

## **Introduction to Cloud Computing**

Cloud Computing is a modern computing paradigm that provides on-demand access to a shared pool of computing resources such as servers, storage, databases, networking, and applications over the internet. Instead of owning physical infrastructure, users can rent resources based on usage, leading to cost efficiency, scalability, and flexibility.

## **Service Delivery Models**

1. Infrastructure as a Service (IaaS) – Provides virtualized hardware resources such as servers, storage, and networking. Example: Amazon EC2, Google Compute Engine.
2. Platform as a Service (PaaS) – Development and deployment environment without managing infrastructure. Example: Google App Engine, Microsoft Azure App Services.
3. Software as a Service (SaaS) – Ready-to-use applications delivered over the internet. Example: Google Workspace, Salesforce.

## **Deployment Models**

1. Public Cloud – Services offered over the public internet. Example: AWS, Azure.
2. Private Cloud – Dedicated to a single organization. Example: VMware Cloud.
3. Hybrid Cloud – Combination of public and private clouds.
4. Community Cloud – Shared among organizations with similar needs.

## **Characteristics of Cloud Computing**

- On-demand self-service
- Broad network access
- Resource pooling
- Rapid elasticity
- Measured service

## **Benefits of Cloud Computing**

- Cost Efficiency
- Scalability & Flexibility
- High Availability & Reliability
- Collaboration & Accessibility
- Focus on Core Business

## **Cloud Computing Platforms and Technologies**

- Amazon Web Services (AWS)
- Microsoft Azure
- Google Cloud Platform (GCP)
- OpenStack
- VMware vCloud
- Technologies: Virtualization, Containerization (Docker, Kubernetes), Serverless, Edge computing.

## **Cloud Native**

Cloud Native refers to building and running applications that fully exploit the advantages of cloud computing. It emphasizes scalability, resilience, and agility using modern practices

such as microservices, containers, DevOps, Continuous Integration (CI), and Continuous Delivery (CD).

## **Continuous Integration and Continuous Delivery**

- Continuous Integration (CI): Developers frequently integrate code into a shared repository, with automated testing. Tools: Jenkins, GitHub Actions. - Continuous Delivery (CD): Extends CI by automatically deploying code to production-like environments. Tools: Spinnaker, ArgoCD.

## **Serverless Computing**

Serverless computing is a cloud-native execution model where the provider manages resource allocation. Developers write small units of code (functions) without managing servers. Example: AWS Lambda, Azure Functions.

## **Advantages of Serverless Cloud Computing**

- Cost Efficiency - Automatic Scaling - Faster Development - High Availability - Reduced Operational Overhead

## **Popular Serverless Computing Platforms**

- AWS Lambda - Microsoft Azure Functions - Google Cloud Functions - IBM Cloud Functions - Oracle Cloud Functions

## **Disadvantages of Serverless Cloud Computing**

- Cold Start Latency - Limited Execution Time - Vendor Lock-in - Debugging & Monitoring Challenges - Not Suitable for All Workloads

## **Virtualization**

Virtualization is the process of creating a virtual version of resources such as servers, storage, networks, or operating systems. It allows multiple instances to run on a single physical machine, improving efficiency and scalability.

## **Virtualization Technology**

- Uses hypervisors (Type 1: Bare-metal, Type 2: Hosted). - Virtual Machines (VMs) - Containers

## **Virtualization Platforms**

- VMware vSphere / ESXi - Microsoft Hyper-V - Oracle VirtualBox - KVM - Citrix XenServer

## **Virtualization Techniques**

1. Full Virtualization – Complete simulation of hardware.
2. Para-Virtualization – Modified OS interacts with hypervisor.
3. OS-Level Virtualization (Containers) – Example: Docker.
4. Desktop Virtualization – Remote desktop solutions.
5. Storage Virtualization – Pools multiple storage devices.
6. Network Virtualization – Virtual networks over physical infra.

## **Pros of Virtualization**

- Efficient Resource Utilization - Cost Savings - Scalability & Flexibility - Isolation - Disaster Recovery - Testing & Development

## **Cons of Virtualization**

- Performance Overhead - High Initial Setup Cost - Complexity - Security Risks - Resource Contention