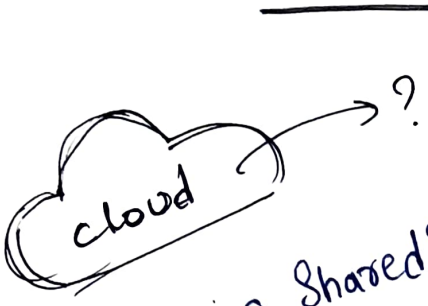
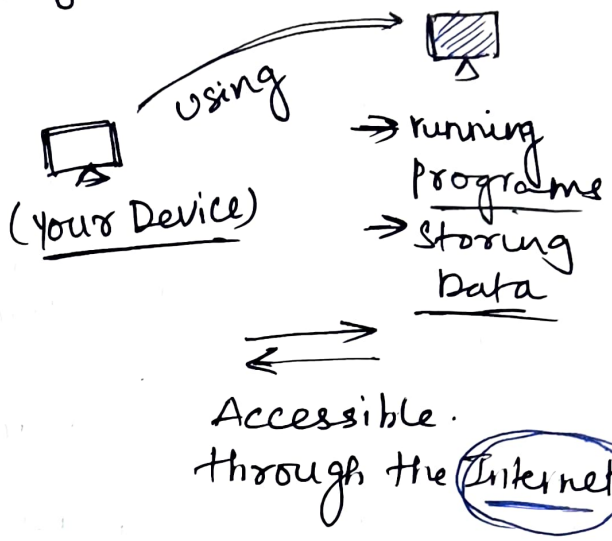
Cloud Computing



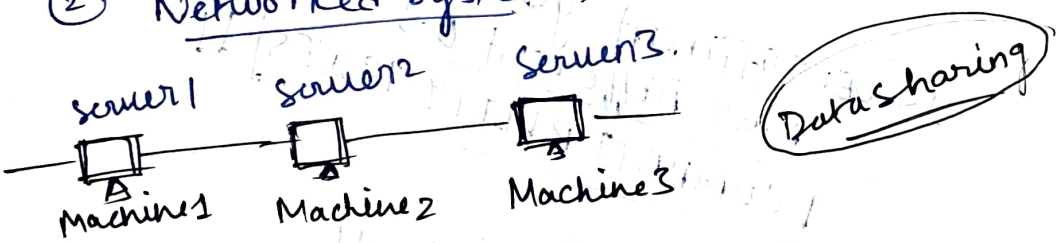
What is getting Shared?

→ Computing Power/ Storage/ Networking/ Software Platforms.

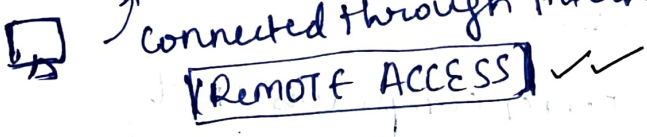


① Single Big Computers :-

② Networked Systems :-



· (work is distributed)



FOUNDATION FOR CLOUD

③ Grid Computing :-

Independent computers connected through Network and solve one very large problem

Grid Computing Architecture:-

Fabric Layer → Actual Hardware.
(Storage, Network, Code, Compute)

Connectivity Layer → Handles Communication & Security.
ex: Grid certificate.

Resource Layer → Managing Individual Machines
(check Availability, schedule tasks, handles Accounting)

Collective Layer → Coordinates Multiple Machines together.
(MDS) Monitoring and Discovery Service)

Application Layer → User interacts.

ISSUE: Diffirent systems have Different O.S
COMPATIBILITY ISSUES !!

Cloud computing → Solution!!

Centralized data-center computers that provide virtual machines, Storage, and platforms over the internet.

1. **Fabric Layer** → Physical data centre.

→ CPU's/GPU's, RAM, Networking switches, Cables, databases etc.

2. Unified Resource Layer → Virtualization ✓

Physical Hardware → Virtual Recourses

→ Virtual Machine
Virtual Storage.
Virtual Networks.
Managed Databases

** Abstraction

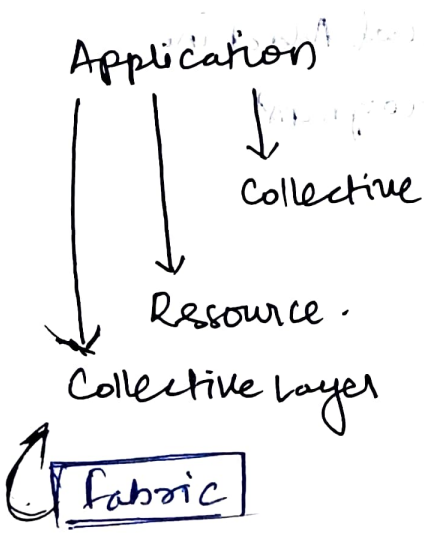
3. Platform Layer → Developer Tools & Services.

provides (runtime environments), web hosting, scheduling systems, Serverless computing

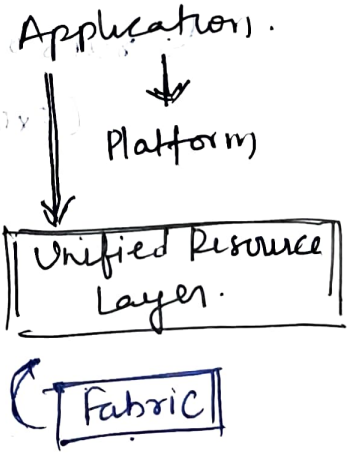
Example: Google App Engine, AWS lamdda, Firebase.

4. Application Layer. → End User Apps

✓ Grid protocol Architecture :-



✓ cloud Architecture



Grid Computing	Cloud Computing
* Custom Hardware	* Commodity Hardware
* Envi.: Library based and customized to H.W	* Environment: Virtualization
* Whole Machine Unit of Resource Allocation	* HW resources fractionally allocated
* Finite allocation of Resources	* Infinite resources available

Understanding Virtual Machine and Runtime Enviroment.

⇒ **Virtual Machine**

Emulates a full computer

- CPU / GPU / RAM / STORAGE
- Networking
- Ability to install any software.
- Operating System.

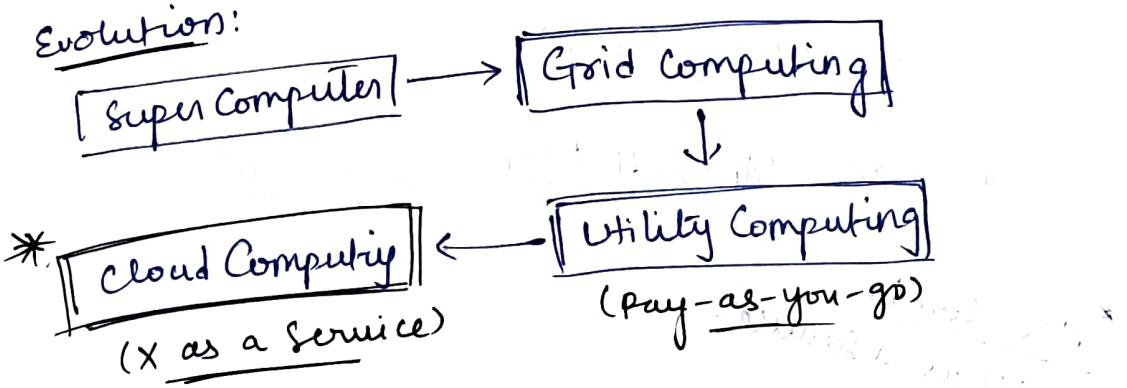
⇒ **Runtime Environment**

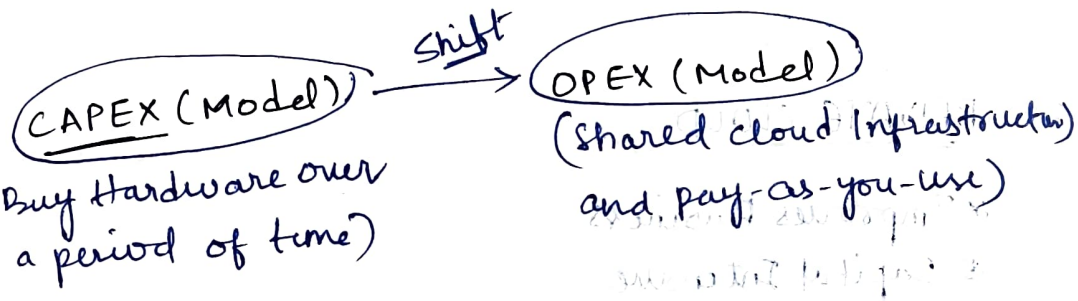
↳ runs on top of Virtual Machine.
(Executes the program)

CLOUD COMPUTING:

A Large Scale distributed pu computing paradigm that is driven by economies of Scale, in which a pool of abstracted, Virtualized, dynamically-scalable, managed computing power, or storage, Platforms, Services are delivered on demand to External Customers over the Internet.

Evolution:





→ Benefits include fair and Efficient Usage of Computational resources.
→ Avoid upfront Infrastructure cost
→ pay-as-you-go.
→ Adjust with fluctuating business demand.

↓

Potential Benefits of OPEX (Model)

Private / Public / Hybrid Cloud

Managed by Single organization Comprising Multiple Consumers

Open use by the general public

1) PUBLIC CLOUD

(Free (or) pay-per-Usage)

✓ Amazon AWS, Microsoft Azure, Google cloud

* Neo clouds.
↳ GPU-as-a service.

2) PRIVATE CLOUD

* Improves Business
* Capital Intensive
* Require Data Centers, Space,

3) HYBRID CLOUD

Aggregation of various of cloud services

Company → Private cloud.
→ Public cloud.

CLOUD BURSTING :-

(Hybrid Cloud Strategy)

Private cloud → during peak → Public cloud

Temporary !!

Hybrid cloud.

↓

Directedted Interconnects

AWS Direct Connect.

Microsoft Azure Express Route.

↳ for low-latency Hybrid AI workflows often Bypassing the public internet entirely for security

(High Bandwidth Connectivity)

↳ Essential for large data transfers, Real time Analytics.

★★ Public cloud (Dir. Inter) → Private cloud.

Bypass public Internet

---

the Concept of Containers :-

(Package an application together with everything it needs to run) → Code, runtime, libraries, settings

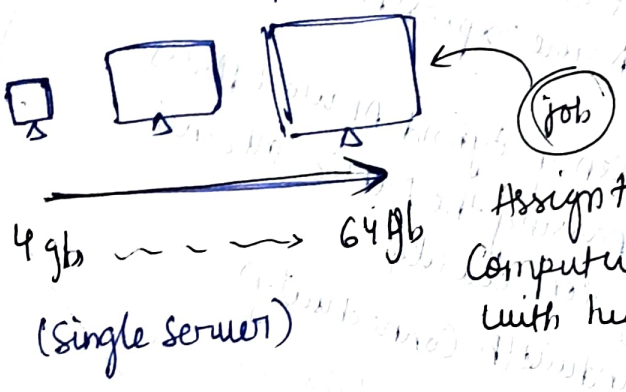
Fast Deployment

⇒ Elasticity and Provisioning

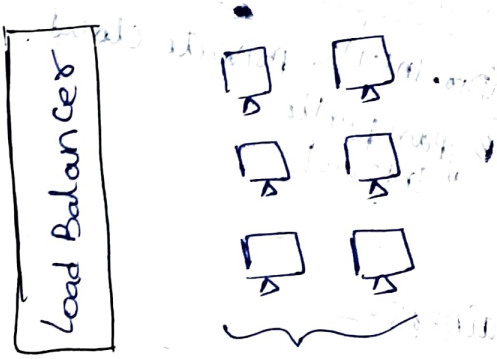
How Systems power (computational) increase when demand grows.

① Vertical Scale-UP :-

Keeps on adding resources to a unit to increase Computational Power.



② Horizontal Scale-out:-



(Add More Servers and Share the work)

Cloud Computing mostly uses Horizontal Scale-out because its cheaper, flexible. etc.

→ Cloud Computing Computation Model.

* hide System-Level details from the developers (cloud will look after it)
* Seperating what from how?
  (developer need to specify the computation that need to be performed)

Resource Management & provisioning:

1. Consumer self-provisioning
   pay as per the usage for cloud services directly to provider
2. Advanced provisioning
   Pay in Advance for resources and services
3. Dynamic provisioning
   provider allocates resources based on consumer usage.

Avoid Over-Provisioning → wastes resources (useless)
Under-Provisioning → Low performance

DATA CENTRE — Secure, specalized physical facility or building that houses IT Infrastructure — such as Servers, Storage Systems, and Networking Equipment.

① On Premises → we own and Manage Everything
(No cloud)

Servers + storage + O.S + Runtime etc.

② IaaS → Infrastructure as a service. ex:
(cloud provides Virtual Machine) → AWS EC2
Storge / Network. → Google cloud Compute Engine

→ we need to handle,
OS setup, libraries, runtime, application

③ CaaS → Container as a service
(we need to package our app into a container)

(cloud Manages servers, scaling, network, Container orchestration).

→ we need to manage.
Container image.
↳ Contains app, runtime, dependencies

ex: → AWS EKS.
→ Google Kubernetes Engine

④ Paas → Platform-as-a-service.
(we need to just upload code)

cloud handles,
Servers, OS, runtime, Scaling, deployment

ex: → AWS Elastic Beanstalk
Google App Engine

(5) Function-as-a-Service (FaaS)

(We need to write small functions, cloud runs them only when needed) → Function Logic

cloud provides:
Servers, runtime, scaling, Execution

Ex: AWS Lambda, Google Cloud Function

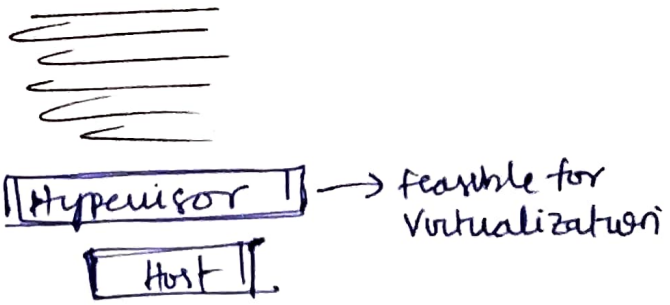
(6) SaaS (Software-as-a-Service)

Cloud provides Everything.

ex: GMail, Google docs, Notion etc

Virtualization:

Virtualization in cloud computing is the foundational technology that allows a single physical server to be split into multiple simulated Environments Called VM's (Virtual Machines) Using software called Hypervisors.



(1) Type1 Hypervisors :- Bare Metal Hypervisors
(Mircosoft-Hyper-V, Open Source KVM)

(2) Type2 Hypervisors :- OS hosted on the Server.
(oracle, Virtual Box, VMware workstation)

Midsem Prep

→ Researches upload a new paper
→ The System understands text meaning + Citation network structure
→ It recommends related papers.
→ It detects trending topics over time (temporal awareness)
→ Publishers can add papers and expand the Citation graph.

Cora citation graph. → add to cloud Storage
(Node features, Paper Metadata)
AWS S3

→ User uploads PDF
(System extracts abstract text).
→ Text Embedding (Sentence Tranformer)

→ Compute Cosine Similarity between query Embedding and stored paper Embeddings
(select top-K Similar nodes)

→ Take top K Similar papers.
(Expand using citations links) 1-hop or 2-hop
Creates a local Subgraph

GNN processing → (cloud) AWS EC2

New Node Embedding = node features +
aggregation of neighbor nodes

(Infrastructure as a service)

Temporal Awareness

Computing Similarity Score:

α. text similarity * β. GNN similarity +
γ. temporal score.

⇒ temporal score = $e^{-(\text{current year} - (\text{year}))}$