K. K. Wagh Institute of Engineering Education and Research, Nashik
(An Autonomous Institute from A. Y. 2022-23)

MODEL ANSWER KEY

Exam Seat No.:
End-Sem Examination-I — Winter 2025
Academic Year: 2025-26 — Sem: I
Class: F.Y — Program: MCA
Branch Code:MCA — Pattern: 2024
Name of Course: Elective I :Cloud Computing — Course Code: 2409505A
Max. Marks: 60 — Duration: 2:30 Hrs.

Instructions: Candidates should read carefully the instructions printed on the Question Paper and on the cover page of the Answer Book, which is provided for their use.
(*keep space*)
1. This question paper contains 2 page(s).
2. Answer to each new question is to be started on a new page.
3. Assume suitable data wherever required, but justify it.
4. Draw the neat labelled diagrams, wherever necessary.
5. The last columns indicate the Course Outcome.

Q. No.	Details	Max. Marks	CO No.	BT Level
Q.1	Summarize the key characteristics and benefits of cloud computing highlighting how they support scalable and flexible computing services. (6 Marks) **Answer:** **Key Characteristics of Cloud Computing:** Cloud computing provides shared computing resources over the internet on demand. Its important characteristics are: • **On-demand self-service:** Users can provision computing resources (VMs, storage, etc.) whenever required, without human intervention from the provider. • **Broad network access:** Services are accessible over the network using standard mechanisms via laptops, mobiles, tablets, etc. • **Resource pooling:** Multiple customers share a common pool of resources like servers and storage; resources are dynamically assigned and reassigned according to demand. • **Rapid elasticity:** Resources can be quickly scaled up or down. To the user, the capacity often appears unlimited. • **Measured service:** Resource usage is monitored, controlled, and billed on a pay-per-use basis. **Benefits for Scalability and Flexibility:** These characteristics support:	[6]	CO1	L2

	• **Scalability:** Rapid elasticity and resource pooling allow applications to handle varying loads without downtime or manual hardware upgrades. • **Cost efficiency:** Measured service and no upfront infrastructure cost reduce financial risk, which is especially beneficial for startups and growing businesses. • **Operational flexibility:** Organizations can quickly experiment, deploy, and update services, adapting to changing business needs and user demands. **Conclusion:** Thus, the combination of on-demand availability, elastic scaling, and pay-as-you-go pricing makes cloud computing a highly scalable and flexible computing model.			
Q.2	Compare the service models IaaS, PaaS, and SaaS with suitable examples from platforms like Amazon EC2. (6 Marks) **Answer:** **Infrastructure as a Service (IaaS):** IaaS provides virtualized computing resources over the internet such as virtual machines, storage, and networks. Users manage OS, runtime, and applications. • **Example: Amazon EC2** offers virtual servers where you choose OS, install software, and manage configurations. **Platform as a Service (PaaS):** PaaS offers a ready-made platform with OS, runtime, and development tools. Developers only focus on writing and deploying applications, not managing underlying infrastructure. • **Examples: AWS Elastic Beanstalk**, **Google App Engine**, **Azure App Service** – these platforms manage scaling, patching, and servers automatically. **Software as a Service (SaaS):** SaaS delivers complete applications over the internet. Users access software through a browser without worrying about installation or maintenance. • **Examples: Gmail**, **Salesforce CRM**, **Microsoft 365**.	[6]	CO2	L2

	Comparison Summary: • IaaS: Maximum control, more management responsibility. • PaaS: Balanced control; easy development and deployment. • SaaS: Minimum control over infrastructure; ready-to-use applications. This layered service model allows organizations to choose the right level of abstraction depending on their needs.			
Q.3	a) Demonstrate the use of Amazon Simple Database Service (SimpleDB) for managing structured data in an e-commerce application scenario. (8 Marks) **Answer:** **Role of SimpleDB in E-commerce:** Amazon SimpleDB is a NoSQL data store for storing structured, queryable data. In an e-commerce application, it can be used to manage product catalogs, user carts, and order details. **Storing Product Information:** Each product can be stored as an item with attributes such as ProductID, Name, Category, Price, Stock, and Brand. This makes it easy to query products by any combination of attributes, like category or price range. **Managing User Cart and Orders:** User cart items can be saved with attributes such as UserID, ProductID, Quantity, and Timestamp. When a user places an order, order items can also be stored and retrieved quickly for order history and tracking. **Query and Indexing Support:** SimpleDB automatically indexes data on all attributes, enabling flexible queries such as: • "Find all products in 'Electronics' with price less than ₹10,000" • "List all orders for a particular user in the last month" **Scalability and Availability:** SimpleDB automatically handles replication and load distribution, supporting growing product lists and user traffic without manual scaling.	[16]	CO3	L3

Integration with Other AWS Services:
It can work with **EC2** (application servers) and **S3** (storing images and documents) to form a complete e-commerce solution.

Conclusion:
By using SimpleDB for structured, attribute-based querying in an e-commerce scenario, the application becomes scalable, flexible, and easy to manage without complex database administration.

OR

b) Show how SQL Azure can be used to support database-as-a-service for a customer relationship management (CRM) system. (8 Marks)

Answer:

SQL Azure as Database-as-a-Service:
SQL Azure (Azure SQL Database) is a fully managed relational database service. It is ideal for CRM systems that require consistent, secure, and scalable management of customer data.

Storing CRM Data:
The CRM system can use SQL Azure to store customer profiles, contact details, interactions, leads, and sales records. Tables such as Customers, Contacts, Opportunities, and Activities can be designed using relational schemas with primary and foreign keys.

High Availability and Backup:
SQL Azure provides built-in backup, point-in-time restore, and automatic high availability across data centers. This ensures customer data is not lost and CRM remains accessible.

Scalability for Growing Users:
As the number of sales agents and customers increases, SQL Azure can scale up or out with minimal downtime, supporting more transactions and larger datasets.

Security and Compliance:
Features like encryption at rest and in transit, firewall rules, and Azure Active Directory integration ensure sensitive customer data is protected. This is crucial for CRM systems dealing with confidential business information.

Integration with Azure Services:
CRM applications hosted on Azure App Service or VMs can directly connect to SQL Azure using secure connection

	strings, simplifying deployment and maintenance. **Conclusion:** SQL Azure supports CRM as a database-as-a-service by offering managed relational storage, high availability, strong security, and scalable performance with minimal administrative overhead.			
	c) Implement the Windows Azure Platform Appliance concept in a hypothetical scenario where a company needs a private cloud setup. (8marks) **Answer:** **Private Cloud Requirement:** Consider a large enterprise (e.g., a bank or government organization) that needs cloud benefits (scalability, self-service) but must keep data on-premises due to security and compliance. **Windows Azure Platform Appliance Concept:** The Windows Azure Platform Appliance was designed as a packaged set of hardware and software that could be installed in the organization's own data center to provide Azure-like cloud services privately. **Setting Up Private Cloud:** The appliance would include: • Pre-configured servers, storage, and networking. • Windows Azure platform software for running applications. • Management tools for provisioning and monitoring resources. **Usage in the Hypothetical Company:** The company's internal IT team can: • Host line-of-business applications on the private cloud. • Provide self-service VM and application deployment to departments. • Enforce internal security policies while still benefiting from cloud elasticity. **Benefits to the Organization:** • **Data Sovereignty:** Data stays within the company's data center.		CO3	L3

- **Compliance:** Easier to meet regulatory requirements.
- **Cloud Experience:** Users get cloud-like features such as self-service and rapid provisioning within a controlled environment.

Conclusion:
The Windows Azure Platform Appliance model enables organizations to implement a private cloud with Azure technologies on-premises, combining control with cloud advantages.

OR

d) Demonstrate the Google App Engine application lifecycle by creating a workflow for developing, deploying, and updating a cloud-based application. (8marks)

Answer:

Development Phase:
Developers create the application using supported languages such as Python, Java, Go, or Node.js. They use Google App Engine SDK and follow the platform's APIs for datastore, authentication, and other services.

Testing Locally:
The application is tested on the local development server (provided by the SDK). This simulates the App Engine environment, allowing developers to debug and verify functionality.

Deployment to App Engine:
Once stable, the application is deployed to Google App Engine using deployment commands. The platform automatically provisions required resources such as instances and load balancing.

Automatic Scaling and Management:
After deployment, App Engine automatically scales the number of instances based on incoming traffic. Developers do not manage servers; the platform handles patching and resource allocation.

Monitoring and Logging:
Google Cloud Console provides logs, performance metrics, and error reports. Developers can monitor response times, errors, and request statistics.

Updating the Application:
New versions of the application can be deployed without downtime. App Engine offers versioning, enabling traffic

	splitting between old and new versions until testing is complete. **Conclusion:** The Google App Engine application lifecycle involves developing locally, deploying to the cloud, automatically scaling, monitoring, and updating seamlessly, enabling rapid and reliable delivery of cloud-based applications.			
Q.4	a) Use Serverless Computing to implement a lightweight event-driven function for a real-time application scenario. (8 marks) **Answer:** **Real-Time Scenario:** Consider a real-time notification system that sends alerts to users when a new order is placed in an e-commerce application. **Serverless Function Concept:** Using services like **AWS Lambda**, **Azure Functions**, or **Google Cloud Functions**, the logic to process events is written as a small function that runs only when triggered. **Event Trigger:** When a new order is stored in the database or posted to a message queue (like SNS, SQS, or Azure Queue), an event is generated that triggers the serverless function. **Processing Within the Function:** The function reads order details, prepares the notification content, and sends an SMS, email, or push notification through integrated services (like SES, Twilio, or Firebase Cloud Messaging). **Scalability and Cost Efficiency:** The function automatically scales based on the number of incoming events. There are no idle server costs because billing is based only on actual execution time and number of invocations. **No Server Management:** Developers focus only on writing the business logic. Cloud provider handles provisioning, updates, and fault tolerance of the underlying compute infrastructure. **Conclusion:** Serverless computing enables lightweight, event-driven applications that are highly scalable, cost-effective, and easy	[16]	CO4	L3

	to maintain, making them ideal for real-time notification and similar use cases. **OR** b) Implement a build and release process for a cloud-based mobile or web application using DevOps tools. (8 marks) **Answer:** **Source Code Management:** Developers commit code to a central repository such as GitHub or GitLab. Branching strategies like feature branches or GitFlow are used to manage changes. **Continuous Integration (CI):** A CI tool like Jenkins, GitHub Actions, or Azure DevOps is configured to automatically build the application whenever code is pushed. It compiles code, runs unit tests, and checks code quality. **Automated Build Creation:** For web applications, build steps may include bundling and minifying front-end assets. For mobile apps, APK/IPA packages are generated. **Continuous Delivery (CD):** After successful builds and tests, the pipeline automatically deploys the application to a staging environment in the cloud (e.g., AWS Elastic Beanstalk, Azure App Service, or Kubernetes cluster) for further testing. **Release to Production:** Once verified, the same pipeline promotes the build from staging to production using blue-green or rolling deployment strategies to reduce downtime and risk. **Monitoring and Feedback:** Application performance is monitored using tools like CloudWatch, Azure Monitor, or Application Insights. Logs and metrics provide feedback to continuously improve the build and release process. **Conclusion:** Using DevOps tools for automated build and release improves reliability, reduces deployment time, and enables frequent, safe releases for cloud-based applications.			
	c) Analyze the case of Spotify using Docker and apply similar containerization strategies to enhance performance and		CO4	L3

	deployment efficiency in another media-streaming platform. (8marks) **Answer:** **Spotify and Docker:** Spotify uses Docker to containerize its microservices, allowing each service (such as user management, playlists, recommendations) to run independently. This improves deployment speed and scalability. **Applying Strategy to Another Media-Streaming Platform:** A similar platform can break its application into microservices: user authentication, media catalog, streaming service, search, and recommendation engine. Each service is packaged into a Docker container. **Benefits of Containerization:** • **Consistency:** Containers ensure that services behave the same across development, testing, and production. • **Scalability:** Popular services like streaming or search can be scaled independently by running more containers. • **Faster Deployment:** New versions can be rolled out quickly by replacing old containers with updated images. • **Resource Efficiency:** Containers share the host OS, allowing more services to run on the same hardware compared to full VMs. **Orchestration and Management:** Using Kubernetes or similar orchestrators, the platform can manage container deployment, scaling, and health checks automatically, just like Spotify does. **Conclusion:** By adopting containerization and orchestration similar to Spotify's approach, another media-streaming platform can achieve better performance, scalability, and deployment efficiency. **OR** d) Apply cloud-based ECG analysis techniques to design a solution that supports remote patient monitoring in the healthcare domain. (8 marks) **Answer:**			

	Remote Monitoring Requirement: Patients at home or in rural areas need continuous ECG monitoring without being physically present in hospitals. **Data Collection:** Wearable ECG devices or smart sensors attached to patients capture ECG signals and send data via mobile or IoT gateways to the cloud. **Cloud Ingestion and Storage:** Data is transmitted securely to cloud services (like AWS IoT, Azure IoT Hub). ECG signals are stored in cloud databases or data lakes for analysis. **Real-Time Analysis:** Cloud-based analytics services or machine learning models analyze ECG patterns to detect abnormalities such as arrhythmia or irregular heartbeats in real-time. **Alerting and Visualization:** When abnormal patterns are detected, alerts are sent to doctors and caregivers via SMS, email, or mobile apps. Dashboards provide visualization of ECG trends over time. **Scalability and Accessibility:** Cloud platforms can handle large numbers of patients simultaneously, and authorized doctors can access data from anywhere, improving healthcare reach. **Conclusion:** Cloud-based ECG analysis enables scalable, real-time, remote patient monitoring, improving early diagnosis and continuous care in the healthcare domain.			
Q.5	a) Apply security controls to prevent malicious intermediary and insufficient authorization attacks while accessing SaaS applications. (8 marks) **Answer:** **Malicious Intermediary Attacks:** These attacks occur when a third party intercepts or alters communication between the user and SaaS provider (Man-in-the-Middle). **Security Controls for Malicious Intermediary:** • **End-to-end encryption using HTTPS/TLS** to protect	[16]	CO5	L3

data in transit.

- **Certificate validation and pinning** to ensure the client communicates only with genuine servers.
- **Use of secure VPNs** when accessing SaaS applications over public networks.

Insufficient Authorization Attacks:
These occur when users gain access to functionalities or data beyond their permissions due to weak access control.

Security Controls for Authorization:

- **Role-Based Access Control (RBAC):** Assign roles such as admin, user, viewer, and strictly limit privileges.
- **Least Privilege Principle:** Users should have only the minimum permissions required.
- **Strong session management:** Proper handling of tokens, session expiry, and logout.
- **Regular audits and access reviews:** Periodic checks to ensure roles and permissions are appropriate.

Conclusion:
By combining secure communication, robust authentication, and strict authorization controls, SaaS applications can be protected from malicious intermediary and insufficient authorization attacks.

OR

b) Illustrate the impact of virtualization attacks and describe how isolation mechanisms can be applied to protect virtualized infrastructures. (8 marks)

Answer:

Impact of Virtualization Attacks:
Virtualized infrastructures host multiple virtual machines (VMs) on a single physical server. Attacks such as hypervisor compromise, VM escape, and side-channel attacks can:

- Allow attackers to break isolation between VMs.
- Expose sensitive data from other tenants.
- Disrupt services by controlling the hypervisor.

Isolation Mechanisms:

- **Strong Hypervisor Security:** Keeping the hypervisor updated and minimized reduces vulnerabilities.
- **VM Isolation:** Using hardware features like Intel VT-

	x, AMD-V, and IOMMU/VT-d to isolate memory and device access. • **Network Isolation:** Virtual LANs (VLANs) and security groups ensure that VMs in different tenant networks cannot communicate directly. • **Access Control and Monitoring:** Strict admin access controls, logging, and monitoring of hypervisor and management interfaces. **Conclusion:** Virtualization attacks can severely compromise multi-tenant environments, but robust isolation and security best practices can significantly reduce these risks.			
	c) Apply the economic principles of cloud computing to create a cost-optimized deployment plan for a startup business. (8marks) **Answer:** **Startup Context:** A startup needs to deploy a web application with limited budget, uncertain traffic, and requirement for quick scaling. **Pay-as-You-Go Model:** Instead of buying servers, the startup uses cloud services and pays only for actual usage (compute hours, storage, and bandwidth). This avoids high upfront capital expenses. **Elasticity and Auto-Scaling:** The startup configures auto-scaling for application servers. During low traffic, fewer instances run, reducing cost; during peak traffic, extra instances are started automatically. **Use of Managed Services:** Instead of managing databases and messaging systems manually, the startup uses managed services (like RDS, DynamoDB, or Azure SQL) to lower operational costs and reduce the need for in-house admin resources. **Right-Sizing and Reserved Instances:** The startup selects appropriate instance sizes and, once usage patterns stabilize, may purchase reserved or savings plans to reduce long-term costs. **Conclusion:** By adopting cloud economic principles—pay-per-use, elasticity, managed services, and right-sizing—the startup achieves a cost-effective, scalable, and flexible deployment		CO5	L3

	plan. **OR** d) Implement edge or fog computing concepts to improve latency and efficiency for a smart city IoT application. (8 marks) **Answer:** **Smart City IoT Scenario:** Sensors across the city (traffic lights, pollution detectors, smart meters) continuously generate data that must be processed quickly. **Edge/Fog Computing Concept:** Instead of sending all raw data to a distant cloud, processing is done closer to the data source (at the network edge or intermediate fog nodes). **Local Processing at Edge/Fog Nodes:** Gateways or fog nodes perform filtering, aggregation, and preliminary analytics. Only important or summarized data is sent to the cloud, reducing bandwidth usage. **Improved Latency:** Decisions such as controlling traffic lights, generating alerts, or adjusting street lighting can be taken locally in milliseconds, without waiting for cloud response. **Cloud Integration:** The central cloud platform still stores long-term data, performs heavy analytics, and runs dashboards for city administrators. **Conclusion:** By using edge and fog computing in smart city IoT, latency is reduced, bandwidth is optimized, and real-time decision-making becomes more efficient and reliable.			