https://github.com/containerd/containerd/blob/main/docs/getting-started.md
- https://lwn.net/Articles/531114/ (the entire series)
- https://blog.quarkslab.com/digging-into-linux-namespaces-part-1.html
- `man {namespaces,cgroups}`
- https://drewdevault.com/2022/11/12/In-praise-of-Plan-9.html
- @jpetazzo - Cgroups, namespaces, and beyond: what are containers made from?
- Containers unplugged: Linux namespaces - Michael Kerrisk
- https://docs.kernel.org/admin-guide/cgroup-v1/cgroups.html
- Hashicorp Nomad - The Two Million Container Challenge
- My VM is Lighter (and Safer) than your Container
- https://firecracker-microvm.github.io/