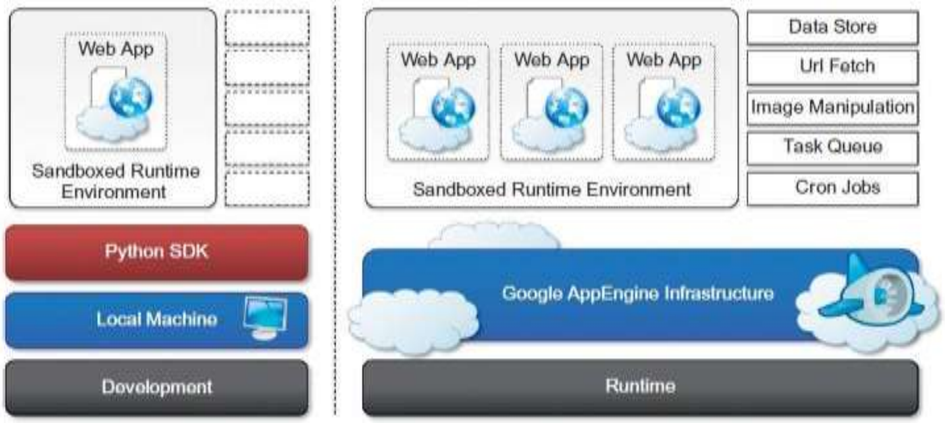
USN

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | |

# Internal Assessment Test 3– Dec 2024

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Sub: | Cloud Computing | | | | | Sub Code: | 21CS72 | Branch: | CSE |
| Date: | 16.12.2024 | Duration: | 90 mins | Max Marks: | 50 | Sem / Sec: | 7 B,C | | OBE |
| | | | Answer any FIVE FULL Questions | | | | MARKS | CO | RBT |

| | | | | |
|---|---|---|---|---|
| 1 (a) | Describe the core components of Google App Engine with a neat diagram | 7M | CO5 | L2 |



- ▶ Infrastructure
- ▶ runtime environment
- ▶ Underlying storage
- ▶ Set of scalable services

Infrastructure

- ▶ AppEngine hosts Web applications
- ▶ primary function is to serve users **requests efficiently.**
- ▶ For each ***HTTP request***
  - ▶ **AppEngine** locates the servers hosting the application that processes the request
  - ▶ evaluates their load
  - ▶ if necessary, allocates additional resources
- ▶ Also monitors **application performance**
- ▶ Collects **statistics** on which the billing is calculated.

Runtime Environment

- ▶ represents the execution context of applications hosted on AppEngine
- ▶ **Sandboxing :** provide the application environment with an **isolated and protected** context
  - ▶ It can execute without **causing threat to the server** or be **influenced by other applications**.
- ▶ **Supported Runtimes** : Java, Python, and Go.

Storage Services

**Storage for semi-structured data**

**Long term storage for static file servers**

- ▶ **Static File Servers :** components that define the graphical layout of the application (CSS files,

plain HTML files, JavaScript files, images, icons, and sound files) or data files.

  - ▶ Hosted in static servers that are not modified often

    - Servers are optimized for storing static content.
- **Data Store:** allows developers to store semi-structured data
    - Designed **to scale and optimized to quickly access** data.
    - Object described in terms **of entity and properties.**
    - A large object database where the object can be retrieved **using a key.**
    - type of the **key and the structure** of the object can vary.
    - Provides facilities to **create indexes** on data.
    - Uses **Optimistic concurrency control**: If one user tries to update an entity that is already being updated, the control returns and the operation fails

Application Services

- **UrlFetch:** Applications can make **synchronous and asynchronous** Web requests
    - integrate these resources into the normal request- handling cycle of the application
    - It can set **deadlines** for requests so that they can be completed (or aborted) within a given time.
- **MemCache :**
    - This is a **distributed in-memory cache** that is optimized for fast access and provides developers

with a volatile store for the objects that are frequently accessed.

- 
    - **Caching algorithm** automatically **removes objects** that are rarely **accessed.**
    - Significantly **reduces access time to data**.

Compute Services

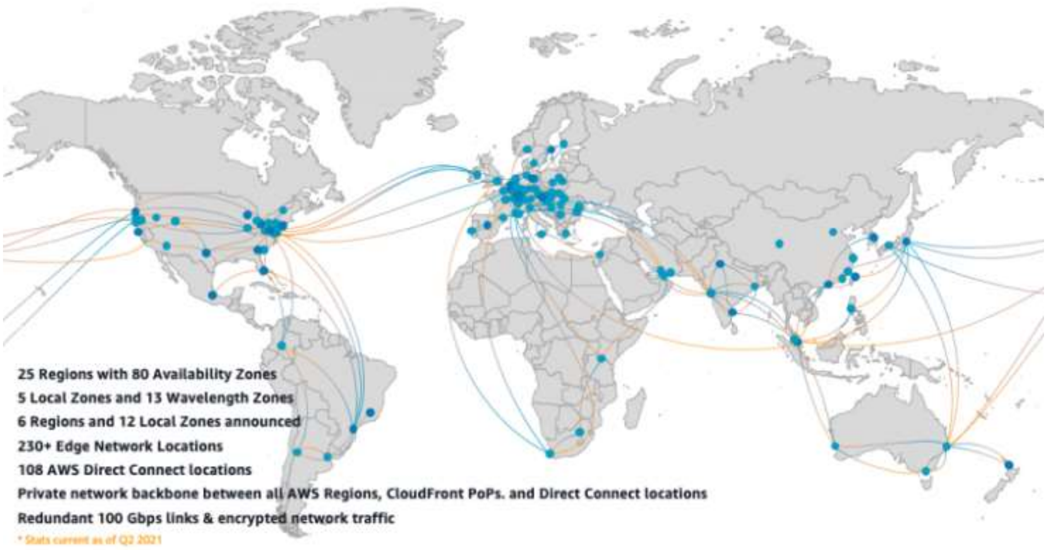
- **Task Queues:** allow applications to **submit a task for a later execution**
    - useful for long computations that cannot be completed within the maximum response time of a request handler.
    - The service allows **users to have up to 10 queues** that can execute tasks at a configurable rate.
- **Cron jobs :** to perform an operation at a specific time of the day.
    - Operates similar to Task Queues
    - But, invokes the request handler specified in the task at a given time
    - does not re-execute the task in case of failure

Application Lifecycle

- AppEngine provides support for almost all the phases characterizing the life cycle of an application:
    - testing and development
    - Deployment
    - Monitoring
- **Application development and testing :**
    - Developers can start building their Web applications on a local development server.
    - It is a self-contained environment that helps developers tune applications without uploading them to AppEngine.
    - AppEngine provides mock implementation of DataStore, MemCache, UrlFetch, and the other

services

- **Java SDK**
    - The SDK supports the development of applications by using the servlet abstraction

| | | | | |
|---|---|---|---|---|
| | ► Google AppEngine plug-in is available for Eclipse.<br>► Developers can easily create Web applications by using the Eclipse Web<br>Platform, which provides a set of tools and components<br>► **Python SDK**<br>► GoogleAppEngineLauncher<br>► Create application identifier<br>► Access from web browser http://<app-id>/.appspot.com<br>► An application is measured against billable quotas, fixed quotas, and per-minute quotas.<br>► **Billable quotas** daily quotas set up by application administrator.<br>► AppEngine will ensure that the application does not exceed these quotas<br>► **Free quotas** are part of the billable quota : part of it that is free<br>► **Fixed quotas** are internal quotas set by AppEngine<br>► Identifies operations application can carry out in the infrastructure or runtime<br>► generally bigger than billable quotas<br>► **Per-minute quotas**<br>► Once an application reaches the quota for a given resource, the resource is depleted and will not be available to the application until the quota is replenished | | | |
| (b) | <br> 25 Regions with 80 Availability Zones<br>5 Local Zones and 13 Wavelength Zones<br>6 Regions and 12 Local Zones announced<br>230+ Edge Network Locations<br>108 AWS Direct Connect locations<br>Private network backbone between all AWS Regions, CloudFront PoPs. and Direct Connect locations<br>Redundant 100 Gbps links & encrypted network traffic<br>* Stats current as of Q2 2021<br>The picture above shows AWS Global cloud infrastructure. Briefly explain i)Regions ii)Availability Zones iii) Edge locations<br>A **region** is a set of datacenters deployed within a latency-defined perimeter and connected through a dedicated regional low-latency network. Eg. Asia Pacific, Canada, Europe<br>**Availability Zones** - separated groups of datacenters within a region. Availability zones are close enough to have low-latency connections to other availability zones. They're connected by a high-performance network<br>**Edge Locations** : **Edge locations are data centers designed to deliver services with the lowest latency possible.** These datacenters across the globe are closer to users than Regions or Availability Zones | 3M | CO5 | L2 |

| | | | | |
|---|---|---|---|---|
| 2 (a) | Explain storage and communication services in AWS<br>Storage<br>▶ S3 Key concepts: accessible through Representational State Transfer (REST) interface<br>▶ The storage is organized in a two-level hierarchy<br>▶ Stored objects cannot be manipulated like standard files<br>▶ Content is not immediately available to users.<br>▶ Requests will occasionally fail.<br>▶ A bucket is a container of objects.<br>▶ virtual drive hosted on the S3 distributed storage<br>▶ provides users with a flat store to which they can add objects.<br>▶ A bucket is located in a specific geographic location eventually replicated fault tolerance and better content distribution.<br>▶ Users create a bucket by sending a PUT request to http://s3.amazonaws.com/ with the name of the bucket and,<br>▶ They may want to specify the availability zone, additional information about the preferred location.<br>▶ Content of a bucket can be listed by sending a GET request<br>▶ The deletion of a bucket is performed by a DELETE request<br>▶ **Objects** constitute the content elements stored in **S3.**<br>▶ Users either **store files** or push to **the S3 text stream** representing the **object's content.**<br>▶ An object is identified by a **name** that needs to be **unique** within the bucket in which the content is stored<br>▶ name cannot be longer than **1,024 bytes** when encoded in **UTF-8,** and it allows almost<br>**any character.**<br>▶ buckets do not support nesting<br>▶ characters normally used **as path separators** are allowed.<br>▶ This actually compensates for the lack of a **structured file system**<br>▶ directories can be **emulated** by properly naming objects.<br>▶ Objects can be tagged with **metadata**, which are passed as **properties** of the **PUT request**<br>▶ **Access Control Policies (ACPs) :** An ACP is a **set of grant permissions.**<br>▶ attached to a resource expressed by means of **XML configuration file.**<br>▶ A policy allows defining up to **100 access rules**<br>▶ Each grants one of the available permissions to a grantee.<br>▶ READ<br>▶ WRITE<br>▶ READ_ACP<br>▶ WRITE_ACP<br>▶ FULL_CONTROL<br>▶ Grantees can be either **single users or groups**: users can be identified by their **canonical IDs** or<br>**email addresses** used for sign up<br>Amazon EBS<br>▶ They accommodate up to **1 TB** of space<br>▶ accessed through a **block device interface**<br>▶ Allows users to format them according to the needs of the instance they are connected to (raw storage, file system, or other) | 7M | CO5 | L2 |

- content of an **EBS volume survives** the instance life cycle and is persisted into **S3**
- EBS volumes can be cloned, used as boot partitions, and constitute durable storage

Amazon Elasticache

- ElastiCache is an implementation of an **elastic memory cache based on a cluster of EC2 instances**.
- provides fast data access from other **EC2 instances** through a

**Memcached-compatible protocol**.

- based on a **cluster of EC2 instances** running the caching software

available through Web services

- An ElastiCache cluster can be **dynamically resized** according to the demand of the client applications.
- automatic **patch management** and **failure detection** and recovery of

**cache nodes**

Structured Storage Solutions

- **Preconfigured EC2 AMIs:** predefined templates featuring an installation of a given database management system
- **Amazon RDS:** A relational database service that relies on the EC2 infrastructure and is managed by Amazon
  - 2 features : multi-AZ deployment and read replicas
  - Optimal for Oracle and MySQL migrated to AWS
- **Amazon SimpleDB:** a lightweight, highly scalable, and flexible data storage solution for applications that do not require a fully relational model for their data
  - semistructured data
  - Based on concepts of domains, items, and attributes
  - Provides fewer constraints on the structure of data entries which leads to improved performance.

SimpleDB

- domains as top-level elements to organize a data store
- Domains are comparable to tables.
- Unlike tables they allow all items to have the same column structure.
- Each item is represented as a collection of attributes expressed as a key-value pair.
- Each domain can grow up to 10 GB
- A single user can allocate a maximum of 250 domains
- Clients can create, delete, modify and make snapshots of domain.
- Select clause supports the following test operators : =, !=. <,>, <=, >=, like, not like, between, is null, is not null, isempty()
- **select from domain_name where every(attribute_name)='value'**
- The select query can extend its boundaries of **a single domain**, hence querying a huge amount of data.

Communication Services

- Virtual Networking : comprises a collection of services that allow AWS users to control the connectivity to and between **compute and storage services**.
- Amazon Virtual Private Cloud (VPC) and Amazon Direct Connect: provides connectivity

solutions in terms of **infrastructure**

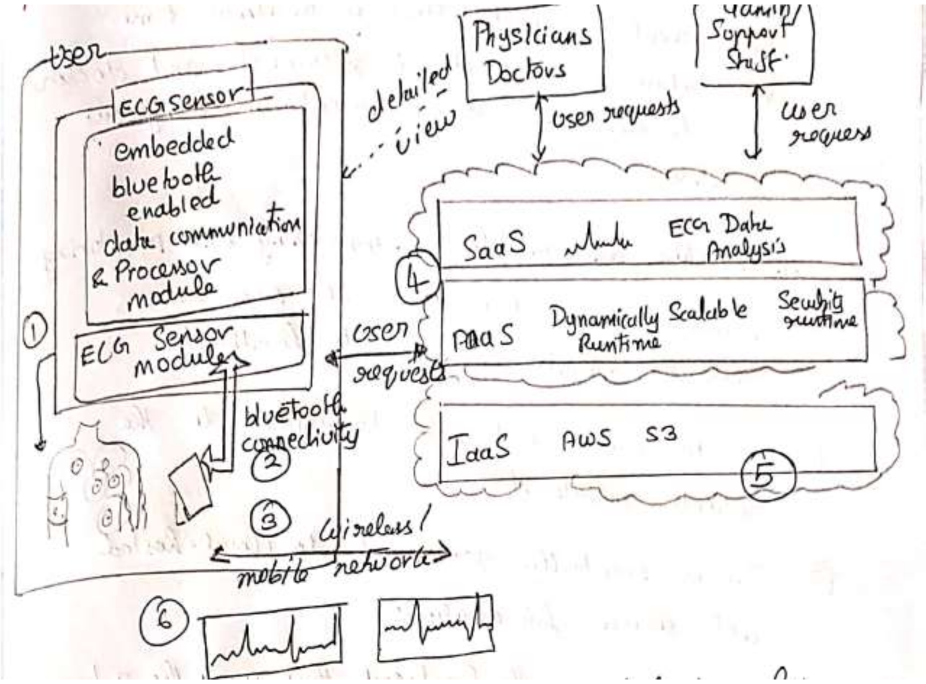
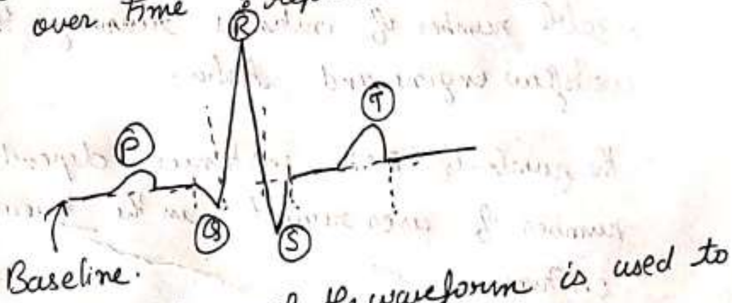
- **Amazon Virtual Private Cloud (VPC)**

    - service providers prepare either templates or customizable network service for advanced configurations.
    - Prepared templates : public subnets, isolated networks, private networks accessing internet through NAT, hybrid networks including AWS resources and private resources.
- **Amazon Direct Connect**
    - Allows AWS users to create **dedicated networks** between the **user private network** and Amazon Direct

Connect locations, called **ports**.

- **Route 53**
    - facilitates connectivity in terms of **naming**
    - implements dynamic DNS services that allow AWS resources to be reached through domain names other than amazon.com domain.
- **Amazon SQS** : is a disconnected model for exchanging **messages** between applications by means of **message queues**.
    - The messages are securely and redundantly stored within the AWS infrastructure.
    - While a message is being read, it is kept locked to avoid spurious processing from other application.
- **Amazon SNS : publish-subscribe** method for connecting heterogeneous applications
    - In SQS, it is necessary to continuously poll a given queue for a new message to process
    - SNS allows applications to be notified when new content of interest is available
    - accessible through a Web service where AWS users can create a topic and other applications can subscribe to.
    - It provides subscribers with different subscription models (**HTTP/HTTPS, email JSON, and SQS**)
- **Amazon SES** : provides AWS users with a scalable email service that leverages the AWS infrastructure
    - Users have to provide an email to SES and SES will send emails on their behalf.
    - A verification mail will be sent to user.
    - User is given a sandbox SES to test the service.
    - it is possible to send either SMTP-compliant emails or raw emails.
    - It also give **statistics to improve email campaigns** for effective communication with customers.

| | | | | |
|---|---|---|---|---|
| (b) | An encoding service integrates with AWS technologies EC2, S3 and CloudFront.<br>Briefly explain each of these services.<br>EC2 is used to use a virtual machine that is powerful enough to run the encoding services. S3 to store the media file in buckets. CloudFront is used to make the service available across the globe in edge locations that reduce transfer times in the content delivery network. | 3M | CO5 | L2 |

| | | | | |
|---|---|---|---|---|
| 3 (a) | With a neat diagram explain how cloud computing model can support ECG monitoring<br><br> User<br>ECG Sensor<br>embedded bluetooth enabled data communication & Processor module<br>ECG Sensor module<br>1<br>bluetooth connectivity<br>2<br>3<br>Wireless/ mobile network<br>6<br>detailed view<br>Physicians Doctors<br>Admin/ Support Staff<br>User requests<br>user requests<br>SaaS ECG Data Analysis<br>4<br>PaaS Dynamically Scalable Runtime<br>Security runtime<br>User requests<br>IaaS AWS S3<br>5<br>.healthcare is a domain where computer technology — several and diverse applications.<br>· Electrocardiogram (ECG) data analysis on the cloud.<br>What is ECG?<br>- electrical manifestation of the contractile activity of the hearts myoccardium.<br>- this activity produces a specific waveform that is repeated over time represents the heartbeat:<br><br> <br>Analysis of the shape of the waveform is used to identify arrhythmias. (irregular heartbeat).<br>Occurs when signals that coordinate the heart's beats | 8M | CO5 | L2 |

- Cloud computing technologies allow remote monitoring of a patients heartbeat data.
- Data analysis can be done in minimal time.
- notification of the first-aid personnel and doctors may be done to reveal potentially dangerous situations.

Infrastructure and model for supporting ECG monitoring:

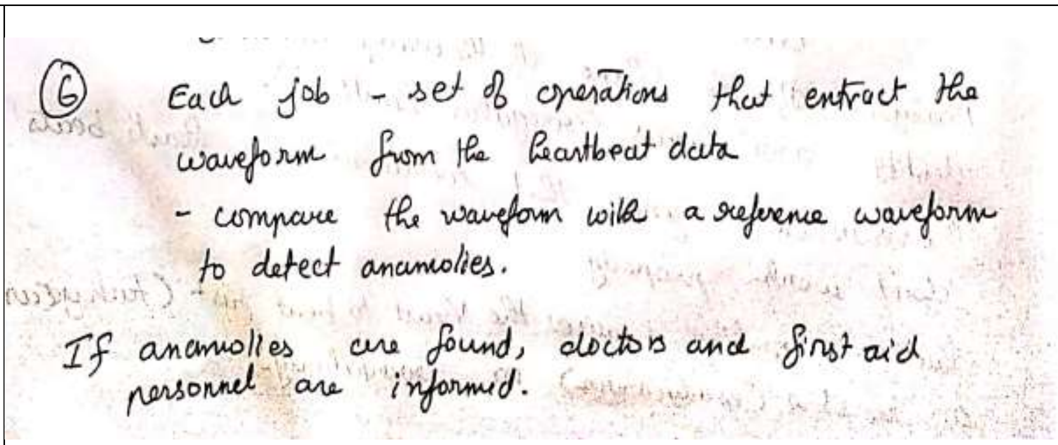
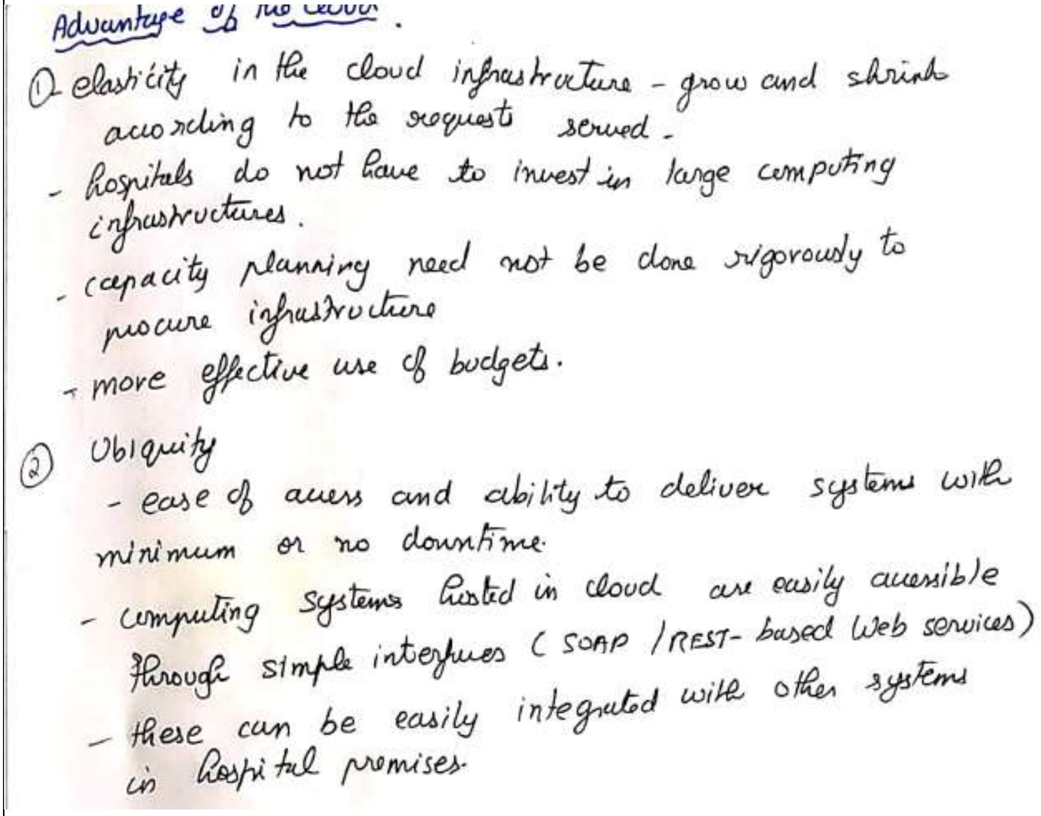
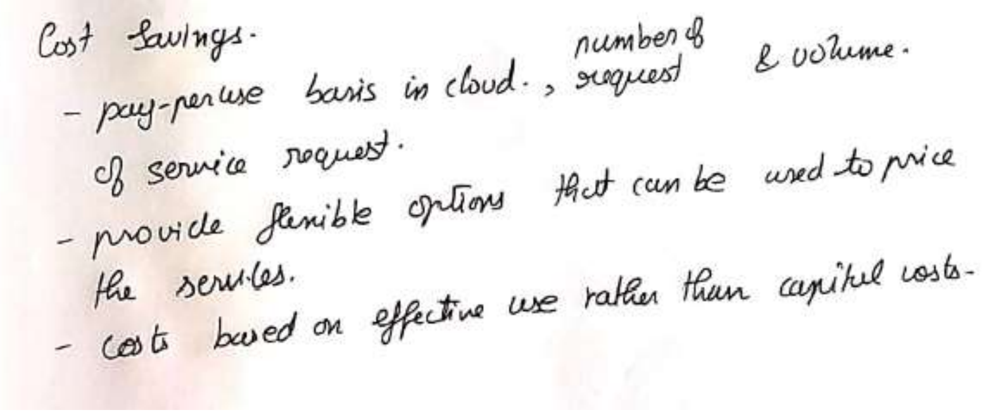
① wearable devices equipped with ECG sensors constantly monitor the patients heartbeat.

② This information is ~~transfer~~ transmitted to the patients mobile device.

③ This is eventually forwarded to cloud-hosted web service for analysis.

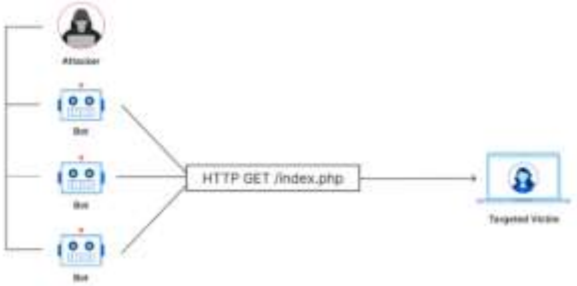
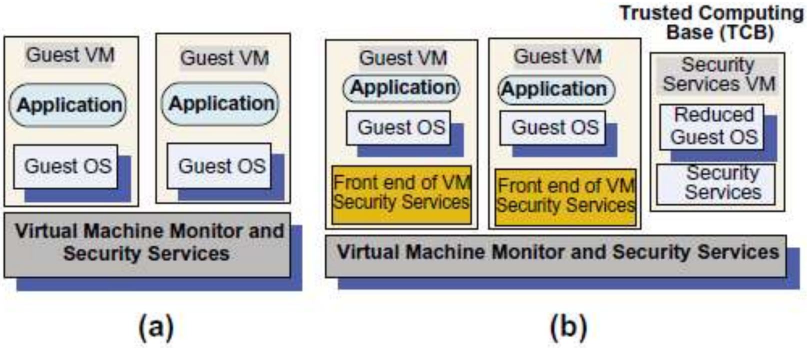
④ Web Service is the front-end that use the 3 layers of the computing stack: SaaS, PaaS, IaaS.

⑤ - Web Service in SaaS application will store the ECG data in Amazon S3.
- It issues a processing request to scalable cloud platform.
- The runtime platform is composed of dynamically sizable number of instances running the workflow engine and Aneka.
- the number of EC2 instances depends on the number of user requests in the queue of each instance

| | | | | |
|---|---|---|---|---|
| | <br>⑥ Each job - set of operations that extract the waveform from the heartbeat data<br>- compare the waveform with a reference waveform to detect anamolies.<br>If anamolies are found, doctors and first aid personnel are informed.<br><br>Advantage of the cloud.<br>① elasticity in the cloud infrastructure - grow and shrink according to the requests served.<br>- hospitals do not have to invest in large computing infrastructures.<br>- capacity planning need not be done rigorously to procure infrastructure<br>- more effective use of budgets.<br>② Ubiquity<br>- ease of access and ability to deliver systems with minimum or no downtime.<br>- computing systems hosted in cloud are easily accessible through simple interfaces ( SOAP / REST- based Web services)<br>- these can be easily integrated with other systems in hospital premises.<br><br>③ Cost savings.<br>- pay-per use basis in cloud., number of request & volume of service request.<br>- provide flexible options that can be used to price the services.<br>- costs based on effective use rather than capital costs. | | | |
| (b) | How could game log processing in online multiplayer games benefit from cloud computing. | 2M | CO5 | L2 |

| | | | | |
|---|---|---|---|---|
| | • Players update the game server hosting the game session, and the server integrates all the updates into a log that is made available to all the players through a TCP port<br>• The client software used for the game con- nects to the log port and, by reading the log, updates the local user interface with the actions of other players.<br>• Game log processing is also utilized to build statistics on players and rank them. These features constitute the additional value of online gaming portals that attract more and more gamers. The processing of game logs is a potentially compute-intensive operation that strongly depends on the number of players online and the number of games monitored. Moreover, gaming portals are Web applications and therefore might suffer from the spiky behavior of users that can randomly generate large amounts of volatile workloads that do not justify capacity planning. | | | |
| 4 (a) | Explain in detail security risks posed by shared images.<br>**Process to create an AMI**<br>• start from a running system, from another AMI, or from the image of a VM and copy the contents of the file system to the *S3*, the so-called *bundling*.<br>• first of the three steps in bundling is to create an image, the second step is to compress and encrypt the image, and the last step is to split the image into several segments and then upload the segments to the *S3*<br>• To use an image, a user has to specify the resources, provide the credentials for login, provide a<br>• firewall configuration, and specify the region<br>• Once instantiated, the user is informed about the public DNS and the virtual machine is made available<br>• A *Linux* system can be accessed using ssh at port 22, whereas the Remote Desktop at port 3389 is used for *Windows*<br>***Results of Audit***<br>• allowed a user to *undelete* files and recover credentials, private keys, or other types of sensitive information with little effort and using standard tools.<br>• 98% of the *Windows* AMIs (249 out of 253) and<br>• 58% of *Linux* AMIs (2,005 out of 3,432) audited had critical vulnerabilities<br>**3 Types of Security Risks**<br>Three types of *security risks* were analyzed: (1) backdoors and leftover credentials, (2) unsolicited connections, and (3) malware<br>**Backdoors and leftover credentials**<br>• To rent a *Linux* AMI, a user must provide the public part of the ssh key, and this key is stored in the authorized_keys in the home directory.<br>• This opens a backdoor for a malicious creator of an AMI who does not remove his own public key from the image and can remotely log into any instance of this AMI.<br>• Another backdoor is opened when the ssh server allows password-based authentication and the malicious creator of an AMI does not remove his own password. | 8M | CO5 | L2 |

|  |  |  |  |  |
|---|---|---|---|---|
|  | • This backdoor is opened even wider as one can extract the password hashes and then crack the passwords using a tool such as John the Ripper<br>**Unsolicited Connections**<br>• Another threat is posed by the omission of the cloud-init script that should be invoked when the image is booted. This script, provided by Amazon, regenerates the host key an ssh server uses to identify itself; the public part of this key is used to authenticate the server. When this key is shared among several systems, these systems become vulnerable to *man-in-the middle* attacks<br>• *Unsolicited connections* pose a serious threat to a system. Outgoing connections allow an outside entity to receive privileged information, e.g., the IP address of an instance and events recorded by a *syslog* daemon to files in the *var/log* directory of a *Linux* system. Such information is available only to users with administrative privileges.<br>• Some of the unsolicited connections are legitimate – for example, connections<br>• to a software update site. It is next to impossible to distinguish legitimate from malicious connections.<br>**Malware**<br>• *Malware*, including viruses, worms, spyware, and trojans, were identified using *ClamAV*, The first trojan carries out keylogging and allows stealing data from the files system and monitoring processes; the AMI also included a tool called *Trojan.Firepass* to decrypt and recover passwords stored by the *Firefox* browser<br>• A malicious agent can recover the *AWS* API keys that are not password protected.<br>• Another avenue for a malicious agent is to recover ssh keys stored in files named id_dsa and id_rsa. Though ssh keys can be protected by a passphrase,13 the audit determined that the majority of ssh keys (54 out of 56) were not password protected<br>• Recovery of deleted files containing sensitive information poses another risk for the provider of an image. When the sectors on the disk containing sensitive information are actually overwritten by another file, recovery of sensitive information is much harder. |  |  |  |
| (b) | ,<br> Amazon Polly<br> <br> <br> <br> amazon Transcribe | 2M | CO5 | L2 |

| | | | | |
|---|---|---|---|---|
| | i) Which AWS service lets you test Reinforcement Learning models on racing using model race cars on physical track? - **DeepRacer**<br>ii) Which AWS service performs text to speech and AI voice generator? – **Amazon Polly** | | | |
| 5 (a) | Briefly describe these attacks i) DDoS ii)Phishing iii) cross-site scripting iv) SQL injection<br>• A **distributed denial-of-service (DDoS)** attack is a malicious attempt to disrupt the normal traffic of a targeted server, service or network by overwhelming the target or its surrounding infrastructure with a flood of Internet traffic.<br><br> Attacker<br>Bot<br>Bot<br>Bot<br>HTTP GET /index.php<br>Targeted Victim<br>• **Phishing** is an attack aiming to gain information from a site database by masquerading as a trustworthy entity. Such information could be names and credit card numbers, Social Security Numbers (SSN), or other personal information stored by online merchants or other service providers.<br>• **SQL injection** is a form of attack typically used against a Web site. An SQL command entered in a Web form causes the contents of a database used by the Web site to be dumped to the attacker or altered. SQL injection can be used against other transaction-processing systems and is successful when the user input is not strongly typed and/or rigorously filtered.<br>• **Cross-site scripting** is the most popular form of attack against Web sites. A browser permits the attacker to insert client scripts into the Web pages and thus bypass the access controls at the Web site. | 4M | CO 4 | L2 |
| (b) | Write short notes on virtual machine security<br><br> Trusted Computing Base (TCB)<br>Guest VM<br>Application<br>Guest OS<br>Guest VM<br>Application<br>Guest OS<br>Virtual Machine Monitor and Security Services<br>Guest VM<br>Application<br>Guest OS<br>Front end of VM Security Services<br>Guest VM<br>Application<br>Guest OS<br>Front end of VM Security Services<br>Security Services VM<br>Reduced Guest OS<br>Security Services<br>Virtual Machine Monitor and Security Services<br>(a)<br>(b)<br>**FIGURE 9.2**<br>(a) Virtual security services provided by the VMM. (b) A dedicated security VM.<br>• The Trusted *computing base* (TCB) is a necessary condition for security in a virtual machine environment; if the TCB is compromised, the security of the entire system is affected. | 6M | CO 4 | L2 |

- A VMM controls the execution of privileged operations and can thus enforce memory isolation as well as disk and network access.
- The VMMs are considerably less complex and better structured than traditional operating systems; thus, they are in a better position to respond to security attacks.
- A major challenge is that a VMM sees only raw data regarding the state of a guest operating system, whereas security services typically operate at a higher logical level, e.g., at the level of a file rather than a disk block perating on the same hardware.
- The state of a guest virtual machine can be saved, restored, cloned, and encrypted by the VMM. Not only can replication ensure reliability, it can also support security, whereas cloning could be used to recognize a malicious application by testing it on a cloned system and observing whether it behaves normally normally.
- Another interesting possibility is to have the guest VM's files moved to a dedicated VM and thus, protect it from attacks
- Sophisticated attackers are able to fingerprint virtual machines and avoid VM *honeypots* designed to study the methods of attack.
- They can also attempt to access VM-logging files and thus recover sensitive data; such files have to be very carefully protected to prevent unauthorized access to cryptographic keys and other sensitive data.

- VM-based intrusion detection systems such as `Livewire` and `Siren`, which exploit the three capabilities of a virtual machine for intrusion detection: isolation, inspection, and interposition.

NIST project has identified the followingVMM-and VM-based threats:

**• VMM-based threats:**

1. **Starvation of resources and denial of service** for someVMs. Probable causes: (a) badly configured resource limits for some VMs; (b) a rogue VM with the capability to bypass resource limits set in the VMM.
2. **VMside-channel attacks**.Malicious attacks on one or more VMs by a rogue VM under the same VMM. Probable causes: (a) lack of proper isolation of inter-VM traffic due to misconfiguration of the virtual network residing in the VMM; (b) limitation of packet inspection devices to handle high-speed traffic, e.g., video traffic; (c) presence of VM instances built from insecure VM images, e.g., a VM image having a guest OS without the latest patches.
3. Buffer overflow attacks.

**VM-based threats:**

1. **Deployment of rogue or insecure VM**. Unauthorized users may create insecure instances from images or may perform unauthorized administrative actions on existing VMs. Probable cause: improper configuration of access controls on VM administrative tasks such as instance creation, launching, suspension, reactivation, and so on.
2. **Presence of insecure and tampered VM images** in the VM image repository. Probable causes: (a) lack of access control to the VM image repository; (b) lack of mechanisms to verify the integrity of the images, e.g., digitally signed image.

| | | | | |
|---|---|---|---|---|
| 6 (a) | State the 4 widely accepted fair information practices for collecting personal identifying information<br>four widely accepted fair information practices:<br>1. *Notice*. Web sites would be required to provide consumers clear and conspicuous notice of their information practices, including what information they collect, how they collect it (e.g., directly or through nonobvious means such as cookies), how they use it, how they provide Choice, Access, and Security to consumers, whether they disclose the information collected to other entities, and whether other entities are collecting information through the site.<br>2. *Choice*.Web sites would be required to offer consumers choices as to how their personal identifying information is used beyond the use for which the information was provided (e.g., to consummate a transaction). Such choice would encompass both internal secondary uses (such as marketing back to consumers) and external secondary uses (such as disclosing data to other entities).<br>3. *Access*.Web sites would be required to offer consumers reasonable access to the information aWeb site has collected about them, including a reasonable opportunity to reviewinformation and to correct inaccuracies or delete information.<br>4. *Security*.Web siteswould be required to take reasonable steps to protect the security of the information they collect from consumers. The Commission recognizes that the implementation of these practices may vary with the nature of the information collected and the uses to which it is put, as well as with technological developments. For this reason, the Commission recommends that any legislation be phrased in general terms and be technologically neutral. | 4M | CO 4 | L2 |
| (b) | Write short notes on the trust necessary in online activities.<br>• *trust* means “assured reliance on the character, ability, strength, or truth of someone or something it enables cooperative behavior, promotes adaptive organizational forms, reduces harmful conflict, decreases transaction costs, facilitates formulation of ad hoc workgroups, and promotes effective responses to crisis .<br>• Two conditions must exist for trust to develop. The first condition is *risk*, the perceived probability of loss; indeed, trust would not be necessary if there were no risk involved, if there is a certainty that an action can succeed.<br>• The second condition is *interdependence*, the idea that the interests of one entity cannot be achieved without reliance on other entities.<br>• A trust relationship goes though three phases:<br>o (1) a building phase, when trust is formed<br>o (2) a stability phase, when trust exists<br>o (3) a dissolution phase, when trust declines.<br>• Utilitarian reasons could be based on the belief that the costly penalties for breach of trust exceed any potential benefits from opportunistic behavior. This is the essence of *deterrence-based* trust. Another reason is the belief that the action involving the other party is in the self-interest of that party. This is the so-called *calculus-based* trust.<br>• After a long sequence of interactions, *relational trust* between entities can develop based on the accumulated experience of dependability and reliance on each other. | 6M | CO 4 | L2 |

| | | | | |
|---|---|---|---|---|
| | • The trust in the Internet “obscures or lacks entirely the dimensions of character and personality, nature of relationship, and institutional character” of traditional trust<br>• The missing identity, personal characteristics, and role definitions are elements we have to deal with in the context of online trust.<br><br>• The Internet offers individuals the ability to obscure or conceal their identities.<br>• The resulting anonymity reduces the cues normally used in judgments of trust.<br>• The identity is critical for developing trust relations; it allows us to base our trust on the past history of interactions with an entity.<br>• Anonymity causes mistrust because identity is associated with accountability and, in the absence of identity, accountability cannot be enforced.<br><br>• To remedy the loss of clues,we need security mechanisms for access control, transparency of identity, and surveillance.<br>• The mechanisms for access control are designed to keep intruders and mischievous agents out. Identity transparency requires that the relationship between a virtual agent and a physical person should be carefully checked through methods such as biometric identification.<br>• Digital signatures and digital certificates are used for identification.<br>• Surveillance could be based on *intrusion detection* or on logging and auditing.<br>• The first option is based on real-time monitoring, the second on offline sifting through audit records.<br>• Credentials are used when an entity is not known.<br>• Credentials are issued by a trusted authority and describe the qualities of the entity using the credential<br>• *Policies* and *reputation* are two ways of determining trust.<br>• Policies reveal the conditions to obtain trust and the actions to take when some of the conditions are met.<br>• Policies require the verification of credentials.Reputation is a quality attributed to an entity based on a relatively long history of interactions with or possibly observations of the entity.<br>• In a computer science context, “trust of a party A to a party B for a service X is the measurable belief<br>• of A in that B behaves dependably for a specified period within a specified context (in relation to service X)” | | | |

CI CCI HOD

| CO-PO and CO-PSO Mapping | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Course Outcomes | | Blooms Level | Modules covered | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 | PSO3 | PSO4 |
| CO 1 | Understand and analyze various cloud computing platforms and service provider. | L2 | 1 | 3 | 2 | - | - | - | 3 | - | - | - | - | - | - | - | 2 | - | 2 |
| CO 2 | Illustrate various virtualization concepts. | L2 | 2 | 3 | 2 | - | - | 2 | - | - | - | - | - | - | - | - | 2 | - | 2 |
| CO 3 | Identify the architecture, infrastructure and delivery models of cloud computing. | L2 | 3 | 3 | 2 | - | - | 2 | - | 3 | - | - | - | - | - | - | 2 | - | 2 |
| CO 4 | Understand the Security aspects of cloud | L2 | 4 | 3 | 2 | - | - | - | 3 | - | - | - | - | - | - | - | 2 | - | 2 |
| CO 5 | Define platforms for development of cloud applications. | L2 | 5 | 3 | 2 | - | - | 3 | 3 | - | - | - | - | - | - | - | 2 | - | 2 |

**CO PO Mapping**

| COGNITIVE LEVEL | REVISED BLOOMS TAXONOMY KEYWORDS |
|---|---|
| L1 | List, define, tell, describe, identify, show, label, collect, examine, tabulate, quote, name, who, when, where, etc. |

| | |
|---|---|
| L2 | summarize, describe, interpret, contrast, predict, associate, distinguish, estimate, differentiate, discuss, extend |
| L3 | Apply, demonstrate, calculate, complete, illustrate, show, solve, examine, modify, relate, change, classify, experiment, discover. |
| L4 | Analyze, separate, order, explain, connect, classify, arrange, divide, compare, select, explain, infer. |
| L5 | Assess, decide, rank, grade, test, measure, recommend, convince, select, judge, explain, discriminate, support, conclude, compare, summarize. |

| PROGRAM OUTCOMES (PO), PROGRAM SPECIFIC OUTCOMES (PSO) | | | | CORRELATION LEVELS | |
|---|---|---|---|---|---|
| PO1 | Engineering knowledge | PO7 | Environment and sustainability | 0 | No Correlation |
| PO2 | Problem analysis | PO8 | Ethics | 1 | Slight/Low |
| PO3 | Design/development of solutions | PO9 | Individual and team work | 2 | Moderate/ Medium |
| PO4 | Conduct investigations of complex problems | PO10 | Communication | 3 | Substantial/ High |
| PO5 | Modern tool usage | PO11 | Project management and finance | | |
| PO6 | The Engineer and society | PO12 | Life-long learning | | |
| PSO1 | Develop applications using different stacks of web and programming technologies | | | | |
| PSO2 | Design and develop secure, parallel,  distributed, networked, and digital systems | | | | |
| PSO3 | Apply software engineering methods to design, develop, test and manage software systems. | | | | |
| PSO4 | Develop  intelligent applications for business and industry | | | | |