

How to run a software on a computer?

Lecture 5

- Running a software
 - Compiling, program file
- Multiple program on a computer
 - Protection-security
 - Isolation
 - Virtualization
- Operating system as a resource manager
- Virtual machine
 - Hypervisor
 - Virtualbox, vmware,
- Hardware Emulators
 - QEMU
- Containerization
 - OS-level containers
 - Dockers
 - Containerized applications
 - Application containers

Compiling a program file

source file

program1.c

program2.py

Main.cpp

Main.java

Binary(program) file (machine code)

- gcc main.c -o pogramfile
 - Generates a binary file
- g++ main.cpp -o programfile
 - Generates a binary file
- javac Main.java
 - Generates a bytecode “.class” file
 - java Main.class runs the file on JVM
- Python main.py
 - compiles/interpretes and runs the code
 - Binary generated in the process

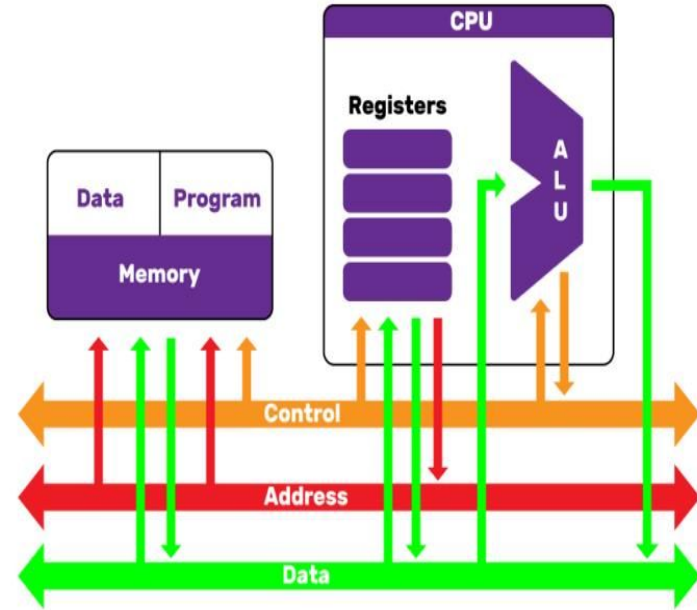
Process

Program: a binary file in hard disk

When we run a program,

- We need to load its data, instructions into memory
- And run on CPU

Process: The program that is loaded into memory



Running one program vs running multiple program

Multi-tasking

- Computers can run many program at the same time
- Many programs use the same memory, CPU, I/O devices

Problems?



Running one program vs running multiple program

Multi-tasking

- Computers can run many program at the same time
- Many programs use the same memory, CPU, I/O devices

Problems

- One program can affect another
- One program can steal another's data
- There may be more than one user

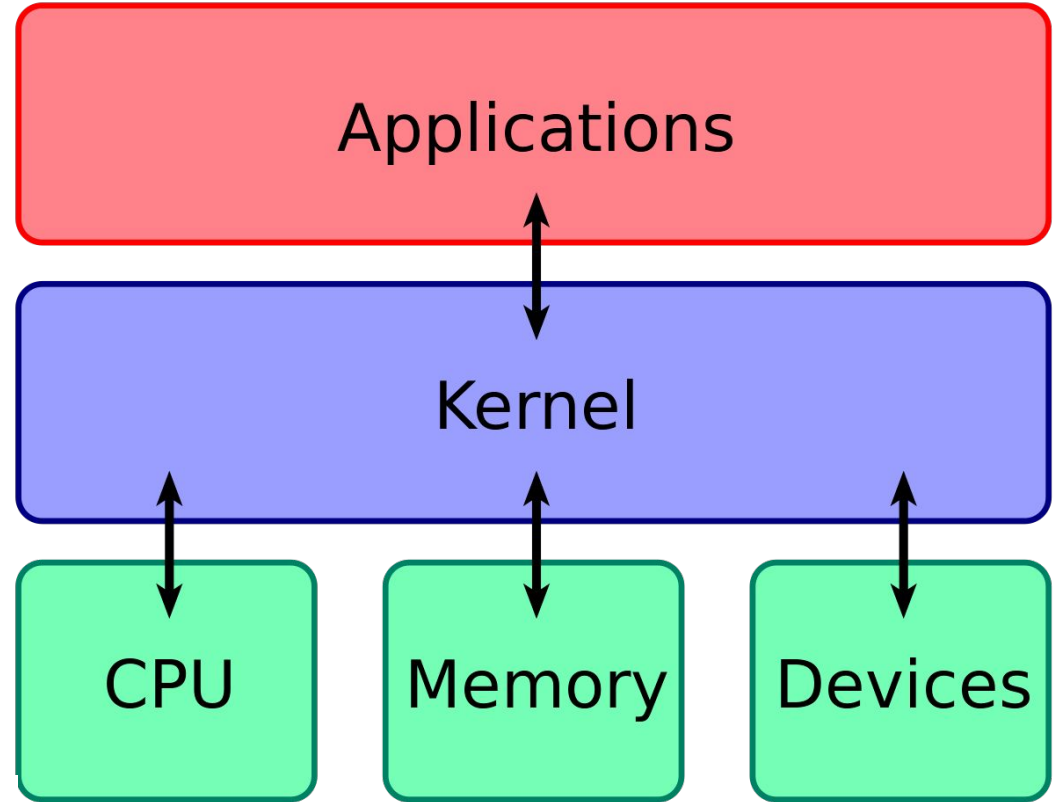
Protection

- How to protect system resources (hardware)?
- How to prevent one program affecting another?
- ..



Operating system

Kernel: the core of os that controls system resources



Protection

Requires

- Isolation of processes
 - Isolate memories they use
 - Protect one process from another



<https://www.patlabelsonline.co.uk/health-and-safety-labels-c203/danger-isolate-before-opening-warning-labels-p1117>

Protection

Requires

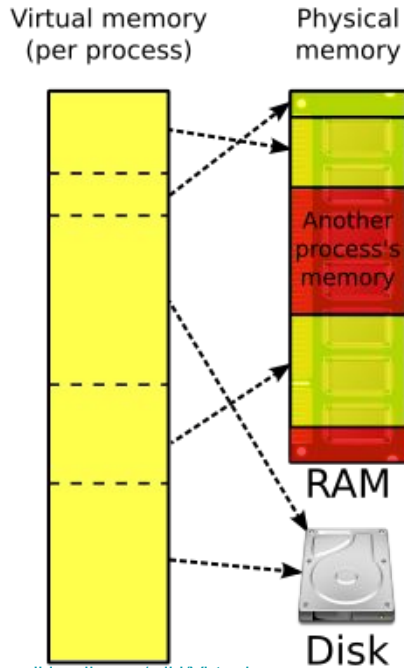
- Isolation of processes
 - Isolate memories they use
 - Protect one process from another



- Virtualization of system resources
 - Process do not know the exact memory layout, or other hardware.
 - Goes through OS to access hardware

Virtualization of system resources

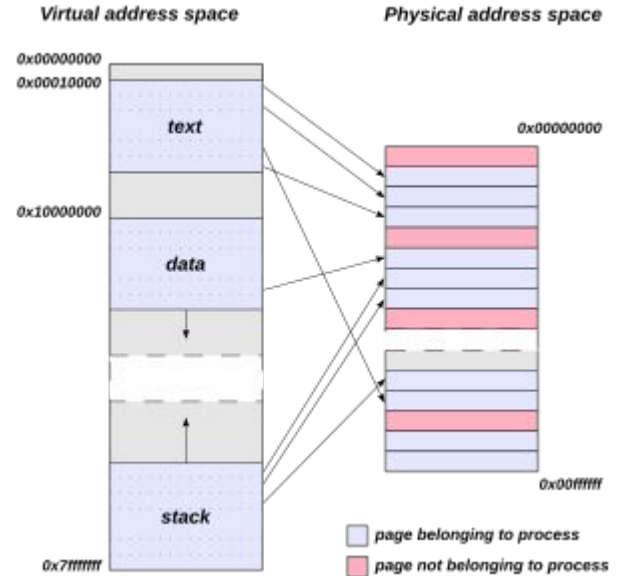
Memory management with Virtual memory



https://en.wikipedia.org/wiki/Virtual_memory

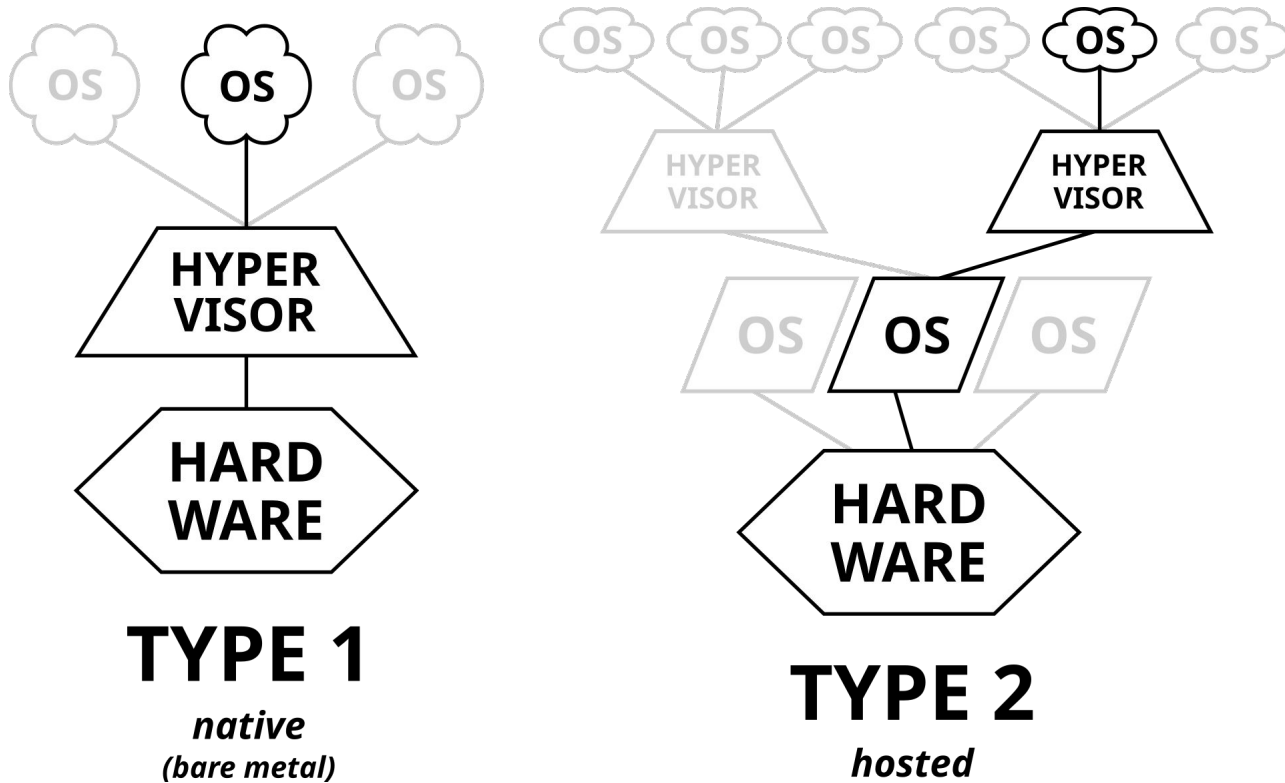
Virtual address space

(set of address available to a process)



https://en.wikipedia.org/wiki/Virtual_address_space

Virtualization of full computer architecture



Hypervisors

a virtual machine monitor (VMM), manages virtual machines (VMs) as they run alongside each other.

Type-1 hypervisor (bare metal hypervisor)

- Interacts directly with the underlying machine
- Installed on the host's physical machine
 - Not through OS
- It can allocate/share resources to VMs

Type-2 hypervisor

- interacts with the underlying hardware through the host machine OS
- Runs as an application

	Type 1 hypervisor	Type 2 hypervisor
Also known as	Bare metal hypervisor.	Hosted hypervisor.
Runs on	Underlying physical host machine hardware.	Underlying operating system (host OS).
Best suited for	Large, resource-intensive, or fixed-use workloads.	Desktop and development environments.
Can it negotiate dedicated resources?	Yes.	No.
Knowledge required	System administrator-level knowledge.	Basic user knowledge.
Examples	VMware ESXi, Microsoft Hyper-V, KVM.	Oracle VM VirtualBox, VMware Workstation, Microsoft Virtual PC.

<https://aws.amazon.com/compare/the-difference-between-type-1-and-type-2-hypervisors/>

Distinction is not clear everytime

[KVM](#) and [bhyve](#) are [kernel modules](#) that effectively convert the host operating system to a type-1 hypervisor

[Comparison of platform virtualization software - Wikipedia](#)

Virtualization on system level

Virtualization allows you to:

- Run software that requires an older versions and different operating systems.
- Experiment with other operating systems.
- Test software on multiple operating systems using multiple virtual machines.

you can run them all on a single desktop or laptop computer.

<https://learn.microsoft.com/en-us/virtualization/hyper-v-on-windows/about/#reasons-to-use-virtualization>

QEMU(Quick Emulator)

QEMU is a generic and open source machine emulator and virtualizer.

It supports various virtualization modes and can emulate a wide range of hardware.

[About QEMU — QEMU documentation](#)

[QEMU](#)

To run on Mac

- <https://mac.getutm.app/>

Installing Linux via Virtual Machine softwares

Download virtual box

- [Downloads – Oracle VirtualBox](#)

You can also download VMware etc. or [Virt-Manager](#),

[How to run an Ubuntu Desktop virtual machine using VirtualBox 7](#)

[How To Install Debian 12 On VirtualBox](#)

On mac

- Mac OS, OSX is a UNIX based OS
 - Similar to linux

To install Linux

- <https://mac.getutm.app/>
- [Ubuntu 22.04 | UTM Documentation](#)

Windows Subsystem for Linux(WSL1-WSL2)

WSL1

You can run Linux/GNU environments

- Command-line tools,
- Utilities,
- Applications

Uses compatibility layer

[Install WSL | Microsoft Learn](#)

[FAQ's about Windows Subsystem for Linux | Microsoft Learn](#)

<https://learn.microsoft.com/en-us/windows/wsl/compare-versions>

WSL2 uses hyper-V architecture to enable virtualization

- WSL1
- +Full linux kernel
-

Another example: android emulators

Uses the virtual machine

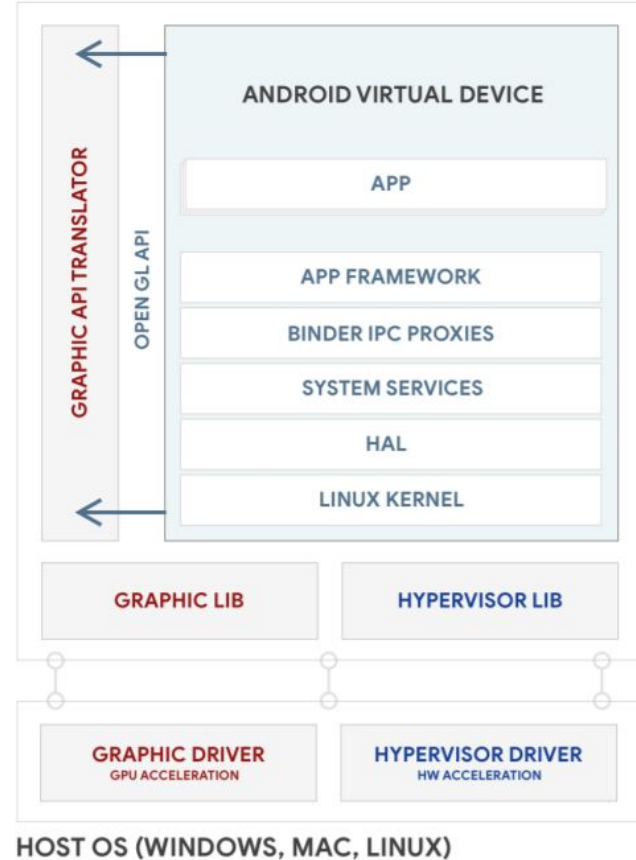
- Android Virtual Device
 - Contains full android software stack

[Use Android Emulator virtual devices](#)

[Cuttlefish virtual Android devices](#)

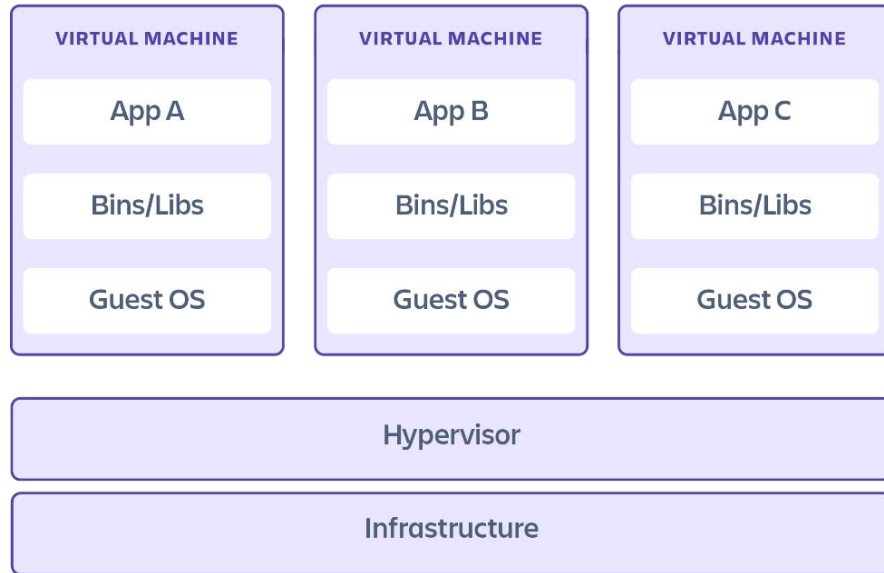
Cuttlefish is a configurable virtual Android device

ANDROID EMULATOR ENGINE

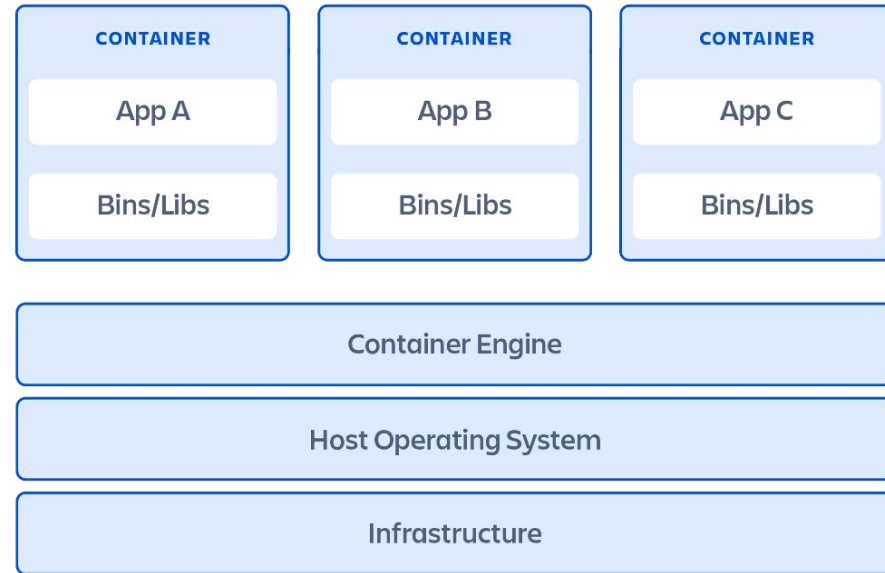


Containerization

Virtual machines



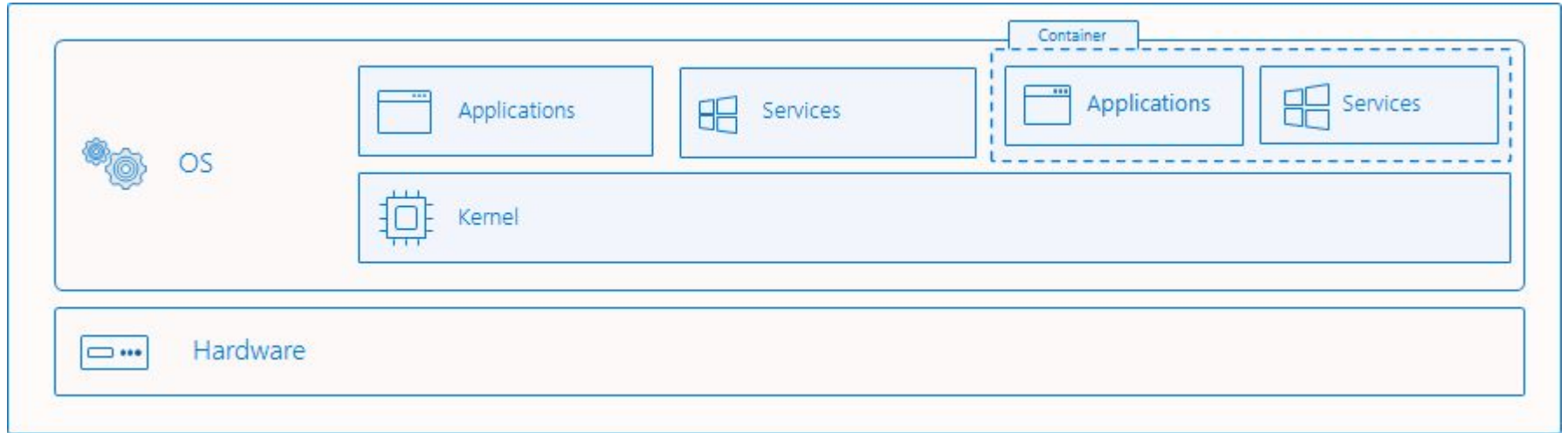
Containers



<https://www.atlassian.com/microservices/cloud-computing/containers-vs-vms>

Example from Windows

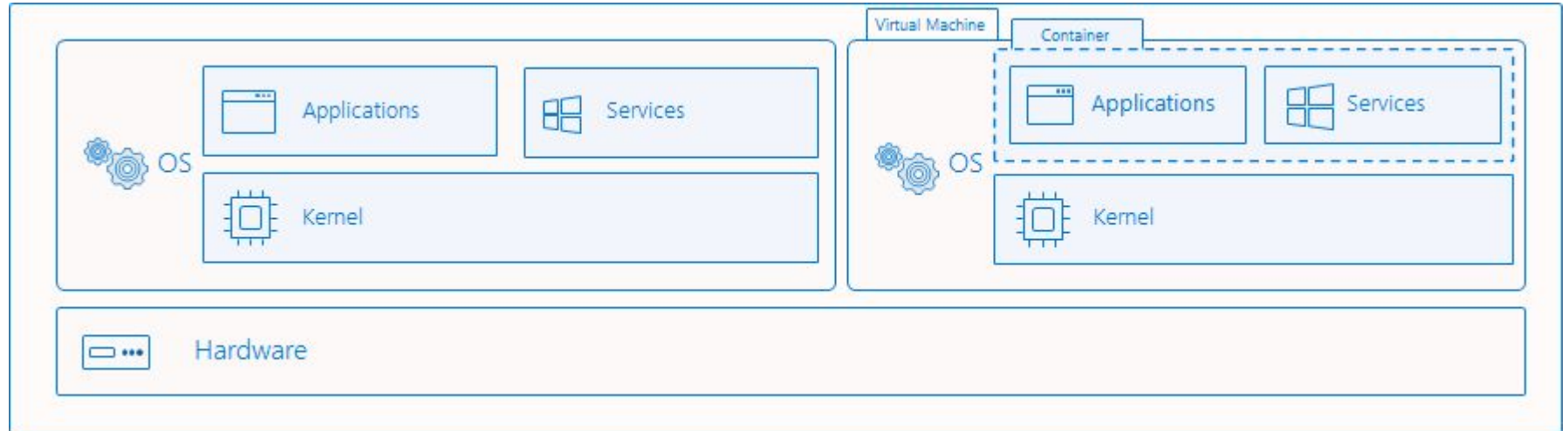
Process isolation with Windows containers



<https://learn.microsoft.com/en-us/virtualization/windowscontainers/manage-containers/hyperv-container>

Example from Windows

Process isolation with virtual machine and containers



<https://learn.microsoft.com/en-us/virtualization/windowscontainers/manage-containers/hyperv-container>

Examples

[Linux Containers](#)

[Linux Containers - LXC - Introduction](#)

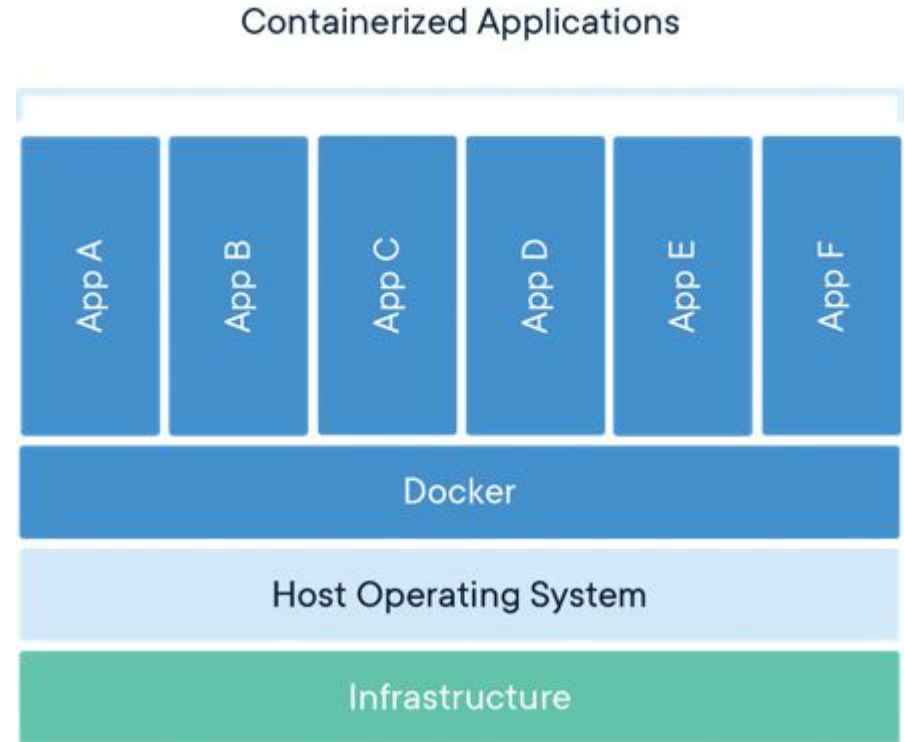
[An Introduction to FreeBSD Jails](#)

- [Chapter 17. Jails and Containers | FreeBSD Documentation Portal](#)

[Podman](#)

<https://podman-desktop.io/>

[Docker](#)



<https://www.docker.com/resources/what-container/>

Installing docker

<https://docs.docker.com/engine/install/>

- [Install Docker Desktop on Linux](#)

[Install Docker Desktop on Windows](#)

- [Docker Desktop WSL 2 backend on Windows](#)

[Install Docker Desktop on Mac](#)

Installing Linux as a docker image

[ubuntu - Official Image | Docker Hub](#)

[debian - Official Image | Docker Hub](#)

https://hub.docker.com/search?type=image&operating_system=linux&image_filter=official&categories=Operating+Systems

Kubernetes

Kubernetes, also known as K8s, is an open source system for managing [containerized applications](#) across multiple hosts.

It provides basic mechanisms to

- deploy,
- scale,
- and manage containerized applications.

[GitHub - kubernetes/kubernetes: Production-Grade Container Scheduling and Management](#)

[Kubernetes](#)

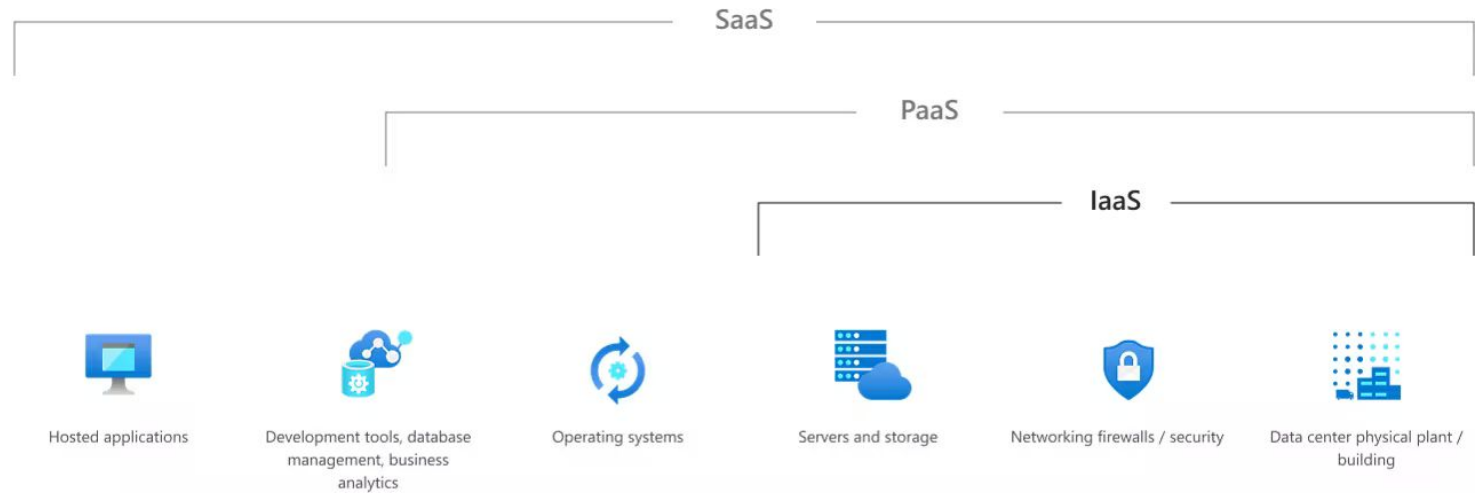
Containerization on cloud computing

Infrastructure as a service (IaaS)

software as a service ([SaaS](#)),

platform as a service ([PaaS](#))

[serverless](#)



When to use?

OS containers

- Linux containers (LXC)
- windows containers
- **When you need extra isolation for some apps!**

Virtual machine

- When you need to run different OS
- Further isolation

Dockers etc.

- Deploying an app,
- Bundled/portable apps

HW:

Install Ubuntu/debian or any other Linux on VirtualBox

Next week

Linux OS

Terminal commands and more