

Cloud computing

Virtualisation

Aleix Llusà Serra

Master's degree in Machine Learning and Cybersecurity for Internet Connected Systems

Universitat Politècnica de Catalunya

<https://epsem.upc.edu>

10 de setembre de 2026



This work is licensed under a Creative Commons "Attribution-ShareAlike 4.0 International" license.



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Master's degree in Machine Learning and Cybersecurity for Internet Connected Systems

1 Introduction to virtualisation

2 Lab

3 Containerisation: Docker



- Software engineering and systems/services deployment
- Virtualisation
- Microservices
- Digital transformation

A virtual machine (**guest**) is like a real physical machine encapsulated in an operating system (**host machine**). Why do we want that?

- Isolation from the hardware
- Isolation from the software of the host OS
- Economic savings (also environmental resources)
- More efficiency, limit overpowering: space, energy...
- Agility when developing and operating systems (DevOps)
- Dynamic resource assignment: CPU, memory, disk...
- Run multiple applications on a single server (vs each new application, a new server (Poulton 2020))

- Hypervisor (hardware virtualisation), simulating physical machines: Qemu/KVM, Xen, VMWare, VirtualBox
 - Full virtualisation: Type 1 (bare metal) Type 2 (hosted)
 - Paravirtualisation: guest is ported (e.g. the kernel) to cooperate with the host hypervisor
- Containerisation (operating-system level), multiple isolated user-space: LXC, Docker

- OS administration, GNU/Linux system administration
- Command-line (Bash and similar)
- Basic network concepts

<https://wiki.debian.org/QEMU>

- 1 Download a GNU/Linux OS installation image
- 2 Create an empty hard disk image
- 3 Boot from the installation image
- 4 Install the OS from zero to the hard disk (no graphics needed)
- 5 Boot the OS
- 6 SSH from your host machine

Alternative: GNOME Boxes (desktop qemu/KVM), or with less potential but more user-friendly Virtualbox OSE (Open Source Edition)

<https://www.virtualbox.org/>

What is Docker?



©Wikimedia Commons. Former logo



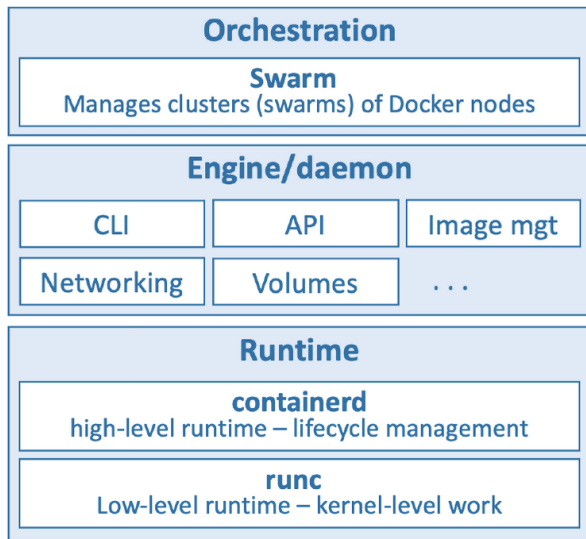
©Wikimedia Commons non-free

Definition

Docker was the magic that made Linux containers usable for mere mortals. Put another way, Docker made containers simple! (Poulton 2020)

Note that containers run in host OS's kernel (as any other process), so they must be compatible.

Docker architecture



Docker engine architecture

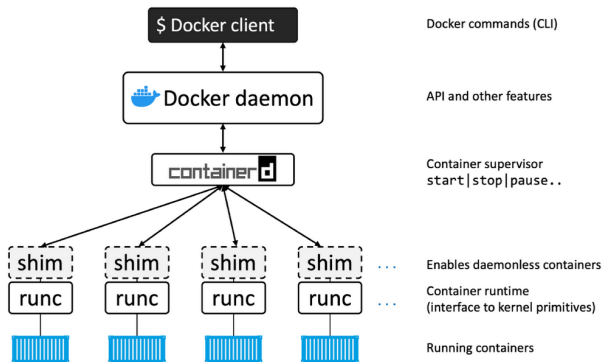


Figure 5.3

© Poulton (2020)

The Open Container Initiative (OCI) <https://opencontainers.org>
Standardisation for fundamental container infrastructure:

- Image format
- Container runtime
- Distribution registries

Mainly oriented to single application or service.

- Build: images
- Run: containers

When created they begin from image (except data).

Documentation: `docs.docker.com` and Poulton (2020)

Lab: <https://labs.play-with-docker.com/>



Poulton, Nigel (2020). **Docker deep dive. Zero to Docker in a single book**. 2a ed. Birmingham: Packt Publishing. ISBN: 9781835087916. URL: https://discovery.upc.edu/permalink/34CSUC_UPC/8e3cvp/alma991005249877806711.