Assignment : Cloud Computing

**Submitted By :- Purv Patel**

**Module 3: System Management And Public Cloud**

**1-What is virtualization and virtualization type?**

### > What is Virtualization?

Virtualization is a technology that allows multiple virtual machines (VMs) to run on a single physical server by abstracting hardware resources. It improves efficiency, scalability, and resource utilization.

Types of Virtualization

1️ Server Virtualization 🖥️

* Runs multiple VMs on a single physical server.
* Example: Hyper-V, VMware ESXi, KVM

2️ Desktop Virtualization 🖥️💻

* Virtual desktops run on a centralized server (VDI).
* Example: Microsoft Remote Desktop Services (RDS), Citrix

3️ Network Virtualization 🌐

* Virtual networks are created over physical infrastructure.
* Example: VLANs, SDN (Software-Defined Networking)

4️ Storage Virtualization 💾

* Pools multiple storage devices into a single storage system.
* Example: SAN (Storage Area Network), NAS (Network-Attached Storage)

5️ Application Virtualization 📦

* Runs applications without installation on the local system.
* Example: Microsoft App-V, VMware ThinApp

6️ Cloud Virtualization ☁️

* Virtual servers, storage, and networks hosted in the cloud.
* Example: AWS, Microsoft Azure, Google Cloud

**2-Type of hypervisor and how to manage it?**

### > Types of Hypervisors & Management

A hypervisor is a software layer that enables virtualization by creating and managing virtual machines (VMs) on a physical host.

Types of Hypervisors

1️ Type 1 (Bare-Metal Hypervisor)

* Runs directly on hardware (no OS required).
* Faster, more secure, and used in enterprise environments.
* Examples:
  + Microsoft Hyper-V
  + VMware ESXi
  + KVM (Kernel-based Virtual Machine)

2️ Type 2 (Hosted Hypervisor)

* Runs on top of an OS (e.g., Windows, Linux, macOS).
* Slower but easier to install for personal use/testing.
* Examples:
  + VMware Workstation
  + Oracle VirtualBox
  + Parallels Desktop

Managing Hypervisors

1. Microsoft Hyper-V (Type 1) Management

* GUI Tools: Hyper-V Manager, Failover Cluster Manager
* PowerShell Commands:
* Get-VM # List all VMs
* Start-VM -Name "VMName" # Start a VM
* Stop-VM -Name "VMName" # Stop a VM

2. VMware ESXi (Type 1) Management

* GUI Tools: vSphere Client, vCenter Server
* CLI Commands (ESXi Shell):
* vim-cmd vmsvc/getallvms # List all VMs
* vim-cmd vmsvc/power.on VMID # Start a VM
* vim-cmd vmsvc/power.off VMID # Stop a VM

3. KVM (Type 1) Management

* GUI Tools: Virt-Manager
* CLI Commands (Linux):
* virsh list --all # List VMs
* virsh start VMName # Start a VM
* virsh shutdown VMName # Stop a VM

4. VirtualBox & VMware Workstation (Type 2) Management

* Managed through GUI or CLI:
* VBoxManage list vms # List VMs
* VBoxManage startvm "VMName" # Start a VM
* VBoxManage controlvm "VMName" poweroff # Stop a VM

**3-Roles of virtualization in cloud computing?**

# > **Roles of Virtualization in Cloud Computing**

Virtualization plays a **crucial role** in **cloud computing** by enabling efficient **resource allocation, scalability, and security**. Below are its key roles:

### **1️ Resource Optimization & Utilization**

**Virtualization allows multiple virtual machines (VMs) to run on a single physical server**, improving resource utilization.  
 **Reduces hardware costs** by consolidating servers.

### **2️ Scalability & Flexibility**

Cloud providers can **quickly deploy or scale VMs** based on demand.  
 Ensures **efficient workload distribution** across data centers.

### **3️ Cost Efficiency**

Eliminates the need for **dedicated physical hardware**, reducing **infrastructure and maintenance costs**.  
 Supports **pay-as-you-go models** in cloud services.

### **4️ Disaster Recovery & High Availability**

Virtual Machines (VMs) can be **replicated and migrated** across servers to ensure **business continuity**.  
 Enables **automated backups and failover mechanisms** in cloud environments.

### **5️ Security & Isolation**

**Each VM is isolated**, preventing **security breaches** from spreading across multiple workloads.  
 Supports **secure multi-tenant environments** in public and private clouds.

### **6️ Multi-Tenancy Support**

Enables **multiple users (tenants) to share cloud resources** securely without interference.  
 Ensures **efficient workload management** for cloud providers.

### **7️ Easy Deployment & Management**

Allows **quick provisioning and de-provisioning** of virtual resources.  
 Supports automation through **orchestration tools (e.g., Kubernetes, Terraform)**.

**4-What is container?**

### > **What is a Container?**

A **container** is a lightweight, portable, and isolated environment that packages an **application and its dependencies** so it can run **consistently across different environments** (development, testing, production, cloud, on-premises).

### **Key Features of Containers**

✅ **Lightweight** – Uses the **host OS kernel**, unlike Virtual Machines (VMs).  
✅ **Fast Deployment** – Starts in **seconds** compared to VMs.  
✅ **Portability** – Runs the same way on **any system** (Windows, Linux, Cloud).  
✅ **Scalability** – Easily **scaled up or down** for cloud-native applications.  
✅ **Isolation** – Each container runs independently but shares the **same OS kernel**.

### **Containers vs. Virtual Machines (VMs)**

| Feature | Containers | Virtual Machines (VMs) |
| --- | --- | --- |
| OS Usage | Shares host OS kernel | Each VM has its own OS |
| Performance | Lightweight | Heavy |
| Startup Time | **Seconds** | **Minutes** |
| Resource Efficiency | **Low overhead** | High resource usage |
| Isolation | Process-level | Full OS-level |

### **Popular Container Technologies**

**Docker** – Most popular container platform.  
 **Kubernetes** – Orchestrates and manages multiple containers.  
 **LXC (Linux Containers)** – Provides system-level virtualization.  
 **Podman** – Alternative to Docker with daemon-less architecture.

### **Example: Running an Nginx Container with Docker**

docker run -d -p 8080:80 nginx

This command runs an **Nginx web server** in a container and maps port **8080** to the container's **port 80**.

**5-What is high availability and live migration in virtualization?**

### >

1️ High Availability (HA) in Virtualization

High Availability (HA) ensures that virtualized services remain operational even if a hardware or software failure occurs. It minimizes downtime by automatically detecting failures and restarting virtual machines (VMs) on a healthy host.

How HA Works:

* A cluster of servers monitors each other.
* If a server hosting VMs fails, the VMs are automatically restarted on another available server.
* HA is commonly used in cloud environments, data centers, and enterprise virtualization platforms.

Examples of HA Solutions:

* VMware vSphere HA – Automatically restarts VMs on another ESXi host.
* Hyper-V Failover Clustering – Moves VMs to another node in the cluster.
* Kubernetes HA – Ensures high availability of containerized applications.

2️ Live Migration in Virtualization

Live Migration allows moving a running virtual machine (VM) from one physical host to another without downtime.

Why Use Live Migration?  
 Load Balancing – Distributes workloads across multiple servers.  
 Hardware Maintenance – Move VMs before shutting down a server for maintenance.  
 Disaster Recovery – Ensures business continuity by transferring VMs to a backup site.

Examples of Live Migration Technologies:

* Hyper-V Live Migration – Moves VMs between Hyper-V hosts.
* VMware vMotion – Enables seamless migration of VMs between ESXi hosts.
* KVM Live Migration – Used in Linux-based virtualization for moving KVM guests.

Example Hyper-V Live Migration Command:

Move-VM -Name "VM1" -DestinationHost "Server2"

(This moves VM1 from one Hyper-V host to another without downtime.)

**6-Storage configuration –describe block storage, file storage and object storage---DAS NAS and SAN**

## > **Storage Configuration: Types & Architectures**

Storage solutions are essential for managing data efficiently in **virtualization, cloud computing, and enterprise environments**.

## **1️ Types of Storage**

### **Block Storage**

* Stores **data in fixed-size blocks** (like hard drives or SSDs).
* Blocks are controlled by **a server OS or application**.
* Used for **high-performance databases, virtual machines (VMs), and transaction processing**.
* **Example:** Amazon EBS, iSCSI, FC (Fibre Channel) storage.

**Pros:**  
 Low latency, high-speed performance.  
 Works well for structured workloads.  
 Ideal for VM storage and databases.

**Cons:**  
 More complex management.  
 Requires additional software for file access.

### **File Storage (Network File System - NFS, SMB/CIFS)**

* Stores **data as files in a hierarchical directory structure** (like a local filesystem).
* Accessible over a **network using protocols like SMB/CIFS (Windows) or NFS (Linux)**.
* Used for **file sharing, user home directories, and collaborative storage**.
* **Example:** Windows File Server, Amazon EFS, NetApp NAS.

**Pros:**  
 Easy to manage and access.  
 Good for shared storage environments.

**Cons:**  
 Slower than block storage for high IOPS workloads.  
 Less scalable compared to object storage.

### **Object Storage**

* Stores **data as objects** with metadata and unique identifiers.
* Designed for **scalability, redundancy, and cloud environments**.
* Used for **backups, media files, and cloud-based applications**.
* **Example:** Amazon S3, Azure Blob Storage, Google Cloud Storage.

**Pros:**  
 Highly scalable and cost-effective.  
 Suitable for unstructured data (e.g., images, videos, backups).

**Cons:**  
 Higher latency compared to block storage.  
 Not ideal for transactional applications.

## **2️ Storage Architectures: DAS, NAS, and SAN**

### **DAS (Direct-Attached Storage)**

* Storage is **directly connected** to a single server (e.g., internal SSDs, external HDDs).
* Used for **personal devices, small business servers, and standalone applications**.

**Pros:**  
 Fast performance (no network overhead).  
 Simple and cost-effective.

**Cons:**  
 Not shareable between multiple servers.  
 Difficult to scale.

### **NAS (Network-Attached Storage)**

* Storage is **connected to a network** and accessible via protocols like **NFS, SMB, or AFP**.
* Used for **file sharing, backup storage, and collaboration**.

**Pros:**  
 Easy to manage and share.  
 Cost-effective for file storage.

**Cons:**  
 Lower performance than SAN for high-speed applications.  
 Network-dependent (can slow down if overloaded).

### **SAN (Storage Area Network)**

* A **high-speed, dedicated storage network** using **Fibre Channel (FC) or iSCSI**.
* Used for **enterprise applications, databases, and virtualization**.

**Pros:**  
 High performance, low latency.  
 Supports block storage and is ideal for VM environments.

**Cons:**  
 Expensive and complex to set up.  
 Requires specialized hardware and expertise.

## **Conclusion**

🔹 **Block Storage** → High-performance, used for VMs & databases.  
🔹 **File Storage** → Simple, good for sharing & collaboration.  
🔹 **Object Storage** → Scalable, best for cloud and backups.

🔹 **DAS** → Direct connection, fast but limited.  
🔹 **NAS** → Shared file storage, good for teams.  
🔹 **SAN** → High-speed storage for enterprise apps.

**7-Describe storage allocation and provisioning. Storage Allocation**

### > Storage Allocation & Provisioning (Short Summary)

1️ Storage Allocation

Storage allocation is the process of assigning disk space to applications, servers, or users based on their needs.

Types:

* Static Allocation – Pre-assigned, fixed storage.
* Dynamic Allocation – Assigned as needed to optimize usage.

2️ Storage Provisioning

Storage provisioning involves configuring and managing storage resources.

Types:

* Thick Provisioning – Pre-allocated space, guarantees performance.
* Thin Provisioning – Allocates storage on demand, reduces waste.
* Automated Tiering – Moves data between SSD, HDD, and archive storage based on usage.