		UNIT-V			
10	a)	Analyze the compute and storage services provided by AWS.	L4	CO4	5 M
	b)	Analyze the application lifecycle in Google App Engine.	L4	CO4	5 M
		OR			
11	a)	Analyze the communication and additional services of AWS.	L4	CO4	5 M
	b)	Describe about the Microsoft Azure core concepts.	L2	CO1	5 M

PVP 23

Code: 23AM4501B, 23DS4501B

III B.Tech - I Semester - Regular Examinations - NOVEMBER 2025

CLOUD COMPUTING

(Common for AIML & DS)

Duration: 3 hours Max. Marks: 70

Note: 1. This question paper contains two Parts A and B.
2. Part-A contains 10 short answer questions. Each Question carries 2 Marks.
3. Part-B contains 5 essay questions with an internal choice from each unit. Each Question carries 10 marks.
4. All parts of Question paper must be answered in one place.

BL – Blooms Level CO – Course Outcome

PART – A

		BL	CO
1.a)	Define the key characteristics of cloud computing.	L2	CO1
1.b)	Explain the advantages of virtualization in cloud environments.	L2	CO1
1.c)	List the major differences between IaaS, PaaS and SaaS.	L2	CO1
1.d)	List out cloud deployment models.	L2	CO1
1.e)	Differentiate SISD, SIMD, MISD and MIMD architectures.	L2	CO1
1.f)	Explain the significance of distributed memory systems in parallel computing.	L2	CO1
1.g)	Describe the concept of Inter-Cloud resource allocation.	L2	CO1
1.h)	Write the challenges of security in hybrid cloud environments.	L2	CO1

1.i)	Explain the core architectural features of Google App Engine.	L2	CO1
1.j)	List out the services provided by AWS.	L2	CO1

PART – B

			BL	CO	Max. Marks
		UNIT-I			
2	a)	Explain the characteristics and benefits of cloud computing.	L2	CO1	5 M
	b)	Explain how virtualization techniques improve performance and computing capacity.	L2	CO1	5 M
		OR			
3	a)	Discuss the role of virtualization in greening initiatives, Increased performance and computing capacity, lack of space, rise of administrative costs, underutilized hardware and software resources.	L2	CO1	5 M
	b)	Explain how to apply virtualization techniques to achieve portability and managed execution.	L3	CO2	5 M
		UNIT-II			
4	a)	Discuss the components of the cloud reference model.	L2	CO1	5 M
	b)	Explain tiered pricing, per unit pricing and subscription-based pricing models.	L2	CO1	5 M

		OR			
5	a)	Compare the characteristics of public and hybrid clouds.	L2	CO1	5 M
	b)	Write the steps to apply cloud interoperability standards, scalability and fault tolerance in cross-platform applications.	L2	CO1	5 M
		UNIT-III			
6	a)	Explain the approaches to parallel programming with examples.	L2	CO1	5 M
	b)	Analyze the levels of parallelism in computing.	L4	CO4	5 M
		OR			
7	a)	Apply message-based communication models in distributed computing.	L3	CO2	5 M
	b)	Describe how to apply service-oriented architecture in developing cloud-based web services.	L3	CO2	5 M
		UNIT-IV			
8	a)	Apply the components of an energy-efficient green cloud architecture.	L3	CO3	5 M
	b)	Analyze the importance of cloud federation stack in federated clouds.	L4	CO4	5 M
		OR			
9	a)	Apply the CIA triad and its relevance in cloud security.	L3	CO3	5 M
	b)	Explain cloud security architecture.	L2	CO1	5 M

Code No:20AM4501B,20DS4501B

III B. Tech – I Semester-Regular Examinations-NOVEMBER 2025
CLOUD COMPUTING
(Common to AIML, DS)

Duration: 3 Hours **Max. Marks: 70**

Note:

1.This question paper contains two Parts A and B.
2. Part-A contains 10 short answer questions. Each Question carries 2 Marks.
3. Part-B contains 5 essay questions with an internal choice from each unit. Each Question carries 10 marks.
4. All parts of Question paper must be answered in one place

PART-A

10X2=20M

Q No	Question	Marks Awarded
1(a)	**Define the key characteristics of Cloud Computing** Ans: Any two characteristics – 2X1=2M	2M
1(b)	**Explain the advantages of virtualization in Cloud Environment** Ans: Any two advantages- 2X1=2M	2M
1(c)	**List the major differences between IaaS, PaaS, SaaS** Ans: Any two differences – 2X1=2M	2M
1(d)	**List out Cloud Deployment Models** Ans: Any two Deployment Models-2X1=2M	2M
1(e)	**Differentiate between SISD, SIMD, MISD and MIMD architectures** Ans: Any two differences among those four architectures-2X1=2M	2M
1(f)	**Explain the significance of Distributed Memory in Parallel Computing** Ans: Importance of Distributed Memory with at least two points-2X1=2M	2M
1(g)	**Describe the concept of Inter-Cloud resource allocation** Ans: Importance of Inter-Cloud Resource allocation – 2X1=2M	2M
1(h)	**Write the challenges of security in Hybrid Cloud environments** Ans: Any two Challenges- 2X1=2M	2M
1(i)	**Explain the core architectural features of Google App Engine** Ans: Any two features of Google App Engine-2X1=2M	2M
1(j)	**List out the services provided by AWS** Ans: Any two services- 2X1=2M	2M

PART - B

5X10=50 M

Q No	Question	Marks Awarded	Total Marks
	UNIT – I		
2(a)	**Explain the characteristics and benefits of Cloud Computing**		5M
	Explanation about any three Characteristics	3 Marks	
	Explanation about any two Benefits	2 Marks	
2(b)	**Explain how virtualization techniques improve performance and computing capacity**		5 M
	Introduction/Definition of Virtualization	1 Marks	
	Overview of Virtualization techniques	2 Marks	
	Explanation about how virtualization improves Performance and computing capacity	2 Marks	
	OR		
3(a)	**Discuss the role of virtualization in Greening Initiatives, Increased performance and computing capacity, lack of space, rise of administrative costs, underutilized hardware and software costs**		5M
	Explanation on the role of Virtualization in Greening Initiatives, Increased performance and computing capacity, lack of space, rise of administrative costs, underutilized hardware and software costs	5X1=5 Marks	
3(b)	**Explain how to apply Virtualization techniques to achieve portability and managed execution**		5M
	Introduction to Virtualization and techniques	2 Marks	
	Use of techniques to achieve portability and managed execution	3Marks	
	UNIT-II		
4(a)	**Discuss the components of Cloud reference model**		5M
	Introduction to cloud reference model	2 Marks	
	Explanation of components of cloud reference model	3 Marks	
4(b)	**Explain the tiered pricing, per unit pricing and subscription-based pricing**		5M
	Explanation on Tiered Pricing	1.5 Marks	
	Explanation on Per-Unit Pricing	1.5 Marks	
	Explanation on Subscription -based Pricing	2 Marks	
	OR		
5(a)	**Compare the characteristics of public and hybrid clouds**		5M
	Any five comparisons between public and hybrid clouds	5 Marks	
5(b)	**Write the steps to apply cloud interoperability standards, scalability and fault tolerance in cross-platform applications**		5 M
	Cloud interoperability standards	2 Marks	
	Scalability and fault tolerance	3 Marks	
	UNIT-III		
6(a)	**Explain the approaches to Parallel Programming with examples**		5M
	Data Parallelism approach	2 Marks	
	Process Parallelism approach	2 Marks	
	Farmer and Worker Model	1 Mark	
6(b)	**Analyze the levels of Parallelism in Computing**		

	Introduction to levels	1 Mark	5M
	Four Levels (Large-grain, Medium-grain, Fine-grain, Very-fine grain)	4 Marks	
	OR		
7(a)	**Apply the message-based communication models in distributed computing**		5M
	Introduction to message-based communication	1 Mark	
	Models of Message based communications	4 Marks	
7(b)	**Describe how to apply Service oriented architecture in cloud-based web services**		5M
	Introduction to SOA	2 Marks	
	Application/Use of SOA in Cloud-based Web Services	3 Marks	
	UNIT-IV		
8(a)	**Apply the components of an energy-efficient in green cloud architecture**		5M
	Introduction to Green Cloud Architecture	2 Marks	
	Role of Green Cloud Components in developing Architecture	3 Marks	
8(b)	**Analyze the importance of Cloud federation stack in federated clouds**		5M
	Introduction to Cloud Federation Stack	2 Marks	
	Analyzing its role in building cloud federation stack	3 Marks	
	OR		
9(a)	**Apply the CIA triad and its relevance in cloud security**		5M
	Introduction to CIA	3 Marks	
	Application of CIA in Cloud Security	2 Marks	
9(b)	**Explain the Cloud Security Architecture**		5M
	Introduction to Cloud Security	1 Mark	
	Cloud Security Architecture-Explanation	4 Marks	
	UNIT - V		
10(a)	**Analyze the compute and storage services provided by AWS**		5M
	Introduction to AWS Services	1 Mark	
	Analyzation of Storage Services	2 Marks	
	Analyzation of Compute Services	2 Marks	
10(b)	**Analyze the application lifecycle in Google APP Engine**		5M
	Introduction to Google APP Engine	1 Mark	
	Analyzation of Life Cycle steps/stages	4 Marks	
	OR		
11(a)	**Analyze the communication and additional services of AWS**		5M
	Introduction to AWS Services	1 Mark	
	Analyzation of Communication Services	2 Marks	
	Analyzation of additional Services	2 Marks	
11(b)	**Describe about Microsoft Azure Core Concepts**		5M
	Introduction to Microsoft Azure	1 Mark	
	Core concepts of Microsoft Azure	4 Marks	

Code No:20AM4501B,20DS4501B

III B. Tech – I Semester-Regular Examinations-NOVEMBER 2025
CLOUD COMPUTING
(Common to AIML, DS)

Duration: 3 Hours **Max. Marks: 70**

Note:

1.This question paper contains two Parts A and B.
2. Part-A contains 10 short answer questions. Each Question carries 2 Marks.
3. Part-B contains 5 essay questions with an internal choice from each unit. Each Question carries 10 marks.
4. All parts of Question paper must be answered in one place

PART-A **10X2=20M**

Q No	Question	Marks Awarded
1(a)	**Define the key characteristics of Cloud Computing** **Characteristics of Cloud Computing (Any two or relevant)** • On-Demand Self-Service • Broad Network Access • Resource Pooling • Rapid Elasticity • Measured Service	2M
1(b)	**Explain the advantages of virtualization in Cloud Environment** **Advantages of Virtualization in Cloud Environment (Any two or relevant)** • Increased Security • Managed Execution • Portability	2M
1(c)	**List the major differences between IaaS, PaaS, SaaS** Major Differences between IaaS, PaaS, SaaS (Any two or relevant) 1. **Control Level:** *Infrastructure as a Service (IaaS)* gives maximum control (OS, runtime, apps), *Platform as a Service (PaaS)* gives moderate control (apps, data), while *Software as a Service (SaaS)* gives minimal control (only usage). 2. **Provider Responsibility:** IaaS providers manage hardware/virtualization, PaaS providers manage hardware + OS + middleware, and SaaS providers manage the entire software stack. 3. **User Responsibility:** In IaaS, users configure OS and applications; in PaaS, users develop and deploy apps; in SaaS, users only use the application. 4. **Target Users:** IaaS is used by system administrators, PaaS by developers, and SaaS by end users. 5. **Examples:** IaaS – AWS EC2, Azure VMs; PaaS – Google App Engine, Heroku; SaaS – Gmail; Office 365.	2M
1(d)	**List out Cloud Deployment Models** Cloud Deployment Models (Any two or relevant) • Public Cloud • Private Cloud • Hybrid Cloud • Community Cloud • Multi Cloud	2M
1(e)	**Differentiate between SISD, SIMD, MISD and MIMD architectures** **Differences (Any two or relevant)** **SISD (Single Instruction Single Data)** has one processor executing one instruction on one data—no parallelism (example: basic uniprocessor).	2M

	SIMD (Single Instruction Multiple Data) uses one instruction on multiple data elements simultaneously, enabling data-parallel processing (example: GPUs, vector processors). **MISD (Multiple Instruction Single Data)** applies multiple instructions to the same data stream mainly for reliability and fault tolerance (example: aerospace redundant systems). **MIMD (Multiple Instruction Multiple Data)** executes multiple instructions on multiple data independently across many processors, giving full parallelism (example: multicore CPUs, clusters). The key differences lie in the number of instructions and data streams processed and the level of parallelism achieved.	
1(f)	**Explain the significance of Distributed Memory in Parallel Computing** Distributed memory is significant in parallel computing because each processor has its own local memory, allowing tasks to run independently and simultaneously. This reduces memory bottlenecks and increases scalability, especially for large systems. It enables efficient handling of massive datasets by distributing them across nodes. Communication occurs only when necessary, improving performance for many parallel algorithms.	2M
1(g)	**Describe the concept of Inter-Cloud resource allocation** Inter-cloud resource allocation refers to the process of distributing computing resources across multiple cloud providers to optimize performance, cost, and reliability. It enables clouds to share workloads, storage, and services dynamically. This approach helps avoid vendor lock-in and improves scalability by using the strengths of different cloud platforms. It also ensures better fault tolerance, as workloads can shift between clouds during failures or peak demand.	2M
1(h)	**Write the challenges of security in Hybrid Cloud environments** **Challenges of Security (Any two or relevant)** • Increased Attack Surface • Inconsistent Security Policies • Loss of Visibility	2M
1(i)	**Explain the core architectural features of Google App Engine** **Google App Engine Core Architectural Features (Any two or relevant)** Application, Services, Environments, Software components, Technologies, Runtime Environments, Application Deployment, Scalability, Data Storage, Service Integration	2M
1(j)	**List out the services provided by AWS** Storage Services Compute Services Communication Services Additional Services	2M

PART – B 5X10=50 M

UNIT-I

2(a) Explain the characteristics and benefits of Cloud Computing 5M

1. **Cloud computing** is a model that delivers computing services over the internet (the "cloud") to offer faster innovation, flexible resources, and economies of scale. Instead of owning and maintaining physical data centers and servers, organizations can use cloud providers' infrastructure to access various resources—such as storage, databases, servers, networking, and software—on-demand.

Characteristics of Cloud Computing (Any two/three)

- **On-Demand Self-Service**: Users can access computing resources whenever needed without human intervention from the provider, enabling quick provisioning.
- **Broad Network Access**: Services are accessible over the internet from various devices, including smartphones, tablets, laptops, and desktops.
- **Resource Pooling**: Resources are shared across multiple users in a multi-tenant model, where computing resources are dynamically allocated based on user demands.
- **Rapid Elasticity**: Resources can be quickly scaled up or down to accommodate fluctuating workloads, allowing businesses to manage peaks and troughs in demand effectively.
- **Measured Service**: Cloud computing uses a pay-as-you-go model, where resource usage is monitored, controlled, and billed based on usage, helping users avoid upfront costs.

Benefits of Cloud Computing (Any two/three)

1. **Cost Savings**: With cloud computing, businesses avoid the costs of purchasing, managing, and upgrading hardware and software infrastructure.
2. **Scalability and Flexibility**: Cloud services can be scaled up or down as needed, providing flexibility to meet changing demands without infrastructure constraints.
3. **Business Continuity**: Cloud providers often have redundancy, backup, and disaster recovery capabilities, ensuring business operations can continue even in case of disruptions.
4. **Accessibility and Collaboration**: Cloud computing allows users to access data and applications from anywhere, facilitating collaboration and remote work.
5. **Innovation and Agility**: Cloud providers continuously update and improve services, giving businesses access to the latest technologies and tools, which enables faster development and innovation.

2(b) Explain how virtualization techniques improve performance and computing capacity- 5M

Virtualization is a technology that enables the creation of multiple virtual instances of computing resources (like servers, storage, and networks) on a single physical machine. This means that a single physical server can be divided into multiple virtual servers, each with its own operating system and applications.

Types of Virtualizations (Any three)

- **Server Virtualization:** multiple virtual servers on a single physical server, allowing for efficient resource utilization and improved server management.
- **Storage Virtualization:** Pools multiple physical storage devices into a single, logical storage pool, simplifying storage management and improving performance.
- **Network Virtualization:** Creates virtual networks on top of a physical network, enabling flexible network configurations and isolation of network traffic.
- **Desktop Virtualization:** Delivers virtual desktops to users, allowing them to access their desktop environment from any device with an internet connection, improving security and flexibility.
- **Application Virtualization:** Separates applications from the underlying operating system, enabling them to run on different hardware and software platforms

Virtualization Improves Performance & Computing Capacity(Any Two Points)

1. **Resource Efficiency:**
   - **Consolidation:** Multiple virtual machines can run on a single physical server, maximizing hardware utilization and reducing costs.
   - **Dynamic Allocation:** Resources can be dynamically allocated to meet changing demands, ensuring optimal performance and cost-effectiveness.
2. **Flexibility and Scalability:**
   - **Rapid Provisioning:** Virtual machines can be created and deployed quickly, enabling rapid scaling of resources up or down as needed.
   - **Agility:** Organizations can adapt to changing business needs without significant hardware investments or lengthy provisioning processes.
3. **Cost Reduction:**
   - **Reduced Hardware Costs:** By consolidating multiple virtual machines onto fewer physical servers, organizations can lower capital expenditures on hardware.
   - **Optimized Resource Utilization:** Efficient resource allocation minimizes energy consumption and operational costs.
4. **Improved Disaster Recovery and Business Continuity:**
   - **Snapshot and Clone Capabilities:** Virtual machines can be easily backed up and cloned, enabling rapid recovery in case of failures or disasters.
   - **Business Continuity:** Virtualization facilitates seamless failover and disaster recovery processes, minimizing downtime and data loss.
5. **Enhanced Security:**
   - **Isolation:** Virtualization isolates virtual machines from each other, reducing the risk of security breaches spreading across the entire infrastructure.
   - **Security Policies:** Granular security policies can be implemented at the virtual machine level, enhancing security posture.

(OR)

3(a) Discuss the role of virtualization in Greening Initiatives, Increased performance and computing capacity, lack of space, rise of administrative costs, underutilized hardware and software costs **-5M**

Role of Virtualization in (Any One Point from each subtopic)

Greening Initiatives:

- Virtualization reduces the number of physical servers, cutting energy use and supporting eco-friendly IT.
- Lower power and cooling needs directly reduce the carbon footprint of data centers.

Increased Performance and Computing Capacity

- Virtualization enables dynamic resource allocation, improving application speed and responsiveness.
- It allows multiple workloads to run efficiently on shared hardware, boosting overall computing capacity.

Addressing Lack of Space

- By consolidating servers, virtualization reduces the need for large physical infrastructure.
- This helps organizations save valuable floor space in data centers.

Rise of Administrative Costs

- Managing many physical servers increases costs, but virtualization centralizes control and simplifies administration.
- Automated provisioning and monitoring reduce manpower requirements, lowering operational expenses.

Underutilized Hardware and Software Costs

- Virtualization ensures idle CPU, memory, and storage are reassigned to active workloads, minimizing waste.
- This maximizes ROI by improving utilization of both hardware and licensed software.

3(b) Explain how to apply Virtualization techniques to achieve portability and managed execution **5M**

Definition

- **Virtualization** is the creation of a virtual version of hardware, software environment, storage, or network. It abstracts physical resources, enabling multiple systems or applications to run on shared infrastructure.

Major Components

1. **Guest** – The system component that interacts with the virtualization layer instead of directly with the host.
2. **Host** – The original physical environment where the guest is managed.
3. **Virtualization Layer** – Software responsible for recreating the same or different environment for the guest to operate.

Role of Virtualization

1. Increased Security

- Virtual machines provide an emulated environment, isolating guest operations from the host.
- Harmful operations can be filtered by the virtualization manager.
- Example: Java Virtual Machine (JVM) and .NET runtime sandbox applications, restricting access to host resources.

2. Managed Execution

- Virtualization enables advanced features:
  - **Sharing** – Multiple guests share the same host resources.
  - **Aggregation** – Multiple hosts combined into one virtual host.
  - **Emulation** – Different environments can be recreated for guest programs.
  - **Isolation** – Guests run in separate environments, preventing interference.

3. Portability

- **Hardware virtualization**: Guests packaged into virtual images can be moved across different machines.
- **Programming-level virtualization**: JVM or .NET assemblies run on any implementation without recompilation. This ensures flexible development and straightforward deployment across platforms.

UNIT-II

4(a) Discuss the components of Cloud reference model **5 M**

Cloud computing supports any IT service that can be consumed as a utility and delivered through a network, most likely the Internet. Such characterization includes quite different aspects: infrastructure, development platforms, application and services

Architecture (Figure Optional)

Cloud computing can be viewed as a layered stack, from hardware appliances to software systems. At the base, large-scale data centers composed of clusters or networked PCs provide computing power. This heterogeneous infrastructure may include databases and storage systems

Cloud Resources (System Infrastructure / IaaS)

- Base layer consisting of physical servers, storage, and network devices.
- Provides fundamental computing resources for higher layers.

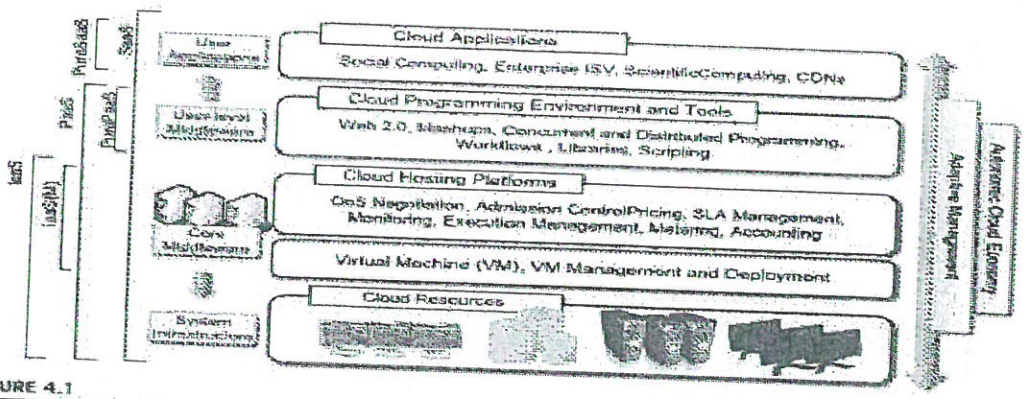


FIGURE 4.1
The cloud computing architecture.

Virtual Machines & Management (Core Middleware / IaaS)

- Virtualization layer enabling multiple VMs on physical hardware.
- Manages deployment, execution, and resource allocation.

Cloud Hosting Platforms (IaaS/M)

- Middleware for service management: QoS negotiation, SLA management, monitoring, accounting, and pricing.

Cloud Programming Environment & Tools (PaaS)

- Platforms for developing cloud applications using workflows, libraries, scripting, mashups, and concurrent programming.

Cloud Applications (SaaS)

- End-user software delivered over the cloud: social computing, enterprise applications, scientific computing, CDNs, etc.

Adaptive Management & Autonomic Cloud Economy

- Cross-layer management ensuring dynamic resource optimization and self-managing cloud operations.

4(b) Explain the tiered pricing, per unit pricing and subscription-based pricing 5 M

Cloud Computing Pricing Models

In terms of the pricing models introduced by cloud computing, we can distinguish three different strategies that are adopted by the providers:

1. **Tiered Pricing**
   In this model, cloud services are offered in several tiers, each of which provides:
   - A fixed computing specification
   - A Service Level Agreement (SLA)
   - A specific price per unit of time

This model is used by Amazon for pricing the EC2 service, which offers different server configurations in terms of computing capacity (such as CPU type and speed, memory), each with different costs per hour.

2. **Per-Unit Pricing**
   This model is more suitable in cases where the principal source of revenue for the cloud provider is determined in terms of units of specific services, such as:
   - Data transfer
   - Memory allocation

In this scenario, customers can configure their systems more efficiently according to the application needs. For example, this model is used by GoGrid, where customers pay according to RAM/hour units for the servers deployed in the GoGrid cloud.

3. **Subscription-Based Pricing**

This is the model used mostly by SaaS providers, in which users pay a periodic subscription fee for the use of the software or specific component services that are integrated into their applications

(OR)

5(a) Compare the characteristics of public and hybrid clouds **5M**

Characteristic	Public Cloud	Hybrid Cloud
Ownership & Infrastructure	Owned by third-party providers; shared infrastructure	Combines private (on-premises) and public cloud infrastructure
Cost	Pay-as-you-go; low upfront cost	Higher setup cost but optimized long-term allocation
Security & Control	Limited control; relies on provider's security	Greater control over sensitive data; customizable security
Scalability	Highly scalable; ideal for variable workloads	Flexible scaling by shifting workloads between public and private
Use Cases	Best for startups, testing, non-sensitive apps	Suitable for enterprises needing agility + compliance
Customization	Standardized services; limited customization	Tailored private cloud + public cloud for general workloads
Reliability	Dependent on provider's uptime and SLA *Example: AWS SLA guarantees*	Combines private reliability with public elasticity *Example: Azure Hybrid ensuring uptime with local + cloud failover*

5(b) Write the steps to apply cloud interoperability standards, scalability and fault tolerance in cross-platform applications **5 M**

Steps to Apply Cloud Interoperability Standards

- Adopt Standard APIs & Protocols – Use open APIs (REST, gRPC) and standard protocols (HTTP, OAuth) to ensure cross-platform compatibility.
- Use Portable VM/Container Formats – Package workloads with Open Virtualization Format (OVF) or containers (Docker/Kubernetes) for migration across providers.
- Implement Data Standards – Store/exchange data in interoperable formats (JSON, XML, SQL) to avoid vendor lock-in.
- Align with Standards Bodies – Follow guidelines from CCIF, DMTF, or Open Cloud Consortium to ensure compliance and portability.

Steps to Apply Scalability

- Design for Elasticity – Architect applications to scale resources (CPU, memory, storage) up or down automatically.
- Use Middleware & Orchestration – Employ cloud middleware (e.g., Kubernetes, OpenStack) for resource allocation, monitoring, and load balancing.
- Enable Horizontal & Vertical Scaling – Add more servers (horizontal) or increase capacity of existing ones (vertical) depending on workload.

Steps to Apply Fault Tolerance

- Redundancy & Replication – Deploy applications across multiple servers or regions to prevent single points of failure.
- Automated Failover – Configure systems to switch to backup resources when primary ones fail.

- Continuous Monitoring & Recovery – Use monitoring tools (e.g., CloudWatch, Prometheus) with automated recovery scripts to maintain uptime.

UNIT-III

6(a) Explain the approaches to Parallel Programming with examples 5 M

A sequential program is one that runs on a single processor and has a single line of control. To make many processors collectively work on a single program, the program must be divided into smaller independent chunks so that each processor can work on separate chunks of the problem. The program decomposed in this way is a parallel program.

A wide variety of parallel programming approaches are available. The most prominent among them are the following:

- Data parallelism
- Process parallelism
- Farmer-and-worker model

These three models are all suitable for task-level parallelism.

In the case of **data parallelism**, the divide-and-conquer technique is used to split data into multiple sets, and each data set is processed on different PEs using the same instruction. This approach is highly suitable to processing on machines based on the SIMD model.

Eg: Image Processing, Scientific Computing,

In the case **of process parallelism**, a given operation has multiple (but distinct) activities that can be processed on multiple processors.

Eg: Web Server and Compiler

In the case of the **farmer- and-worker model**, a job distribution approach is used: one processor is configured as master and all other remaining PEs are designated as slaves; the master assigns jobs to slave PEs and, on completion, they inform the master, which in turn collects results.

Eg: Search Engine Indexing

6(b) Analyze the levels of Parallelism in Computing 5 M

The level of parallelism depends on how big or small the parts of the program are (called *grain size*) that can be run at the same time. Table 2.1 shows different types of grain sizes used for parallelism. The main goal of all these methods is to make the processor work better by reducing waiting time (latency).

To do this, another task should be ready to run when one task takes a long time. For example, you can run two or more single-threaded programs at the same time, like compiling code, formatting text, searching a database, or simulating a device.

- Large grain (or task level)
- Medium grain (or control level)
- Fine grain (data level)
- Very fine grain (multiple-instruction issue)

Table 2.1 Levels of Parallelism

Grain Size	Code Item	Parallelized By
Large	Separate and heavyweight process	Programmer
Medium	Function or procedure	Programmer
Fine	Loop or instruction block	Parallelizing compiler
Very fine	Instruction	Processor

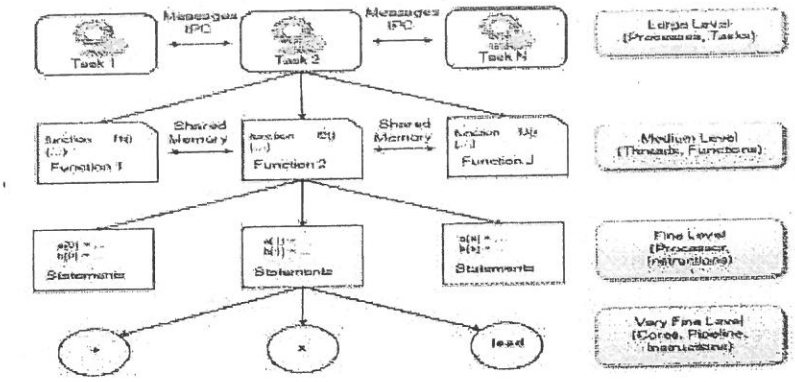


FIGURE 2.7
Levels of parallelism in an application.

(OR)

7(a) Apply the Message based communication models in Distributed Computing – 5M

The concept of a **message** has played a key role in the development of models and technologies for distributed computing. As defined by **Coulouris et al.**, a distributed system is *"one in which components located at networked computers communicate and coordinate their actions only by passing messages."*

In this context, a **message** means any piece of information sent from one part of the system to another. It could be a request to run a remote function, a shared data object, or just a small data packet. These messages are **limited in size and time**, and they do **not rely on continuous data streaming**.

➢ **Models for message-based communication:**

We've learned that **message-based communication** is a key part of how components in distributed systems interact. But how messages are exchanged—and between how many components—can vary. A common way to organize this is the **client/server model**, where one component (the server) handles requests from others (clients). This is a **point-to-point** setup where one client talks directly to one server. However, there are other communication models too. Let's look at the three most common ones:

1. Point-to-Point Message Model

This model involves communication between two specific components—like one computer sending a message directly to another.

There are two main types:

- **Direct Communication**: The message is sent straight to the receiver and processed immediately.
- **Queue-Based Communication**: The message goes into a queue (a line) on the receiver's side and is processed later.

Use case: Best for one-to-one or many-to-one communication, like a client requesting data from a specific server.

2. Publish-and-Subscribe Message Model

This model is about **notifications**. Here, there are two main roles:

- **Publisher**: Sends out messages when something important happens.
- **Subscriber**: Signs up to receive messages about certain topics or events.

There are two ways messages are shared:

- **Push**: The publisher actively sends messages to all subscribers.
- **Pull**: Subscribers regularly check (poll) for new messages from the publisher.

Use case: Good for one-to-many communication, like news updates, stock prices, or notifications. The publisher doesn't need to know who the subscribers are.

3. Request-Reply Message Model

In this model, every message (request) sent by one component expects a **reply**. It's like asking a question and waiting for an answer.

- Often used in **point-to-point** communication, like a client-server system.
- Less common in **publish-subscribe** models, which are based on broadcasting notifications.

Use case: Ideal when a response is required, like making a query to a database and getting the result back.

7(b) Describe how to apply Service oriented architecture in cloud-based web services- 5M

SOA is a way of designing software by organizing it as a group of **interacting services**. These services are **independent**, **reusable**, and can work across different systems. SOA helps create a **decentralized system** where components are loosely connected.

Service-Oriented Architecture (SOA) in cloud-based web services is applied by designing applications as reusable, loosely coupled services that communicate over standard protocols, enabling interoperability, scalability, and flexibility across platforms.

Two Main Roles in SOA

1. Service Provider

- Creates and manages services.
- Publishes service details in a **registry** (like a catalog).
- Also provides a **service contract** that explains:
  - What the service does
  - How to use it
  - What it requires
  - Any fees (if applicable)

2. Service Consumer

- Finds service info (metadata) from the registry.
- Builds a **client** to connect and use the service.
- Can belong to a different organization or business area than the provider.

Key Steps to Apply SOA in Cloud-Based Web Services

1. Service Identification & Design

- Break down business processes into independent services (e.g., authentication, payment, reporting).
- Each service should represent a self-contained business function with well-defined inputs/outputs.

2. Standardized Interfaces

- Use web service standards like SOAP, REST, or gRPC for communication.
- Ensure services are accessible via platform-neutral protocols (HTTP, XML, JSON).
- This guarantees interoperability across different systems and programming languages.

3. Service Registry & Discovery

- Publish services in a registry (like UDDI or API Gateway).

- Consumers can discover and bind to services dynamically, supporting cross-platform integration.

4. **Loose Coupling & Reusability**
   - Services should be independent of each other, so changes in one don't break others.
   - Example: A single authentication service reused across multiple applications instead of duplicating code.
5. **Cloud Deployment & Scalability**
   - Deploy services on cloud platforms (AWS, Azure, GCP) to leverage elastic scaling.
   - Use orchestration tools (Kubernetes, Docker Swarm) to manage service scaling and availability.
6. **Fault Tolerance & Reliability**
   - Implement redundancy and failover mechanisms at the service level.
   - Use cloud-native monitoring (AWS CloudWatch, Azure Monitor) to detect failures and reroute requests.
7. **Security & Governance**
   - Apply identity management, encryption, and access control across services.
   - Define Service Level Agreements (SLAs) to ensure performance and compliance.

UNIT-IV

8(a) Apply the components of an energy-efficient in green cloud architecture ~~to optimize resources, reduce power use, and ensure sustainable performance~~ - 5 M

A High-Level architecture for supporting energy-efficient resource allocation in Green Cloud computing infrastructure in fig 11.2. It consists of four main components

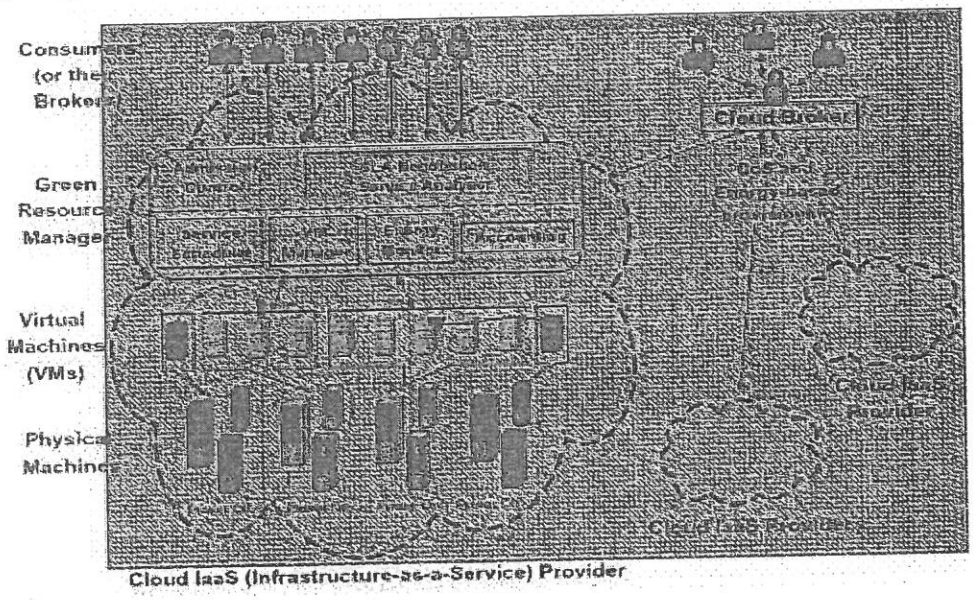


FIGURE 11.2
High-level system architectural framework for green cloud computing.

- **Consumers/brokers**. Cloud consumers or their brokers submit service requests from anywhere in the world to the cloud. It is important to note that there can be a difference between cloud consumers and users of deployed services. For instance, a consumer can be a company deploying a Web application, which presents varying workloads according to the number of "users" accessing it.

- **Green Resource Allocator.** Acts as the interface between the cloud infrastructure and consumers. It requires the interaction of the following components to support energy-efficient resource management:
  - **Green Negotiator.** Negotiates with the consumers/brokers to finalize the SLAs with specified

prices and penalties (for violations of SLAs) between the cloud provider and the consumer, depending on the consumer's QoS requirements and energy-saving schemes. In Web applications, for instance, the QoS metric can be 95% of requests being served in less than 3 seconds.

- **Service Analyzer.** Interprets and analyzes the service requirements of a submitted request before deciding whether to accept or reject it. Hence, it needs the latest load and energy information from VM Manager and Energy Monitor, respectively.
- **Consumer Profiler.** Gathers specific characteristics of consumers so that important consumers can be granted special privileges and prioritized over other consumers.
- **Pricing.** Decides how service requests are charged to manage the supply and demand of computing resources and facilitate prioritizing service allocations effectively.
- **Energy Monitor.** Observes and determines which physical machines to power on or off.
- **Service Scheduler.** Assigns requests to VMs and determines resource entitlements for allocated VMs. It also decides when VMs are to be added or removed to meet demand.
- **VM Manager.** Keeps track of the availability of VMs and their resource entitlements. It is also in charge of migrating VMs across physical machines.
- **Accounting.** Maintains the actual usage of resources by requests to compute usage costs. Historical usage information can also be used to improve service allocation decisions.

- **VMs.** Multiple VMs can be dynamically started and stopped on a single physical machine to meet accepted requests, hence providing maximum flexibility to configure various partitions of resources on the same physical machine to different specific requirements of service requests. Multiple VMs can also run concurrently applications based on different operating system environments on a single physical machine. In addition, by dynamically migrating VMs across physical machines, workloads can be consolidated and unused resources can be put on a low-power state, turned off, or configured to operate at low performance levels (e.g., using Dynamic Voltage and Frequency Scaling, or DVFS) to save energy.

Physical machines. The underlying physical computing servers provide hardware infrastructure for creating virtualized resources to meet service demands

8(b) Analyze the importance of Cloud federation stack in federated clouds – 5M

Creating a cloud federation involves research and development at different levels: conceptual, logical and operational, and infrastructural. Figure 11.7 provides a comprehensive view of the challenges faced in designing and implementing an organizational structure that coordinates together cloud services that belong to different administrative domains and makes them operate within a context of a single unified service middleware.

Each cloud federation level presents different challenges and operates at a different layer of the IT stack. It then requires the use of different approaches and technologies. Taken together, the solutions to the challenges faced at each of these levels constitute a reference model for a cloud federation

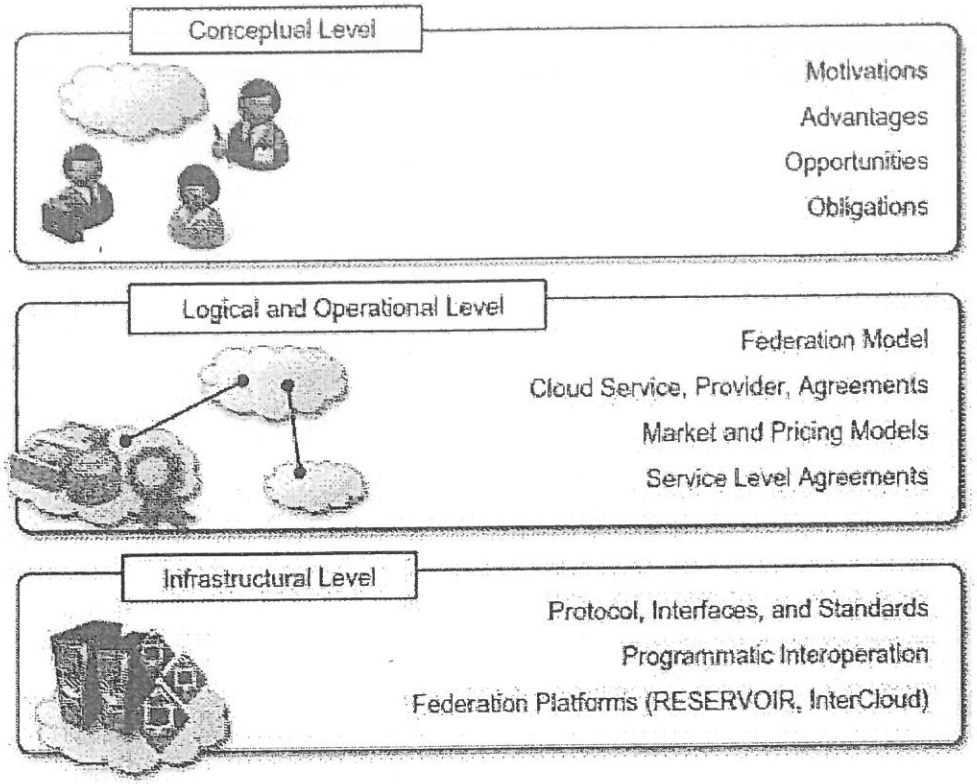


FIGURE 11.7

Cloud federation reference stack.

(OR)

9(a) Apply the CIA triad and its relevance in cloud security - 5 M

The **CIA Triad** (Confidentiality, Integrity, and Availability) is the foundational model for security policies. In the context of cloud security, it is used to define, implement, and evaluate the protection mechanisms for data and services hosted in cloud environments.

1. Confidentiality (C)

Confidentiality ensures that data is kept **private** and is only accessed by authorized individuals or systems.

- **Relevance in Cloud Security:** Protecting sensitive customer data and intellectual property from unauthorized disclosure, especially given the shared responsibility model.

- **Implementation Steps:**

- **Encryption:** Using strong **AES-256 encryption** for data both **in transit** (using TLS/SSL for communication) and **at rest** (encrypting cloud storage buckets, databases, and virtual machine disks).
- **Access Control:** Implementing the **principle of least privilege** using **Role-Based Access Control (RBAC)** or Attribute-Based Access Control (ABAC) to restrict resource access only to those who absolutely need it.
- **Key Management:** Securely managing encryption keys using cloud-native services like **Key Management Service (KMS)** or dedicated hardware security modules (HSMs).
- **Data Masking:** Hiding or obfuscating sensitive data fields in development or testing environments.

2. **Integrity (I) (1.5 Marks)**

Integrity ensures that data is **accurate, complete, and trustworthy**, and that it has not been modified in an unauthorized manner.

- **Relevance in Cloud Security:** Guaranteeing that configuration files, logs, and stored data are reliable and have not been tampered with by malicious actors or faulty processes.
  - **Implementation Steps:**
    - **Hashing and Digital Signatures:** Using **cryptographic hashing** (e.g., SHA-256) to verify data integrity after transmission or retrieval. **Digital signatures** verify both the integrity and the source's authenticity.
    - **Version Control and Immutable Infrastructure:** Storing configurations and code in version control systems and using **Immutable Infrastructure** principles (never modify a running server; replace it).
    - **Logging and Monitoring:** Implementing comprehensive **audit logs** that record every event and change to critical systems. The integrity of the logs themselves must be protected.

3. **Availability (A)**

Availability ensures that authorized users can **reliably access** the systems, data, and applications when required.

- **Relevance in Cloud Security:** Ensuring service continuity and uptime, which is a core expectation of cloud computing. Failures directly impact business operations and customer trust.
  - **Implementation Steps:**
    - **Redundancy and Failover:** Deploying services across **multiple Availability Zones (AZs)** or geographic regions to prevent outages from single points of failure.
    - **Disaster Recovery (DR):** Implementing a robust **Disaster Recovery plan** with regular backups and defined Recovery Time Objectives (RTO) and Recovery Point Objectives (RPO).
    - **DDoS Protection:** Using cloud-native Distributed Denial of Service (DDoS) protection services to filter malicious traffic and ensure legitimate users can reach the application.
    - **Scalability:** Utilizing **auto-scaling** features to dynamically adjust capacity and maintain performance during peak load periods.

9(b) Explain the Cloud Security Architecture **5 M**

Cloud security architecture is the strategic blueprint for an organization's cloud environment, encompassing all the technologies, policies, and controls designed to protect data, applications, and infrastructure. It's not a single product but a comprehensive framework that guides how an organization will secure its resources in the cloud.

A well-designed cloud security architecture is built on core principles and incorporates a multi-layered approach to address the unique threats and complexities of cloud environments.

Core Principles of Cloud Security Architecture

1. **Shared Responsibility Model**: The foundation of any cloud security architecture is a clear understanding of what the customer is responsible for versus what the cloud service provider (CSP) is responsible for. This model defines the boundaries for security controls.
2. **Defense-in-Depth**: This principle advocates for a layered security approach. If one security

control fails, another one is in place to provide a backup defense. This includes layers like network security, identity and access management, data encryption, and application security.

3. **Principle of Least Privilege**: Users and systems should only be granted the minimum permissions necessary to perform their required tasks. This significantly reduces the attack surface and minimizes the potential damage from a compromised account.
4. **Zero Trust**: A zero-trust model assumes that no user or system, inside or outside the network, should be trusted by default. Every access request is verified based on identity, context, and other security signals.
5. **Confidentiality, Integrity, and Availability (CIA Triad)**: This is a classic security model that is highly relevant in the cloud.

- Confidentiality: Protecting data from unauthorized access.
- Integrity: Ensuring that data is accurate, complete, and has not been tampered with.
- Availability: Guaranteeing that authorized users can access data and services when needed.

Key Components of a Cloud Security Architecture

A robust cloud security architecture is composed of several critical components that work together to provide comprehensive protection.

- Identity and Access Management (IAM)
- Data Security
- Network Security
- Security Monitoring and Event Management
- Application Security
- Cloud Security Posture Management (CSPM)
- Incident Response and Disaster Recovery

UNIT V

10(a) Analyze the compute and storage services provided by AWS 5 M

AWS offers a wide range of compute services to support different workloads, from simple web hosting to high-performance computing. Key services include:

AWS provides a wide range of compute and storage services designed to deliver scalable, reliable, and cost-efficient cloud solutions.

1. Compute Services Analysis

1. **Amazon EC2 (Elastic Compute Cloud):**
   Offers resizable virtual machines with different instance types (general-purpose, compute-optimized, GPU, memory-optimized). It supports auto-scaling and elastic load balancing, enabling dynamic resource allocation based on workload demand.

2. **AWS Lambda:**
   A serverless compute service that runs code without provisioning servers. It provides automatic scaling, event-driven execution, and pay-per-use pricing, making it efficient for microservices and lightweight applications.

3. **Amazon ECS & EKS:**
   ECS manages Docker containers, while EKS supports Kubernetes workloads. They simplify container orchestration, improve application portability, and enhance deployment automation.

4. **AWS Elastic Beanstalk:**
   A fully managed platform for deploying applications where AWS handles provisioning, scaling, and monitoring, reducing operational overhead.

Analysis:
AWS compute services focus on scalability, elasticity, and operational efficiency, supporting both VM-based and serverless architectures.

2. Storage Services Analysis

1. **Amazon S3 (Simple Storage Service):**
   A highly durable object storage service offering 11 nines durability. Supports versioning, lifecycle policies, and multiple storage classes (Standard, IA, Glacier), making it suitable for backups, big data, and static web hosting.

2. **Amazon EBS (Elastic Block Store):**
   Provides block-level persistent storage for EC2 instances. It supports provisioned IOPS, snapshots, and high availability, ideal for databases and transactional workloads.

3. **Amazon EFS (Elastic File System):**
   A scalable NFS file system for Linux-based workloads. Automatically grows and shrinks based on the amount of stored data, supporting shared access across multiple EC2 instances.

4. **Amazon Glacier / S3 Glacier:**
   Low-cost archival storage with retrieval options suitable for long-term backups and compliance data.

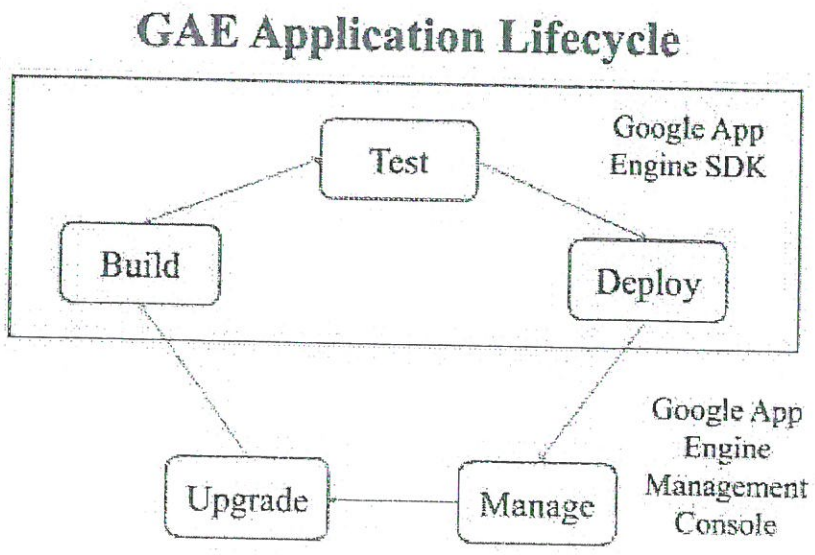
Analysis:
AWS storage services offer flexibility across object, block, and file storage, ensuring durability, high throughput, automated management, and cost optimization for different data types.

10(b) Analyze the application lifecycle in Google APP Engine 5 M

Google App Engine is a Platform-as-a-Service (PaaS) that allows developers to build and deploy applications without managing the underlying infrastructure. It supports automatic scaling, multiple programming languages (Python, Java, Node.js, Go, PHP), and integrates with other Google Cloud services.

The application lifecycle in GAE consists of several stages, from development to deployment, scaling, and maintenance.

Application Lifecycle in Google App Engine (GAE)



Test: Developers rigorously test their applications in the local development environment to ensure functionality and identify potential issues.

Deploy: After successful testing, the application is deployed to Google App Engine using the 'gcloud app deploy' command, and a specific version is assigned for version control.

Manage: In this phase, developers use monitoring tools like Stackdriver to oversee the application's performance, resource usage, and user interactions.

Upgrade: Enhancements or bug fixes are made to the application, and new versions are deployed. Features like traffic splitting may be utilized for seamless updates.

Build: An iterative phase involving continuous improvement based on user feedback, evolving requirements, and emerging technologies.

(OR)

11(a) Analyze the communication and additional services of AWS - 5M

Communication Services Analysis

AWS provides several communication and messaging services that enable reliable, scalable interaction between distributed applications.

a) Amazon SNS (Simple Notification Service)

A fully managed publish–subscribe messaging service. It delivers messages to multiple subscribers (mobile devices, email, HTTP endpoints).

Analysis: Supports high-throughput notifications and decouples microservices for better scalability.

b) Amazon SQS (Simple Queue Service)

A distributed message queuing service that stores messages until they are processed.

Analysis: Ensures reliable message delivery, fault-tolerance, and helps build loosely coupled and resilient applications.

c) Amazon SES (Simple Email Service)

A cloud-based email sending service for applications and businesses.

Analysis: Provides cost-effective bulk email delivery with high deliverability and easy integration with other AWS services.

d) Amazon API Gateway

Manages, publishes, and secures APIs for backend services.

Analysis: Enables communication between client applications and cloud services with monitoring, throttling, and authentication.

2. Additional AWS Services Analysis

These services extend AWS functionality beyond communication, supporting monitoring, security, deployment, and management.

a) AWS CloudWatch

Monitors resource usage and application performance.

Analysis: Improves operational visibility through metrics, logs, alarms, and automated actions.

b) AWS CloudTrail

Tracks user activity and API calls for governance and auditing.

Analysis: Enhances security, compliance, and troubleshooting through event history.

c) AWS IAM (Identity and Access Management)

Controls user access and permissions.

Analysis: Ensures secure operations by applying least-privilege and role-based access control.

d) AWS CloudFormation
Automates infrastructure deployment using templates.
Analysis: Supports Infrastructure as Code (IaC), improves consistency, reduces manual errors, and accelerates provisioning.
e) AWS Route 53
A highly available DNS and routing service.
Analysis: Enhances global application performance using domain management, routing policies, and health checks.

11(b) Describe about Microsoft Azure Core Concepts 5M

Microsoft Azure is a cloud computing platform and suite of services that allows individuals and businesses to create, deploy, and manage applications and services. It provides a wide range of cloud-based solutions that use Microsoft-operated data centres.

Azure Core Concepts

1. Compute Services:

Azure Virtual Machines (VMs) allow users to run virtualized Windows or Linux servers in the cloud, providing flexibility and scalability. Azure App Service supports the deployment of web and mobile applications without managing infrastructure, while Azure Functions enables serverless computing for event-driven scenarios.

2. **Storage Services:**

Azure Storage provides a suite of scalable and secure storage solutions. Blob Storage is ideal for large amounts of unstructured data, Table Storage is a NoSQL data store for semi-structured data, Queue Storage facilitates communication between components, and Azure Files offers fully managed file shares.

3. **Networking:**

Azure Virtual Network enables the creation of isolated and customizable networks. Azure Load Balancer distributes incoming network traffic across multiple servers to enhance availability and responsiveness.

4. **Databases:**

Azure offers managed database services like Azure SQL Database for relational databases, Cosmos DB for globally distributed NoSQL databases, and Azure Database for MySQL and PostgreSQL to cater to various database needs.

5. Identity and Access Management:

Azure Active Directory (AD) is a comprehensive identity and access management service. It allows users to securely sign in and access Azure resources, providing a single sign-on experience.

6. Security and Compliance:

Azure Security Center helps safeguard applications from threats, Azure Policy allows the enforcement of organizational standards, and Azure Blueprints facilitates the deployment of compliant environments to meet regulatory requirements.

7. Monitoring and Management:

Azure Monitor provides a unified view of application performance and health. Azure Resource Manager (ARM) simplifies resource management and deployment through templates, allowing consistent and repeatable deployments.

8. Serverless Computing:

Azure Functions enables developers to write and deploy code without worrying about the underlying infrastructure. This serverless model allows for efficient handling of event driven workloads, ensuring optimal resource usage and cost-effectiveness.

9. **Artificial Intelligence (AI) and Machine Learning:**

Azure AI and Machine Learning services empower developers and data scientists to build, train, and deploy machine learning models. Azure Cognitive Services offer prebuilt AI capabilities, such as vision, speech, and language understanding.

10. **DevOps:**

Azure DevOps provides a set of tools for collaborative software development and delivery. Azure Repos supports version control, Azure Pipelines enables continuous integration and continuous deployment (CI/CD), and Azure Boards facilitates agile project management.