This chapter presents some of the representative Cloud [illegible] ture-as-a-Service (IaaS) and Platform-as-a-Service (PaaS) services in market. It provides some insights and practical issues around the architecture of the major Cloud computing technologies and their service offerings.

9.1 AMAZON WEB SERVICES

Amazon Web Services (AWS) is a platform allowing the development of flexible applications by providing solutions for elastic infrastructure scalability, messaging, and file and data storage. The platform is accessible through SOAP or RESTful Web service interfaces and provides a Web based console where users can administrate and monitor the resources required as well as their expenses computed on a pay as you go basis.

Table 9.1 Some Examples of Cloud-Computing Offerings

Vendor / Product	Service Type	Description
Amazon Web Services	IaaS, PaaS, SaaS	Amazon Web Services (AWS) is a collection of Web services providing developers with compute, storage, and more advanced services. AWS is mostly popular for IaaS services and primarily for its elastic compute service EC2.

(*Continued*)

a specific image. Images come with already preinstalled operating system and a software stack, and instances can be configured for what regards memory, number of processors, and storage. Users are provided with credentials to remotely access the instance and further configure or install software, if needed.

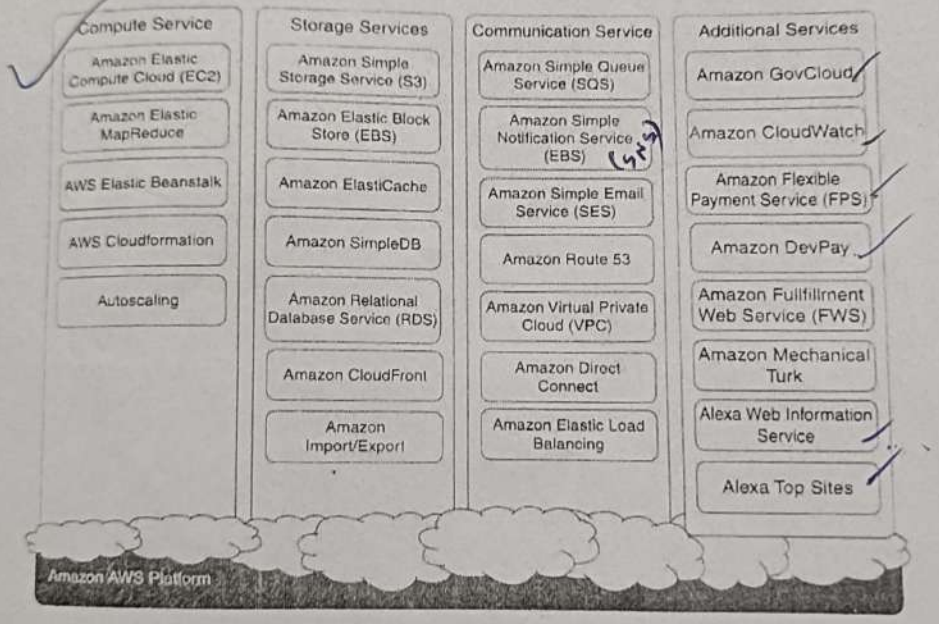


Fig. 9.1. Amazon Web Services Ecosystem.

1. Amazon Machine Image (AMI)

AMIs are templates from which it is possible to create a virtual machine. They are stored into the Amazon S3 and identified by a unique identifier in the form of *ami-xxxxxx* and a manifest XML file. An AMI contains a physical file system layout with a predefined operating system installed. These two are specified by the *Amazon Ramdisk Image* (*ARI*, id: *ari-yyyyyy*) and the *Amazon Kernel Image* (*AKI*, id: *aki-zzzzzz*), which are part of the configuration of the template. AMIs are either created from scratch or "bundled" from existing EC2 instances running. A common practice is to prepare new AMIs to create an instance from a pre-existing AMI, log into it once it is booted and running, and install all the software needed and, by using the tools provided by Amazon, the instance is converted into a new image. Once an AMI is created, it is stored into an S3 bucket and the user can decide whether to make it available to other users or keep it for personal use. Finally, it is also possible to associate a product code with a given AMI, thus allowing the owner of the AMI to get revenue every time this AMI is used to create EC2 instances.

2. EC2 Instance

EC2 instances represent virtual machines. They are created by using AMI as templates, which are specialized by selecting the number of cores, their computing power, and the installed memory. The

Google App Engine (GAE) is a Platform as a Service (PaaS) offering from Google Cloud that allows developers to build and deploy applications without having to manage the underlying infrastructure.

Key Features:

- Automatic scaling: GAE automatically adjusts the number of instances based on traffic.
- Supports multiple languages: Such as Python, Java, Node.js, Go, PHP, Ruby, and .NET.
- Fully managed: Google handles server provisioning, load balancing, patching, and scaling.
- Integrated services: Easy integration with other Google Cloud services like Cloud Datastore, Cloud SQL, and Cl torage.

- Integrated services: Easy integration with other Google Cloud services like Cloud Datastore, Cloud SQL, and Cloud Storage.
- Flexible environments:
  - Standard Environment: Uses sandboxed runtime environments and is optimized for quick scaling.
  - Flexible Environment: Runs on custom Docker containers and supports more control over the environment.

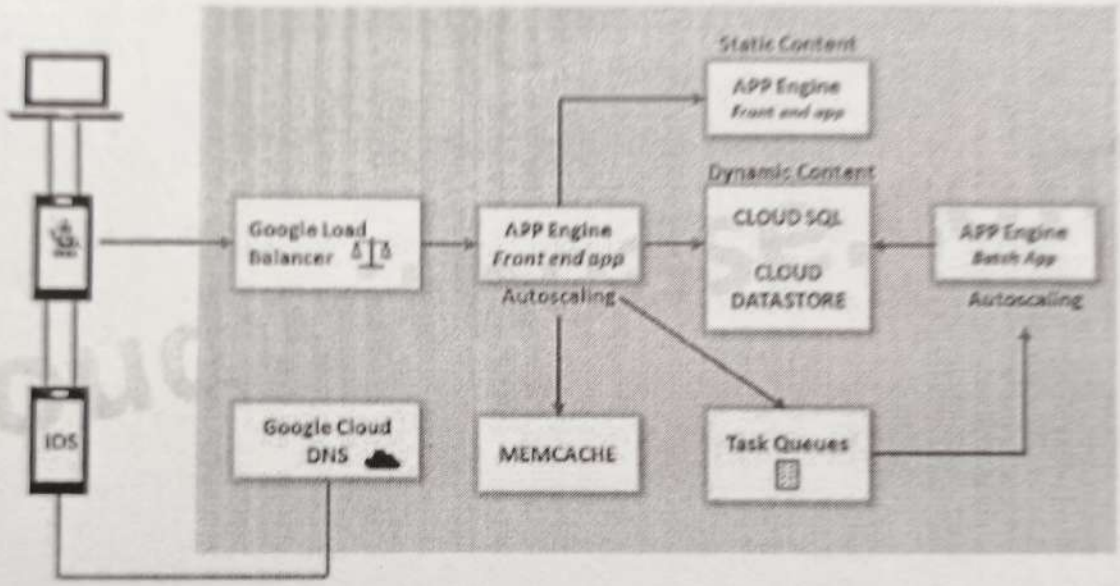
Use Cases:

- Web and mobile app backends
- APIs and microservices
- Event-driven applications

Would you like a simple example of how an app runs

Cloudwithease

Google App Engine Architecture



Google App Engine vs Google Compute Engine ...

Visit ›

Images may be subject to copyright. Learn More

Share

Save

with distributed, Scalable Cloud architec
-tures. ②

23/4/25

- Virtualization

Virtualization is a large umbrella of technologies & concepts that are meant to Provide an abstract environment — whether virtual hardware / operating system — to run applications. It is one of the fundamental components of Cloud computing, specialy in case of infrastruct. based services.

virtualization technologies have gained a renewed interest by the community.

Reasons

1. Increase performence & computing capacity

Now a days super computers can Provide immense computing Power that can accomodate the execution of ~~tools~~ 100's/1000's machines.

2. Underutilize of hardware / Software resources

H/W & S/W underutilization is occuring due to -

1. increase Performence / Computing capacity

2) Effect of limited / sporadic use of resources.
[3] Computers today are so powerful that in most of the cases only a fraction of their capacity is used by an application / system.] ||

ii) Lack of space
The continuous need for additional capacity like storage / computing powers makes data centers grow quickly. [Companies like google / microsoft expand their infrastruct by building data centers that are able to host thousands of nodes.] ||

iv) Greening initiatives [It aims to reduce energy consumption & env. impact by running multiple virtual machines on fewer physical servers.] [This improves resource efficiency, lowers power & cooling needs & reduces e-waste, supporting sustainable & eco-friendly IT operations.

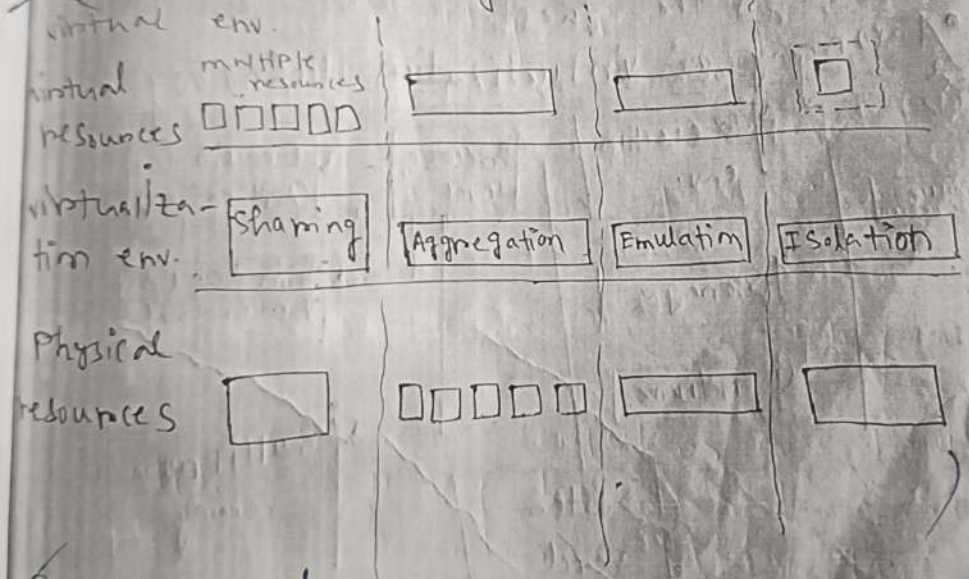
v) Rise of administrative cost [virthalization can raise admin costs because it needs skilled staff & special tools.] [managing many virtual systems can be more complex.] ||

Characteristics of virtual environments

1) Increase security - (virtual env's increase security by keeping systems seperate, making backups easy & controlling access better)

2) manage execution - sharing
(sharing of resources, aggregation of resources, emultion, isolation).

- simulate - Executing the plan from vocal to virtual env.



- Emulation - (Guests are ~~ext~~ executed within an env. that is controlled by the virtualization layer which is a program that ~~is allowing for~~ controlling & ~~tuning~~ tuning the env. ~~that~~ which is exposed to guest)

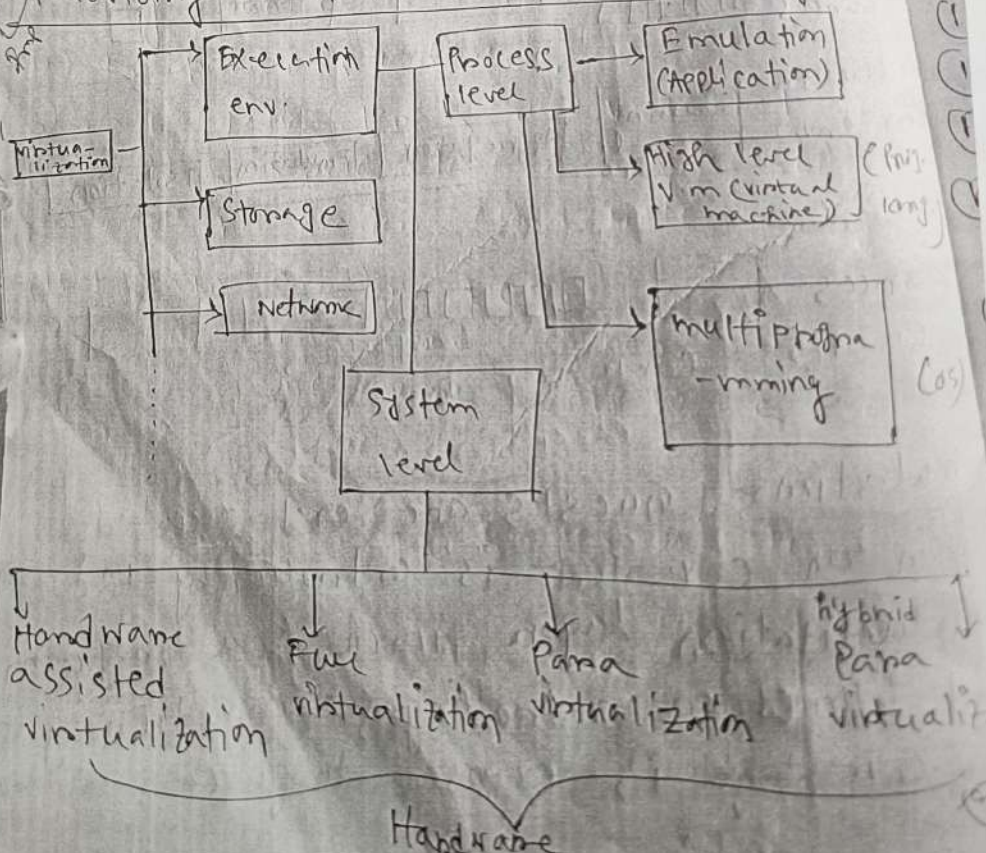
[For instance, a complete diff. env. with respect to the host can be emu -lated for allowing the execution of guest ~~requirement~~ requiring specific chara characteristics that are not present in the physical machine.] ||

- Isolation

[virtualization allows guest, whether their applications/ other entities with a complete seperate env. in which they are executed.

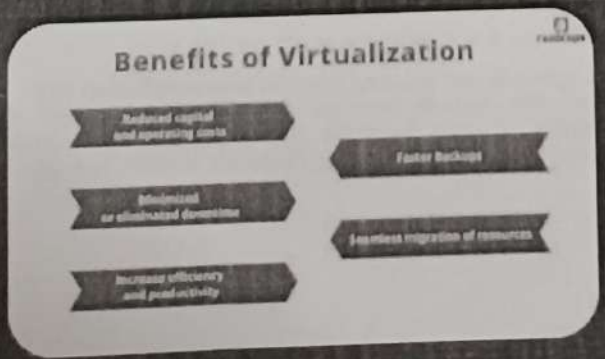
Secondly it provides seperation b/w the host & the guest.]

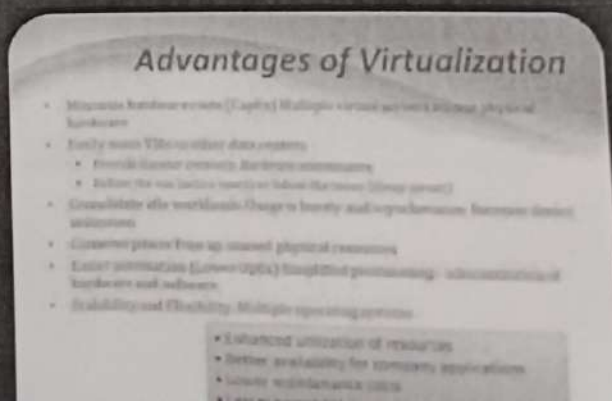
- Taxonomy of virtualization technology



Benefits of Virtualization

- Encapsulation - VMs can be described in a file
  - Possible to 'snapshot'
  - Easy to move
- Enables running multiple operating systems
- Consolidation & use of unused computation power
- Resource management
- High availability & disaster recovery
- Create "Base Environment"
- Safe testing of new software
- Easy Management





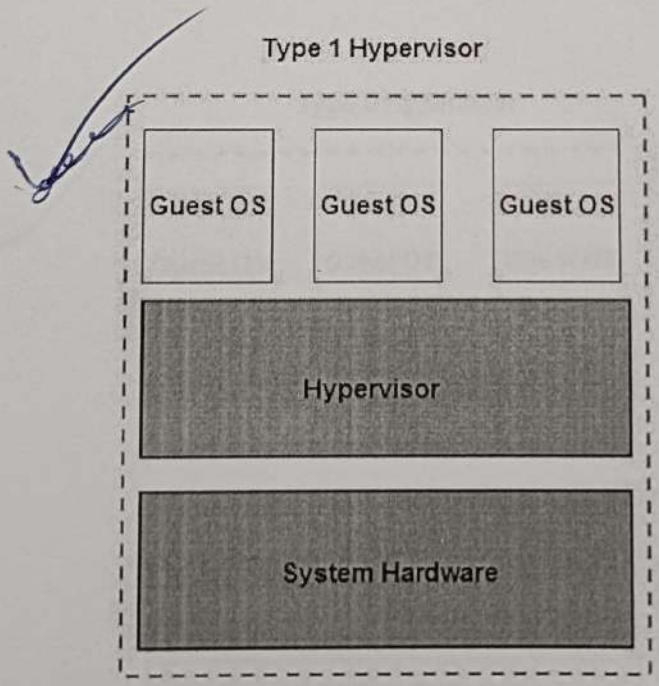
Virtual Machines provide an environment that is logically separated from the underlying hardware.

The machine on which the virtual machine is created is known as **host machine** and **virtual machine**is referred as a **guest machine**. This virtual machine is managed by a software or firmware, which is known as **hypervisor**.

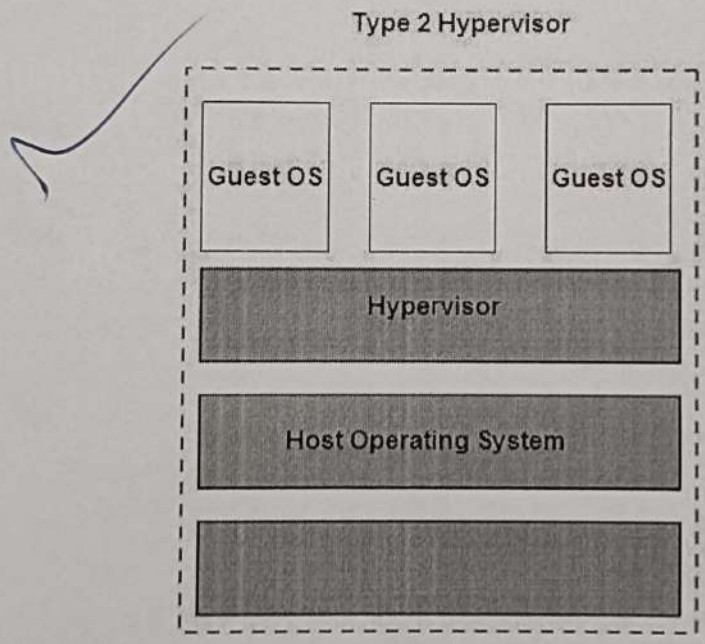
HYPERVISOR

Hypervisor is a firmware or low-level program that acts as a Virtual Machine Manager. There are two types of hypervisor:

Type 1 hypervisor runs on bare system. **LynxSecure, RTS Hypervisor, Oracle VM, Sun xVM Server, VirtualLogic VLX** are examples of Type 1 hypervisor. The following diagram shows the Type 1 hypervisor.



The type1 hypervisor does not have any host operating system because they are installed on a bare system. **Type 2 hypervisor** is a software interface that emulates the devices with which a system normally interacts. **Containers, KVM, Microsoft Hyper V, VMWare Fusion, Virtual Server 2005 R2, Windows Virtual PC** and **VMWare workstation 6.0** are examples of Type 2 hypervisor. The following diagram shows the Type 2 hypervisor.



Types of Hardware Virtualization

Here are the three types of hardware virtualization:

1. Full Virtualization
2. Emulation Virtualization
3. Paravirtualization

FULL VIRTUALIZATION

In **Full Virtualization**, the underlying hardware is completely simulated. Guest software does not require any modification to run.

chatgpt.com

ChatGPT

hypervisor clustering architecture in cloud computing

Hypervisor Clustering Architecture in Cloud Computing refers to the use of multiple hypervisors (virtual machine monitors) working together in a coordinated cluster to provide high availability, scalability, and resource management for virtualized environments. Here's a breakdown of the architecture and its components:

1. Key Concepts

- Hypervisor: Software that creates and runs

Ask anything

2. Components of Hypervisor Clustering Architecture

1. Clustered Hypervisors:
   - Multiple physical hosts running hypervisors.
   - Connected via a high-speed network.
2. Centralized Management Server:
   - Manages and monitors the cluster (e.g., VMware vCenter, Microsoft SCVMM).
   - Handles VM migration, load balancing, and policy enforcement.
3. Shared Storage:
   - Network-attached or SAN storage accessible by all hypervisors.
   - Enables VM mi[illegible]tion and data redundancy.

redundancy.

4. VMs and Workloads:
   - Deployed across the hypervisor cluster.
   - Can move between hosts using technologies like Live Migration or vMotion.
5. Heartbeat and Failover Mechanism:
   - Ensures health monitoring of hypervisors.
   - Automatically fails over VMs if a host goes down.

3. Benefits

- High Availability: If one hypervisor fails, VMs can automatically restart on another.

Host = Hypervisor

3. Benefits

- High Availability: If one hypervisor fails, VMs can automatically restart on another.
- Scalability: Easy to add more hosts to the cluster.
- Load Balancing: Even distribution of workloads across hosts.
- Maintenance Without Downtime: Live migration allows updates and maintenance with minimal service interruption.

4. Example Use Case

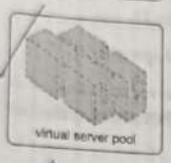
VMware vSphere HA Cluster:

11.2 Resource Pooling Architecture

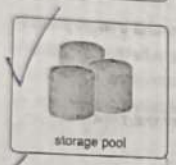
A *resource pooling architecture* is based on the use of one or more resource pools, in which identical IT resources are grouped and maintained by a system that automatically ensures that they remain synchronized.

Provided here are common examples of resource pools:

Physical server pools are composed of networked servers that have been installed with operating systems and other necessary programs and/or applications and are ready for immediate use.

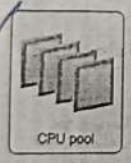


Virtual server pools are usually configured using one of several available templates chosen by the cloud consumer during provisioning. For example, a cloud consumer can set up a pool of mid-tier Windows servers with 4 GB of RAM or a pool of low-tier Ubuntu servers with 2 GB of RAM.

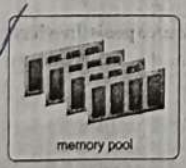


Storage pools, or cloud storage device pools, consist of file-based or block-based storage structures that contain empty and/or filled cloud storage devices.

Network pools (or interconnect pools) are composed of different preconfigured network connectivity devices. For example, a pool of virtual firewall devices or physical network switches can be created for redundant connectivity, load balancing, or link aggregation.



CPU pools are ready to be allocated to virtual servers, and are typically broken down into individual processing cores.



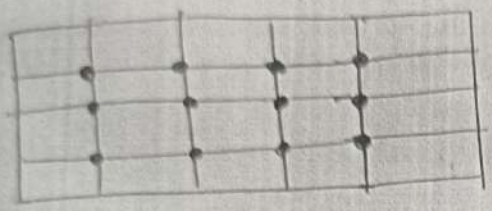
Pools of physical RAM can be used in newly provisioned physical servers or to vertically scale physical servers.

Dedicated pools can be created for each type of IT resource and individual pools can be grouped into a larger pool, in which case each individual pool becomes a sub-pool (Figure 11.2).

Disadv -
Redundancy & failure.

- Grid Computing

Grid Computing provides a platform in which computing resources are organised into one or more logical pools.



These pools are ~~collecting~~ collectively coordinated to provide high performance distributed grids. It is also called super virtual computer.

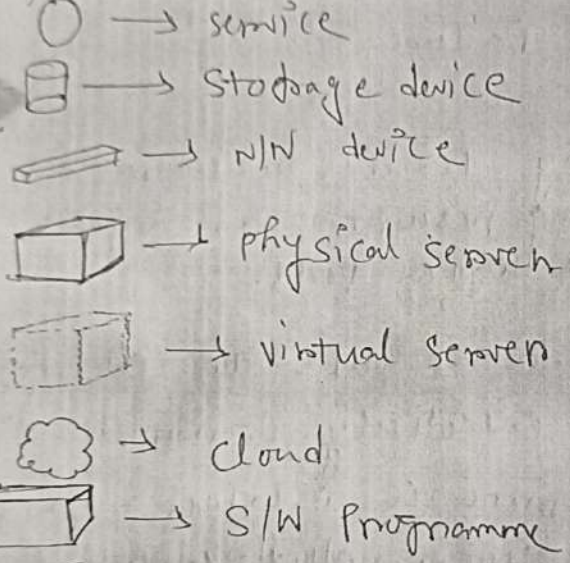
- Cloud

A cloud refers to distinct IT environment that is designed for the purpose of remotely Provisioning scalable & major IT resources.

The term originated from the visual rep of the internet infrastructure in early n/w diagrams.)

Components

IT resources (An IT resource is a physical or virtual IT related artifact that can be either software (such as virtual or custom software) or hardware based (such as physical servers or a physical device).



On-Premise:

(An IT resource that is hosted in a conventional IT enterprise within an organisational boundary is considered to be located on the premises of the IT enterprise as on-premise.)

Cloud consumers V/S cloud providers

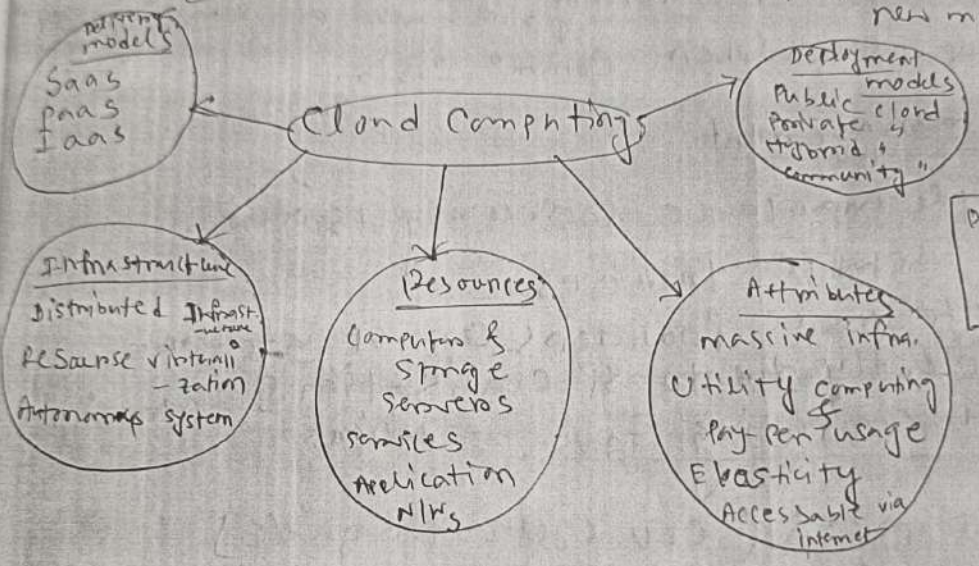
(The party that provides cloud based

IT anfine sources is the cloud providers, the party that uses cloud services that are called cloud consumers)

27/2/25

- Cloud computing models, resources & attributes

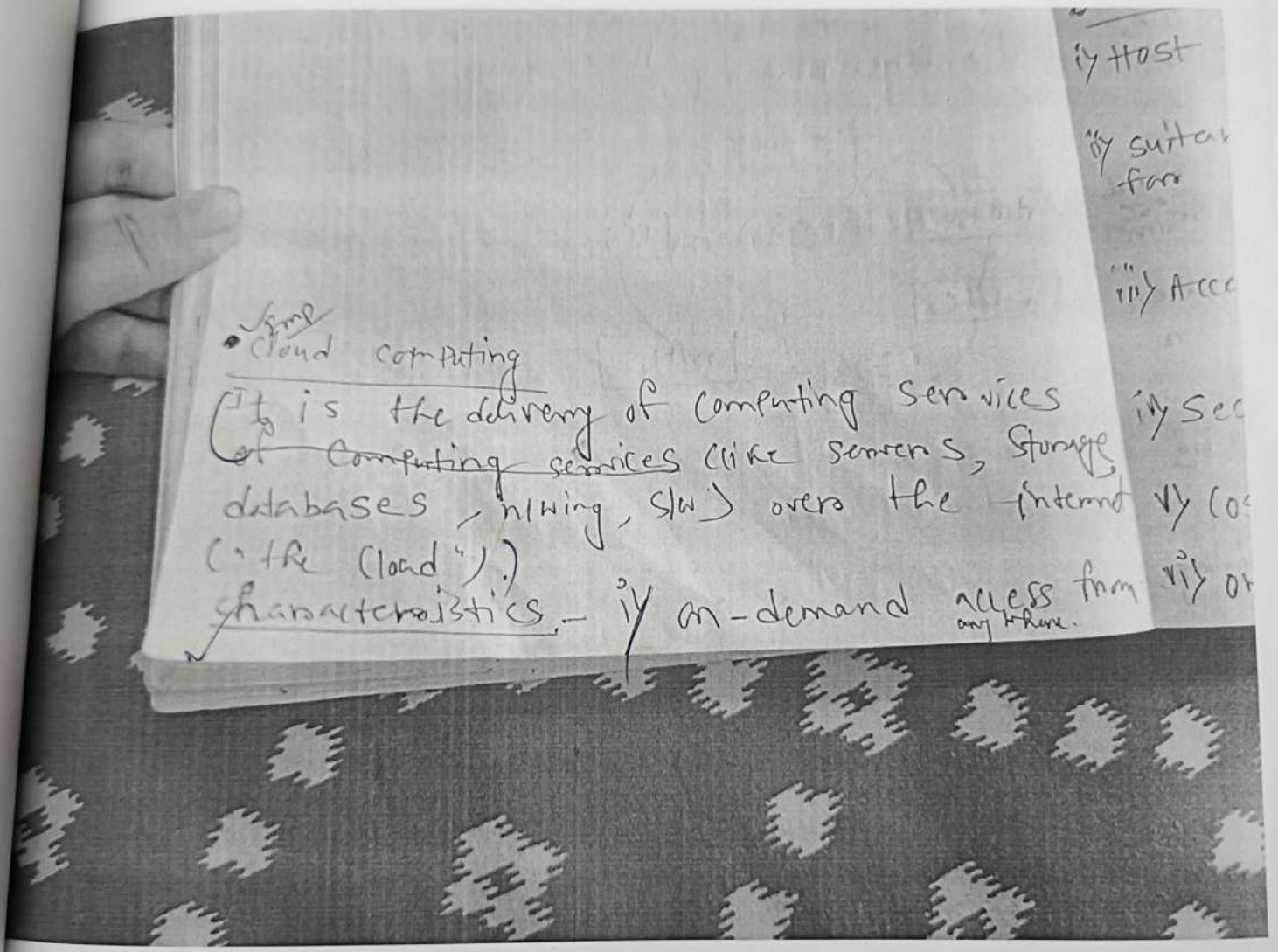
Here there are 3 models (they are old models, nowadays there are many new models)



- Public cloud ≈ Saas / Paas / Iaas (can be any)
- Resource virtualization - Google servers (google drive)
- Autonomous system - weather forecasting
- Cloud computing - characteristics (1-7-0) not this

1) shared resources & resource management

1) It uses shared pool of resources
2) Scalable & elastic
3) Elastic computing - The ability of dynamically & on demand acquiring computing resources & supporting



• Cloud computing

It is the delivery of computing services (like servers, storage, databases, n/w ing, s/w) over the internet ("the cloud").

characteristics - i) on-demand access from any where.

ii) host

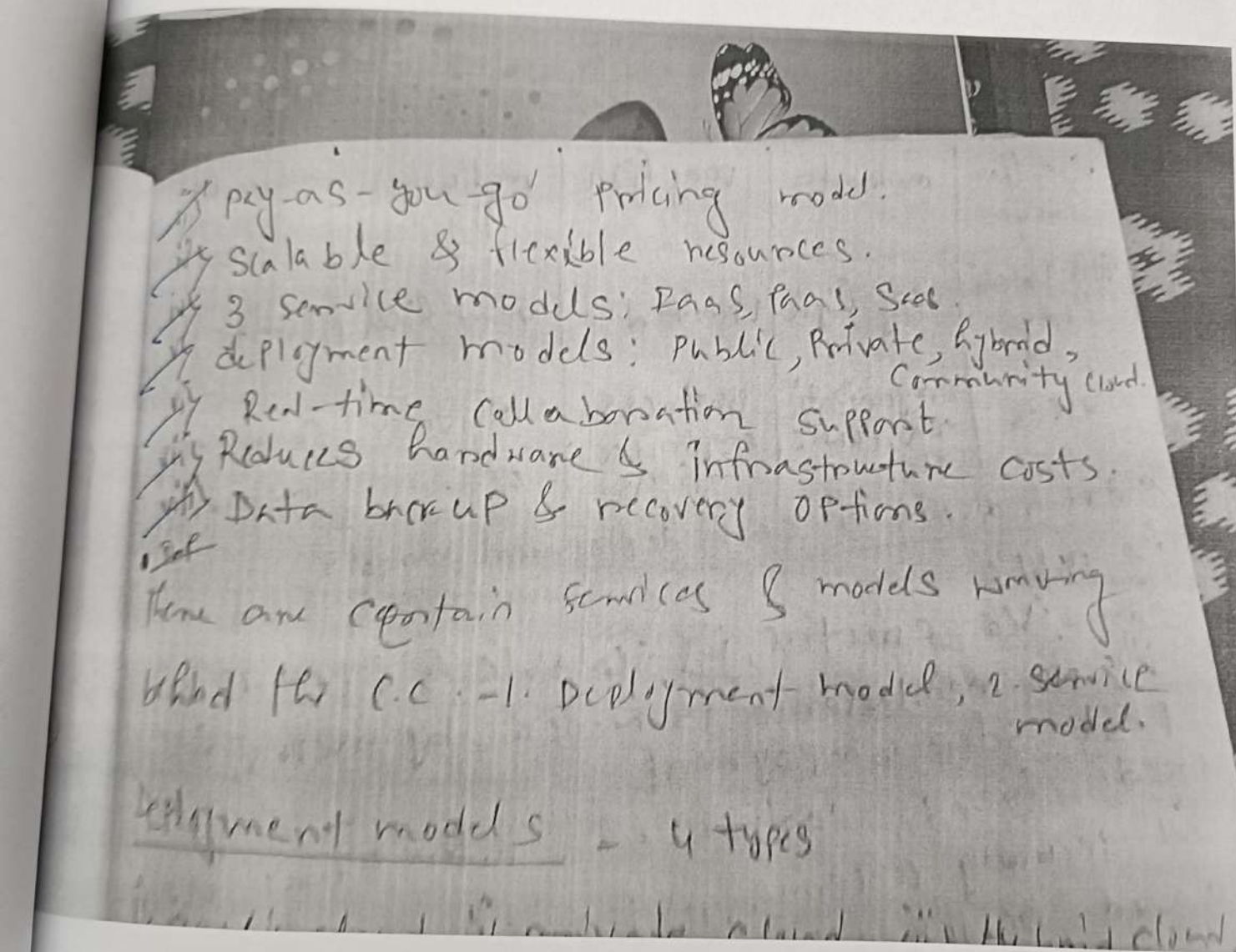
iii) suita for

iv) Acc

i) Se

ii) Co

iii) O



i) pay-as-you-go pricing model.
ii) Scalable & flexible resources.
iii) 3 service models: IaaS, PaaS, SaaS.
iv) deployment models: Public, Private, hybrid, Community cloud.
v) Real-time collaboration support.
vi) Reduces hardware & infrastructure costs.
vii) Data backup & recovery options.

Ref
There are certain services & models working behind the C.C -1. Deployment model, 2. Service model.

Deployment models - 4 types

iii) management.

a) the maintenance & security is operated by service providers.

b) the service providers can operate more efficiently due to specialisation & centralization.

Cloud computing advs-

i) Resources (CPU cycles, storage etc) are shared.

ii) multiplexing leads to a higher resource utilization (when multiple users/applications share a system)

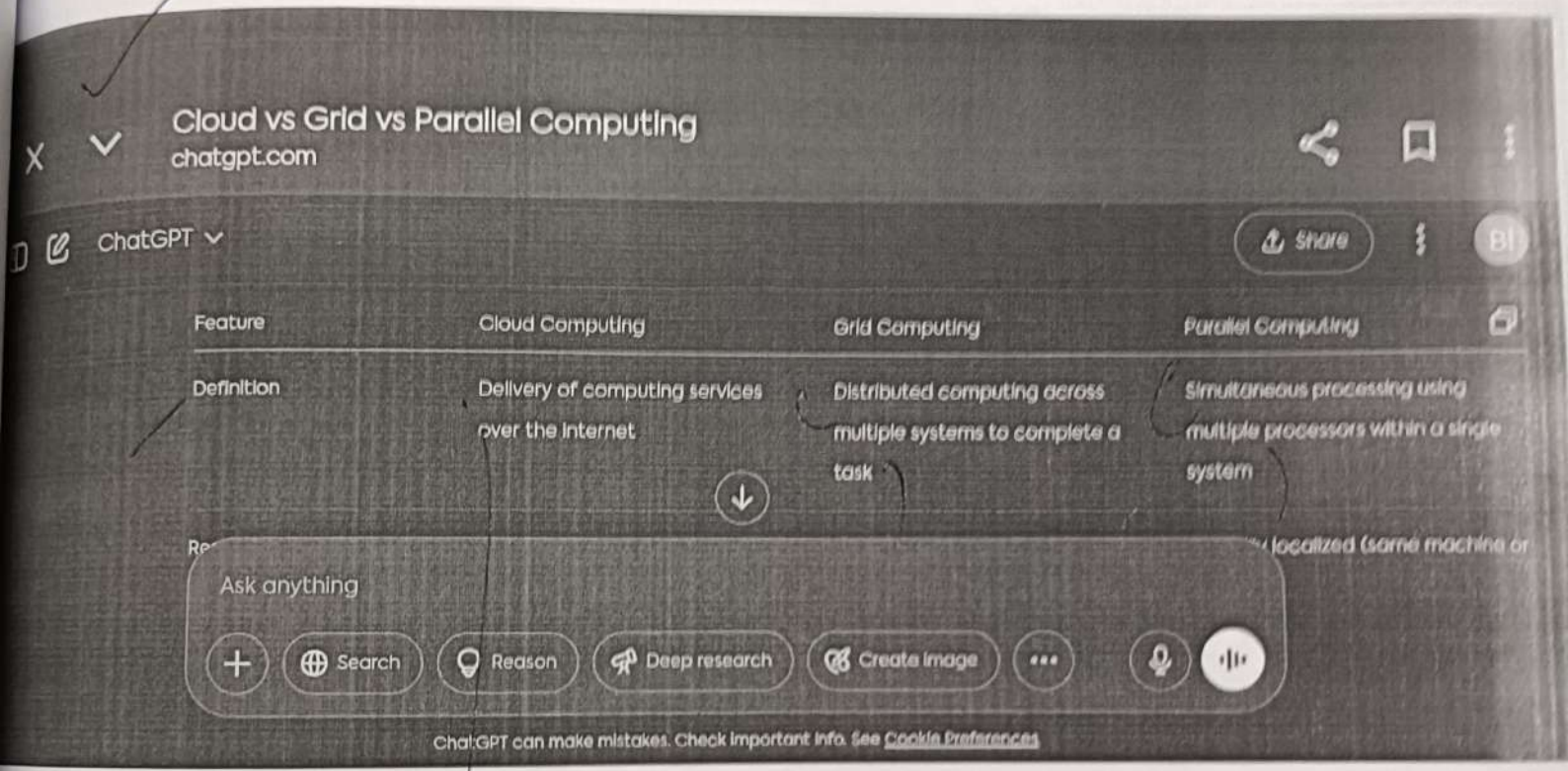
iii) Eliminates the initial investment costs.

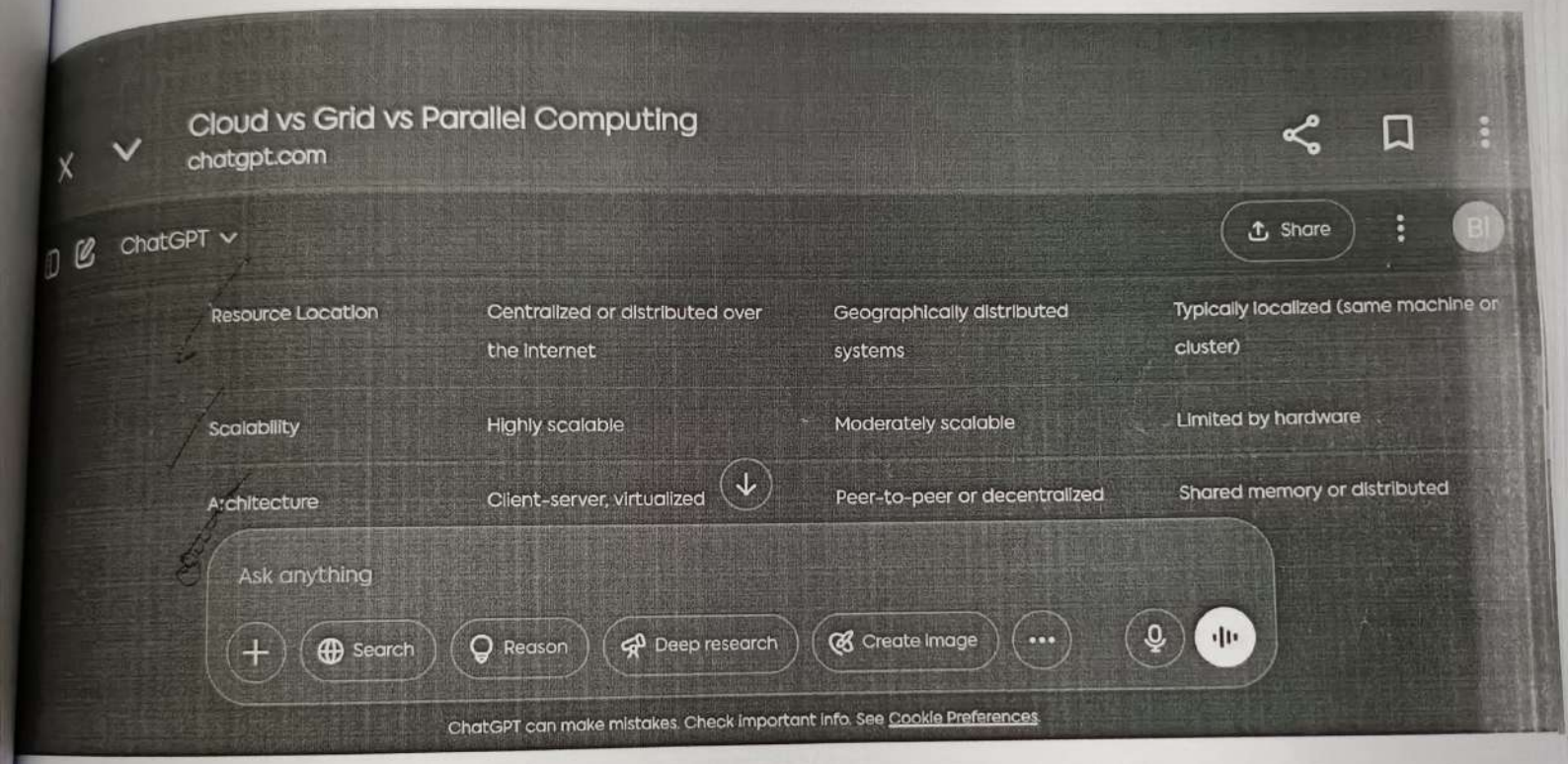
iv) Cost reduction [concentration of resources creates the opportunity to pay as you go for computing).]

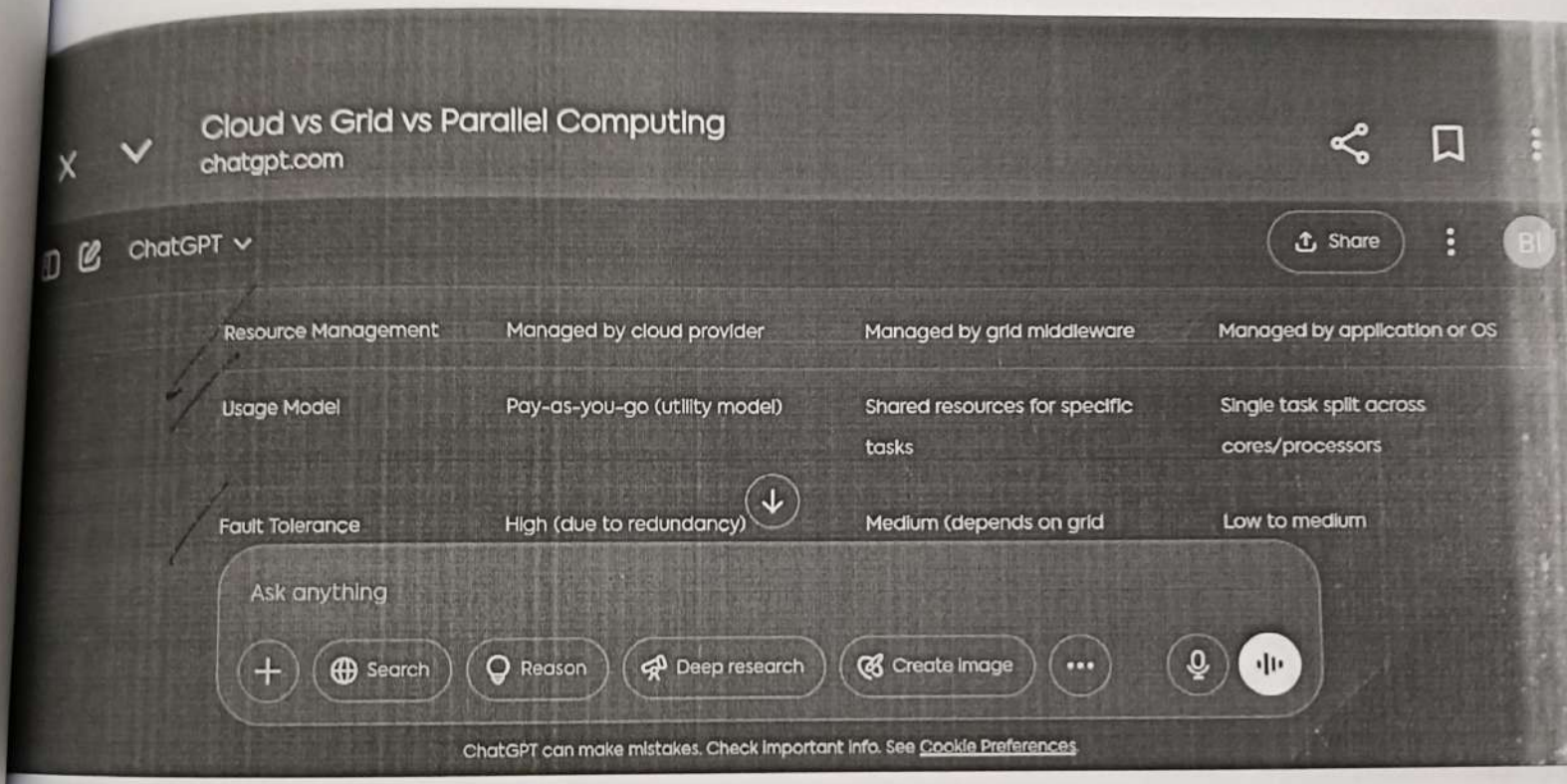
v) Elasticity (the ability to accomodate workloads with very large peak-to-avg ratios)

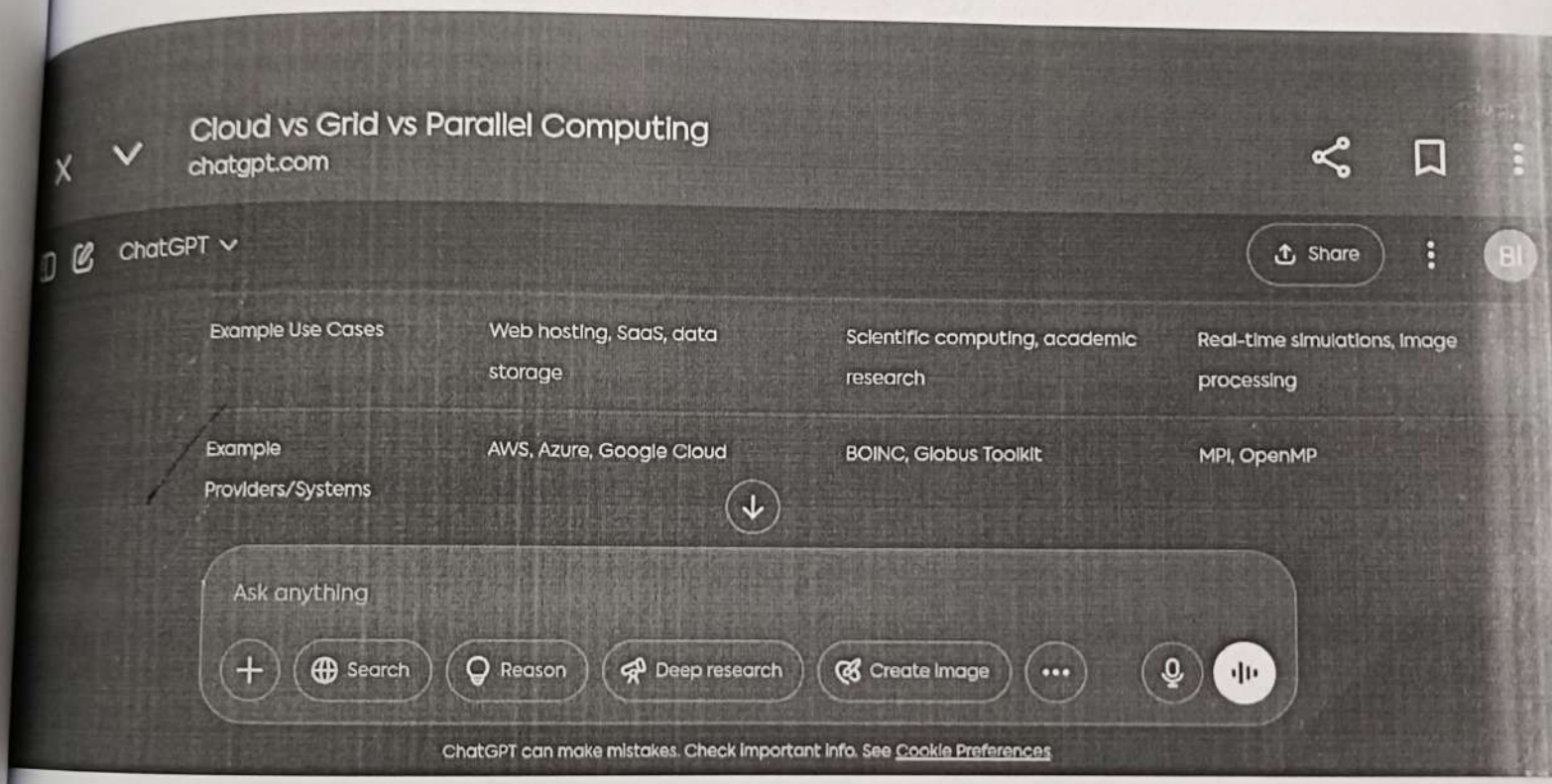
vi) User convinence (virtualization allows users to operate in familiar env [rather than idyosyncratic ones])

vii) Data sharing facilitates collaborative activities.


Cloud vs Grid vs Parallel Computing
chatgpt.com
ChatGPT
Share
Feature
Cloud Computing
Grid Computing
Parallel Computing
Definition
Delivery of computing services over the internet
Distributed computing across multiple systems to complete a task
Simultaneous processing using multiple processors within a single system
localized (same machine or
Ask anything
Search
Reason
Deep research
Create image
ChatGPT can make mistakes. Check important info. See Cookie Preferences


Cloud vs Grid vs Parallel Computing
chatgpt.com
ChatGPT
Share
Resource Location
Centralized or distributed over the internet
Geographically distributed systems
Typically localized (same machine or cluster)
Scalability
Highly scalable
Moderately scalable
Limited by hardware
Architecture
Client-server, virtualized
Peer-to-peer or decentralized
Shared memory or distributed
Ask anything
Search
Reason
Deep research
Create image
ChatGPT can make mistakes. Check important info. See Cookie Preferences


Cloud vs Grid vs Parallel Computing
chatgpt.com
ChatGPT
Share
Resource Management
Managed by cloud provider
Managed by grid middleware
Managed by application or OS
Usage Model
Pay-as-you-go (utility model)
Shared resources for specific tasks
Single task split across cores/processors
Fault Tolerance
High (due to redundancy)
Medium (depends on grid
Low to medium
Ask anything
Search
Reason
Deep research
Create image
ChatGPT can make mistakes. Check important info. See Cookie Preferences.


Cloud vs Grid vs Parallel Computing
chatgpt.com
ChatGPT
Share
Example Use Cases
Web hosting, SaaS, data storage
Scientific computing, academic research
Real-time simulations, image processing
Example Providers/Systems
AWS, Azure, Google Cloud
BOINC, Globus Toolkit
MPI, OpenMP
Ask anything
Search
Reason
Deep research
Create image
ChatGPT can make mistakes. Check important info. See Cookie Preferences

Virtualized Resource

In subject area: Computer Science

A 'Virtualized Resource' refers to a resource that is abstracted from the physical infrastructure through virtualization technologies, allowing it to be dynamically allocated and migrated across different physical machines in Cloud Computing environments.

AI generated definition based on: Computer Networks, 2013

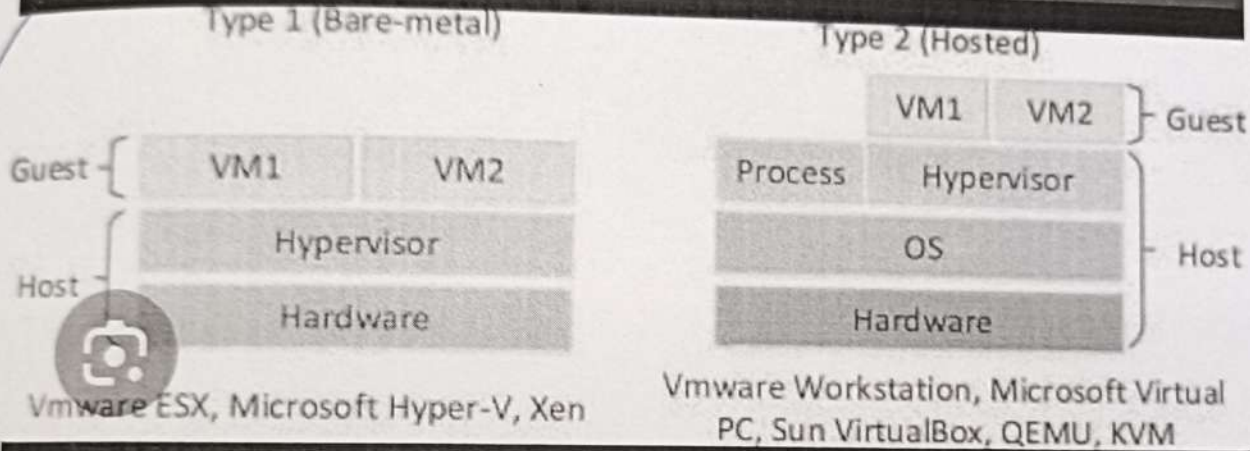
On this page

Chapters and Articles

You might find these chapters and articles relevant to this topic.

Mini review

ScienceDirect.com



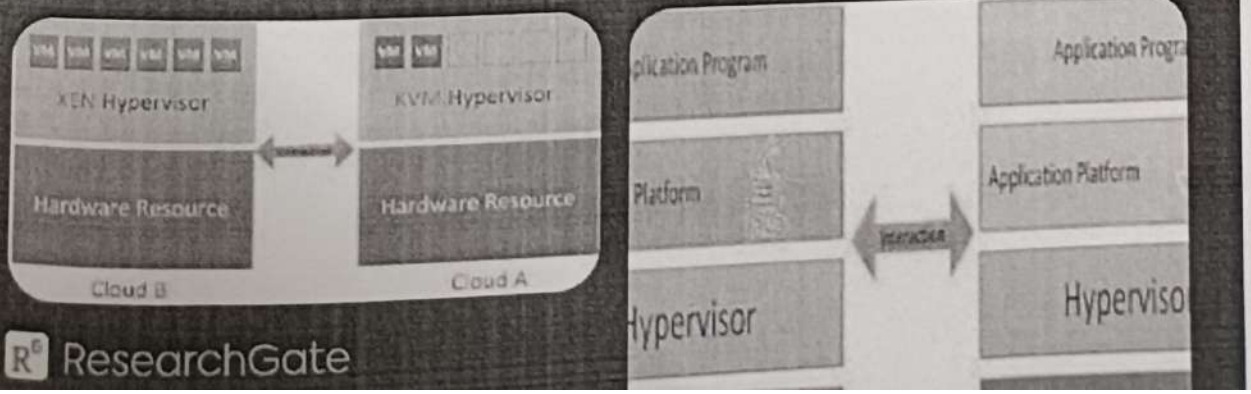
Virtualized Resource - an overview | ScienceDirect...

Visit ›

Images may be subject to copyright. Learn More

Share

Save



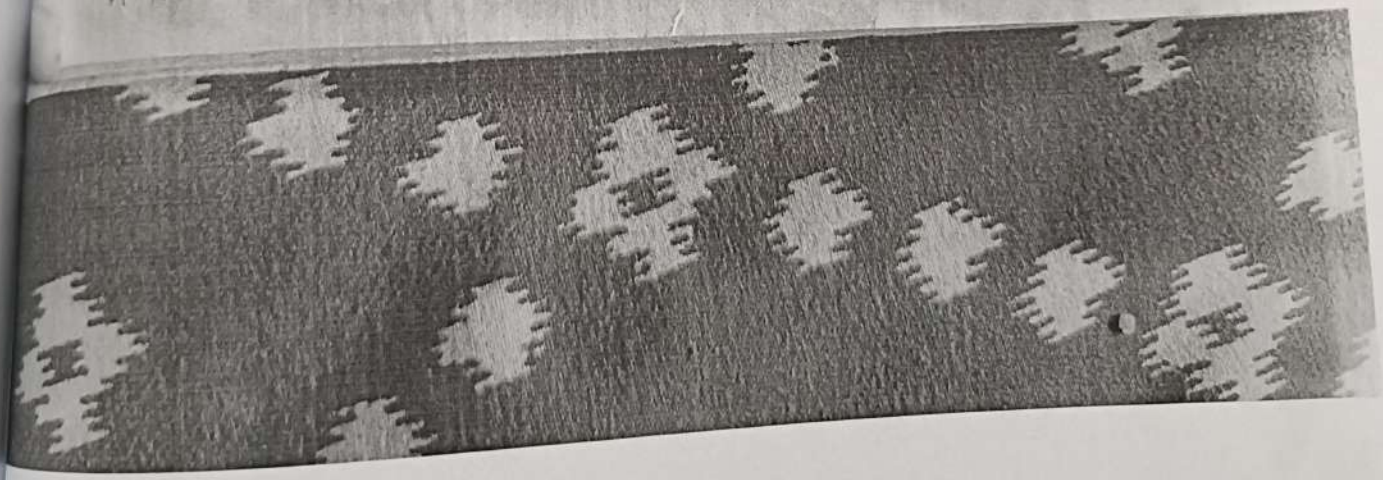
ResearchGate

Chap - 3
Pg = 26 - 44 (Book from which we need to study)

Imp

• Scaling

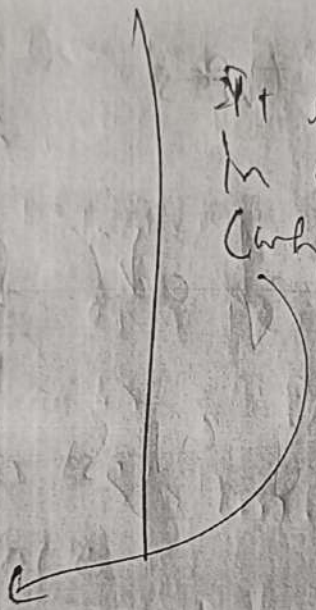
Scaling, from an IT resource perspective, refers the ability of the IT resource to handled increased or decreased usage



demands. There are 2 types of scaling: horizontal & vertical)

1) Horizontal — Scaling out & Scaling in
= Scaling up & Scaling down

Horizontal	vertical
i) Scaling out & Scaling in	i) Scaling up & Scaling down
ii) The allocating & releasing of IT resources that are of the same type is refd. to as horizontal Scaling. The horizontal allocation of resources is refd. to as Scaling out & the releasing of resources is refd. to as Scaling in. It is a common form of Scaling with the cloud environment.	ii) when an existing IT resource is replaced by another with higher or lower capacity, then it's called vertical Scaling. The replacing of an IT resource with another that has a higher capacity is refd. to as scaling up & the replacing of the IT resource with another that has a lower capacity is refd. to as Scaling down



It is less common in C. environment. (why?)

~~It is~~ vertical Scaling is less common in cloud environment.

Why?

ps the infrastructure management

Integration challenges - Compatibility with existing on-premise systems is low.

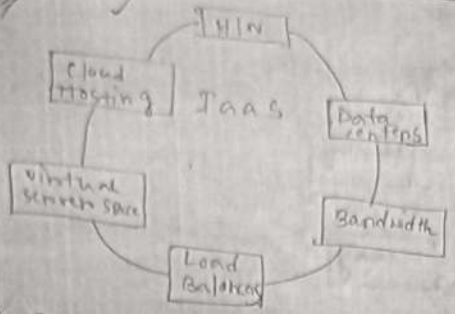
Performance issues - Latency & speed variations

- cloud delivery ~~(deployment)~~ model based on n/w conditions.

iii) Iaas, iii) paas, iii) saas

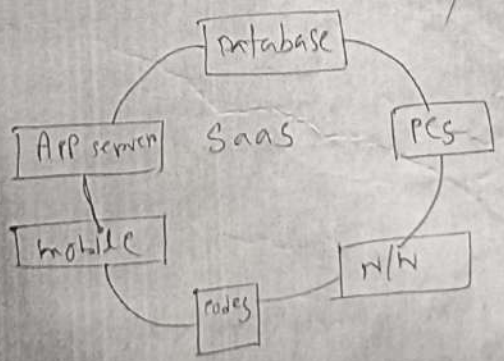
The three main cloud delivery models are Infrastructure as service (Iaas), Paas, Saas.

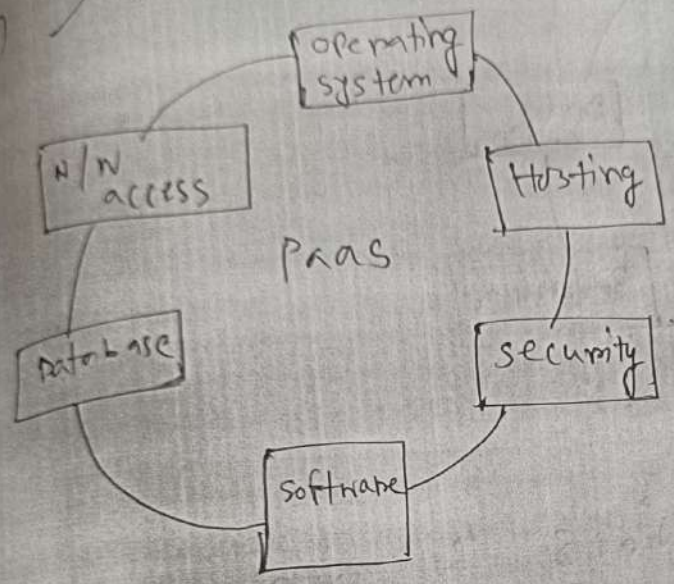
Iaas - (It provides the users the ability to use the computing & storage resources.
Eg - Google drive etc.)



Paas — It Provides the users the ability to develop & deploy application in cloud using the development tools, APIs, S/W libraries & services provided by the cloud service Provider (CSP). Eg- ~~Azure~~ IBM Cloud, Red Hat Openshift etc. ⊛)

S/W as a service (Saas) — Provides users a complete S/W application / the user interface to the application. Eg- google sheet, Powerpoint etc.)





16/4/25

CW

- Cloud → manipulation
  - accessing the services provided by Cloud
  - configuration
- (Core Prov)

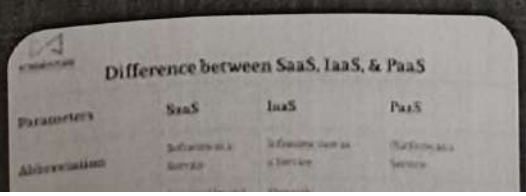
MultiQoS

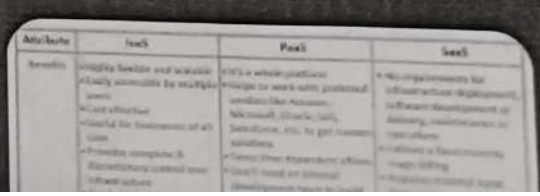
Variables	SaaS	PaaS	IaaS
Full form	Software as a service	Parameter as a service	Infrastructure as a service
Used for	It is used for cloud computing software development	It is used for application development in cloud computing	It is used for any cloud computing development
Type of users	Any users who use it	It is used by software and application developers	Network architects use it
Access	SaaS gives access to any software that you want to use	It provides the necessary tools for development	You can store data and also use virtual machines
Technical know-how	If you want to use SaaS, you do not have to be a technical expert	You need to have a piece of proper knowledge of developing applications	You need to have excellent technical knowledge
Popularity	Most consumers and businesses focus on SaaS	PaaS is primarily popular among application developers	IaaS is popular among researchers and developers
Cost	The cost of using SaaS is minimum as you use the software according to your requirements	The cost of using PaaS is low	The cost of using IaaS is the highest

eg. Google sheet PPT IBM Cloud RedHat Openshift Google drive

SaaS Vs PaaS Vs IaaS: What's the Difference...

Visit ›





• Types of clouds - Computing deploy ment models are described below:

(means deployment models)

IaaS, PaaS, SaaS = C.C. models - delivery

i) Public - It's a commonly adopted cloud model, where the cloud services provider owns the infrastructure & openly provides access to it for the public to consume.
Eg - google cloud, amazon aws etc.

ii) Private - It can be thought of as an environment that is fully owned & managed by a single tenant. It's a c.c. env. that is dedicated to a single organisation. Its also known as corporate or internal cloud. Eg - Azure stack.

iii) Hybrid - It's a computing env. that combines public & private clouds, & some on-premises data centers. It allows data & applications to move b/w the different envs. Eg - Netflix uses a hybrid c.m. to store & manage a large amount of video contents

iv) Other types; Community / Federated
It's a c. infrastructure in which multiple organizations share resources & services based on common operational & regulatory requirements. It's a semi-public cloud. Eg - Healthcare)

• Why C.C is successful, when other paradigms have failed?
- gms

i) It's in a better position to exploit recent advances in S/w, h/w, storage & Processor technologies promoted by the same companies who provide cloud services.

... can contain services & ...

What is CC - 1. Deployment model, 2 service model.

Deployment models - 4 types

i) public cloud, ii) private cloud, iii) Hybrid cloud, iv) Community cloud (Done Pw)

- Diff b/w Private & Public cloud (vice versa cotains clouds)
- Adv & disadv of each deployment models

Features	Public	Private	Hybrid	Community
Host	service Providers	Enterprise	Enterprise	Community (3rd Party)
Suitable for	Large enterprise	Large enterprise	Small & mid-size	Financial, health & legal compines.
Access	Internet	~~Internet~~, VPN, Intranet	Intranet, VPN	Intranet, VPN
Security	Low	most Secured	moderate	Secured
Cost	Cheapest	High cost	cost effecti-ve	Cost effective
Owners	Service Providers	Enterprise	Enterprise	Community

iv) Reliability	moderate	very high	medium-high	very high
v) scalability	very high	Limited	very high	Limited
vi) users	organization, public etc	Business organizations	Business org.	Community members

6. Adv & disadv of each deployment models—

i) Public

Adv – i) Cost effective, ii) scalable & flexible, iii) No maintenance is needed by users, iv) Fast setup & deployment

Disadv – i) Less control over infrastruct., ii) Potential security concerns, iii) Limited customization, iv) shared resources can affect performance.

ii) Private

Adv – i) Greater control & customization, ii) Enhanced security & privacy, iii) Better performance, iv) Strict regulations.

Disadv – i) High cost, ii) Requires in-house expertise, iii) Longer deployment time, iv) Scalability can be limited.

iii) Hybrid

Adv – i) Flexible & scalable, ii) Balanced cost & control iii) Improved business continuity, iv) Allows sensitive data to stay on Private cloud.

Disadv – i) Complex management, ii) Integration challenges, iii) Security risks in data transfer, iv) Higher setup costs than public cloud

iv) Community

Adv – i) Cost effective, ii) Better security than Public cloud, iii) shared responsibilities & costs, iv) Customizable for group needs.

Disadv – i) Limited scalability, ii) Governance challenges, iii) Costly than public c., iv) slow adoption & updates

service m

the cl

types

i) Iaas,

Ans

• Rahu

usi

Ser

w

• Ber

Ans

Hypervisor T...
chatgpt.com

ChatGPT

Hypervisor Type 1 (Bare Metal Hypervisor):

- Installed directly on physical hardware (no operating system needed).
- Manages virtual machines (VMs) directly.
- Examples: VMware ESXI, Microsoft Hyper-V, Xen.

How it works:

1. Installed on the server hardware.
2. Creates and runs multiple VMs.
3. Each VM runs its own operating system.
4. Offers high performance and better security.

Hypervisor Type 2 (Hosted Hypervisor):

- Installed on top of a regular operating system

Ask anything

Hypervisor T...
chatgpt.com

ChatGPT

Hypervisor Type 2 (Hosted Hypervisor):

- Installed on top of a regular operating system (like Windows or Linux).
- Runs VMs like any other software application.
- Examples: VMware Workstation, Oracle VirtualBox.

How it works:

1. Installed on an existing OS (e.g., Windows).
2. Creates VMs as applications on the OS.
3. Easier to set up but not as fast or secure as Type 1.

Ask anything

usless security than type 1

Amazon Sim...
chatgpt.com

ChatGPT

Feature	Amazon SimpleDB	Amazon RDS
Type	NoSQL (Key-Value Store)	Relational Database Service
Data Model	Schema-less, attribute-value pairs	Structured, schema-based tables
Query Language	Proprietary SimpleDB query language	Standard SQL
Use Case	Lightweight, non-relational data storage	Complex, relational data workloads
Scalability	Automatically scales horizontally	Vertical and limited horizontal scaling (read replicas)

Ask anything

Scalability	Automatically scales horizontally	Vertical and limited horizontal scaling (read replicas)
Transaction Support	Limited support	Full ACID transaction support
Consistency	Eventual consistency	Strong consistency
Indexing	Automatic indexing on all attributes	User-defined indexes
Backup & Recovery	Manual via application	Automated backups, snapshots, point-in-time recovery
Availability	Designed for high availability	Supports Multi-AZ deployments for high availability

App Engine ...
chatgpt.com

ChatGPT

1. Standard Environment

- Languages supported:
  - Python
  - Java
  - Node.js
  - Go
  - PHP
  - Ruby
- Fast startup, automatic scaling, limited customization.

2. Flexible Environment

- Languages supported:
  - Same as Standard (Python, Java, Node.js, etc.)

Ask anything

← Deployment tools currently supported by GAE

- Languages supported:
  - Same as Standard (Python, Java, Node.js, etc.)
  - Plus: .NET Core and any language using Docker
- More contro, supports custom runtimes, slower startup, scales with containers.

3. Custom Runtimes

- Use your own Dockerfile
- Deploy any language or tool (as long as it runs in Linux)

Want a quick table comparing them?

with AppEngine services. As happens for the Python environment [illegible] removed or generate a run-time exception. Also, developers can include third party libraries into their applications as long as they are implemented in pure Go.

3. Storage

AppEngine provides different types of storage, which operate differently according to the volatility of the data they are designed for. There are three different level of storage: in memory-cache, storage for semi-structured data, and long-term storage for static data. In this section, we describe *DataStore* and the use of static file servers and we cover *MemCache* in the application services section.

(a) Static File Servers. Web applications are composed by dynamic and static data. Dynamic data is a result of the logic of the application and the interaction with the user. Static data often is mostly constituted by the components that define the graphical layout of the application (css files, plain html files, javascript files, images, icons, and sound files) or data files. These files can be hosted on static file servers, since they are not frequently modified. Such servers are optimized for serving static content and users can specify how dynamic content should be served when uploading their applications to AppEngine.

(b) DataStore. DataStore is a service allowing developers to store semi-structured data. The service is designed to scale and optimized to quickly access data. DataStore can be considered as a large object database where to store objects that can be retrieved by a specified key. Both the type of the key and the structure of the object can vary.

With respect to the traditional Web applications backed by a relational database, DataStore imposes less constraint on the regularity of the data but, at the same time, does not implement some of the features of the relational model (such as reference constraints and join operations). These design decisions originated from a careful analysis of data usage patterns for Web applications and have been taken in order to obtain a more scalable and efficient data store. The underlying infrastructure of *DataStore* is based on *Bigtable* [93], which is a redundant, distributed, and semi-structured data store that organizes data in the form of tables (See Section 8.2.1).

DataStore provides high-level abstractions that simplify the interaction with Bigtable. Developers define their data in terms of *entity* and *properties*, and these are persisted and maintained by the service into tables in *Bigtable*. An entity constitutes the level of granularity for the storage and it identifies a collection of properties, which define the data it stores. Properties are defined according to one of the several primitive types supported by the service. Each entity has associated a key, which is either provided by the user of created automatically by AppEngine. An entity is associated with a *named kind* that is used by AppEngine to optimize its retrieval from Bigtable. Although entities and properties seem to be similar to rows and tables in SQL, there are a few differences that have to be taken account. Entity of the same kind might not have the same properties, and properties of the same name might contain values of different types. Moreover, properties can store different version of the same values. Finally, keys are immutable elements and once created, they cannot be changed anymore.

DataStore also provides facilities for creating indexes on data and to update data within the context a transaction. Indexes are used to support and speed-up queries. A query can return zero or more ects of the same kind or simply the corresponding keys. It is possible to query the data store either specifying the key or conditions on the values of the properties. Returned result sets can be sorted ner by key value or properties value. Even though the queries are quite similar to SQL queries, their plementation is substantially different. DataStore has been designed to be extremely fast in returning ult sets. In order to do so, it needs to know in advance all the possible queries that can be done for on kind because it stores for each of them a separate index. The indexes are provided by the user

1. Primitive Data Types

- String
- Integer
- Float (Double)
- Boolean (True/False)
- Null

2. Date and Time

- Timestamp (date and time with timezone info)

3. Complex Types

- List/Array (of values of the supported types)
- Embedded Entity (a nested dictionary/object)
- GeoPoint (latitude and longitude)
- Blob (binary data like images, files)

- Timestamp (date and time with timezone info)

3. Complex Types

- List/Array (of values of the supported types)
- Embedded Entity (a nested dictionary/object)
- GeoPoint (latitude and longitude)
- Blob (binary data like images, files)

4. Key

- Used to identify and reference entities (like primary key)

Each entity in Datastore is like a flexible JSON object — it doesn't need a fixed schema.

Would you like an example entity structure?

25/2/25

• Cluster Computing

(A cluster is a group of independent IT resources that are interconnected & works as a single system.)

Dis Adv -

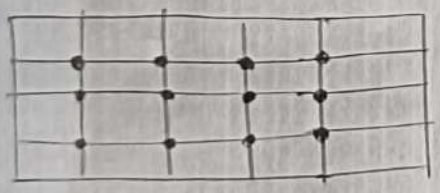
Availability of the resources.

Adv -

Redundancy & failure.

• Grid Computing

Grid Computing provides a platform in which computing resources are organised into one or more logical pools.



These pools are ~~collecting~~ collectively coordinated to provide high performance distributed grids. It is also called super virtual computer.

• Cloud

A cloud refers to distinct IT environment that is designed for the purpose of remotely provisioning scalable & major IT resources.

The term is originated from ... virtual

Feature	Internet	Cloud
Definition	A global network connecting computers	A service that provides computing resources via the internet
Function	Enables communication and data sharing	Offers storage, computing power, and software services
Access	Used to access websites, email, social media, etc.	Used to access online services like Google Drive, AWS, Microsoft Azure
Infrastructure	Physical network of servers, cables, and devices	Virtual infrastructure managed by cloud providers

Dep

Infrastructure	Physical network of servers, cables, and devices	Virtual infrastructure managed by cloud providers
Dependency	Independent; cloud depends on the internet	Dependent on the internet to function
Example	Browsing websites, sending emails	Storing files on OneDrive or using Google Docs

Let me know if you'd like this in PDF or image format for easier sharing.

iv) Dropbox uses a hybrid c.m. to store & manage a large amount of video contents

iv) Others types; Community / Federated

It's a c. infrastructure in which multiple organizations share resources & services based on common operational & regulatory requirements. It's a semi-Public cloud. Eg - Heroku etc.

Why C.C is successful, when other paradigms have failed?

i) It's in a better position to exploit recent advances in S/w, h/wing, storage & Processor technologies promoted by the same companies who provide cloud services.

ii) Economical reasons - It is used

S/w requirement are changing due to the growth of hardware rapid

for Enterprise computing; it's adoption by industrial organisations, government etc has a huge impact on the economy.

iii) Infrastruct. management reasons.

a) A single cloud consists of a mostly homogeneous set of hardware & S/N resources.

b) the resources are in a single admini-tive domain (AD).

- Challenges of C.C

i) security risks - Data ~~bas~~ breaches

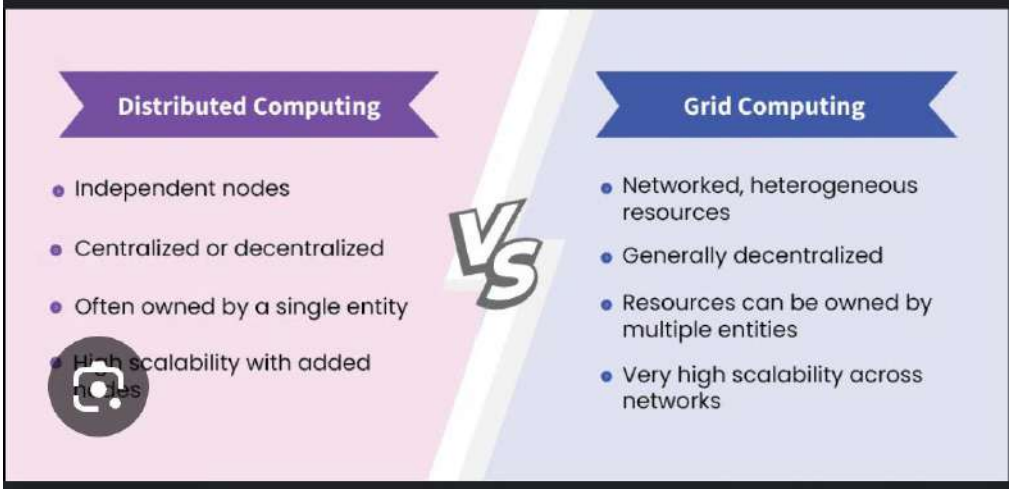
ChatGPT

Here are one-line, simple and brief definitions:

- Full Virtualization: Simulates complete hardware, allowing unmodified OS to run as if on real hardware.
- Hardware-Assisted Virtualization: Uses CPU features (like Intel VT or AMD-V) to boost virtual machine performance.
- Paravirtualization: OS is modified to interact directly with the hypervisor for better performance.
- Emulated Virtualization: Entire hardware is mimicked in software, allowing any OS to run but with slower performance.

Ask anything

GeeksforGeeks



Distributed Computing vs. Grid Computing |...

Visit >

Images may be subject to copyright. Learn More

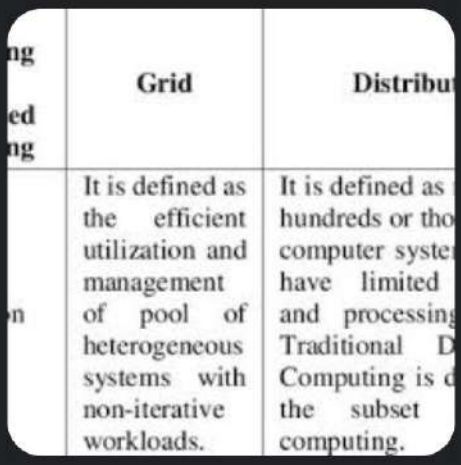
Share Save

DifferenceBetween.net

Grid Computing VERSU...

GeeksforGeeks

Distributed System vs. ...

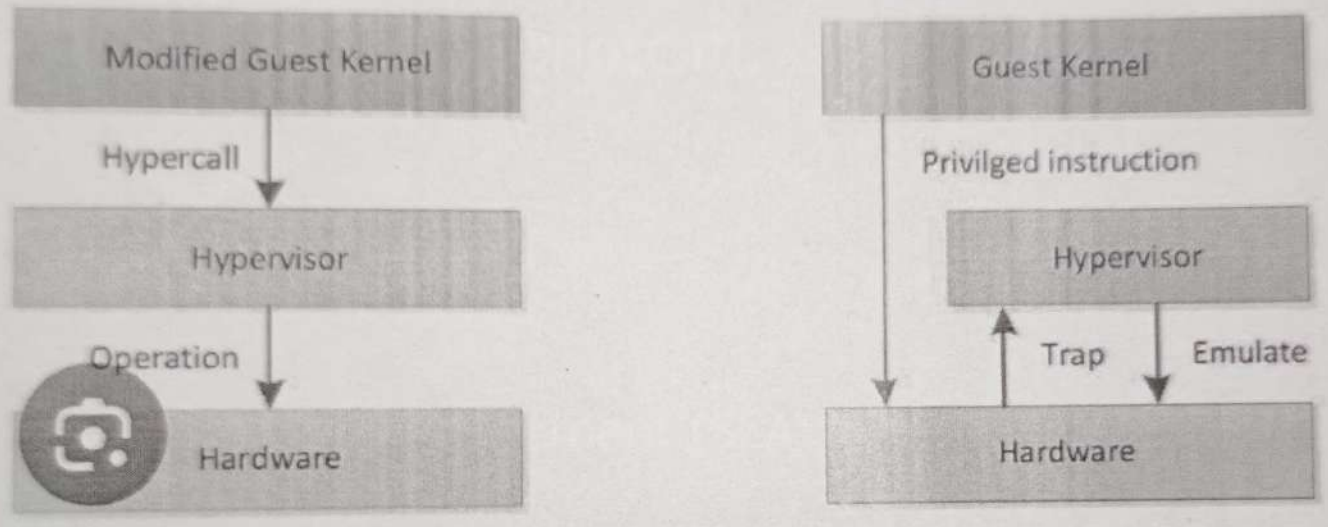


ResearchGate

ITsecurity Demand

Home Search Saved Notifications

Medium

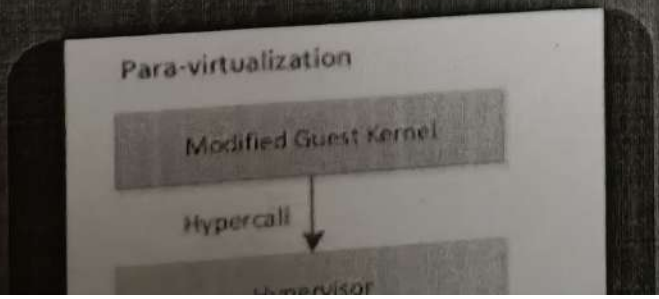


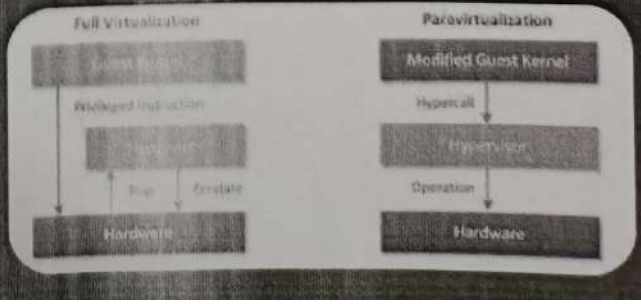
Understanding Hypervisors: Exploring...

Visit >

Share

Save





SLA in Cloud...
chatgpt.com

ChatGPT

A Service Level Agreement (SLA) in cloud computing is a promise between the cloud provider and the user about how well the service will work. It clearly states things like:

- How often the service will be available (like 99.9% uptime)
- How fast problems will be fixed
- What happens if the service doesn't meet these promises

It helps users know what to expect and what they'll get if things go wrong.

Ask anything

9.1.2 Storage Services

AWS provides a collection of services for data storage and information management. The core service in this area is represented by Amazon *Simple Storage Service (S3)*. This is a distributed object store that allows users to store information in different. The core components of S3 are two: *buckets* and *objects*. Buckets represent virtual containers where to store objects, while objects represent the content that is actually stored. Objects can also be enriched with metadata that can be used to tag the content stored with additional information.

1. S3 Key Concepts

As the name suggest, S3 has been designed to provide a simple storage service accessible through a REST interface, which is quite similar to a distribute file system but that presents some important differences that allow the infrastructure to be highly efficient:

The storage is organized in a two level hierarchy. S3 organizes its storage space into buckets that cannot be further partitioned. This means that it is not possible to create directories or other kinds of physical groupings for objects stored in a bucket. Despite this, there are less limitations in naming objects, and this allows users to simulate directories and create logical groupings.

Objects stored cannot be manipulated like standard files. S3 has been designed to essentially provide storage for objects that will not change over time. Therefore, it does not allow renaming, modifying, or relocating an object. Once an object has been added to a bucket, its content and position is immutable, and the only way to change one of them is to remove the object from the store and add it again.

Content is not immediately available to users. The main design goal of S3 is to provide an eventually consistent data store. As a result, being a large distributed storage facility, changes are not immediately reflected. For instance, S3 uses replication to provide redundancy and efficiently serve objects across the globe; this introduces latencies when adding objects to the store—especially large ones—which are not available instantly across the entire globe.

Request will occasionally fail. Due to the large distribute infrastructure being managed, requests for object may occasionally fail. Under certain conditions, S3 can decide to drop a request by returning an internal server error. Therefore, it is expected to have a small failure rate during day-to-day operations, which generally it does not identify a persistent failure.

Access to S3 is provided with RESTful Web services. These express all the operations that can be performed against the storage in the form of HTTP requests (GET, PUT, DELETE, HEAD, and POST), which operate differently according to the element they address. As a rule of thumb, PUT/POST requests add new content to the store, GET/HEAD requests are used to retrieve content and information, while DELETE requests are used to remove elements or information attached to it.

(a) Resource Naming. Buckets, objects, and attached metadata are made accessible through a REST interface. Therefore, they are represented by *Uniform Resource Identifiers (URIs)* under the *s3.amazonaws.com* domain. All the operations are then performed by expressing the entity they are directed to in the form of a request for a URI.

Amazon offers three different ways of addressing a bucket:

Canonical Form: http://s3.amazonaws.com/bukect_name/. The bucket name is expressed as a path component of the domain name s3.amazonaws.com. This is the naming convention that has less restriction in terms of allowed characters, since all the characters that are allowed for a path component can be used.

Subdomain form: http://bucketname.s3.amazon.com/. Alternatively, it is also possible to reference a bucket as a subdomain of s3.amazonaws.com. In order to express a bucket name into this form, the name has to:

- be between 3 to 63 characters long;
- contain only letters, numbers, periods, and dashes;
- start with a letter or a number;
- contain at least one letter; and
- have no fragments between periods that start with a dash, end with a dash, or are an empty string.

This form is equivalent to the previous one when it can be used but it is the one to be preferred since it [illegible] for all the geographical locations serving resources stored in S3.

SLA in Cloud...
chatgpt.com

ChatGPT

Amazon CloudWatch is a monitoring service from AWS that helps you track and understand what's happening in your cloud resources.

In simple words, CloudWatch watches your AWS services (like EC2, RDS, Lambda) and gives you:

- Metrics (e.g., CPU usage, memory, disk activity)
- Logs (detailed records of what your applications are doing)
- Alarms (alerts when something goes wrong or crosses a limit)

It helps you keep your apps running smoothly by spotting problems early.

Would you like an example of how CloudWatch is used?

Ask anything

SLA in Cloud...
chatgpt.com

ChatGPT

Amazon EC2 (Elastic Compute Cloud) is a service from AWS that lets you run virtual computers (called instances) in the cloud.

In simple words, EC2 is like renting a computer on the internet. You can use it to:

- Run websites or apps
- Store files
- Process data
- Test software

You choose how powerful the computer is, how long you want it, and what software runs on it.

Would you like to know how to launch an EC2 instance?

Ask anything

SQL Azure (now called Azure SQL Database):

SQL Azure is a cloud-based relational database service from Microsoft, part of the Azure platform. It is built on SQL Server technology and provides:

- High availability and scalability without the need to manage hardware or infrastructure.
- Supports T-SQL, ADO.NET, ODBC, and other common interfaces.
- Offers automatic backups, geo-replication, and intelligent performance tuning.

Use Case: Ideal for applications needing a scalable, secure, and fully managed database in the cloud.

AppFabric (retired service):

Windows Azure AppFabric was a set of middleware services that helped developers connect applications and services in the cloud and on-premises. It included:

- Service Bus: For messaging and communication between distributed applications.
- Access Control: For federated identity management and authentication.

Note: AppFabric has been retired, and its features are now integrated into Azure Service Bus and Azure Active Directory.

Use Case (when active): Used to simplify building and managing distributed cloud applications.

Let me know if you need their architecture or

Here are the main Azure Storage Services in points:

1. Blob Storage
    - Stores unstructured data like images, videos, documents.
    - Ideal for large-scale data storage and backups.
2. File Storage (Azure Files)
    - Fully managed file shares in the cloud.
    - Supports SMB and NFS protocols for mounting.
3. Queue Storage
    - Messaging system for communication between applications.
    - Ensures reliable message delivery.
4. Table Storage

3. Queue Storage
    - Messaging system for communication between applications.
    - Ensures reliable message delivery.
4. Table Storage
    - NoSQL key-value storage for structured, non-relational data.
    - Suitable for applications with large-scale datasets.
5. Disk Storage
    - High-performance, durable block storage for Azure VMs.
    - Includes SSD and HDD options.

Let me know if you want use cases or comparisons for each.