2.Virtualization

Contents page

2. virtualization

2.1 What is Virtualization?

Virtualization is the creation of a virtual (rather than actual) version of something, such as a hardware platform, operating system, storage device, or network resources. It's essentially creating a simulated environment within a real one. In the context of operating systems, it means running multiple operating systems (called "guest OSs") on a single physical machine (called the "host"). Each guest OS operates independently, as if it were running on its own dedicated hardware.

2.2 Why Virtualization?

• Resource Optimization: Maximizes hardware utilization. Instead of a server running at 10% capacity, virtualization allows multiple virtual machines (VMs) to share the same hardware, increasing overall resource usage.

• Cost Savings: Reduces capital expenditure (CAPEX) by requiring fewer physical servers. Also, reduces operational expenditure (OPEX) through lower power consumption, cooling, and maintenance costs.

• Isolation and Security: VMs are isolated from each other. If one VM is compromised, it doesn't necessarily affect other VMs on the same host. This enhances security.

• Flexibility and Agility: VMs can be easily created, cloned, backed up, and moved between physical servers. This provides greater flexibility for scaling resources and responding to changing business needs.

• Testing and Development: Ideal for testing software in different environments without needing separate physical machines. Allows developers to quickly create and destroy VMs for experimentation.

• Legacy Application Support: Can run older operating systems and applications that are not compatible with modern hardware or OSs.

• Disaster Recovery: VMs can be easily backed up and restored, providing a fast and efficient disaster recovery solution.

2.3 How Virtualization Works :

Virtualization is achieved through a piece of software called a hypervisor (also known as a virtual machine monitor or VMM). The hypervisor sits between the physical hardware and the guest operating systems, managing resource allocation and providing an abstraction layer. There are two main types of hypervisors:

• Type 1 (Bare-Metal Hypervisors): These run directly on the hardware. Examples include VMware ESXi, Microsoft Hyper-V Server (core installation), and Xen. They are more efficient because they don't have the overhead of a host OS.

• Type 2 (Hosted Hypervisors): These run on top of an existing operating system (like Windows, macOS, or Linux). Examples include VMware Workstation, VirtualBox, and Parallels Desktop. They are easier to set up and use for desktop virtualization.

Key Components:

1. Hardware: The physical server with CPU, memory, storage, and network interfaces. Modern CPUs often have hardware virtualization extensions (Intel VT-x or AMD-V) that improve performance.

2. Hypervisor: The virtualization software that manages the VMs.

3. Virtual Machines (VMs): Each VM consists of:

- Virtual Hardware: Simulated CPU, memory, network interface, disk controller, etc.
- Guest OS: The operating system running inside the VM (e.g., Windows, Linux, macOS).
- Applications: The software running on the guest OS.

Process:

1. The hypervisor allocates resources (CPU, memory, disk space) from the physical hardware to each VM.

2. The guest OS believes it has direct access to the hardware, but the hypervisor intercepts and translates hardware requests.

3. The hypervisor ensures that VMs are isolated from each other and that resources are fairly shared.

4. Hardware virtualization extensions in the CPU can assist the hypervisor in these tasks, improving performance.