

Cloud Computing (315325)

Model Question Paper — Set A · Model Answers

Q.1

a) Define cloud computing (NIST).

2 M

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

Example: Google Drive, Amazon EC2, Gmail.

(Correct definition 2 M)

b) Four essential characteristics of cloud computing.

2 M

1. **On-demand self-service** — user gets server time / storage automatically without human interaction with the provider.
2. **Broad network access** — services are available over the network through standard devices (mobile, laptop, PC).
3. **Resource pooling** — provider's resources are pooled to serve many consumers (multi-tenant model).
4. **Rapid elasticity** — resources can be scaled out / in quickly as per demand.
5. **Measured service** — resource usage is metered; user pays as per use.

(Any four, ½ M each)

c) Hypervisor with examples.

2 M

Hypervisor (Virtual Machine Monitor, VMM) is the software layer that creates and runs virtual machines. It sits between the physical hardware and the guest operating systems, shares CPU, memory, storage and I/O among the VMs and keeps them isolated from each other.

Examples: VMware ESXi, Xen, Microsoft Hyper-V, KVM (Type-1); Oracle VirtualBox, VMware Workstation (Type-2).

(Definition 1 M, two examples 1 M)

d) Types of SLA with examples.

2 M

1. **Infrastructure SLA** — the provider guarantees infrastructure: e.g. *hardware / server availability 99.95 %*, power availability, data-centre network availability.
2. **Application SLA** — guarantees on the application's performance: e.g. *web page response time < 2 seconds*, transactions per second, database latency.

(Each type with example 1 M)

e) Policy and organizational risks.

2 M

1. Lock-in (difficult to move data / applications to another provider)

2. Loss of governance (control is handed to the provider)
3. Compliance challenges (provider may not meet legal / industry standards)
4. Loss of business reputation due to co-tenant activities
5. Cloud service termination or failure of the provider
6. Cloud provider acquisition / supply-chain failure

(Any four, ½ M each)

f) Examples of cloud-based ubiquitous computing for students.

2 M

Ubiquitous computing means computing available *anytime, anywhere, on any device*. Cloud makes it possible because data and applications stay on the cloud, not on one device.

- **Google Drive / Google Docs** — notes and assignments opened and edited from phone, laptop or college lab PC.
- **Google Classroom / Microsoft Teams / YouTube** — lectures and study material streamed on any device without installing or downloading.

(Any two examples, 1 M each)

g) NameNode and DataNode in HDFS.

2 M

- **NameNode** (master): stores the **metadata** — directory tree, file → block mapping and block → DataNode locations; handles open / close / rename requests and decides where replicas are placed.
- **DataNode** (slave): stores the actual **data blocks** on its local disks, serves read / write requests of clients and sends **heartbeats and block reports** to the NameNode.

(Each 1 M)

Q.2

a) SaaS model with advantages and disadvantages.

4 M

Software as a Service (SaaS) is a service model in which a complete application is hosted by the provider and delivered to users over the Internet, usually through a web browser, on a subscription / pay-per-use basis. The provider manages everything — infrastructure, OS, runtime, data storage and the application itself; the user only uses the software.

Examples: Gmail, Google Docs, Microsoft 365, Salesforce CRM, Zoom.

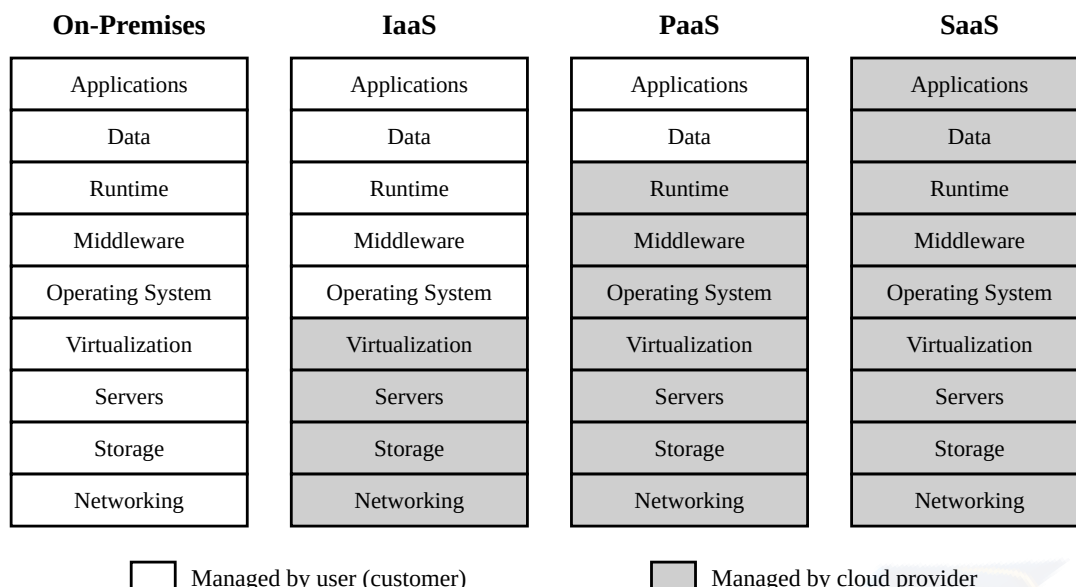


Fig.: Division of management responsibility in IaaS, PaaS and SaaS

Advantages:

1. No installation, maintenance or upgrades on user machines — the provider patches and updates.
2. Low starting cost (subscription instead of licence + hardware) and accessible from any device, anywhere.

Disadvantages:

1. Data is stored with the provider — security / privacy concerns and vendor lock-in.
2. Needs a reliable Internet connection; limited customization of the application.

(Explanation 2 M, two advantages 1 M, two disadvantages 1 M)

b) Virtualization reference model.

4 M

Virtualization creates a virtual (software-based) version of a resource such as hardware, OS, storage or network. Every virtualized environment has three major components:

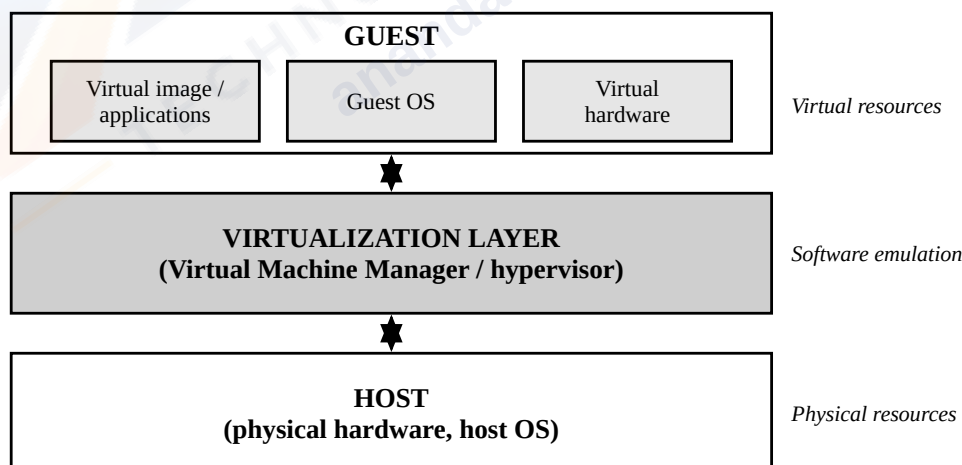


Fig.: Virtualization reference model

1. **Guest:** the system component that interacts with the virtualization layer instead of the real host — e.g. a guest OS with its applications running inside a virtual machine (virtual image).

2. **Host:** the original environment where the guest is supposed to be managed — the physical hardware (and host OS, if any).
3. **Virtualization layer:** the software (Virtual Machine Manager / hypervisor) that recreates the same or a different environment for the guest. It emulates virtual hardware, maps virtual resources to physical resources and isolates guests.

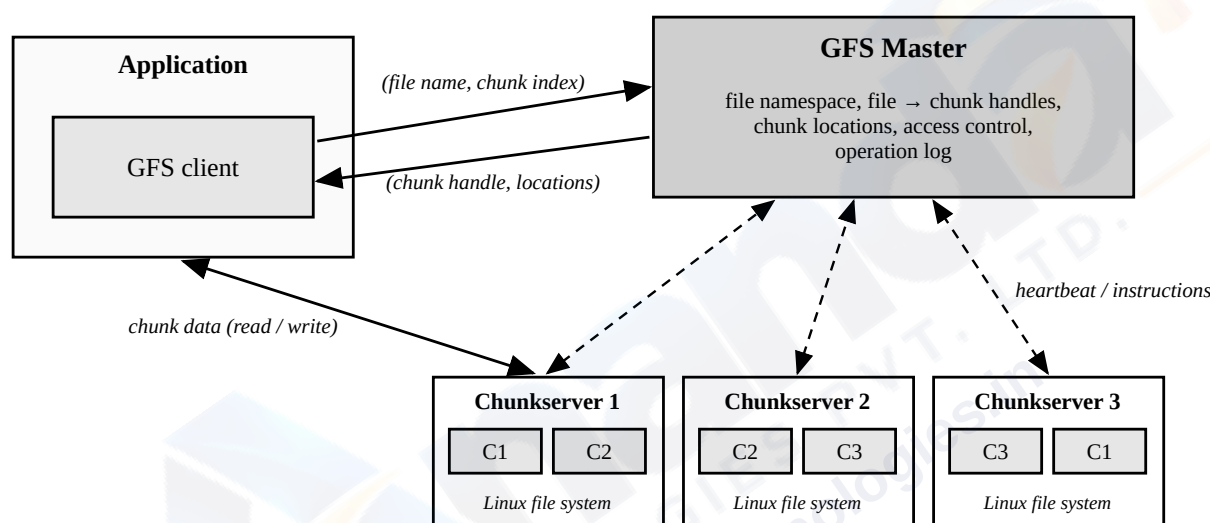
Characteristics of virtualized environment: increased security (guest is isolated in a controlled environment), managed execution — sharing, aggregation, emulation, isolation, and portability (a VM image can be moved to another host).

(Diagram 2 M, explanation of components 2 M)

c) GFS architecture.

4 M

Google File System (GFS) is a scalable distributed file system developed by Google for large, data-intensive applications running on clusters of cheap commodity machines. A GFS cluster has **one master, many chunkservers and many clients**.



Each chunk = 64 MB, stored as 3 replicas on different chunkservers

Fig.: Google File System (GFS) architecture

- **GFS Master:** keeps all metadata — file namespace, file-to-chunk mapping, locations of chunk replicas and access control; uses an operation log for recovery. It handles chunk lease management, garbage collection and re-replication. It does not carry file data.
- **Chunkservers:** files are divided into fixed-size **chunks of 64 MB**, each identified by a 64-bit chunk handle. Chunkservers store chunks as Linux files; each chunk is **replicated (default 3 copies)** on different chunkservers.
- **GFS client:** library linked to the application. It asks the master *which* chunkserver has the chunk, then reads / writes data **directly** with the chunkserver, so the master does not become a bottleneck.
- **Heartbeat:** the master periodically exchanges heartbeat messages with chunkservers to give instructions and collect their state.

(Diagram 2 M, explanation 2 M)

d) Identification, authentication, authorization and auditing in IAM.

4 M

Identity and Access Management (IAM) is the framework of policies and technologies that ensures that the **right users get the right access to the right cloud resources** at the right time.

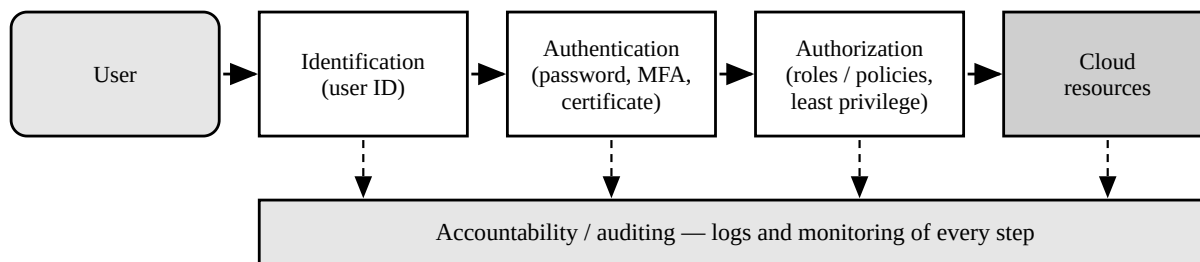


Fig.: Identity and access management (IAM) flow

- 1. Identification:** the user claims an identity, e.g. user name, e-mail ID or account ID. Each user / service gets a unique digital identity.
- 2. Authentication:** the claimed identity is verified — password, OTP, multi-factor authentication (MFA), digital certificate, biometrics.
- 3. Authorization:** after authentication the system decides *what* the user may do — permissions given through roles and policies (RBAC), following the principle of least privilege.
- 4. Auditing / accountability:** all access and actions are logged and monitored (e.g. AWS CloudTrail) so that misuse can be detected and compliance can be proved.

(Each term 1 M)

Q.3

a) Type-1 vs Type-2 hypervisor.

4 M

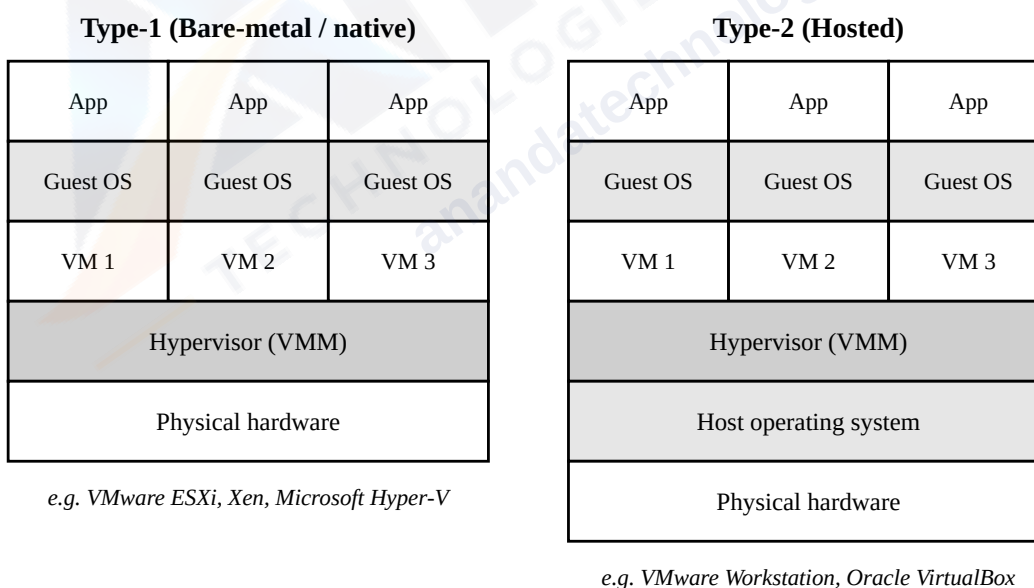


Fig.: Type-1 (bare-metal) and Type-2 (hosted) hypervisor

Point	Type-1 hypervisor (bare metal)	Type-2 hypervisor (hosted)
Installation	Installed directly on physical hardware.	Installed as an application on a host operating system.

Host OS	No host OS needed.	Needs a host OS (Windows / Linux / macOS).
Performance	Better performance, less overhead — direct hardware access.	Lower performance — every request passes through the host OS.
Security / reliability	More secure and reliable; failure of one OS does not affect others.	Less secure; a crash or attack on the host OS affects all VMs.
Used for	Data centres, servers, cloud (IaaS).	Desktops, testing, learning, development.
Examples	VMware ESXi, Xen, Microsoft Hyper-V, KVM.	Oracle VirtualBox, VMware Workstation, Parallels Desktop.

(Any four points, 1 M each)

b) Five phases of SLA life cycle.

4 M

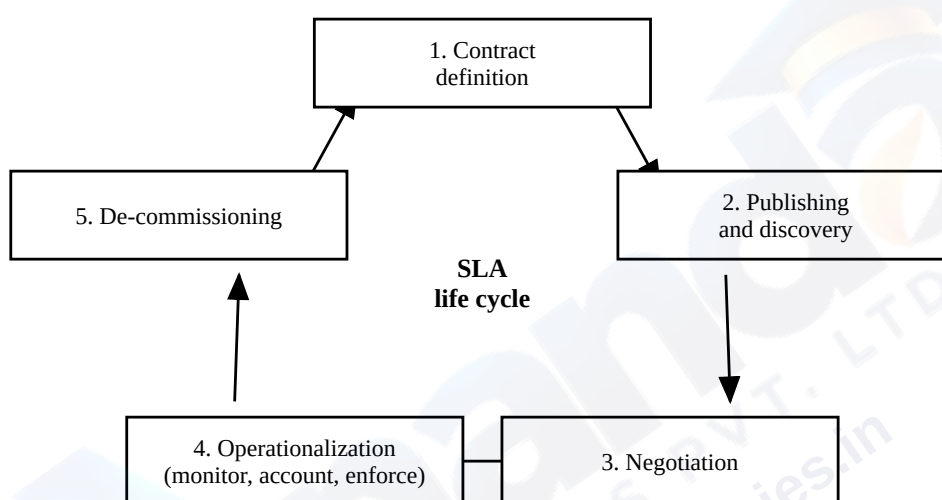


Fig.: SLA life cycle (five phases)

- 1. Contract definition:** the provider defines standard SLA templates for its service offerings (service, QoS parameters, price, penalties).
- 2. Publishing and discovery:** the provider publishes its offerings (with SLA templates) in a catalogue; consumers search and discover suitable providers.
- 3. Negotiation:** consumer and provider negotiate the terms (QoS targets, price, penalties) and the final SLA is signed — a customised SLA.
- 4. Operationalization:** the SLA is put into operation — *SLA monitoring* (measure QoS), *SLA accounting* (record compliance / violations for billing) and *SLA enforcement* (take corrective action / apply penalties).
- 5. De-commissioning:** the relationship ends; all activities under the SLA are terminated, resources released and final settlement done.

(Five phases with explanation 4 M)

c) ZigBee device types and topologies.

ZigBee is a low-power, low-data-rate, short-range wireless standard built on **IEEE 802.15.4** (2.4 GHz worldwide; 868 / 915 MHz regional; up to 250 kbps; about 10–100 m). It is widely used in IoT — home automation, smart meters, sensors.

Device types:

1. **ZigBee Coordinator (ZC):** one per network; starts the network, selects channel and PAN ID, stores security keys; acts as bridge to other networks.
2. **ZigBee Router (ZR):** relays data from other devices, extends the range, allows end devices to join.
3. **ZigBee End Device (ZED):** simple, battery-powered sensor / actuator; talks only to its parent; sleeps most of the time to save power.

Topologies: **Star** — all end devices talk to the coordinator; **Tree** — coordinator at root, routers form branches; **Mesh** — routers interconnect so data can take alternate paths (self-healing).

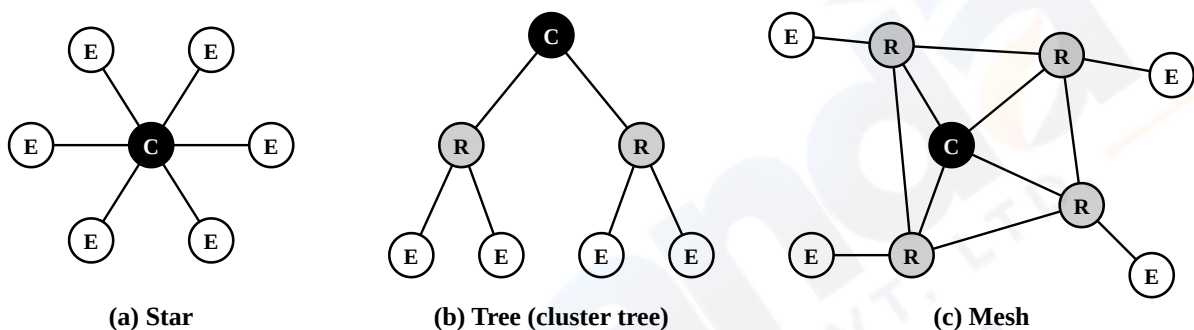


Fig.: ZigBee network topologies (C = coordinator, R = router, E = end device)

(Device types 2 M, topologies with diagram 2 M)

d) Cloud storage system architecture.

Cloud storage keeps data on remote, provider-managed servers and makes it available over the Internet as a service (e.g. Amazon S3, Google Drive). Its architecture is layered:

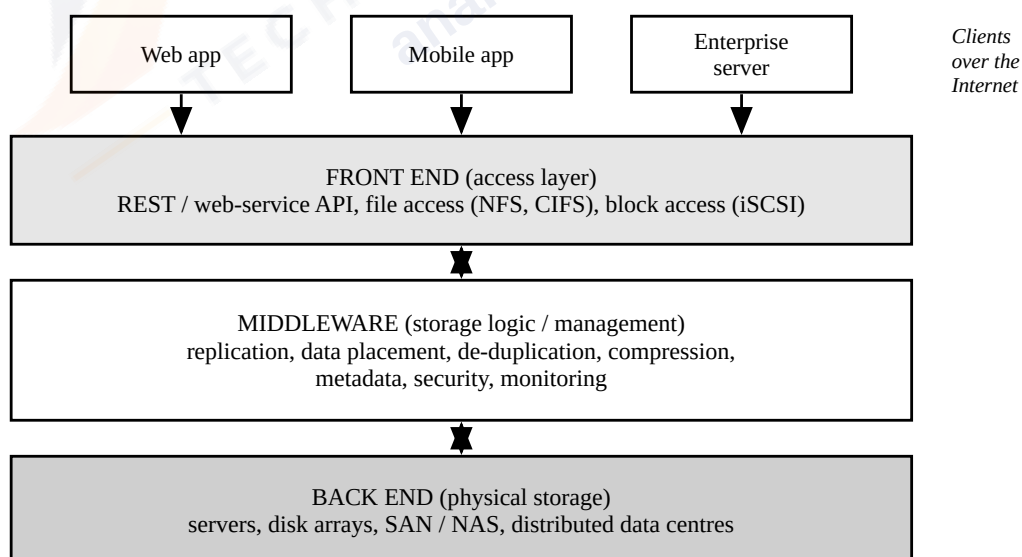


Fig.: Generic cloud storage system architecture

1. **Front end (access layer):** exposes the storage to clients through APIs — REST / web-service APIs for object storage, file protocols (NFS, CIFS) and block protocols (iSCSI).
2. **Middleware (storage logic):** implements storage features — replication for reliability, data placement and load balancing, de-duplication and compression to reduce space, metadata management, security and monitoring.
3. **Back end (physical storage):** actual disks, storage arrays, SAN / NAS spread over geographically distributed data centres.
4. Clients reach the front end over the Internet; the storage is elastic, pay-per-use and highly available because data is replicated in multiple locations.

(Diagram 2 M, explanation 2 M)

Q.4

a) Live VM migration steps.

4 M

Live migration moves a *running* VM from one physical host to another with almost zero downtime; users stay connected. It is done with the **pre-copy** approach in the following stages:

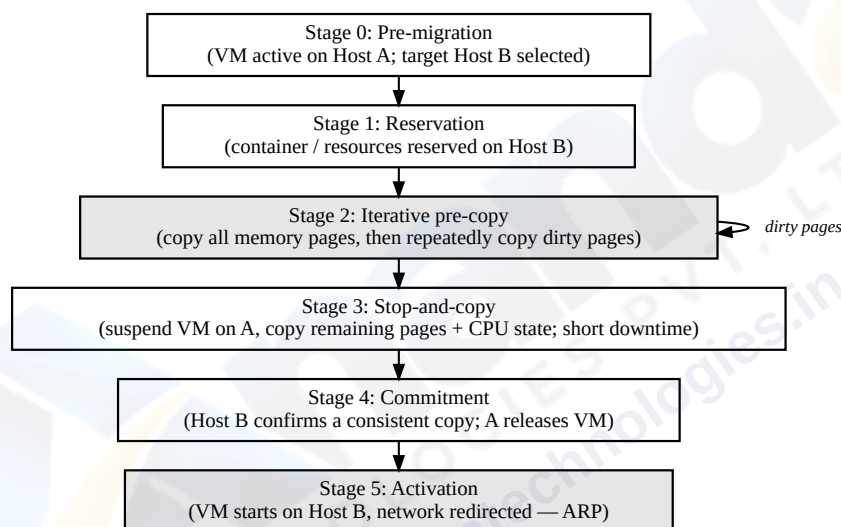


Fig.: Live migration of a VM from Host A to Host B (pre-copy)

1. **Stage 0 – Pre-migration:** VM is active on host A; a target host B with enough resources is selected (shared storage, e.g. SAN, is used so disks need not be copied).
2. **Stage 1 – Reservation:** resources (a VM container) are reserved on host B.
3. **Stage 2 – Iterative pre-copy:** all memory pages are copied to B while the VM keeps running; pages modified (dirtied) during copying are sent again in further rounds.
4. **Stage 3 – Stop-and-copy:** VM on A is suspended briefly; remaining dirty pages and CPU state are copied — this is the only downtime (milliseconds).
5. **Stage 4 – Commitment:** B confirms that it has a consistent image; A can release the VM.
6. **Stage 5 – Activation:** VM starts on B; network traffic is redirected (ARP), device drivers re-attached.

(Diagram 2 M, steps 2 M)

b) Technical risks in cloud.

4 M

1. **Resource exhaustion (under / over-provisioning):** provider may not have enough resources at peak time, so service becomes unavailable.
2. **Isolation failure:** failure of the mechanisms that separate storage, memory and routing between tenants (e.g. hypervisor attacks) — one tenant may access another's data.
3. **Malicious insider:** a dishonest employee of the provider can misuse privileged access.
4. **Management interface compromise:** the customer web portal / APIs are reachable over the Internet; if hacked, all resources can be controlled.
5. **Interception of data in transit / data leakage:** data travelling between user and cloud may be sniffed if not encrypted.
6. **Insecure or incomplete data deletion:** deleted data may still remain on disks reused by others.
7. **Distributed denial of service (DDoS)** and loss of encryption keys.

(Any four risks with explanation, 1 M each)

c) Google App Engine features and architecture.

4 M

Google App Engine (GAE) is Google's **Platform as a Service (PaaS)** for developing and hosting web applications in Google-managed data centres. The developer only writes and deploys code; Google manages servers, scaling and patching.

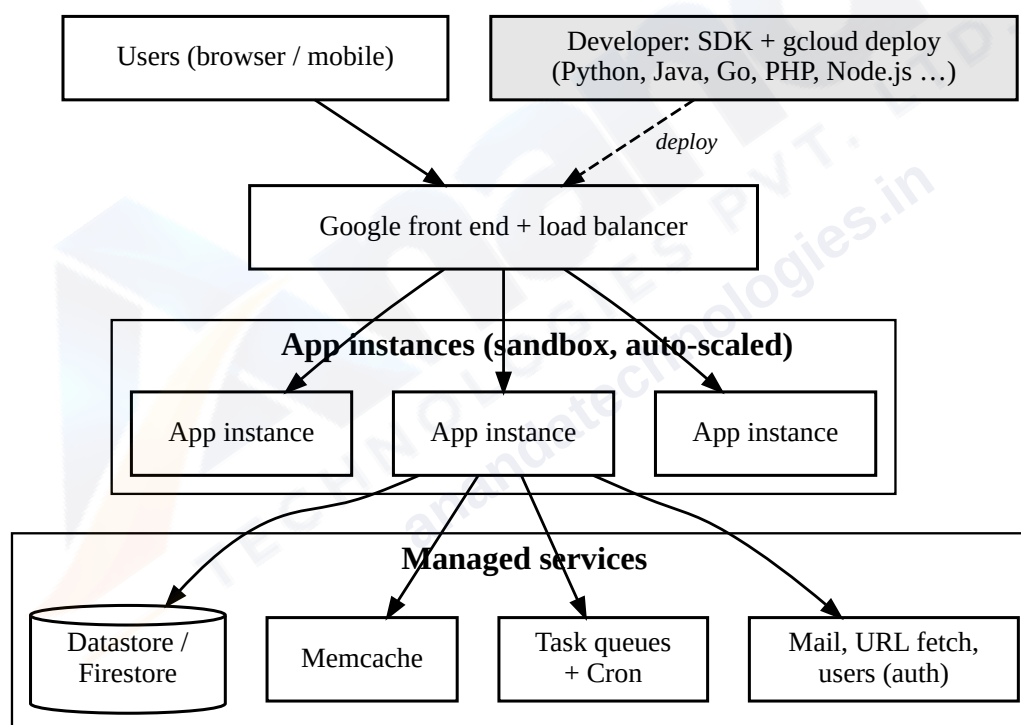


Fig.: Google App Engine (PaaS) architecture

- **Automatic scaling and load balancing** — instances are added or removed according to traffic.
- **Multiple language runtimes** — Python, Java, Go, PHP, Node.js, Ruby, .NET (standard and flexible environments).
- **Sandbox** — each app runs in a secure, isolated environment.
- **Managed services** — Datastore / Firestore (NoSQL), Memcache, task queues, cron jobs, user authentication, logging.

- **Pay-per-use with free quota;** versioning and traffic splitting for safe updates; SDK for local development.

(Architecture / explanation 2 M, any four features 2 M)

d) Architectural components of cloud computing.

4 M

Cloud computing architecture is divided into two main parts connected through a network (Internet):

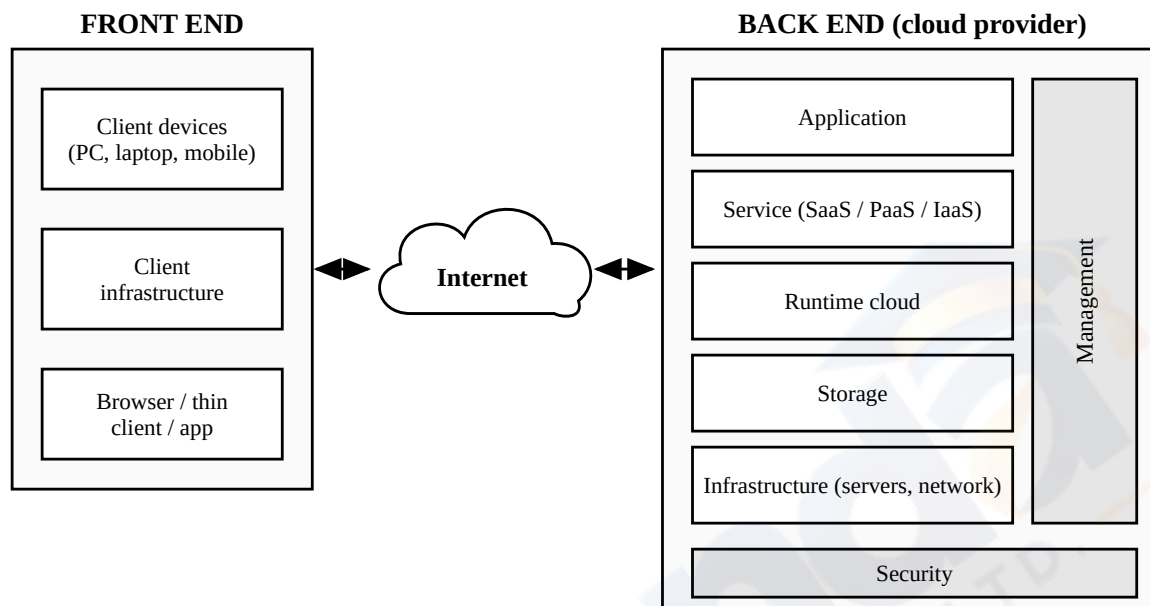


Fig.: Architecture of cloud computing (front end and back end)

1. **Front end:** the client side used by the user — client devices (PC, laptop, mobile), client infrastructure and interfaces such as a web browser, thin client or mobile app used to access cloud services.
2. **Back end:** the provider side — *application* (software offered), *service* (SaaS / PaaS / IaaS), *runtime cloud* (execution environment for VMs), *storage*, and *infrastructure* (servers, network, virtualization software).
3. **Management and security:** back-end components that allocate resources, monitor usage and protect the cloud (authentication, firewalls, encryption).
4. **Network / Internet:** connects front end and back end.

NIST describes the same structure in terms of actors — cloud consumer, provider, broker, auditor and carrier:

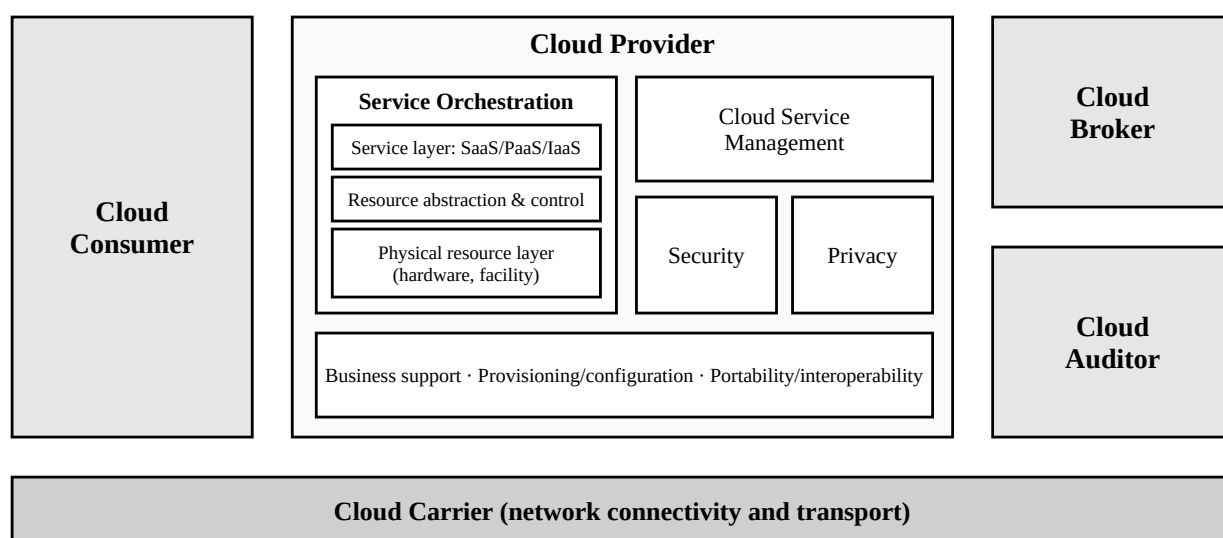


Fig.: NIST cloud computing reference architecture (actors)

(Diagram 2 M, explanation 2 M)

e) ECG analysis in the cloud and its access.

4 M

ECG (electrocardiogram) monitoring is an important **cloud-based healthcare IoT application**. Instead of keeping the patient in hospital, the heartbeat is monitored remotely and analysed in the cloud.

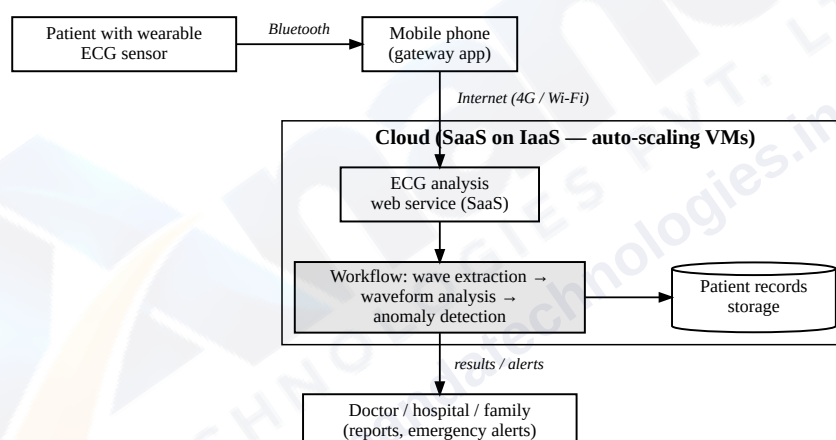


Fig.: ECG data analysis using the cloud

- 1. Data capture:** the patient wears an ECG sensor which records the heart's electrical signal continuously.
- 2. Transfer:** the sensor sends data via Bluetooth to the patient's mobile phone, which forwards it over the Internet (4G / Wi-Fi) to the cloud web service.
- 3. Analysis in the cloud:** an ECG analysis service (SaaS) running on IaaS VMs executes a workflow — extract waveform, analyse waves (heart rate, QRS complex), compare with normal patterns and detect anomalies. VMs scale automatically as the number of patients grows.
- 4. Access:** results are stored in cloud storage; doctors, hospitals and family members access reports from any browser / mobile app; in an emergency an alert is sent immediately.

Benefits: remote 24×7 monitoring, no costly hardware at hospital, elastic computing, pay-per-use.

(Diagram 2 M, explanation 2 M)

Q.5

a) Deployment models with advantage / disadvantage.

6 M

A deployment model defines **who owns, manages and can use** the cloud infrastructure.

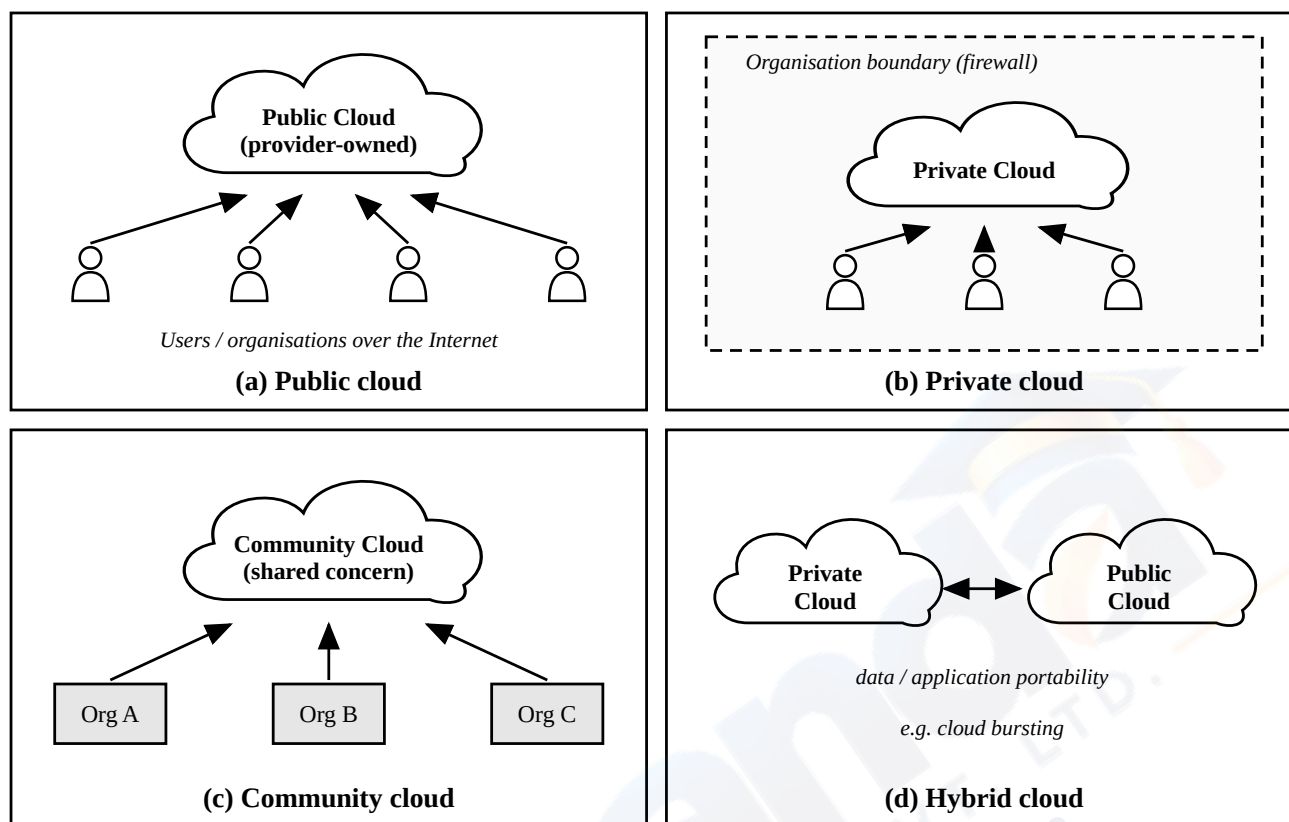


Fig.: Cloud deployment models

- 1. Public cloud:** owned and operated by a third-party provider and offered to the general public over the Internet on pay-per-use basis (AWS, Microsoft Azure, Google Cloud).
Advantage: no capital cost, highly scalable. *Disadvantage:* less control and data-privacy concerns.
- 2. Private cloud:** used by a single organisation; may be on-premises or hosted by a third party (e.g. OpenStack / VMware in a company data centre).
Advantage: high security and control, can be customised. *Disadvantage:* costly to set up and maintain; limited scalability.
- 3. Community cloud:** shared by several organisations with common concerns (security, compliance, mission) — e.g. government departments, banks, universities.
Advantage: cost shared among members, meets common compliance. *Disadvantage:* limited capacity; data-sharing and governance issues between members.
- 4. Hybrid cloud:** combination of two or more clouds (private + public) bound by technology that enables data and application portability — e.g. sensitive data on private cloud, extra load sent to public cloud (cloud bursting).
Advantage: flexibility — security of private with scalability of public. *Disadvantage:* complex to manage and integrate; security across both clouds.

(Diagram 2 M, each model with advantage and disadvantage 1 M)

b) Xen architecture and guest OS management.

6 M

Xen is an open-source **Type-1 (bare-metal) hypervisor** originally from the University of Cambridge. It is based on **paravirtualization**: guest OS kernels are modified to call the hypervisor through **hypercalls** instead of executing privileged instructions. (With hardware support — Intel VT / AMD-V — Xen can also run unmodified guests.) Citrix XenServer and early Amazon EC2 used Xen.

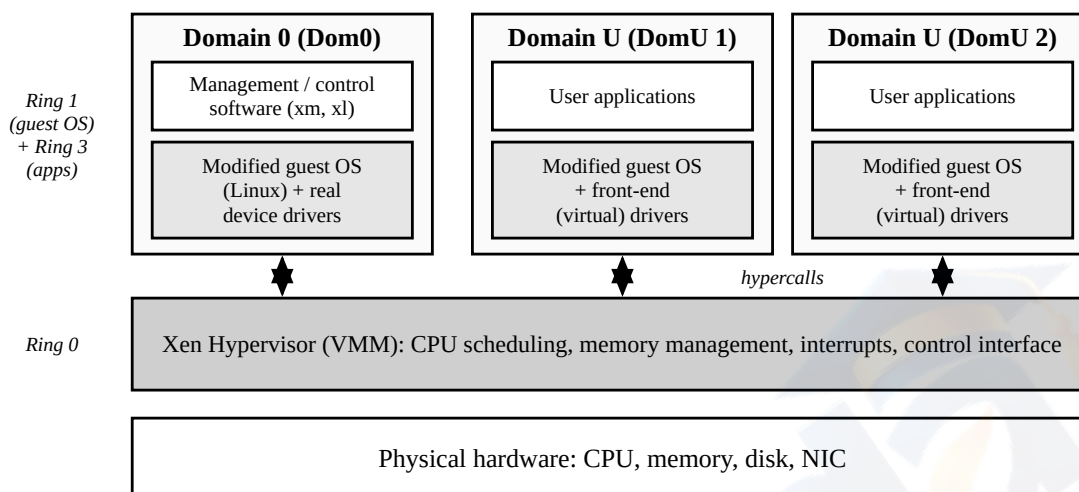


Fig.: Xen architecture (paravirtualization)

Components:

1. **Xen hypervisor:** runs directly on hardware in the most privileged mode (Ring 0); handles CPU scheduling, memory partitioning and interrupts. It is kept small and does not contain device drivers.
2. **Domain 0 (Dom0):** the first, privileged VM started at boot. It runs a modified Linux kernel with the real device drivers and the management tools; it can access hardware directly.
3. **Domain U (DomU):** unprivileged guest VMs. Their modified kernels run in Ring 1 and applications in Ring 3; they access devices through split (front-end / back-end) drivers via Dom0.

Guest OS management:

- All guest management is done from Dom0 using the **control interface** and tools (xm / xl, XenCenter): create, start, pause, resume, shut down and destroy DomU guests.
- Dom0 allocates CPU and memory to each guest, configures virtual network and disks and can **live-migrate** a guest to another Xen host.
- Guests request privileged operations through hypercalls; the hypervisor validates them, which keeps guests isolated and secure.

(Diagram 2 M, components 2 M, guest OS management 2 M)

c) Multi-tenancy and trust management.

6 M

1) Multi-tenancy

Multi-tenancy means that a **single instance of software / infrastructure serves many customers (tenants)**. Tenants share the same application, database and servers, but each tenant's data and configuration is logically isolated.

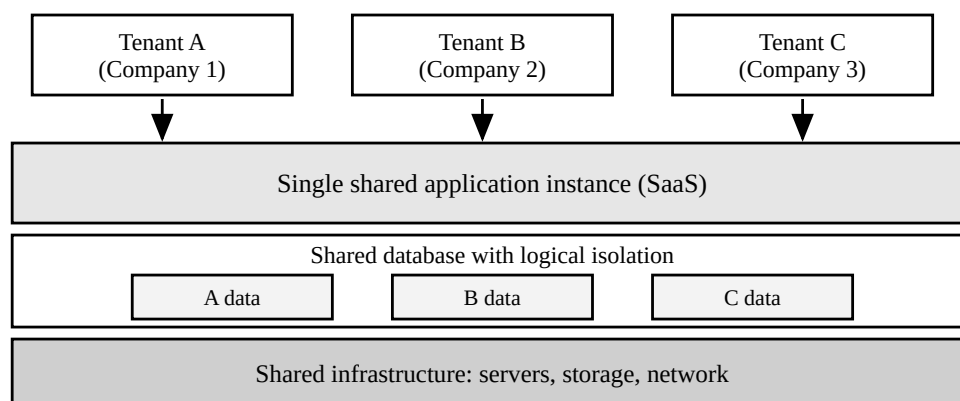


Fig.: Multi-tenancy — many tenants share one application instance / infrastructure

- Benefits: better resource utilisation, lower cost per customer, easy upgrades (one instance).
- Security issues: data leakage between tenants if isolation fails, side-channel attacks between co-located VMs, one tenant's heavy load affecting others ('noisy neighbour').
- Controls: strong logical isolation (tenant ID in every query, separate schema), encryption per tenant, hypervisor hardening, access control and monitoring.

2) Trust management

In cloud the user hands data and processing to a third party, so the user must **trust** the provider. Trust management is the process of establishing, evaluating and maintaining this trust between cloud users and providers (and among federated providers).

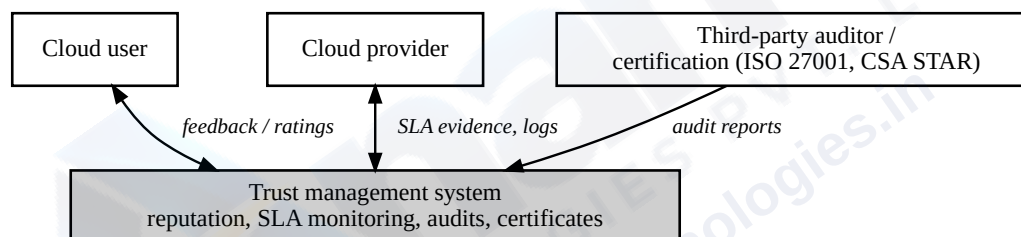


Fig.: Trust management between cloud user and provider

- **SLA-based trust:** clear SLA with QoS guarantees and penalties; SLA monitoring proves compliance.
- **Reputation / feedback:** ratings and feedback from other users help select reliable providers.
- **Certification and audits:** third-party audits and certificates (ISO 27001, SOC 2, CSA STAR) give evidence of security practices.
- **Transparency:** logs, data-location information and incident reports provided to customers.
- Role: trust management reduces risk, helps the user choose a provider and makes the provider accountable.

(Multi-tenancy 3 M, trust management 3 M)

Q.6

a) VDC architecture and benefits of server virtualization.

6 M

A **Virtualized Data Centre (VDC)** is a data centre in which compute, storage, network, desktop and application resources are virtualized and pooled, so that they can be provisioned on demand —

it is the foundation of a cloud.

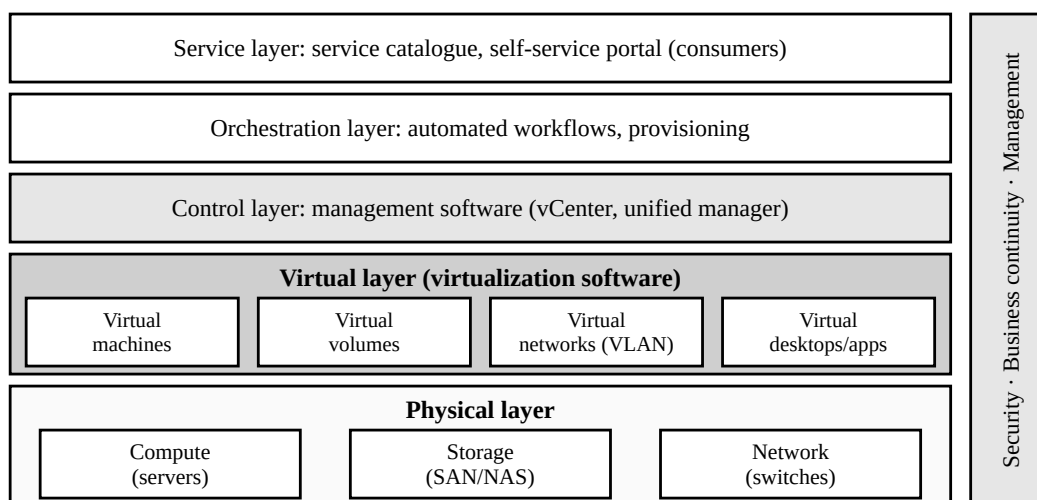


Fig.: Virtualized Data Centre (VDC) architecture

1. **Physical layer:** compute (servers), storage (SAN / NAS) and network (switches, routers).
2. **Virtual layer:** virtualization software abstracts physical resources into virtual machines, virtual volumes and virtual networks (VLAN, virtual switches).
3. **Control layer:** management software (e.g. VMware vCenter) that configures, allocates and monitors resources.
4. **Orchestration and service layers:** automate provisioning and offer services through a catalogue / self-service portal; security, business continuity and management span all layers.

Benefits of server virtualization in VDC:

- **Server consolidation** — many VMs on one physical server; fewer servers, better utilisation.
- **Reduced cost** — less hardware, power, cooling and floor space.
- **Rapid provisioning** — new server (VM) from a template in minutes.
- **High availability** — live migration and failover of VMs without downtime.
- **Isolation** — fault in one VM does not affect others; easy backup / snapshots.

(Architecture with diagram 3 M, benefits 3 M)

b) Data security technologies and data security risks.

6 M

Data security technologies:

1. **Encryption:** data encrypted *at rest* (AES-256 on disks / storage) and *in transit* (TLS / HTTPS, VPN); server-side or client-side.
2. **Key management:** secure generation, storage and rotation of keys using key management services / HSMs.
3. **Access control and IAM:** authentication, MFA and role-based authorization so only authorised users can read data.
4. **Data masking / tokenization and anonymisation:** sensitive fields hidden or replaced.
5. **Data integrity checking:** hashing (SHA-256), digital signatures, provable data possession.
6. **Data Loss Prevention (DLP), backup and replication:** stop leakage and recover lost data.

Data security risks:

- Data breach / unauthorised access
- Data loss (deletion, disaster, provider failure)
- Data leakage between tenants
- Insecure or incomplete data deletion
- Unknown data location and jurisdiction problems
- Malicious insiders and insecure APIs

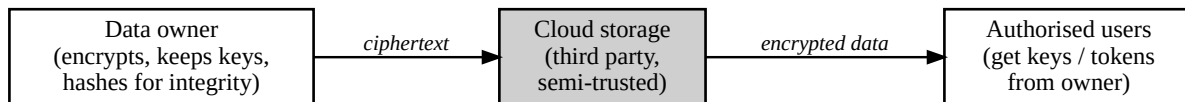


Fig.: Data outsourcing — data protected before it leaves the owner

(Technologies 4 M, risks 2 M)

c) Benefits of Amazon EC2 and S3.

6 M

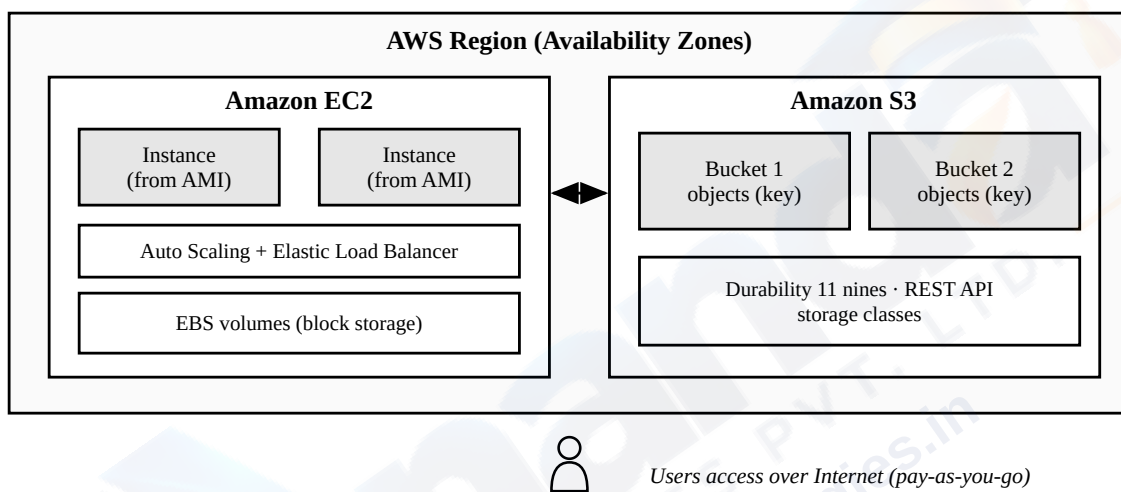


Fig.: Amazon EC2 and S3 working together

Amazon EC2 (Elastic Compute Cloud) — IaaS that provides resizable virtual servers (instances) created from Amazon Machine Images (AMIs).

1. **Elastic** — instances can be started or stopped in minutes; Auto Scaling adds / removes instances with load.
2. **Pay-as-you-go** — on-demand, reserved and spot pricing; no hardware investment.
3. **Choice** — many instance types (CPU, memory, GPU optimised) and OS (Linux, Windows).
4. **Reliable and secure** — multiple availability zones, security groups (firewall), key pairs, Elastic Load Balancing and EBS block storage.

Amazon S3 (Simple Storage Service) — object storage service; data stored as objects in buckets, accessed by key through REST API / console.

1. **Highly durable** — designed for 99.999999999 % (11 nines) durability by storing copies across multiple facilities.
2. **Unlimited and scalable storage**, pay only for storage used and requests made.
3. **Storage classes** (Standard, Infrequent Access, Glacier) to reduce cost; lifecycle rules.

4. **Security** — encryption, bucket policies, IAM; used for backup, static websites, data lakes; integrates with EC2.

(EC2 benefits 3 M, S3 benefits 3 M)

Note: This is a **model question paper prepared by Ananda Technologies** only for practice and exam preparation, based on the pattern of previous MSBTE question papers and the K-scheme syllabus. It is **not an official MSBTE paper**, and it does **not** mean that these questions will come in the exam. Answers follow standard textbook points; marks shown are indicative. Free for students — anandatechnologies.in

