Virtualization

It is the technique of splitting a physical resource into as many logical resource as we want example: Cpu,Memory

**Virtualization**

* Creates multiple "virtual" machines (VMs) on a single physical machine.
* Each VM runs its own OS and apps, isolated from others.
* **Virtualization** is the process of creating a virtual version of a physical resource, such as a server, operating system, storage device.
* Virtualization is a technology that transforms hardware into software.

### What Is A Virtual Machine: VM, is a digitized version of a physical computer.

### Virtual machines can run programs and operating systems, store data, connect to networks, and do other computing functions.

* A virtual machine (VM) is **a software-based computer that exists within another computer's operating system**, often used for the purposes of testing, backing up
* A VM is a software-based computer that runs an OS and apps like a real machine, but is managed by a hypervisor.

### ****Advantages of Virtual Machines****

1. **Isolation:** Each VM is separate — safe from crashes or malware in others.
2. **Run multiple OSs:** Linux on Windows? No problem.
3. **Snapshots:** Save VM state, revert anytime.
4. **Cost-saving:** Use one physical machine for many VMs.
5. **Safe testing:** Perfect for testing apps or updates without risk.
6. **Portability:** Move VMs between systems or cloud easily.

### ****Disadvantages of Virtual Machines****

1. **Slower than physical machines:** Because they share hardware.
2. **Resource-heavy:** Each VM needs RAM, CPU, disk space.
3. **Complex setup for large scale:** Especially with networking.
4. **Requires hypervisor knowledge:** Some learning curve.
5. **Less efficient than containers:** VMs use full OS; containers are lighter.

### What is a Hypervisor?

A **hypervisor is the software that creates and manages VMs**.

Hypervisor is the software or firmware that creates and run virtual machine. virtual machine also called virtual machine manager (VMM).

**Type 1 (Bare-metal)or Native Hypervisor:** Firmware

* Runs **directly on the hardware**
* for Enterprise..(fast, used in data centers).
* Examples: VMware ESXi, Microsoft Hyper-V, KVM …. Vmware suite 🡪vsphere
* Runs directly on hardware (used in production).

**Type 2 (Hosted):**

* Runs **on top of an OS** (easy for personal use).
* Examples: VirtualBox, VMware Workstation, Parallels …..
* Runs on a host OS (used on laptops/dev machines).

Window server

Ubuntu

Window server

OS

OS

AD

Mac

Mac

OS

OS

VM

VM

VM

VM

VM

VM

VM

Hypervisor ESXI

Hardware

Operating system

Hardware

**Type 1 (Bare-metal)or Native Hypervisor**

**Type 2 (Hosted):**

### Types of Virtualization

* **OS Virtualization** – Multiple OSs on one machine using VMs.
* **Application Virtualization** – Run apps without installing on host.
* **Storage/Network Virtualization** – Pool resources across multiple devices.

### Bare-metal vs Hosted

|  |  |  |
| --- | --- | --- |
| **Feature** | **Bare-metal (Type 1)** | **Hosted (Type 2)** |
| Performance | Better | Lower (some overhead) |
| Use Case | Servers, cloud | Developers, learners |
| Runs on | Direct on hardware | Inside an OS |

### VM vs Container?

|  |  |  |
| --- | --- | --- |
| **Feature** | **VM** | **Container** |
| OS Layer | Has full guest OS | Shares host OS |
| Size | Heavy (GBs) | Lightweight (MBs) |
| Speed | Slower to start | Very fast to start |
| Isolation | Strong (full OS separation) | Less isolated |
|  |  |  |

### How do VMs support disaster recovery?

**A:** You can take **snapshots** of VMs and **restore** them quickly in case of failure.

### Why use VMs in the cloud?

**A:** Cloud providers (AWS, Azure) run VMs on data center servers to host apps and services. They offer flexibility, scalability, and isolation.

## Virtualization vs Containers vs Cloud

|  |  |  |  |
| --- | --- | --- | --- |
| **Feature** | **Virtual Machine** | **Container** | **Cloud** |
| OS | Full OS | Shared OS | Uses both VMs & containers |
| Use Case | Isolation, legacy apps | Microservices | Hosting, scalability |
| Examples | VMware, Hyper-V | Docker | AWS EC2, Azure VM |

* **VM = Your own house** (independent, isolated)
* **Container = Apartment in shared building** (faster, lighter)
* **Cloud = Renting house/apartment on-demand** (on someone else's land)

### Types of Virtualization:

|  |  |
| --- | --- |
| **Type** | **Purpose** |
| **Server** | Run multiple server OSes on one machine. |
| **Desktop** | Run different desktop environments (e.g., test apps safely). |
| **Network** | Simulate networks for testing or optimization. |
| **Storage** | Combine multiple storage devices into one. |
| **Application** | Run an app in a virtual environment without installing it natively. |

## ****Key Differences Table****

|  |  |  |
| --- | --- | --- |
| **Feature** | **Virtual Machines** | **Containers** |
| Boot time | Minutes | Seconds |
| OS per instance | Full OS | Share Host OS |
| Isolation | Strong | Moderate |
| Size | GBs | MBs |
| Use case | OS-level testing, legacy apps | Microservices, DevOps |

​**VMware**: holds a significant position in the virtualization market, with an estimated 42.97% share in the virtualization platform segment as of 2025 .

* VMware offers better stability and flexibility compared to XenServer and Hyper-V.
* VMware vSphere is the brand name for VMware’s suite of virtualization products. It was previously known as VMware Infrastructure.
* vSphere includes the following components:  
  → VMware ESXi  
  → vCenter Server  
  → vSphere Client  
  → vSphere Web Client
* Other key features include:  
  → vMotion  
  → High Availability (HA)  
  → Distributed Resource Scheduler (DRS)  
  → Fault Tolerance (FT)

**Difference between vCenter Server and vSphere Client:**

* **vSphere Client** is a GUI tool for remotely connecting to and managing a single ESXi host from a Windows PC, allowing basic VM management and configuration.
* **vCenter Server** is needed to manage multiple ESXi hosts from a centralized console, enabling advanced features like vMotion, HA, DRS, and Fault Tolerance.
* **vCenter Server** provides richer features and greater control compared to vSphere Client, supporting centralized management for multiple hosts and VMs.
* To access features like DRS, HA, vMotion, and Fault Tolerance, **vCenter Server** is essential.
* **vCenter Server 6.7** offers both Flash and HTML5 interfaces for management.

**VMware High Availability (HA):**  ensures minimal downtime during ESXi host failure by automatically restarting VMs on other available hosts, keeping them operational with little disruption.

**Migration** is the process of moving a virtual machine from one host or datastore to another within a VMware environment. This helps optimize resources and maintain performance.

**Types of Migration:**

* **Cold Migration**: Moving a VM while it is powered off.
* **vMotion**: Live migration of a running VM from one host to another without downtime.
* **Suspended Migration**: Migrating a VM that is in a suspended state.
* **Virtual to Virtual (V2V)**: Migrating a VM from one virtual environment to another.
* **Physical to Virtual (P2V)**: Converting a physical server into a virtual machine.

A **Datastore** stores VM files, log files, virtual disks, and ISO images. There are two types:

* **VMFS (VMware File System)**
* **NFS (Network File System)**

**Cold Migration:**

* Involves moving a VM to another host while it is powered off.
* The VM must be powered off during migration.
* Cold migrations are more flexible than vMotion, allowing VM movement between data centers as long as both are under the same vCenter Server instance.
* The chances of failure are lower compared to hot migration (vMotion). Hot means running

**Suspended Migration: Mostly done in Troubleshooting**

* Involves migrating a VM that is in a suspended state (like a paused state).
* The VM resumes from the exact point where it was suspended.
* Both suspended migration and vMotion are considered "hot" migrations since the VM is still running during the process.
* The primary reason for suspending a VM is for **troubleshooting** on an ESXi host.

**vMotion:**

* Involves migrating a **powered-on** VM from one ESXi host to another without causing any downtime.
* **vMotion** moves a running VM to a different ESXi host within the same cluster.
* **Storage vMotion** migrates the VM’s files between datastores while the VM remains powered on.
* **vMotion** is also known as **Live Migration**.

**P2V Migration (Physical to Virtual):**

* Converts a physical machine (e.g., a physical web server) into a virtual machine.
* Tools like **VMware Converter** are used to create a VM copy of the physical server on an ESXi host.

**V2V Migration (Virtual to Virtual):**

* Similar to P2V, but the source is already a virtual machine.
* For example, migrating a VM from **Hyper-V** or **VMware Workstation** to **ESXi** is considered V2V.

**High Availability & Fault Tolerance:** HA is a VMware feature that automatically restarts virtual machines on another host if the original host fails, ensuring minimal downtime.

* When a host crashes or fails, HA automatically restarts the affected VMs on another available host.
* There is minimal downtime, only during the VM restart time.
* HA automatically detects host failures and takes action without requiring admin intervention.
* It's a fully automated process designed to reduce service disruption.
* No passive standby ESXi host or extra VM is required.
* If a host fails, the affected VM restarts on any other available running host.
* **HA does not use vMotion.**
* To use HA, it must be enabled in the **Cluster Settings** in vCenter.

**Requirements for HA:**

* A **Cluster** created in vCenter
* **Shared Storage** accessible by all hosts
* A configured **vCenter Server**
* HA must be enabled in the **Cluster Settings**

**Key Points:**

* No need for a passive standby host or extra VMs
* VMs restart on any available host after a failure
* **HA does not use vMotion**
* Fully automated and requires no manual intervention during failure

**Before HA:**  
Before HA, if a single ESXi host failed, all the virtual machines on it would go down. This was known as the "all your eggs in one basket" issue, which made some companies hesitant to adopt virtualization.

**With HA:**

* **HA continuously monitors** host health and ensures that in case of failure, affected VMs can be restarted on other hosts.
* **Resource Check:** HA reserves spare capacity in the cluster to guarantee that VMs can restart after a host failure.

**Pre-requisites for VMware vSphere HA:**

* All hosts must be **licensed** for VMware HA.
* At least **two ESXi hosts** are required in the cluster.
* Each host must have a **unique hostname**.
* Hosts should be configured with **static IP addresses**. If using DHCP, ensure IPs persist across reboots.
* Virtual machines must be located on **shared storage**, not local storage—otherwise, they can't be failed over during a host failure.

**Additional HA Requirements & Architecture:**

* All hosts in a VMware HA cluster must have **DNS properly configured**.
* **HA uses a Master-Slave architecture**:
  + When HA is enabled on a cluster, an **election process** takes place among all ESXi hosts.
  + The host with access to the most datastores and VMs typically becomes the **Master**.
  + The remaining hosts act as **Slaves**.
  + If the Master host fails or crashes, a **new election** is automatically triggered to select a new Master.

**HA Failover Time:**  
In a setup with **64 hosts and 6000 VMs**, the total time from when the vCenter Server VM stopped responding to when the vSphere Web Client became active again was approximately **460 seconds (about 7 minutes)**.

* **HA response time**: Around **30–40 seconds** to detect failure and start the failover process.

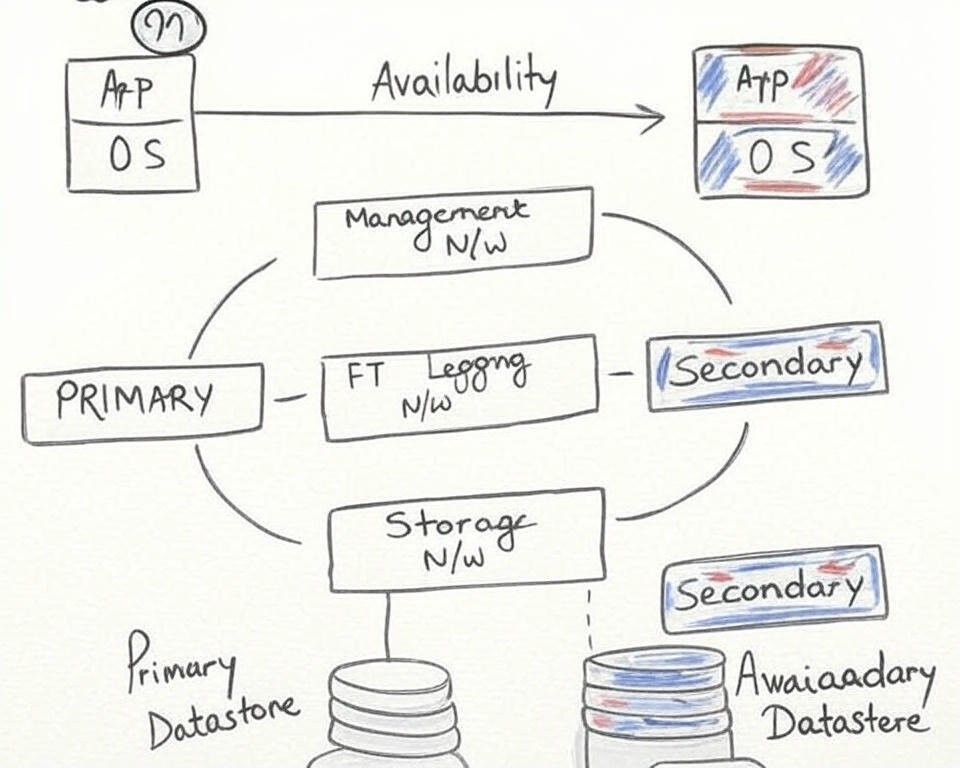
**How HA Works:**  
HA operates at the **ESXi host level**. If any ESXi host fails, **HA automatically restarts the affected virtual machines (VMs)** on another available host in the cluster—ensuring minimal downtime and service continuity.

**Fault Tolerance (FT):**

* The goal of FT is similar to HA, but **FT provides zero downtime and no restarts**, ensuring **continuous availability**.
* Designed for **mission-critical applications** like robotic surgery, autopilot systems, or spacecraft missions.
* Uses **VMware FT Lockstep Technology** to keep the primary and secondary VMs in sync.
* A **secondary VM** is created on a different host using **Distributed Resource Scheduler (DRS)**.
* The secondary VM is a **real-time replica** of the primary VM.
* **FT VMs cannot run on the same host** to prevent total loss in case of host failure.

**Lockstep Technology in Fault Tolerance (FT):**

* The **Primary and Secondary VMs work in lockstep**, meaning the current state and events of the primary VM are continuously sent to the secondary VM.
* If the **primary VM fails**, the **secondary VM takes over instantly** with zero downtime.
* FT requires an **extra standby VM**, making it a **costlier solution**, but ideal for critical systems where even a few seconds of downtime is unacceptable.



**More on Fault Tolerance (FT):**

* FT avoids the **"split-brain" scenario**, where both primary and secondary VMs could become active after a failure—leading to data conflicts.
* The **primary and secondary VMs exchange heartbeats continuously** to monitor each other's status and ensure FT is maintained.
* FT **works best with DRS (Distributed Resource Scheduler)** but **requires FT to be enabled on supported hardware**.

**Virtual Machine Template:**

* A **VM Template** is a master copy of a fully configured virtual machine.
* It allows **rapid deployment** of multiple identical VMs.
* Useful for maintaining consistency across new virtual machines based on a pre-configured setup.

**Distributed Resource Scheduler (DRS):**

* **DRS** is a **cluster-level feature** managed by **vCenter Server** that balances resource usage across hosts.
* It ensures optimal performance by **automatically distributing VM workloads** based on resource availability.

**Key Capabilities of a DRS-Enabled Cluster:**

1. **Initial VM Placement** – Chooses the best host when a VM is powered on.
2. **Load Balancing** – Moves VMs across hosts to maintain balanced CPU and memory usage.
3. **Power Management** – Shuts down underutilized hosts and consolidates workloads to save power.

**How it works:**

* As VM workloads **expand and contract** over time, DRS ensures hosts are not over or underutilized.
* **vMotion is a core requirement** for DRS, enabling live migration of VMs without downtime.

**Main Goals of DRS:**

* **Keep all ESXi hosts healthy and well-utilized** by automatically balancing the load across the cluster.
* Ensure **VMs always have enough resources** (CPU, memory) to run efficiently.
* Allow **zero-downtime server maintenance** by moving VMs automatically.
* By default, **DRS checks every 5 minutes** to evaluate if workload balancing is needed.

**DRS Automation Levels:**  
DRS offers **three automation modes** to control how it manages VM placement and migrations:

1. **Manual**
   * DRS **suggests** the best host when a VM is powered on.
   * The **user must manually choose** the host and approve all migrations.
2. **Partially Automated**
   * DRS **automatically selects** a host when the VM is powered on.
   * However, **migration suggestions still require user approval**.
3. **Fully Automated:**

* DRS **automatically selects** the best host for a VM when it is powered on.
* It also **automatically performs vMotion migrations** to balance the load—no user intervention needed.
* This mode offers **maximum efficiency** and **zero-downtime optimization** of cluster resources.

**Summary of DRS Automation Levels:**

* **Manual:** Suggestions only; user decides.
* **Partially Automated:** Auto placement, manual migration.
* **Fully Automated:** Auto placement and auto migration.