

THE VIRTUAL REALM

Harsh Karwa
Vaishnavi Ladda



#01

Devops

#02

Virtualisation

#03

Podman

#04

Vagrant

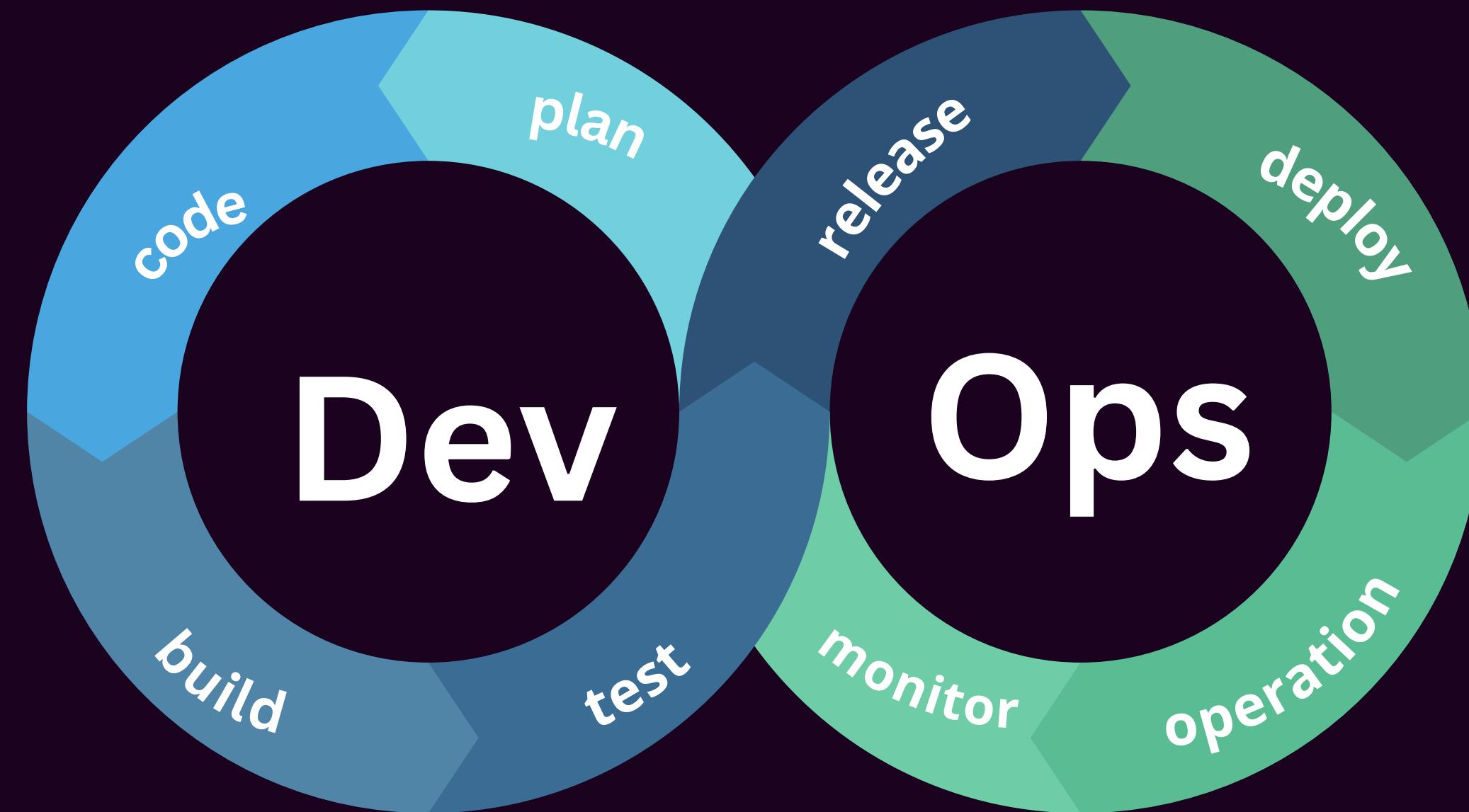
Table of Contents

WHAT IS DEVOPS ?

DevOps = Development + Operations

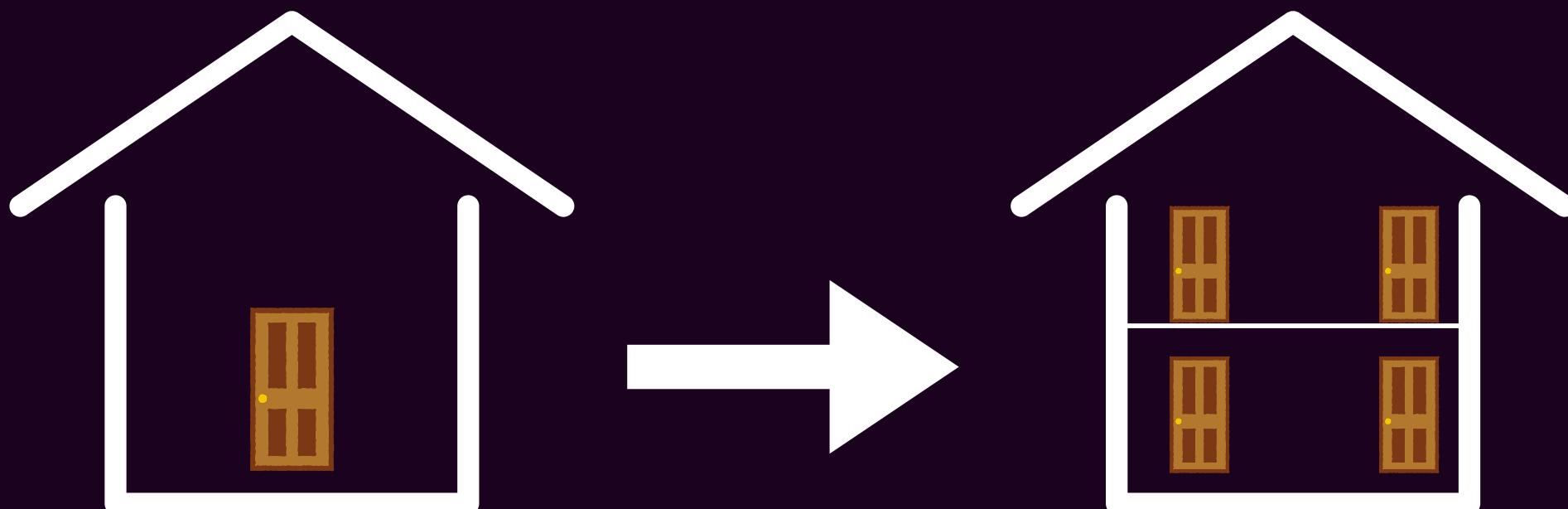
1. There are 2 teams : Development team and the Operational team
2. These teams have to work in sync with each other
3. This follows Continuous Integration and Continuous Deployment

WHAT IS DEVOPS ?

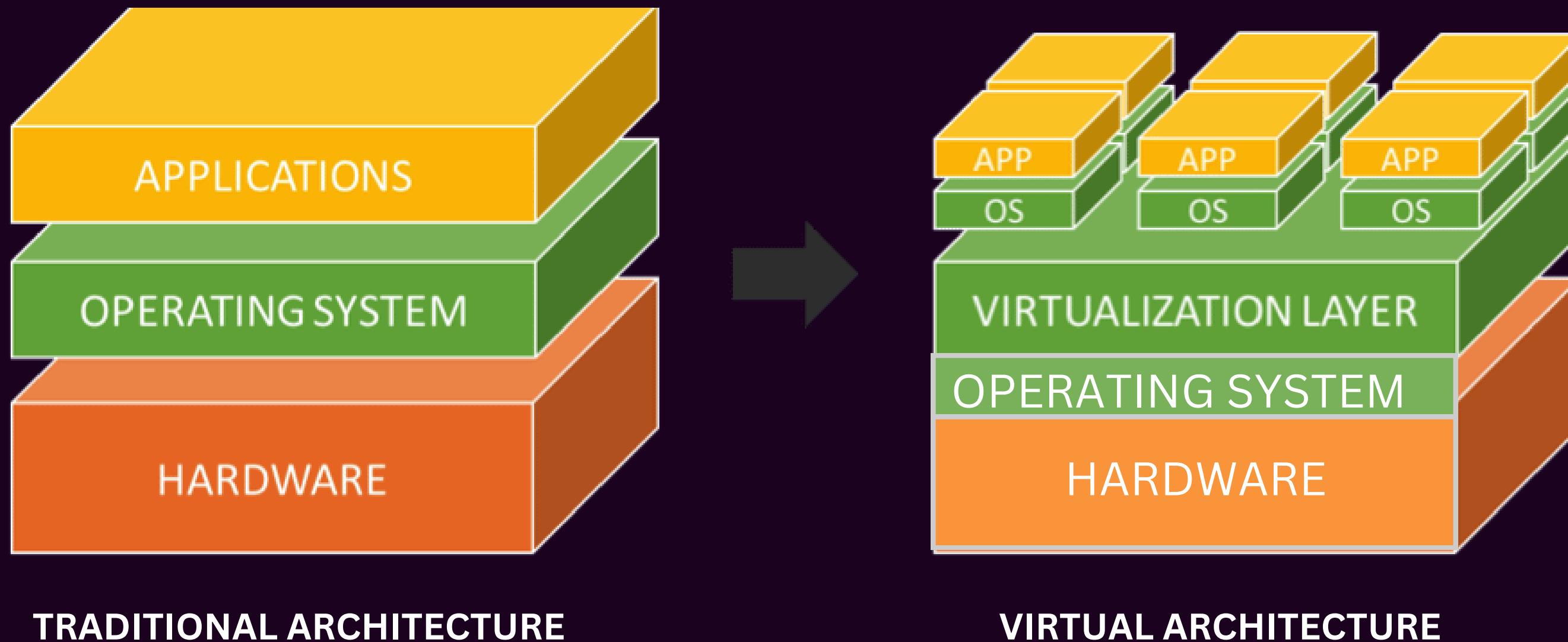


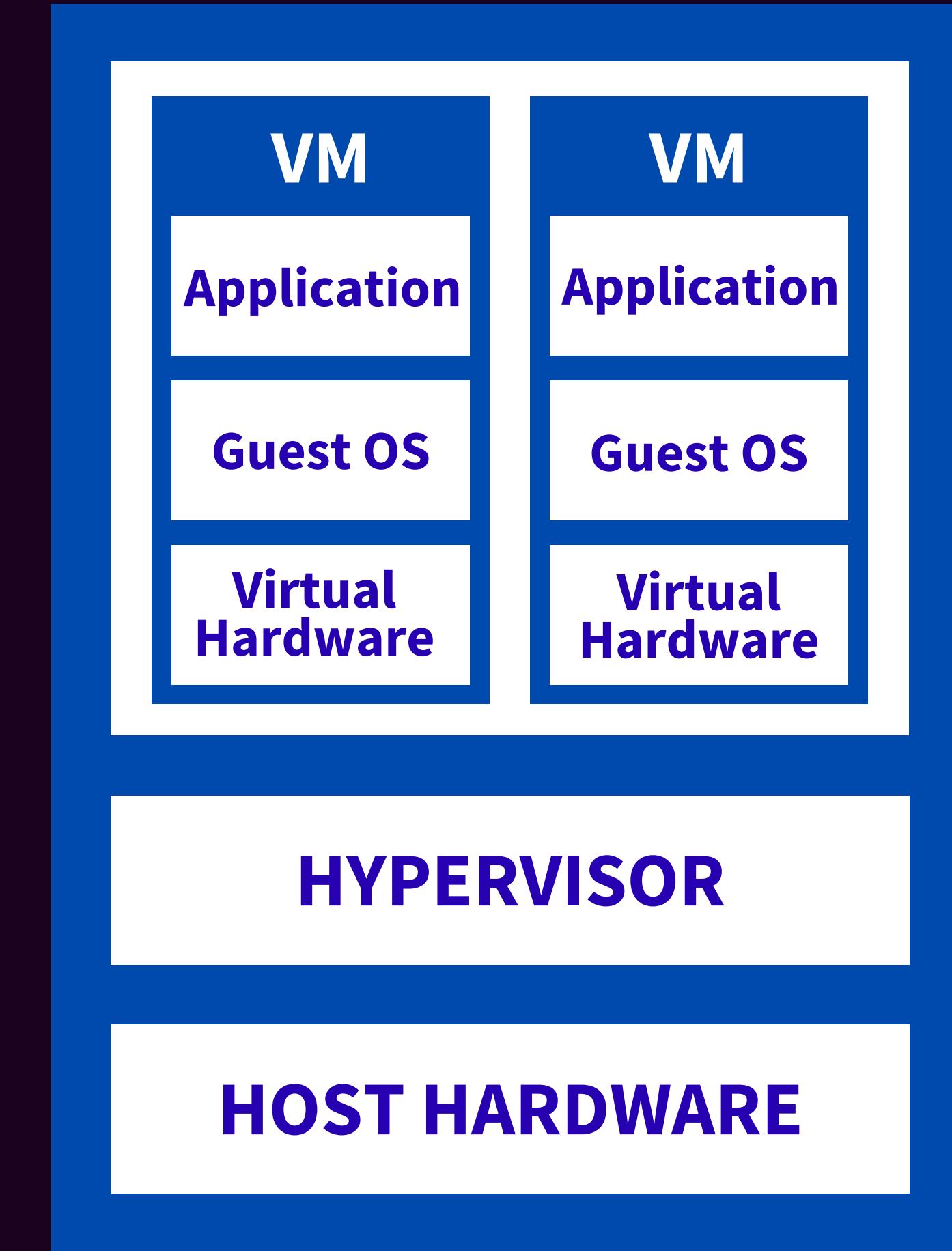
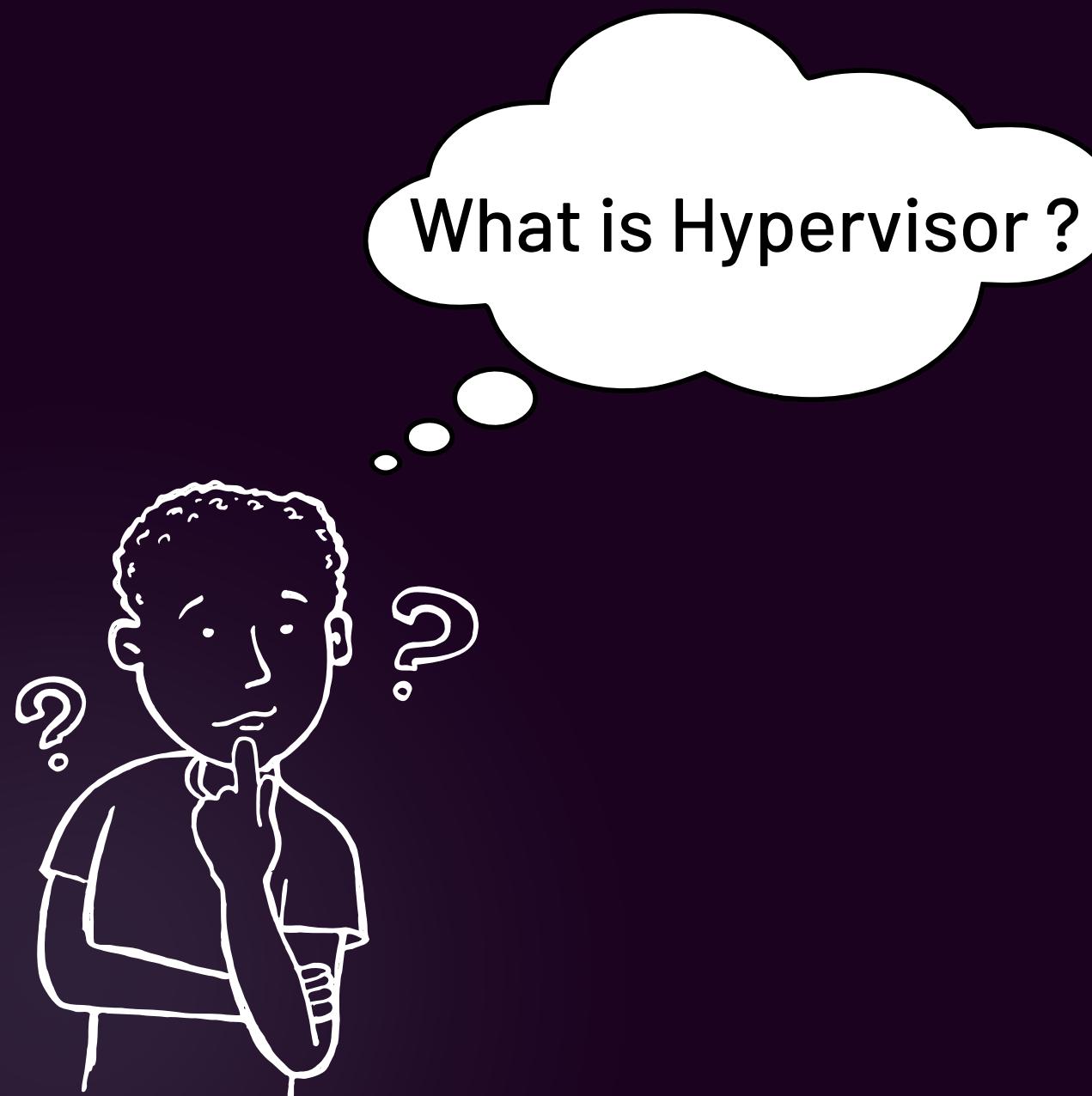
WHAT IS VIRTUALISATION ?

In computing, virtualization refers to the act of creating a virtual (rather than actual) version of something, this includes virtual computer hardware, virtual storage devices and virtual network resources.



Physical Server VS Virtualisation





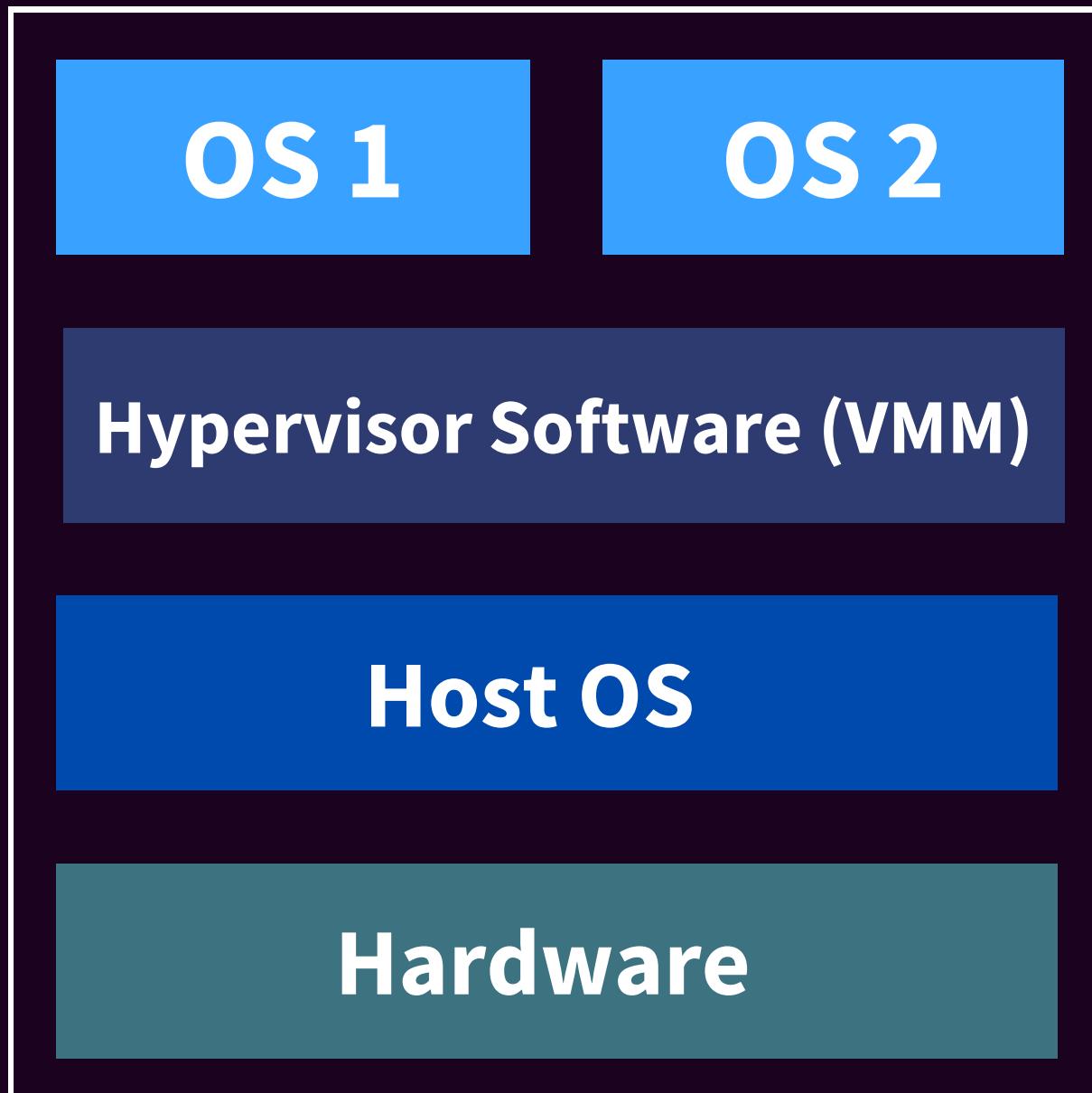
What is a Hypervisor ?

A hypervisor is also called a virtual machine manager(VMM). #01

It is a software/firmware that creates and runs virtual machines. #02

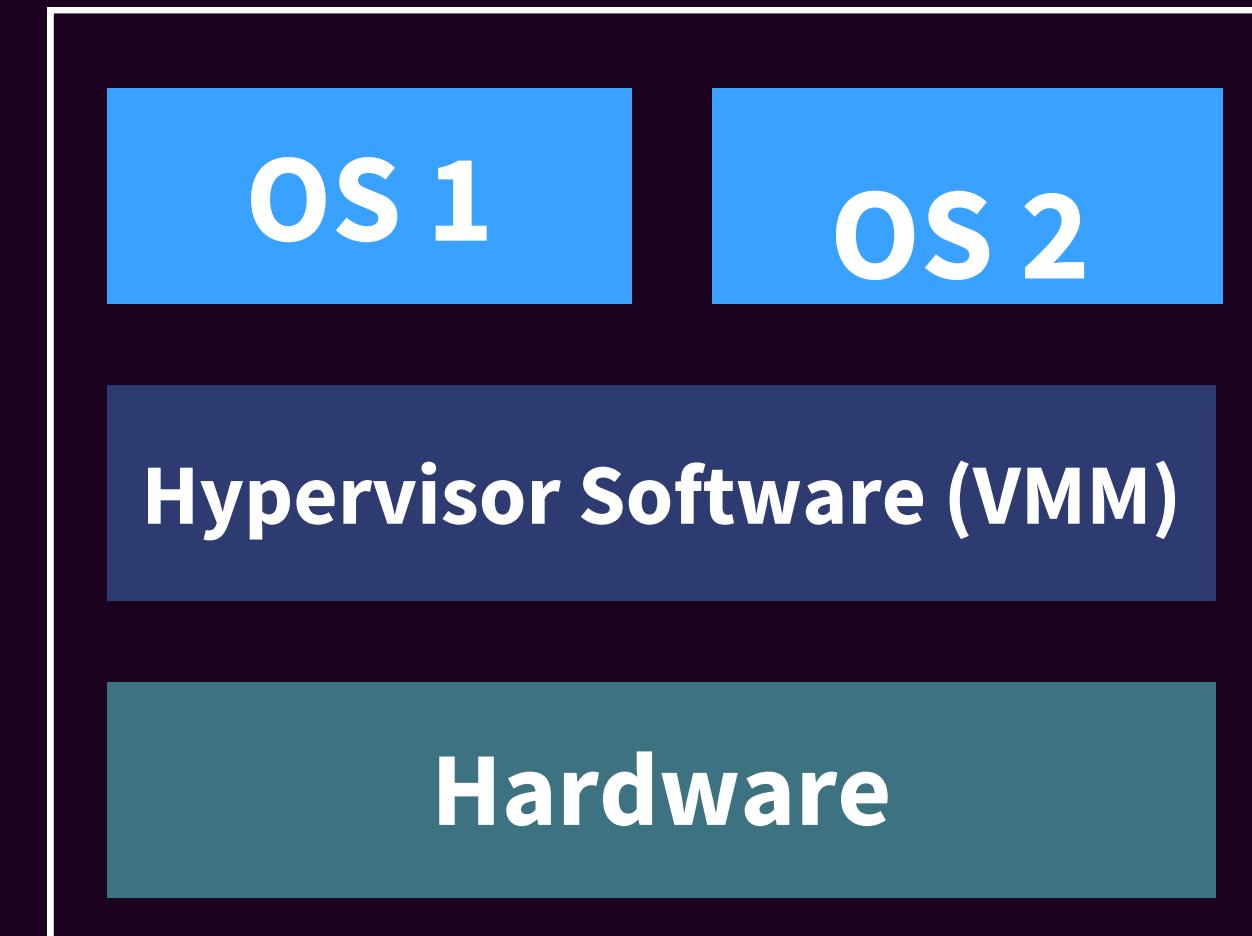
It allows one host computer to support multiple guest VMs by virtually sharing its resources. #03

Types of Hypervisors



Hosted Architecture

Ex. Oracle VM VirtualBox



Bare-Metal Architecture

Ex. VMware ESXi.

Type-1: Bare metal hypervisor

- Runs directly on the host's hardware to manage guest operating systems
- Very efficient because they have direct access to the physical hardware resources

Type-2: Hosted Hypervisor

- Software is installed on an operating system
- Hypervisor asks the operating system to make hardware calls

Physical Server VS Virtualisation

- | | |
|---|--|
| <ul style="list-style-type: none">➤ Resources are not shared between multiple users➤ Hardware is used directly by an OS➤ Includes memory, hard-drive, processor, network connection and OS➤ It is used to run a single instance of an OS | <ul style="list-style-type: none">➤ Resources are shared between multiple users➤ A hypervisor manages the virtualized resources➤ It is software based that emulates all the functions of a physical server➤ Runs an independent OS on top of the hypervisor |
|---|--|

WHAT IS CONTAINERIZATION?

Developer:

It works on my
computer

Product Manager:

Yes, but we are NOT going
to give your computer to
the customer



 [Product Interview](#)

Container

A container is a standard unit of software that packages up code and all its dependencies, so the applications runs quickly and reliably from one computing environment to another.

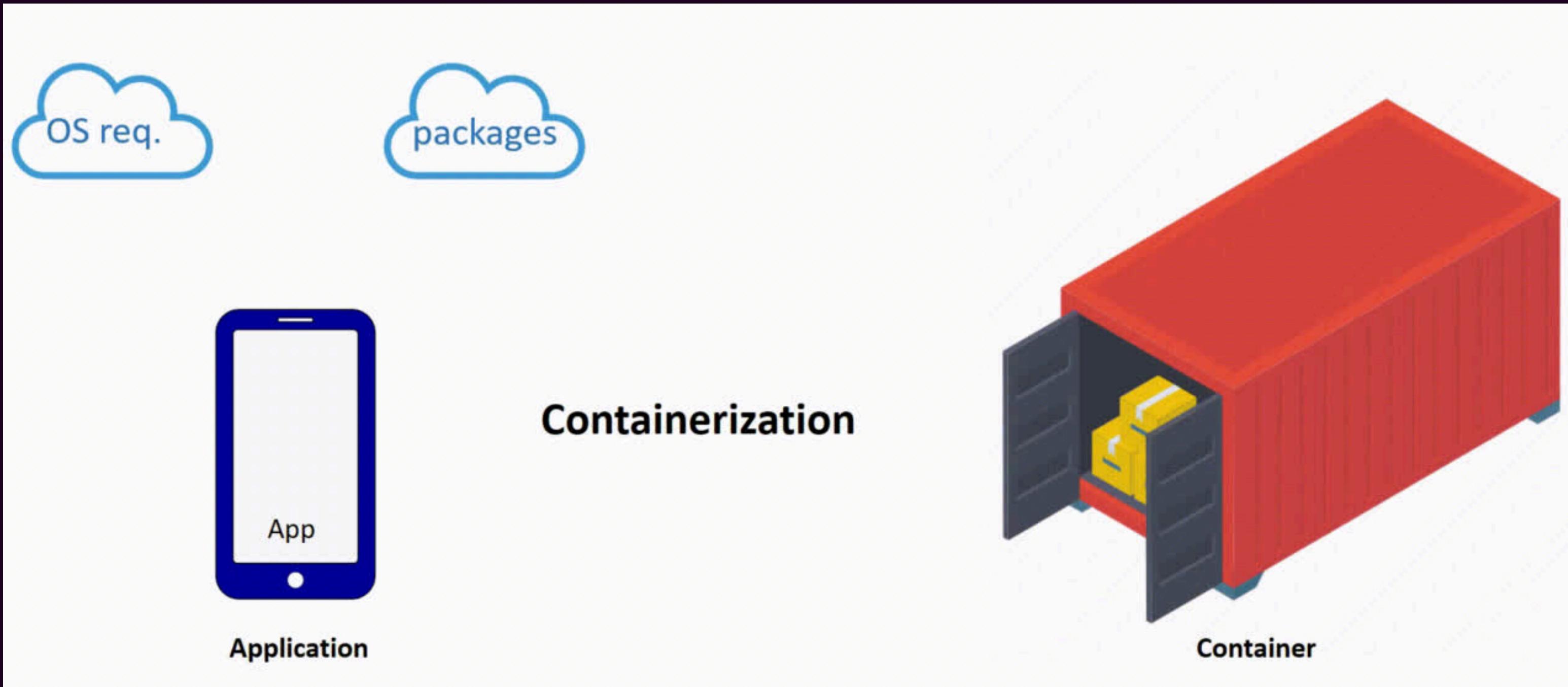


Containerization

- Containerization is a process that bundles an application code with all the files and libraries it needs to run on any infrastructure.
- It is Operating System level Virtualization.
- Containers are said to share the host system's kernel with other containers.



Containerization



What is Podman?

- ▶ Podman (the POD manager) is an open source tool for developing, managing, and running containers on your Linux systems.
- ▶ Podman was initially released in September 2017 by Red Hat.
- ▶ Podman is primarily written in the Go programming language



How Podman Works ?

- Process-based approach: Manages containers as individual processes, eliminating the need for a central daemon.
- Uses runC: Utilizes the industry-standard runC container runtime to execute containers.
- Provides a command-line interface: Allows users to interact with containers via the command line.

Usecases of Podman ?

- DevOps and CI/CD pipelines: Useful for building and testing applications in automated workflows.
- Security-focused environments: Preferred in environments with strict security requirements due to its daemonless operation.
- Multi-container applications: Suitable for managing complex applications composed of multiple containers.

Why Podman?

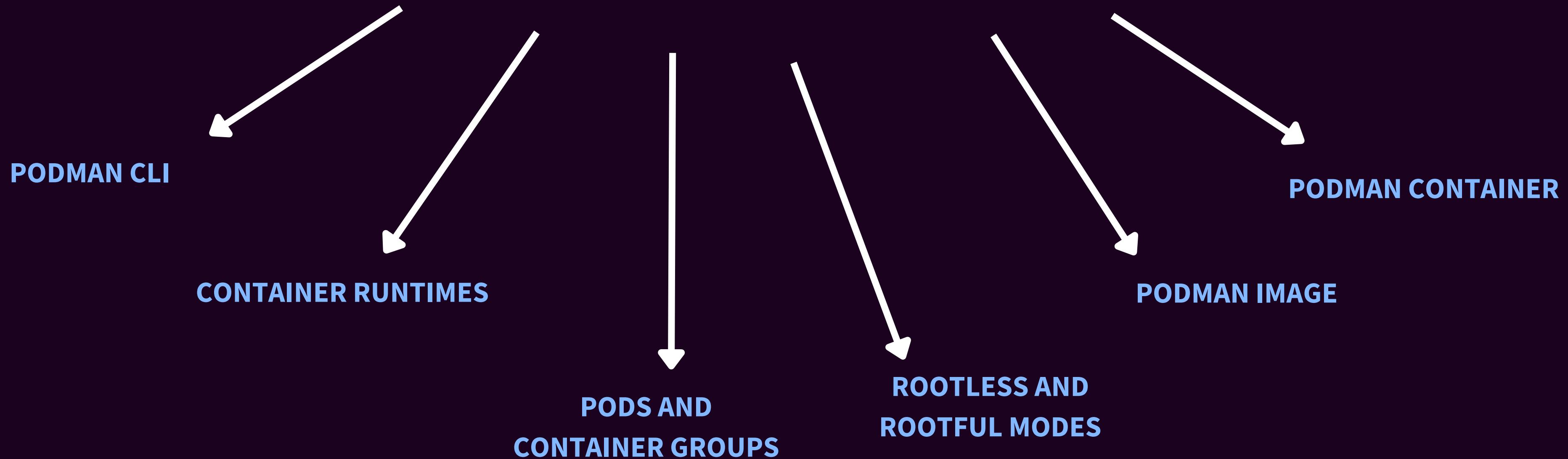
Daemonless Operations

Enhanced Security

Compatibility with Docker

Support for Pod-Based Workflows

PODMAN COMPONENTS

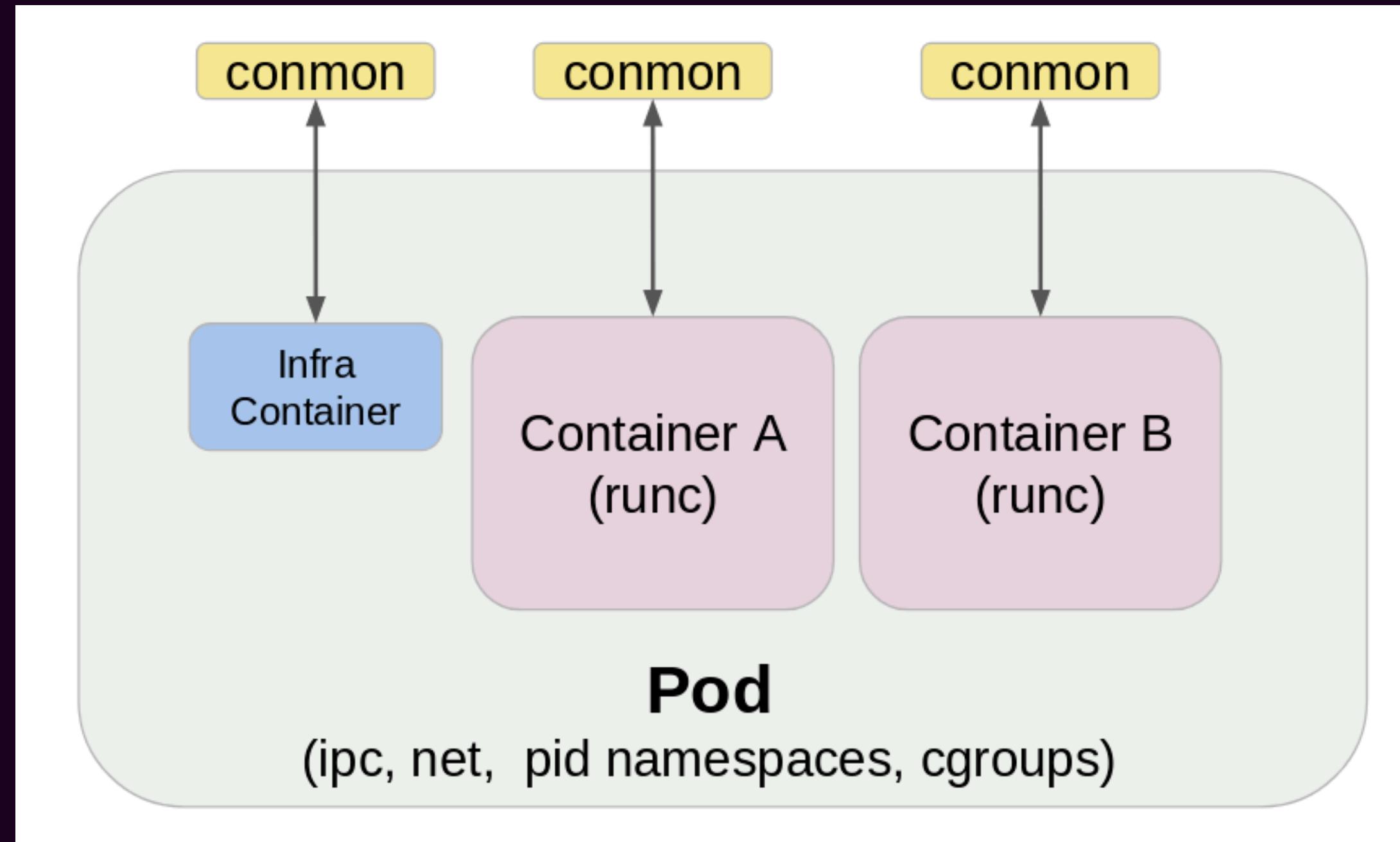




Flow For Building Podman Container



What are PODS?



Docker vs Podman

ARCHITECTURE

Docker

- Docker uses a client-server architecture. It relies on a daemon process (`dockerd`) that runs in the background and manages containers.
- The Docker CLI communicates with this daemon to issue commands.

Podman

- Podman, on the other hand, uses a daemonless architecture.
- Each Podman command directly communicates with the container engine, which makes it more lightweight.

Rootless Operation

Docker

- By default, Docker requires root privileges to run. This means that non-root users need to use sudo to interact with Docker.
- There is a rootless mode available, but it has some limitations.

Podman

- Podman is designed to be used by both root and non-root users.
- It does not require root privileges to run containers.

Networking

Docker

- Docker provides a built-in networking system that allows containers to communicate with each other and the outside world.
- It also supports various network plugins for more advanced networking configurations.

Podman

- Podman leverages the host system's networking directly, which can simplify networking configuration.
- It can also utilize plugins for more complex networking setups.

Operating System Support

Docker

- Docker is well-supported on various Linux distributions and also has versions available for Windows and macOS.

- However, on non-Linux platforms, it uses a lightweight VM (Virtual Machine) to run containers.

Podman

- Podman is primarily focused on Linux systems.

- It is designed to run natively on Linux without the need for a VM.

VAGRANT

The major problem with virtual machines are:

1. OS Installations
2. Time Consuming
3. Manual Setup
4. Tough to replicate VMs
5. Documentation of Multi VMs



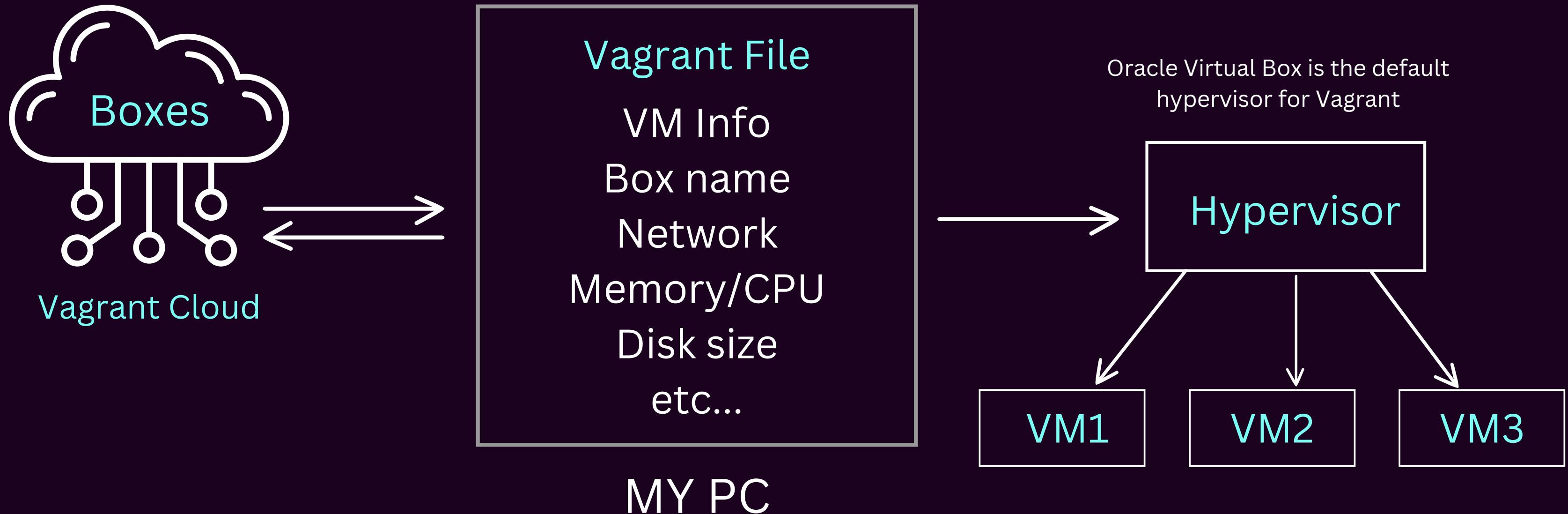
VAGRANT

Vagrant solves all these problems as :

1. OS Installations are not required
2. VM setup through images (Vagrant Boxes)
3. Images/Boxes are available on the vagrant cloud
4. Manage VMs with a file i.e. Vagrant File
5. Can make VM changes automatic through Vagrant File



VAGRANT WORKFLOW



Common Commands :

► **vagrant init :**

This command is used to create a vagrant file which has all the configurations of the VM. The vagrant file is editable according to your needs

► **vagrant up :**

Firstly it will search for the specifications of VM provided in your local machine and if not found it will pull the box from the cloud and setup the VM

► **vagrant halt/reload/destroy :**

It will perform the given actions on the VM. If the resources are being under utilised then it will destroy or halt the VM

► **vagrant ssh :**

It is used to login into your previous linux VMs

Thank you!