Cloud Computing - 315325

❖ 1.Fundamentals of Cloud Computing

- **Syllabus**

At the end of this unit, you should be able to understand and comprehend the following syllabus topics:

1.1 Definition of Cloud Computing, Characteristics of Cloud computing.

1.2 Cloud Deployment. Models (Introduction, advantages and disadvantages): Public Cloud, Private Cloud, Community Cloud, Hybrid Cloud.

1.3 Cloud Service Models (Function, advantages, disadvantages): laaS, PaaS, SaaS

14 Cloud cost benefits

1.5 Architectural and Infrastructural components of Cloud Computing.

❖ Introduction to Cloud Computing

Before I discuss anything on cloud computing with you, it is important for you to understand the mindset of a service and what makes it different from the regular product consumption.

Table 1.1.1 : Consuming Services Vs Owning Products

Comparison Attribute	Taxi Service (Service Consumption)	Car Ownership (Product Consumption)
Ownership	Service Provider	You
Maintenance, Taxes, Cleaning	Service Provider	You
Fuel	Service Provider	You
Overnight Parking	Service Provider	You
Liability / Commitment	Only during the service consumption	Long-term or until lifetime
Consumption Model	On demand, self-service via app	Always available
Availability	Anywhere in the world	Only in the city you own the car
Used by	Anyone in the world	Whomsoever you choose
Price / Billing	Pay as you go (computed for actual distance)	Capital and Operational Investment
Number of cars you can use	Nearly unlimited	Limited to cars that you own

It is crucial to understand the above characteristics of service consumption. I would be referring to this basic block of information several times throughout this unit as an

analogy to explain various topics on cloud computing. You could refer to Table 1.1.1 for quickly understanding the topics as I discuss them with you.

❖ What is Cloud Computing?

- **Note:**- When talking about **cloud computing**, you might come across different definitions. However, for clarity and consistency, I will be using the definitions and terms provided by **NIST (National Institute of Standards and Technology)**.NIST's definitions are widely accepted in the tech industry and are considered the most accurate. Even though other definitions might mean the same thing, using NIST's standard helps avoid confusion.

➢ **Definition:-** Cloud computing means **using computers and services over the internet** instead of owning them yourself.

- Imagine you need 5 computers to do some work. Instead of buying them, setting them up, and maintaining them, you just **rent them from a cloud provider** (like Amazon Web Services, Microsoft Azure, or Google Cloud). You access them through the internet, and you only pay for the time you use them.
- Here's what makes cloud computing special:

➢ **No need to buy hardware** – You don't have to purchase computers or servers.
➢ **Pay as you go** – You only pay for how long you use the service.
➢ **Access from anywhere** – Since it's on the internet, you can use it from any location.
➢ **No maintenance hassle** – You don't have to worry about electricity, storage, or keeping the system running.
➢ **Easy to scale** – Need more computers tomorrow? Just update your plan and get them instantly.

1.1.3 Goals of Cloud Computing

Fig. 1.1.1 shows the high-level goals of cloud computing.

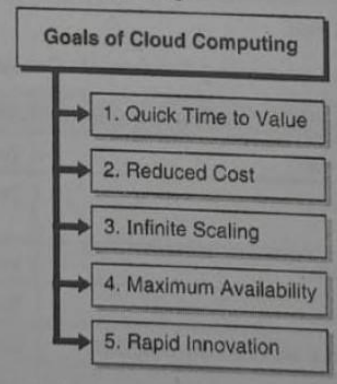


Fig. 1.1.1 : Goals of Cloud Computing

➢ **Quick Time to Value**
Cloud computing lets you start using IT resources (like servers or storage) **right away** without wasting time setting up hardware.
Example: Just like you don't worry about how electricity is generated — you just plug in and use it — cloud lets you focus on your business instead of managing computers.

➢ **Lower Costs**
You don't need to **buy expensive equipment** or run your own data center (which needs space, electricity, staff, etc.).
You only pay for what you use, like your electricity bill.
Example: You don't build your own power station — you use electricity from the grid and pay based on usage.

❖ **Scales as You Grow**

Cloud providers have a **huge amount of resources**. If your needs grow, you can quickly get more computing power without delay.

Example: You don't stop to check how much electricity you have — you just keep using it. Similarly, cloud gives you more computing power when you need it, like it's unlimited.

➢ **High Availability**

Cloud services are **very reliable**. If one server goes down, another one quickly takes its place — often without you even noticing.

Cloud providers also guarantee uptime through **Service Level Agreements (SLAs)**, often referred to in terms of **"nines"** (like 99.9% or 99.999% availability).

Example: It's like the power grid — if one power line fails, electricity is usually rerouted, so your lights stay on.

Table 1.1.2 : What Different "Nines" of Availability Mean

Availability	Annual Downtime
99.9% ("three nines")	8.77 hours
99.95% ("three nines five")	4.38 hours
99.99% ("four nines")	52.60 minutes
99.995% ("four nines five")	26.30 minutes
99.999% ("five nines")	5.26 minutes
99.9999% ("six nines")	31.56 seconds
99.99999% ("seven nines")	3.16 seconds
99.999999% ("eight nines")	315.58 milliseconds
99.9999999% ("nine nines")	31.56 milliseconds

It is not unusual for a cloud service provider to provide an SLA of "five nines" which just means a downtime of 5.26 minutes in a year!

For example, here is the SLA from Microsoft Azure for some of its services.

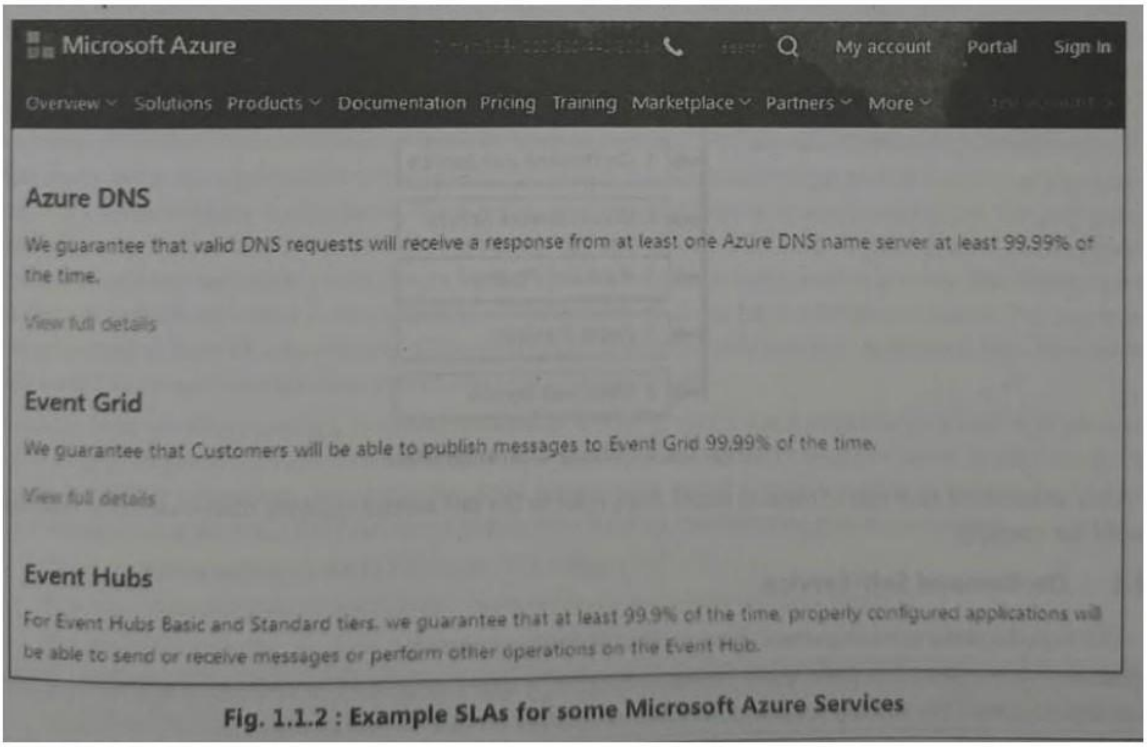


Fig. 1.1.2 : Example SLAs for some Microsoft Azure Services

5. Resources Can Be Released and Reused

When you're done using cloud resources (like storage or servers), **you can give them back**, and they become available for someone else to use.

Example:

Imagine you rented 20 GB of storage space for 3 days. When you don't need it anymore, you **cancel your subscription**, and that space goes back into the shared pool. Then, someone else can use it. This way, resources are **used efficiently** and **shared among many users**, just like a library book being returned for others to borrow.

❖ Rapid Elasticity

➢ **Definition**:- **Rapid elasticity** is a key feature of cloud computing that means **you can quickly and automatically scale resources up or down** based on your needs.

- **Elasticity** means that cloud resources can **grow or shrink easily** based on your needs — just like a **rubber band** can stretch and return to its original size.
- Example:
- Suppose you start with **100 GB of storage**.
- A few days later, you need **500 GB more** — the cloud lets you **instantly add it**.
- Later, if you only need **300 GB less**, you can **release it just as easily**.
- This happens **automatically or with very little effort**, and you're only charged for what you use.
- *Why It Matters:-**You're never stuck with too much or too little. The cloud adjusts to your needs, so you don't waste money or run out of resources.***

➢ Measured Service

➢ **Definition:-** Cloud systems automatically control and optimize resource use by

- leveraging a metering capability st some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be monitored, controlled, and reported, providing transparency for both the provider and consumer of the utilized service.
- Cloud billing is quite easy to understand. Just like other services, cloud services are charged based on how much you actually use. A good example is a taxi ride — you're charged depending on the type of taxi, the distance traveled, and the time taken. Similarly, in cloud computing, you are charged based on what you've subscribed to and how much you've used. You don't pay separately for things like hardware, electricity, or maintenance. For example, if you use a virtual computer (server) for 5 hours, you only pay for those 5 hours — nothing more, nothing less.
- To make this possible, cloud providers use special tools to measure how much of their services you're using. This tracking helps them calculate your bill accurately. Different services are measured and billed in different ways. For instance, CPU usage is billed based on the processor's speed and type, while RAM is billed based on how much memory you use. Other factors, like the location of the server (data center), can also affect pricing.
- To keep things transparent, cloud providers usually offer a service catalog — a kind of menu — that lists all the services they provide along with their prices. This helps users understand what they're paying for and manage their usage and budget more effectively.
- **Exam Tip:-** You can remember the 5 characteristics of cloud using the acronym DREAM.
  D- on Demand self-service
  R- Resource pooling
  E-Rapid Elasticity
  A -broad network Access M-Measured service

❖ Cloud Delivery (Service) Models

- Cloud service delivery models describe **how cloud services are provided and used**. To understand this better, let's first think of something we all know — **living in an apartment**. Imagine you have three choices of apartments to live in. Each one offers a different level of setup and responsibility. This is similar to how cloud services work.

➢ Fully Furnished Apartment – Like SaaS (Software as a Service)

- In this case, the apartment is fully ready to live in. It has furniture, lighting, kitchen setup, and all other basic things already provided.
  You just bring your personal items and move in. You don't have to fix or set up anything, and you probably can't change much either.
- Easy to use
  Everything is managed for you
  Less control or customization
- This is like SaaS, where the cloud provider gives you ready-to-use software (like Gmail, Microsoft 365, etc.). You just use the application — no setup or maintenance

required.

➢ Semi-Furnished Apartment – Like PaaS (Platform as a Service)

- Here, you get the basic setup — walls, plumbing, wiring — but you must bring in your own furniture, appliances, and other comforts.
  You have more control and flexibility over how the place looks and feels.
- Basic structure is ready
  You add your own setup
  Some effort needed from your side
- This is like PaaS, where the provider gives you a platform (like servers and tools) to build your applications, but you are responsible for developing and managing your own software on top of it.

➢ Unfurnished Apartment – Like IaaS (Infrastructure as a Service)

- In this case, you only get the bare structure — just the walls and floors. You have to do everything else: install wiring, plumbing, furniture, lighting, etc.
  It's more work, but you get complete control over how everything is set up.
- Full control and flexibility
  High effort and responsibility
- This is like IaaS, where the provider gives you raw computing resources (like virtual machines, storage, and networks), and you do everything else, including installing software and managing systems.

Table 1.3.1 : Comparison between Fully, Semi and Unfurnished Apartment

Comparison Attribute	Fully furnished	Semi-furnished	Unfurnished
What you bring?	Your personal stuff	Your personal stuff + other amenities as you like	Everything except core building infrastructure
Flexibility to change	Least	Somewhat flexible	Flexible
Investment required	Least	To some extent	Significant
Rent / Charge	Highest	Medium	Low
Liability / Commitment	Could be short term	Could be short or medium term	Long-term

❖ Cloud Service Models

- Now that you have a fair idea of service delivery models, let's understand cloud delivery (service) models. As per NIST, there are three cloud service models:

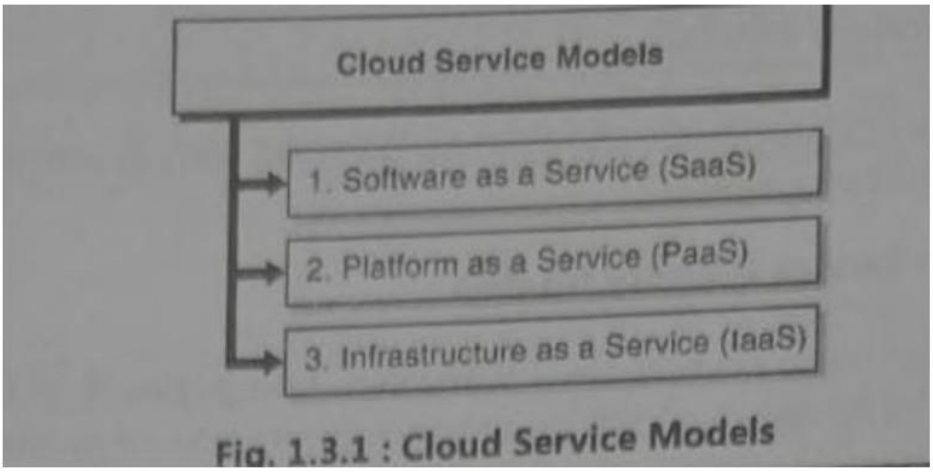


Fig. 1.3.1 : Cloud Service Models

❖ Software as a Service (SaaS)

➢ **Definition:-** Software as a Service (SaaS) is a cloud model where the user gets ready-to-use applications from the provider. You don't need to install, manage, or update the software yourself. The provider runs everything on their cloud servers, and you just use it through the internet.

- These applications are usually accessed through a web browser (like Gmail on Chrome) or a mobile app (like WhatsApp on your phone). You don't need to worry about the servers, storage, or operating systems where the software runs. The only thing you do is use the software.
- It's just like living in a fully furnished apartment. You don't need to buy furniture, set up the kitchen, or do any repairs. Everything is already there. You just move in with your personal belongings and start living. Similarly, in SaaS, you bring your data and start using the application.
- Think about Dropbox. You use it to store your files. You don't manage the servers or storage behind it. Same with apps like Facebook, Instagram, Twitter, or WhatsApp. These companies provide the platform, and you only need to create an account and upload your data. The rest is handled by the service provider.

❖ Advantages of Saas

- **Immediate use**
  SaaS apps are ready-to-use. You don't need to install or set up anything. For example, when you create a Gmail account, you can start sending emails instantly.
- **Low cost**
  You don't have to buy expensive hardware or hire people to maintain it. For example, imagine if you had to build and run Instagram yourself—it would need servers, storage, and a team. Instead, you just download the app and start using it for free.
- **Quick upgrades**
  When a new version comes, the provider updates it for you. You don't have to reinstall or make changes. For example, Google keeps improving Gmail, but as a user, you don't face any trouble—it just works.

- **Scalability (Massive scale)**
  SaaS can grow with your needs. If you need more storage or more users, the provider gives it instantly. For example, Google Drive can give you 2 TB storage in seconds if you upgrade your plan.
- **Access from anywhere**
  You can use SaaS apps from anywhere in the world if you have internet. You can open Google Drive from your laptop at home or your phone while traveling. The experience remains the same.

❖ Disadvantages of SaaS

- **Limited control**
  You can't change how the app works. It's built for everyone, not for individual customization. For example, you can't add your own unique features to WhatsApp—it works the same for all.
- **Internet requirement**
  SaaS needs internet. Without it, you can't use the service. For example, you can't back up files to Google Drive if you are offline. Some apps allow limited offline use, but full features come back only when you reconnect.
- **Vendor lock-in**
  It's hard to switch from one SaaS provider to another. For example, moving your WhatsApp chats to Telegram or your Facebook data to another app is very difficult. This makes you "locked" with one provider.
- **Security and privacy issues**
  Your data is stored on the provider's servers, so you depend on them to protect it. If their system is hacked, your private data can be leaked. For example, Facebook data leaks have exposed millions of users' information.

❖ Platform as a Service (PaaS)

- **Definition** PaaS (Platform as a Service) means the cloud provider gives you a **ready-made development platform**. You don't manage the servers, storage, or operating systems. Instead, you just use the platform to build, test, and run your applications.
- Earlier, developers had to set up everything on their own computer—install programming languages, databases, testing tools, etc. This was slow, expensive, and hard to maintain. As software became bigger and more complex, developers needed faster and easier environments.
- That's where **PaaS came in**. It provides all the tools, libraries, and environments in the cloud, so developers can **focus only on writing code** instead of wasting time setting up hardware and software.
- With PaaS, the cloud provider prepares the infrastructure and development tools (like databases, programming frameworks, servers). A developer simply logs in, chooses what they need, and starts coding. Everything is ready—no need to worry about hardware or OS.
- For example:

- You want to build a website. With PaaS, you get the web server, database, and coding tools already set up. You just write your program and run it.

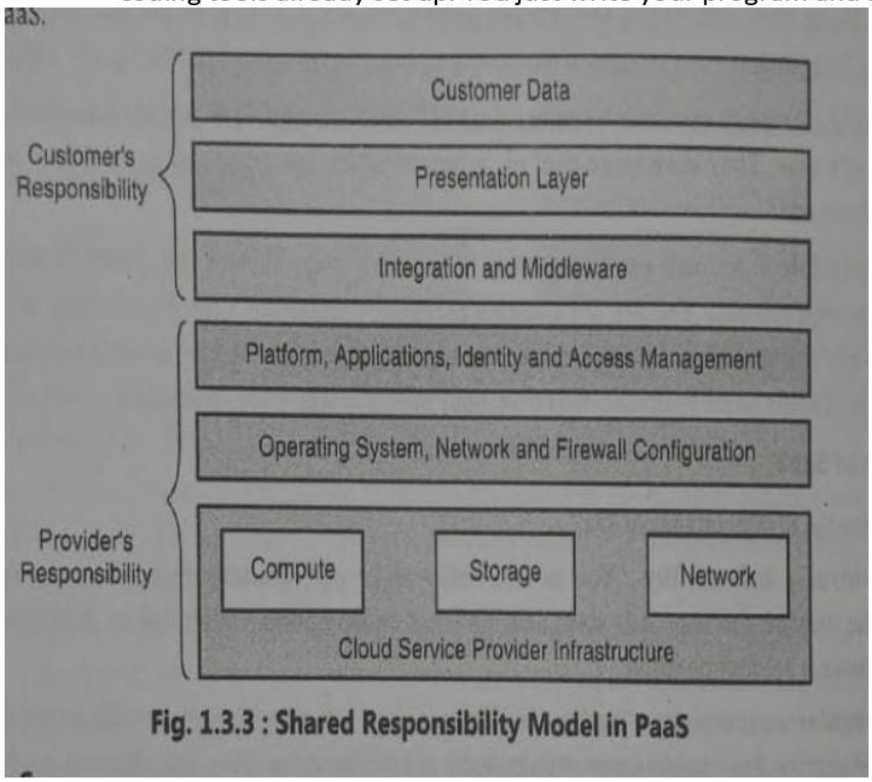


Fig. 1.3.3 : Shared Responsibility Model in PaaS

❖ Advantages of PaaS

- **Quick setup**
  PaaS gives developers a ready-to-use environment. All the tools, libraries, and software packages are already available. This saves time because developers can immediately start coding instead of wasting hours or days setting up systems.
- **Low cost**
  Some development tools are very expensive if you buy them separately. With PaaS, the cloud provider gives these tools as part of the service at no extra cost. This makes software development much cheaper.
- **Always up-to-date**
  The provider regularly updates the platform with the latest libraries and tools. Developers don't need to worry about installing updates or managing versions—they can just focus on building their apps.
- **Scalability**
  If your app becomes popular, PaaS can easily handle the extra load. For example, you may start with 10 users, and later grow to 1,000 or even 1 million. The provider automatically gives you more resources so your app runs smoothly.
- **High availability**
  PaaS providers guarantee that resources like storage or servers will always be available. For example, if you need 100 GB of disk space, the provider ensures it is accessible 24/7. This helps developers host apps with confidence, without worrying

❖ Disadvantages of Paas

- **Limited environments**
  Some applications need very special hardware or software environments. If the PaaS provider does not support these, developers may not be able to build or run such specialized apps.
- **Internet dependency**
  Since PaaS is cloud-based, you must be connected to the internet. If you're offline (for example, on an airplane), you cannot use the platform until you reconnect.
- **Vendor lock-in**
  If you build your app on one provider's PaaS, moving it to another provider can be very difficult. Each provider has its own setup and configurations. This may cause delays or disruptions if you ever want to switch.
- **Security risks**
  Your application code and customer data are stored on the provider's servers. If there's a security breach or hack, your data and your users' data may be at risk. This can cause serious damage to your business.

❖ Infrastructure as a Service (IaaS)

- **Definition:-** IaaS (Infrastructure as a Service) is a cloud model where the provider gives you the **basic building blocks of computing**—like storage, servers, and networking—over the internet. You don't own the physical machines, but you can create **virtual machines (VMs)** and install your own operating systems and applications on them.

- Think of IaaS like **renting a computer lab**. The provider manages the building, electricity, cooling, and physical computers. You just sit down, install your own software, and use the machines as you like.

- In cloud terms:
- The provider manages hardware (servers, racks, power, networking).
- You control everything above the operating system (OS, apps, databases, firewalls, etc.)
- Flexibility of IaaS
- IaaS gives the **maximum flexibility** compared to SaaS and PaaS. You can:
- Install any operating system (Windows, Linux, etc.).
- Run any software you want.
- Set up your own firewalls, storage, and databases.
- It's almost like having your own **data center**, but you don't have to worry about physical servers, cooling systems, or hiring staff to maintain them.
- Examples of IaaS
- **Amazon EC2 (Elastic Compute Cloud)** – lets you rent virtual servers.
- **Amazon S3 (Simple Storage Service)** – cloud storage.
- **Microsoft Azure VM** – virtual machines hosted in Azure.
- **Google Cloud VM** – customizable virtual servers.

- The Fig. 1.3.4 depicts the respective responsibilities of cloud service provider and the customer (user or consumer) in IaaS.

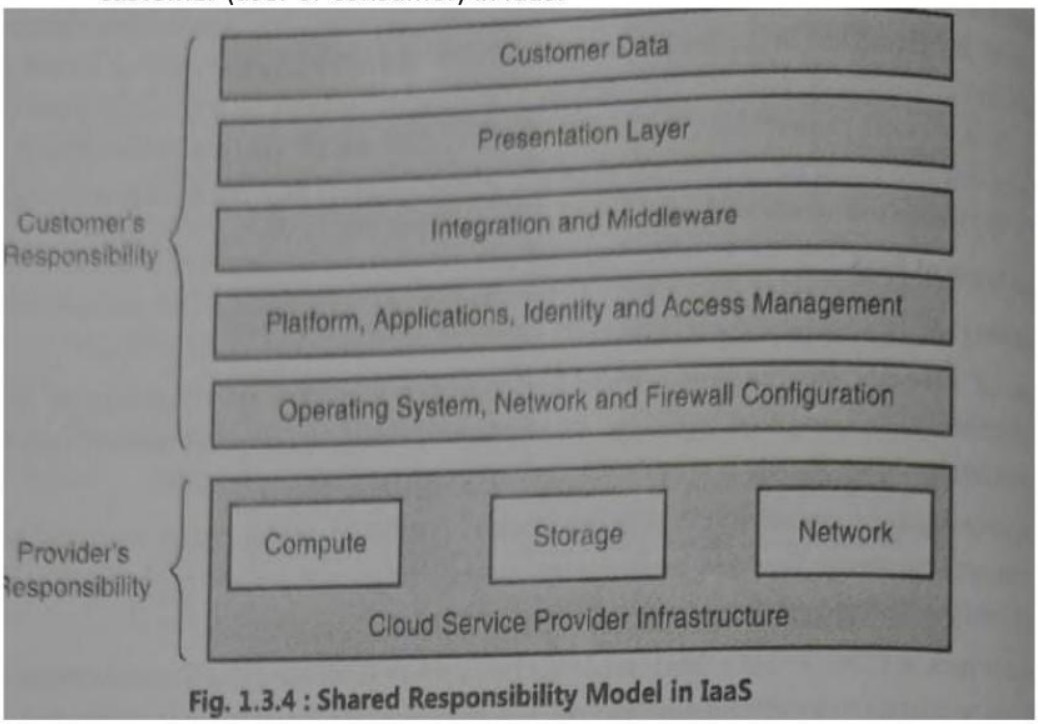


Fig. 1.3.4 : Shared Responsibility Model in IaaS

❖ Advantages of Iaas

- **Full control**
  With IaaS, you get maximum flexibility. You can choose your operating system, configure your own network, select the speed of storage, and even decide from which location (data center) your resources should come. It's almost like building your own data center, but virtually.
- **No hardware management**
  You don't need to worry about buying or managing physical devices like servers, routers, or storage. The provider handles all that. You only manage what's inside your virtual machines.
- **Cost savings**
  Building your own data center requires huge investment (land, servers, cooling, staff). With IaaS, you pay only for what you use. For example, a startup can rent cloud servers at low cost instead of spending crores on its own data center.
- **High performance**
  IaaS providers use very powerful and expensive hardware that most companies cannot afford individually. Since they buy hardware in bulk and share it among many customers, you get top-class performance at a reasonable price.
- **Strong security**
  IaaS providers host big clients like banks and government agencies, so they design their systems with very strict security standards. Even if you're a small company, you benefit from the same high-level security that large organizations demand.

❖ Disadvantages of Iaas

- **High cost if not managed**
  IaaS is the most expensive cloud option because it gives you maximum resources and flexibility. If you don't monitor your usage properly, the bill can become very high.With IaaS, you must manage everything above the operating system. This means you need to set up firewalls, install security patches, and manage access rules yourself. This is known as the **shared responsibility model**.
- **Legal and compliance issues**
  In some industries (like healthcare, finance), storing data on cloud providers can have legal implications. Organizations must follow compliance rules like **PCI DSS, HIPAA, ISO**, etc. That's why providers get certifications, but still, companies must carefully check the rules.
- **Internet requirement**
  Just like SaaS and PaaS, IaaS also needs a stable, high-speed internet connection. Without it, you cannot use or manage your cloud resources properly

➢ Comparison between SaaS, PaaS, IaaS

Table 1.3.2 : Resource Management-wise Comparison between IaaS, PaaS and SaaS

Resource	Your Datacentre	IaaS	PaaS	SaaS
Data	You manage	You manage	You manage	You manage
Application	You manage	You manage	You manage	Provider manages
Runtime / Middleware	You manage	You manage	Provider manages	Provider manages
OS	You manage	You manage	Provider manages	Provider manages
Virtualization	You manage	Provider manages	Provider manages	Provider manages
Physical Servers	You manage	Provider manages	Provider manages	Provider manages
Physical Storage	You manage	Provider manages	Provider manages	Provider manages
Physical Networking	You manage	Provider manages	Provider manages	Provider manages
Datacentre Operations	You manage	Provider manages	Provider manages	Provider manages

II. Characteristics-Wise Comparison

Table 1.3.3 : Characteristics-Wise Comparison between IaaS, PaaS and SaaS

Characteristics	IaaS	PaaS	SaaS
Used By	Organisations	Developers	End users
Used for	Building datacentre	Application Development	Application consumption
Costs	High	Medium	Low
Flexibility	High	Medium	Low
Provider responsibility	Least	Medium	High
Your responsibility	High	Medium	Low
Example	Amazon EC2	Amazon RDS	Dropbox

1.3.4 Cloud Pyramid

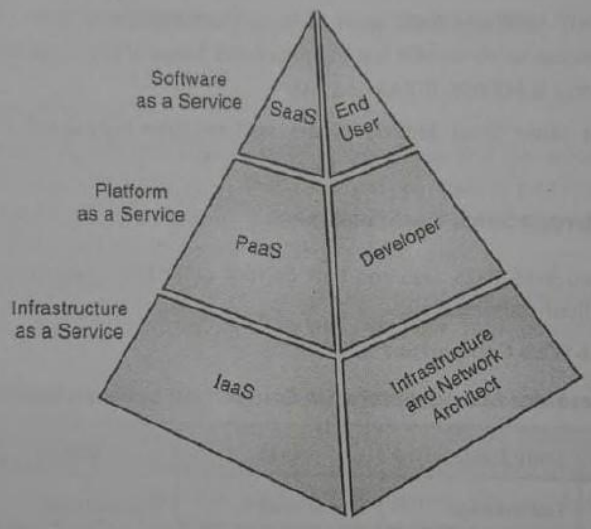
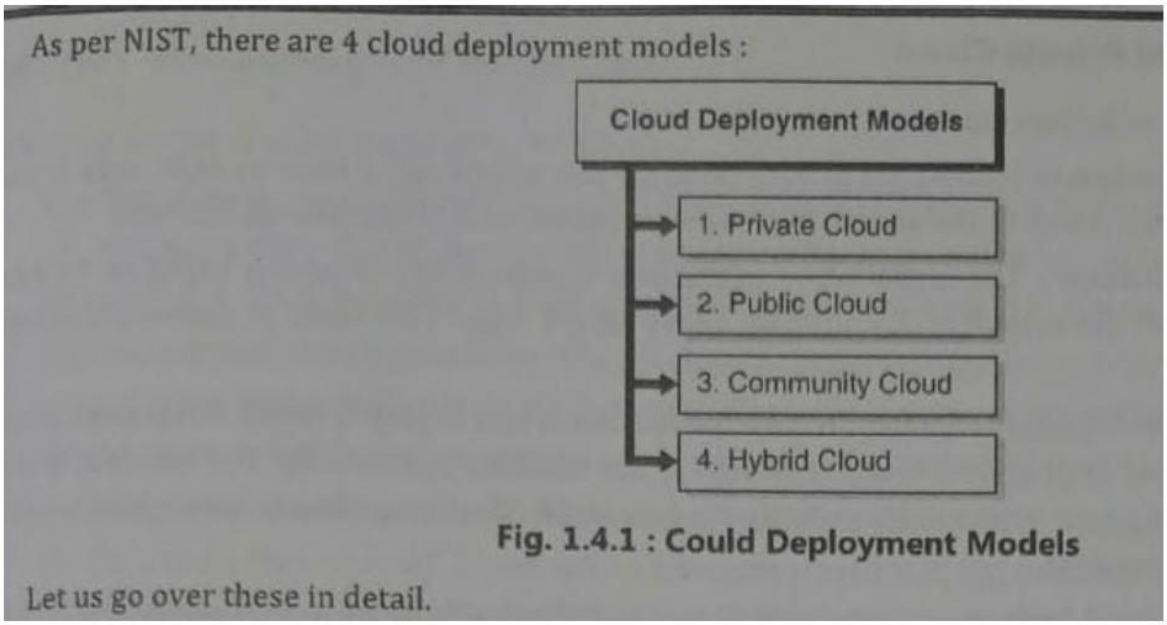


Fig. 1.3.5 : Cloud Pyramid

- You can layover the 3 service models as a cloud pyramid. The bottom most layer is laas. laas has the broadest width and covers everything from OS and above. Next layer is PaaS that packages several libraries and tools for software development and application hosting. The topmost layer is SaaS, which abstracts the underlying complexities and directly delivers the application to the end user.

- Cloud Deployment Models
- the services are required to be consumed over network, sometimes, in certain scenarios, the network connectivity Cloud deployment models describe the location of the cloud services as deployed by the cloud provider. While al restrictive. Let us learn some cloud deployment models.



❖ Private Cloud

- **Definition**: The cloud infrastructure is provisioned for exclusive use by a single organisation comprising multiple consumers (e.g. business units). It may be owned, managed, and operated by the organisation, a third party, or some combination of them, and it may exist on or off premises.

➢ Advantages of Private Cloud

- **High flexibility**
  A private cloud is built only for one organization, so it can be designed exactly as per their needs. The company decides everything—from storage size to security features—making it highly customizable.
- **Strong security and privacy**
  Since the private cloud is used only by one organization, no outsiders can access it. Data usually moves through a private network (not the public internet), which makes it much safer and more private.

- **Best for critical business needs**
  If a company runs important applications that cannot go down (like banking systems or airline booking systems), a private cloud is a good choice because it gives stronger availability guarantees.
- **Better performance**
  In a private cloud, the network is not shared with other companies. This means faster data transfer and better performance. Also, if the cloud is inside the company premises, data travels a shorter distance, further improving speed.
- **Easier compliance**
  Many industries (like finance or healthcare) have strict rules about data. Since resources in a private cloud are not shared with others, it is easier for a company to meet compliance requirements.

➢ Disadvantages of Private Cloud

- **High cost**
  Building a private cloud is very expensive. The company has to buy hardware, set up data centers, and pay staff to manage everything. This makes it costlier than public cloud options.
- **Wasted resources**
  Sometimes, companies overestimate their needs and buy extra resources. But if those resources are not fully used, they go to waste, leading to low efficiency.
- **Limited capacity**
  Unlike public clouds, private clouds cannot scale instantly. If the company sets up 5 TB of storage, it cannot suddenly expand to 6 TB without buying and installing new hardware.
- **Vendor or design lock-in**
  Since building a private cloud takes a lot of money and time, management may feel "stuck" with it. Switching to another model (like public cloud) or changing hardware later becomes difficult.

❖ Public Cloud

- **Definition**: The cloud infrastructure is provisioned for open use by the general public. It may be owned, manage and operated by a business, academic, or government organisation, or
- some combination of them. It exists on premises of the cloud provider.

➢ Advantages of Public Cloud

- **Low cost**
  Public cloud is the cheapest option. Providers buy hardware and resources in bulk at low prices and share them among many customers. This makes services affordable for everyone.
- **High availability**
  Public cloud providers promise very high uptime (availability). If they fail, they compensate customers with refunds or credits under **Service Level Agreements**

(SLAs). This makes them reliable for businesses.

- **Wide variety of services**
  Because public clouds serve millions of users worldwide, they offer a very large catalogue of services—storage, networking, databases, AI, analytics, and more. Almost every IT need can be met with public cloud services.
- **Fast innovation**
  Public cloud providers like AWS, Azure, and Google Cloud keep adding new features and tools quickly. This helps businesses adopt the latest technology without waiting for years.
- **Quick setup**
  It is very easy to start using public cloud. You just create an account, and within minutes, you can launch servers, storage, or applications. You can begin small and grow later.
- **Global reach**
  Public cloud providers have data centers across the world. If your business is in India and you want to serve customers in the UK, you can instantly deploy services in a UK data center for faster performance.

➢ Disadvantages of Public Cloud

- **Limited control**
  You don't have full control over how the provider manages the infrastructure. If there are service changes or disruptions, you cannot control or prevent them.
- **Less customization**
  Public cloud services are designed for mass use. You can only use them as they are. If you want specific changes or custom features, they may not be possible.
- **Security responsibility**
  Public cloud providers give you tools to secure your resources, but the responsibility is yours. If you misconfigure security, your data may be exposed. Since everything is accessible over the internet, hackers may also try to attack.
- **Internet requirement**
  Public cloud depends on a strong internet connection. Without reliable, high-speed connectivity, it is hard to manage or use your cloud resources effectively.

❖ Community Cloud

Definition: The cloud infrastructure is provisioned for exclusive use by a specific community of consumers from organisations that have shared concerns (e.g., mission, security requirements, policy, and compliance considerations). It may be owned, managed, and operated by one or more of the organisations in the community, a third party, or some combination of them, and it may exist on or off premises.

➢ Advantages of Community Cloud

- **Shared cost**
  Instead of each company building its own private cloud (which is costly), several companies come together and share the investment. This makes it cheaper for each

member.

- **Designed for specific needs**
  Unlike public cloud (which is made for everyone), a community cloud is built for a particular group of organizations. For example, hospitals can create a healthcare community cloud with services tailored to medical needs.
- **Easier compliance**
  Since all organizations in the community belong to the same industry, they usually have similar legal and regulatory requirements. The community cloud focuses only on those rules, making compliance easier to achieve.

❖ Disadvantages of Community Cloud

- **Lack of trust**
  Community clouds are usually shared by companies in the same industry, who may also be competitors. Because of this, there can be trust issues, and some companies may hesitate to join or fully adopt it.
- **Limited resources**
  Community clouds do not scale as much as public clouds. The capacity is limited to what the group planned initially. If one company suddenly needs more resources, it may not be possible to provide them instantly.

❖ Hybrid Cloud

Definition: The cloud infrastructure is a composition of two or more distinct cloud infrastructures (private community, or public) that remain unique entities, but are bound together by standardized or proprieser technology that enables data and application portability (e.g., cloud bursting for load balancing between clouds

➢ Advantages of Hybrid Cloud

- **Reuse existing infrastructure**
  With a hybrid cloud, a company can continue using its own data center while also taking advantage of public or private cloud services. This avoids wasting existing investments and makes the transition to cloud smoother.
- **Better security**
  Sensitive data or critical operations can stay in the company's own secure data center, while less-sensitive workloads can run in the cloud. This way, organizations balance strong security with cloud benefits.
- **Scalability (Cloud bursting)**
  If demand suddenly increases, the extra load can "burst" into the cloud without overloading the local data center. When demand goes down, those extra cloud resources are released. This saves money since you pay only for temporary extra usage.
- **Flexibility**
  Hybrid cloud allows businesses to design their system according to their needs. For

example, specialized hardware can stay in-house, while regular projects run on the cloud. This mix offers the best of both worlds.

➢ Disadvantages of Hybrid Cloud

- **Complex setup**
  Building a hybrid cloud requires advanced technical skills in networking, hardware, and cloud. It is more complicated than using only public or private cloud.
- **Interoperability issues**
  Your own data center and the cloud may use different hardware, speeds, or management tools. Making them work together smoothly can be difficult and may require ongoing adjustments.
- **Security management**
  Data and applications move between the local data center and the cloud. This requires strong security controls and access policies. If not managed carefully,

1.4.5 Comparison of Cloud Deployment Models

Table 1.4.1 : Comparison between different Cloud Deployment Models

Comparison Attribute	Private / Community	Public	Hybrid
Infrastructure Managed By	Organisation or Third-party provider	Third-party provider	Both organisation and Third-party provider
Infrastructure Owned By	Organisation or Third-party provider	Third-party provider	Both organisation and Third-party provider
Infrastructure Location	On-premises or Off-premises	Off-premises	Both On-premises or Off-premises
Consumed by	Exclusively trusted only	Anyone (Untrusted)	Both Trusted and Untrusted
Costs	High	Low	Medium
Time required to setup	High	Low	Medium

1.4.6 Summary of Cloud Characteristics, Service Model and Deployment Model

Whatever you have learnt in the previous sections, can be summarised in one simple block diagram.

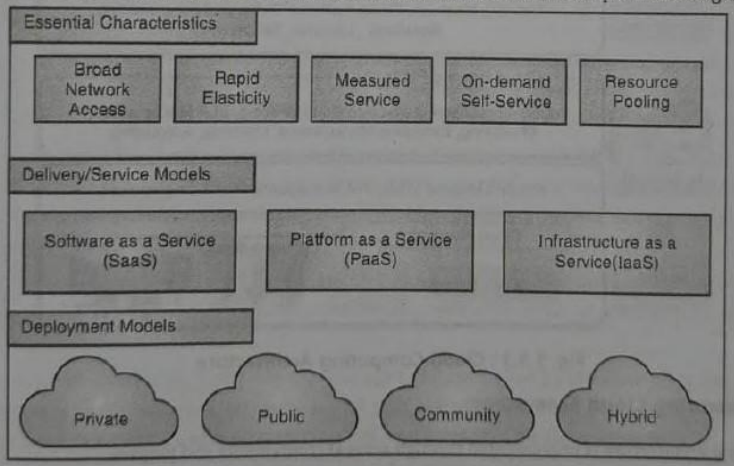


Fig. 1.4.2 : Summary of Cloud Characteristics, Service Model, and Deployment Model

sensitive data could be at risk.

Layers of Cloud Architecture Implementation

Cloud architecture implementation consists of several layers. On top of these layers is the browser running desktops and mobile devices which a user uses to access applications hosted on the cloud environment.

Cloud user uses Cloud services and applications which form the next two layers. These services and application run on software platforms (e.g. Oracle, SAP, Net etc) which forms the next layer of the cloud architect Further down the architecture comprises of infrastructure layer having servers, data base, storage, CPU etc

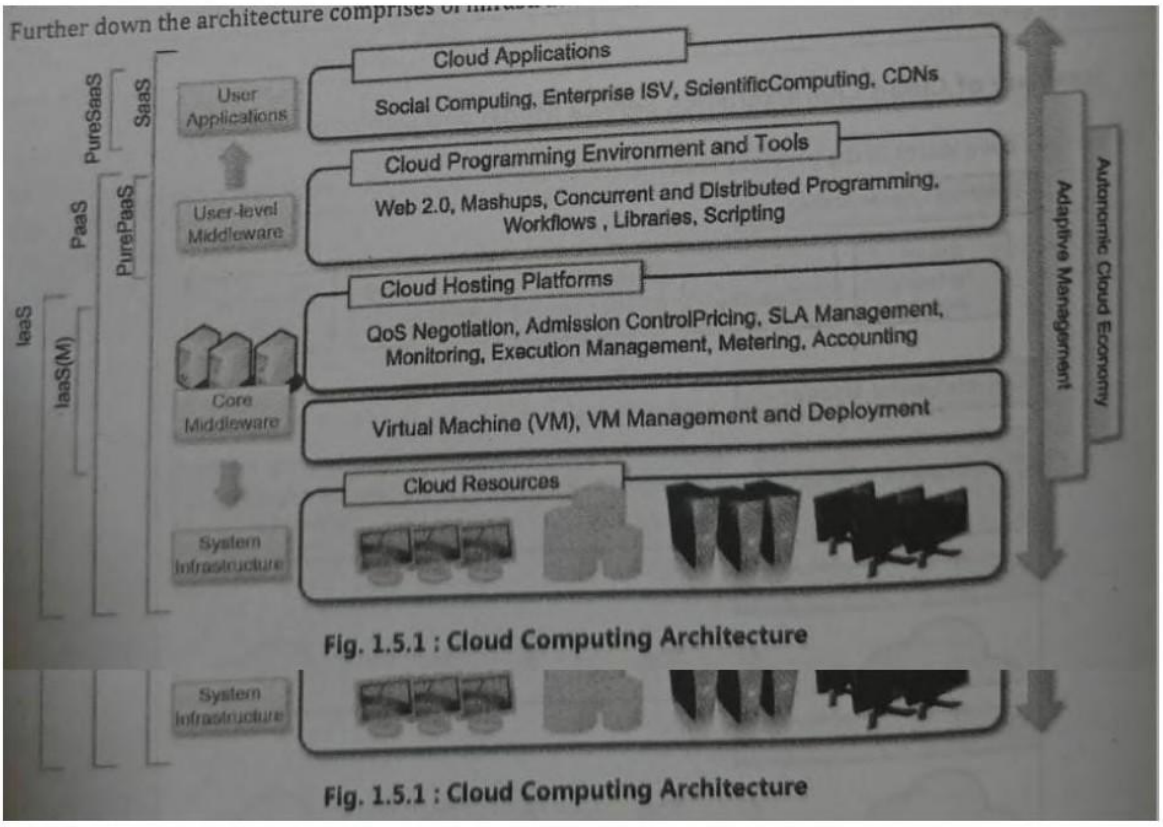


Fig. 1.5.1 : Cloud Computing Architecture

Fig. 1.5.1 : Cloud Computing Architecture

Understanding Cloud Ecosystem

1. Users / Cloud Subscribers

These are the **end users** who use cloud services.

If you use **SaaS** (like Gmail, WhatsApp), you are the user.

Developers who use **PaaS** to build applications are also users.

Companies renting servers on **IaaS** (like Netflix using AWS) are also cloud users.

2. Cloud Brokers

Brokers act as **middlemen between users and providers**.
They simplify the complexity of the cloud for users.
Example: A car company giving you a navigation system actually acts as a broker—it uses cloud services but gives you a simple interface.
Roles of a broker:
Service Intermediation: Adds extra services like security, performance monitoring, or identity management.
Service Aggregation: Combines services from multiple providers (e.g., integrating data storage + analytics).
Service Arbitrage: Chooses the best services from multiple providers to give users better pricing or performance.

3. Service Conceptualizer

These are the **creators of cloud services**.
They **design and build applications** and then host them on the cloud (using SaaS, PaaS, or IaaS).
Example: A developer building a photo-editing app and hosting it on AWS for users to access.

4. Resource Allocator

This player ensures **resources are properly assigned** between users and providers.
Manages CPU, storage, memory, and network usage.
Ensures the cloud runs smoothly without overloading.
Guarantees **Service Levels (SLAs)** like uptime, availability, and performance.

5. Cloud Providers (The Central Player)

These are the companies that **own and operate the infrastructure**.
Examples: **Amazon AWS, Microsoft Azure, Google Cloud, IBM Cloud**.
Depending on the service model:
SaaS: The provider manages everything (apps + infrastructure). Example: Google manages Gmail completely.
PaaS: The provider manages infrastructure + environment; users build apps. Example: Google App Engine.
IaaS: The provider manages only hardware and virtualization; users control OS, apps, and storage. Example: Amazon EC2.

6. Service Request Examiner and Controller

This system (human or automated) checks incoming **requests for resources** and decides how to allocate them.
Example: If one app suddenly needs more CPU power, the controller decides whether to grant it based on **priority and availability**.

7. Pricer

This component **calculates the cost** of resources used.

Works with different pricing plans (pay-as-you-go, subscription).
Example: AWS charging you hourly for the VM you run.

8. VM Monitor (Virtual Machine Monitor)

Also known as the **Hypervisor**.
It manages all **virtual machines (VMs)** running on physical hardware.
Ensures VMs are isolated, secure, and get the required resources.
Example: VMware, Xen, Hyper-V.
Cloud Architectural Components

1. Two Main Components

- **Front-End**: This is what the **user sees and interacts with**. It includes devices like laptops, mobiles, tablets, and the web browser or app through which the user accesses the cloud. Example: When you log into Gmail on your phone, that's the front-end.
- **Back-End**: This is the **hidden part** where the actual cloud operations happen. It includes servers, databases, applications, and storage systems that process and store your data. Example: The Google servers that store all Gmail emails.

2. Sub-Architectures of Cloud

Cloud computing has three sub-architectures:

a) Business Architecture

Focuses on the **business side** of cloud services.
It includes pricing, service contracts, cost models, and how services are marketed and delivered to customers.
Example: How AWS designs its "Pay-as-you-go" pricing model.

b) Technical Architecture

Focuses on the **technology design**.
It decides which cloud platform to use, how different components (servers, databases, middleware, security tools) connect and work together.
Example: Choosing Microsoft Azure for hosting apps, designing system flow, and ensuring security.

c) Operational Architecture

Focuses on the **practical running of the cloud**.
It covers feasibility, network availability, where the data is hosted, monitoring system performance, and legal aspects of hosting.
Example: Deciding whether data should be hosted in India or Europe depending on legal rules.

---

3. Management in Cloud

Managing cloud services involves:
Event management: Handling alerts, failures, or unusual activity.
Configuration and compliance: Ensuring systems meet legal and security standards.
Provisioning: Allocating resources (CPU, memory, storage) as needed.
Workload balancing: Distributing tasks across servers to avoid overload.
Service integration: Making different services work smoothly together.

---

4. Security in Cloud

Security is one of the most important aspects of cloud computing. It includes:
Identity and access management (IAM) → Making sure only the right people access the system.
Data encryption → Protecting stored and transferred data.
User segregation → Keeping different users' data separate.
VM isolation → Virtual machines must not interfere with each other.
Secure migration → Data should remain safe when moved from one cloud to another.
Virtual network isolation → Keeping each customer's network separate.
Security monitoring → Continuously checking for threats and attacks

- **Security in Cloud**

Cloud security ensures that the cloud is safe, reliable, and trustworthy. It mainly covers:
Access authentication and authorization → Only the right users should be able to log in and use cloud services.
Uninterrupted availability → Services should always be running without downtime.
Confidentiality → Customer data must remain private.
Identity management → Every subscriber's identity should be managed securely.

- **Agile Cloud Infrastructure**

An **agile cloud infrastructure** means a cloud that is flexible, fast, and secure. It provides virtual servers that can be quickly used for:
Development and testing new applications.
Hosting websites.
Running small pilot projects.
Performing research and data analysis.
This type of infrastructure is usually delivered as **IaaS (Infrastructure as a Service)**.

➢ Features of IaaS Security-enabled Cloud

- **Software licensing options**

Users can choose from a catalogue of ready-to-use software provided by the cloud.
No need to buy and install software separately.

- **Flexibility to deploy software**

Users can install open-source software (like Linux tools) or even their own licensed software on virtual machines.

- **Workload choices**

Users can select the right kind of workloads (software products, databases, or services) depending on their needs.

- **Custom server setup**

Users can configure their own virtual servers (x86-based) and storage in the cloud.

- **Enterprise file sharing**

Companies can create private, global, or public **file-sharing systems** using the cloud.

- **Network security features**

Facilities like **VPNs, firewalls, and traffic filtering** keep data and applications safe.

- **Create higher-level services**

Users can even design their own **PaaS (platforms)** or **SaaS (applications)** on top of the IaaS

infrastructure.

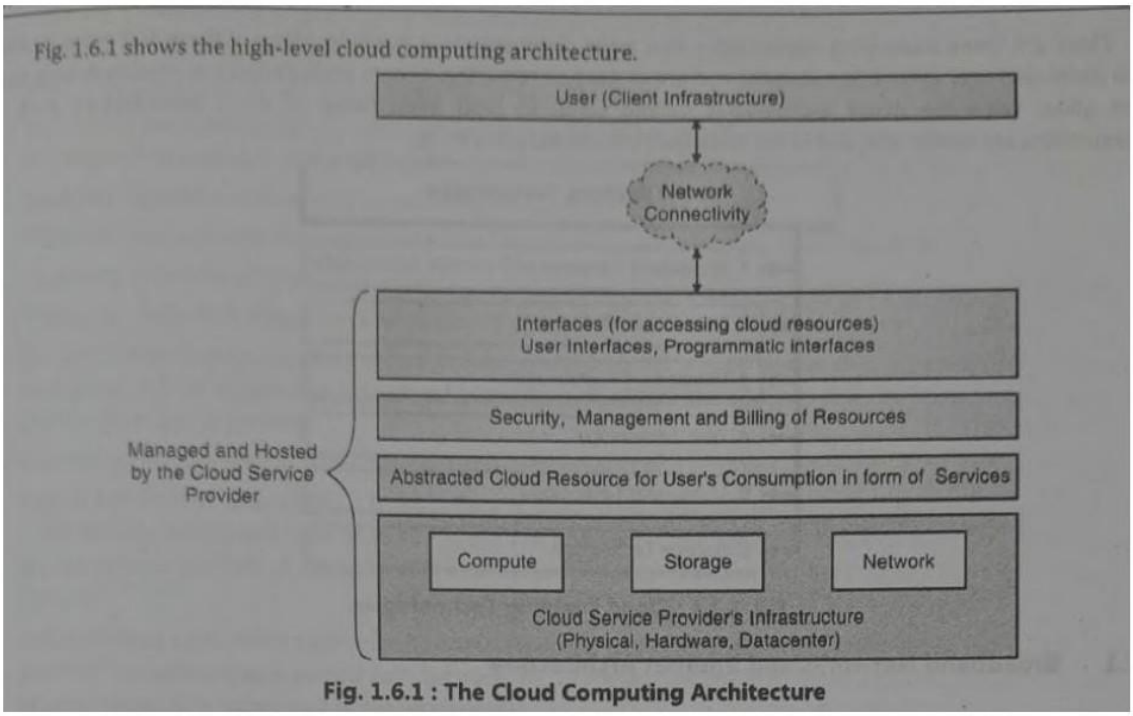


Fig. 1.6.1 : The Cloud Computing Architecture

On the cloud service provider side, you have

➢ Cloud Service Provider's Key Components

- **Infrastructure**

Cloud providers (like AWS, Azure, Google Cloud) build a **global network of datacentres**, divided into **regions** and **zones**.
These datacentres are modern, secure, and designed by experts.
No matter where you are in the world, you can access this infrastructure via the internet.

- **2. Abstracted Cloud Resources**

Instead of giving you direct hardware (like physical servers), the provider gives you **services**.
These services include **servers, storage, networking, and databases**.
They are packaged into different plans, with **uptime guarantees (SLAs)** and **pricing options**.
Example: You can choose a small VM for testing or a large high-performance VM for production.

- **3. Security, Management, and Billing**

Cloud resources are delivered with **security controls**.
You can manage users by creating accounts or integrating with your company's directory.
Billing is **pay-as-you-use**, similar to your **electricity or phone bill**.
Example: If you use 50 GB of cloud storage, you are billed only for that usage.

- **4. Interfaces (How You Access the Cloud)**

Cloud providers give different ways to access resources:
Web Interface (UI) → Easy, user-friendly dashboard.
APIs/CLI → Programmatic access for developers.

Networking Options → You can connect via public internet or secure private connections. Some companies even **extend their datacentre** into the cloud (Hybrid model).

➢ 🔑 Concept Building – Cloud Enabling Technologies

Cloud computing is possible because of a set of **underlying technologies** developed over decades. Some key ones are:

Virtualization → Allows multiple virtual machines to run on one physical server.
Networking → High-speed networks connect users to datacentres worldwide.
Storage Technologies → Distributed storage systems allow storing and accessing huge amounts of data.
APIs & Automation → Enable easy integration and automatic scaling of services.
Security Technologies → Identity management, encryption, and firewalls make cloud safe.

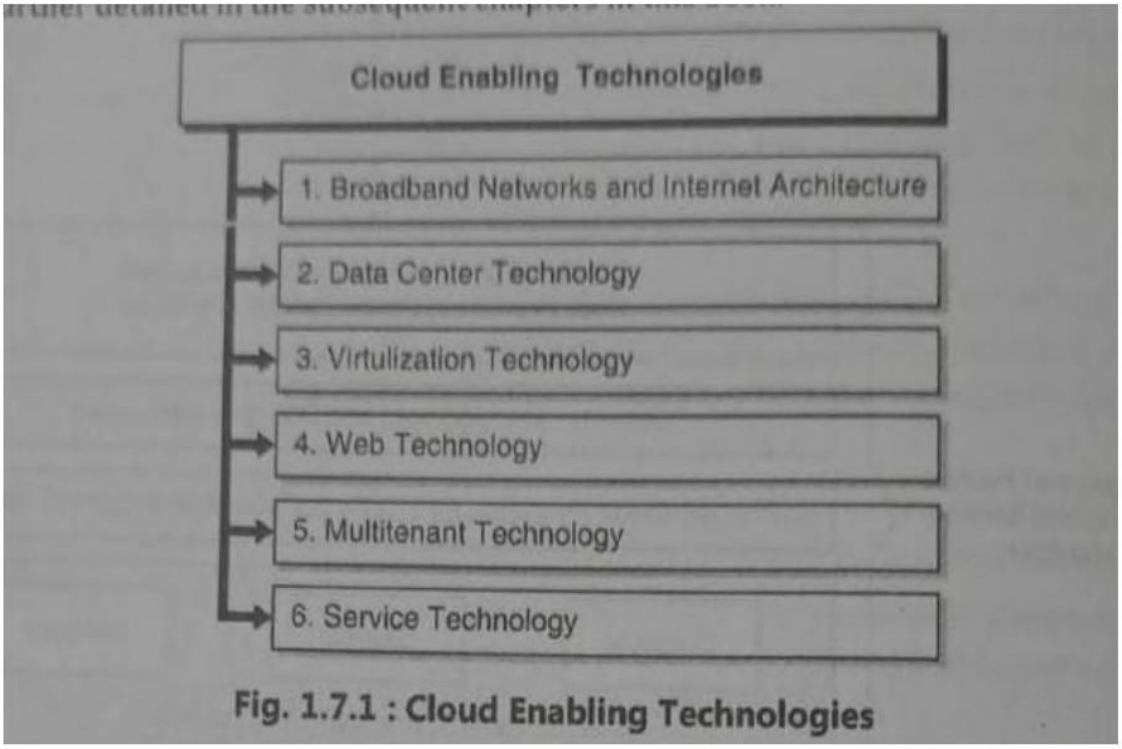


Fig. 1.7.1 : Cloud Enabling Technologies

Broadband Networks and Internet Architecture

High speed networking is the core of utilizing cloud services. Without high speed networking, your da consumption experience could be extremely poor.

The internet is commonly defined as the network of networks. It connects several networks via intercommoned technologies and gives you an experience of worldwide connectivity. The Internet architecture is based various protocols such as HTTP and TCP/IP. Each network follows the protocols and thus can send and recue data packets that are switched and routed through several points.

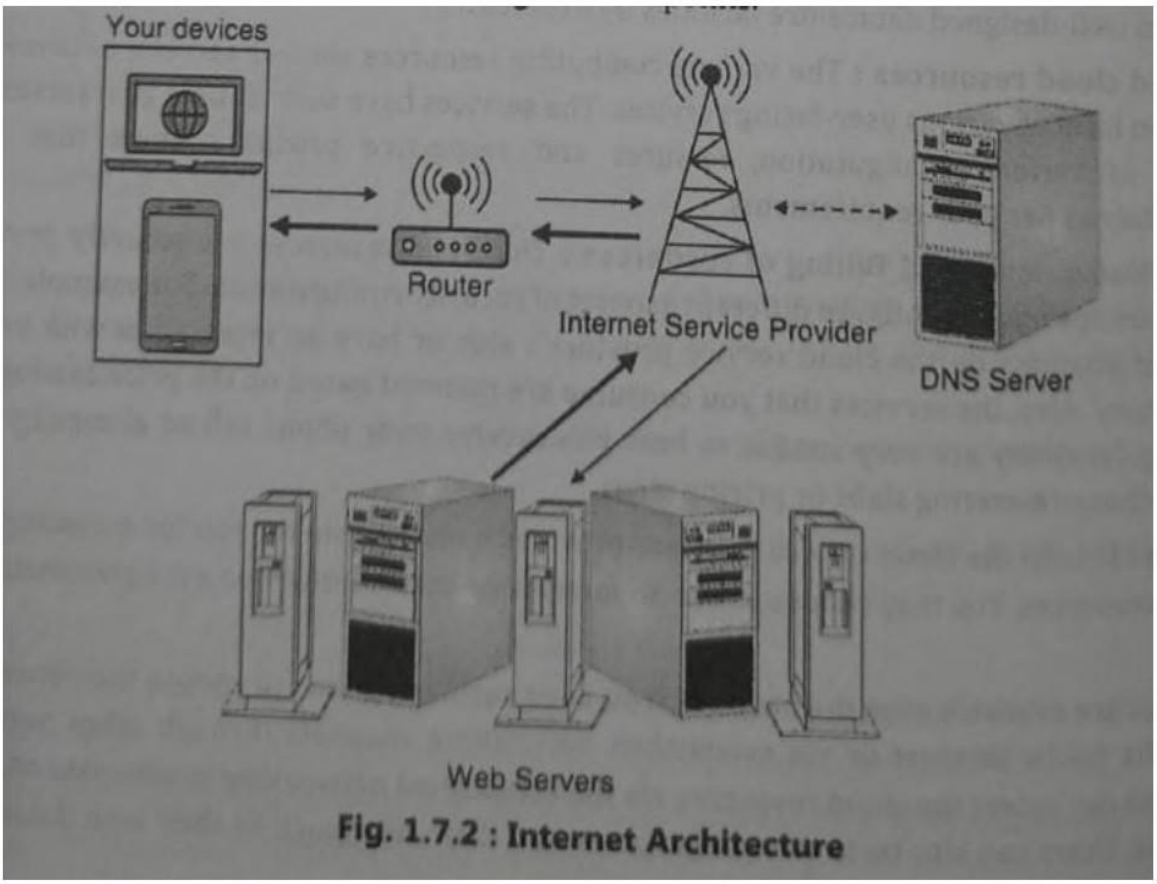


Fig. 1.7.2 : Internet Architecture

➢ Data Centre Basics

1. Facility (The Building)

A **data centre** is a special building where all cloud hardware is kept.

The location is chosen carefully:

Avoid areas with **floods, earthquakes, lightning, or natural disasters**.

Avoid **remote areas** (mountains, deserts, forests) that are hard to reach.

Must have **basic facilities** like water, electricity, and transport access.

---

2. Operational Infrastructure

A data centre must run **24x7 without failure**.

Needs:

Continuous electricity (backup generators).

Security surveillance (CCTV, guards).

Fireproof walls to prevent damage.

HVAC (Heating, Ventilation, Air-Conditioning) to keep servers cool and running properly.

---

3. IT Resources (The Hardware)

Inside the data centre, you'll find:

Servers (computers that run apps and store data).

Racks (to hold servers).

Switches and routers (for networking).

Cables and connections.

Hardware is usually **modular and standardized**, meaning it's easy to install, manage, and replace.
Buying in bulk reduces **costs** and **maintenance burden**.

---

4. Administrative and Operational Staff
Since data centres must be up **all the time (24x7)**, staff are always on duty.
Teams include:
Server administrators.
Storage administrators.
Network engineers.
Application managers.
Electrical engineers.
Maintenance staff (housekeeping, technicians).
Their job: **fix outages quickly** and ensure smooth operation.

➤ 💻 Virtualization Technology

Definition (Simple)
Virtualization = creating a **virtual (software-based) version** of hardware like servers, storage, or networks.
👉 Instead of buying 10 physical servers, you can create 10 **virtual servers** on **one machine**.
Saves **money** (less hardware needed).
Saves **time** (no waiting for physical setup).
Increases **efficiency** (better use of resources).

Analogy
Without virtualization → Each parent drives their kid separately to school → causes **traffic, fuel wastage, more cars needed**.
With virtualization → A **school bus** carries all kids together → saves **fuel, money, and space**.
Similarly, virtualization allows many users to share one machine effectively, just like many kids sharing one bus.

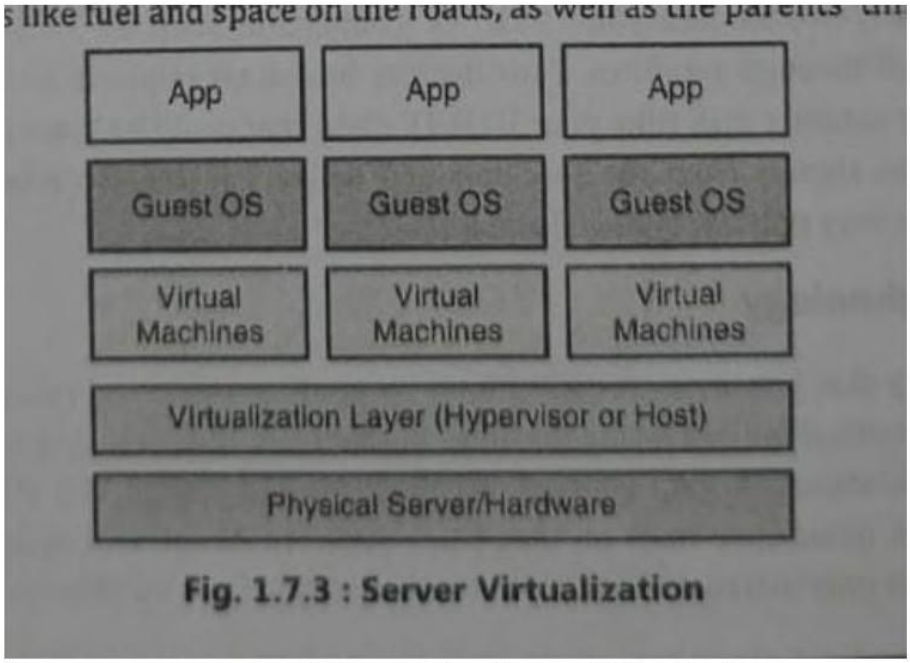


Fig. 1.7.3 : Server Virtualization

There are various types of virtualization. Types of Virtualization:

- Server Virtualization
- Network Virtualization
- Storage Virtualization
- Application Virtualization

➢ Web Technology

🌐 **Web Technologies – Simple Explanation**

Web technologies are the tools and systems that allow us to:
Access websites (through browsers).
View digital content (like videos, images, and text).
Interact with applications (like Facebook, YouTube, or Google Docs).
They include **languages, scripts, protocols, and APIs** that work together in the background.

📖 **Evolution of the Web**

Web 1.0 (First Generation)
Known as **Read-Only Web**.
Websites were **static** (just plain HTML pages).
Only developers or companies could create content.
Example: Early websites in the 1990s (like old Yahoo pages).

Web 2.0 (Second Generation)
Known as **Read-Write Web**.
Users can **interact and create content** (comments, posts, uploads).
Pages became **dynamic** (changed based on user input).
Example: Facebook, YouTube, Wikipedia, Instagram.

Web 3.0 (Third Generation / Current)
Known as the **Semantic Web**.

Uses **AI and data** to understand what users want.
Content is **personalized** (different for every user).
Smarter interactions (voice search, recommendations).
Example: Google Assistant, Netflix suggestions, Chatbots.

➢ 🔑 Key Web Technologies

Browsers
Software used to access the internet.
Examples: **Google Chrome, Microsoft Edge, Safari, Firefox**.
They display websites and run web apps.
HTML (HyperText Markup Language)
The **backbone of web pages**.
Defines the **structure** of a webpage (headings, paragraphs, tables, images).
Uses **tags** (like `<h1>`, `<p>`, `<img>`).
Example: Without HTML, a web page would just be plain text.

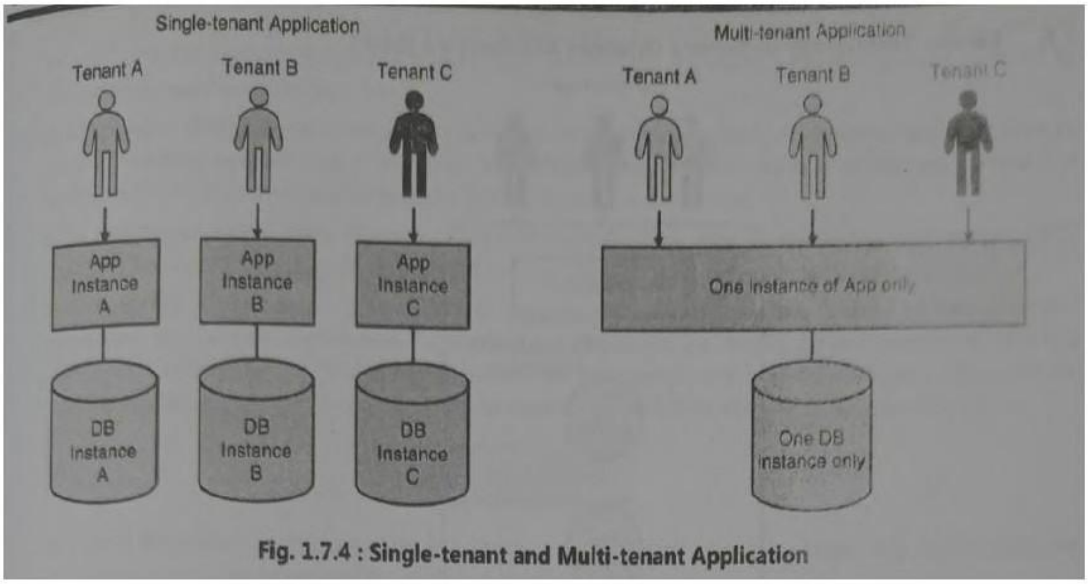


Fig. 1.7.4 : Single-tenant and Multi-tenant Application

➢ ✅ Advantages of Multi-tenancy

- Only **one application instance** serves many users → no need to run separate copies.
- **Easier to maintain & upgrade** → changes are applied once for all.
- **Fast onboarding** → new users can join quickly.

➢ ❌ Disadvantages of Multi-tenancy

- **Complex to build** → designing one system for many users is difficult.
- **Single point of failure** → if the main system fails, all users are affected.
- **Limited customization** → since all users share the same code base

Comparison Attribute	Single-tenant app	Multi-tenant app
Cost of development	Low	High
Complexity of development	Low	High
Ease of maintenance and support	Low	High
Ease of operation	Low	High
Security	High	Low
Resource utilisation	Low	High
Impact of disruption or outage	Low	High

➢ Characteristics of a Service

- **Business-focused** → Represents an activity that gives a specific business outcome.
- **Self-contained** → Works independently.
- **Can depend on others** → May need other services to function.
- **Provides outputs** → Its results can be used by other services.
- **Loosely coupled** → Multiple services can be combined to form an application.
- **Black box** → Users don't see the internal working, only the output.

➢ Core Elements of Service Technology

- **WSDL (Web Services Description Language)**
- Describes how a service works (like a **blueprint**).
- Defines endpoints and message formats.
- **UDDI (Universal Description, Discovery, and Integration)**
- A **directory** where services are registered and discovered.
- Like a **yellow pages** for web services.
- **SOAP (Simple Object Access Protocol)**
- A protocol using **XML** for communication.
- Allows apps on different platforms (Windows/Linux) to talk to each other.
- **REST (Representational State Transfer)**
- A **lightweight style** for web services.
- Uses APIs (HTTP methods: GET, POST, PUT, DELETE).
- Simple, fast, and widely used (e.g., social media APIs).
- **Service Agents**
- Act like **managers/brokers** for services.
- Help in discovery, negotiation, and monitoring.
- Can notify admins if a service fails.
- ---