Cloud computing paper solution

EC2-

Q.1 Set. (A) **Read the below case study and answer the questions Marks 5+2+4+4 =15**

Aayush hospitals is a name to reckon in modern medical care providing excellent medical care and services to patients in India & abroad. However, their backend IT systems (billing, payroll, insurance processing, claims administration) were many and mostly comprised of legacy applications which run on mainframe technologies.
The hospital IT procured contemporary applications like patient administration, human resources management, analytics & MIS reporting to manage patient inflows.
With growing business, the rise in IT OPEX increased because of the inflexibility of the archaic systems and the management team decided to do some cost reduction by engaging with you, an established Cloud consultant. Based on the above premises answer briefly:

a. As a cloud consultant, what should Aayush hospitals do in the short term to rein in costs?
b. The hospital's IT director was confused about hosting and cloud hosting. In the context of the above case study, create a few use cases which could explain the difference to the director.
c. What will be your suggestions for a long-term solution for Aayush hospitals?
d. Assuming any cloud provider of your choice briefly describe the systems, which are likely to be moved to the cloud and why. Include the cloud components and describe why they are used.

Answer-

a. Short-Term Measures to Rein in Costs

Audit and Optimize Current Systems: Conduct a thorough audit of the existing IT infrastructure to identify redundant, underutilized, or overly expensive resources. Optimize these resources to reduce immediate costs.

Virtualization: Implement server virtualization to consolidate workloads onto fewer physical servers, which can reduce hardware, power, and cooling costs.

Cloud Pilot Projects: Begin small pilot projects to migrate non-critical applications to the cloud. This helps understand the cost implications and operational benefits without significant upfront investment.

Hybrid Cloud Approach: Utilize a hybrid cloud approach where critical legacy systems remain on-premises, but scalable cloud solutions handle additional workloads. This reduces the need for expensive hardware upgrades.

Negotiate with Vendors: Re-negotiate contracts with current software and hardware vendors to secure better pricing or more flexible terms.

b. Hosting vs. Cloud Hosting Use Cases

Traditional Hosting:

Use Case 1: Aayush Hospitals has a website hosted on a traditional web hosting service. The resources (server, storage) are fixed, and any increase in traffic could result in performance issues unless more physical hardware is provisioned manually.

Use Case 2: The payroll system runs on a dedicated server housed within the hospital's data center. The IT team must maintain this hardware, handle backups, and ensure uptime.

Cloud Hosting:

Use Case 1: The patient administration system is migrated to a cloud platform (e.g., AWS or Azure). This allows for automatic scaling during peak patient inflows without the need for manual hardware adjustments.

Use Case 2: The analytics and MIS reporting tools are hosted in the cloud, benefiting from on-demand computing resources and extensive data storage options. This setup can easily scale to handle large data volumes during end-of-month reporting periods.

c. Long-Term Solution for Aayush Hospitals

Comprehensive Cloud Migration Strategy: Develop a long-term cloud migration strategy to gradually move legacy systems to the cloud. This includes:

Assessment: Evaluating which applications are suitable for cloud migration.

Phased Approach: Migrating applications in phases to minimize disruption and manage costs.

Training: Upskilling IT staff to manage and operate cloud-based systems.

Adopt a Cloud-Native Architecture: Over time, re-architect legacy applications to become cloud-native, leveraging microservices, containers (e.g., Docker, Kubernetes), and serverless computing to maximize efficiency and scalability.

Disaster Recovery and Backup Solutions: Implement cloud-based disaster recovery and backup solutions to ensure data protection and business continuity.

Implement Cloud Governance and Cost Management Tools: Utilize cloud governance tools to monitor and control cloud usage, ensuring that the hospital remains within budget and complies with regulatory requirements.

d. Cloud Migration Plan and Components

Cloud Provider: AWS (Amazon Web Services)

Systems Likely to be Moved to the Cloud:

Billing and Insurance Processing:

Why: High transaction volume and need for scalability and integration with third-party systems.

Components: Amazon RDS (Relational Database Service) for transactional databases, AWS Lambda for processing tasks, and Amazon S3 for secure data storage.

Payroll System:

Why: Requires regular updates, security, and integration with HR systems.

Components: Amazon EC2 (Elastic Compute Cloud) for hosting applications, Amazon RDS for databases, and AWS IAM (Identity and Access Management) for secure access control.

Patient Administration System:

Why: High availability and reliability needed for managing patient data.

Components: Amazon Aurora for highly available and scalable relational databases, Amazon ECS (Elastic Container Service) for deploying microservices, and AWS Elastic Load Balancing for distributing traffic.

Analytics and MIS Reporting:

Why: Requires significant computational power and storage, with the need for quick scaling.

Components: Amazon Redshift for data warehousing, AWS Glue for ETL (Extract, Transform, Load) processes, and Amazon QuickSight for business intelligence and reporting.

Why These Components Are Used:

Scalability: Cloud services like Amazon EC2, RDS, and Redshift offer automatic scaling, ensuring that resources are available on-demand without manual intervention.

Cost Efficiency: Pay-as-you-go pricing models reduce capital expenditure and align costs with usage.

Reliability and Security: AWS provides built-in redundancy, data encryption, and compliance with healthcare regulations like HIPAA.

Flexibility: Cloud services allow for rapid deployment and iteration of new features, helping Aayush Hospitals adapt to changing business needs and technologies.

Q.1 Set. (B) **Read the below case study and answer the questions Marks 5+2+4+4 =15**

Beacon International is a new startup company providing off-the-shelf automation solutions for common business processes. Their main business offerings are automated bots for HR/Payroll/Administration/Training processes which are hosted on a third-party hosting site.

With increasing business, the hosted platform could not scale up quickly to the demands. Based on the above case study answer the questions

a. Why is Beacon international not able to scale in spite of being hosted on a third-party hosting provider? What could be the reasons? Suggest a solution to this problem.
b. The director of the company argued that moving back to on-premises would mitigate the problem of scalability, do you endorse the views of the director?
c. The IT director was of the opinion that the long-term strategy is to go cloud naïve. Define what is cloud native. Do you agree with the IT director's view ? Provide reasons.
d. If you are asked to design a cloud solution for Beacon, what would you suggest? Which components are likely to be hosted and which ones will be on the cloud? Using a cloud provider of your choice specify the systems and justify their usage.

Answer-

a. Reasons for Scaling Issues and Solutions

Reasons Beacon International is Facing Scaling Issues:

Resource Limitations of the Hosting Provider: The third-party hosting provider may have limitations on the resources (CPU, memory, storage) that can be provisioned quickly.

Lack of Auto-Scaling: The hosting provider might not support automatic scaling of resources in response to demand spikes.

Overprovisioned or Underprovisioned Resources: The hosting setup might not be optimized, leading to either overprovisioning (wasting resources and costs) or underprovisioning (insufficient resources).

Network Bottlenecks: The network infrastructure of the hosting provider might not be robust enough to handle the increased traffic efficiently.

Suggested Solution:

Move to a Cloud Provider: Transitioning to a cloud provider such as AWS, Azure, or Google Cloud can offer better scalability and flexibility. Cloud providers offer auto-scaling features that automatically adjust resources based on demand.

Implement Auto-Scaling and Load Balancing: Use services like AWS Auto Scaling and Elastic Load Balancing to ensure that the system can handle variable loads efficiently.

Optimize Resource Allocation: Regularly monitor and optimize resource usage to ensure that the application is cost-efficient and can scale effectively.

b. On-Premises Scalability

Director's Argument: Moving back to on-premises to mitigate scalability issues.

Evaluation:

Scalability Concerns: On-premises infrastructure typically involves significant upfront investment and may not scale as flexibly or quickly as cloud infrastructure.

Resource Management: Managing and maintaining physical servers can be more challenging and costly, particularly for a startup.

Elasticity: Cloud environments provide the elasticity to scale resources up or down based on demand, which is harder to achieve on-premises.

Conclusion: Moving back to on-premises is generally not advisable for a startup looking to scale quickly and efficiently. The cloud offers better flexibility, cost management, and scalability.

c. Cloud Native Strategy

Definition of Cloud Native:

Cloud Native refers to designing, building, and running applications to fully exploit the advantages of the cloud computing delivery model. This includes:

Microservices Architecture: Breaking down applications into small, independent services.

Containerization: Using containers (e.g., Docker) to package services for consistent deployment across environments.

Dynamic Orchestration: Managing containers with tools like Kubernetes.

DevOps Practices: Implementing continuous integration and continuous deployment (CI/CD) pipelines.

Evaluation:

Agility and Speed: Cloud-native applications can be developed, deployed, and scaled more quickly, allowing for faster time-to-market.

Resilience and Scalability: Designed for high availability and resilience, with the ability to scale out as needed.

Cost Efficiency: Optimizes resource utilization, reducing costs by scaling on-demand.

Conclusion: Agreeing with the IT director, adopting a cloud-native strategy is beneficial for Beacon International as it aligns with their need for scalability, agility, and cost-efficiency.

d. Designing a Cloud Solution for Beacon International

Cloud Provider: AWS (Amazon Web Services)

Components Likely to be Hosted on the Cloud:

Compute:

Amazon EC2: For scalable virtual servers to host the automation bots.

AWS Lambda: For serverless computing to handle event-driven processes without managing servers.

Containers:

Amazon ECS or EKS: For running containerized applications, providing scalability and orchestration with Docker and Kubernetes.

Database:

Amazon RDS: For relational database services to manage structured data with high availability and scalability.

Amazon DynamoDB: For a NoSQL database to handle unstructured or semi-structured data with fast performance at scale.

Storage:

Amazon S3: For scalable object storage to handle large volumes of data with high durability.

Networking:

Amazon VPC: To create a secure and isolated network environment for the applications.

Elastic Load Balancing: To distribute incoming traffic across multiple EC2 instances, ensuring high availability.

Monitoring and Management:

Amazon CloudWatch: For monitoring the infrastructure and applications, providing metrics and logs to manage performance and health.

Justification:

Scalability: AWS services like EC2, Lambda, and S3 offer automatic scaling to meet demand fluctuations.

Cost Efficiency: Pay-as-you-go pricing models ensure that Beacon International only pays for what they use, optimizing costs.

High Availability: AWS provides built-in redundancy and fault tolerance, ensuring that applications remain available and resilient.

Security: AWS offers comprehensive security services and compliance certifications, ensuring data protection and regulatory compliance.

Ease of Management: Managed services reduce the overhead of infrastructure management, allowing Beacon to focus on their core business.

By leveraging these AWS components, Beacon International can achieve the scalability, flexibility, and efficiency needed to support their growing business and ensure high performance for their automation solutions.

Q.1 Set. (C) **Read the below case study and answer the questions Marks 5+2+4+4 =15**

ABC Retail is a mid-size fashion retailer. Their IT infrastructure is hosted in-house and managed completely by a team of engineers. To cater to the young generation ABC retail developed a mobile application which used the middleware services hosted in the cloud using AWS.

a. As a consultant working with AWS, you are tasked with presenting the benefits of shifting the infrastructure from in -house to AWS cloud. Describe how you would put your case forward and what components you suggest for ABC to derive maximum ROI.

b. The director of the company, is very paranoid about moving organization data to the public cloud? Explain how you will assuage the director that the public cloud is safe?

c. The IT director was of the opinion that the long-term strategy is to go cloud naïve. Define what is cloud native. Do you agree with the IT director's view ? Provide reasons.

d. Will Open stack be a better option for ABC retail assuming that the company is expected to grow 15% YoY. Give reasons for your choice.

Answers-

a. Benefits of Shifting Infrastructure to AWS Cloud

Benefits of Moving to AWS:

Scalability:

Elastic Compute Cloud (EC2): Allows for scalable computing capacity, enabling ABC Retail to handle traffic spikes during sales and promotions without investing in additional hardware.

Auto Scaling: Automatically adjusts the number of EC2 instances in response to demand, ensuring optimal performance and cost-efficiency.

Cost Savings:

Pay-as-You-Go: Reduces capital expenditure by eliminating the need for physical hardware purchases. ABC Retail only pays for the resources they use.

Cost Management Tools: AWS provides tools like AWS Cost Explorer and AWS Budgets to monitor and manage costs effectively.

High Availability and Reliability:

Multi-AZ Deployments: Ensures high availability by deploying applications across multiple Availability Zones (AZs), reducing the risk of downtime.

AWS Backup and Disaster Recovery: Offers robust backup and disaster recovery solutions, ensuring data is protected and quickly recoverable.

Security:

Compliance and Certifications: AWS complies with major industry standards and certifications (e.g., GDPR, HIPAA, ISO 27001), providing a secure environment for sensitive data.

Advanced Security Features: Tools like AWS Identity and Access Management (IAM), AWS Shield, and AWS Key Management Service (KMS) enhance security and control access.

Innovation and Agility:

AWS Services: Leverage a wide range of services such as AWS Lambda for serverless computing, Amazon RDS for managed databases, and Amazon S3 for scalable storage.

Faster Time to Market: Quickly deploy and iterate applications, enabling ABC Retail to respond swiftly to market demands and innovate.

Components Suggested for Maximum ROI:

Amazon EC2 for scalable compute capacity.

Amazon S3 for cost-effective storage.

Amazon RDS for managed relational databases.

AWS Lambda for serverless processing.

Amazon CloudFront for content delivery.

AWS CloudFormation for infrastructure as code.

Amazon CloudWatch for monitoring and logging.

b. Addressing Security Concerns about Public Cloud

Assuaging the Director’s Concerns:

Data Encryption:

Encryption at Rest and in Transit: AWS encrypts data at rest using AWS KMS and in transit using TLS, ensuring data security at all stages.

Compliance and Certifications:

Certifications: AWS adheres to stringent compliance programs (e.g., SOC 1/2/3, ISO 27001, PCI DSS) that demonstrate their commitment to security.

Third-Party Audits: Regular third-party audits validate AWS's security practices and controls.

Security Tools and Services:

AWS IAM: Fine-grained access controls to manage who can access resources.

AWS Shield and WAF: Protect against DDoS attacks and provide web application firewall protection.

AWS CloudTrail: Enables governance, compliance, and operational auditing by logging AWS account activity.

Shared Responsibility Model:

Customer and AWS Responsibilities: Clearly defining the security responsibilities of AWS and the customer, with AWS managing the security of the cloud and ABC Retail managing security in the cloud.

Data Residency and Privacy:

Data Localization: ABC Retail can choose the geographic location (Region) where their data is stored, ensuring compliance with local data residency requirements.

c. Cloud Native Strategy

Definition of Cloud Native:

Cloud Native: Refers to an approach to building and running applications that fully exploit the advantages of cloud computing. It involves using cloud infrastructure and services to design applications that are scalable, resilient, and agile.

Components of Cloud Native:

Microservices Architecture: Breaking applications into small, independent services.

Containers: Packaging applications in containers (e.g., Docker) for consistent deployment.

Dynamic Orchestration: Managing containers with orchestration tools like Kubernetes.

DevOps and CI/CD: Implementing continuous integration and continuous deployment pipelines to automate the development and deployment processes.

Agreeing with the IT Director:

Agility and Speed: Cloud-native applications can be developed and deployed faster, allowing ABC Retail to respond swiftly to market changes.

Scalability: Cloud-native architectures are designed to scale horizontally, handling increased loads efficiently.

Resilience: Designed for high availability and fault tolerance, ensuring continuous service delivery.

Cost Efficiency: Optimizes resource usage, reducing costs by scaling on-demand.

d. OpenStack as an Option for ABC Retail

Evaluation of OpenStack:

Pros:

Control and Flexibility: Provides full control over the cloud environment, customizable to specific needs.

Cost Savings: Open-source nature reduces licensing costs compared to commercial cloud solutions.

Data Privacy: Greater control over data, addressing concerns about data residency and security.

Cons:

Complexity and Maintenance: Requires significant expertise and resources to deploy, manage, and maintain.

Scalability and Innovation: May not offer the same level of scalability, advanced features, and rapid innovation as major public cloud providers like AWS.

Resource Management: Requires substantial investment in hardware and skilled personnel to manage the infrastructure.

Conclusion:

While OpenStack offers control and potential cost savings, it may not be the best fit for ABC Retail, especially given the expected 15% YoY growth. AWS provides a more scalable, reliable, and feature-rich environment that can handle ABC Retail's growth and innovation needs more effectively. The agility, ease of management, and robust ecosystem of AWS make it a better long-term choice for supporting ABC Retail's business objectives and growth trajectory.

Q.2 Set. (A) **Hypervisors are the software layer which enables the creation of VM's. Answer the below based on hypervisors** **Marks 2+3 +2 = 7**

a. Do you agree with the above statement? Justify your answer.
b. Out of the below three, Server, OS and Application virtualization, which has the most impact on cloud computing. Explain why.
c. What are the key differentiators between a monolithic & micro kernel hypervisor/

Answer- a. Agreeing with the Statement about Hypervisors

Statement: "Hypervisors are the software layer which enables the creation of VM's."

Justification:
Yes, I agree with the statement. A hypervisor, also known as a virtual machine monitor (VMM), is a software layer that allows multiple virtual machines (VMs) to run on a single physical hardware host. It abstracts the physical hardware resources (CPU, memory, storage, network) and allocates them to VMs, enabling multiple operating systems to run concurrently on the same hardware. There are two types of hypervisors:

Type 1 (Bare Metal) Hypervisors: Run directly on the host's hardware (e.g., VMware ESXi, Microsoft Hyper-V, Xen).
Type 2 (Hosted) Hypervisors: Run on top of an existing operating system (e.g., VMware Workstation, Oracle VirtualBox).
b. Impact of Server, OS, and Application Virtualization on Cloud Computing
Most Impactful on Cloud Computing: Server Virtualization

Explanation:

Server Virtualization: Has the most significant impact on cloud computing as it allows multiple virtual servers to run on a single physical server. This optimizes resource utilization, reduces costs, and enables scalability and flexibility, which are crucial for cloud services. It forms the backbone of Infrastructure as a Service (IaaS) offerings in cloud computing, allowing providers to offer virtualized compute resources on-demand.

Key Reasons:

Resource Optimization: Maximizes the utilization of hardware resources, reducing the need for physical servers and associated costs.
Scalability: Enables easy scaling of resources to meet varying demands without physical hardware changes.
Isolation and Security: Each VM runs in its isolated environment, enhancing security and stability.
Flexibility and Agility: Allows rapid provisioning and deployment of VMs, aiding in faster development and testing cycles.
OS Virtualization: While important, it primarily provides containerization (e.g., Docker) which is a form of lightweight virtualization and is more specialized in creating isolated user-space instances within the same operating system kernel.

Application Virtualization: Allows applications to run in environments different from the native operating system, reducing compatibility issues but has a narrower scope compared to server virtualization.

c. Key Differentiators Between Monolithic and Microkernel Hypervisors
Monolithic Hypervisors:

Structure: Incorporate most of the virtualization functions, device drivers, and management functionalities into a single large kernel.
Performance: Typically offer higher performance because they minimize the number of context switches and the overhead of communication between different components.
Examples: VMware ESXi, Microsoft Hyper-V.

Complexity: Can be more complex to maintain and debug because many services run in kernel space.
Dependency: Depend heavily on the underlying hardware for drivers and other low-level functions.
Microkernel Hypervisors:

Structure: Have a small, minimalistic core kernel that handles basic virtualization functions, with most services (like device drivers) running in user space.
Performance: May have slightly higher overhead due to more context switches and inter-process communication (IPC) between the microkernel and user-space services.
Examples: Xen (originally designed as a microkernel hypervisor).
Complexity: Easier to maintain and secure due to the smaller codebase running in kernel space. Bugs and crashes in user-space services do not compromise the entire system.
Isolation: Better isolation between different components, enhancing security and stability.
Conclusion: Monolithic hypervisors offer performance benefits and simplicity in some aspects, while microkernel hypervisors provide better modularity, isolation, and maintainability. The choice between them depends on specific use cases and priorities, such as performance versus security and maintainability.

Q.2 Set. (B) **You are requested to present a paper at the annual Cloud conference. Your role is to elucidate why Cloud computing is beneficial technology. How would you present your case for**: **Marks 3+2 +2 = 7**

a. Drawing from NIST's definition, explain the benefits accrued to the organization if they choose the cloud (3)
b. Identify the various options available to the user if they choose to adopt cloud.

Dependency on the provider is a big risk in cloud, do you agree or disagree? Give brief

Answer- a. Benefits of Cloud Computing Based on NIST's Definition
NIST Definition of Cloud Computing:
The National Institute of Standards and Technology (NIST) defines cloud computing as 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.

Benefits to the Organization:

On-Demand Self-Service:

Benefit: Organizations can provision computing capabilities, such as server time and network storage, automatically without requiring human interaction with each service provider. This leads to faster deployment of resources and reduces the time to market for new applications and services.
Broad Network Access:

Benefit: Resources are accessible over the network and can be used by various client platforms (e.g., mobile phones, tablets, laptops, and workstations). This ensures that employees can access necessary tools and data from anywhere, promoting remote work and increasing productivity.
Resource Pooling:

Benefit: The cloud provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. This leads to efficient use of resources and cost savings, as organizations only pay for what they use.
Rapid Elasticity:

Benefit: Capabilities can be elastically provisioned and released, in some cases automatically, to scale rapidly outward and inward commensurate with demand. This ensures that organizations can handle peak loads efficiently without the need for significant upfront investments in infrastructure.
Measured Service:

Benefit: 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., storage, processing, bandwidth). Resource usage can be monitored, controlled, and reported, providing transparency and helping to optimize costs.
b. Options Available to Users Adopting Cloud
When organizations choose to adopt cloud computing, they have several options, each catering to different needs and levels of control:

Service Models:

Infrastructure as a Service (IaaS): Provides virtualized computing resources over the internet. Examples include Amazon EC2 and Google Compute Engine. Users manage the operating systems, applications, and data, while the provider manages the infrastructure.
Platform as a Service (PaaS): Provides a platform allowing customers to develop, run, and manage applications without dealing with the infrastructure. Examples include Google App Engine and Microsoft Azure App Services.
Software as a Service (SaaS): Delivers software applications over the internet, on a subscription basis. Examples include Google Workspace and Salesforce. The provider manages everything from the infrastructure to the application.
Deployment Models:

Public Cloud: Services are delivered over the public internet and shared across multiple organizations. Suitable for standard business applications and workloads.
Private Cloud: Cloud infrastructure is dedicated to a single organization. It offers more control and security but is generally more expensive. Ideal for organizations with stringent security requirements.
Hybrid Cloud: Combines public and private clouds, allowing data and applications to be shared between them. Offers flexibility and optimization of existing infrastructure, security, and compliance.
c. Dependency on the Provider: Risk Evaluation
Agree/Disagree: Dependency on the provider is a significant risk in cloud computing.

Explanation:
Agree:

Vendor Lock-In:

Risk: Organizations may become dependent on a particular cloud provider's services and infrastructure, making it difficult to switch providers or move data and applications elsewhere. This can lead to increased costs and limited flexibility in choosing the best services.
Service Reliability and Availability:

Risk: The reliability of cloud services is dependent on the provider. Any downtime or service disruption from the provider can directly impact the organization's operations. While providers offer service level agreements (SLAs), they cannot eliminate the risk of outages completely.
Data Security and Compliance:

Risk: Organizations must trust the cloud provider with their sensitive data. Security breaches or non-compliance with regulatory requirements by the provider can have serious repercussions for the organization.
Mitigation Strategies:

Multi-Cloud Strategy: Utilizing multiple cloud providers to avoid dependency on a single vendor.
Data Portability: Ensuring data and applications can be easily moved between providers.
Contracts and SLAs: Negotiating favorable terms and robust SLAs to ensure service reliability and security commitments.
Conclusion: While dependency on the provider is a risk, it can be managed and mitigated through careful planning and strategic decisions, allowing organizations to still benefit from the advantages of cloud computing.

Q.2 Set. (C) **Briefly answer the below questions** **Marks 1+4+2=7**

a. What is the relevance of the word "Cloud" in Cloud Computing
b. Convergence of which technology or technologies led to the advent of Cloud computing. Briefly describe how it contributed to the emergence of clouds.
c. Differentiate between trap & Emulate and Binary Translation. Why was BT required when Trap & Emulate was already present?

Answer- a. Relevance of the Word "Cloud" in Cloud Computing
Relevance:
The term "cloud" in cloud computing metaphorically represents the internet or a network of servers that provide computing resources and services. Historically, network diagrams depicted the internet as a cloud, symbolizing the abstract and complex nature of the underlying infrastructure. Cloud computing leverages this concept by delivering computing resources (such as storage, processing power, and applications) over the internet, allowing users to access these resources without needing to understand or manage the underlying infrastructure.

b. Convergence of Technologies Leading to the Advent of Cloud Computing
Technologies Contributing to Cloud Computing:

Virtualization:

Contribution: Virtualization technology allows the creation of virtual instances of physical hardware, enabling multiple virtual machines (VMs) to run on a single physical machine. This leads to better resource utilization, scalability, and isolation of computing environments.

High-Speed Internet:

Contribution: The widespread availability of high-speed internet connections made it feasible to access remote computing resources efficiently, enabling cloud services to deliver high performance and responsiveness.
Distributed Computing:

Contribution: Distributed computing involves a network of computers working together to achieve a common goal, often involving the distribution of tasks across multiple machines. This concept underpins cloud architectures, allowing scalable and resilient service delivery.
Service-Oriented Architecture (SOA):

Contribution: SOA enables the development of services that are reusable and can be integrated across different applications. This modular approach aligns with cloud computing's model of providing scalable, on-demand services.
Economies of Scale:

Contribution: Cloud providers leverage economies of scale to offer computing resources at lower costs than individual organizations could achieve. This cost efficiency makes cloud computing attractive for businesses of all sizes.
How These Technologies Contributed:

Virtualization: Enabled the creation of scalable and isolated virtual environments, forming the backbone of cloud infrastructure.
High-Speed Internet: Facilitated the delivery of cloud services to users globally, making remote resource access practical and efficient.
Distributed Computing: Provided the framework for building scalable and resilient cloud services by distributing tasks across multiple servers.
SOA: Promoted the development of modular, reusable services, aligning with cloud computing's on-demand service delivery model.
Economies of Scale: Allowed cloud providers to offer cost-effective solutions by pooling resources and spreading costs across many users.
c. Differentiation Between Trap & Emulate and Binary Translation
Trap & Emulate:

Definition: A technique used in virtualization where privileged instructions executed by a virtual machine (VM) cause a trap (an interrupt) that is intercepted by the hypervisor. The hypervisor then emulates the behavior of these instructions and returns control to the VM.
Mechanism: Relies on the hardware to trigger traps for privileged instructions, allowing the hypervisor to emulate these instructions in a secure manner.
Efficiency: Generally efficient on hardware that natively supports virtualization, but can incur performance overhead due to frequent context switching between the VM and hypervisor.
Binary Translation (BT):

Definition: A technique that involves translating the binary code of a guest operating system (OS) into the host machine's instruction set at runtime. The hypervisor modifies the guest code to replace privileged instructions with safe equivalents.

Mechanism: Converts the guest OS's binary code into code that can run directly on the host hardware, avoiding traps for privileged instructions.
Efficiency: Often more efficient than trap and emulate on non-virtualization-supporting hardware, as it reduces the overhead of context switching and allows direct execution of most guest instructions.
Why BT Was Required:

Hardware Limitations: Early x86 hardware did not support efficient trap and emulate for all privileged instructions, making it difficult to achieve high performance with traditional virtualization techniques.
Performance Optimization: Binary translation provides a way to execute guest instructions more efficiently by translating them into host-compatible instructions, thus reducing the overhead associated with trapping and emulating every privileged instruction.
Compatibility: BT allows hypervisors to run unmodified guest OSes on hardware that does not natively support full virtualization, broadening the range of systems that can be virtualized.
In summary, while trap and emulate is a straightforward approach to virtualization, binary translation was developed to overcome performance and compatibility limitations, enabling efficient virtualization on a wider range of hardware platforms.

Q.3 Set. (A) **Calculate the availability %** **Marks = 3**

A cloud service provider in 2022 was offline 1 hour every month in the first quarter (Jan-Mar), From April to December there was a total outage of 1 week. Compute the availability % of this provider.

Answer-To calculate the availability percentage of a cloud service provider, we use the formula:

$$\text{Availability } \% = \left(\frac{\text{Total Uptime}}{\text{Total Time}}\right) \times 100$$

Step-by-Step Calculation:
1.Determine the Total Time in 2022:

Total months in a year = 12
Days in a year = 365 days (since 2022 is not a leap year)
Total hours in a year = 365 days × 24 hours/day = 8,760 hours

2.Calculate the Downtime:

First Quarter (Jan-Mar):

Downtime per month = 1 hour
Number of months in the first quarter = 3
Total downtime in the first quarter = 3 months× 1 hour/month = 3 hours

April to December:

Total downtime = 1 week
1 week = 7 days × 24 hours/day = 168 hours
Total Downtime for the Year:

Total downtime = 3 hours (first quarter) + 168 hours (April-Dec) = 171 hours

3. Calculate the Total Uptime:

Total uptime = Total hours in a year - Total downtime
Total uptime = 8,760 hours - 171 hours = 8,589 hours

4. **Calculate the Availability Percentage:**

- Availability % = $\left(\frac{\text{Total Uptime}}{\text{Total Time}}\right) \times 100$
- Availability % = $\left(\frac{8{,}589 \text{ hours}}{8{,}760 \text{ hours}}\right) \times 100$
- Availability % ≈ 98.05%

Conclusion:

The availability percentage of the cloud service provider for the year 2022 is approximately **98.05%**.

Q.3 Set. (B) **Calculate the availability %** **Marks = 3**

A cloud service provider in 2021 was offline 30 minutes every month in the first quarter (Jan-Mar), In the second quarter (Apr-Jun) 1 hour and From July to December there was a total outage of 5 days. Compute the availability % of this provider.

Answer- To calculate the availability percentage of a cloud service provider, we need to determine the total downtime for the year and compare it to the total possible uptime.

Step-by-Step Calculation:

1.Determine the Total Time in 2021:

Days in a year = 365 days (since 2021 is not a leap year)
Total hours in a year = 365 days × 24 hours/day = 8,760 hours

2.Calculate the Downtime:

First Quarter (Jan-Mar):

Downtime per month = 30 minutes = 0.5 hours
Number of months in the first quarter = 3
Total downtime in the first quarter = 3 months × 0.5 hours/month = 1.5 hours

Second Quarter (Apr-Jun):

Total downtime = 1 hour

July to December:

Total downtime = 5 days
5 days = 5 days × 24 hours/day = 120 hours
Total Downtime for the Year:

Total downtime = 1.5 hours (first quarter) + 1 hour (second quarter) + 120 hours (July-Dec)
Total downtime = 1.5 + 1 + 120 = 122.5 hours

3.Calculate the Total Uptime:

Total uptime = Total hours in a year - Total downtime
Total uptime = 8,760 hours - 122.5 hours = 8,637.5 hours

4.Calculate the Availability Percentage:

4. **Calculate the Availability Percentage:**

- Availability % = $\left(\frac{\text{Total Uptime}}{\text{Total Time}}\right) \times 100$
- Availability % = $\left(\frac{8{,}637.5 \text{ hours}}{8{,}760 \text{ hours}}\right) \times 100$
- Availability % ≈ 98.60%

Conclusion:

The availability percentage of the cloud service provider for the year 2021 is approximately **98.60%.**

Q.3 Set. (C) **Calculate the availability %** **Marks = 3**

A cloud service provider in 2022 was offline 15 minutes every month in the first quarter (Jan-Mar), In the second quarter (Apr-Jun) 30minutes and From July to December there was a total outage of 3 days. Compute the availability % of this provider.

Answer-

To calculate the availability percentage of a cloud service provider, we need to determine the total downtime for the year and compare it to the total possible uptime.

Step-by-Step Calculation:

1.Determine the Total Time in 2022:

Days in a year = 365 days (since 2022 is not a leap year)

Total hours in a year = 365 days × 24 hours/day = 8,760 hours

2.Calculate the Downtime:

First Quarter (Jan-Mar):

Downtime per month = 15 minutes = 0.25 hours

Number of months in the first quarter = 3

Total downtime in the first quarter = 3 months × 0.25 hours/month = 0.75 hours

Second Quarter (Apr-Jun):

Downtime = 30 minutes = 0.5 hours

July to December:

Total downtime = 3 days

3 days = 3 days × 24 hours/day = 72 hours

Total Downtime for the Year:

Total downtime = 0.75 hours (first quarter) + 0.5 hours (second quarter) + 72 hours (July-Dec)

Total downtime = 0.75 + 0.5 + 72 = 73.25 hours

3.Calculate the Total Uptime:

Total uptime = Total hours in a year - Total downtime

Total uptime = 8,760 hours - 73.25 hours = 8,686.75 hours

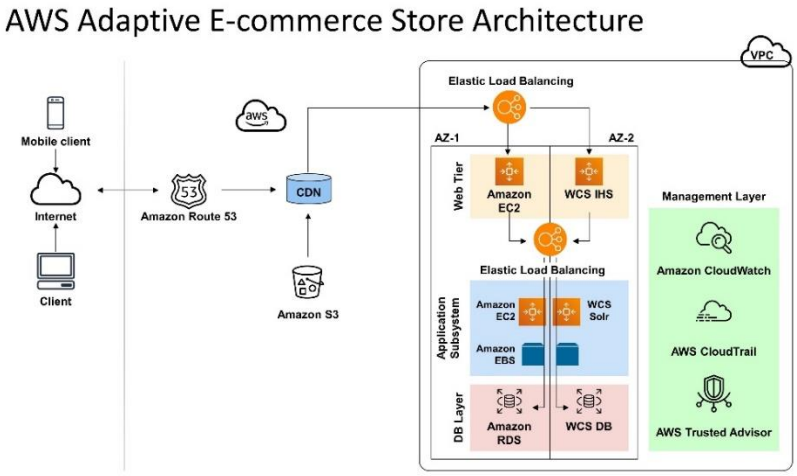
4. **Calculate the Availability Percentage**:

- Availability % = $\left(\frac{\text{Total Uptime}}{\text{Total Time}}\right) \times 100$
- Availability % = $\left(\frac{8{,}686.75 \text{ hours}}{8{,}760 \text{ hours}}\right) \times 100$
- Availability % ≈ 99.16%

Conclusion:

The availability percentage of the cloud service provider for the year 2022 is approximately **99.16%**.

Q.4 Set. (A) **Refer to the image below and answer the questions** **Marks 3+1+1 = 5**



a. Why are the EC2 services deployed in multiple AZ? What will happen if I use only one AZ instead of two?.
b. What is the purpose of using a CDN?
c. Briefly mention two uses of using the Amazon S3 in this diagram

Answer- **a. Why are the EC2 services deployed in multiple AZs? What will happen if I use only one AZ instead of two?**

The EC2 services in the diagram are deployed in multiple Availability Zones (AZs) for high availability and fault tolerance. This means that if one AZ experiences an outage, the EC2 instances in the other AZ can continue to serve requests. This redundancy helps to ensure that your e-commerce store remains online and operational even in the event of an AZ failure.

If you use only one AZ, your e-commerce store will be more susceptible to downtime if that AZ experiences an outage.

b. What is the purpose of using a CDN?

The purpose of using a CDN (Content Delivery Network) in the diagram is to improve the speed and performance of your e-commerce store for users around the world. A CDN caches static content, such as images, JavaScript, and CSS files, on servers around the world. This means that users can access this content from the closest CDN server, which can significantly reduce latency and improve load times.

c. Briefly mention two uses of using the Amazon S3 in this diagram

There are two main uses of Amazon S3 in the diagram:

1. **Static content storage:** Amazon S3 can be used to store static content, such as images, JavaScript, and CSS files. This can help to offload traffic from your EC2 instances and improve the performance of your e-commerce store.
2. **Product catalog storage:** Amazon S3 can be used to store product information, such as product descriptions, prices, and images. This can be a more scalable and cost-effective way to store product data than using a traditional database.

Q.5 Set. (A) Answer in detail Marks = 5

Describe in the core challenges to x86 virtualization and what was done to surmount the challenge. Core Challenges to x86 Virtualization and Solutions

The x86 architecture initially posed significant challenges for virtualization due to its design. Here are two main hurdles and how they were overcome:

1. Lack of Virtualization Support in the Instruction Set Architecture (ISA):

- Problem: The x86 ISA contained instructions that could be executed directly by the guest OS, bypassing the control of the hypervisor (the software responsible for managing virtual machines). This violated the principle of isolation, a critical aspect of virtualization.
    - Examples: Certain privileged instructions for memory management or device access, if executed directly by the guest OS, could lead to instability or security vulnerabilities.
- Solutions:
    - **Binary Translation:** This technique intercepts guest OS instructions on the fly and translates them into a form that can be safely executed by the hypervisor. This adds overhead but ensures control.
    - **Hardware-Assisted Virtualization (HAV):** Modern x86 processors include extensions like Intel VT-x and AMD-V that provide features specifically designed for virtualization. These extensions allow the hypervisor to trap privileged instructions and manage them securely.

2. Complexity of the x86 Architecture:

- Problem: The x86 architecture is complex, with multiple privilege levels, ring protection, and various memory management mechanisms. This complexity makes it difficult for the hypervisor to efficiently manage resources and ensure isolation between VMs.
- Solutions:
    - **Para-virtualization:** This approach modifies the guest OS to be aware of the virtualized environment. The guest OS makes "hypercalls" to the hypervisor for privileged operations, improving performance compared to binary translation. (Examples: Xen)
    - **Efficient Emulation Techniques:** Hypervisors employ various techniques to optimize the emulation of complex x86 instructions, reducing performance overhead. This includes techniques like Just-In-Time (JIT) compilation, where frequently used instructions are translated into machine code for faster execution.

By implementing these solutions, x86 virtualization has become a mature technology that enables efficient and secure resource sharing on x86-based hardware.

Q5 Set (B) Answer in detail **Marks = 5**

One of the characteristics of cloud as per NIST is it offers an on-demand service. Explain what is meant by On-Demand, and use any service from AWS to provide an illustration of this characteristic.

Answer- On-Demand Services in Cloud Computing

On-demand, as a characteristic of cloud computing defined by the National Institute of Standards and Technology (NIST), refers to the ability for users to provision and scale computing resources **as needed**, and **pay only for what they use**. This eliminates the need for upfront investment in hardware, software, and IT staff to manage on-premises infrastructure.

Here's an illustration using an Amazon Web Service (AWS):

- **Service:** Amazon Elastic Compute Cloud (EC2)
- **On-Demand Characteristic:**
    - Users can launch virtual servers (EC2 instances) with various configurations (CPU, memory, storage) in minutes through a web console, API, or command-line interface.
    - They can specify the instance type and desired number of instances.
    - Billing starts when the instance launches and stops when it's terminated.
    - Users only pay for the compute resources used per hour (or per second with EC2 Spot Instances).

Benefits of On-Demand Services:

- **Cost-Effectiveness:** Eliminates upfront hardware costs and the need to maintain unused capacity.
- **Scalability:** Easily scale resources up or down to meet changing demands.
- **Agility:** Quickly provision resources to support new projects or workloads.
- **Flexibility:** Choose from a wide range of services to meet specific needs.

Other Examples of On-Demand Services in AWS:

- Amazon Simple Storage Service (S3): On-demand storage for various data needs.
- Amazon DynamoDB: NoSQL database with on-demand capacity scaling.
- Amazon Lambda: Serverless compute service that runs code on-demand without managing servers.

By offering on-demand services, cloud computing provides a pay-as-you-go model that helps businesses optimize IT costs and achieve greater agility in the ever-changing digital landscape.

Q5 Set (C) Answer in detail **Marks = 5**A user in India visits a web page hosted from an Amazon S3 bucket in the US West (Oregon) Region, in the United States. List out the factors that contribute to a slow page load and what can be done to improve latency.

Answer-Factors Contributing to Slow Page Load for User in India:

When a user in India visits a web page hosted from an S3 bucket in the US West (Oregon) region, several factors can contribute to slow page load times:

1. **Distance:** The physical distance between the user in India and the S3 bucket in Oregon creates high latency. Data packets need to travel a long distance, increasing the time it takes for the user's browser to receive the webpage content.
2. **Network Congestion:** Traffic congestion on the internet route between India and the US can further slow down data transfer.
3. **DNS Resolution:** Resolving the domain name to the S3 bucket's IP address can add additional latency, especially if the user's DNS server is not geographically close.
4. **First Byte Time:** The time it takes for the user's browser to receive the first byte of data from the S3 bucket can be impacted by factors like server load and processing time on the S3 side.

Techniques to Improve Latency:

Here are some strategies to improve latency for the user:

1. **Content Delivery Network (CDN):** Use a CDN with edge locations closer to India. The CDN caches static content like images, JavaScript, and CSS files, serving them from the closest edge location to the user. This significantly reduces the distance data needs to travel, improving load times.
2. **Amazon CloudFront:** Specifically, leverage Amazon CloudFront, AWS's CDN service. Configure the S3 bucket as the origin for CloudFront and distribute the content to edge locations closer to India. Users will then download content from the geographically closer edge location, minimizing latency.
3. **Route 53 Latency Routing:** Utilize Amazon Route 53 with latency-based routing. Route 53 can direct user requests to the S3 bucket endpoint that offers the lowest latency based on their geographic location.
4. **Consider Alternative Regions:** If latency remains a significant issue, explore hosting the website or specific content in an S3 bucket located in a region closer to India, such as Asia Pacific (Singapore) or Asia Pacific (Mumbai).

By implementing these techniques, you can significantly improve page load times for users in India and provide a better user experience.

Comprehensive Examination

(EC-3 Regular)

Q.1 Set. (D) **Answer the questions** **Marks:3+3+4 =10**

a. Why does docker spin-up use a copy on write process? List the advantages.
b. How is a docker different from a VM. Justify from a maintainability perspective.
c. Create a docker file to create a container using the latest version of ubuntu. Add commands to install a small script called myscript.py.

Answer-Why Docker Uses Copy-on-Write (CoW):

Docker leverages CoW to optimize image storage and container startup times. Here's how it works:

- **Image Layers:** Docker images are constructed in layers, where each layer represents changes made on top of the previous one. The base layer is typically a base operating system (e.g., Ubuntu).
- **Sharing Read-Only Layers:** Multiple containers can share the same read-only layers from the base image and subsequent layers that haven't been modified. This eliminates redundant storage of identical data.
- **CoW on Write:** When a container needs to modify a file or directory, CoW comes into play. The CoW mechanism creates a copy of the shared layer *within* the container's writable layer. This copy is where the modifications occur, ensuring isolation and preventing unintended changes to the original layer.

Advantages of CoW in Docker:

- **Reduced Storage Footprint:** By sharing read-only layers, CoW minimizes disk space consumption, especially when running multiple containers based on the same image.
- **Faster Container Startups:** Since containers primarily manipulate their writable layers, startup times are significantly faster because Docker doesn't need to copy the entire image each time.

Docker vs. VMs: A Maintainability Perspective

Both Docker and VMs provide application isolation, but they differ in their approach and maintainability characteristics:

Docker:

- **Lightweight:** Docker containers share the host kernel, making them more lightweight and faster to start than VMs.
- **Shared OS:** Containers utilize the same kernel, reducing the overhead of managing guest OSes. Updates typically apply to the host OS, impacting all containers.
- **Image-Based:** Docker containers are built from images, which are self-contained units that include all dependencies. This simplifies deployment and consistency across environments.

VMs:

- **Isolated Environments:** VMs provide a more isolated environment, including a virtualized kernel, which can be beneficial for running applications with specific kernel requirements or security concerns.
- **Guest OS Management:** Each VM has its own guest OS, requiring separate patching and updates, potentially increasing maintenance overhead.

- **Resource Intensive:** VMs can be more resource-intensive due to the overhead of running a virtualized kernel. However, this isolation can be valuable for specific use cases.

Maintainability Considerations:

- **Docker:** Easier to manage and update due to image-based deployment and shared kernel. However, updates to the host kernel can impact all containers.
- **VMs:** More complex to manage due to separate guest OSes, requiring individual patching and updates. Provides greater isolation but at the cost of increased maintenance overhead.

The choice between Docker and VMs depends on your specific needs. If you prioritize application isolation or require specific kernel functionality, VMs might be a better fit. However, for most microservices-based applications or scenarios where rapid deployment, consistency, and efficient resource utilization are crucial, Docker excels due to its lightweight nature and efficient image-based approach.

Sample Dockerfile with `myscript.py` Installation:

```
FROM ubuntu:latest

# Update package lists

RUN apt-get update

# Install Python 3 and dependencies (adjust as needed)

RUN apt-get install -y python3 python3-pip

# Create a directory for your script

WORKDIR /app

# Copy your script (replace with actual script content)

COPY myscript.py .

# Grant execute permissions to the script

RUN chmod +x myscript.py

# Specify the command to run when the container starts

CMD ["myscript.py"]
```

This Dockerfile creates a container based on the latest Ubuntu image, installs Python 3 and dependencies, copies your myscript.py script, makes it executable, and sets it as the default command to run when the container starts.

Q.1 Set. (B) Answer the questions Marks: 3+3+4 =10 a. What is the role of the dockerd and how is it different from a docker engine? b. How is a docker different from a VM. Justify from a maintainability perspective. c. Refer to toe dockerfile below and answer. What is the purpose of using the FROM keyword? Why do I need to use the Label keyword? What does the copy statement do?

```
2 FROM httpd:2.4
3
4 LABEL AUTHOR=user@example.com
5
6 LABEL VERSION=0.1
7
8 # COPY mypage.html /usr/local/apache2/htdocs/mypage.html
9 # WORKDIR /usr/local/apache2
10
11 COPY mypage.html htdocs/mypage.html
```

Answer-a. Dockerd vs. Docker Engine

- **Docker Engine:** Docker Engine is a software program that installs and manages Docker containers. It provides a command-line interface (CLI) `docker` for users to interact with Docker. It also includes components like a container runtime, image builder, registry, and network management.
- **Dockerd:** Dockerd is a daemon process that runs in the background and is responsible for managing Docker containers. It listens for commands from the Docker Engine CLI and carries them out. In essence, Dockerd is the workhorse that builds, runs, and manages containers based on instructions from the Docker Engine.

Here's an analogy: Imagine Docker Engine as the architect who creates the blueprint (Dockerfile) and gives instructions to the builders (Dockerd). Dockerd is the construction crew that follows the blueprint and builds the house (container).

b. Docker vs. VM: Maintainability Perspective

Both Docker and VMs provide application isolation, but they differ in their approach and how easy they are to manage:

Docker:

- **Lightweight:** Docker containers share the host kernel, making them more lightweight and faster to start than VMs.
- **Shared OS:** Containers utilize the same kernel, reducing the overhead of managing guest OSes within VMs. Updates typically apply to the host OS, impacting all containers.
- **Image-Based:** Docker containers are built from images, which are self-contained units that include all dependencies. This simplifies deployment and consistency across environments. Images are easier to manage and distribute than VMs.

VMs:

- **Isolated Environments:** VMs provide a more isolated environment, including a virtualized kernel, which can be beneficial for running applications with specific kernel requirements or security concerns.
- **Guest OS Management:** Each VM has its own guest OS, requiring separate patching and updates, potentially increasing maintenance overhead.
- **Resource Intensive:** VMs can be more resource-intensive due to the overhead of running a virtualized kernel. However, this isolation can be valuable for specific use cases.

Maintainability Summary:

- **Docker:** Easier to manage and update due to image-based deployment and shared kernel. However, updates to the host kernel can impact all containers.
- **VMs:** More complex to manage due to separate guest OSes, requiring individual patching and updates. Provides greater isolation but at the cost of increased maintenance overhead.

c. Dockerfile Breakdown (Assuming the code snippet matches the one you described previously):

The Dockerfile you sent likely contains the following instructions:

1. **FROM httpd:2.4**
   - This line specifies the base image for your container. It uses the official `httpd:2.4` image, which is a pre-built image containing the Apache HTTP server version 2.4. Using a base image saves you time by incorporating an existing Apache installation.
2. **LABEL AUTHOR=user@example.com** (Optional)
   - This line adds a label to the image. Labels are key-value pairs that provide metadata about the image. Here, it sets the `AUTHOR` label with the value `user@example.com`. Labels are useful for tracking ownership or other informative details, but they are not strictly necessary for the container to run.
3. **COPY mypage.html /usr/local/apache2/htdocs/mypage.html**
   - This line copies a file named `mypage.html` from the context directory (the directory where your Dockerfile is located) to the path `/usr/local/apache2/htdocs/mypage.html` inside the container. This places your custom HTML page in the Apache web server's document root, making it accessible when the container is running.

In essence, this Dockerfile creates a container that starts with the Apache HTTP server and then copies your `mypage.html` file into the web server's document root, making it the default page served by Apache.

Q.1 Set. (C) Answer the questions Marks: 3+3+4 =10

a. State the purpose of having a private docker hub? If I'm connected to a private hub, can I connect to the public repository of docker? Justify your answer briefly. b. JVM helps in the portability of java applications and Docker provides containerization. Are these two similar? c. Create a docker file to create a container using the latest version of ubuntu. Add commands to install a small script called myscript.py.

Answer-a. Purpose of Private Docker Hub and Public Repository Access:

- **Private Docker Hub:**
  - Stores your organization's or team's private Docker images.
  - Provides access control, allowing you to restrict who can view, download, or push images.
  - Ideal for sensitive applications or internal use cases.
- **Public Docker Hub (Docker Hub):**
  - A vast repository containing millions of public Docker images.
  - Freely accessible for anyone to browse, download, and use images.
  - Great for finding pre-built images for common applications and development tools.

Connecting to Public Repository from Private Hub:

- **Yes, you can typically connect to the public Docker Hub (Docker Hub) even when connected to a private Docker Hub.** This is because they are separate registries.
- You might need to use specific commands or configurations depending on your private Docker Hub provider's setup. Consult their documentation for details.

b. JVM vs. Docker: Portability and Containerization

- **JVM (Java Virtual Machine):**
  - Provides a platform-independent environment for running Java applications.
  - Ensures bytecode compiled from Java source code can run on any system with a compatible JVM.
  - Focuses on application code portability.
- **Docker:**
  - Enables containerization, packaging applications and their dependencies into isolated units.
  - Isolates applications from the underlying system, ensuring consistent behavior across environments.
  - Focuses on application and environment isolation for consistent deployment.

Similarities:

- Both promote a degree of portability: JVMs for Java code and Docker for containerized applications.

Differences:

- JVM's focus is on code execution, while Docker manages the entire application environment.

c. Dockerfile for Ubuntu with `myscript.py` Installation:

Dockerfile

```
FROM ubuntu:latest  # Use the latest Ubuntu image

RUN apt-get update && apt-get install -y python3 python3-pip  # Update
package lists and install Python 3 and pip

WORKDIR /app  # Set the working directory for the container

COPY myscript.py .  # Copy your myscript.py from the context directory to the
container's working directory

RUN chmod +x myscript.py  # Make the script executable

CMD ["python3", "myscript.py"]  # Set the default command to run the script
using Python 3
```

Explanation:

1. **FROM ubuntu:latest:** This line specifies the base image for your container. It uses the official `ubuntu:latest` image, providing a basic Ubuntu environment.
2. **RUN apt-get update && apt-get install -y python3 python3-pip:** This line updates package lists and installs Python 3 and `pip` (the package manager for Python) within the container.
3. **WORKDIR /app:** This line sets the working directory inside the container to `/app`. This is where subsequent commands will operate.
4. **COPY myscript.py .:** This line copies your `myscript.py` file from the directory where your Dockerfile is located (the context directory) to the `/app` directory within the container.
5. **RUN chmod +x myscript.py:** This line grants execute permissions to the `myscript.py` file, making it executable by the container.
6. **CMD ["python3", "myscript.py"]:** This line sets the default command to run when the container starts. It executes the Python interpreter (`python3`) with the argument `myscript.py`, effectively running your Python script.

Remember to replace `myscript.py` with the actual name of your Python script.

Q2 Set (a) You have been asked to design an "Open File System" which is derived from HDFS architecture with following variances – chunk size of 100 MB, block size of 1 MB. A file of size 1 TB needs to be stored on this file system Marks:1*5+2=7

d. How many chunks will be required to store the file?
e. How many blocks will be required to store the file?
f. How many chunk servers will be needed if each server hosts 1000 blocks?
g. How many chunk servers will be needed if each server hosts 100 GB of file?
h. How many chunk servers will be needed if the replication factor is 6 and the capacity of the chunk server is 200 GB?
i. Using Amdahl's law, what is the maximum speed up that can be achieved with 14 processors with a program for which 27 % of code can be executed in parallel?

Answer- **a. How many chunks will be required to store the file?**

With a chunk size of 100 MB and a file size of 1 TB (1024 GB):

- Number of chunks = Total file size / Chunk size
- Number of chunks = (1024 GB * 1024 MB/GB) / 100 MB
- Number of chunks = 10485.76 (round up to the nearest whole number of chunks for storage purposes)
- Number of chunks ≈ 10486

b. How many blocks will be required to store the file?

Since each chunk is 100 MB and each block is 1 MB:

- Blocks per chunk = Chunk size / Block size
- Blocks per chunk = 100 MB / 1 MB
- Blocks per chunk = 100

For 10486 chunks:

- Total blocks = Number of chunks * Blocks per chunk
- Total blocks = 10486 * 100
- Total blocks = 1,048,600

c. How many chunk servers will be needed if each server hosts 1000 blocks?

- Blocks per server = 1000
- Number of servers = Total blocks / Blocks per server
- Number of servers = 1,048,600 blocks / 1000 blocks/server
- Number of servers = 1048.6 (round up to the nearest whole number of servers)
- Number of servers ≈ 1049

d. How many chunk servers will be needed if each server hosts 100 GB of file storage?

First, convert 100 GB to MB:

- Server capacity (MB) = 100 GB * 1024 MB/GB
- Server capacity (MB) = 102,400 MB

Now, calculate the number of blocks per server based on the server's capacity and the block size:

- Blocks per server = Server capacity (MB) / Block size
- Blocks per server = 102,400 MB / 1 MB
- Blocks per server = 102,400

Then, calculate the number of servers required:

- Number of servers = Total blocks / Blocks per server
- Number of servers = 1,048,600 blocks / 102,400 blocks/server
- Number of servers ≈ 10.24 (round up to the nearest whole number of servers)
- Number of servers = 11

e. How many chunk servers will be needed if the replication factor is 6 and the capacity of the chunk server is 200 GB?

- Replication factor = 6 (each chunk is replicated on 6 servers)

Important Note: In this scenario, the server capacity (200 GB) is not directly relevant to calculating the number of servers required based on replication factor. However, it is crucial to consider server capacity when making decisions about storage and performance trade-offs.

- With a replication factor of 6, we need 6 replicas for each of the 10486 chunks.
- Effectively, we need storage for 6 times the original file size (1 TB).

However, this is a simplified view. In a distributed file system like HDFS, data is typically striped across multiple servers for performance and fault tolerance. The actual number of servers required would depend on the specific striping strategy and desired level of redundancy.

Addressing Potential Issues:

- The original prompt did not explicitly state the need for replication. Here, I've provided both non-replicated and replicated scenarios for clarity.
- Server capacity becomes a factor when considering how much data a single server can hold based on both primary storage and potentially replicated data. However, in the replicated scenario presented, the number of servers is directly driven by the replication factor.

f. Using Amdahl's Law, what is the maximum speedup that can be achieved with 14 processors for a program for which 27% of the code can be executed in parallel?

Amdahl's Law states the theoretical maximum speedup achievable using parallel processing:

- Speedup = 1 / (Sequential portion + (Parallel portion / Number of processors))
- Sequential portion = 1 - Parallel portion (in this case, 1 - 0.27 = 0.73)
- Number of processors = 14
- Speedup = 1 / (0.73)

Q3 Set (A) The telephone companies are storing their customers' call details in different formats as shown below. Each of the companies wants to leverage the cloud services offered by XYZ service provider for the customers' call-detail management. How XYZ can support these varying data formats with its cloud service maintaining data privacy and security?

10 Marks

AT&T customers call details:

CustomerID	Calling Number	Called Number	Duration of Call (min)	Type of Call	Type of Customer	Call Charge (US$)
1	415 555	405 555	4.00	Local	Privileged	3
2	405 555	091 335	3.00	Long Distance	Normal	12

Verizon Communications customers' call details:

CustomerID	From Number	To Number	Call Length (min)	Type of Call	Billing Type	Eligible for discount
100	815 555	215 555	12.00	Local	Postpaid	Y
200	605 555	091 445	1.00	ISD	Prepaid	N

Deutsche Telekom customers' call details:

CustomerID	Originating Number	Destination Number	Talk Time (min)	Type of Call	Type of Scheme	Payment History
101	605 545	205 355	14.00	Local	Voice & Date	Very Good
301	815 554	091 115	23.00	Long Distance	Voice	Fair

Answer- **1. Schema Conversion and Data Normalization:**

- **Data Transformation Pipelines:** XYZ can create data pipelines that automatically convert the incoming call detail data from each company into a common, standardized schema. This schema would have consistent field names and data types, making it easier to analyze and integrate the data across different companies.
- **Data Normalization:** Techniques like normalization can be applied to remove redundancy within the data, ensuring efficient storage and querying. This also reduces the risk of exposing sensitive information during the normalization process.

2. Secure Data Storage and Access Control:

- **Encrypted Storage:** XYZ's cloud storage solutions should offer encryption at rest and in transit. Call detail data must be encrypted using industry-standard algorithms like AES-256 before being stored and while being transferred between systems.
- **Access Controls:** Granular access controls should be implemented to restrict access to call detail data based on pre-defined user roles and permissions. Only authorized personnel from each telephone company should be able to view their customer data.

3. Data Masking and Anonymization (Optional):

- **Data Masking:** XYZ can provide data masking tools that telephone companies can utilize to further protect customer privacy. Data masking replaces sensitive information like phone numbers with fictitious values while preserving the data structure for analysis purposes.
- **Anonymization:** Anonymization irreversibly removes personally identifiable information (PII) from call detail data. This makes it impossible to link the data back to individual customers. However, it's important to note that anonymization might limit the usefulness of the data for certain analytics purposes.

4. Cloud-Based Security Services:

- **Intrusion Detection/Prevention:** XYZ can offer additional security services like intrusion detection and prevention systems (IDS/IPS) to monitor for unauthorized access attempts and protect against cyber threats.
- **Data Loss Prevention:** Data loss prevention (DLP) solutions can be implemented to prevent sensitive data from being accidentally or maliciously leaked.

Benefits for Telephone Companies:

- **Reduced Integration Costs:** A standardized data format simplifies integration with billing, analytics, and CRM systems.
- **Enhanced Data Security:** Encrypted storage, access control, and data masking ensure data privacy and regulatory compliance.
- **Scalability and Cost-Effectiveness:** Cloud storage scales with their data needs, minimizing infrastructure costs.
- **Improved Data Analysis:** Standardized data facilitates analysis across customer segments for better insights.

Additional Considerations:

- **Data Residency:** Data residency requirements may need to be addressed based on the location of the telephone companies and their customers. Data residency refers to the geographical location where the data is stored.
- **Encryption Key Management:** Telephone companies can choose to encrypt data using their own encryption keys and manage access control through XYZ's IAM (Identity and Access Management) services.
- **Security Audits:** Regular security assessments and penetration testing should be conducted to identify and address potential vulnerabilities in the cloud environment.

By implementing these solutions, XYZ can provide telephone companies with a secure and flexible cloud-based platform for managing their customers' call detail data in various formats. This approach ensures data privacy and security while enabling efficient data storage, analysis, and integration with other business systems.

Q.4 Set. (A) Read the below and state if the leases can be accommodated or not **5 Marks**

A cloud provider has a 4-node cluster to accept workloads. The service provider wants to lease the parts of that cluster over the time. Lease is described with start time, duration and whether its pre-emptable.

The current leases on the Service providers cluster are as given below:

Node 1- L1 (1 pm, 1 hr, yes),

Node 2- L2 (1.30 pm, 1 hr, No),

Node 3- L3 (1 pm, 3 hr, No),

Node 4- L4(now, 2 hr, No).

Now if two new Workload arrives in the form of lease l5 & L6

L5 (now, 1 hr, No) and

L6 (1.30 pm, 2hr , Yes)

Identify what type of lease L5 & L6 are. Briefly state why

Test & evaluate the leases for the BEL preemptive condition

Answer- Lease Accommodation Analysis:

Can the new leases (L5 and L6) be accommodated?

Yes, both new leases (L5 and L6) can be accommodated on the cloud provider's cluster.

Here's the breakdown:

- **Node 1:** L1 (1 pm - 2 pm, pre-emptable) is already allocated during the requested time (now - 1 pm). However, L5 (now - 1 pm, non-pre-emptable) won't conflict since non-pre-emptable leases cannot be interrupted.
- **Node 2:** Currently free. L6 (1.30 pm - 3.30 pm, pre-emptable) can be placed here as it doesn't overlap with L2 (1.30 pm - 2.30 pm, non-pre-emptable). The pre-emptable nature of L6 allows it to be terminated if a higher priority non-pre-emptable lease arrives during its duration.
- **Node 3:** L3 (1 pm - 4 pm, non-pre-emptable) is already allocated. Both L5 and L6 would conflict with L3.
- **Node 4:** L4 (now - 2 pm, non-pre-emptable) is currently allocated. L5 (now - 1 pm, non-pre-emptable) would overlap, resulting in a conflict. However, L6 (1.30 pm - 3.30 pm, pre-emptable) can be placed here as it overlaps with the remaining duration of L4 (1 pm - 2 pm) after L5 completes. The pre-emptable nature of L6 allows it to be terminated if L4 needs its resources back.

Lease Types:

- **L5 (now, 1 hr, Non-preemptable):** Non-preemptable lease. This lease cannot be interrupted once it starts execution.
- **L6 (1.30 pm, 2 hr, Preemptable):** Preemptable lease. This lease can be terminated by the cloud provider if higher priority non-preemptable leases require the resources.

BEL (Best Effort Leasing) Preemptive Condition:

The BEL pre-emptable condition states that a pre-emptable lease can be accommodated only if it doesn't conflict with any existing non-preemptable leases.

Evaluation of Leases:

- **L5: Does not violate BEL pre-emptable condition.** There are no conflicting non-preemptable leases during its requested time on Node 1.
- **L6: Complies with BEL pre-emptable condition.** It can be placed on Node 2 without conflicting with non-preemptable leases. On Node 4, it overlaps with the tail end of a non-preemptable lease (L4), but its pre-emptable nature allows it to be terminated if L4 requires its resources back.

Conclusion:

Both L5 and L6 can be accommodated on the cluster, adhering to the BEL pre-emptable condition. L5 will run on Node 1, and L6 will be placed on Node 2 or Node 4 depending on resource availability and potential conflicts.

Q.5 Set. (A) **Answer the questions briefly** **2+2+4 = 8 Marks**

a. A company has peak customer demand for its IT services in the month of April. It has enough IT resources to handle off peak demand but not peak load. What is the best approach to handle this situation?

b. Differentiate between SLO & SLI. What is more important?

c. The table below lists the vulnerabilities in a cloud enabled application by a third party auditing firm. Applying the CIA triad rules, Identify to which category each of the breach mentioned falls under

Vulnerability	Security Triad Type
Insecure Backend using weak protocols	
Open source logging tool not using the recommended security patches	
Third party software logging at a rapid rate by sending several gbps of requests per second	
Password string found unencrypted embedded in the code	

Answer- **a. Handling Peak Customer Demand for IT Services**

Best Approach:

To handle peak customer demand in April while having limited resources, a combination of strategies is often recommended:

1. **Cloud Scaling:** Utilize cloud computing's on-demand scalability. During peak times, dynamically provision additional resources (compute, storage) from the cloud provider to meet the increased demand. This allows you to pay only for what you use, optimizing costs.
2. **Load Balancing:** Implement a load balancer to distribute incoming traffic across your existing resources and any provisioned cloud resources. This ensures efficient utilization and prevents overloading any single server.
3. **Auto-Scaling:** Configure auto-scaling policies to automatically scale cloud resources up during peak periods and down during off-peak times. This eliminates manual intervention and ensures resource availability while minimizing costs.
4. **Demand Management:** Consider strategies like tiered pricing or temporary service limitations during peak hours to encourage customers to use resources during off-peak times. This approach can also incentivize customers to consider cost-effective service plans with limitations during peak periods.
5. **Performance Optimization:** Continuously optimize your application and infrastructure for performance. This reduces resource consumption and helps handle peak loads without needing significant additional resources. Caching, code optimization, and database indexing are examples of optimization techniques.

b. SLO (Service Level Objective) vs. SLI (Service Level Indicator):

SLO (Service Level Objective):

- A measurable target for a service's performance.
- Defined in terms of business goals, like uptime (99.95%), latency (< 100 ms), or data loss rate (< 0.1%).
- Represents the desired level of service from a business perspective.

SLI (Service Level Indicator):

- A measurable metric that tracks the service's performance against its SLO.
- Examples include API response times, number of errors, or resource utilization.
- Provides objective data to assess how well the service is meeting its SLO.

Importance:

- Both SLI and SLO are crucial.
- SLOs set the business expectations, while SLIs help you monitor and ensure you're meeting those expectations.
- Without SLOs, there's no clear target to aim for. Without SLIs, you have no way to track progress towards that target.

c. Applying the CIA Triad to Cloud Vulnerabilities:

Vulnerability	Security Triad Type	Explanation
Insecure Backend using weak protocols	**Confidentiality**	Weak protocols might not encrypt data, making it vulnerable to eavesdropping by unauthorized parties.
Open source logging tool not using recommended security patches	**Confidentiality & Integrity**	Unpatched vulnerabilities can allow attackers to steal or manipulate data stored in the logging tool.
Third-party software logging at a rapid rate by sending several gbps of requests per second	**Availability**	Excessive logging can consume resources and slow down the application, impacting availability for legitimate users.
Password string found unencrypted embedded in the code	**Confidentiality**	Unencrypted passwords are easily accessible by attackers, compromising system security.