Unit -1 : Introduction To Cloud computing

* Cloud Computing : Store, Access & process data over Internet.
(x Not locally)

→ No need to buy expensive H/W & S/W, just "rent"
- → Computing power
- → Storage
- → other Resources.

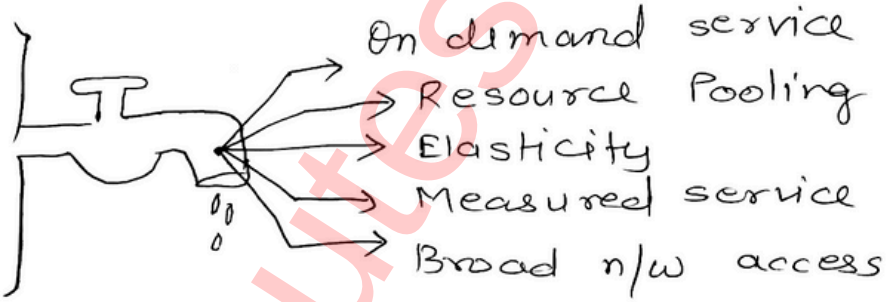
[Using someone else's computer over the Internet].

→ Benefits :
- → Saves money
- → Scalable
- → Accessible anywhor
- → Automatic updates

Eg: ◦ Students using Google drive to store.
◦ Companies using AWS to run websites.
◦ Developeres deploying apps on Azure

* Characteristics of CC:

- On demand self service
  (Use when you want)
- Broad N/w Access
  (Access from anywhere)
- Resource Pooling
  (Shared resources)
- Rapid Elasticity
  (Auto-scale up/down)
- Measured service
  (Pay for usage only)



* History & Evolution of CC:

Mainframe (1960s)
↓
Client-server (1980s)
↓
Grid computing (1990s)
↓
virtualization (2000s)
↓
CC

* Components of CC:

① Client device : used to access Cloud.
[laptops, mobiles]

② Application (SaaS)
↳ Users can use directly through cloud, no need to install or manage.
[Gmail, Google docs, Canva]

③ Platform (PaaS)
↳ Development platform provided by cloud for programmers to build & deploy appln.
[Google App engine, Microsoft Azure App service, AWS Elastic Beanstalk]

④ Infrastructure (IaaS)
↳ provides VMs, servers, storage & n/w.
[Amazon EC2, GCS, Microsoft Azure VM]

⑤ Cloud storage: All data, files & backups are stored in remote server
[Google drive, Dropbox, icloud].

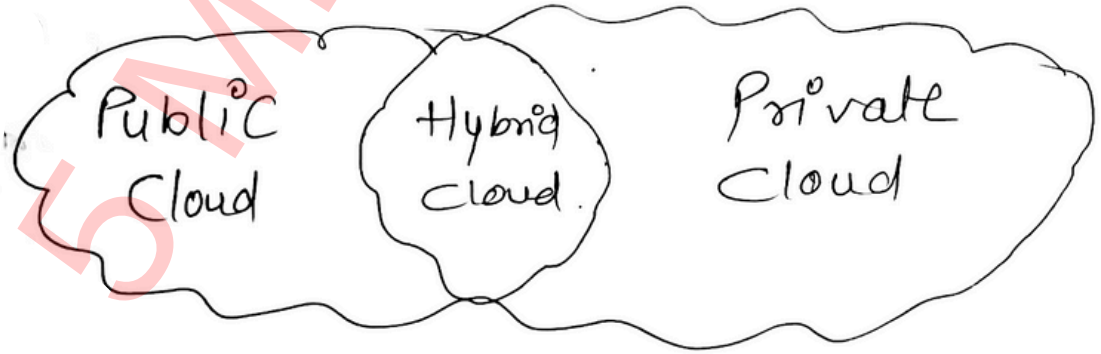
* Cloud computing Layers: (service Models)

SaaS
[On demand]
[End users]
[No Installation]
[No Mgmt]
[Platform Independence]
[Internet required]
[Eg: food ordering Apps]

PaaS
[OS x UI ✓]
[Runtime environment]
[offer development & deployment tools]
[Developers use it]
[More control]
[No need to buy H/W & S/W]
[Eg: No app, Go Hotel, I eat]

IaaS
[OS ✓]
[VMs]
[Storage]
[Provides Infrastructure]
[Enhanced Scalability]
[Max control]
[system administrator]
[I cook]

* Deployment Models:



* Advantages of CC:

→ Scalability: ↑ ↓

→ Cost Saving: Pay only what you use

→ Accessibility: Work from anywhere.

→ Reliability: Automatic backup/update

→ Automation: No installation
No manual maintenance.

eg:- Netflix scales ↑ ↓ automatically..
Google Doc, collaborative work.

* Challenges / limitations of CC:

→ Security & Privacy Issues
"customer info on cloud."

→ Internet Dependency.

→ Limited control

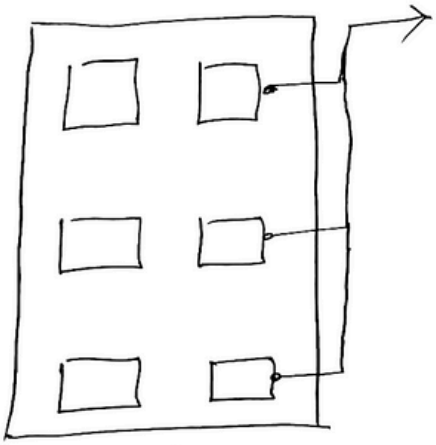
→ Data loss

→ Hard to switch betn clouds.

→ Hidden costs.

→ Technical Complexity.

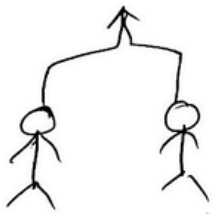
Unit-2 : Virtualization



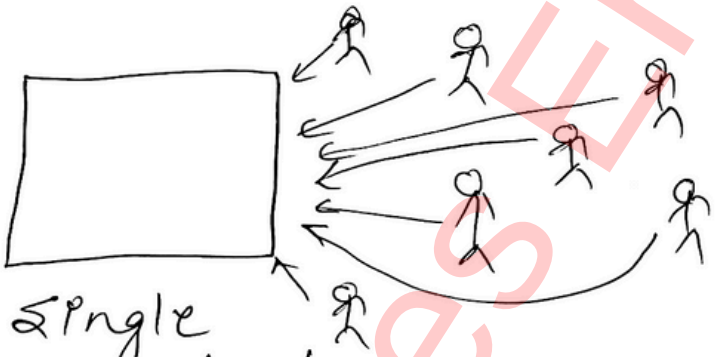
Rooms (VMs)

physical server (Building)

Applications



(Live in same building share wall, water & electricity)



single physical m/c

s/w that makes this possible

↓

"Hypervisor"

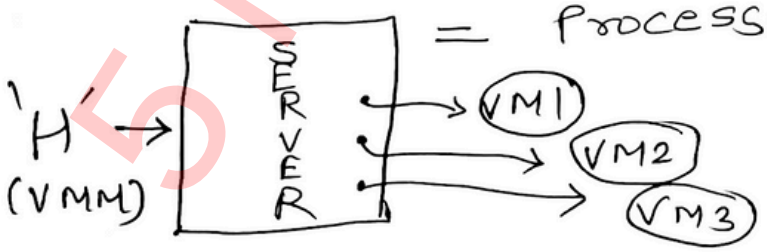
→ Maximizing H/w utilization.

→ Saving cost

→ More flexibility.

→ Simple maintenance.

= Processor + RAM + ROM + N/w



* Types of Virtualization:

- Hardware ✓ (VMware vSphere, Microsoft Hyper V)
- OS ✓ (Docker)
- Server ✓ (VMware ESXi)
- Storage ✓ (SAN, NAS)
  - SAN: storage Area N/w
  - NAS: N/w attached storage
- N/w ✓ (Virtual LAN, SDN)
- Desktop ✓ (VMware Horizon)
- Application ✓ (Microsoft App-V)

* Types of Hypervisors:

(1) Bare Metal-H (Direct Hypervisor)

↳ Run directly on physical H/W (without any OS below it).

eg:- VMware vSphere, Microsoft Hyer-V

→ No middle man landlord manage all apartments himself.

→ faster, more secure, complex to setup.

(2) Hosted Hypervisor

↳ Runs on top of existing OS. (windows, macOS, linux etc)

eg:→ VMware workstation

→ slower, less secure, simple to setup

→ Rents a flat & sublets rooms.

* Benefits of Virtualization:

- Efficient use of H/W
- Cost saving.
- Isolation.
- flexibility
- Portability.

* Challenges of Virtualization:

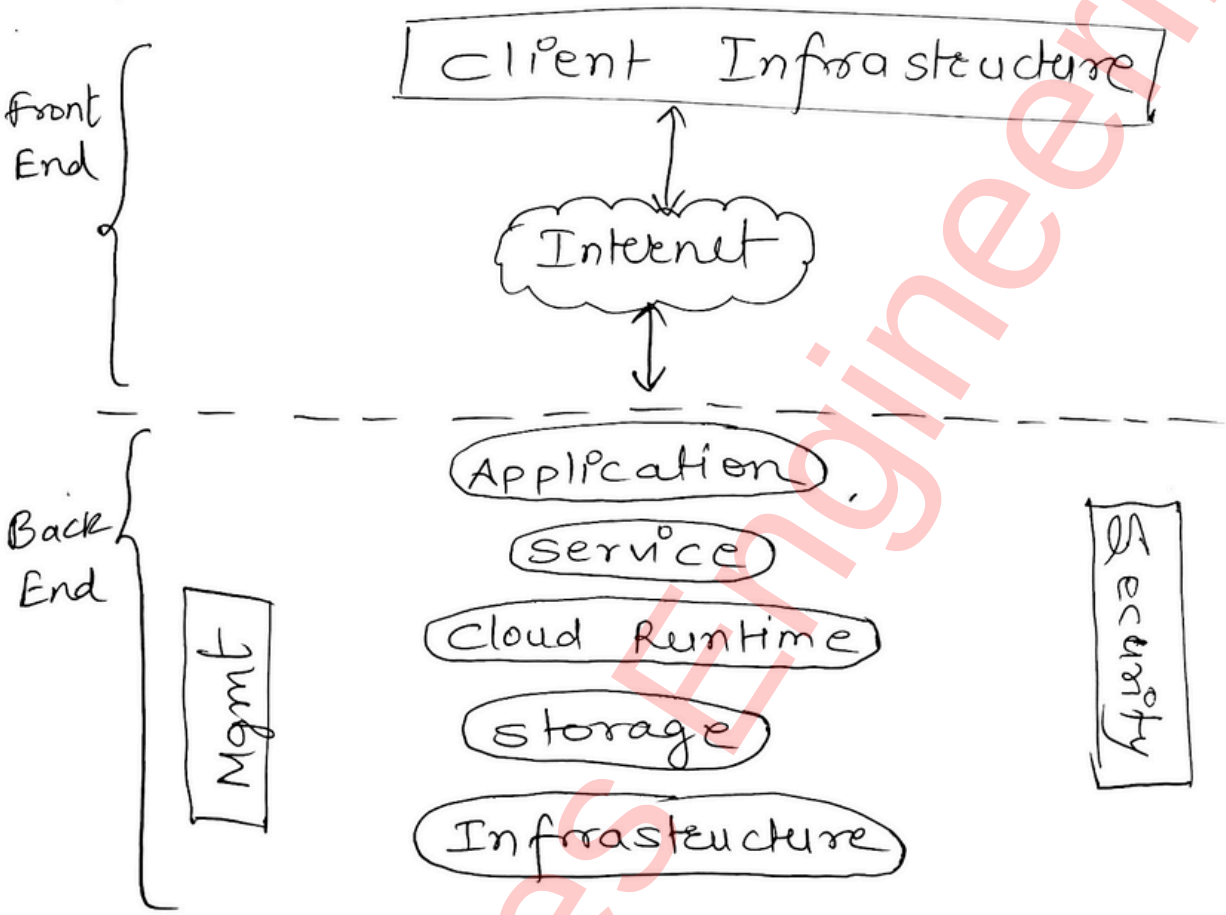
- Performance Overhead (slow down)
- Security Risks
- Single point failure
- Management Complexity
- Compatibility Issues
- N/W bottlenecks

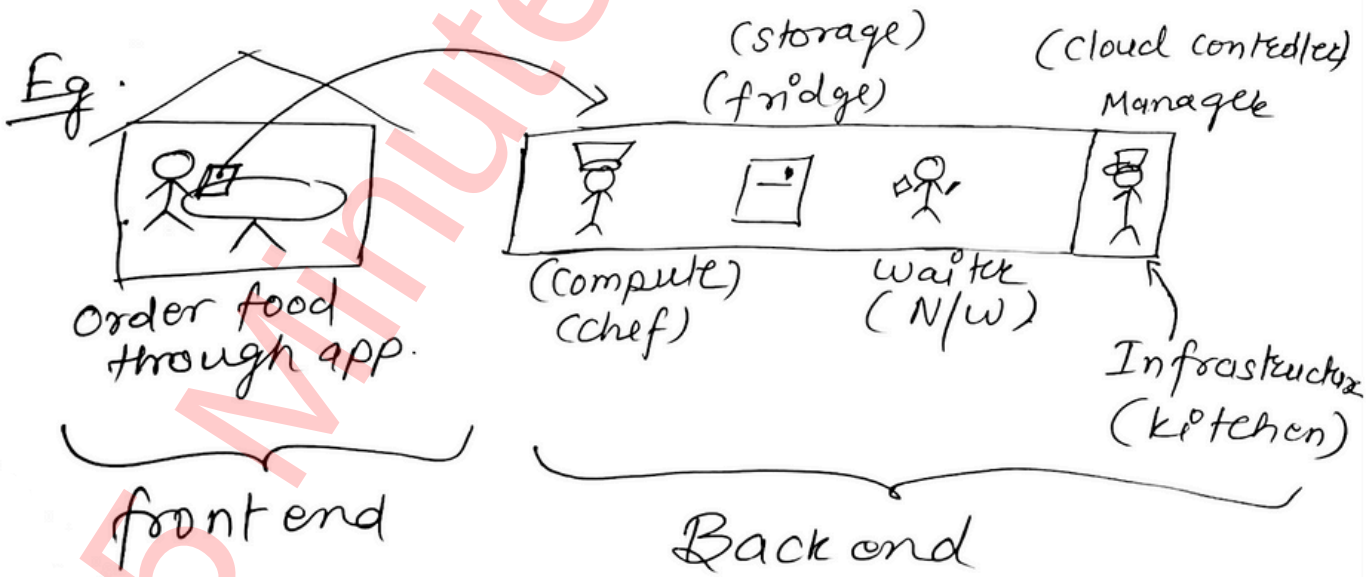
* Live Migration:

- Moving a running VM from one physical host to another without stopping it.

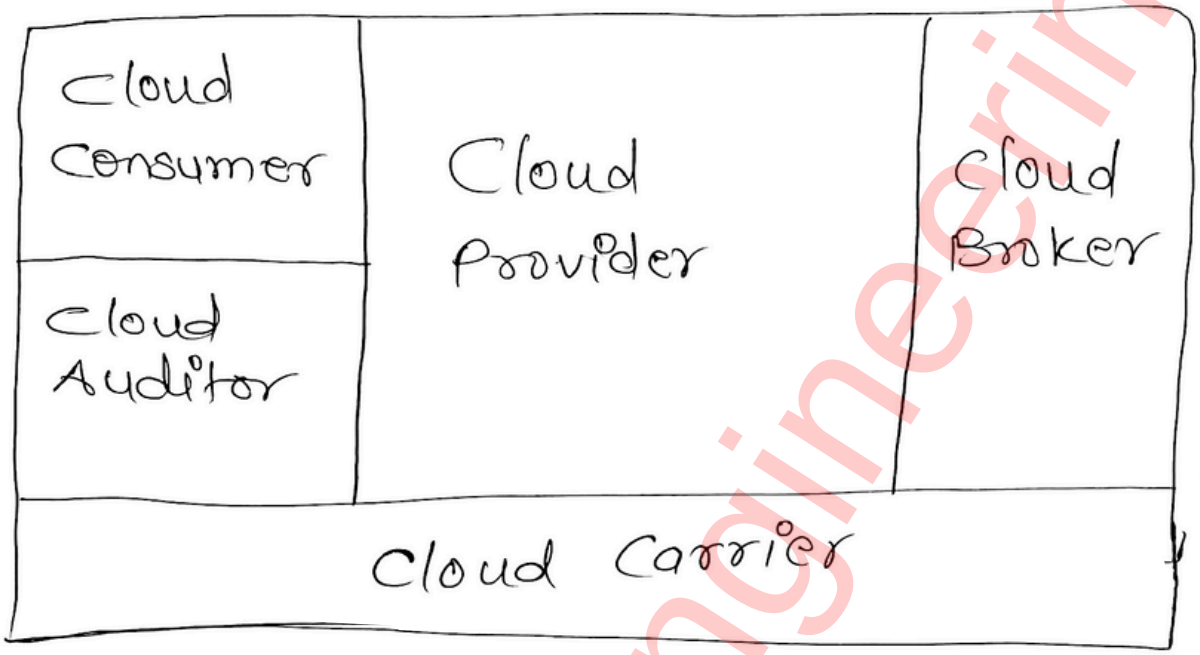
→ To prevent failures / downtime / lags.

Unit-3 : Cloud Architecture Computing





* NIST Cloud Reference Architecture



→ cloud consumer: YOU
→ cloud Provider: AWS, GCP, Azure.
→ Cloud Broker: Middleman
→ cloud Carrier: n/w that connect users to cloud (ISP)
→ cloud Auditor: check performance & security.

Eg: Online shopping:
- You (consumer)
- Amazon (provider)
- Delivery Boy (Carrier)
- Price comparison website (broker)
- Quality control Inspector (auditor).

* Major Cloud service Providers.

① Amazon web services (AWS)

→ Amazon, 2006, 31% global share

→ Services: EC2 (virtual servers [IaaS])
S3 (object storage)
RDS (Database service)
lambda (Serverless computing)

→ Used by: Netflix, Airbnb, NASA, LinkedIn.

② Microsoft Azure

→ Microsoft, 2010, 24% market share.

→ VMs, App services, SQL Database, Azure Active directory, Azure data factory, Azure Synapse Analytics.

→ Used by: Adobe, eBay, samsung, *

③ Google cloud Platform (GCP)

→ Google, 2011,

→ Services: compute engine, App engine, cloud storage, BigQuery.

→ used by: Spotify, PayPal, Twitter.

Unit-4

* Cloud storage systems

↳ Storing your data on remote servers.

- files
- photos
- videos
- docs
- backups

① Upload file from your device.
② file is sent to cloud provider's data centre via internet.
③ The provider stores the file on multiple servers.
④ You can access it anytime, anywhere.

⇒ Types of cloud storage systems

① Object storage:
- → Best for large amount of Unstructured data (images, videos).
- → Each file (object) has unique ID & metadata.
- → Data is stored in 'Buckets'
- → Scales infinitely.
- → eg: Amazon S3, Google cloud storage, Azure blob storage.

② File Storage
↳ Best for structured data, user directories & team collaboration.
→ works like normal file system.
→ eg:- Google drive, dropbox, onedrive.

③ Block Storage
↳ Best for Databases, high performace workloads
→ splits data in fixed size blocks.
→ Each block has unique addr.
→ faster Read/Write.
→ eg:- Azure Disk storage, AWS Elastic Block store (EBS)

* Cloud Networking
↳ Combine all Virtual n/ws, routers, switches, load balancers to connect with all cloud components & make them communicate efficiently.

→ Components
- Virtual N/ws
- Subnets
- Routers
- Load Balancer
- firewall
- VPN
- DNS
- CDN

* Cloud Storage Architecture.

↳ structure/design of how data is stored, manage & accessed in the cloud.

→ 3 Layers

1) Physical Layer [Infrastructure Layer]

→ Data centers, physical servers, Hard drives... where all data physically lives.

→ eg:- shelves in a library.

2) Control Layer [Management Layer]

→ Brain of cloud storage system.

→ Manages storage, replication & retrival, Access control.

→ eg:- librarian of library.

3) Application Layer [User Interface Layer]

→ Top layer: user interact with.

→ Authetication (login), upload/download file sharing & collaboration tools.

→ eg: You come to borrow or return books.

* Data Replication

↳ keeping Multiple copies of same data across different servers or locations (data centers) in Cloud.

(To ensure availability, reliability & fault tolerance, Disaster Recovery & Low Latency).

Types:

→ Synchronous Replication:

→ Data is updated in all replicas at the same time. (strong consistency)

→ Asynchronous Replication

→ Data is first written to primary server & then copied later to other secondary servers. (fast writes)

→ Semi-Synchronous Replication

→ Hybrid: one or two replicas are updated immediately, & rest are updated later.

→ Notify your family first then all friends & relatives.

→ Balance of speed & consistency.

* Data Consistency & Models.

↳ Data Replicated in Multiple places must Agree on the same value.

" All users/everywhere see same data value".

→ Consistency Models (Types)

① Steong Consistency:

↳ Reads give most recent or latest data every time.

→ Always up-to-date.

eg → everyone updates notebook the moment teacher writes.

② Eventual Consistency:

→ Data updates slowly spread to all replicas.

→ Eventually but not immediately.

eg → Rumor spreads eventually.

③ Casual consistency:

↳ Logical order maintained

→ If Ops B depend on A, then all users must see A befor B.

→ Reading chapter 1 before chapter 2.

④ Read-Your-write Consistency:
↳ After you write/update, you always see your own update, even if others don't yet.
→ (Great for user experience but Inconsistent for others briefly).

⑤ Session consistency:
↳ When active/logged in you always see a consistent view of your data, but once you log out, the guarantee ends.

* Scalability & Load Balancing:

Scalability ↓
Grow/shrink resources based on demand

- Types:
  - ↳ Vertical.S (Scale up/down) Add more power (CPU, RAM).
  - → Horizotal.S (Scale out/In) Add more servers to handle more load.

Load Balancing ↓
work is evenly distributed among servers.

Types
- → Static.LB (fixed distribution)
- → Dynamic LB (Adjust dynamically on real-time traffic & performance demand)

* Cloud Resource Management:

↳ Efficiently controlling, allocating, monitoring & optimizing all cloud resources (servers, n/w, application, storage) to ensure smooth performance.

eg: Manager of a Hotel ✓.

• Key Processes in CRM:

① Resource Provisioning:
(Allocating required resources to user)

② Resource Scheduling:
(which resource will do what task & when – to ensure no server overload)

③ Resource Monitoring:
(Tracking how resources are used).

④ Resource optimization:
(Improving resource usage to save cost & boost performance).
By removing idle servers or resizing instances.

Unit-5 Cloud Security, privacy & Trust.

↳ Set of protocols, technologies, policies, practices, & rules designed to protect cloud data, application and infrastructure from threats & unauthorized access.

→ Cloud Security layers

① Physical SL : (Physical data centers & H/W)

② Network SL : (Protects CIA traits of data over n/w)

③ Host SL : (Protects VMs & OS)

④ Application SL : (Protects S/W & APIs)

⑤ Data SL : (Protects data from theft or loss)

⑥ User Access SL : (Manages user identities & permissions)

→ Cloud Security threats / challenges

① Data Breaches
② Account Hacking
③ Insecure APIs
④ Denial of Service
⑤ Data loss / corruption.

Eg Castle protection

→ Walls (firewalls)
→ Gaurds (Authentication)
→ Camera (Monitoring system)
→ Locks (Encryption)
→ Alaram (Intrusion detection)

* Identity & Access Management (IAM)

→ Only the right people can access right resources in cloud.

[Who is allowed to do what on which resource & when]

Eg:- company office:

Identity = Employee ID Card.

Authentication = Scanning your ID to enter.

Authorization = which rooms you can enter.

Audit = CCTV footage tracking your movements.

→ Types of Access control models in IAM

① Discretionary AC = Owner controls who get access

② Mandatory AC = system enforces access rules on individuals strictly.

③ Role Based AC = Access based on job role.

④ Attribute Base AC = Based on user + resource + environment attributes.
(If user = admin & time < 6PM)

* Cloud Privacy: Protecting user's personal & sensitive data/logs stored in cloud from disclosure.

→ More focus is on keeping data

CONFIDENTIAL ✓ (Closing curtains)

SAFE ≈ X (Locking house)

↓ +

Both are required

privacy + security.

(Control & Consent)

* key Principles of Cloud Privacy
  - → Data Ownership
  - → Data Control
  - → Transparency
  - → Consent
  - → Data Minimization
  - → Accountability.

* Trust in cloud computing
  - ↳ confidence users have in cloud provider's ability to secure, manage & respect their data & privacy.

[They won't : Lose it, Leak it, Misuse it]

* Type of Trust in CC
  - ↳ User Trust : I trust Google drive.
  - → Provider Trust : AWS trusts you won't host illegal content
  - → Mutual Trust : Like tenant & landlord both keep their promises.

→ Cloud Trust layers

- Security (Encryption, IAM)
- Privacy (Data control & Consent)
- Reliability (Uptime, Backup)
- Transparency (Logs & No Hidden cost)
- Compliance (ISO, GDPR)
- Accountability (Audit trails).

Unit-6 Cloud Deployment, Monitoring & Performance

→ Cloud Deployment

↳ way cloud resources (servers, storage & app^n) are set up, managed & made available to users.

Types of Cloud Deployment Models:

↳ ① Public cloud: Its shared way. Accessible over internet, Pay as you go pricing.

eg:- AWS, Azure & GCP.

[Living in a apartment building]

② Private cloud : when the cloud infra. is used by a single organization.
→ (Private house) ✓
→ Everything (server, storage, n/w, app^n) is dedicated to one organization.
→ Exclusive access, high level security & control.
→ eg: IBM privite cloud, Openstack.

③ Hybrid cloud = Public + Private
↳ Allowing data & app^n to move betn them.
→ keep sensitive data in private. & less critical workload in public
eg → AWS outposts, Azure stack.
→ 2 homes → separate bungalow.
↳ flat in apartment. (Rented).

④ Community cloud : Infrastructure shared by several organizations that have common goal, policies or interest.
eg: Universities shaing a cloud for research
→ Shared apartment for doctors/defence

→ 5 Phases of Cloud Deployment

① Planning (Define goal / strategy) [why & what]

② Deployment Design (N/w, storage, security) [How]

③ Implementation (Setup & More App^n) [Build]

④ Testing & Integration (Validate) [verify]

⑤ Monitoring & optimization. (Manage, scale) [Improve]

→ Service level Agreement (SLA)

↳ formal contract bet^n provider & Consumer that defines level of service expected:

- Uptime (%) guarantee
- Performance speed.
- support availability.
- Penalties if service fails.

eg:- Mobile plan recharge.
(All offering must be made available)

* Cloud Monitoring
  - observing, tracking & analyzing the operations, performance, & security of cloud systems in real time.

→ Metrics like CPU usage, memory usage & traffic (%) ↑↓.

- Components of CM
  - Infrastructur. M (CPU usage, RAM, uptime)
  - Appln M (Response time, API errors)
  - Database. M (Query speed, replication lag)
  - N/w. M (Latency, Bandwidth usage)
  - Security. M (Unsual login attempts)
  - User Activity. M (IAM tracking & audit logs)

- Steps of CM
  - Data collection
  - Data Aggregation
  - Data Analysis
  - Alerting.
  - Visualization
  - Action/ Auto-Response.

* Performance Metrics in Cloud:

↳ key measurable indicators that show how well cloud system is working.

→ Types:

① Compute Metrics
- → CPU & memory utilization (%)
- → Disk I/O
- → Uptime

② Storage Metrics
- → Storage Latency
- → Throughput (vol. of data transferred per sec).
- → I/O op/sec.
- → Storage capacity (Available Vs used)
- → Data Durability.

③ Network Metrics
- → N/W Latency
- → Bandwidth
- → Packet loss (%)
- → N/W throughput.

④ App^n Metrics
- → Response time
- → Error Rate (%)
- → Uptime (%)
- → Concurrent users.

Unit-7 Advanced Cloud Concepts

① Serverless Computing.

→ Don't own or manage the servers. You just mention what needs to happen, → cloud runs it automatically whenever needed & you only pay when it runs.

eg:- AWS lambda, Azure functions.

② Containers:

↳ small, lightweigh box that carries everything an appln needs like code, tools & libraries.

So that it runs exactly the same everywhere.

eg:- Docker

③ Container Orchestration

↳ 100's of containers, you need a manager to organize, start, stop & balance work & flow.

→ That manager is an orchestrator.

eg:- kubernetes.

④ Edge Computing:

↳ Instead of sending all work to big central cloud server, we do a small, quick processing near the user (at 'edge' of n/w) (IOT devices

⑤ Fog Computing: Its the middle layer bet^n Edge & Cloud.
→ It processes & filters data near the user before sending it to main cloud.
eg:- Cisco IOx, Iot gateways.

CC	Vs	GC
→ Client - server		→ Distributed.
→ CMs		→ DMS
→ Resources are used in centralized manner		→ Resources are used in collaborative manner
→ High Accessibility		→ Low Accessibility
→ Paisa 'Nikal baba		→ Rehne do, bhai hai tu mera.
→ Iaas, Paas, Saas		→ DC, DI.

⑥ AI in Cloud Computing:
↳ using Cloud platforms (AWS, Azure) to create, train & use AI systems, models without needing own Super computer.