Cloud Computing

aws Azure Alibaba Cloud

Koustubh Juvekar

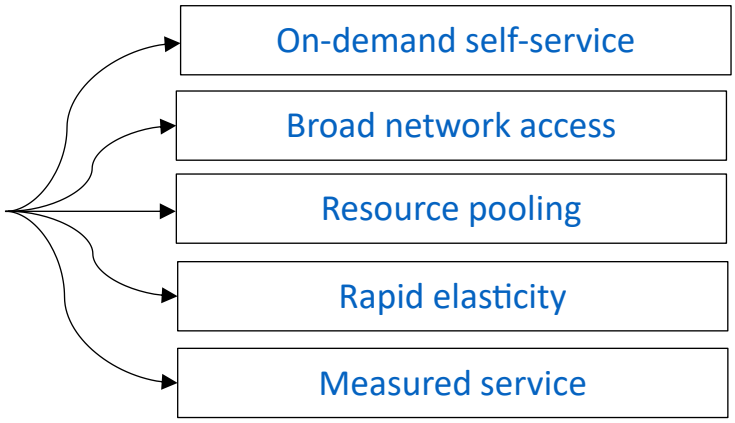
INDEX

* *All company names and examples used are for educational purposes only and are the property of their respective owners.*
* *Blue text is clickable.*
* *Click on the orange corners to get back to index*

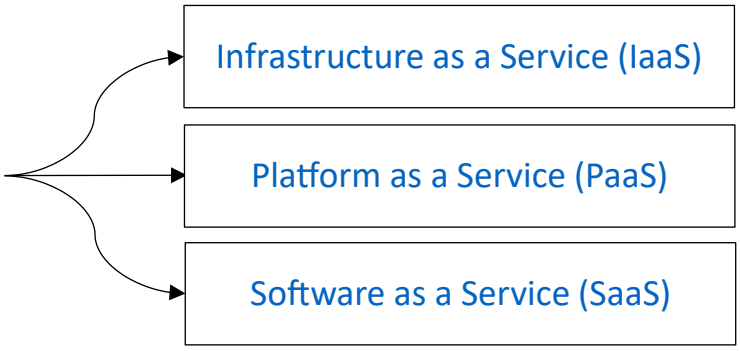
NIST Cloud computing:

"Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computer resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model is composed of *five essential characteristics, three service models, and four deployment models.* "

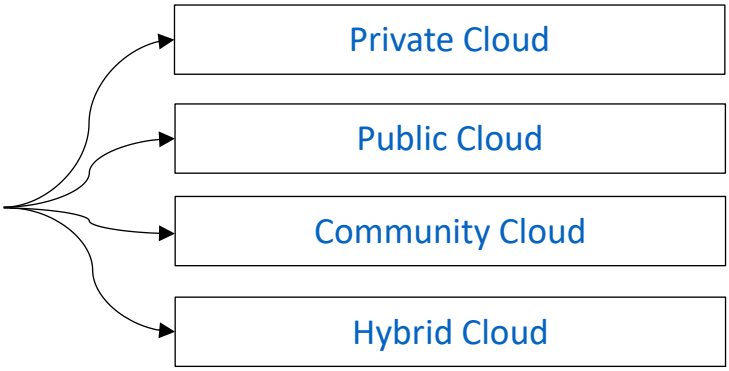
1. Five essential characteristics



2. Three service models



3. Four deployment models



Cloud computing is the on-demand delivery of IT resources over the Internet with pay-as-you-go pricing.

Five essential characteristics in details:

1. On-Demand Self-Service:

→ You can access and use cloud services (like storage, servers, or software) anytime on your own, without needing approval from anyone or a company.

→ Access cloud resources via web portal or API.

→ Users can get cloud resources quickly, within minutes.

→ Supports **pay-as-you-go** model – pay only for what you used.

→ **Available to all** – individuals, startups, and large enterprises.

→ **Easy signup** – usually requires just a credit card and shows estimated cost upfront.

Technical Example:

➲ In cloud computing, a developer can log into AWS or Azure, select what they need, and launch a virtual machine in seconds—no IT admin needed and no need to contact AWS or Azure or any cloud provider. That's self-service in action!

Real Life Examples:

➲ **Like Netflix or YouTube:**

You choose what movie or video you want and watch it instantly; no one needs to approve or set it up for you.

➲ **Buying on Amazon:**

You select what you want (e.g., a laptop), pay, and it ships. No one needs to manually process your order.

2. Broad network access:

→ Access cloud services from anywhere (any location) using any device (phone, laptop, tablet); you just need an internet connection.

→ Use web browsers or software clients to connect to cloud apps easily.

→ Supports BYOD (Bring Your Own Device)

→ Ensures secure and smooth access in today's flexible, remote-friendly IT environments.

Just for Knowledge...

A BYOD (Bring Your Own Device) policy lets employees use their personal devices, like laptops or smartphones, for official work. Many organizations do not allow this due to security risks, but few has begun recently. When personal devices are allowed, they must meet security rules and be used as specified by the policy. This means installing security tools that enable remote access and check the device for malware or compliance. The policy also sets rules for safely storing company data on personal devices and securely deleting it when no longer needed.

How **Broad access network** and BYOD policies connected?

Broad Network Access is an important feature of cloud computing that helps support BYOD (Bring Your Own Device). Many companies now let employees use their personal phones, laptops, or tablets for work to save money on hardware and maintenance. Broad Network Access makes this possible by allowing users to connect to cloud services from any device, anywhere, using the internet. This means employees can safely and easily access work systems whether they are at home, in the office, or traveling.

Technical Example:

- Using the internet, you can securely connect to a virtual desktop (like Amazon Workspaces or Azure Virtual Desktop) from any device; a laptop, tablet, or smartphone.
- It doesn't matter whether you're on a home Wi-Fi, corporate network, or even a public network. This shows Broad Network Access, as the cloud service is accessible from multiple platforms and locations.

Real Life Examples:

- **Google Drive**

  Access your files from a laptop at work, a tablet at home, or a phone while traveling, all through a browser or app.

- **Netflix (as a cloud service consumer)**

  Watch the same content from your Smart TV, mobile app, or browser, using different networks like Wi-Fi, 4G, or even public hotspots.

3. Resource Pooling:

→ Resource pooling means cloud providers share computing resources (like servers, storage) among many users to use them more efficiently.

→ Instead of giving each user a separate server, resources are grouped and assigned as needed based on demand.

→ Also known as **multi-tenancy** – multiple users (tenants) share the same physical infrastructure.

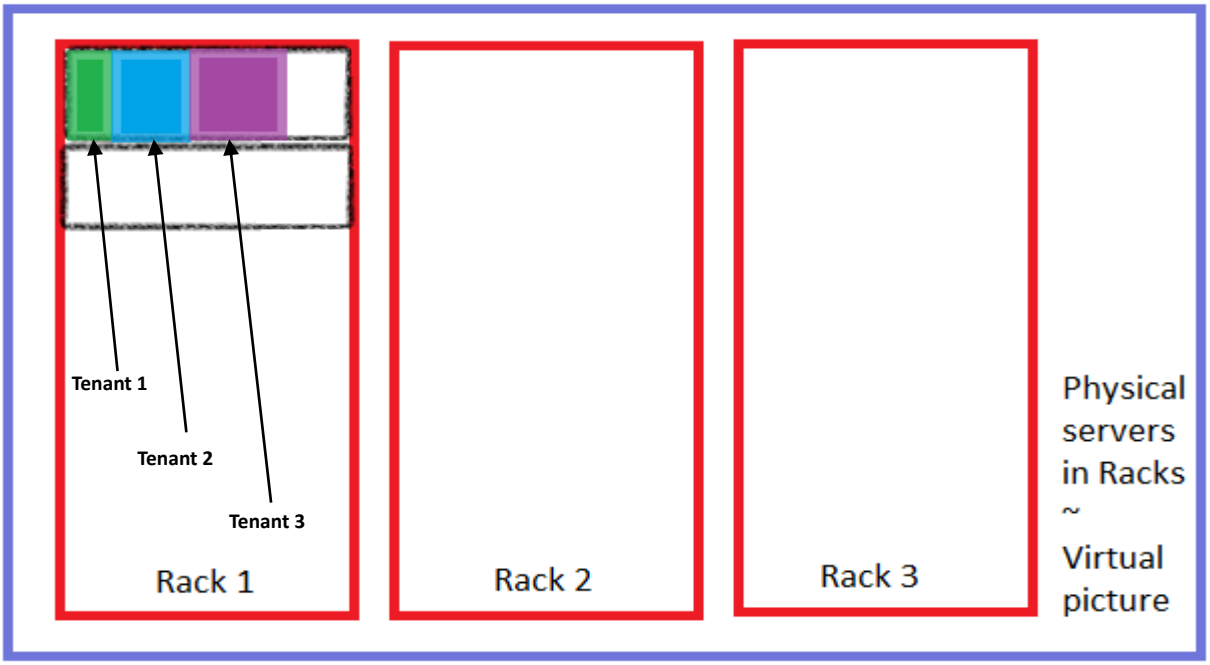
→ Users don't see the physical location of their resources – everything looks virtual.

→ Users may choose a general location (like region or data center), but specific details are hidden; this is called **abstraction**.

→ **Examples of resources:**

- Server pooling
- Storage pooling
- Memory pooling
- Network bandwidth.

Let's understand concept with one example, Server pooling:



In above given example, we don't know exact physical location of resource. But cloud provider has pooled all physical resources together and is using a multi-tenancy model where we can see a single physical server (within a rack, black coloured) runs multiple virtual machines (VMs) using a hypervisor.

The hypervisor ensures isolation between tenants, so despite sharing the same physical server, each VM remains secure and independent.

These VMs belong to different customers (Tenant 1, Tenant 2, Tenant 3) but share the same physical hardware, ensuring efficient use of resources. When many such servers are grouped together, they form a server pool.

Pooling: Pooling means putting things together to share and use them better.

Technical Example:

- In AWS, resources like CPU, memory, and storage are pooled in a large data center. When a customer needs a virtual machine, the system dynamically assigns available resources from the pool, rather than reserving a fixed server for each user.

Real Life Examples:

- **Public Transport**

  A bus or metro system serves multiple passengers rather than each person owning a separate vehicle—shared resources for efficiency.

- **Shared Office Workspaces (Co-Working Spaces)**

  Few companies offer pooled office spaces where multiple businesses share workstations, internet, and meeting rooms instead of renting an entire office.

4. Rapid elasticity

→ In cloud computing, resources like storage and computing power can quickly scale up or down based on need.

→ This is possible because cloud services are software-controlled, not tied to fixed physical hardware.

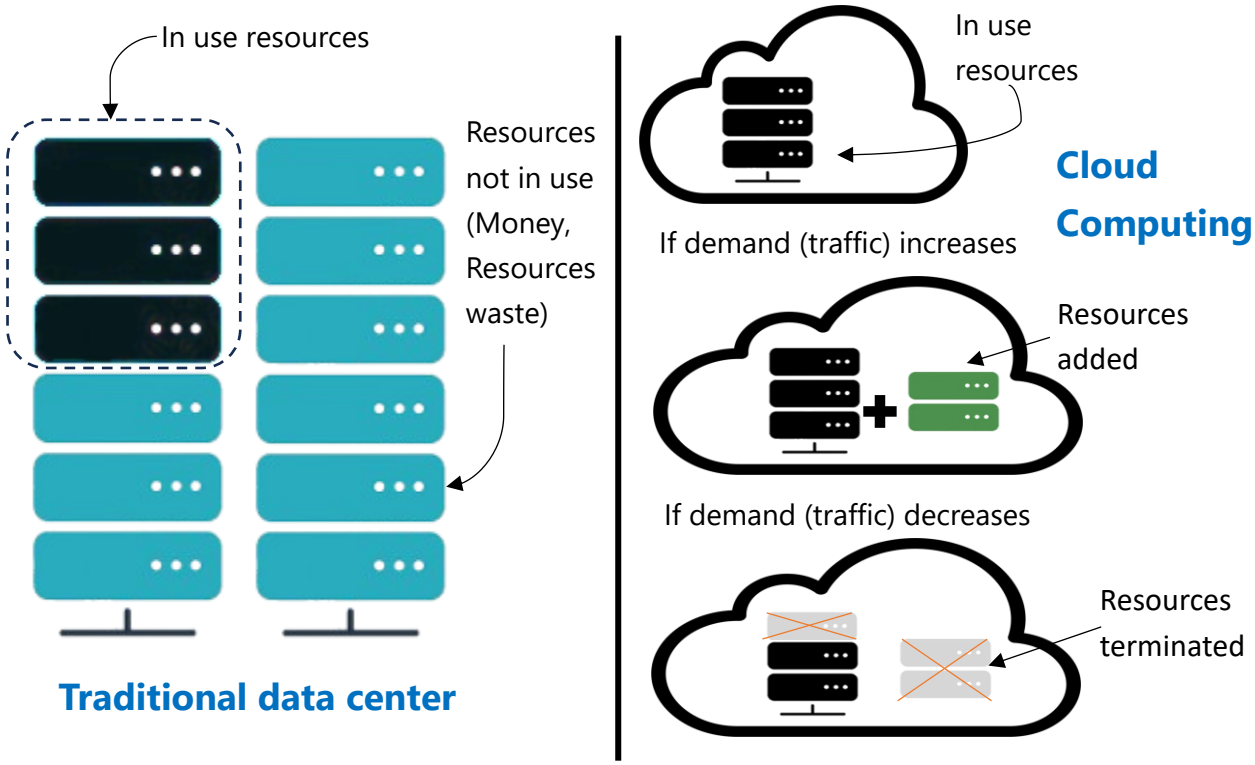
→ You can manually adjust resources via a web portal, or let the cloud provider automatically scale based on demand.

→ Helps save money and maintain performance during high traffic or usage spikes.

→ In traditional data centers, extra hardware had to be set up in advance, even if it was rarely used — leading to waste.

→ With the cloud, you only use what you need, when you need it — reducing costs and waste.

This flexible feature is known as **Rapid Elasticity.**



Technical Example:

- **AWS EC2 Auto Scaling:**

  In AWS, Auto Scaling Groups (ASGs) can automatically launch or terminate EC2 instances based on metrics like CPU usage or network traffic.

  For example: If CPU usage goes above 70%, AWS automatically adds more EC2 instances to handle the load. When CPU usage drops below 30%, it removes the extra instances. This ensures performance and cost-efficiency, an exact demonstration of Rapid Elasticity!

Real Life Examples:

- **E-commerce during a Big Sale (e.g., Amazon on Prime Day)**

  During events like Black Friday or Diwali sales, traffic spikes massively. Cloud systems automatically scale up to handle millions of users and scale back down when the sale ends.

- **Streaming Services (e.g. JioHotstar)**

  When a thrilling moment happens in a match, like a last over finish or when a legendary player like MS Dhoni walks onto the field, millions of users start streaming instantly. Streaming servers automatically scale up during the event and scale down afterward, maintaining performance without wasting resources.

5. Measured service

→ Cloud providers automatically track how much of each resource you use; like storage, CPU, memory, number of VMs, or active user accounts.

→ Usage is monitored and reported in real time, helping both users and providers see exactly what's being used.

→ Enables accurate billing, resource usage tracking, and helps set limits (like for auto-scaling).

→ In traditional data centers, such tracking was difficult and less transparent.

→ This feature supports **pay-as-you-go, charge-per-use** or **pay-per-use** pricing models.

Technical Example:

➲ **AWS Lambda Billing**

AWS Lambda automatically tracks each function invocation's duration (in 100 ms increments) and memory usage, then bills you only for the exact compute time and resources you consumed.

Real Life Examples:

➲ **Electricity Meter**

Your home electric meter measures kilowatt-hours used, and your utility bill reflects exactly how much electricity you consumed.

➲ **Mobile Data Plan:**

Your phone tracks how much mobile data you use (e.g., 2 GB, 5 GB). At the end of the month, you're either charged based on your usage or it's deducted from your plan.

Three service models in details:

1. **Infrastructure as a Service (IaaS)**

   Definition: Infrastructure as a service (IaaS) is the on-demand availability of highly scalable computing resources as services over the internet. It eliminates the need for enterprises to procure, configure, or manage infrastructure themselves, and they only pay for what they use.

2. **Platform as a Service (PaaS)**

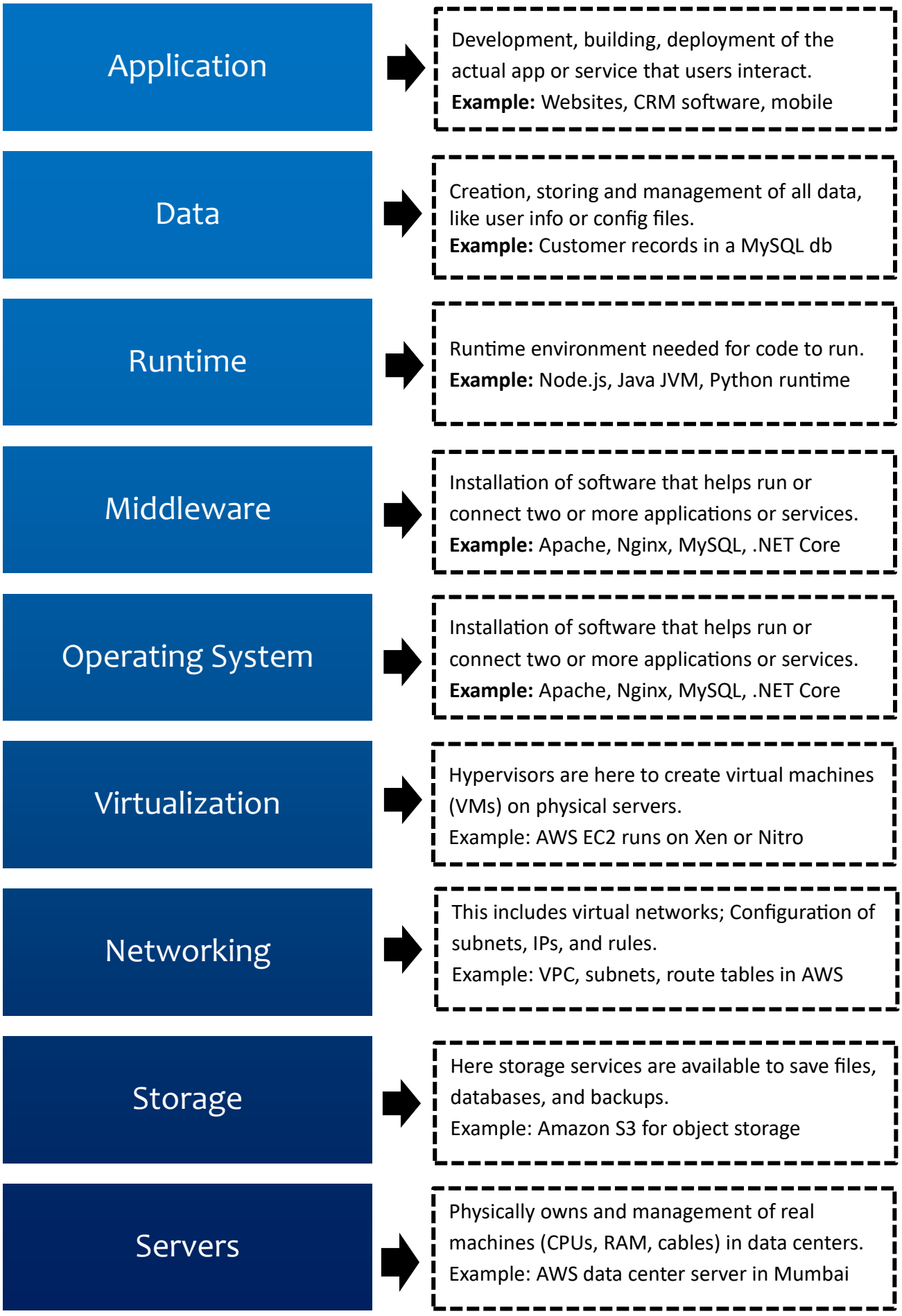
   Definition: Platform as a Service (PaaS) is a complete cloud environment that includes everything developers need to build, run, and manage applications—from servers and operating systems to all the networking, storage, middleware, tools, and more.

3. **Software as a Service (SaaS)**

   Definition: SaaS stands for software as a service. It is a model in which the software is centrally hosted and accessed by the user via a web browser using the internet.

Let's see...

- Every application, whether a simple website or a complex system, relies on below given model to work efficiently in real world. These all layers in the model define what components are needed, who manages them, and how they interact to ensure smooth operation.
- Whether deployed on a personal laptop, a private server, or in the cloud, these layers form a structured model that optimizes performance, security, and scalability.


Application
Development, building, deployment of the actual app or service that users interact.
Example: Websites, CRM software, mobile
Data
Creation, storing and management of all data, like user info or config files.
Example: Customer records in a MySQL db
Runtime
Runtime environment needed for code to run.
Example: Node.js, Java JVM, Python runtime
Middleware
Installation of software that helps run or connect two or more applications or services.
Example: Apache, Nginx, MySQL, .NET Core
Operating System
Installation of software that helps run or connect two or more applications or services.
Example: Apache, Nginx, MySQL, .NET Core
Virtualization
Hypervisors are here to create virtual machines (VMs) on physical servers.
Example: AWS EC2 runs on Xen or Nitro
Networking
This includes virtual networks; Configuration of subnets, IPs, and rules.
Example: VPC, subnets, route tables in AWS
Storage
Here storage services are available to save files, databases, and backups.
Example: Amazon S3 for object storage
Servers
Physically owns and management of real machines (CPUs, RAM, cables) in data centers.
Example: AWS data center server in Mumbai

Above given diagram is a simple service model that is used in real world to host, deploy websites, application, services, etc. What happens in each layer is given in dotted box.

Now, cloud service providers like AWS, GCP, Azure, etc. they took this model and distinguished their services in three types, so that consumer (we) can select service model as per need or requirement.

Let's elaborate...

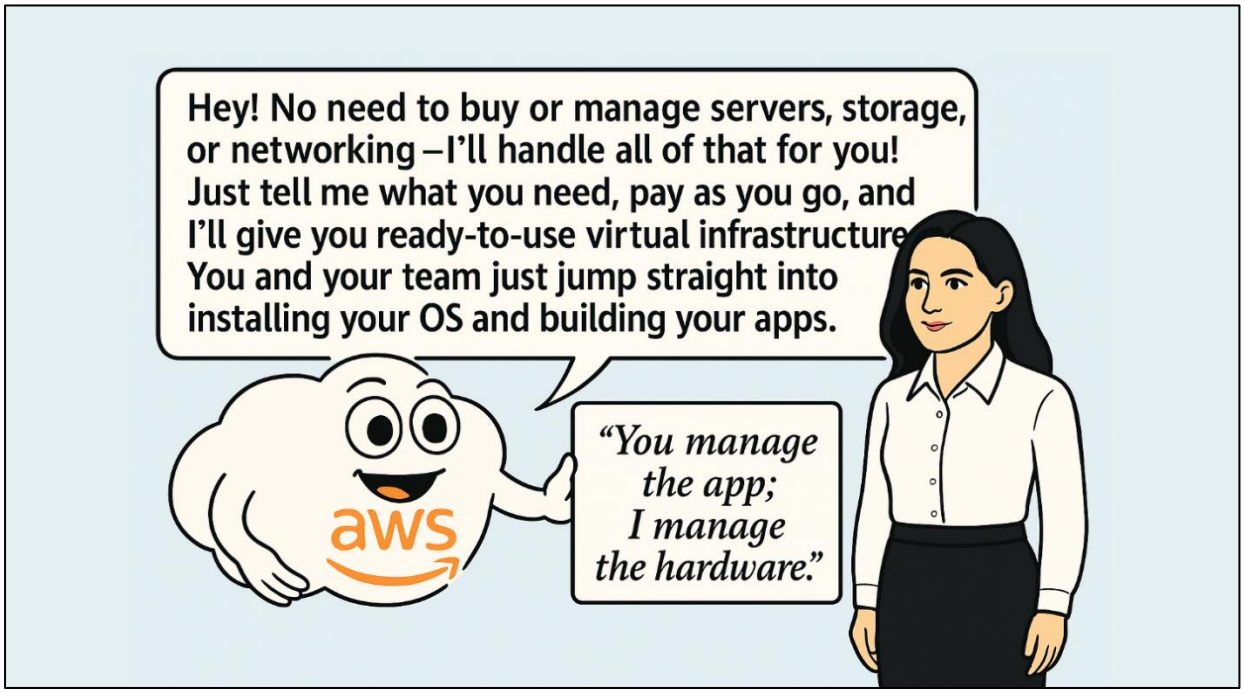
1. On premises / On-site

→ The on-premises deployment model means that **all hardware, software, and data** are hosted inside your company's own building or data center.

→ All the layers and services given in the diagram, those are **set up, maintained, and managed by you or your own IT team**.

- ✓ **Physical setup required** – You must build and manage your own data center (not cloud-based).
- ✓ **Full control** – You manage everything: hardware, software, security, updates, maintenance.
- ✓ **High security responsibility** – Both physical and digital security are fully your responsibility.
- ✓ **High upfront cost** – Large initial investment in:
  - Building or space
  - Servers and networking equipment
  - Cooling systems
  - Power backup and infrastructure
- ✓ **Skilled IT staff needed** – To maintain, monitor, and secure all systems and infrastructure.

2. Infrastructure as a Service (IaaS)

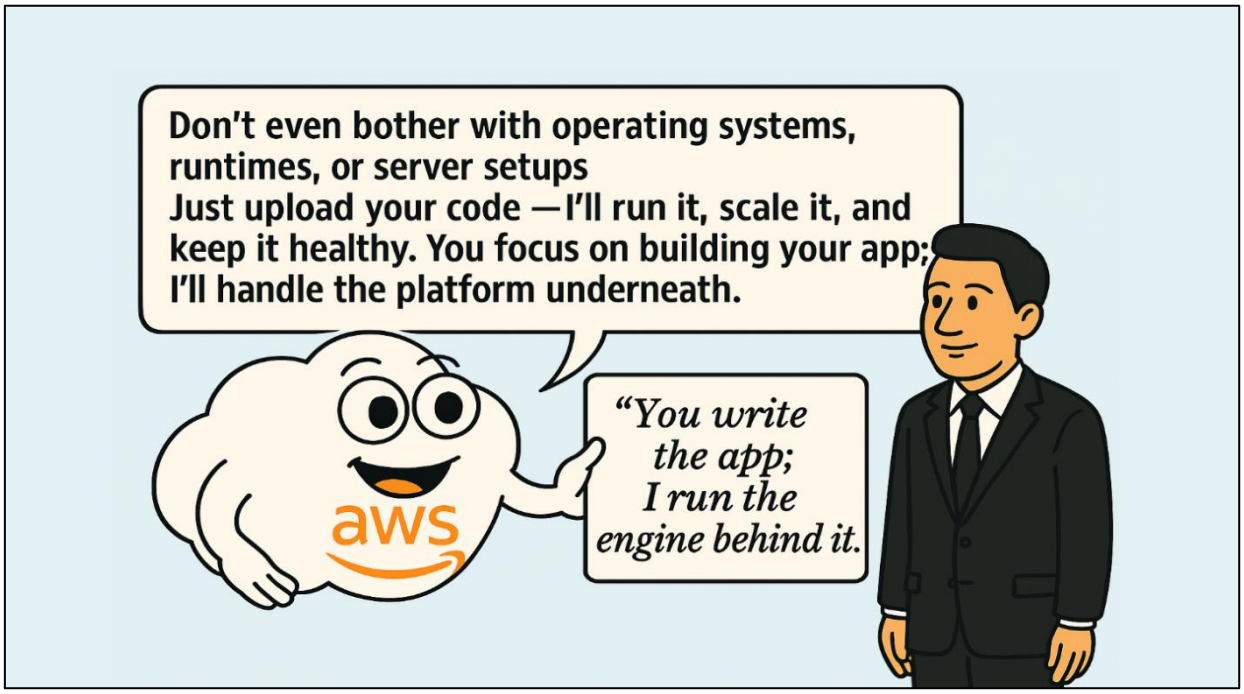


→ Suppose, you need to run a website or an app for your company. But instead of buying your own physical servers and building a data center, you use cloud services like AWS, Azure, or GCP to get virtual servers, storage, and networks over the internet. That's **IaaS**.

→ With IaaS, you get the basic building blocks of IT, like virtual machines, storage, and networking — from a cloud provider.

You can	You don't have to manage
✅ Install your own operating system (like Linux or Windows)	❌ Physical servers
✅ Deploy your apps and data, use your own environment for development and deployment.	❌ Data centers
✅ Configure settings like firewalls or IP addresses.	❌ Hardware maintenance
	❌ Cooling, power, cables

3. Platform as a Service (PaaS)

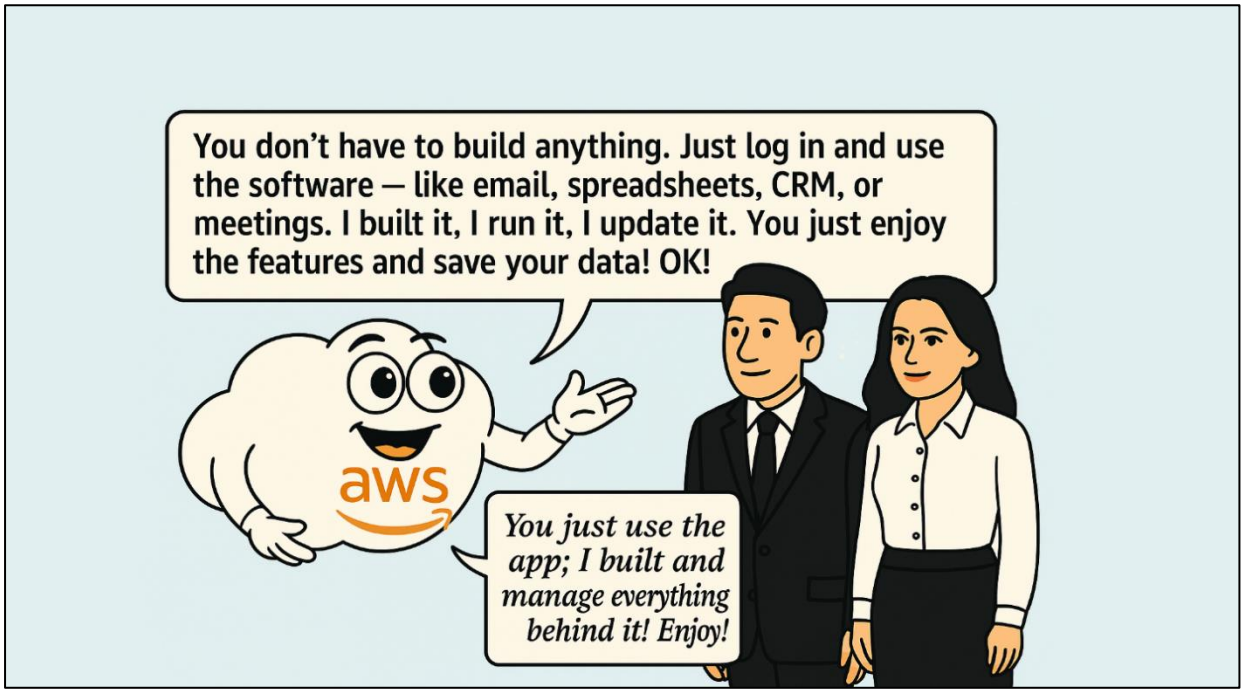


→ You want to build and run a web app, but you don't want to manage servers, OS, or runtime environments.

→ With PaaS, you just focus on writing code ; the cloud provider takes care of everything else behind the scenes; the platform runs it for you.

You can	You don't have to manage
✅ Focus only on your application code	❌ Operating system
✅ Deploy apps easily without worrying about servers	❌ Servers or storage
✅ Use built-in tools for testing, deployment, scaling	❌ Runtime (like Node.js, Java, Python)
	❌ Infrastructure updates or patching

4. Software as a Service (SaaS)



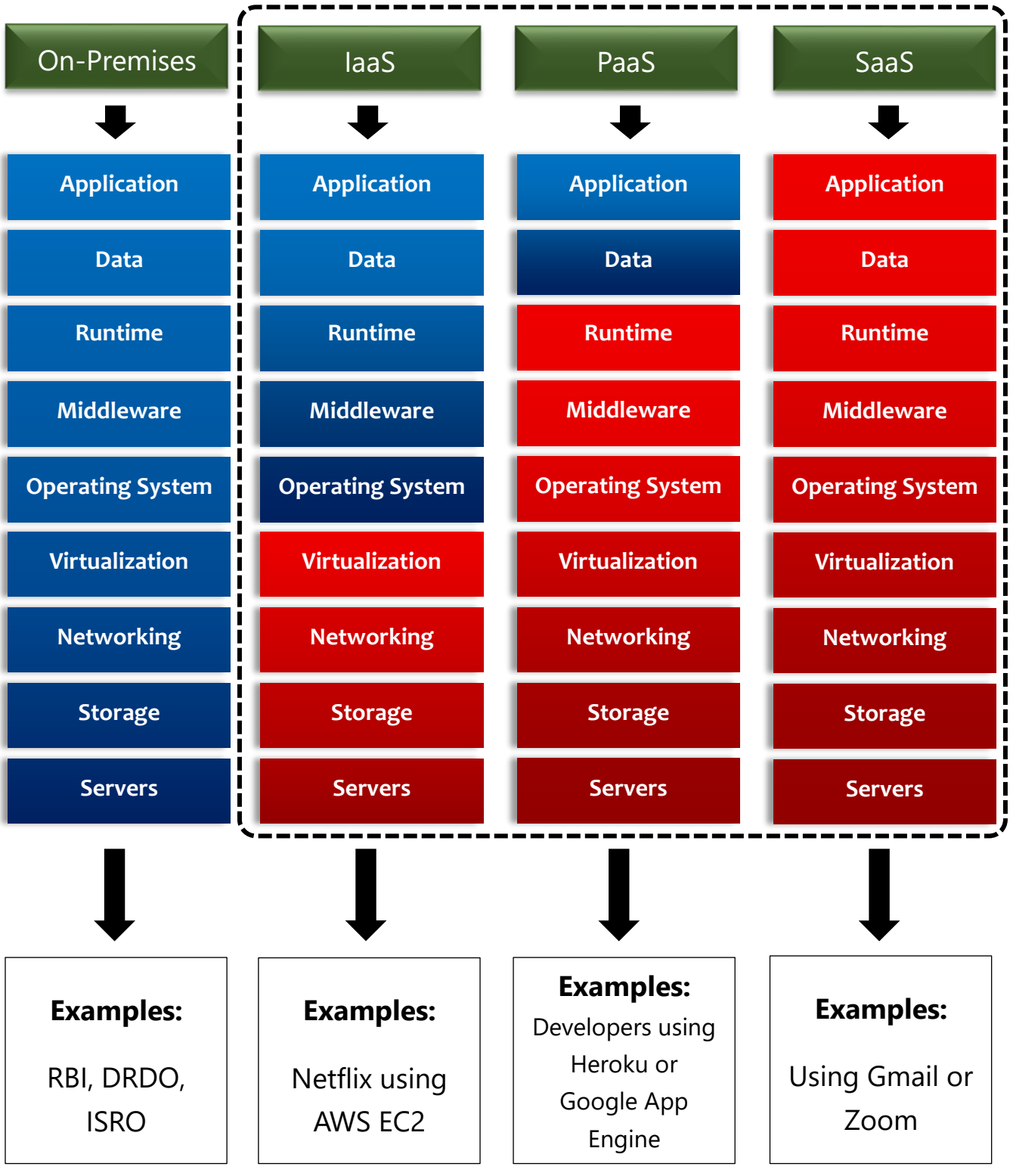
→ You want to use a tool like Gmail, Zoom, or Microsoft Word; but you don't want to install, update, or maintain anything.

→ SaaS gives you fully built applications over the internet.

→ You just sign in and use it; nothing to manage.

You can	You don't have to manage
✅ Access the software from your browser	❌ Servers
✅ Use it instantly (no installation)	❌ OS
✅ Save and work with your own data	❌ Application code
	❌ Updates, bug fixes, scaling

Final Diagram of On-premises, IaaS, PaaS and SaaS:



→ You (Consumer) manage → (any individual, Organization, company)

→ Service provider manages → (AWS, GCP, Azure, etc.)

Remember all layers like "A Day Reminds Me Of Very Nice Sunny Skies."

Four deployment models in details:

1. Private Cloud:

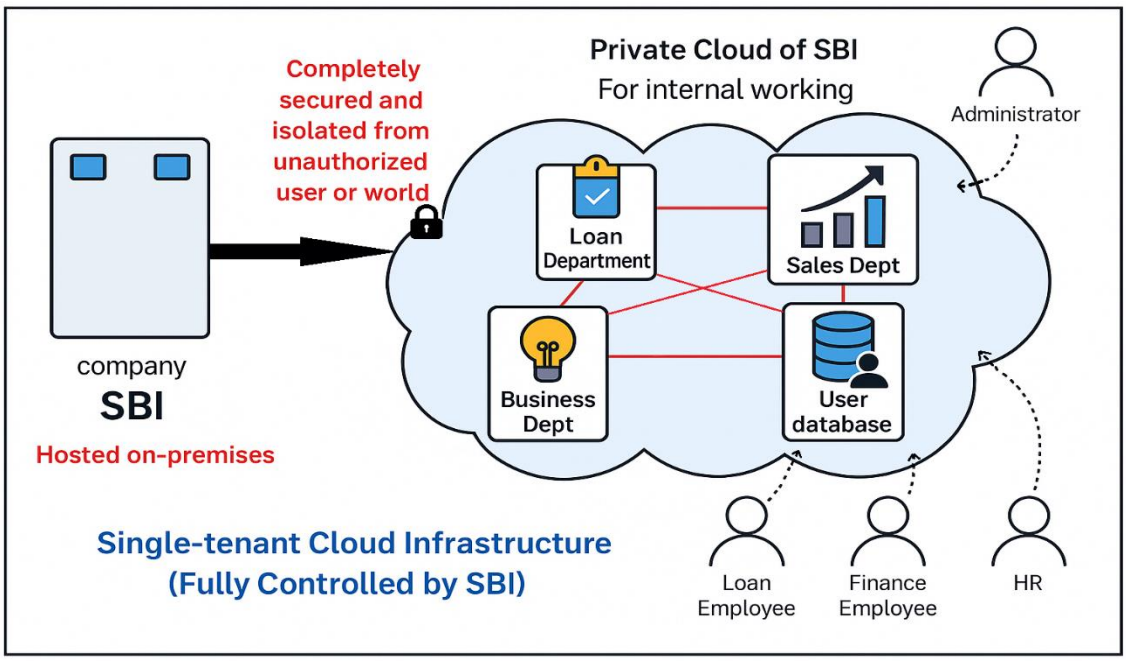
Definition: A private cloud is a cloud computing environment dedicated to a single organization.

→ Cloud infrastructure provides things like CPU and storage, which you can set up yourself whenever you need using a self-service portal.

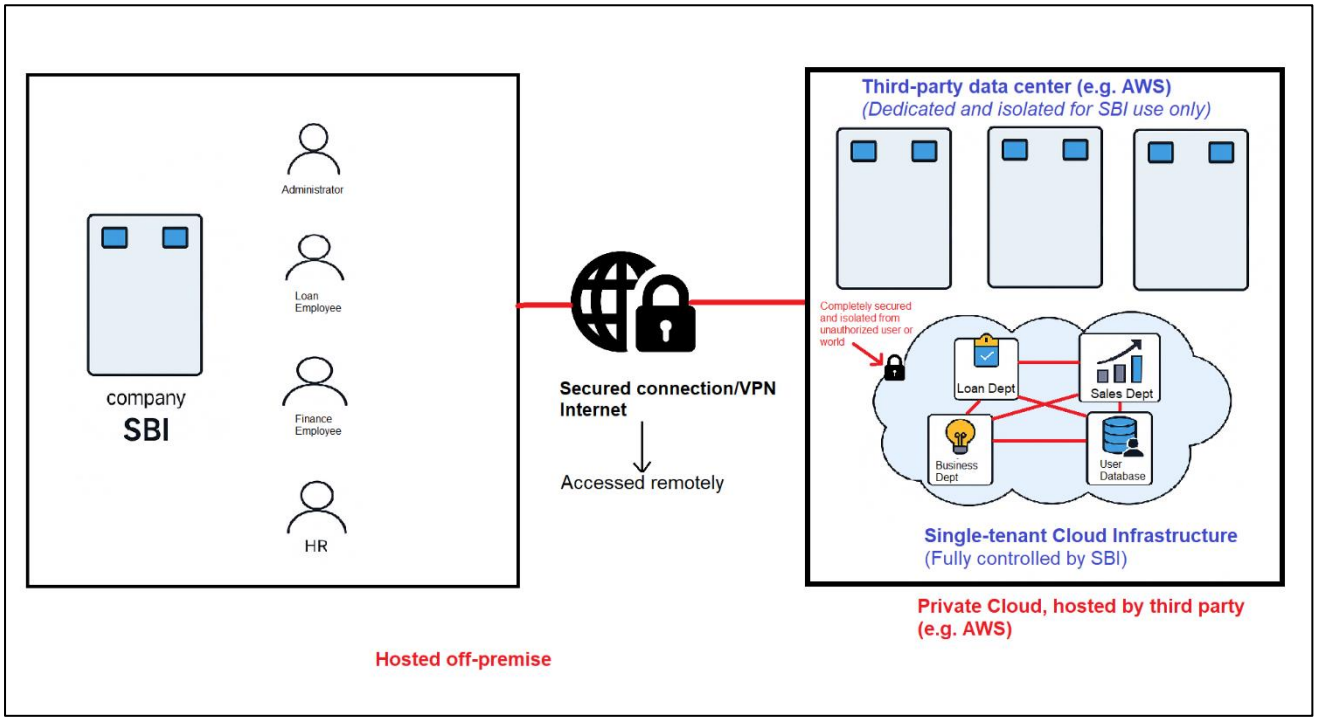
→ In a private cloud, all resources are isolated and in the control of one organization. So, the private cloud is also called the internal or corporate cloud.

→ Deployment Options:

- **On-premises** – hosted in the organization's own data center.
- **Off-premises** – hosted by a third-party provider, but still dedicated and private.



Hosted on Premises *(SBI has its own physical servers and infrastructure inside its office/data center.)*



Hosted off Premises *(SBI does not own physical servers and infrastructure, it is hosted by a third-party provider (AWS), but still dedicated and private..)*

Example:

The **Government of India** runs its own private cloud called **MeghRaj**, managed by NIC, to host sensitive government applications and citizen data securely.

2. Public Cloud:

Definition: A public cloud is a cloud computing environment open for use by the general public.

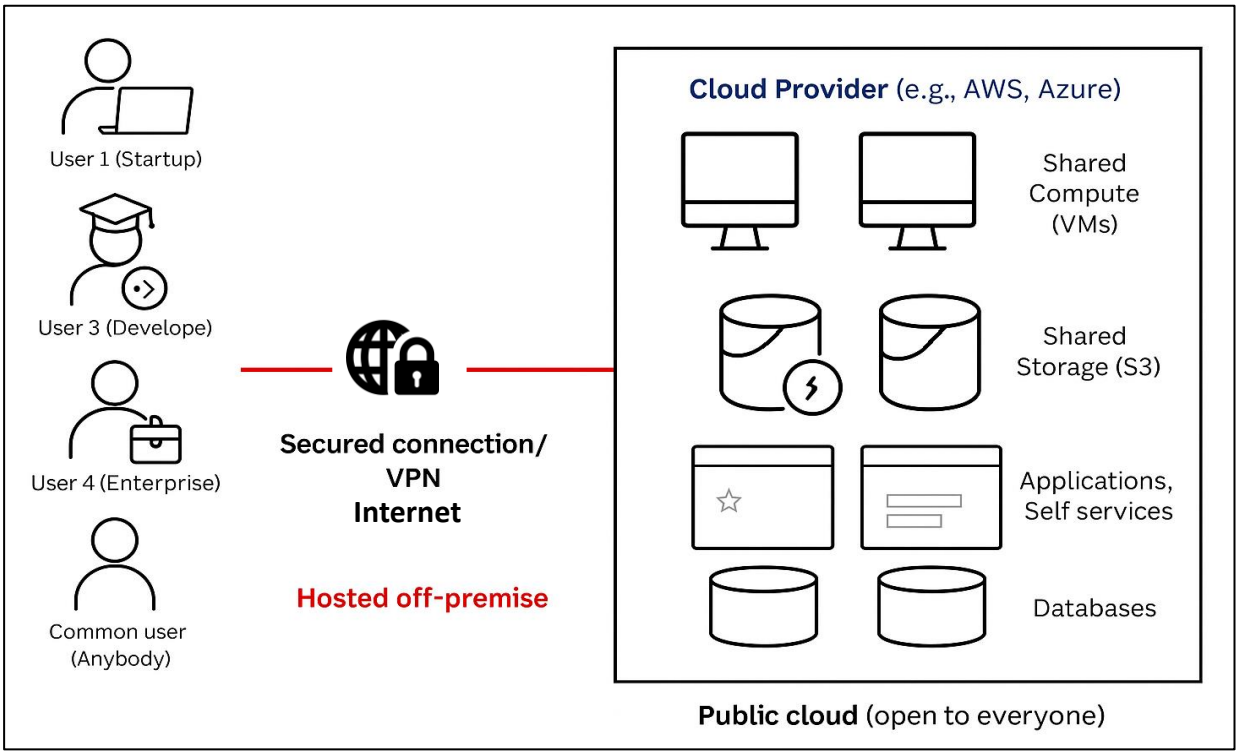
→ Cloud resources (like servers, storage, apps) → owned and managed by a third-party provider (AWS, GCP, etc.)

→ It may be owned, managed, and operated by a business, academic, or government organization, or some combination of them.

→ Cost-effective → pay-as-you-go.

→ Deployment Options:

Off-premises → In the data centers of cloud providers.



Example:

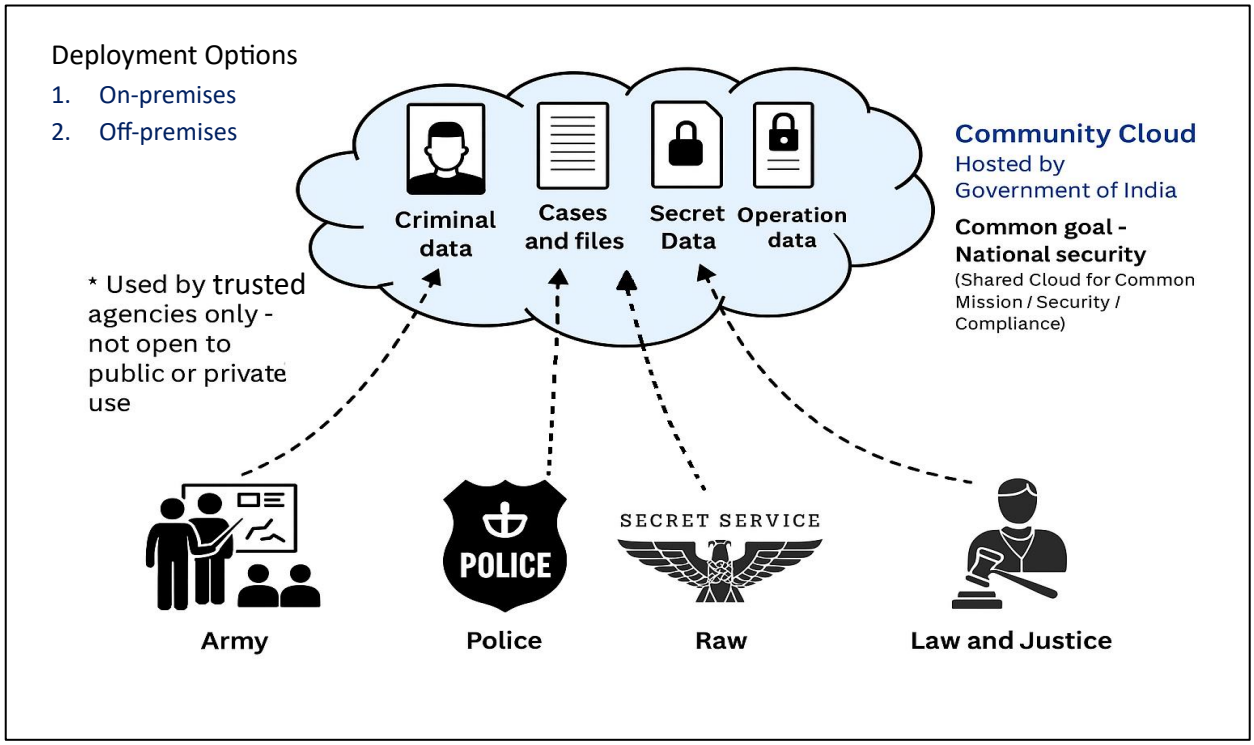
Students and professionals use **Google Drive**, which runs on Google Cloud Platform, to store documents, collaborate in real time, and access files from any device.

3. Community Cloud:

Definition: A community cloud is a cloud environment shared by multiple organizations that have similar needs or concerns; like security, policy, or compliance.

→ Shared by a group of related organizations (e.g., banks, hospitals, govt depts) with common goals like security, policy, or compliance.

→ Owned/managed by one or more members or third party in the community, and can be on-premises or off-premises.



Example:

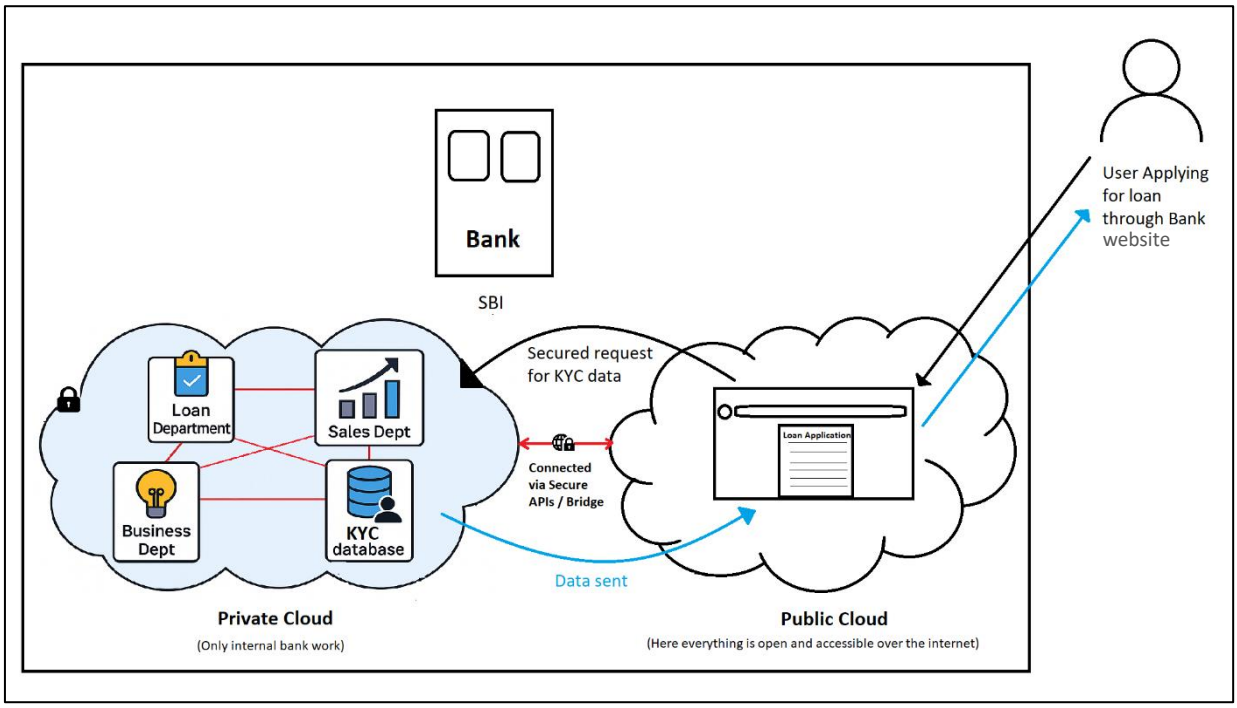
Suppose it's a **national security** matter. The Indian Government uses a community cloud where agencies like ATS, Police, Army, RAW, and Intelligence Bureau store and access data related to crimes, terrorism, operations, tips, and criminals. Since they all work towards the common goal of national security, they share this secure cloud environment while following strict policies and data regulations.

4. Hybrid Cloud:

Definition: A Hybrid Cloud combines two or more types of cloud infrastructures (e.g., private, public, community), which work together but remain separate systems.

→ Connected via APIs or software

→ Data/app portability between clouds



Example:

A bank runs customer data securely on its private cloud, but uses a public cloud (like AWS) to handle heavy website traffic during loan application season.