ABHYASMITRA.IN

COMPREHENSIVE UNIVERSITY STUDY NOTES

CLOUD COMPUTING

Unit I — Introduction to Cloud Computing

Topics Covered in this Unit

Cloud Fundamentals: Formal NIST Definition, Historical Evolution & Modern Importance

Comprehensive Advantages & Disadvantages of Cloud Computing (CapEx vs OpEx)

Essential Characteristics: On-demand Self-service, Broad Network Access, Resource Pooling, Elasticity & Measured Service

Categories of Clouds & Deployment Models: Public, Private, Hybrid, and Community Clouds

Cloud Service Models: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS)

Shared Responsibility Model and Extended XaaS Architectures

Cloud Computing Architecture: Layered Reference Model, Front-End vs Back-End Ecosystem

Cloud Storage Fundamentals: Distributed Storage, Block/File/Object Models, CAP Theorem & Data Management

High-Yield University Exam Question Bank with Detailed Model Answers & Unit Summary

Compiled in a structured, easy-to-understand, textbook style format
Specially curated for B.Tech / B.E. / BCA / MCA / B.Sc Computer Science & IT Students

COURSE SYLLABUS & MAPPING

[UNIT I SYLLABUS — PRESCRIBED UNIVERSITY CURRICULUM]

This section contains the official syllabus for Unit I. Verify with your university curriculum mapping.

Unit I Syllabus Breakdown:

Unit I **-** Introduction to Cloud Computing: Cloud Fundamentals: Definition, Importance of cloud computing, Advantages and Disadvantages of Cloud Computing, Characteristics, Categories of Clouds: Private clouds, Public clouds Cloud Service Models: SaaS, PaaS, IaaS, Cloud Architecture, Cloud Storage: Distributed Data Storage, Data management, Cloud Deployment Models

• Course Code: ____________________ • Semester / Year: ____________________

TABLE OF CONTENTS

1.1 Cloud Fundamentals & Overview

Formal NIST Definition: Cloud Computing

According to the National Institute of Standards and Technology (NIST SP 800-145):

"Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction."

Historical Evolution of Cloud Computing

Cloud computing did not emerge overnight; it is the natural convergence of several decades of evolution in distributed systems, virtualization, networking, and utility billing models:

- **1. Mainframe Computing (1950s–1960s):** Centralized compute mainframes (IBM) where multiple dumb terminals accessed CPU time via time-sharing mechanisms.
- **2. Client-Server Architecture (1980s–1990s):** Decentralization where personal computers (clients) requested specific resources or services from specialized central servers.
- **3. Cluster Computing (1990s):** Connecting homogeneous physical computers across high-speed LANs to function as a single high-performance computing system.
- **4. Grid Computing (Late 1990s–Early 2000s):** Aggregating geographically dispersed, heterogeneous computational resources to solve massive, complex scientific problems (e.g., SETI@home).
- **5. Utility Computing (2000s):** Packaging computing resources as a metered service, analogous to traditional public utilities like water, electricity, and telecommunications.
- **6. Cloud Computing (2006–Present):** Amazon launched EC2 and S3 in 2006, combining hypervisor-driven hardware virtualization, elastic web APIs, multi-tenant isolation, and automated pay-as-you-go provisioning.

HISTORICAL EVOLUTION TOWARDS CLOUD COMPUTING

```
+---------------------+
| Mainframe Era       | --> Centralized Time-Sharing
+---------------------+
           |
           v
+---------------------+
| Client-Server Era   | --> Distributed Enterprise Networking
+---------------------+
           |
           v
```

```
+--------------------+
| Cluster & Grid Era  | --> Heterogeneous Resource Aggregation
+--------------------+
           |
           v
+--------------------+
| Utility Computing   | --> Pay-per-use Billing Paradigm
+--------------------+
           |
           v
+-------------------------------------------------------------+
| CLOUD COMPUTING: Virtualized, Elastic, On-Demand, Multi-tenant |
+-------------------------------------------------------------+
```

Figure: Evolutionary timeline of compute paradigms leading to contemporary cloud platforms

Importance of Cloud Computing in Modern IT

Cloud computing serves as the foundational backbone of contemporary digital transformation, artificial intelligence, big data analytics, and mobile applications:

- • **Agility and Fast Time-to-Market:** Developers can spin up virtual machines, managed databases, and AI pipelines in seconds via web consoles or Infrastructure as Code (IaC) without waiting months for physical server procurement.
- • **Global Reach & Low Latency:** Hyperscale cloud providers operate data centers across dozens of worldwide geographic regions and edge locations, enabling microsecond data access for global end-users.
- • **High Availability & Business Continuity:** Automated failover, distributed data replication across isolated Availability Zones (AZs), and built-in disaster recovery mechanisms guarantee 99.99%–99.999% uptime SLAs.
- • **Democratization of Advanced Technology:** Startups and academic researchers can utilize petabyte-scale storage, TPUs, GPUs, and quantum processors without massive upfront capital investment.

Advantages and Disadvantages of Cloud Computing

A comprehensive engineering evaluation requires understanding both the operational benefits and technical trade-offs of cloud adoption:

Dimensions	Advantages of Cloud Computing	Disadvantages / Key Challenges
1. Financial & Cost Structure	Eliminates Capital Expenditure (CapEx) for physical servers and real-estate data centers; converts to predictable Operational Expenditure (OpEx). Pay only for consumed compute seconds and storage bytes.	Unmonitored cloud sprawl can lead to unexpected cloud bill shocks. Data egress fees (moving data out of cloud) can become financially prohibitive at enterprise scale.

Dimensions	Advantages of Cloud Computing	Disadvantages / Key Challenges
2. Scalability & Elasticity	Horizontal auto-scaling adjusts server capacity dynamically in response to real-time traffic spikes (e.g., Black Friday sales) and contracts during off-peak hours.	Application architecture must be designed stateless and decoupled to fully exploit horizontal elasticity; legacy monolithic software requires costly refactoring.
3. Reliability & Maintenance	Providers handle hardware maintenance, OS hypervisor patching, uninterruptible power supplies (UPS), and redundant network cooling automatically.	Outages at cloud hyperscalers (e.g., AWS us-east-1 downtime) can disrupt thousands of reliant global applications simultaneously outside user control.
4. Security & Compliance	Enterprise-grade physical security, dedicated DDoS mitigation hardware, continuous SOC 2/ISO 27001 compliance, and automated encryption at rest and in transit.	Multi-tenant resource sharing introduces potential side-channel risks, hypervisor escape vulnerabilities, and complex cross-border data sovereignty compliance (GDPR).
5. Operational Flexibility	Teams can work seamlessly from any location worldwide through ubiquitous internet connectivity and browser-based interfaces.	Total dependency on continuous, high-speed internet connectivity. High network latency can degrade performance for ultra-real-time industrial robotics.
6. Vendor Relationships	Access to cutting-edge managed AI/ML models, serverless frameworks, and specialized database engines without internal R&D overhead.	Severe Vendor Lock-in risks due to proprietary APIs (e.g., AWS DynamoDB vs Google Cloud Spanner), making cloud migration complex and expensive.

NOTE / EXAM POINTER

EXAM POINTER: In university examinations, questions frequently ask students to differentiate between CapEx (Capital Expenditure) and OpEx (Operational Expenditure). Highlight that CapEx requires large upfront fixed investments with multi-year hardware depreciation, whereas OpEx is a variable pay-as-you-go expense directly aligned with immediate business operations.

1.2 Essential Characteristics of Cloud Computing

As formally standardized in the NIST SP 800-145 framework, a computing system must possess five foundational characteristics to be classified as genuine cloud computing:

- **1. On-Demand Self-Service:** A consumer can unilaterally provision computing capabilities, such as server time, compute cores, virtual disks, and network configurations, automatically as needed without requiring human interaction with the service provider. Provisioning is executed instantly through self-service web portals, command-line interfaces (CLI), or RESTful APIs.
- **2. Broad Network Access:** Capabilities are available over the standard network (Internet or private VPN) and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, tablets, laptops, workstations, and IoT gateways).

- **3. Resource Pooling (Multi-Tenancy):** The provider's computing resources are pooled together to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. There is a sense of location independence in that the customer generally has no control or knowledge over the exact physical location of the resources (e.g., exact rack, motherboard, or data center room).
- **4. Rapid Elasticity:** Capabilities can be elastically provisioned and released, in some cases automatically, to scale rapidly outward (scale-out) and inward (scale-in) commensurate with real-time demand. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be appropriated in any quantity at any time.
- **5. Measured Service (Pay-as-you-go):** Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., CPU hours, storage gigabytes, active user accounts, bandwidth consumed). Resource usage can be continuously monitored, controlled, reported, and audited, providing transparency for both the provider and consumer.

THE 5 NIST ESSENTIAL CHARACTERISTICS OF CLOUD COMPUTING

```
                          +-------------------------+
                          |    CLOUD COMPUTING      |
                          +-------------------------+
                                       |
         +-----------------+----------+----------+-----------------+
         |                 |                     |                 |
         v                 v                     v                 v
  +--------------+  +--------------+      +--------------+  +--------------+
  |  On-Demand   |  |    Broad     |      |   Resource   |  |    Rapid     |
  | Self-Service |  |Network Access|      |   Pooling    |  |  Elasticity  |
  +--------------+  +--------------+      +--------------+  +--------------+
         |                 |                     |                 |
         +-----------------+----------+----------+-----------------+
                                      |
                                      v
                               +--------------+
                               |   Measured   |
                               |   Service    |
                               +--------------+
```

Figure: The five core architectural pillars defining true cloud environments (NIST SP 800-145)

1.3 Cloud Service Models: IaaS, PaaS, and SaaS

Cloud computing services are organized hierarchically into three primary delivery models, traditionally known as the SPI Model (Software, Platform, Infrastructure as a Service). Each model defines a distinct boundary of management responsibility between the cloud provider and the cloud consumer:

1. Infrastructure as a Service (IaaS)

- • **Architectural Scope:** The consumer is provided with fundamental computing resources—processing power, physical/virtual machines, block/file storage, and virtual networking (IP addresses, subnets, routers, firewalls).
- • **Consumer Control:** The consumer does not manage or control the underlying physical data center infrastructure but has total administrative control over operating systems (Linux/Windows), storage allocation, deployed applications, runtime frameworks, and network security ingress/egress rules.
- • **Target Audience & Use Cases:** System administrators, network architects, and DevOps engineers. Ideal for lift-and-shift enterprise migrations, custom distributed clusters, big data processing, and raw high-performance computing (HPC).
- • **Industry Examples:** Amazon EC2, Google Compute Engine (GCE), Microsoft Azure Virtual Machines, DigitalOcean Droplets, Linode.

2. Platform as a Service (PaaS)

- • **Architectural Scope:** The provider delivers a complete managed runtime environment, middleware, database management engines, operating systems, and web servers. The consumer deploys application code without managing the underlying VMs or hardware.
- • **Consumer Control:** The consumer controls only the deployed application code and configuration settings. The provider automatically manages OS security patching, runtime updates, auto-scaling, load balancing, and hardware redundancy.
- • **Target Audience & Use Cases:** Software developers, web engineers, and agile software teams. Ideal for rapid web application development, microservices, mobile app backends, and continuous integration/continuous deployment (CI/CD) pipelines.
- • **Industry Examples:** AWS Elastic Beanstalk, Google App Engine, Heroku, Microsoft Azure App Services, Red Hat OpenShift.

3. Software as a Service (SaaS)

- • **Architectural Scope:** The provider delivers fully functional, turn-key end-user software applications accessible over the internet via a web browser, desktop client, or mobile app.
- • **Consumer Control:** The consumer has zero control over infrastructure, operating systems, or application codebase. Users manage only user-specific configuration settings, data uploads, and access permissions.
- • **Target Audience & Use Cases:** Business end-users, corporate employees, and general consumers. Ideal for enterprise productivity, customer relationship management (CRM), email, collaborative document editing, and billing.
- • **Industry Examples:** Google Workspace (Gmail, Docs), Microsoft 365, Salesforce CRM, Dropbox, Zoom, Netflix, Workday.

Shared Responsibility Model across Service Models

THE CLOUD SHARED RESPONSIBILITY STACK (PROVIDER VS CONSUMER)

```
Layer / Component          |   IaaS Model   |   PaaS Model   |   SaaS Model   |
---------------------------+----------------+----------------+----------------+
Applications               |  USER MANAGES  |  USER MANAGES  | PROVIDER MNGS  |
Data & User Access         |  USER MANAGES  |  USER MANAGES  |  USER MANAGES  |
Runtime Environment        |  USER MANAGES  | PROVIDER MNGS  | PROVIDER MNGS  |
Middleware                 |  USER MANAGES  | PROVIDER MNGS  | PROVIDER MNGS  |
Operating System (OS)      |  USER MANAGES  | PROVIDER MNGS  | PROVIDER MNGS  |
Virtualization / Hypervis  | PROVIDER MNGS  | PROVIDER MNGS  | PROVIDER MNGS  |
Physical Servers & Storage| PROVIDER MNGS  | PROVIDER MNGS  | PROVIDER MNGS  |
Physical Datacenter & Net  | PROVIDER MNGS  | PROVIDER MNGS  | PROVIDER MNGS  |
```

Figure: Comparison of operational boundaries across IaaS, PaaS, and SaaS

Comprehensive Comparison Matrix: IaaS vs PaaS vs SaaS

Comparison Criteria	Infrastructure as a Service (IaaS)	Platform as a Service (PaaS)	Software as a Service (SaaS)
Core Focus	Raw computing, storage, and networking hardware abstraction.	Managed application development, execution, and runtime stack.	Ready-to-use end-user software applications over web.
Target Users	Cloud architects, DevOps, System Administrators.	Software developers, QA testers, Data Engineers.	End-users, business enterprises, consumers.
Level of Control	Highest level of control over OS, networking, and software stack.	Moderate control over application code and configurations only.	Lowest control; only manage personal user data and settings.
Provider Responsibility	Physical data centers, power, cooling, hardware, hypervisors.	Hardware, OS, middleware, runtime, scaling, database engines.	Complete application lifecycle, bug fixes, updates, security.
Cost Model	Pay per vCPU-hour, RAM GB-hour, Storage GB/month, Egress IO.	Pay per platform instance, runtime tier, or resource consumption.	Subscription per user/seat per month/year, or freemium.
Key Trade-off	High management complexity and administrative overhead.	Risk of framework and runtime vendor lock-in.	Limited customizability and dependency on vendor uptime.

1.4 Cloud Deployment Models & Categories of Clouds

Cloud deployment models specify the environmental parameters of the cloud infrastructure, including the ownership, accessibility, scale, and location of hardware assets:

1. Public Cloud

- • **Architectural Concept:** The cloud infrastructure is owned and operated by a third-party commercial cloud provider (e.g., AWS, Azure, Google Cloud) and is made available to the general public or large industry groups over the public internet on a multi-tenant basis.
- • **Key Characteristics:** Massive scalability, near-infinite resource pools, zero upfront infrastructure investment, automated multi-region replication, and standard pay-as-you-go pricing.
- • **Trade-offs:** Shared multi-tenant hardware can cause noisy-neighbor performance fluctuations; stringent regulatory requirements (e.g., banking secrets) may restrict public cloud hosting in certain jurisdictions.

2. Private Cloud (Internal / Enterprise Cloud)

- • **Architectural Concept:** The cloud infrastructure is provisioned for exclusive use by a single organization comprising multiple consumers (e.g., business units). It may be owned, managed, and operated by the organization itself (On-Premises Private Cloud) or hosted by a third party (Hosted Private Cloud).
- • **Key Technologies:** OpenStack, VMware vSphere/Cloud Foundation, Microsoft Azure Stack, Apache CloudStack.
- • **Key Characteristics:** Maximum data privacy, dedicated single-tenant hardware, full customization of security perimeters, deterministic network performance, and total adherence to strict compliance regimes (e.g., defense, healthcare).
- • **Trade-offs:** High CapEx for physical servers, specialized IT staff required for hypervisor and datacenter maintenance, limited elastic bursting capacity.

3. Hybrid Cloud

- • **Architectural Concept:** The cloud infrastructure is a composition of two or more distinct cloud infrastructures (private, community, or public) that remain unique entities, but are bound together by standardized or proprietary technology that enables data and application portability.
- • **Key Use Case — Cloud Bursting:** An organization runs its baseline core workload (e.g., customer database) in its secure private cloud. During seasonal traffic spikes, the application dynamically 'bursts' excess stateless compute workloads into public cloud VMs without buying new physical servers.
- • **Key Technologies:** Kubernetes (KubeFed), AWS Outposts, Azure Arc, Google Anthos, HashiCorp Terraform.

4. Community Cloud

- • **Architectural Concept:** The cloud infrastructure is provisioned for exclusive use by a specific community of consumers from organizations that have shared concerns (e.g., mission, security requirements, policy, and compliance considerations).
- • **Real-World Examples:** Government Community Clouds (e.g., AWS GovCloud for US Federal agencies), Healthcare research clouds for genomic data sharing, Inter-banking clearing clouds.

Feature / Parameter	Public Cloud	Private Cloud	Hybrid Cloud	Community Cloud
Ownership & Management	Third-party cloud service provider (AWS, Azure, GCP).	Single enterprise or outsourced managed provider.	Shared between enterprise and public cloud providers.	Shared among participating organizations or third party.
Tenancy Model	Multi-tenant (shared physical servers).	Single-tenant (dedicated physical infrastructure).	Mixed (single-tenant private + multi-tenant public).	Multi-tenant within a bounded, trusted community.
Security & Control	Moderate; provider controls perimeter.	Highest; custom hardware firewalls & air-gaps.	High; sensitive data on-prem, apps in public.	High; tailored to community compliance rules.
Cost Structure	Zero CapEx, variable OpEx (pay-as-you-go).	High initial CapEx + ongoing OpEx (maintenance).	Balanced CapEx and OpEx optimization.	Cost shared among participating organizations.
Elasticity & Scaling	Near-infinite, instant dynamic scaling.	Limited by physical datacenter capacity.	Dynamic hybrid bursting capabilities.	Moderate; shared community capacity.

1.5 Cloud Computing Architecture & Reference Model

Cloud architecture refers to the underlying components and sub-components required for cloud computing. It is conceptually divided into two major operational segments: the Front-End and the Back-End, connected across an IP network:

- **1. Front-End Architecture:** The client-side interface visible to users. Includes web browsers (Chrome, Edge), mobile applications, CLI tools, and fat-client desktop dashboards used to access cloud services.
- **2. Back-End Architecture:** The entire server-side cloud ecosystem managed by the cloud provider. Includes physical server blades, hypervisors, distributed storage arrays, software-defined networks (SDN), security authorization engines, load balancers, and management middleware.

HIERARCHICAL CLOUD ARCHITECTURE REFERENCE MODEL

```
+==============================================================+
|                   FRONT-END CLIENT TIER                      |
|    Web Browsers | Mobile Apps | REST APIs | CLI Terminals    |
+==============================================================+
                              |  (Internet / Direct Connect)
                              v
+==============================================================+
|                   BACK-END CLOUD ECOSYSTEM                   |
+--------------------------------------------------------------+
| 1. APPLICATION LAYER (SaaS): CRM, Email, ERP, Analytics     |
+--------------------------------------------------------------+
| 2. PLATFORM LAYER (PaaS): Java/.NET/Node Runtimes, DB Engine|
+--------------------------------------------------------------+
| 3. INFRASTRUCTURE LAYER (IaaS): VM Instances, Virtual Disks |
+--------------------------------------------------------------+
| 4. VIRTUALIZATION LAYER: Type-1 Hypervisors, KVM, Containers|
+--------------------------------------------------------------+
| 5. PHYSICAL HARDWARE LAYER: CPU Blades, SAN/NAS, SDN Switches|
+--------------------------------------------------------------+
| 6. MANAGEMENT & SECURITY: IAM, Billing, Monitoring, SLA Mgmt|
+==============================================================+
```

Figure: Layered architectural reference model of enterprise cloud infrastructure

Core Back-End Subsystems

- • **Hypervisor / Virtualization Layer:** Abstracts physical hardware into multiple isolated Virtual Machines (VMs), allowing efficient multi-tenant sharing of physical memory, CPU, and network interfaces.
- • **Cloud Storage Controller:** Manages distributed block, file, and object data across clusters of redundant storage arrays with automated RAID and cross-region replication.
- • **Management & Orchestration Engine:** Coordinates provisioning, resource scheduling, load balancing, health monitoring, and auto-scaling rules (e.g., OpenStack Nova/Neutron, Kubernetes control plane).
- • **Security & Identity Subsystem:** Enforces authentication, Role-Based Access Control (RBAC), API rate-limiting, and end-to-end cryptographic key management (KMS).

1.6 Cloud Storage: Distributed Storage & Data Management

Cloud storage is a model of networked computer data storage where data is stored in distributed logical pools across multiple physical storage servers hosted in data centers:

1. Types of Cloud Storage

- • **Block Storage:** Data is split into fixed-size raw blocks without file metadata. Each block has a unique address. Ideal for operating system boot volumes, transactional databases (MySQL, Oracle), and low-latency random read/writes. Examples: AWS Elastic Block Store (EBS), Azure Managed Disks.

- • **File Storage:** Data is stored as a hierarchical directory tree of files and folders using standard network file sharing protocols (NFS, SMB/CIFS). Ideal for shared content management, team collaboration repositories, and legacy applications. Examples: AWS Elastic File System (EFS), Azure Files.
- • **Object Storage:** Data is managed as distinct discrete objects containing raw payload data, comprehensive customizable metadata, and a globally unique identifier (UUID). Objects are accessed directly via HTTP/HTTPS REST APIs (GET, PUT, DELETE). Virtually unlimited scale, ideal for unstructured big data, media files, backups, and data lakes. Examples: Amazon S3, Google Cloud Storage, Azure Blob Storage.

Storage Paradigm	Underlying Architecture	Primary Protocols	Optimal Workloads & Use Cases
Block Storage	Raw virtual hard disks; sector-level access.	iSCSI, Fibre Channel, NVMe-oF.	OS root volumes, high-performance relational DBs, transactional systems.
File Storage	Hierarchical file directory and folder trees.	NFS (v3/v4), SMB, CIFS.	Shared home directories, media editing pipelines, content management.
Object Storage	Flat address space; object = Data + Metadata + ID.	HTTP / HTTPS REST APIs (S3 API).	Unstructured big data, static web assets, machine learning datasets, archives.

2. Distributed Data Storage Principles & Data Management

- • **Replication & Fault Tolerance:** To survive hardware failures, cloud storage engines maintain multiple synchronous or asynchronous replicas of every data block (typically 3x replication across distinct fault domains).
- • **Data Partitioning and Sharding:** Massive datasets are horizontally divided into smaller, autonomous partitions (shards) distributed across independent database nodes to prevent single-node bottlenecks.
- • **The CAP Theorem (Brewer's Theorem):** States that any distributed data store can simultaneously guarantee at most two out of three core properties:
  - **1. Consistency (C):** Every read receives the most recent write or an error.
  - **2. Availability (A):** Every non-failing node returns a valid response for every request.
  - **3. Partition Tolerance (P):** The system continues to function despite arbitrary message loss or network partitions. Since network partitions are inevitable in physical cloud networks, cloud systems must choose between CP (Consistency + Partition Tolerance, e.g., HBase, MongoDB) and

AP (Availability + Partition Tolerance / Eventual Consistency, e.g., Amazon DynamoDB, Apache Cassandra).

> **NOTE / EXAM POINTER**
> EXAM POINTER: Master the CAP Theorem trade-off for university exams. Remember that in large-scale distributed cloud systems, Partition Tolerance (P) is non-negotiable because physical network cables, switches, and routers can always fail. Therefore, system designers must decide whether to favor strong Consistency (CP) or high Availability (AP).

1.7 High-Yield University Exam Question Bank & Model Answers

Part A: Short Answer Questions (2–3 Marks)

- **Q1. State the NIST definition of Cloud Computing.**

Answer: Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction (NIST SP 800-145).

- **Q2. What is Multi-Tenancy in cloud computing?**

Answer: Multi-tenancy is an architectural principle where a single instance of software or physical hardware infrastructure serves multiple distinct customer organizations (tenants), while ensuring complete logical data isolation, security boundaries, and customized resource allocation for each tenant.

- **Q3. Differentiate between Cloud Scalability and Cloud Elasticity.**

Answer: Scalability is the static or planned ability of a system to handle increasing workload by adding resources (e.g., upgrading RAM or adding servers over months). Elasticity is the dynamic, automated ability of a system to adapt to real-time workload fluctuations by provisioning resources during traffic surges and deprovisioning them during lulls within seconds.

- **Q4. What is Cloud Bursting?**

Answer: Cloud bursting is a hybrid cloud deployment configuration where an application primarily executes on a private cloud or on-premises datacenter, but automatically 'bursts' excess compute workloads into a public cloud during peak traffic surges to maintain seamless performance.

Part B: Long Answer Questions (8–10 Marks)

- **Q5. Compare and contrast IaaS, PaaS, and SaaS with suitable architecture diagrams and real-world examples.**

Model Answer Outline:

- 1. Introduction & Formal Definition of the SPI Model.
- 2. Detailed technical breakdown of IaaS: Scope (virtual hardware, raw storage, VMs), User Control (OS, runtime, apps), Examples (AWS EC2, GCE).
- 3. Detailed technical breakdown of PaaS: Scope (managed runtime, database, OS), User Control (Application code only), Examples (AWS Elastic Beanstalk, Heroku).
- 4. Detailed technical breakdown of SaaS: Scope (turn-key software), User Control (configurations and data), Examples (Microsoft 365, Salesforce).
- 5. Comprehensive comparative table across parameters: Target users, management burden, flexibility, pricing models, and vendor lock-in risk.
- 6. Complete Shared Responsibility Stack diagram depicting the boundary between Cloud Provider and Cloud Customer.

- **Q6. Explain the four Cloud Deployment Models in detail. Discuss the factors an enterprise must consider when choosing between Public and Private Cloud.**

Model Answer Outline:

- 1. Definition and architecture of Public, Private, Hybrid, and Community Cloud.
- 2. Public Cloud analysis: Architecture, Multi-tenancy, CapEx-to-OpEx benefit, Scalability, and Security considerations.
- 3. Private Cloud analysis: Single-tenancy, On-premises vs Hosted, Air-gapped security, High upfront CapEx, and Compliance governance.
- 4. Hybrid Cloud & Community Cloud architectures: Interoperability, Cloud bursting, and industry consortiums.
- 5. Decision Framework: Data Sensitivity & Regulatory Compliance, Budget & Cash Flow (CapEx vs OpEx), Workload Predictability, IT Operational Expertise, and Latency Requirements.

1.8 Unit I Summary & Quick Revision Sheet

Core Concept	Key Takeaways & University Exam Summary
NIST Characteristics	5 Essential Pillars: On-Demand Self-Service, Broad Network Access, Resource Pooling, Rapid Elasticity, Measured Service.
SPI Service Models	IaaS (Hardware/VMs - User manages OS to Apps), PaaS (Runtime/OS managed - User manages Apps), SaaS (Complete app managed by vendor).
Deployment Models	Public (Multi-tenant, provider owned), Private (Single-tenant, dedicated), Hybrid (Private + Public orchestration), Community (Shared domain-specific).

Core Concept	Key Takeaways & University Exam Summary
Cloud Storage Types	Block Storage (OS/DB, low latency raw blocks), File Storage (NFS/SMB hierarchical trees), Object Storage (REST HTTP, flat UUID space, big data).
CAP Theorem	Distributed systems can only guarantee 2 of 3: Consistency, Availability, Partition Tolerance. Cloud networks must prioritize CP or AP.