

Cloud Computing (315325)

Model Question Paper — Set C · Model Answers

Q.1

a) Private cloud with examples.

2 M

Private cloud: a cloud infrastructure operated **solely for a single organisation**. It may be managed by the organisation or a third party and may be located on-premises or off-premises. It gives cloud features (self-service, elasticity, pooling) with high security and control.

Examples: a company data centre running OpenStack or VMware vSphere / vCloud; Microsoft Azure Stack; a bank's internal cloud; ISRO / NIC internal clouds.

(Definition 1 M, two examples 1 M)

b) Virtualization and its types.

2 M

Virtualization is the technique of creating a **virtual (software-based) version** of a computing resource — server, OS, storage, network — so that one physical resource can act as many logical resources, or many physical resources as one.

Types: 1) Server / hardware virtualization 2) Storage virtualization 3) Network virtualization 4) Desktop virtualization 5) Application virtualization.

(Definition 1 M, any four types 1 M)

c) GFS vs HDFS.

2 M

Point	GFS	HDFS
Owner / source	Proprietary file system of Google (C++).	Open source, part of Apache Hadoop (Java).
Master / slaves	Master and chunkservers.	NameNode and DataNodes.
Unit size	Chunk of 64 MB.	Block of 128 MB (default, Hadoop 2+; 64 MB in Hadoop 1).
Writes	Supports multiple concurrent writers (record append).	Single writer, append-only (write-once-read-many).

(Any two points, 1 M each)

d) Phases of cloud service life cycle.

2 M

1. Service planning
2. Service creation
3. Service operation
4. Service termination

(½ M each)

e) Multi-tenancy.

2 M

Multi-tenancy is an architecture in which a **single instance of an application / infrastructure serves multiple customers (tenants)**. Tenants share the same software, database and hardware, but each tenant's data and configuration are logically isolated and invisible to others.

Example: Salesforce or Gmail — thousands of companies use the same application instance.

(Definition 2 M)

f) Technical risks.

2 M

1. Resource exhaustion (under / over-provisioning)
2. Isolation failure between tenants
3. Malicious insider at the provider
4. Management interface compromise
5. Interception of data in transit / data leakage
6. Insecure or ineffective deletion of data
7. Distributed denial of service (DDoS)
8. Loss of encryption keys

(Any four, ½ M each)

g) Ubiquitous computing and supporting trends.

2 M

Ubiquitous computing (pervasive computing) is the concept of computing being available **everywhere and anytime**, on any device, embedded in everyday objects, so that users get services without thinking about the machine.

Cloud trends supporting it: mobile cloud computing / SaaS accessed from browsers; cloud storage with sync (Google Drive); IoT with cloud back-end; edge / fog computing.

(Definition 1 M, two trends 1 M)

Q.2

a) Public vs private vs hybrid cloud.

4 M

Point	Public cloud	Private cloud	Hybrid cloud
Ownership	Third-party provider	Single organisation (or hosted for it)	Combination of private and public
Users	General public / many organisations	Only one organisation	Organisation; uses public for extra load
Security / control	Lower control; shared infrastructure	Highest security and control	Sensitive data kept private; good control
Cost	No capital cost; pay per use	High capital and maintenance cost	Moderate
Scalability	Very high (practically unlimited)	Limited to own hardware	High (cloud bursting to public)

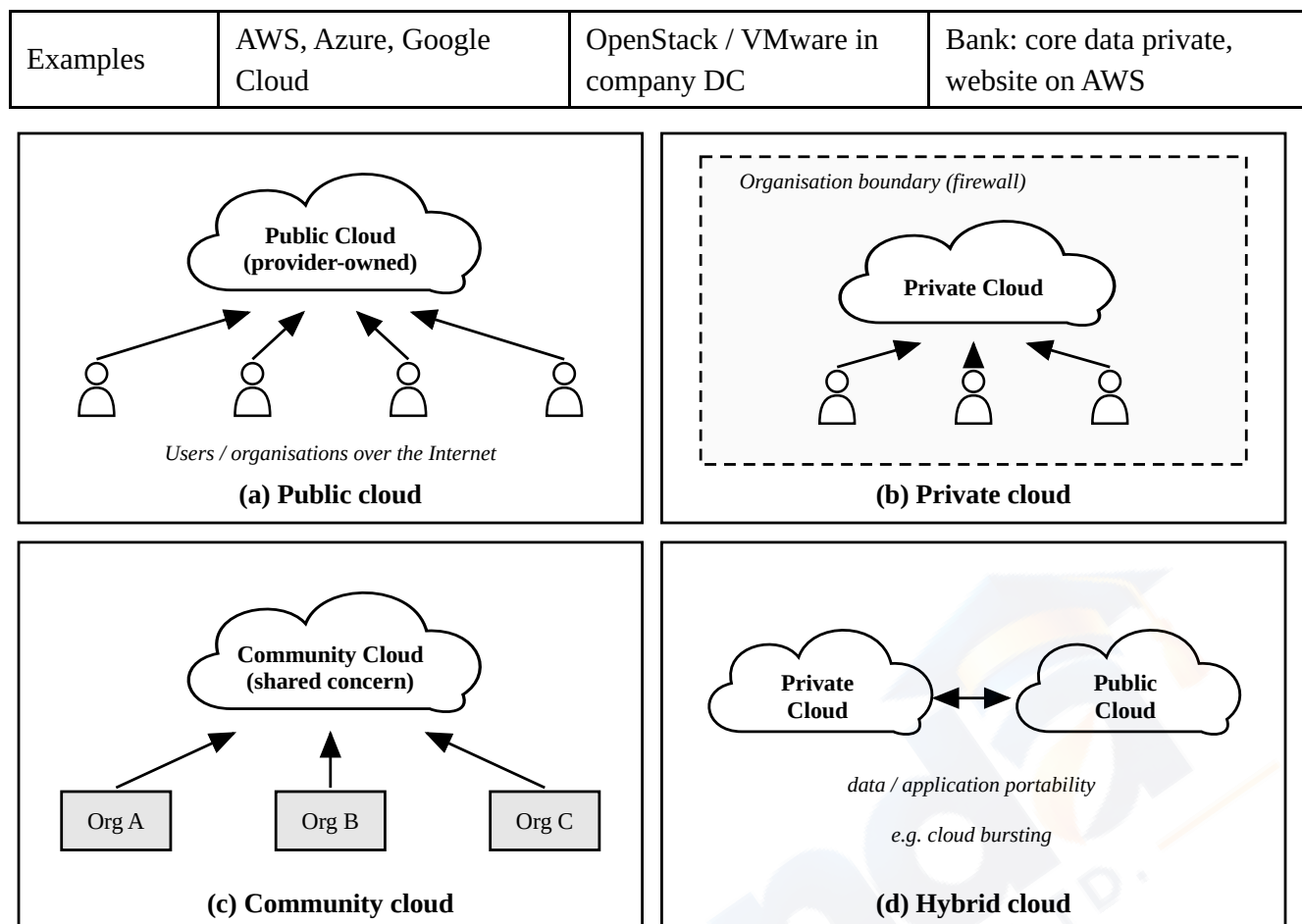


Fig.: Cloud deployment models

(Any four points, 1 M each)

b) VM migration and VM management.

4 M

VM migration is the process of moving a virtual machine (its memory, CPU state, and if needed its storage) from one physical host to another. It is used for load balancing, server maintenance, fault tolerance and power saving (consolidation).

- **Cold (offline) migration:** VM is shut down, its files are moved, then it is restarted on the new host — downtime occurs.
- **Live (hot) migration:** VM keeps running while memory pages are copied iteratively (pre-copy); only a few milliseconds of downtime — e.g. VMware vMotion, Xen / KVM live migration.

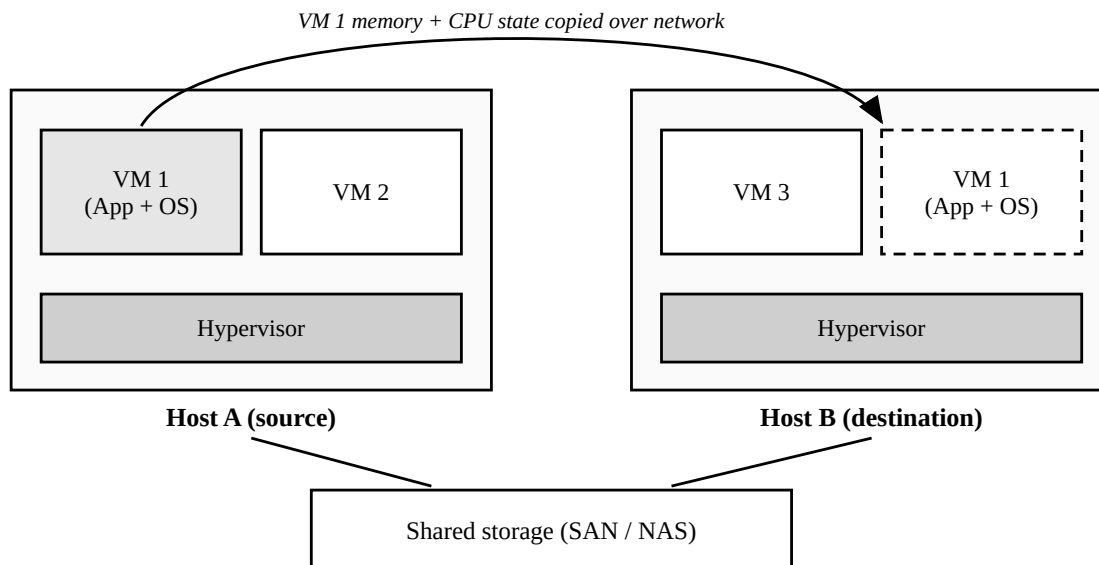


Fig.: VM migration between physical hosts sharing storage

VM management is the set of activities to control VMs over their life cycle, done with tools such as VMware vCenter, XenCenter, OpenStack Nova, virt-manager:

- Provisioning — create VMs from templates, allocate CPU, RAM, disk, network.
- Monitoring — performance, resource usage, health; alerts.
- Placement, load balancing, migration and consolidation.
- Snapshots, backup and recovery; patching and updates.
- Security and access control; de-provisioning when not needed.

(VM migration 2 M, VM management 2 M)

c) VDC components; desktop and application virtualization.

4 M

Components of a VDC environment: **compute** (physical servers with hypervisors / VMs), **storage** (SAN / NAS, virtual volumes), **network** (switches, routers, VLANs, virtual switches), plus **applications, desktops and management software**.

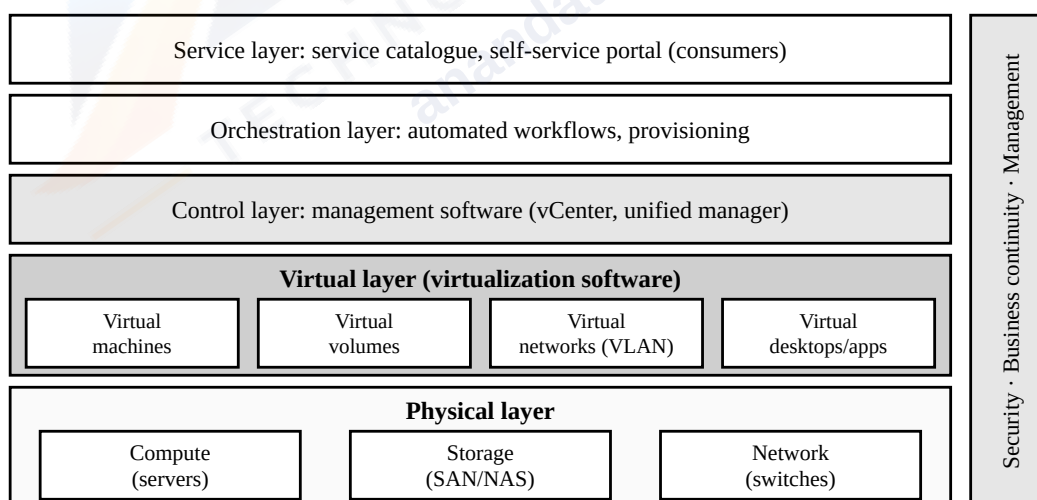


Fig.: Virtualized Data Centre (VDC) architecture

Benefits of desktop virtualization (VDI):

- Desktops run on central servers — access from any device / location.

- Central management, patching and backup; data does not leave the data centre (more secure).
- Cheap thin clients can be used; quick provisioning of new desktops.

Benefits of application virtualization:

- Applications run without installation, isolated from the OS — no conflicts between versions.
- Easy, central deployment and update; faster roll-out to many users; simplified licence tracking.

(Components 1 M, desktop virtualization 1½ M, application virtualization 1½ M)

d) Recommended IAM feature: Single Sign-On.

4 M

Recommended feature: Single Sign-On (SSO) through a central identity provider, preferably with multi-factor authentication (MFA). With SSO an employee logs in **once** with one set of credentials and gets access to all permitted SaaS applications.

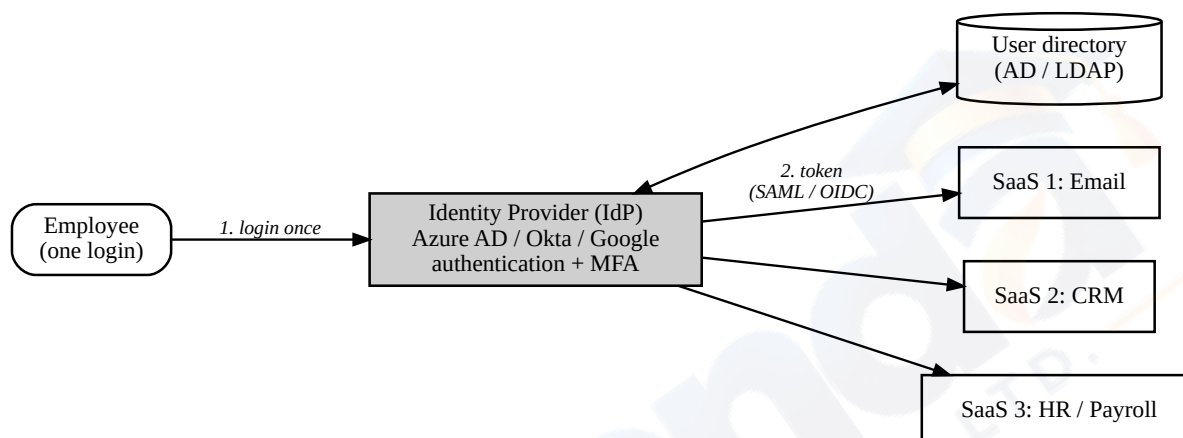


Fig.: Single Sign-On (SSO) using a central identity provider

1. The employee opens a SaaS application (e.g. CRM); it redirects the user to the company's **Identity Provider (IdP)** — e.g. Azure AD / Microsoft Entra ID, Okta, Google Workspace.
2. The IdP authenticates the user once (password + OTP / MFA) against the company directory.
3. The IdP sends a signed token / assertion (**SAML** or **OpenID Connect** / **OAuth 2.0**) to the application; the app trusts the IdP and logs the user in.
4. For other apps (e-mail, payroll) the existing IdP session is reused — no new password.

Why: fewer passwords (less reuse / weak passwords), central control — disabling one account removes access to all apps, better audit logs, less help-desk load.

(Correct feature 1 M, working with diagram 3 M)

Q.3

a) Characteristics of virtualized environment.

4 M

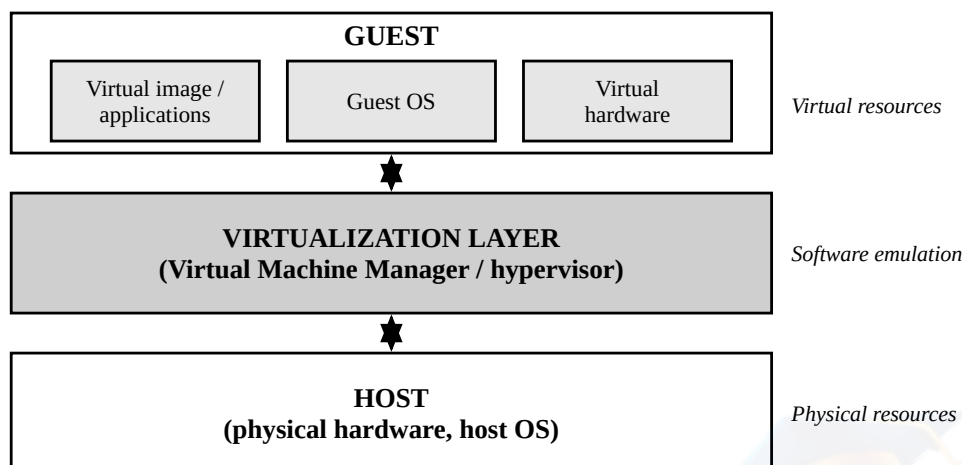


Fig.: Virtualization reference model

- 1. Increased security:** the virtualization layer controls all operations of the guest; the guest runs in an isolated, sandboxed environment, so harmful operations cannot reach the host or other guests. Sensitive data of the host can be hidden.
- 2. Managed execution:** the virtualization layer manages how guests use resources —
 - Sharing:* many guests share one physical host, improving utilisation.
 - Aggregation:* a group of separate hosts appears as a single virtual host (e.g. cluster).
 - Emulation:* an environment different from the host can be emulated (e.g. different CPU / device).
 - Isolation:* each guest gets a separate environment; failure of one guest does not affect others.
- 3. Performance tuning:** resources given to a guest (memory, CPU share, bandwidth) can be finely controlled to meet QoS / SLA.
- 4. Portability:** a guest packaged as a virtual image can be moved and executed on different hosts / hypervisors; users can carry their environment with them.

(Any four characteristics, 1 M each)

b) Cloud service life cycle.

4 M

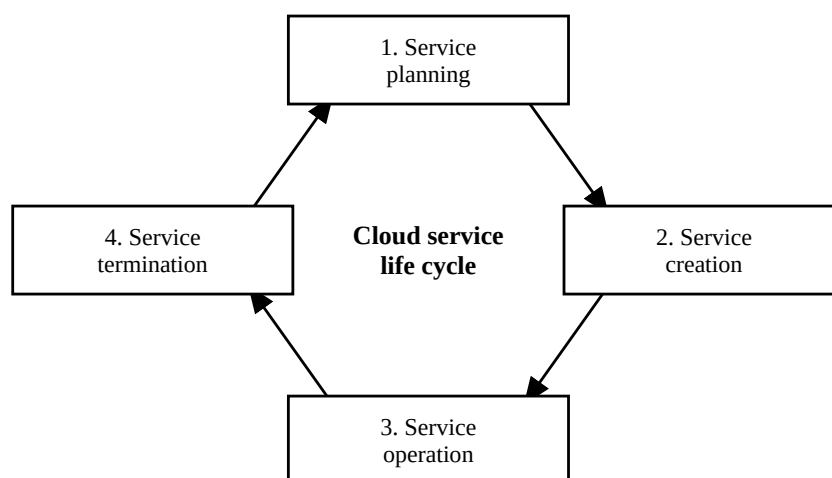


Fig.: Cloud service life cycle

1. **Service planning:** identify business requirements, define the service offering (what, for whom, SLA, price), check feasibility and plan resources.
2. **Service creation:** design and build the service — create templates / service catalogue entries, orchestrate infrastructure, configure security, test and publish the service to consumers.
3. **Service operation:** the service is delivered — provisioning on request, monitoring, metering and billing, SLA management, capacity and incident management, updates.
4. **Service termination:** service (or a subscription) is withdrawn — data returned / securely deleted, resources de-provisioned and released, final billing and closure.

(Diagram 1 M, four phases 3 M)

c) Wireless sensor networks and GPS for IoT.

4 M

1) Wireless Sensor Network (WSN)

A WSN is a network of many small, battery-powered **sensor nodes** spread over an area that sense physical conditions (temperature, humidity, motion, pollution), process the data and send it wirelessly — hop by hop — to a **sink / gateway**, which forwards it to the Internet / cloud.

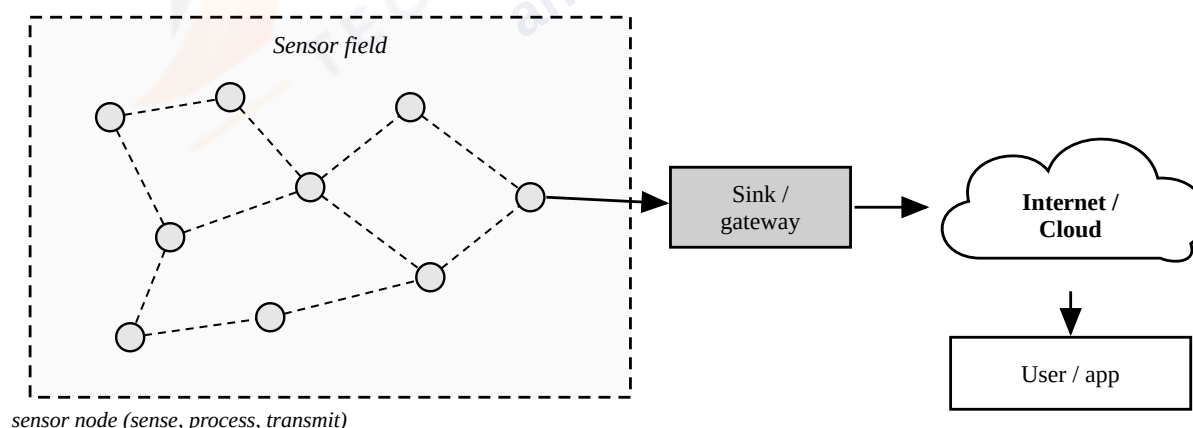


Fig.: Wireless sensor network connected to the cloud

- Sensor node = sensing unit + microcontroller + radio transceiver + battery.
- Uses low-power protocols (ZigBee / IEEE 802.15.4, LoRa, BLE).

- IoT use: smart agriculture, environment monitoring, smart cities, industrial monitoring.

2) GPS (Global Positioning System)

GPS is a satellite navigation system (a constellation of about 24+ satellites). A receiver measures the travel time of signals from several satellites and calculates its position by trilateration — **at least four satellites** are needed for latitude, longitude, altitude and time.

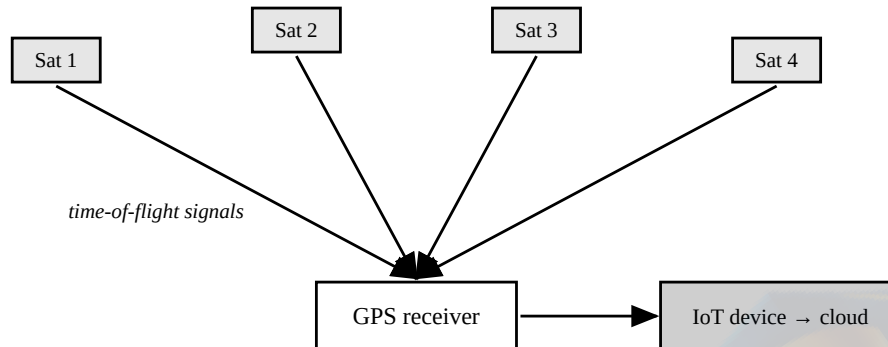


Fig.: GPS — receiver position fixed using signals from four or more satellites

- Gives location and time to IoT devices.
- IoT use: vehicle and fleet tracking, cab apps, logistics, geo-fencing, child / pet trackers, precision farming.

(WSN 2 M, GPS 2 M)

d) Data outsourcing and metadata security.

4 M

1) Data outsourcing

Data outsourcing means the data owner stores its data on, and lets it be processed by, an external **cloud provider** that is not fully trusted. The owner loses physical control of the data, so security must be ensured by the owner's own measures.

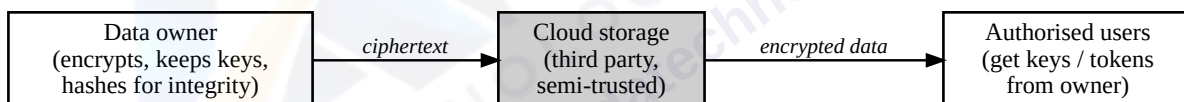


Fig.: Data outsourcing — data protected before it leaves the owner

- **Confidentiality:** encrypt data before outsourcing; keep keys with the owner.
- **Integrity:** verify that the provider has not modified or lost data — hashes, MACs, provable data possession / proof of retrievability.
- **Access control:** only authorised users get keys / tokens; query privacy for searching encrypted data.

2) Metadata security

Metadata is 'data about data' — file names, owner, size, location, timestamps, access lists, tags, encryption-key IDs. In cloud file systems (e.g. GFS / HDFS master) metadata controls access to all data, so if it is leaked or altered, the data can be exposed or lost.

- Encrypt and access-control metadata just like data; protect the metadata server.
- Integrity checks and audit logs on metadata changes; replicate / back up metadata to avoid single point of failure.

- Avoid putting sensitive information in file names / tags.

(Data outsourcing 2 M, metadata security 2 M)

Q.4

a) Role of middleware with layered architecture.

4 M

Middleware is the software layer between the cloud's physical resources / operating systems and the applications. It hides the complexity of distributed resources and provides common services.

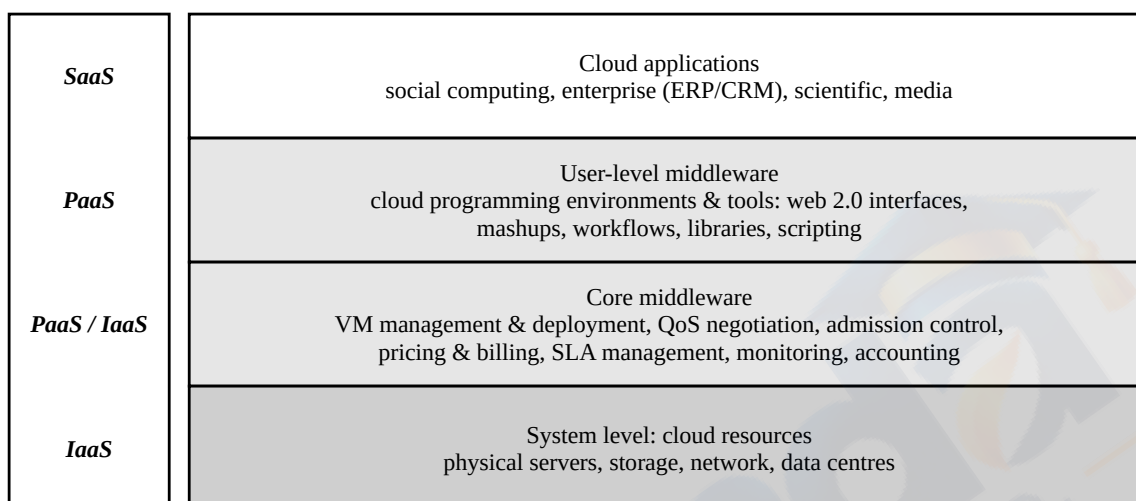


Fig.: Layered cloud computing architecture (middleware layers)

- **Core middleware:** manages the physical infrastructure as a cloud — VM management and deployment, QoS negotiation and admission control, SLA management, pricing / billing, metering, monitoring and accounting.
- **User-level middleware:** gives developers programming environments and tools — web 2.0 interfaces, libraries, APIs, workflow and scripting tools (the basis of PaaS).
- **Communication and integration:** message queues, APIs and service buses connect different applications and services (e.g. RabbitMQ, Amazon SQS).
- **Other roles:** load balancing, security (authentication, encryption), data management and interoperability across heterogeneous systems.

(Diagram 2 M, role 2 M)

b) Purpose of virtualization layer; hypervisor structures.

4 M

Purpose of virtualization layer (VMM / hypervisor):

- Creates the virtual environment (virtual CPU, memory, disk, NIC) in which the guest runs — it acts as the 'host' for the guest.
- Maps virtual resources to real physical resources and schedules them among VMs (sharing, aggregation).
- Isolates guests from each other and from the host — security and fault isolation.
- Translates / traps privileged instructions of guests; enables management features such as snapshot, migration and monitoring.

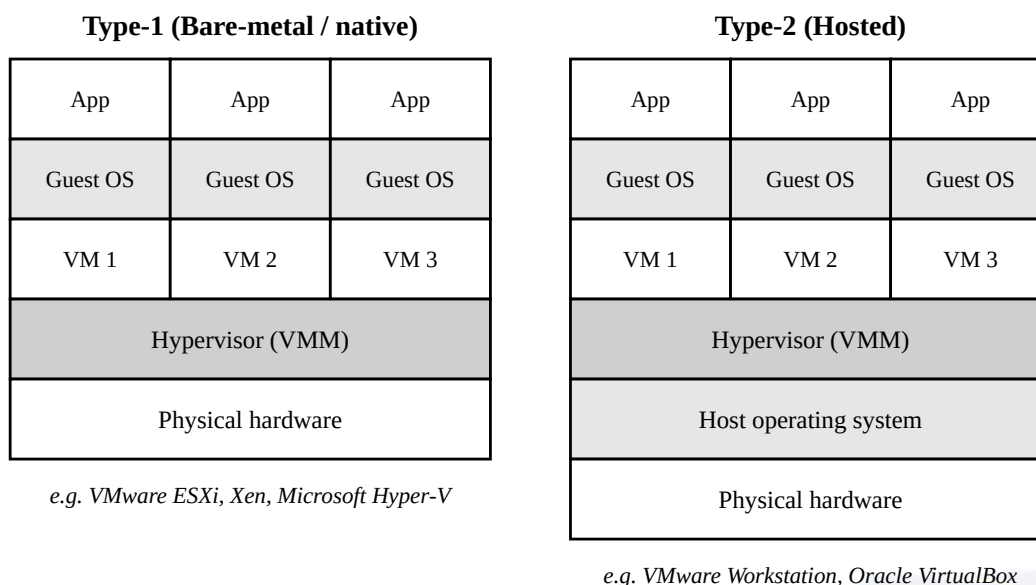


Fig.: Type-1 (bare-metal) and Type-2 (hosted) hypervisor

(Purpose 2 M, diagrams 2 M)

c) Operationalization phase; SLA for IaaS customer.

4 M

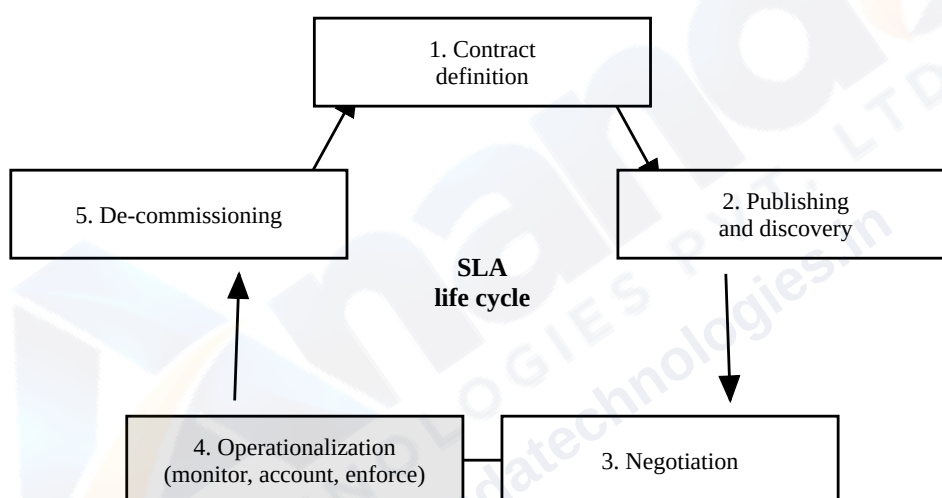


Fig.: SLA life cycle (five phases)

Operationalization is the fourth phase of the SLA life cycle, in which the signed SLA is actually put into operation for the running service. It has three activities:

1. **SLA monitoring:** continuously measure the agreed QoS parameters (availability, response time, throughput) and compare them with targets.
2. **SLA accounting:** record compliance and violations over the billing period; calculate charges and service credits / penalties.
3. **SLA enforcement:** when a violation occurs or is predicted, take corrective action — allocate more resources, migrate VMs, notify the customer, apply penalties.

IaaS customer needs an Infrastructure SLA, because he manages his own OS and applications; what the provider guarantees is the infrastructure — e.g. VM / server availability 99.95 %, power

and network availability, storage durability.

(Operationalization 3 M, SLA type 1 M)

d) Server-side vs client-side encryption.

4 M

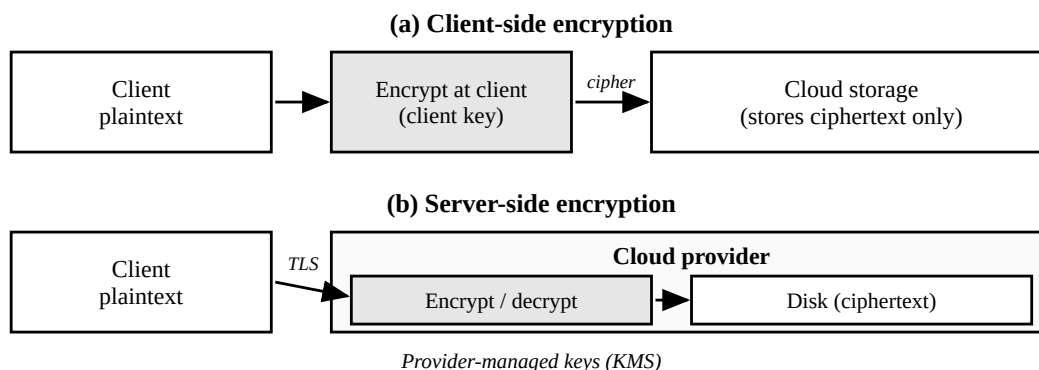


Fig.: Client-side vs server-side encryption

Point	Server-side encryption (SSE)	Client-side encryption (CSE)
Where encrypted	Data encrypted by the cloud provider after it arrives at the server, before writing to disk.	Data encrypted on the client's machine before it is uploaded.
Key management	Keys managed by provider (or customer keys stored in provider KMS).	Keys created and kept by the customer.
Data seen by provider	Provider can see plaintext during processing.	Provider sees only ciphertext.
Effort / performance	Transparent and easy; no load on client.	Extra processing and key handling on client side.
Security level / trust	Requires trust in the provider.	Higher confidentiality; no need to trust provider.
Example	Amazon S3 SSE-S3 / SSE-KMS.	Encrypting files with own tool / SDK before upload (e.g. S3 encryption client).

(Any four points, 1 M each)

e) CRM vs ERP in cloud.

4 M

Point	Cloud CRM	Cloud ERP
Full form / purpose	Customer Relationship Management — manages customers, leads, sales and support.	Enterprise Resource Planning — integrates internal processes: finance, HR, inventory, production.
Focus	Front office; increase sales and customer satisfaction.	Back office; reduce cost and improve efficiency.
Main users	Sales, marketing and customer-service teams.	Accounts, HR, purchase, store and management.
Data handled	Contacts, leads, opportunities, campaigns, tickets.	Ledgers, payroll, stock, purchase orders, supply chain.

Examples (SaaS)	Salesforce Sales Cloud, Zoho CRM, HubSpot.	Oracle NetSuite, SAP S/4HANA Cloud, Zoho Books / Odoo.
-----------------	--	--

(Any four points, 1 M each)

Q.5

a) IaaS model.

6 M

Infrastructure as a Service (IaaS) is the lowest cloud service model, in which the provider delivers **virtualized computing infrastructure** — virtual machines, storage, networks, load balancers — over the Internet on demand and on pay-per-use basis. The user installs and controls the OS, middleware and applications; the provider manages physical hardware, virtualization, storage and networking.

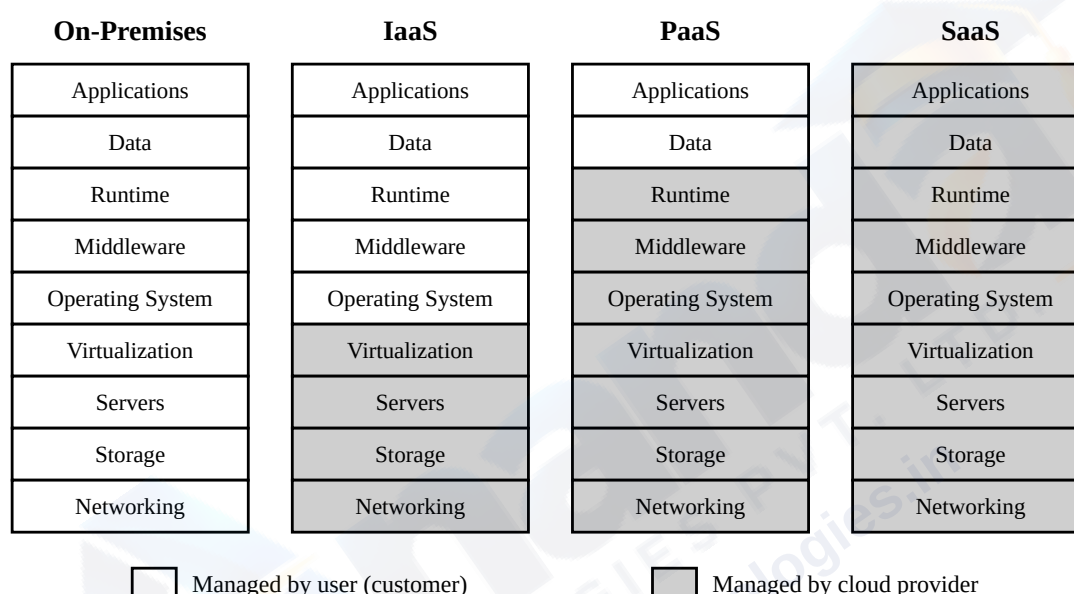


Fig.: Division of management responsibility in IaaS, PaaS and SaaS

Functions:

- Provision / release virtual servers of chosen size (CPU, RAM) and OS image.
- Block, file and object storage; backup and snapshots.
- Virtual networking — IP addresses, firewalls / security groups, load balancers, VPN.
- Monitoring, metering and billing; auto-scaling; self-service portal and APIs.

Advantages:

1. No capital investment in hardware; pay only for what is used.
2. Highly scalable and elastic — add servers in minutes.
3. Full control over OS and software stack (flexibility).
4. Provider handles hardware failures, data-centre security and power.

Disadvantages:

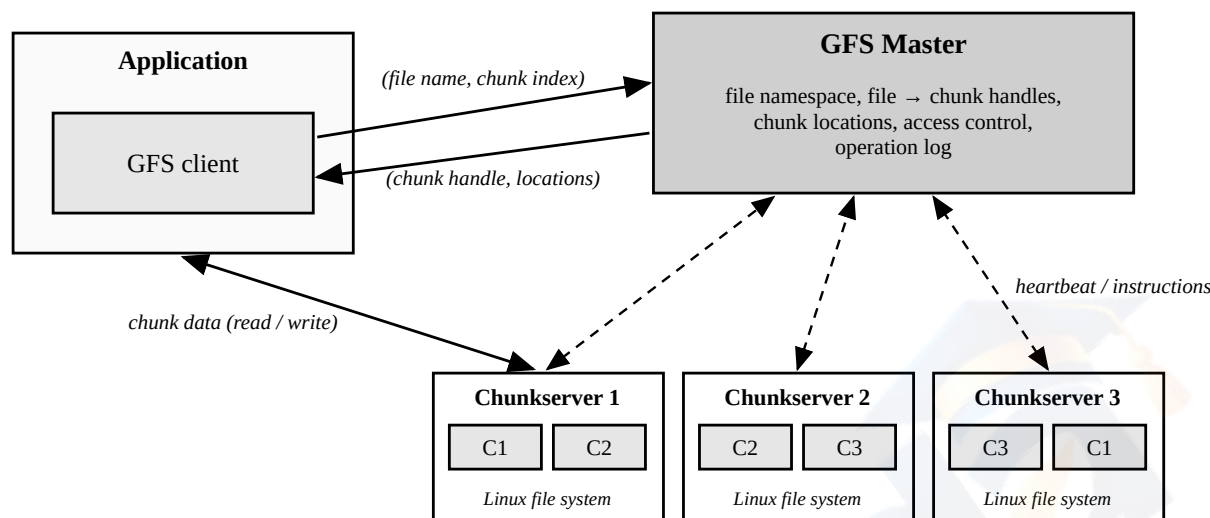
1. User must still manage OS, patches, security of VMs — needs skilled staff.
2. Security / compliance concerns in multi-tenant infrastructure; dependence on Internet.
3. Costs can rise if resources are not monitored; possible vendor lock-in.

Example: Amazon EC2 + EBS + S3, Microsoft Azure Virtual Machines, Google Compute Engine, OpenStack (private IaaS). A startup rents two EC2 instances for its website and adds more during a festival sale.

(Concept 1 M, functions 1 M, advantages 2 M, disadvantages 1 M, example 1 M)

b) Google File System.

6 M



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

Fig.: Google File System (GFS) architecture

Components:

1. **Master:** single master stores all metadata (namespace, file-to-chunk mapping, chunk locations), manages leases, garbage collection and re-replication; communicates with chunkservers using heartbeats.
2. **Chunkservers:** store 64 MB chunks as Linux files; each chunk replicated on 3 chunkservers by default.
3. **Clients:** get metadata from the master, then transfer data directly with chunkservers.

Features / characteristics:

- Designed for very large files and huge clusters of commodity hardware where failures are normal.
- Large chunk size (64 MB) — fewer metadata entries and less client-master interaction.
- Fault tolerance through replication, checksums and fast recovery (operation log, checkpoints).
- Optimised for large streaming reads and **atomic record append** by many clients concurrently; relaxed consistency model.

Advantages:

- High throughput and scalability (thousands of nodes, petabytes).
- Reliability — automatic re-replication of lost chunks; low cost commodity hardware.

Disadvantages:

- Single master can become a bottleneck / single point of failure (mitigated by shadow masters).
- Not suitable for many small files or low-latency random access; proprietary to Google.

(Components with diagram 2 M, features 2 M, advantages 1 M, disadvantages 1 M)

c) ZigBee and smart-home application.

ZigBee is a wireless communication standard for **low-power, low-data-rate, short-range** networks, built on the IEEE 802.15.4 physical and MAC layers. It is maintained by the Connectivity Standards Alliance (formerly ZigBee Alliance).

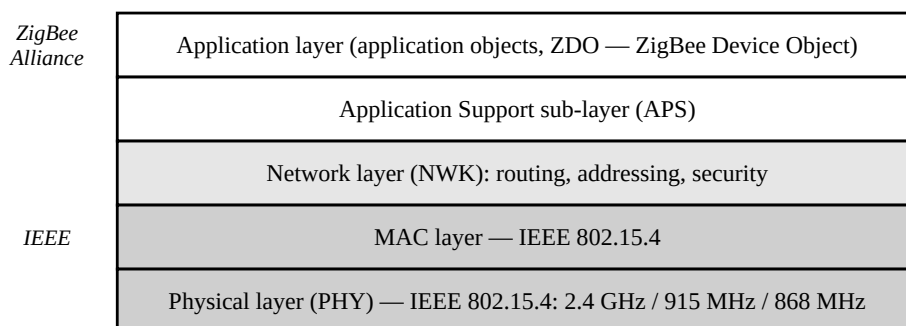


Fig.: ZigBee protocol stack

- Frequency: 2.4 GHz (16 channels, worldwide), 915 MHz (America), 868 MHz (Europe); data rate up to 250 kbps; range 10–100 m.
- Device types: Coordinator, Router, End device; topologies: star, tree, mesh (self-healing).
- Very low power — end devices sleep; batteries last months to years; 128-bit AES security; low cost.

Application — ZigBee-based smart home:

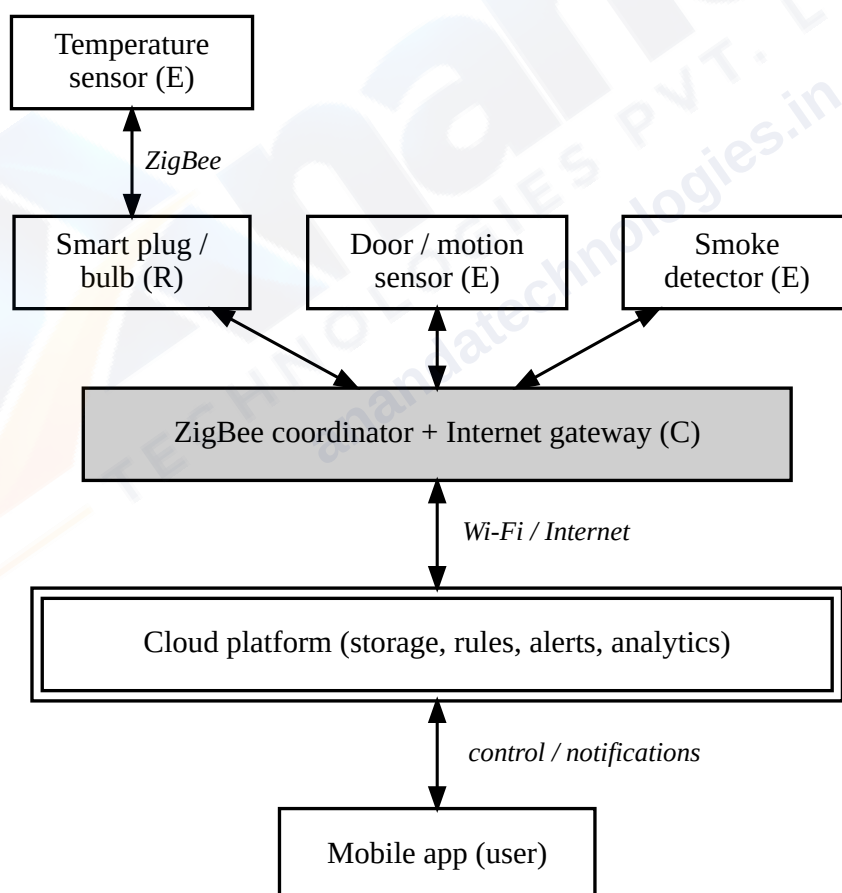


Fig.: ZigBee-based smart home with cloud monitoring

1. Sensors (temperature, door / motion, smoke) are ZigBee end devices; smart plugs / bulbs act as routers and extend the mesh.
2. All devices send readings to the ZigBee coordinator in the home hub / gateway (e.g. Amazon Echo with ZigBee hub, Philips Hue bridge).
3. The gateway forwards data over Wi-Fi / Internet to a cloud platform which stores data, runs rules (e.g. 'if smoke detected → alarm + notify') and analytics.
4. The owner monitors and controls lights, AC and locks from a mobile app anywhere; alerts are pushed instantly.

Other ZigBee applications: smart energy meters, industrial monitoring, healthcare wearables, street-light control.

(ZigBee explanation 3 M, application with diagram 3 M)

Q.6

a) Characteristics and cost benefits of cloud computing.

6 M

Five essential characteristics (NIST):

1. **On-demand self-service:** the user provisions computing resources (server time, storage) automatically through a portal / API, without human interaction with the provider.
2. **Broad network access:** services are available over the network through standard mechanisms, from thin or thick clients — mobiles, laptops, tablets.
3. **Resource pooling:** the provider's resources are pooled to serve many consumers using a multi-tenant model; resources are dynamically assigned; the user generally does not know the exact location.
4. **Rapid elasticity:** capabilities are scaled out and in quickly, often automatically, as per demand; they appear unlimited.
5. **Measured service:** usage is automatically metered, monitored and reported; the user pays for what is used.

Cost benefits:

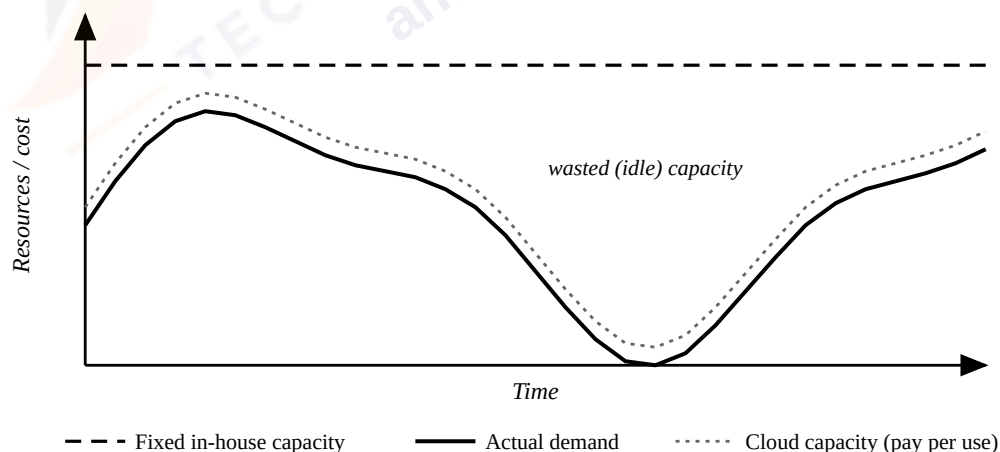


Fig.: Fixed in-house capacity vs elastic cloud capacity

- Capital expenditure becomes operating expenditure — no upfront purchase of servers and licences.
- Pay-per-use — no payment for idle capacity; elasticity avoids buying for peak load.
- Lower maintenance, power, cooling, space and IT staff cost.
- Economies of scale of large providers; faster time-to-market and lower risk for new projects.

(Five characteristics 3 M ($\frac{1}{2}$ M + explanation each), cost benefits 3 M)

b) Virtualization for Linux + Windows on one server; advantages and disadvantages.

6 M

How virtualization helps: a hypervisor (e.g. VMware ESXi, Microsoft Hyper-V, KVM) is installed on the physical server. It creates separate virtual machines — one VM runs **Linux**, another runs **Windows** — each with its own virtual CPU, memory, disk and network card. Both OS run **at the same time**, fully isolated, on the same hardware.

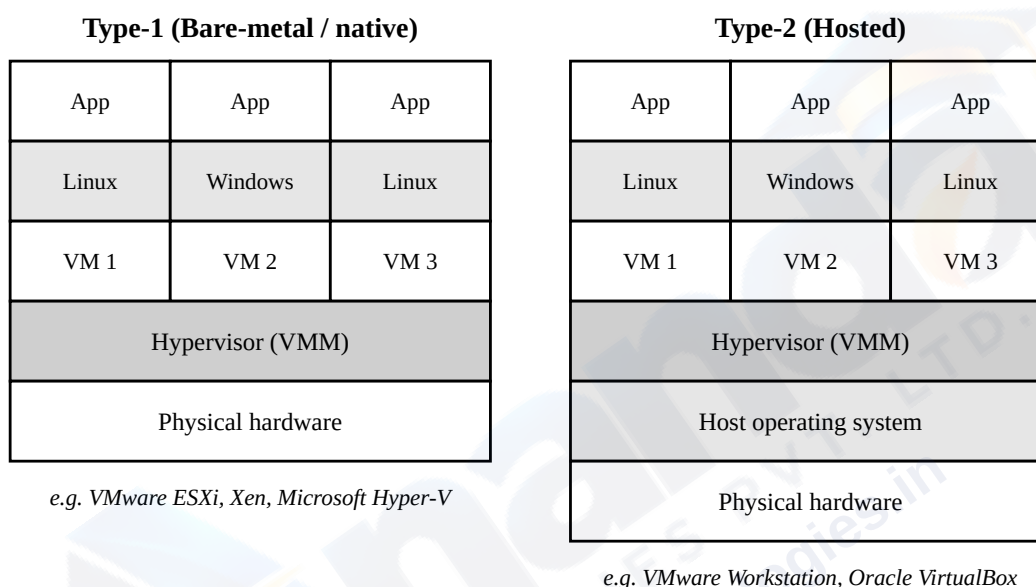


Fig.: Linux and Windows guest OS running as VMs on one physical server (Type-1 is used on servers)

- Fewer physical servers are needed (server consolidation) → hardware, power, cooling and space cost reduced.
- Server utilisation increases from typically 10–20 % to much higher levels.
- New VMs can be created from templates in minutes; VMs can be moved (migrated) between hosts — more flexibility.

Advantages of virtualization:

1. Reduced hardware and operating cost; less energy use (green IT).
2. Better resource utilisation and server consolidation.
3. Fast provisioning and easy scaling; snapshots and cloning.
4. Isolation — failure or attack in one VM does not affect others.
5. High availability, easy backup, disaster recovery and live migration.
6. Run multiple / legacy OS and testing environments on one machine.

Disadvantages of virtualization:

1. High initial cost of powerful servers and hypervisor licences.
2. Performance overhead compared with bare hardware for some workloads.

3. Single point of failure — if the host fails, all its VMs fail (unless HA is configured).
4. Needs skilled administrators; VM sprawl and licensing complexity; hypervisor-level security risks.

(How it helps 2 M, advantages 2 M, disadvantages 2 M)

c) Benefits of GAE, OpenStack and OpenNebula.

6 M

1) Google App Engine (PaaS)

- Fully managed serverless platform — no servers to manage; developers deploy code directly.
- Automatic scaling and load balancing; pay-per-use with free quota.
- Integrated services: Datastore / Firestore, Cloud SQL, Memcache, task queues, cron, logging; supports Python, Java, Go, PHP, Node.js etc.

2) OpenStack (open-source IaaS)

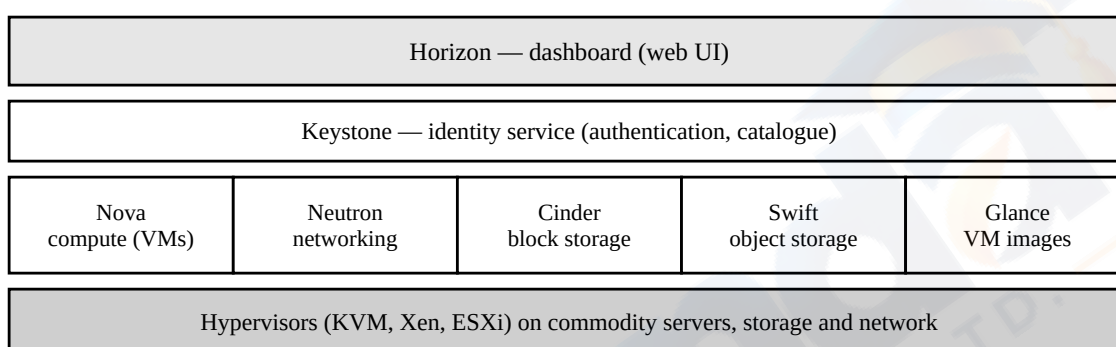


Fig.: OpenStack core components

- Free, open-source software to build private and public IaaS clouds on commodity hardware — no licence cost, no vendor lock-in.
- Modular services: Nova (compute), Swift (object storage), Cinder (block storage), Neutron (network), Keystone (identity), Glance (images), Horizon (dashboard).
- Supports many hypervisors (KVM, Xen, ESXi), highly scalable, large community and industry support; standard APIs.

3) OpenNebula (open-source cloud management)

- Simple, lightweight platform to build and manage private, hybrid and edge clouds from existing virtualized data centres.
- Supports KVM, VMware vCenter, LXC containers; can burst to public clouds (hybrid).
- Features: VM templates, multi-tenancy with user groups / quotas, self-service portal (Sunstone), monitoring and accounting; easy to install and maintain.

(Each platform 2 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