

# VIRTUALIZATION SOFTWARES



**PRESENTED BY**

Rutvi Shah

Vandeet Shah

Raj Randive

# VIRTUALIZATION TECHNIQUES

# 1. Full Virtualization

- This technique emulates the complete hardware environment
- Allows any guest OS to run on the VM
- VMware and VirtualBox utilise Full Virtualization

## 2. Para Virtualization

- This technique modifies the guest OS to interact with the Hypervisor
- Offers better performance than full virtualization but requires compatibility between guest OS and Hypervisor
- Xen and KVM utilise Para Virtualization

# 3. Hardware assisted Virtualization

- This technique leverages special features built into modern processors to improve virtualisation performance and security
- Features like translating guest OS memory addresses to physical memory addresses.
- VMware ESXi and Xen utilise hardware assisted virtualisation

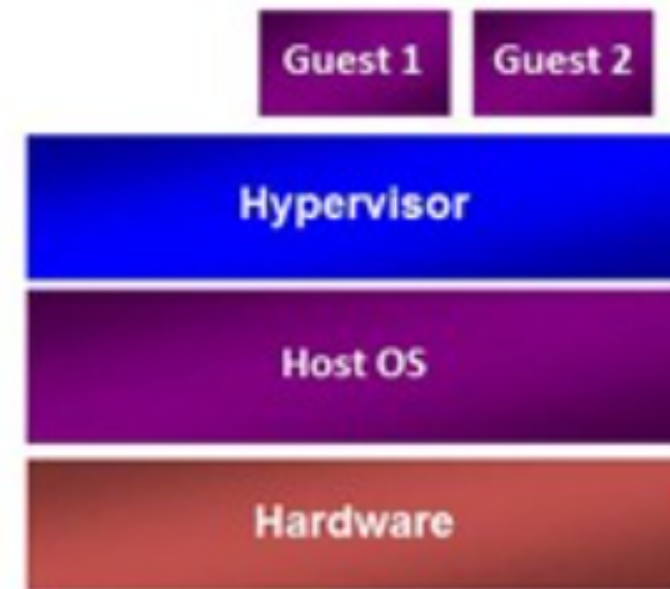


# ARCHITECTURAL DIFFERENCES

# Hypervisor Design:

Two approaches

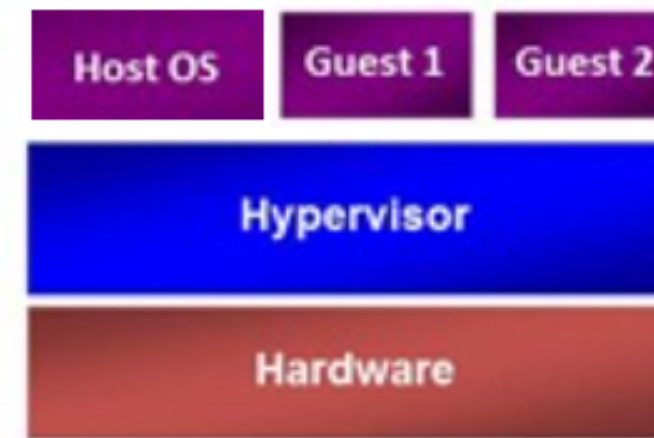
## Type 2 Hypervisor



Examples:

Virtual PC & Virtual Server  
VMware Workstation  
KVM

## Type 1 Hypervisor



Examples:

Hyper-V  
Xen  
VMware ESX

VMware:

Type: VMware primarily uses hardware virtualization, which involves the use of a type-1 hypervisor.

QEMU (Quick Emulator):

Type: QEMU is a versatile emulator that can be used as both a Type 1 and a Type 2 hypervisor i.e it can support both hardware virtualization and software virtualization (emulation).

## Xen Hypervisor:

Type: Xen is a Type 1 hypervisor. Xen uses a combination of hardware virtualization and para-virtualization. Hardware virtualization is employed for certain privileged instructions, while para-virtualization requires modifications to the guest operating systems to make them aware of the virtualization layer.

## Oracle VirtualBox:

Type: VirtualBox is a Type 2 hypervisor. VirtualBox primarily uses software virtualization, also known as hardware-assisted virtualization. It relies on the host operating system for resource management and uses dynamic binary translation to run guest operating systems on the host hardware.

## KVM (Kernel-based Virtual Machine):

Type: KVM is a Type 1 hypervisor that is integrated into the Linux kernel. KVM primarily uses hardware virtualization, allowing the Linux kernel to act as a hypervisor. It leverages virtualization extensions (Intel VT-x or AMD-V) for efficient virtual machine execution.



# What are the Methodologies??

## 1) VMware:

uses a technique called hardware-assisted virtualization where it leverages features provided by modern CPUs (Intel VT-x or AMD-V) to improve the efficiency of virtualization

## 2) QEMU:

QEMU utilizes **dynamic binary translation** to emulate the guest CPU instructions

Dynamic binary translation is particularly useful in scenarios where direct execution of guest instructions on the host is not feasible due to architectural differences. This process occurs dynamically at runtime, translating the binary code of guest instructions to equivalent instructions that can be executed on the host system.

### 3) KVM:

KVM leverages hardware virtualization extensions (Intel VT-x or AMD-V) to provide a direct interface to the host's CPU for virtualization. It works as a kernel module and relies on the Linux kernel's scheduler and memory management.

### 4) XEN:

Xen uses a microkernel design where the hypervisor runs as a separate, lightweight kernel (the "Xen hypervisor") above the hardware. It allows multiple operating systems to run concurrently in isolated domains (known as Xen domains or VMs). Each domain has its own kernel, and the hypervisor manages their access to hardware resources.

### 5) Oracle:

VirtualBox runs as a user-space application on the host operating system. It uses a combination of dynamic binary translation and various device drivers to emulate hardware and provide a virtualized environment for guest operating systems

**Thank You**