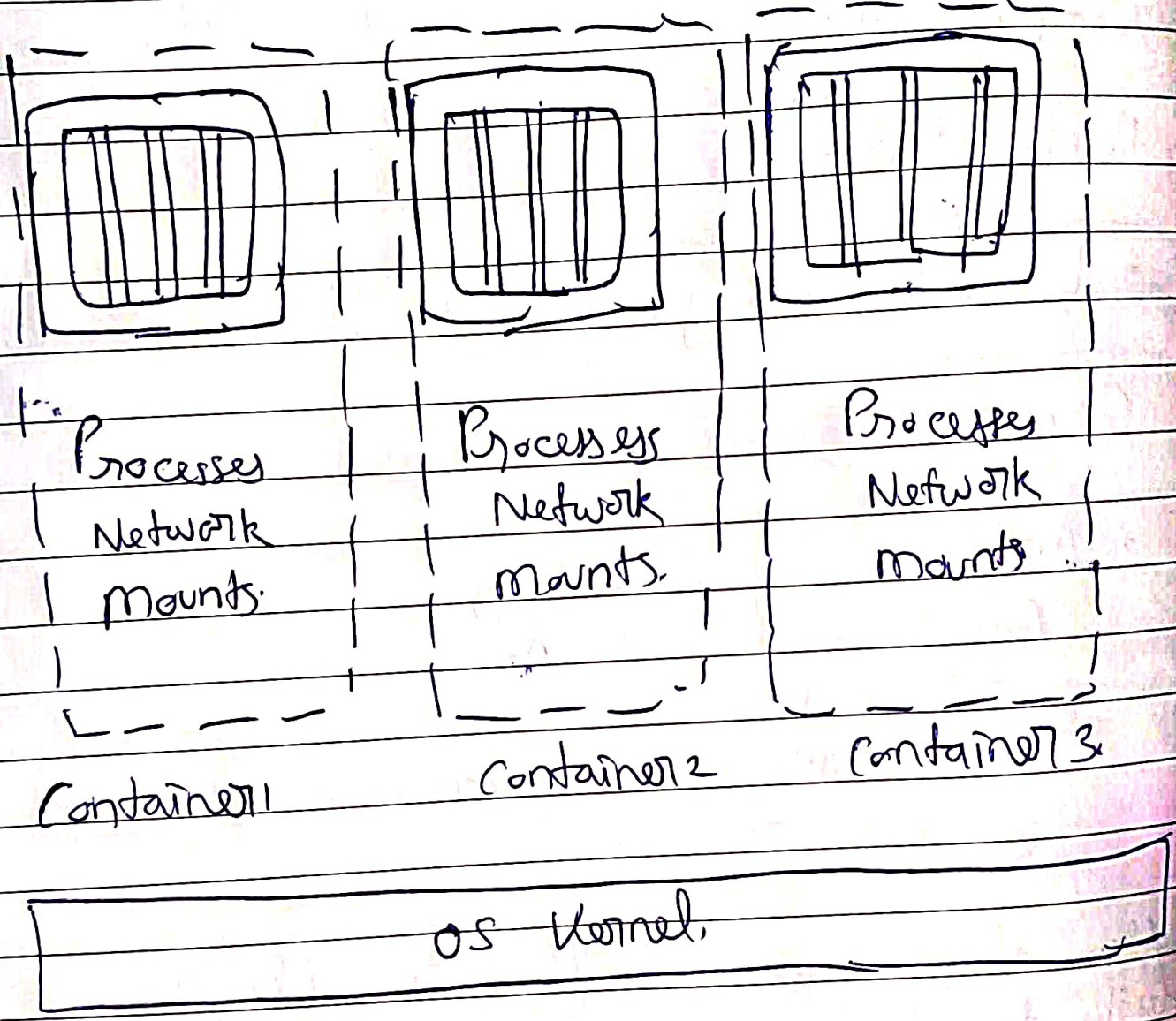
(23rd Dec 2025)

(Docker)

What are containers?

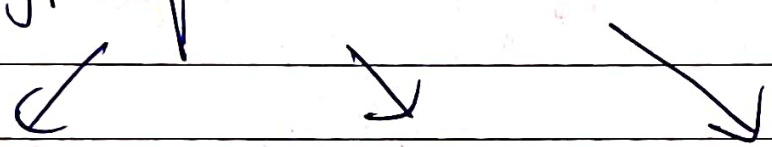


A container is a separate environment which has its own processes, network and mounts, but share the same OS Kernel.

Each container gets its own.

* Process space → Isolation of process across containers
* Network stack → own IP Address, NAT
* File system mount points → Its own files & directories.
* User namespace → Each user has its own namespace

Types of Containers.



(Linux Containers) → [Docker started with this].

LXC was heavy as Virtual Machines, it had system init processes and everything And since it was Linux so heavily coupled with Linux systems making it ~~difficult to~~ tightly coupled.

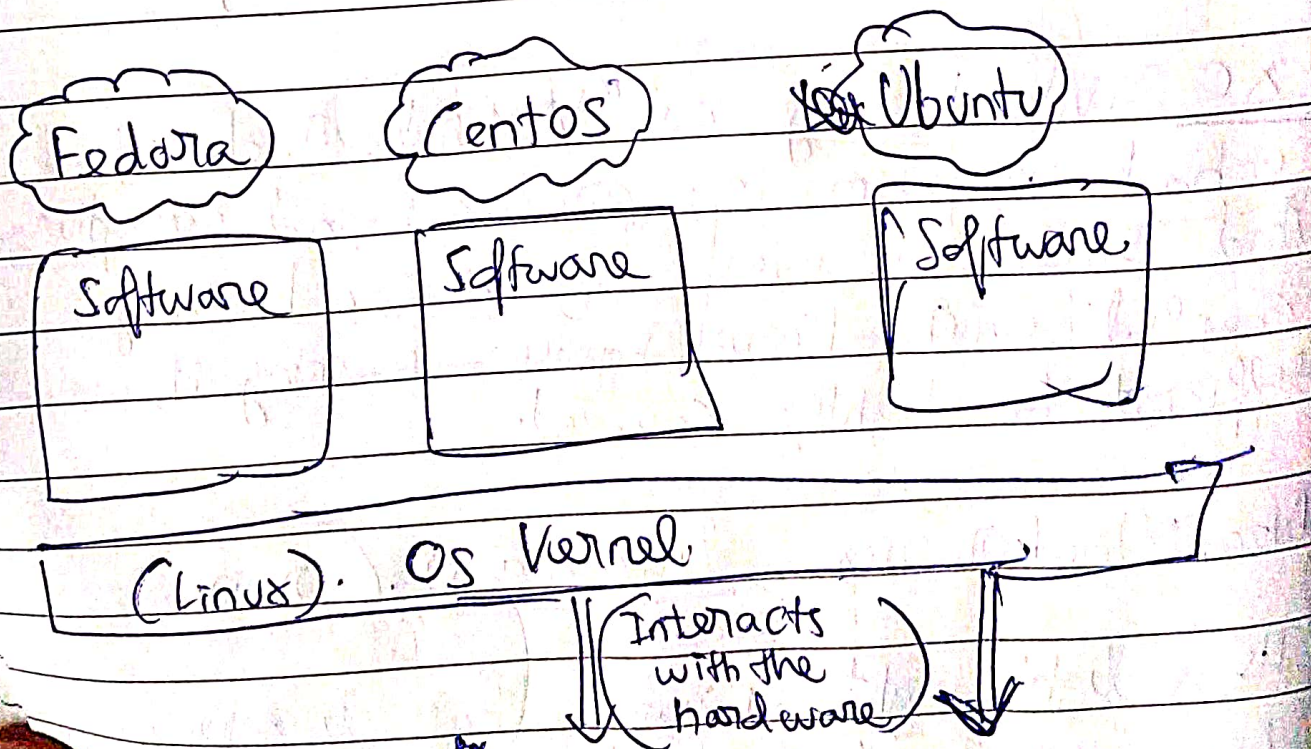
Now Docker uses libcontainer. (runc)

→ Its a light weight library that directly uses Linux kernel features → Essentially bare minimum needed to create and run containers.

libcontainer/runc is like assembling a computer from raw components yourself and get exactly what you want.

Docker wanted the direct control to innovate faster and create containers that were lighter and more purpose-built to run a single app/service rather than full system environment

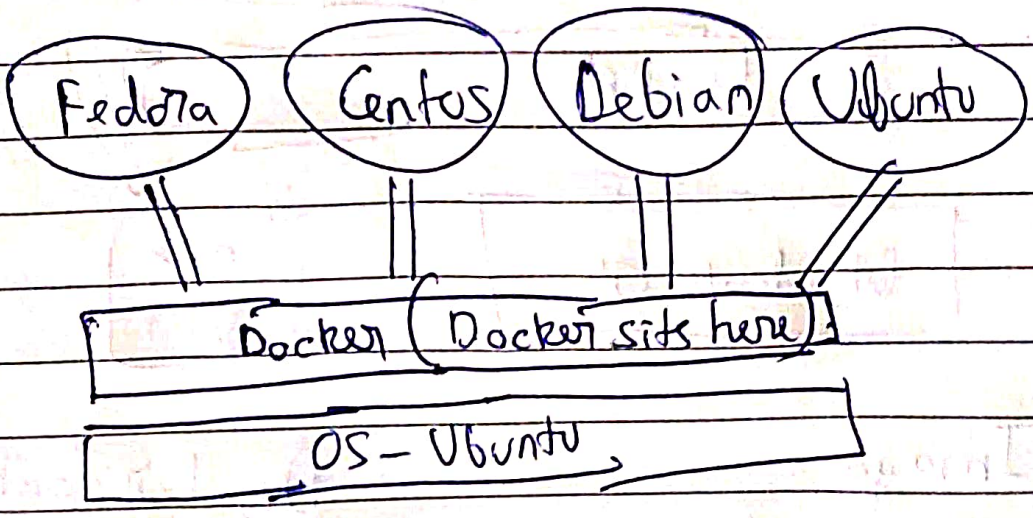
Now lets dive a level deeper to solidify our understanding



So these operating systems like Centos, Ubuntu, Fedora, debian all use Linux Kernel to interact with the hardware.

Only thing differentiating ~~them from~~ is the software interacting with the kernel. like the UI etc.
Hence we see different UI, for different OS.

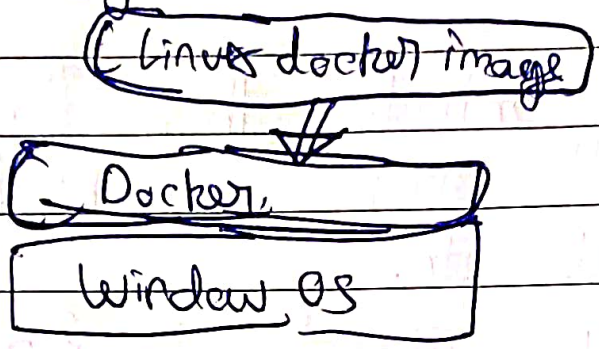
Ubuntu ~~Container~~ Docker image can also run on Centos machine and vice-versa, as all are compatible.



***

But windows docker image won't be able to run on Linux Kernel.

But in practical if you try to run a linux docker image on a window operating system.



This will run fine. The instructor said because, this Docker on windows spins up a linux OS VM, one which this docker image runs.

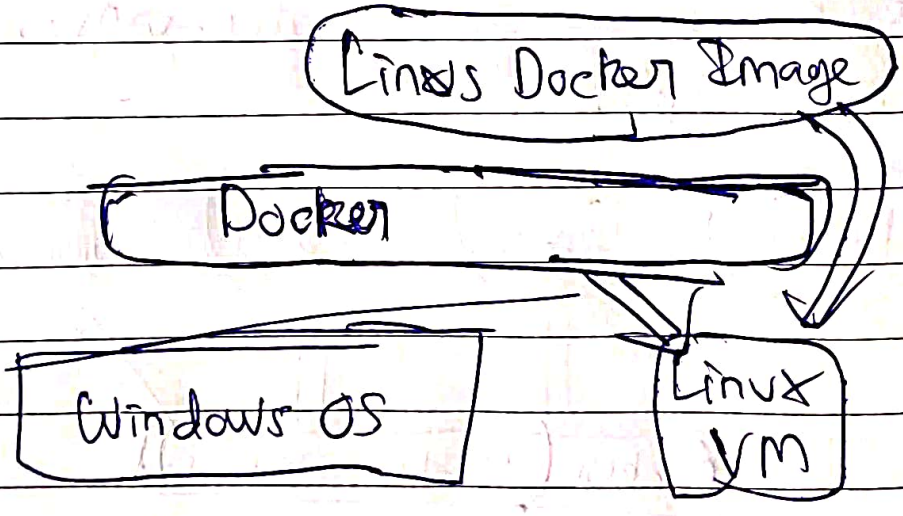


Image (Package, template, Plan) ⟹ Docker Container. [Its the running instance of the image].