

Cloud Computing (315325)

Model Question Paper — Set B · Model Answers

Q.1

a) Community cloud.

2 M

Community cloud: a cloud infrastructure that is shared by **several organisations having common concerns** (mission, security requirements, policy, compliance). It may be managed by the organisations themselves or by a third party, and may be on- or off-premises.

Example: a government cloud shared by different government departments (e.g. MeghRaj – GI Cloud), or a cloud shared by banks / hospitals / universities.

(Definition 1 M, example 1 M)

b) Characteristics of virtualized environment.

2 M

1. **Increased security** — the guest runs in an isolated, controlled environment; host is protected from harmful guest operations.
2. **Managed execution — sharing:** many guests share the same physical host resources.
3. **Managed execution — aggregation:** several physical hosts can be combined and shown to a guest as one resource.
4. **Managed execution — emulation and isolation:** a different environment can be emulated for the guest; each guest is isolated from others.
5. **Performance tuning** — resources given to a guest can be controlled (e.g. memory, CPU share).
6. **Portability** — a virtual image can be moved and run on another host.

(Any four, ½ M each)

c) VM consolidation.

2 M

VM consolidation is the technique of **placing (migrating) many under-utilised virtual machines onto fewer physical servers**, so that server utilisation increases and the idle servers can be switched off or put to sleep to save power and cost. It may be static (planned placement) or dynamic (using live migration at run time).

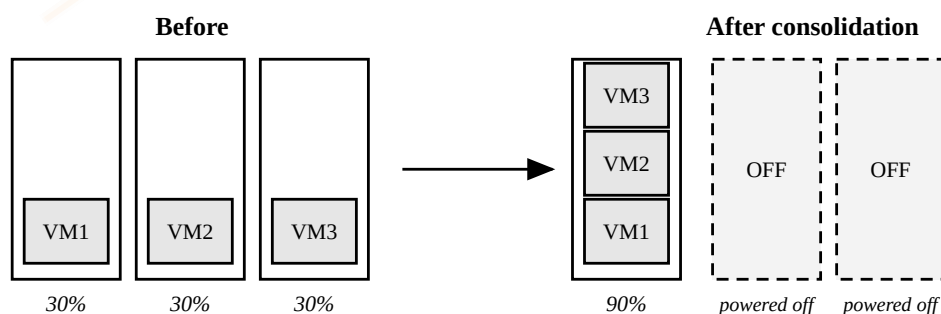


Fig.: VM consolidation — under-used servers merged, idle servers switched off

(Definition 2 M)

d) Five phases of SLA management.

2 M

1. Feasibility analysis
2. On-boarding
3. Pre-production
4. Production
5. Termination

*(All five phases 2 M)***e) Legal risks of cloud computing.**

2 M

1. **Subpoena and e-discovery** — data may be seized by authorities; other tenants' data may be exposed.
2. **Changes of jurisdiction** — data stored in another country is governed by that country's laws.
3. **Data protection risks** — difficult for the customer to check how the provider handles personal data.
4. **Licensing risks** — software licences charged per machine may be violated in the cloud.

*(Any four, ½ M each)***f) Features of ZigBee.**

2 M

1. Based on IEEE 802.15.4; operates at 2.4 GHz (worldwide), 915 MHz and 868 MHz.
2. Low data rate — up to 250 kbps.
3. Very low power — battery life of months / years; devices sleep most of the time.
4. Short range — about 10–100 m; extended by mesh routing.
5. Supports star, tree and mesh topologies; large networks (up to about 65,000 nodes).
6. Secure — 128-bit AES encryption; low cost.

*(Any four, ½ M each)***g) Rapid elasticity and measured service.**

2 M

- **Rapid elasticity:** computing capabilities can be provisioned and released quickly — automatically scaling out when demand rises and scaling in when it falls. To the user, resources appear unlimited. e.g. auto-scaling of web servers during a sale.
- **Measured service:** cloud systems automatically meter resource usage (storage, CPU hours, bandwidth, active users); usage is monitored, reported and billed, giving transparency to both provider and user (pay-per-use).

*(Each 1 M)***Q.2****a) Recommended model: PaaS.**

4 M

Recommended model: Platform as a Service (PaaS). The team only wants to write and deploy code; PaaS gives a ready platform — runtime, middleware, database, web server, development tools — on the provider's infrastructure. The provider manages servers, OS, patching and scaling; the developer manages only the **application and its data**.

Examples: Google App Engine, Heroku, Microsoft Azure App Service, AWS Elastic Beanstalk.

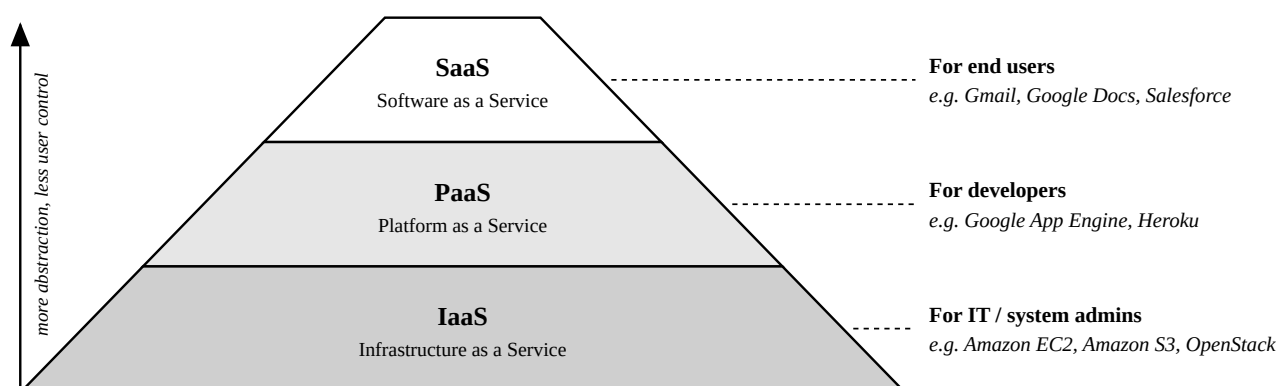


Fig.: Cloud service models (SPI model) with examples

Advantages:

1. Faster development and deployment — no server set-up; built-in scaling, load balancing and database services.
2. Lower cost — no hardware / OS licences or system administrators; pay per use.

Disadvantages:

1. Vendor lock-in — application depends on the provider's APIs and services; migration is difficult.
2. Less control — cannot change OS / runtime versions; limited language or framework support.

(Correct model with justification 2 M, advantages 1 M, disadvantages 1 M)

b) Live migration vs cold migration.

4 M

Point	Live (hot) migration	Cold (offline) migration
VM state	VM is running while it is moved.	VM is powered off (or suspended) before moving.
Downtime	Near-zero (milliseconds) — users stay connected.	Service unavailable during the whole migration.
What is copied	Memory pages, CPU state and device state (iteratively — pre-copy / post-copy).	Only VM files / disk image and configuration.
Speed / resources	Complex; needs high network bandwidth and usually shared storage.	Simple and safe; network load lower; can move between different storage.
Compatibility	Source and destination CPUs / hypervisors must be compatible.	Fewer restrictions on hardware compatibility.
Use	Critical services, load balancing, maintenance without downtime (VMware vMotion, Xen live migration).	Planned maintenance, moving non-critical VMs, cross-site moves.

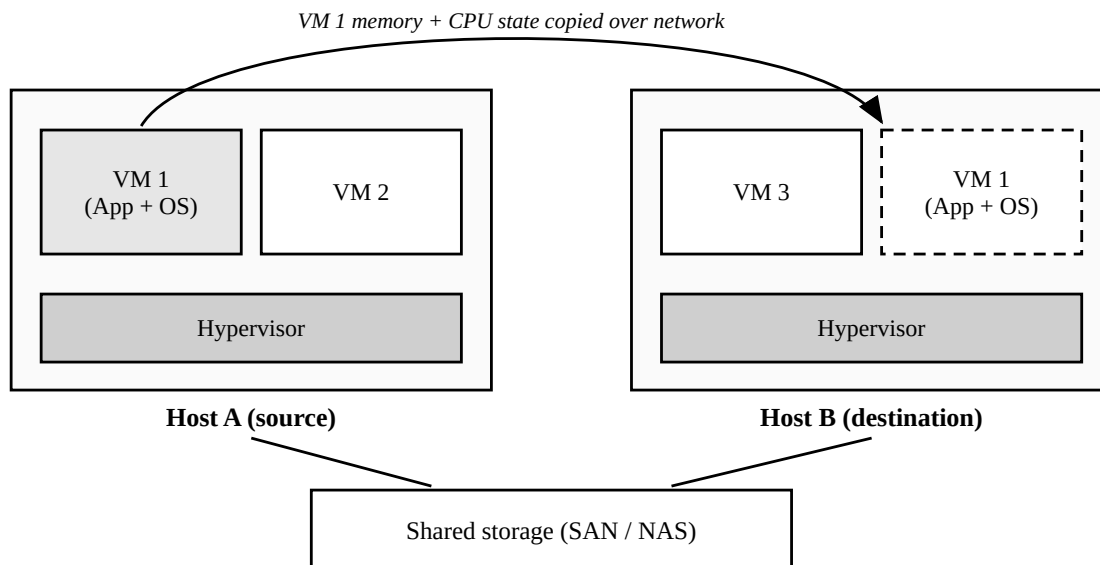


Fig.: VM migration between physical hosts sharing storage

(Any four points, 1 M each)

c) HDFS heartbeat, replication and balancing.

4 M

HDFS (Hadoop Distributed File System) has one **NameNode** (metadata) and many **DataNodes** (data blocks, default size 128 MB in Hadoop 2 and later).

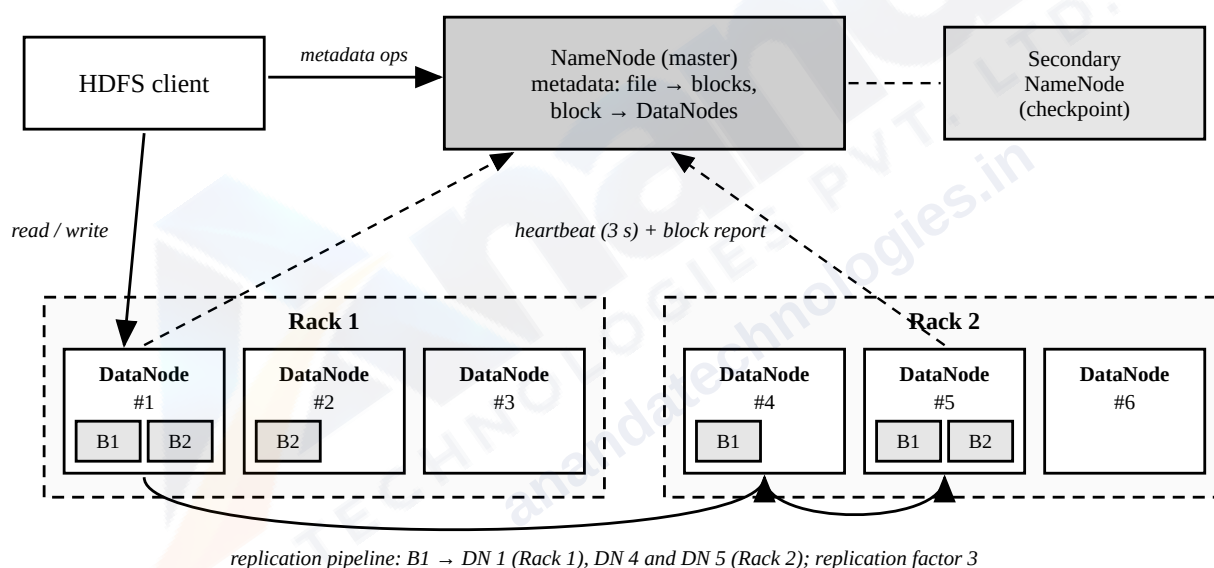


Fig.: HDFS architecture (NameNode, DataNodes, rack-aware replication)

- Heartbeat:** every DataNode sends a heartbeat signal to the NameNode periodically (default every 3 seconds) to show it is alive, along with periodic *block reports* (list of blocks it holds). If no heartbeat is received for a timeout period (about 10 minutes by default), the NameNode marks the DataNode dead and re-replicates its blocks on other nodes.
- Replication:** each block is stored on several DataNodes — default **replication factor 3**. Rack-aware placement: first replica on the writer's node, second on a node in a different rack, third on another node of that second rack. Blocks are written through a pipeline of DataNodes. This gives fault tolerance and faster reads.

3. **Balancing:** as nodes are added or data grows, disks become unevenly used. The HDFS **balancer** tool moves blocks from over-utilised to under-utilised DataNodes until every node's usage is within a threshold of the cluster average, without breaking the replica placement rules.

(Heartbeat 1 M, replication 2 M (with diagram), balancing 1 M)

d) Content-level security and Security-as-a-Service.

4 M

Content-level security (CLS) applies protection to the **content (data / document) itself** rather than only to the network or server. Each item carries its own encryption and access policy (e.g. rights-management on a document), so it stays protected wherever it is stored or moved — even in a third-party cloud.

Pros	Cons
Protection travels with the data, independent of location / provider.	Complex key and policy management for large volumes of data.
Fine-grained control — who can open, edit, print, forward.	Processing / encryption overhead; slower access.
Reduces impact of breaches in the infrastructure.	Searching / indexing encrypted content is difficult; needs compatible tools.

Features of Security-as-a-Service (SECaaS): security delivered from the cloud on a subscription basis —

1. Identity and access management (SSO, MFA)
2. Data loss prevention and encryption / key management
3. Web and e-mail security (filtering of malware, spam, phishing)
4. Intrusion detection / prevention and network security
5. Security information and event management (SIEM), log monitoring
6. Vulnerability scanning, security assessment, business continuity / backup

(CLS with pros and cons 2 M, any four SECaaS features 2 M)

Q.3

a) VMware full virtualization reference model.

4 M

VMware implements **full virtualization**: the guest OS runs **unmodified** and is not aware that it is virtualized. The whole hardware is emulated by the VMM.

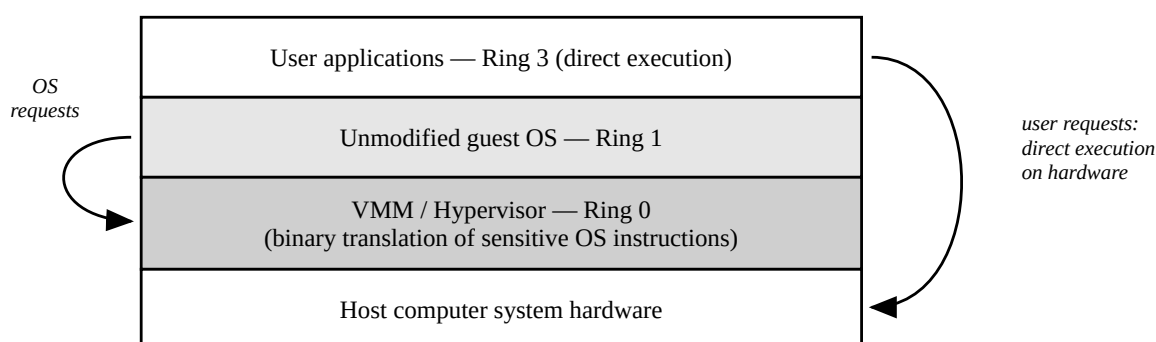


Fig.: VMware full virtualization reference model (binary translation)

1. **Direct execution:** user-level application code (Ring 3) runs directly on the processor for high performance.
2. **Binary translation:** the guest OS is moved to Ring 1; its privileged / sensitive kernel instructions (which cannot be virtualized directly on x86) are trapped and **translated on the fly** by the VMM into safe instruction sequences. Translated code is cached for reuse.
3. **VMM / hypervisor in Ring 0:** controls the hardware, emulates virtual devices (CPU, memory, NIC, disk) and isolates VMs.
4. **Implementations:** hosted (Type-2) — VMware Workstation / Player on a host OS; bare-metal (Type-1) — VMware ESXi in data centres. Modern versions also use hardware-assisted virtualization (Intel VT-x / AMD-V).

(Diagram 2 M, explanation 2 M)

b) Infrastructure SLA vs application SLA; e-commerce case.

4 M

Point	Infrastructure SLA	Application SLA
Covers	Infrastructure provided by the provider — servers, power, network, data centre.	Performance of the hosted application as seen by its users.
Typical metrics	Hardware availability (e.g. 99.95 %), power availability, network availability, backbone latency.	Web-page response time, transactions per second, DB / web-server latency, application uptime.
Mainly for	IaaS customers who run their own software.	SaaS / PaaS customers who use an application.
Who measures	Measured at the data-centre / VM level.	Measured end-to-end at the application level.

For the e-commerce company on a SaaS shopping platform, the application SLA is more useful. The company does not manage any servers; what matters to its business is that the store pages open fast, checkout transactions succeed and the site is available during sales. e.g. an application SLA can promise '99.9 % store availability per month and page response time under 2 seconds for 95 % of requests, with service credits if violated'. The infrastructure SLA is the provider's internal concern.

(Differences 2 M, choice with justification and example 2 M)

c) RFID for IoT.

4 M

RFID (Radio Frequency Identification) uses radio waves to automatically identify and track objects that carry a tag — without line of sight or contact. It gives every 'thing' a unique identity, which is the first requirement of IoT.

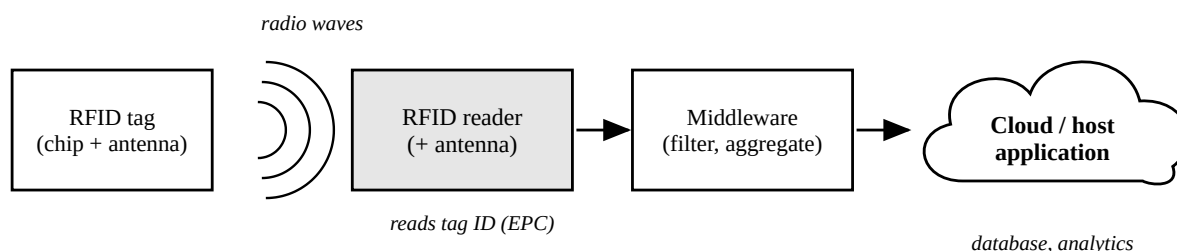


Fig.: RFID system as an IoT enabling technology

1. **RFID tag:** microchip + antenna attached to the object; stores a unique ID (e.g. EPC). *Passive* tags have no battery (powered by reader's field); *active* tags have a battery and longer range.
2. **RFID reader:** sends radio signals, energises the tag and reads its data (LF, HF, UHF bands).
3. **Middleware:** filters duplicate reads, aggregates events and forwards them to applications.
4. **Cloud / host application:** stores and analyses data — inventory, tracking, alerts.

IoT applications: supply-chain and warehouse tracking, retail billing, toll collection (FASTag), library management, animal tracking, access cards.

(Diagram 2 M, explanation 2 M)

d) Role of trust management.

4 M

When an organisation moves to the cloud, its data and processing are controlled by a third party.

Trust management establishes, measures and maintains the confidence of users that the provider will behave as expected — securely, reliably and as per the SLA.

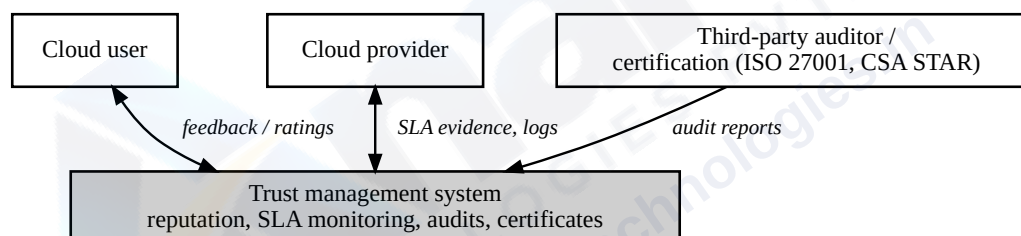


Fig.: Trust management between cloud user and provider

1. **Selecting a provider:** reputation and feedback-based trust models help users pick dependable providers.
2. **SLA verification:** continuous monitoring of SLA parameters gives evidence whether promises are kept; violations reduce the trust score.
3. **Certification and auditing:** third-party audits (ISO 27001, SOC 2, CSA STAR) and transparency reports prove security controls.
4. **Security between providers and users:** trusted identity (federated IAM), encryption, logging and accountability make both sides responsible; in federated clouds providers also evaluate trust in each other before sharing load.

(Any four points, 1 M each)

Q.4

a) Life cycle of a virtual machine.

4 M

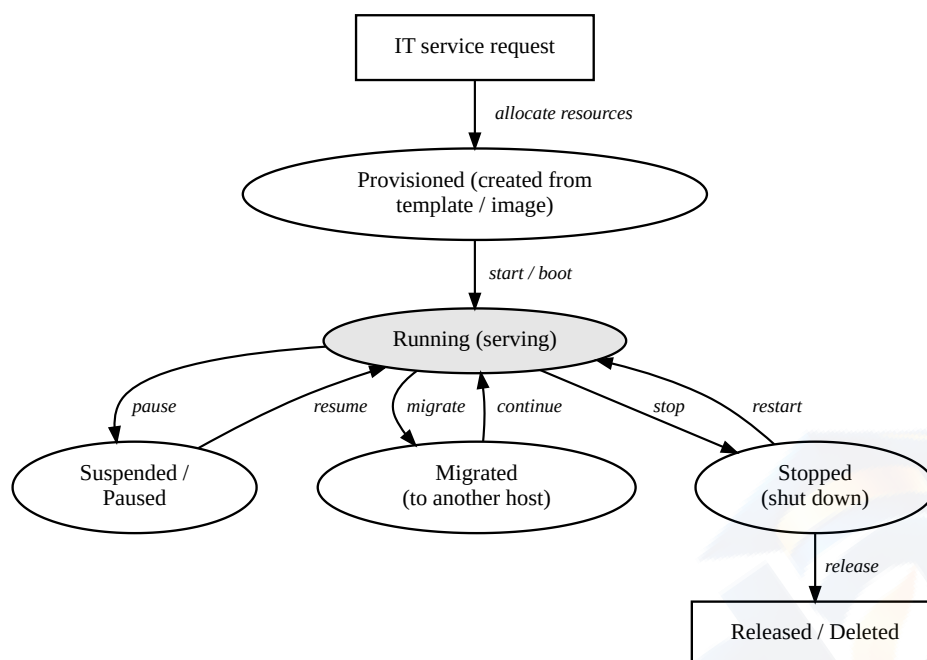


Fig.: Life cycle of a virtual machine

- IT service request:** a user / application requests a server; the request is checked against available resources.
- Provisioning:** a VM is created from a template or image; CPU, memory, storage and network are allocated, OS and applications configured.
- Running (serving):** the VM is started and serves the request. During this phase it may be *paused / suspended* and *resumed*, or *migrated* to another host for load balancing or maintenance.
- Stopped / released:** when the service is no longer required the VM is shut down and its resources are released back to the pool; its image may be deleted or archived.

(Diagram 2 M, explanation 2 M)

b) Federated cloud architecture.

4 M

Cloud federation interconnects the cloud environments of **two or more providers** so that they can share resources and balance load. When one provider's data centre is overloaded, requests are served by a partner provider — giving the user better QoS, availability and lower cost.

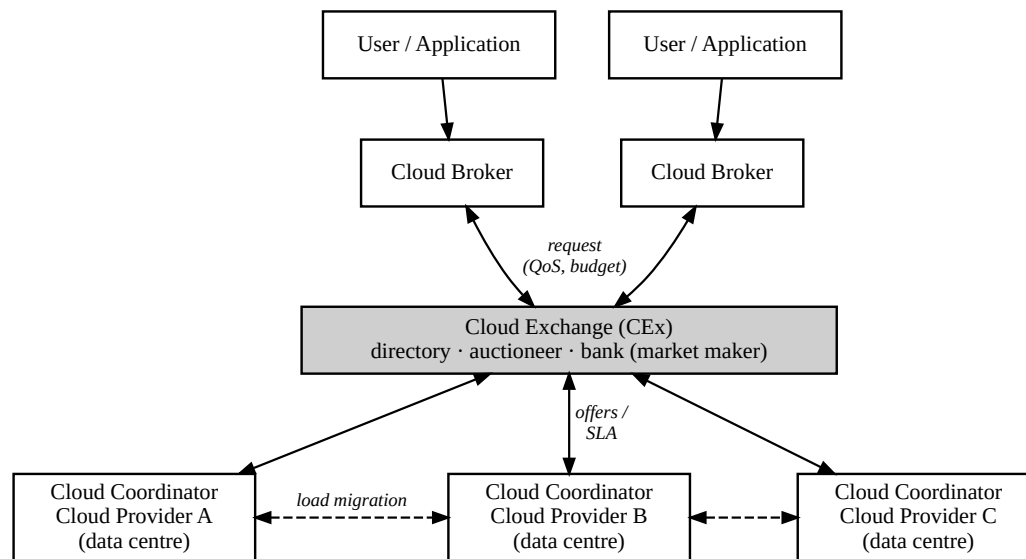


Fig.: Federated cloud (InterCloud) architecture with Cloud Exchange

1. **Cloud Coordinator:** one per provider; exports the provider's services to the federation, monitors its load and decides when to borrow or lend resources.
2. **Cloud Broker:** acts for the user — discovers suitable providers through the exchange, negotiates price and SLA, and deploys the application.
3. **Cloud Exchange (CEx):** market maker that brings brokers and coordinators together; keeps a directory of offers, matches demand and supply, handles auction and payment.
4. **Federation models:** *horizontal* (providers at the same layer, e.g. IaaS-IaaS), *vertical* (across layers SaaS-PaaS-IaaS) and *hybrid / inter-cloud*.

(Diagram 2 M, explanation 2 M)

c) Importance of access control; models.

4 M

Access control decides **who** can access **which** cloud resource and **what** operations they may perform. It is enforced after authentication.

Importance:

1. Protects confidential data in a shared, multi-tenant environment from unauthorised users and other tenants.
2. Enforces the principle of least privilege — reduces damage from compromised accounts and insider threats.
3. Required for regulatory compliance and audit (who accessed what and when).
4. Allows safe sharing and delegation — temporary or role-based access for employees, partners and applications.

Access control models:

- **Role-Based Access Control (RBAC):** permissions assigned to roles (admin, developer, viewer); users get roles.
- **Attribute-Based Access Control (ABAC):** decision based on attributes — user department, time, location, resource tags.
- Also: Discretionary (DAC) and Mandatory (MAC) access control.

(Importance 2 M, two models 2 M)

d) Cost benefits of cloud computing.

4 M

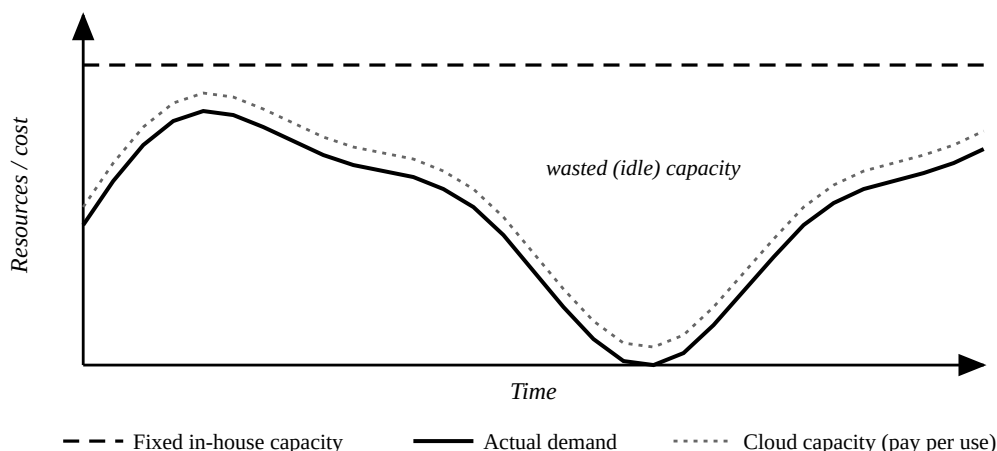


Fig.: Fixed in-house capacity vs elastic cloud capacity

1. **No capital expenditure (CapEx → OpEx):** no need to buy servers, storage, licences and data-centre space; pay a monthly operating cost.
2. **Pay-as-you-go:** pay only for resources actually used; no money wasted on idle, over-provisioned capacity (see figure).
3. **Reduced maintenance and staff cost:** hardware repair, upgrades, patching, power and cooling are handled by the provider.
4. **Economies of scale:** large providers buy hardware and energy cheaply and pass the saving to users.
5. **Elasticity:** scale up for peak load and scale down later — avoids buying for peak demand.
6. **Faster time to market and lower risk:** start projects without big investment; shut them down without loss.

(Any four benefits, 1 M each)

e) CRM and ERP in the cloud.

4 M

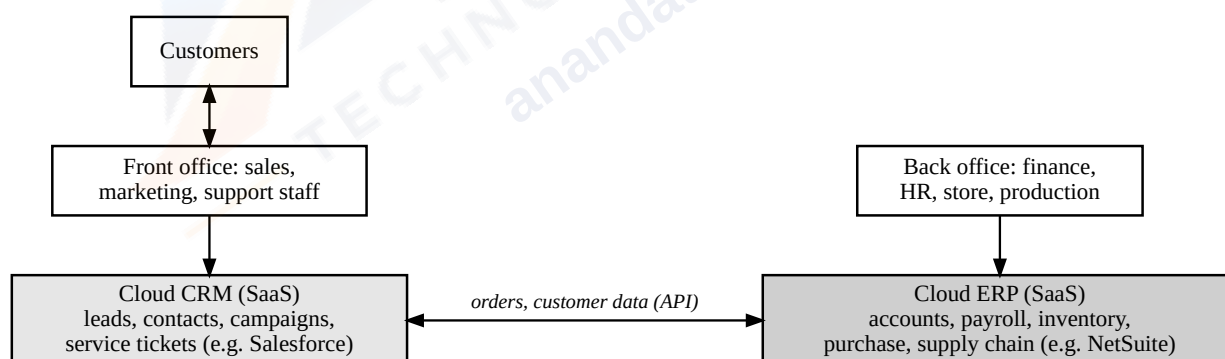


Fig.: Cloud CRM and cloud ERP serving a business

- **Cloud CRM (Customer Relationship Management):** SaaS application that manages a company's interactions with customers — leads, contacts, sales pipeline, marketing campaigns and customer support. Example: Salesforce.com Sales Cloud, Zoho CRM, Microsoft Dynamics 365. Accessible to sales staff from anywhere, including mobile.

- **Cloud ERP (Enterprise Resource Planning):** SaaS application that integrates internal business processes — finance and accounting, HR and payroll, inventory, purchase, manufacturing and supply chain — in one database. Example: Oracle NetSuite, SAP Business ByDesign / S/4HANA Cloud.
- **Benefits:** no installation or servers, subscription pricing suitable for small and medium businesses, automatic updates, real-time data and easy integration between CRM and ERP through APIs.
- **Difference:** CRM is customer-facing (front office, increases sales); ERP is internal (back office, reduces cost and improves efficiency).

(CRM 2 M, ERP 2 M)

Q.5

a) Types of virtualization.

6 M

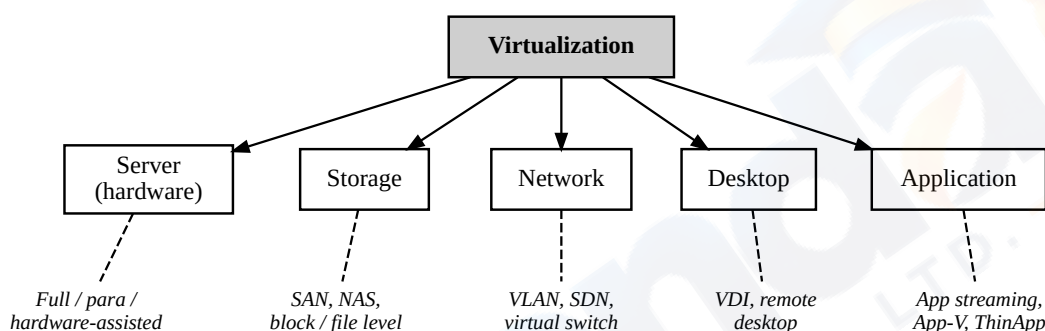


Fig.: Types of virtualization

1. **Server (hardware) virtualization:** one physical server is divided into many virtual machines using a hypervisor; each VM runs its own OS. Techniques: full virtualization, paravirtualization, hardware-assisted. *Benefit:* server consolidation, lower cost.
2. **Storage virtualization:** physical storage from many devices is pooled and presented as a single logical storage (virtual volumes / LUNs); implemented in SAN / NAS at block or file level. *Benefit:* better utilisation, easy management, non-disruptive data migration.
3. **Network virtualization:** physical network is divided into logical networks (VLAN) or combined; software-defined networking (SDN), virtual switches and virtual NICs. *Benefit:* isolation, flexible network provisioning.
4. **Desktop virtualization:** the user's desktop (OS + apps) runs on a central server (VDI) and is accessed remotely from any device. *Benefit:* central management, security, access from anywhere.
5. **Application virtualization:** applications are encapsulated from the underlying OS and streamed / run in an isolated environment without installation (e.g. Microsoft App-V, VMware ThinApp). *Benefit:* no conflicts, easy deployment.

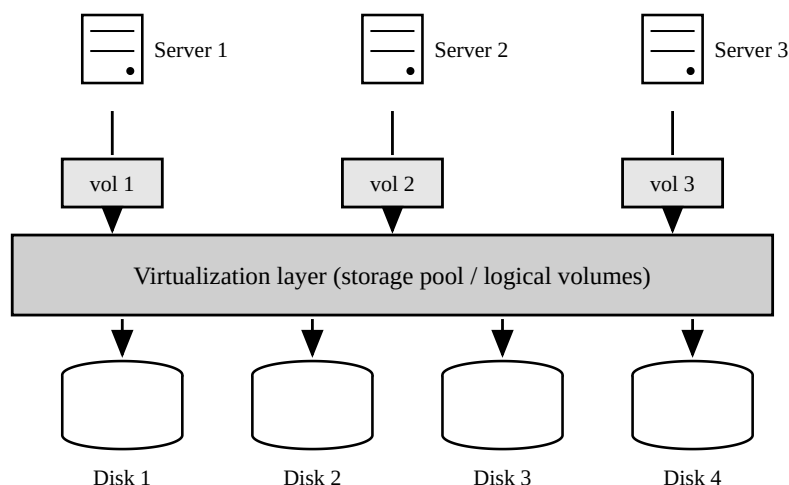


Fig.: Storage virtualization — physical disks pooled into virtual volumes

(Diagram 1 M, each type 1 M)

b) Five phases of SLA management.

6 M

SLA management covers the whole period from deciding whether a customer's application can be hosted to the end of the contract (as described for hosting providers):

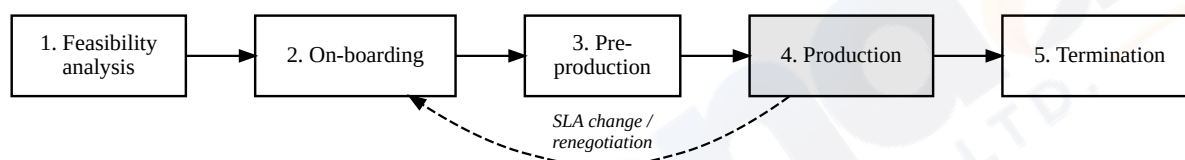


Fig.: Five phases of SLA management

1. **Feasibility analysis:** the provider studies the customer's application — technical feasibility (resources, architecture), infrastructure feasibility (capacity available) and financial feasibility (cost, price). Decides whether to accept the application.
2. **On-boarding:** the application is moved to the provider's platform; it is packaged, tested, and its performance is measured to create policies (e.g. how many VMs per load). SLA terms are finalised; if not acceptable, negotiation is repeated.
3. **Pre-production:** the application is hosted in a staging (simulated production) environment; the customer verifies and validates it against the SLA before it goes live.
4. **Production:** the application is live for end users; the provider monitors SLA parameters, scales resources and handles violations. If customer needs change, the SLA is re-negotiated (back to on-boarding).
5. **Termination:** when the customer ends the contract the application is removed, customer data is handed back / archived and resources are released.

(Diagram 1 M, five phases 5 M)

c) Policy & organizational, technical and legal risks.

6 M

Cloud risks are commonly grouped (ENISA classification) into three types:

1) Policy and organizational risks

- Lock-in — dependence on one provider's formats / APIs.
- Loss of governance — physical and logical control handed to the provider.

- Compliance challenges — provider may not satisfy standards (PCI-DSS, ISO, HIPAA).
- Loss of business reputation due to co-tenant activities; provider failure, termination or acquisition; supply-chain failure.

2) Technical risks

- Resource exhaustion — under- or over-provisioning.
- Isolation failure between tenants; malicious insider at the provider.
- Management interface compromise; interception of data in transit; data leakage.
- Insecure / ineffective data deletion; DDoS attacks; loss of encryption keys.

3) Legal risks

- Subpoena and e-discovery — seizure of shared hardware exposes data.
- Changes of jurisdiction — data stored in other countries.
- Data protection risks — customer cannot verify provider's handling of personal data.
- Licensing risks — per-machine software licences in a dynamic cloud.

(Each type with any three / four risks 2 M)

Q.6

a) IaaS, PaaS, SaaS with diagram and comparison.

6 M

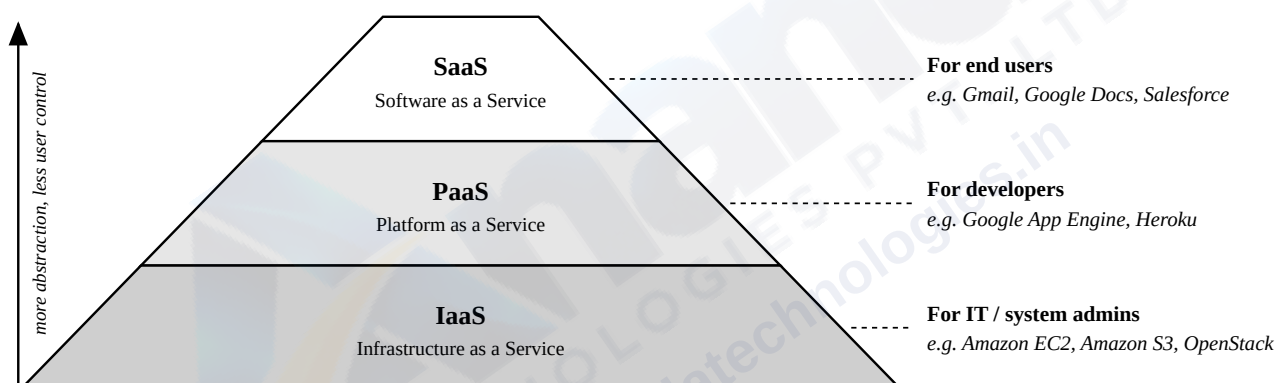


Fig.: Cloud service models (SPI model) with examples

- **IaaS:** provider supplies virtualized infrastructure — VMs, storage, networks; user installs and manages OS, middleware and apps. e.g. Amazon EC2, Google Compute Engine.
- **PaaS:** provider supplies a platform (runtime, database, tools) to develop and deploy apps; user manages only code and data. e.g. Google App Engine, Heroku.
- **SaaS:** complete application delivered over the Internet; user only uses it. e.g. Gmail, Salesforce, Microsoft 365.

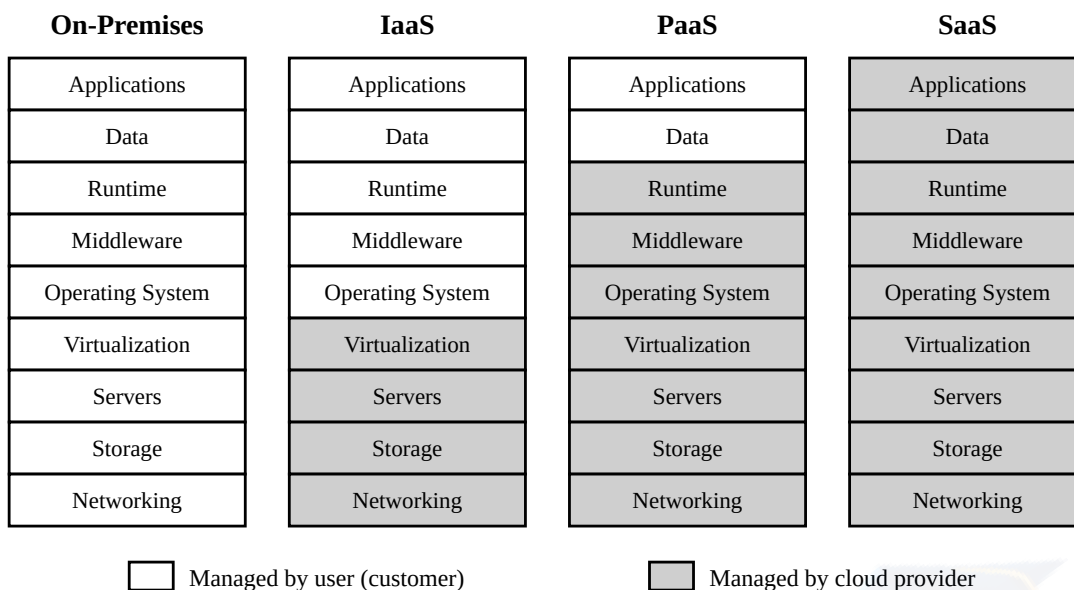


Fig.: Division of management responsibility in IaaS, PaaS and SaaS

Point	IaaS	PaaS	SaaS
Provides	Virtual servers, storage, network	Runtime, middleware, DB, dev tools	Ready-to-use application
User manages	OS, middleware, runtime, apps, data	Applications and data	Nothing (only settings / own data)
Users	System / network administrators	Developers	End users
Control / flexibility	Highest	Medium	Lowest
Technical knowledge needed	High	Medium (programming)	Low
Pricing	Per VM-hour, GB, bandwidth	Per app instance / resources used	Per user per month (subscription)
Examples	Amazon EC2, S3, Azure VMs	Google App Engine, Heroku	Gmail, Salesforce, Office 365

(Explanation with diagram 3 M, any six points of comparison 3 M)

b) Cloud trends supporting ubiquitous computing.

6 M

Ubiquitous (pervasive) computing means that computing is available **anytime, anywhere, on any device**, often invisibly. The cloud provides the always-available back end — data, processing and applications live in the cloud, and any connected device simply acts as an access point.

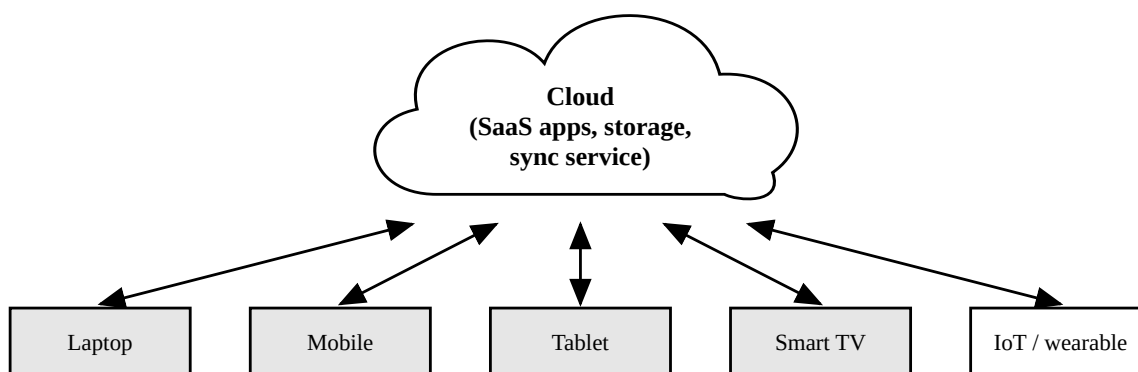


Fig.: Cloud-supported ubiquitous computing — same data/apps from any device, anywhere

Cloud trends that support it:

1. **SaaS and web / browser-based applications** — no installation; work from any browser.
2. **Mobile cloud computing** — heavy processing and storage moved from phones to the cloud; apps sync across devices.
3. **Cloud storage and synchronisation** — Google Drive, OneDrive, iCloud keep the same files on all devices.
4. **Streaming services** — video, music and even games / desktops streamed on demand without downloading.
5. **IoT + cloud, edge / fog computing** — sensors and wearables send data to cloud; edge nodes give low-latency processing near the user.
6. **Serverless and API-based services** — voice assistants and smart apps call cloud functions from any device.

Example: A student watches a lecture on YouTube on a phone while travelling, opens the same notes in Google Docs on the college lab PC, and submits the assignment through Google Classroom from a laptop at home — nothing is installed or downloaded; the cloud keeps everything in sync.

(Concept with diagram 2 M, trends 3 M, example 1 M)

c) Benefits of Google App Engine for developers; features.

6 M

Google App Engine (GAE) is a fully managed, serverless **PaaS** from Google Cloud for building and running web applications and APIs.

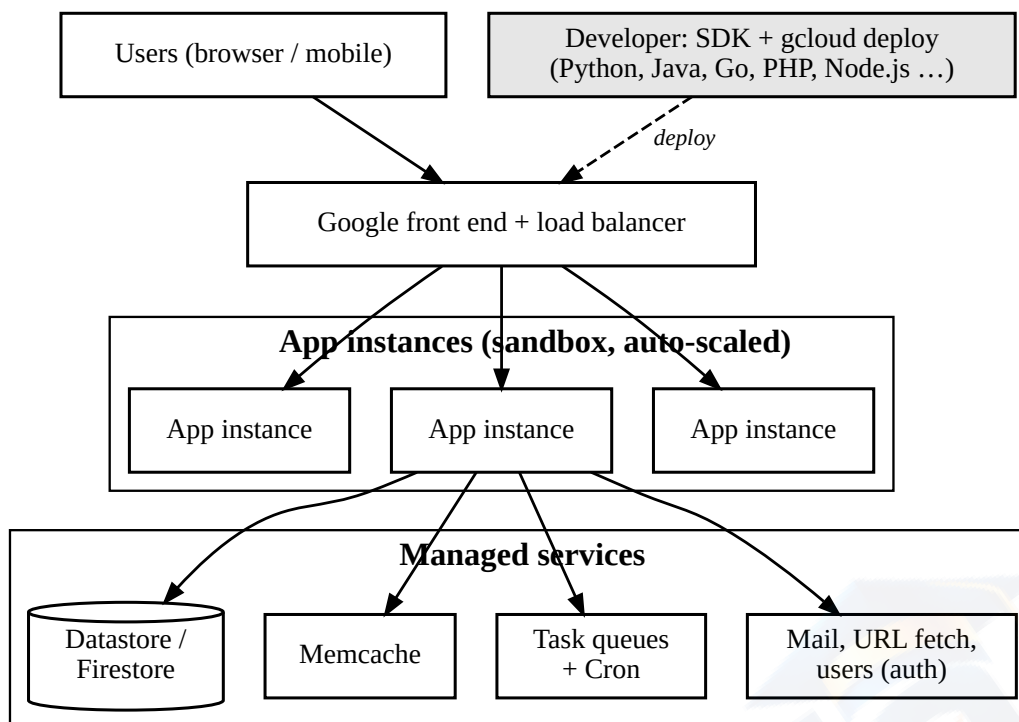


Fig.: Google App Engine (PaaS) architecture

Benefits for developers:

1. **Focus only on code** — no server provisioning, OS patching or capacity planning.
2. **Automatic scaling** — scales from zero to many instances as traffic grows, and back down; built-in load balancing.
3. **Rapid deployment and versioning** — deploy with one command (gcloud app deploy); multiple versions, traffic splitting, easy rollback.
4. **Integrated services** — Datastore / Firestore, Cloud SQL, Memcache, task queues, cron, authentication, logging and monitoring.
5. **Low cost** — pay only for resources used, free daily quota for small apps.
6. **Security and reliability** — runs on Google's infrastructure with sandboxing, firewall and managed SSL.

Main features: popular languages (Python, Java, Go, PHP, Node.js, Ruby, .NET), standard and flexible environments, SDK for local testing, application security scanning, custom domains.

(Any six benefits 4 M, features 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