

## Virtual Machines (VMs)

A Virtual Machine is like a "fake" computer running inside your real computer. It has its own operating system (like Windows or Linux) and acts just like a real computer.

But, because it's a "fake" computer, it needs extra resources to run (memory, disk space, etc.), which makes it slower and heavier to run.

Ex: VMs are like renting an apartment: You have your own space with your own kitchen, bathroom, etc. But it's a bit heavy and takes time to set up.

## Containers

Containers are like mini-computers that don't need their own operating system. They share the same operating system as the host computer (the real one).

They take up less space and are much faster because they don't have to load a full OS every time. They just run the application and its necessary parts.

Ex: Containers are like renting a room in a shared house: You share the kitchen and bathroom with others, but you have your own space (the room) that's ready to use right away, and it's much lighter and quicker to set up.

### Key Differences:

#### 1. What Do They Run?

- VMs: Run a full operating system (like Windows, Linux), with its own kernel (the core part of the OS).
- Containers: Run on top of the host OS and share the kernel. They just package the application and needed files.

#### 2. Resources (Memory, CPU, etc.):

- VMs need a lot of resources because they run a full operating system. Think of it like a whole computer system inside your computer.
- Containers are lightweight. They share the operating system, so they don't need as many resources.

#### 3. Performance (Speed):

- VMs are slower to start because they need to boot up a whole operating system.
- Containers start almost instantly because they don't need to load an entire OS.

#### 4. Isolation (Security):

- VMs are like totally separate computers. They don't interfere with each other, making them more secure.
- Containers are more like rooms in the same building. They share some parts (like the OS) but are still isolated from each other. They are not as secure as VMs.

## Docker

Docker is a platform that allows developers to package applications and all their dependencies into something called a container, so they can run anywhere, consistently.

Docker lets you create a "box" (container) that holds your app, along with everything it needs — so it runs the same on your computer, a server, or in the cloud.

## Why Do We Need Docker?

Without Docker:

- An app might work on your machine but not on someone else's
- You need to install the right Python/Java/Node version, libraries, tools, configs — manually

With Docker:

- Your package everything together
- Your app runs identically everywhere

## What is a Container?

Docker is a container

A container is a lightweight, standalone environment where your app runs.

Think of it as a mini-computer inside your real computer:

- Has its own file system
- Its own libraries
- Runs your app in isolation

But unlike a virtual machine, containers:

- Are much faster
- Use less memory
- Start in seconds

## How Docker Works (Simplified)

1. You create a Dockerfile that tells Docker how to build your app environment.

```
FROM python:3.9
WORKDIR /app
COPY . .
RUN pip install -r requirements.txt
CMD ["python", "app.py"]
```

2. Build the image (a snapshot of the app):

```
docker build -t my-app .
```

3. Run a container from that image:

```
docker run -p 5000:5000 my-app
```

Now your app is running in its own portable box!

---

Need of container : compatibility/dependency, long setup time, diff dev/test/prod environment

with docker (elexy container uses) : Containersize application

container : completely isolated env.

OS woking : OS kernal and set of software