Draft Study Material

Cloud Computing Assistant

(Qualification Pack: Ref. Id. NIE/ITS/Q1201, NSQF Level 3)

Sector: Information Technology-Information Technology Enable Services (IT-ITeS)

(Grade X)

विद्यया ऽ मृतमश्नुते

एन सी ई आर टी
NCERT

PSS CENTRAL INSTITUTE OF VOCATIONAL EDUCATION

(a constituent unit of NCERT, under Ministry of Education, Government of India)

Shyamla Hills, Bhopal- 462 002, M.P., India

https://www.psscive.ac.in

© **PSS Central Institute of Vocational Education, Bhopal 2025**

No part of this publication may be reproduced, stored in a retrieval system or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise without the prior permission of the publisher.

Preface

Vocational Education is a dynamic and evolving field, and ensuring that every student has access to quality learning materials is of paramount importance. The journey of the PSS Central Institute of Vocational Education (PSSCIVE) toward producing comprehensive and inclusive study material is rigorous and time-consuming, requiring thorough research, expert consultation, and publication by the National Council of Educational Research and Training (NCERT). However, the absence of finalized study material should not impede the educational progress of our students. In response to this necessity, we present the draft study material, a provisional yet comprehensive guide, designed to bridge the gap between teaching and learning, until the official version of the study material is made available by the NCERT. The draft study material provides a structured and accessible set of materials for teachers and students to utilize in the interim period. The content is aligned with the prescribed curriculum to ensure that students remain on track with their learning objectives.

The contents of the modules are curated to provide continuity in education and maintain the momentum of teaching-learning in vocational education. It encompasses essential concepts and skills aligned with the curriculum and educational standards. We extend our gratitude to the academicians, vocational educators, subject matter experts, industry experts, academic consultants, and all other people who contributed their expertise and insights to the creation of the draft study material.

Teachers are encouraged to use the draft modules of the study material as a guide and supplement their teaching with additional resources and activities that cater to their students' unique learning styles and needs. Collaboration and feedback are vital; therefore, we welcome suggestions for improvement, especially by the teachers, in improving upon the content of the study material.

This material is copyrighted and should not be printed without the permission of the NCERT-PSSCIVE.

Deepak Paliwal
(Joint Director)
PSSCIVE, Bhopal

STUDY MATERIAL DEVELOPMENT COMMITTEE

Members

Akshya Arya, Assistant Professor, Department of Computer Applications, JECRC University, Ramchandrapura Industrial Area, Sitapura, Vidhani, Jaipur, Rajasthan 302022

Dinesh Chotia, Computer Instructor, Govt. HSS, Noowan, Churu, Rajsthan

Kuldeep Malviya, SRA, Department of Engineering and Technology, PSSCIVE, NCERT, Bhopal, Madhya Pradesh

Muskan Gupta, Assistant Professor at Nirmal College, Hindaun City, Karauli, Rajasthan.

Prakash Khanale, Professor and Head, Department of Computer Science, DSM College, Parbhani, Maharashtra

Member Coordinator

Deepak D. Shudhalwar, Professor (CSE), Head, Department of Engineering and Technology, PSSCIVE, NCERT, Bhopal, Madhya Pradesh

S.No.	Title	Page No.
	Module 1. Cloud Platforms & Architecture	**6**
	Module Overview	6
	Learning Outcome	6
	Module structure	6
	Session 1. Global Cloud Providers	7
	Session 2. Cloud Application Workflow	16
	Session 3. Cloud Compute Power in Virtual Machines	28
	Module 2. Cloud Security & Data Protection	**39**
	Module Overview	39
	Learning Outcome	39
	Module structure	39
	Session 1. Authentication and Multi-Factor Authentication for Secure Cloud Access	40
	Session 2. Managing Access in Cloud Computing using IAM (Users, Roles and Permissions)	58
	Session 3. Encryption Basics and Secure Data Storage in Cloud Computing	73
	Session 4. Secure Web Communication using HTTPS and SSL-TLS	88
	Module 3. Modern Cloud Applications & Services	**101**
	Module Overview	101
	Learning Outcome	101
	Module structure	101
	Session 1. Building Modern Applications Using Cloud Services	102
	Session 2. E-Commerce Systems and Secure Digital Payments	112
	Session 3. Introduction to IoT and Edge Computing	121
	Session 4. Real-Time Location Services and Navigation Systems	127
	Module 4. Cloud Deployment & Operations	**125**
	Module Overview	125
	Learning Outcome	125
	Module structure	125
	Session 1. Application Deployment Workflow	132
	Session 2. Introduction to Website Hosting and Cloud Deployment	139
	Session 3. Backup and Restore Planning in Cloud Computing	146
	Session 4. Cloud Project Development and Collaboration	152
	Answer Key	**157**

Module 1. Cloud Platforms & Architecture

This module on ***Cloud Platforms & Architecture*** helps you to understand the basic concepts of cloud computing and its use in modern digital services. In Session 1, you will learn about the limitations of traditional computing and how cloud platforms provide flexible and cost-effective solutions; by the end of the session, they are able to identify key features of cloud computing, explain its advantages, and recognize major cloud service providers. In Session 2, you will understand how cloud applications work using the client–server model; they learn how data flows between clients, servers, and databases, and are able to describe the request–response process in real-life applications. In Session 3, you will learn about virtualization and virtual machines (VMs); they understand the role of hypervisors and are able to explain how cloud providers deliver computing power efficiently using shared resources. Overall, the module enables students to build foundational knowledge of cloud computing and relate it to everyday technologies such as websites, mobile applications, and online platforms.

Session 1. Global Cloud Providers

Rohan is a bright student from a small town in India who has a brilliant idea. He wants to create an online platform called **"Young Creative Hub"** where students from across the country can share their artwork, stories, and creative projects.

Rohan spends weeks designing the platform and planning its features. He is excited and ready to launch the website. However, one important question troubles him:

"Where will this website live?"

At first, Rohan thinks of buying a powerful computer and keeping it in his room to run the website. Soon he realizes the risks:

- What if the computer overheats?
- What if thousands of users visit the website at the same time?
- What if there is a power failure?
- What if hackers try to attack the system?

Running a website from a single computer is risky and unreliable. These were the same challenges faced by people who wanted to build online services before cloud computing became popular. Figure 1.1 illustrates the scenario.

Fig. 1.1: Website Scenario

To understand how cloud computing solves this problem, let us first examine how computing worked earlier.

1.1. The Traditional Way of Computing

Before the cloud, if you wanted to launch a website, you had to face four major "Boss Levels" (Challenges):

Table 1.1 Shows Challenges of Traditional IT Infrastructure Management

Challenge	**Explanation**
Hardware Cost	Organizations had to purchase expensive servers costing from ₹50,000 to several lakhs.
Facilities	Servers require uninterrupted power supply, backup generators, and cooling systems to prevent overheating.
Skilled Staff	Experts such as system administrators, network engineers, and security specialists were needed to manage the infrastructure.
Capacity Planning	It was difficult to predict how many users would visit the website. If the server was too small, it could crash; if it was too large, money would be wasted.
Maintenance	Hardware needed regular repairs, upgrades, and replacement after a few years.
Security	Data centers required security systems such as cameras, locks, and fire protection.

The Result:

Because of these challenges, traditional computing was:

- *Extremely Expensive:* Required huge money at the start.
- *Incredibly Slow:* Took a lot of time to set up.
- *Highly Inflexible:* Very difficult to increase or decrease resources based on need. Figure 1.2 shows traditional computing model.

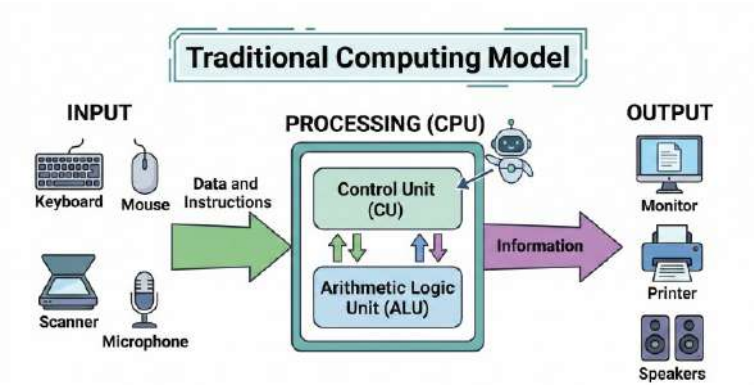


Fig. 1.2: The Traditional Computing Model

1.2 The Cloud Solution: A New Way of Computing

A Cloud Platform is a revolutionary service that provides computing resources like servers and storage over the internet on-demand. Instead of buying

expensive hardware, you simply rent what you need from companies called Cloud Service Providers (CSPs).

Using a "Pay-as-you-go" model, you only pay for the resources you actually use and can stop anytime, just like paying for electricity. The best part is that providers like Amazon (AWS), Microsoft (Azure), and Google (GCP) handle all the difficult tasks like power, cooling, and maintenance so you can focus on your work.

Let us compare the traditional model with the cloud model side by side:

1.3 Key Characteristics of Cloud Platforms

The National Institute of Standards and Technology (NIST), a US government agency, has defined five essential characteristics that all true cloud platforms must possess. Understanding these characteristics is fundamental to understanding cloud computing.

On-demand Self-service: You can get computing resources (like storage) automatically with a few clicks. You don't need to call or wait for a provider to help you.

Broad Network Access: You can access your data and services from anywhere in the world using the internet on any device (Phone, Tablet, or Laptop).

Resource Pooling: Many customers share the same physical hardware (servers), but their data remains private and secure—just like families sharing an apartment building.

Rapid Elasticity: You can instantly increase or decrease your resources based on your needs. It feels like you have unlimited power that grows or shrinks with your traffic.

Measured Service: The cloud provider tracks exactly how much you use (like an electricity meter) so you only pay for what you actually consume.

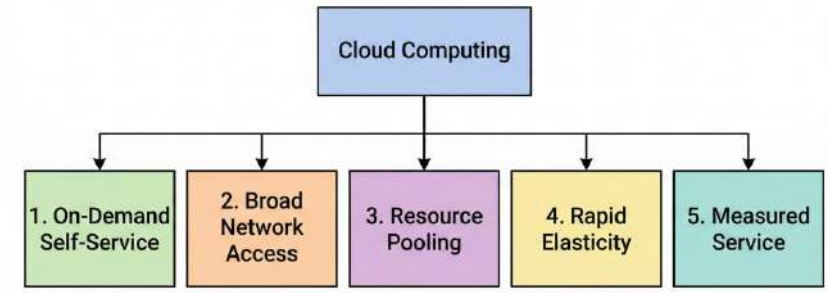


Fig. 1.3: The Five Essential Characteristics of Cloud Computing

1.4 What Can You Do with a Cloud Platform?

Cloud platforms offer a vast range of services that help businesses and individuals work faster and smarter. Here are the most common uses:

Run Applications: You can host websites and mobile apps on virtual servers. Instead of one physical machine, your app can run on multiple servers across the globe for better speed and reliability.

Store and Backup Data: Cloud offers virtually unlimited storage for photos, videos, and documents. It is designed to be durable (your data won't get lost) and highly available (you can access it anytime).

Analyze Data: Businesses use powerful cloud tools to process massive amounts of data. This helps them understand customer behavior and find patterns to improve their work.

Use Artificial Intelligence (AI): Cloud providers offer ready-to-use AI services. Developers can easily add features like Image Recognition (identifying faces), Speech-to-Text, Language Translation, and Chatbots to their apps.

Develop and Test: Developers can quickly create "test environments" to experiment with new software without buying new hardware or affecting their main website.

1.5 Who Uses Cloud Platforms?

One of the most remarkable things about cloud computing is the incredible diversity of its users. Cloud platforms are used by virtually every type of organization and individual imaginable—from solo developers and students to the world's largest corporations and government agencies.

1.6 Major Cloud Service Providers

There are many Cloud Service Providers (CSPs) globally, but three major companies dominate the market. They own massive data centers and provide computing power to users over the internet.

Table 1.2 shows list of Major Cloud Service Providers:

amazon web services	Launched in 2006, it is the world's oldest and most widely used cloud platform.
Azure	Started in 2010, it is very popular among businesses that already use Microsoft software like Windows and Office.
Google Cloud Platform	Launched in 2011, it provides the same infrastructure that Google uses for Search and YouTube. It is well-known for Data Analytics and AI.

1.2.2 Other Cloud Providers

While AWS, Azure, and GCP dominate the global market, several other providers serve important niches or dominate specific regions.

IBM Cloud	Oracle Cloud	Alibaba Cloud	MeghRaj (GI Cloud)

1.3.1 Cloud Regions

A Region is a specific geographic location (like Mumbai, London, or Singapore) where a cloud provider operates multiple data centers. Cloud providers build these regions all over the world so that businesses can host their applications closer to their users. For example, if your users are in India, you would choose the 'Mumbai Region' to make your website load faster. Regions are kept far apart from each other so that even if a natural disaster happens in one country, the services in other regions remain safe.

Availability Zones (AZs)

An Availability Zone (AZ) is a single, isolated data center (or a group of them) located inside a specific Region. Every Region is made up of multiple AZs (usually 3 or more) that are connected by high-speed fiber cables but are physically separate. They have their own independent power, cooling, and security. The main goal of AZs is to provide backup; if one data center (AZ) fails due to a power cut or fire, your application continues to run smoothly from another AZ in the same region.

Regions vs. Availability Zones: A Comparison

Table 1.3: Regions vs. Availability Zones

Feature	**Region**	**Availability Zone (AZ)**
Definition	A broad geographic area	A distinct location within a region
Composition	Contains two or more Availability Zones	Contains one or more physical data centers
Distance Apart	Hundreds or thousands of kilometers	Several kilometers
Purpose	Geographic isolation, data residency, low latency for regional users	High availability, fault tolerance within a region
Failure Impact	Failure of an entire region is rare but possible	Failure of one AZ does not affect others in the same region
Analogy	A city (Mumbai)	A neighborhood within the city (Bandra, Andheri)
Practical Activity 1.1. Research Cloud Providers		

Objective

To identify major cloud service providers, understand the types of services they offer, compare their features and offerings, and develop basic research and analytical skills.

Material Required

- Computer/Laptop with internet access
- Web browser
- Notebook and pen

Procedure

Step 1. Open a web browser and visit the official websites of the cloud providers: Amazon Web Services, Microsoft Azure, Google Cloud

Step 2. Navigate through each website and explore sections such as:

- Products/Services
- Solutions
- Pricing
- Free tier or trial options

Step 3. Identify and note at least three services offered by each provider, such as: Compute (Virtual Machines), Storage, Database

Step 4. Prepare a comparison table in your notebook with the following headings:

- Name of Cloud Provider
- Key Services
- Special Features
- Free Tier Availability

Step 5. Analyze the information collected and identify similarities and differences among the providers.

Step 6. Write a short conclusion (4–5 lines) stating which cloud provider you find most useful and why.

Observation

All cloud providers offer core services like compute, storage, and databases, provide unique tools and features, ensure scalability and internet accessibility, and include free tier options for beginners to explore cloud computing.

Practical Activity 1.2. Draw a Cloud Architecture (Multi-Region, Multi-

AZ)

Objective

To understand the structure of cloud architecture, illustrate how multi-region and multi-Availability Zone setups ensure reliability and availability, and relate these concepts to real-world applications such as Zomato or Swiggy.

Material Required

- A4 size sheet / notebook
- Pencil and eraser
- Ruler
- Colored pens/pencils (optional)

Procedure

Step 1. Draw a cloud user interface at the top (mobile users accessing the app).

Step 2. Below it, draw the Internet as a connecting layer.

Step 3. Draw a Load Balancer to distribute user requests.

Step 4. Divide the page into two regions:

- Region 1 (e.g., North India)
- Region 2 (e.g., South India)

Step 5. Inside each region, draw 2–3 Availability Zones (AZs).

Step 6. In each AZ, include:

- Web Server
- Application Server
- Database

Step 7. Connect:

- Users → Load Balancer → Regions
- Load Balancer → AZs

Step 8. Draw arrows between the two regions to represent data replication and backup.

Step 9. Label all components clearly.

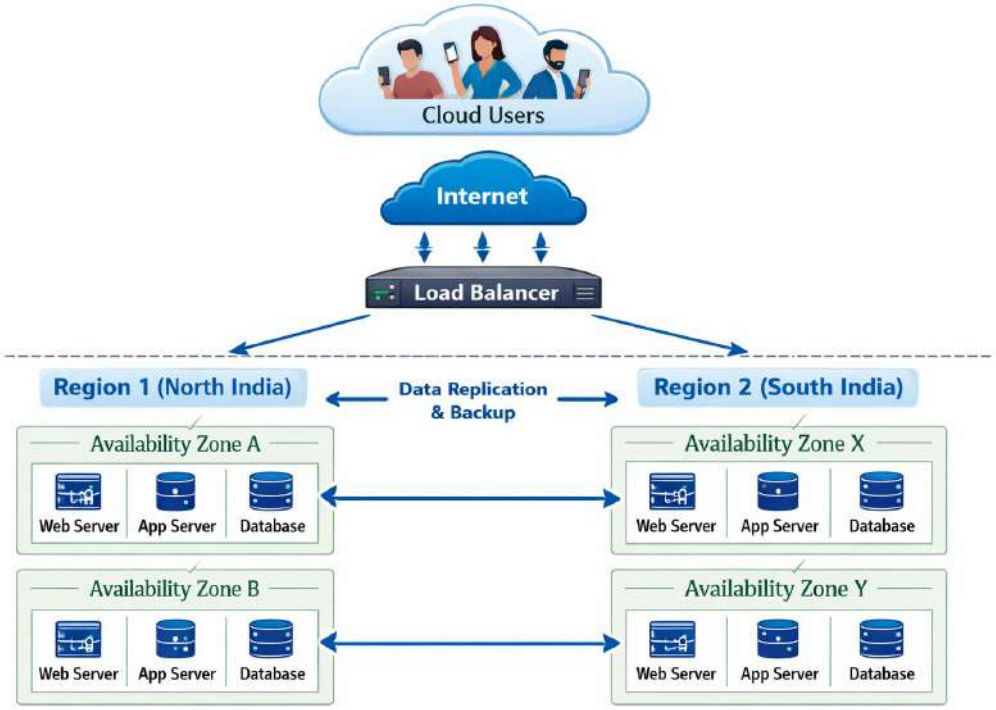


Fig. Cloud-based infrastructure with regional redundancy

Observation

Multiple regions support disaster recovery, Availability Zones ensure service continuity, load balancing enhances performance and traffic management, and applications remain accessible and reliable even during heavy load or failures.

Practical Activity 1.3. Group Discussion on "If Cloud Computing Did Not Exist, How Would Our Daily Lives Be Different?"

Objective

To understand the role of cloud computing in everyday life, analyze the dependence of modern services on cloud technology, develop critical thinking and communication skills, and evaluate its impact on key sectors such as streaming, social media, online shopping, and education.

Material Required

- Chart paper / A4 sheets
- Pens / markers
- Sticky notes (optional)
- Internet-enabled device (for reference, if available)

Procedure

Step 1. Divide the class into four groups.

Step 2. Assign each group one area:

1. Streaming services (e.g., Netflix, Hotstar)
2. Social media (e.g., Instagram, Facebook)
3. Online shopping (e.g., Amazon, Flipkart)
4. Education platforms (e.g., Byju's, Unacademy)

Step 3. Ask each group to discuss:

5. How the assigned service works using cloud computing.
6. What changes would occur if cloud computing did not exist.
7. Difficulties users might face in such a scenario.

Step 4. Students prepare key points on chart paper.

Step 5. Each group presents their findings to the class (3–5 minutes per group).

Step 6. Conduct a brief class discussion to compare all areas and summarize key ideas.

Observation / Result

Students observe that cloud computing enables easy data access, storage, and sharing over the internet, and without it, services like streaming, social media, online shopping, and digital education would be limited or inefficient, highlighting its essential role in modern digital life.

Summary

In this session, you learned about *cloud computing* and how it has improved the way websites and applications are created and managed. Earlier, in the traditional computing system, organizations had to spend a lot of money on hardware, maintenance, and security, which made the process expensive and inflexible. Cloud computing provides resources like servers and storage over the internet on a pay-as-you-go basis, making it more affordable and efficient. I also learned about the five key characteristics defined by the National Institute of Standards and Technology, such as on-demand self-service, broad network access, resource pooling, rapid elasticity, and measured service. Additionally, I understood the role of major cloud providers like Amazon Web Services, Microsoft Azure, and Google Cloud Platform. Overall, cloud computing makes technology faster, scalable, reliable, and plays an important role in our daily digital life.

Check Your Progress

A. Multiple Choice Questions (MCQs)

1. What is the main advantage of cloud computing? (a) Requires more hardware (b) Pay only for what you use (c) Needs more maintenance (d) Works without internet
2. Which of the following is a cloud service provider? (a) Windows (b) Linux (c) Amazon Web Services (d) Python
3. What does "rapid elasticity" mean in cloud computing? (a) Slow performance (b) Fixed resources (c) Increase or decrease resources quickly (d) No internet access
4. Which component ensures backup within a region? (a) Region (b) Availability Zone (c) Server (d) Internet
5. In traditional computing, which was a major challenge? (a) Low cost (b) Easy scaling (c) High hardware cost (d) Unlimited storage

B. Fill in the Blanks

1. Cloud computing works on a _____ model.
2. _____ allows users to access services from anywhere.
3. Multiple data centers in one region are called _____.
4. _____ computing was expensive and inflexible.
5. Cloud providers charge based on _____ service.

C. True or False

1. Cloud computing requires buying physical servers.
2. Availability Zones help in fault tolerance.
3. Cloud services cannot be accessed on mobile devices.
4. Resource pooling means sharing hardware among users.
5. Cloud computing is slower than traditional computing.

D. Short Answer Questions

1. What is cloud computing?
2. Give any two features of cloud platforms.
3. Name any two cloud service providers.
4. What is a Region in cloud computing?
5. Why is cloud computing more reliable than traditional computing?

Session 2. Cloud Application Workflow

In the previous session, we explored how the Cloud provides a massive 'power grid' of resources like storage and apps. Now, we will see how your own devices actually talk to that grid. This is done through the Client-Server Model, which is the basic language of the internet.

2.1 Client

A Client is a device or software that requests information or services from another computer. When you use a phone, laptop, or tablet to browse the internet, it acts as a client and sends a request. For example, when you open a website in a browser or use an app like Instagram, the browser or app works as the client. It is similar to a customer in a restaurant who places an order and waits for it to be served.

2.2 Server

A Server is a powerful computer that provides resources and services to clients. It runs 24/7 and waits for requests from devices such as phones or computers. In the restaurant analogy, the server is like the kitchen that receives orders, prepares them, and sends the results back. Servers are more powerful than regular PCs because they handle many requests at the same time. Examples include web servers, email servers, and game servers. In cloud computing, many servers are virtual servers that run on shared hardware.

2.3 Client-Servers Communication

Clients and servers talk using a Request-Response Pattern. First, the client sends a Request (like a click), and the server sends back a Response (like a web page). For this to work, both must speak the same "language" called a Protocol (most common is HTTPS). During the journey, data is broken into tiny pieces called Packets, which travel through the internet and get reassembled at the destination in a fraction of a second.

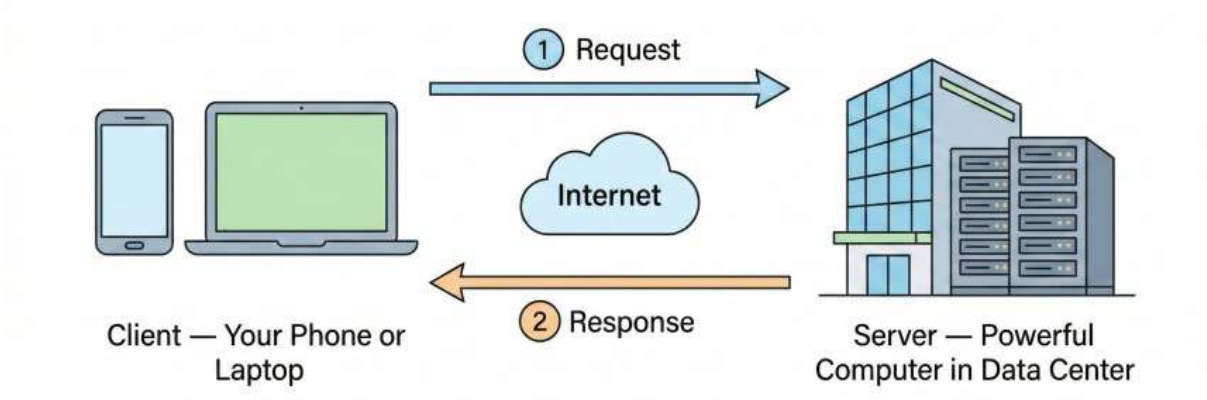


Fig. 2.1. Client-Server Communication Diagram

2.5. The User Request - Server - Database – Response Model

The client–server model is used in most online activities like Instagram, WhatsApp, Google, and YouTube. Your device acts as the client and sends requests to servers, which return the required data. The client always starts communication, and servers respond only after receiving a request.

In cloud applications, servers work with a database, which stores information. The process begins with a user request. The server processes it, asks the database for data, and sends the final response back to the user.

Steps in the Process

Step 1. User Request: Your action (click/search) sends a request from your device using HTTPS.

Step 2. Data Transfer: The request is broken into packets and travels through the internet to the server.

Step 3. Server Processing: The server reads the request and checks user details.

Step 4. Database Query: The server asks the database for required information.

Step 5. Data Return: The database sends the data back to the server.

Step 6. Response to User: The server formats the data and sends it back to your device.

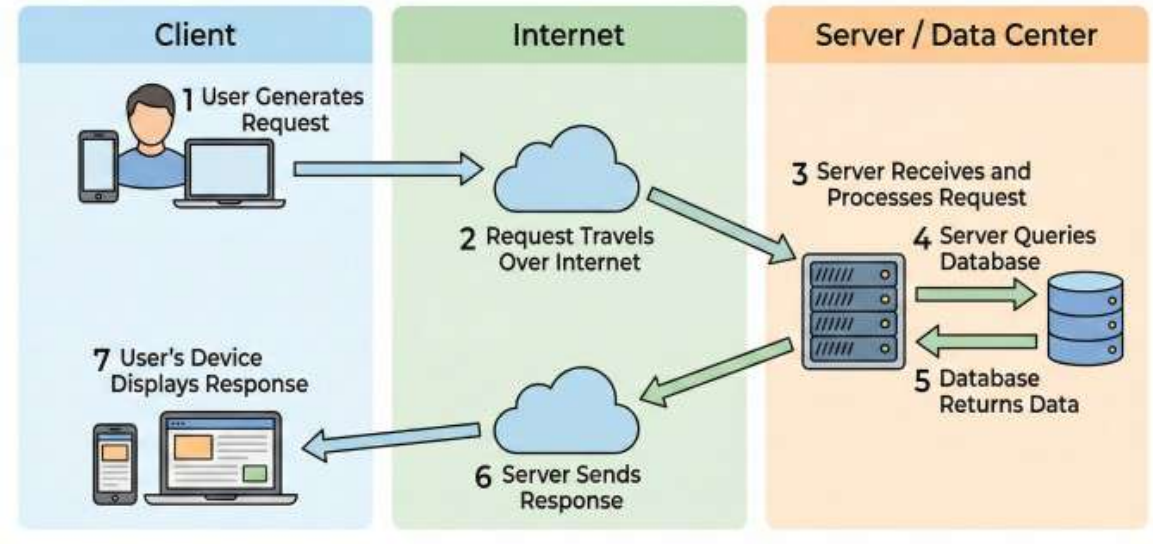


Fig. 1.13: Complete Request-Response Workflow

2.6 Real-World Example: Ordering Food on Zomato/Swiggy

Ordering food shows how the Client–Server–Database model works.

1. Opening the App (Request): When you open Zomato, your phone (Client) sends a request to the server to show nearby restaurants based on your location.

2. Fetching Data (Database Query): The server asks the database for restaurant details. The database returns names, ratings, and menus, which are shown on your phone.

3. Placing the Order: When you place an order, the server checks payment and stores order details in the database with a unique Order ID.

4. Order Confirmation (Response): The server sends a response confirming your order and showing delivery time.

5. Tracking: The delivery partner's location is sent to the server. The server updates the database and shows live tracking on your app.

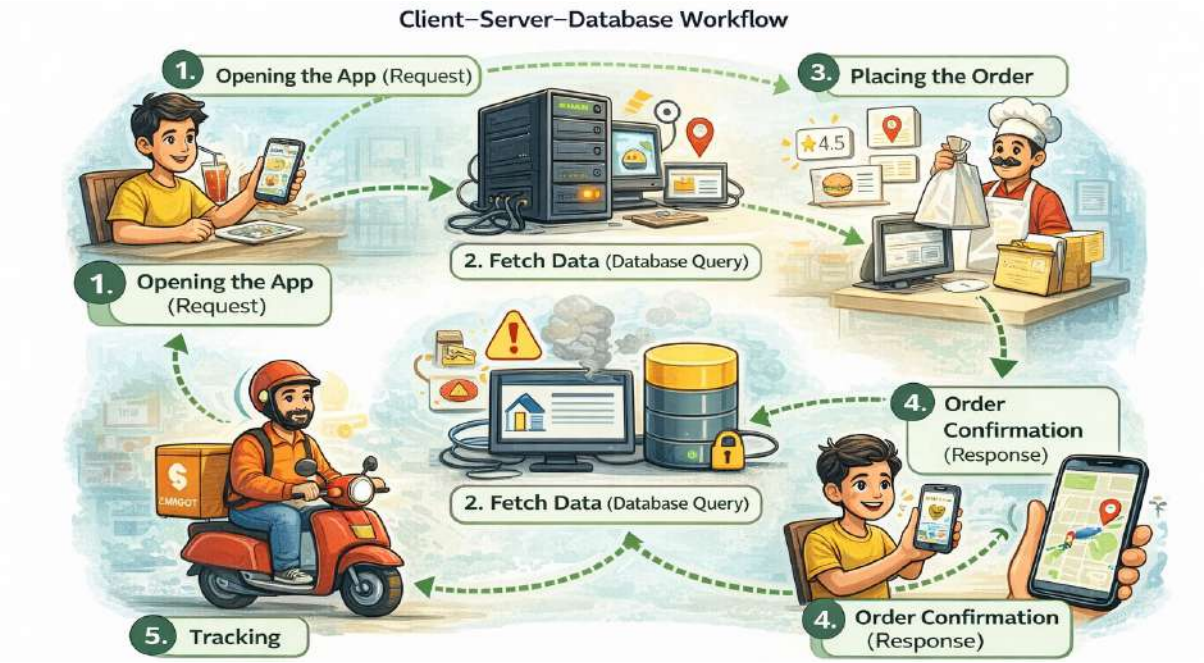


Fig. 1.14: Food Delivery App Workflow

2.4.1 Flow Diagram Explanation

A Flow Diagram is a visual tool that helps us see the path data takes through a system. It shows how different components connect and work together. In these diagrams, we use standard symbols: a device icon for the client, a cloud for the internet, a rectangle for the server, and a cylinder for the database. Arrows show the direction of data flow. These diagrams are essential for developers to plan systems, troubleshoot errors, and explain complex cloud architectures in a simple way.

2.4.2 Tracing the Path of Data

Let's trace the complete journey of a single request: The journey begins when a user takes an action on their client device. This action is packaged as a request and sent over the internet in small pieces called packets. These packets travel through routers and cables until they reach the data center. There, the server reassembles the packets and determines what the user

needs. If data is required, the server sends a query to the database, which - finds the information and sends it back. Finally, the server creates a response, breaks it into packets, and sends it back to the client device, which displays the result on the screen. This entire cycle happens in a fraction of a second.

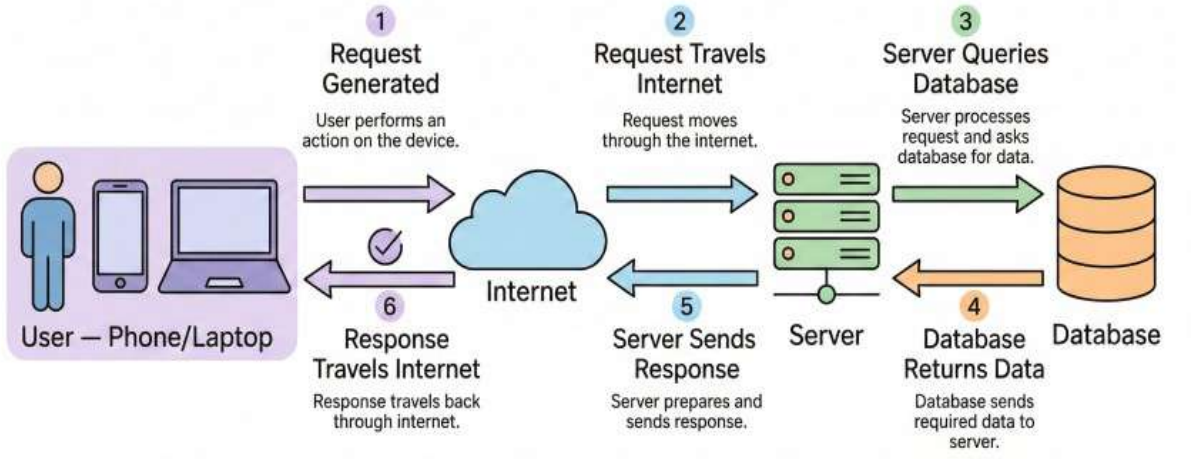


Fig. 1.15: Complete Data Flow Diagram

Key Terms Review

Below is a compact table of key terms and definitions for quick reference.

Client — Device/app that requests data (e.g., browser, mobile app)

1. Server — Computer that provides data/services to clients
2. Protocol — Rules for communication (e.g., HTTP, HTTPS)
3. Packet — Small piece of data sent over a network
4. Request — Message sent by client asking for data
5. Response — Message sent by server with data
6. Database — Organized storage of data
7. Query — Request to get data from database
8. IP Address — Unique number identifying a device
9. Router — Device that directs data between networks

Practical Activity 2.1. Tracking Your Online Actions (Client–Server Interaction)

Objective

To observe how everyday online activities use the **client–server model** and exchange data over the internet.

Materials Required

- Computer or smartphone
- Internet connection
- Notebook or worksheet

Procedure

Step 1. For one hour, record every online activity you perform.

Step 2. Activities may include visiting websites, using apps, sending messages, or watching videos.

Step 3. For each activity, note the device used (client).

Step 4. Identify the possible server or service provider involved.

Step 5. Record the type of data requested or sent during that activity.

Step 6. Fill in the observation table below.

Step 7. After one hour, review your observations and discuss them with your classmates.

Observation Table

Time	Online Action	Client Device	Server / Service	Data Requested or Sent
10:00 AM	Search information	Laptop	Google Search servers	Search query and webpage results
10:10 AM	Send message	Smartphone	WhatsApp servers	Text message data
10:20 AM	Watch video	Smartphone	YouTube servers	Video streaming data
10:30 AM	Check email	Laptop	Gmail servers	Email messages
10:40 AM	Browse website	Laptop	Website hosting server	Webpage content

Result

The activity shows that most online actions involve a client device requesting data from a server, which then sends the required information back.

Practical Activity 2.2. Draw a Workflow Diagram of an Online Service

Materials Required

- Notebook or chart paper
- Pencil and ruler
- Computer or smartphone (optional for reference)

Procedure

Step 1. Select an online service that you use frequently, such as YouTube, Instagram, or Google Maps.

Step 2. Identify the client device used to access the service (phone, tablet, or laptop).

Step 3. Draw a diagram showing the flow of a request from the client to the internet.

Step 4. Show how the request reaches the server of the service provider.

Step 5: Indicate how the server may interact with a database to retrieve information.

Step 6. Draw arrows showing how the response travels back to the client device.

Step 7. Label each step with a short explanation of what happens.

Step 8. Present the diagram to the class and explain the workflow of the request.

Observation

Students draw and label a **workflow diagram** showing the path of data from the client device to the server and back.

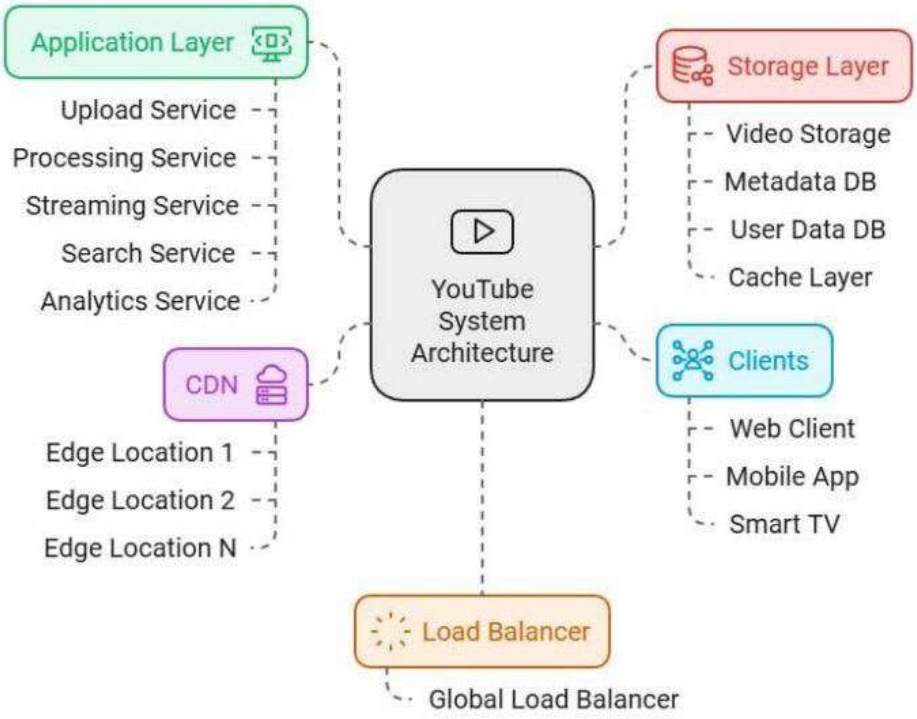


Fig. : Workflow diagram

Result

The activity demonstrates how online services follow a client–server workflow to process user requests and deliver information.

Practical Activity 2.3. Group Discussion – Impact of Global Server Failure on Daily Life – Group Discussion Activity

Materials Required

- Notebook or worksheet
- Pen or pencil

Procedure

Step 1. Form groups of **4–5 students** and discuss the question: "What would happen if all the servers in the world stopped working for one day?" Think about how different areas of daily life would be affected. Consider the following sectors:

- Communication
- Entertainment
- Shopping and e-commerce
- Banking and payments
- Education
- Transportation

Step 2. List all the online services or apps that would stop working if servers were unavailable.

- Record your observations in the table below.
- Summarize your group's findings.
- Present the results and discuss how society depends on servers.

Observation Table

Area of Daily Life	**Online Service / Example**	**Possible Impact if Servers Stop**
Communication	WhatsApp	Messages cannot be sent or received
Entertainment	YouTube	Videos cannot be streamed
Shopping	Amazon	Online purchases cannot be made
Banking	Online banking apps	Transactions and payments stop
Education	Google Classroom	Online classes and assignments stop
Transportation	Google Maps	Navigation and location services stop

Result

Students identify many services that depend on servers and understand how digital infrastructure supports modern life.

Practical Activity 2.4. Research Assignment – How Online Services Handle Millions of Requests

Objective

To understand how large online services use multiple servers, databases, and data centers to handle millions of user requests.

Materials Required

- Computer or smartphone with internet access
- Notebook or worksheet
- Web browser for research

Procedure

Step 1. Choose a popular online service such as Google Search, YouTube, or WhatsApp.

Step 2. Research how the selected service handles millions of user requests every second.

Step 3. Identify how the service uses multiple servers to manage large amounts of traffic efficiently.

Step 4. Find out how databases are used for storing and managing user data, messages, videos, or search results.

Step 5. Explore how data centres located in different regions of the world help in delivering faster services.

Step 6. Understand how these technologies (servers, databases, and data centers) help the service provide fast, reliable, and uninterrupted performance.

Step 7. Note down all the important points from your research in a clear and organized manner.

Step 8. Write a one-page report summarizing your findings in simple language.

Step 9. Share your report with the class.

Observation Points

Aspect to Research	**Key Information Found**
Name of Online Service	
Number of Users / Requests	

Use of Multiple Servers		
Role of Databases		
Use of Data Centers		
Benefits (speed, reliability, availability)		

Summary

In this session, students learned about the client–server model and how cloud applications work in real life. They understood that a client (such as a mobile phone or computer) sends a request through the internet, which is processed by a server. The server may interact with a database to retrieve information and then sends a response back to the client. Students also learned about concepts like protocols, data packets, and the complete request–response workflow. Through examples and activities, they understood how everyday services like apps and websites depend on this system, and how flow diagrams can be used to represent the path of data. Overall, the session helped them understand how modern digital services function efficiently.

Check Your Progress

A. Multiple Choice Questions (MCQs)

1. A device that sends a request to a server is called a: (a) Server (b) Client (c) Database (d) Router
2. Which component processes user requests and sends responses? (a) Client (b) Internet (c) Server (d) Protocol
3. What is the role of a database? (a) To send requests (b) To store and manage data (c) To connect networks (d) To control internet speed
4. Which protocol is commonly used for secure communication? (a) FTP (b) HTTP (c) HTTPS (d) TCP
5. Data sent over the internet is broken into small units called: (a) Files (b) Packets (c) Signals (d) Bytes

B. Fill in the Blanks

1. A _____ sends requests to the server.
2. A _____ provides services and data to clients.
3. Data travels over the internet in small pieces called _____.
4. The rules of communication between client and server are called _____.
5. A _____ stores organized information in cloud applications.

C. True or False

1. The client starts the communication in a client–server model.

2. Servers only work when a user is online.
3. Databases are used to store and retrieve data.
4. Packets are large blocks of data sent at once.
5. Online services like apps and websites depend on servers.

D. Short Answer Questions

1. What happens when you click a link on a website? Explain the process briefly.
2. How does a server use a database to respond to a user request?
3. Why are data packets important in internet communication?
4. How does the client–server model help apps like YouTube or Instagram work smoothly?
5. What problems might occur if the server does not respond to a client request?

Session 3. Cloud Compute Power in Virtual Machines

In this session, your will explore the concept of virtualization and understand how cloud computing delivers powerful computing resources using Virtual Machines (VMs). You will learn about the limitations of traditional physical servers and how virtualization overcomes these challenges by enabling multiple virtual systems to run on a single machine. By the end of the session, students will be able to define virtualization, explain the role of a hypervisor, identify the components of a virtual machine, and understand how cloud service providers offer scalable and cost-effective compute power through VM services.

3.1. The Problem with Physical Servers

Traditional physical servers had four major limitations that made computing expensive and slow. Virtualization was invented to solve these specific problems:

3.1.1 High Cost of Hardware

Purchasing physical servers is extremely expensive because they are specialized machines built to run 24/7. Beyond the hardware cost, companies also have to pay for expensive software licenses, cooling systems, and electricity, which makes it very hard for small startups to start a digital business.

3.1.2 Underutilization of Resources

Most expensive servers sit idle (doing nothing) for 80-90% of the time. This is because servers are bought to handle "Peak Load" (maximum traffic), but for the rest of the day, they use very little of their actual power, leading to a massive waste of money and capacity.

3.1.3 Difficulty in Scaling

Increasing the power of a physical server (Scaling) is difficult and slow. If an app suddenly gets millions of new users, adding more RAM or buying a new server can take too long, often requiring the system to be shut down, which causes downtime for users.

3.1.4 Long Procurement and Setup Times

Buying and setting up a new physical server is a very slow process that can take weeks or even months. From getting budget approvals and waiting for delivery to physically installing cables and software, the long delay makes it impossible for businesses to react quickly to new opportunities.

3.2. Introduction to Virtualization

Virtualization is the technology that solved the problems of physical servers. It allows one single physical server to do the work of many computers.

Virtualization is the process of creating a software-based (virtual) version of a computer. Instead of running just one Operating System (like Windows) on a physical server, virtualization allows you to run **multiple Operating Systems** at the same time on the same hardware. Each "virtual computer" is called a **Virtual Machine (VM)**. It feels like a real computer but shares its brainpower (CPU, RAM) with others on the same machine.

3.2.1 The Hypervisor: The Manager

The "magic" behind virtualization is a software called the Hypervisor. It sits between the physical hardware and the Virtual Machines. Its job is to distribute the physical CPU, memory, and storage to each VM and ensure they don't interfere with each other.

Type 1 (Bare Metal)	Runs directly on the hardware (Used in Cloud/Data Centers).
Type 2 (Hosted)	Runs like an app on an OS (Used on personal laptops).

3.2.2 Analogy: The Apartment Building

Think of a physical server as a single large house. Only one family can live there. If the family is small, space is wasted. Virtualization is like turning that house into an Apartment Building.

The building is the Physical Server.
Each Apartment is a Virtual Machine (VM).
The Building Manager is the Hypervisor.
The Families are the Apps/OS. Every family has its own private space (Isolation), but they all share the same land, water, and electricity (Physical Resources).

3.2.3 Benefits of Virtualization

Virtualization changed everything by providing these benefits:

- ***Server Consolidation:*** You can run many apps on one server instead of buying ten, saving money and electricity.
- ***Isolation:*** If one VM crashes or gets a virus, the others stay safe and keep working.
- ***Hardware Independence:*** Since a VM is just a "file," you can easily move it from one physical server to another without any setup.
- ***Disaster Recovery:*** Backing up a VM is as easy as copying a file, making it fast to fix things if a server fails.
- ***Multiple OS:*** You can run Linux and Windows on the same physical machine at the same time.

3.3. Virtual Machine (VM)

Now that we understand the concept of virtualization, let us examine virtual machines themselves in more detail. What exactly is a virtual machine, and what components make it work?

3.3.1 Definition: A Software-based Computer

A **Virtual Machine (VM)** is like a "computer within a computer." It is a software version of a physical computer that runs its own operating system (OS) and apps. Even though it shares the same physical hardware with other VMs, it is completely independent. To the user, a VM feels and works exactly like a real, physical computer, but it is actually just a protected environment created by the hypervisor.

3.3.2 Components of a VM

Just like a real computer, a VM needs specific parts to work, but these are all "virtual":

vCPU (Virtual CPU)	This is the brainpower assigned to the VM. The hypervisor gives each VM a slice of time on the real physical processor.
vRAM (Virtual Memory)	This is the memory allocated to the VM to run programs. The hypervisor ensures one VM cannot "peek" into another VM's memory.
Virtual Disk	Instead of a physical hard drive, a VM uses a large **file** to store its OS, apps, and data. This makes it very easy to copy or move the entire VM.
Virtual Network (vNIC)	This acts like a network card, allowing the VM to connect to the internet and talk to other computers.

3.3.3 Guest Operating System

The OS running inside a VM is called the **Guest Operating System**. One of the coolest things about virtualization is that you can have different Guest OSs on the same physical machine. For example, one physical server can run a Windows VM, a Linux VM, and an Ubuntu VM all at the same time. The Guest OS doesn't even know it's virtual; it thinks it has its own dedicated hardware.

3.3.4 How VMs Run on Physical Hosts

The physical computer that runs VMs is called the **Host**. It uses a **Hypervisor** as a manager to share its physical resources (CPU, RAM, Storage) among all the VMs. For example, when a VM needs to process data, the hypervisor "schedules" a tiny bit of time on the real CPU for that VM. This switching happens so fast (in milliseconds) that every VM feels like it is running

continuously on its own. It is a perfectly coordinated system that ensures efficiency and security

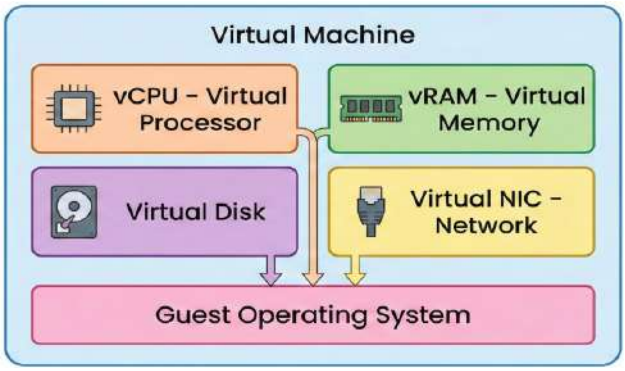


Fig. 3.1: Components of a Virtual Machine

3.4. Cloud Compute Power: VM Services

Once virtualization is set up in a data center, cloud providers sell that "Compute Power" to us as a service. Each big company has its own name for these Virtual Machines.

3.4.1 AWS: Amazon EC2

Amazon's service is called **EC2 (Elastic Compute Cloud)**. They call their virtual machines **"Instances."** You can choose an **AMI (Amazon Machine Image)**, which is like a pre-ready template of an operating system (Windows or Linux) to start your VM instantly.

3.4.2 Azure: Azure Virtual Machines

Microsoft's service is simply called **Azure Virtual Machines**. It is very popular with businesses that already use Microsoft tools like Windows Server or Office, as it integrates perfectly with them.

3.4.3 GCP: Google Compute Engine

Google's service is called **Compute Engine**. It runs on the same powerful infrastructure that runs YouTube and Gmail. A unique feature of Google is that it allows you to create **Custom Machine Types**, where you can pick the exact amount of RAM and CPU you want.

3.4.4 Key Features of Cloud VMs

All these providers offer four revolutionary features:

Create in Minutes	You don't have to wait weeks for a physical server. You can launch a VM with just a few clicks.
Choose Your Power	You can pick exactly how many **vCPUs** and how much **RAM** you need for your specific task.
Pay-as-you-go	You only pay for the minutes or hours your VM is actually running. If you turn it off, you stop paying.
Scale Easily	If your website suddenly gets famous, you can add 10 more VMs in seconds to handle the traffic. This is called **Elasticity**.

3.5. Use Cases for Virtual Machines

Virtual machines (VMs) are versatile and used for many tasks. The most common uses are:

Hosting Websites: Websites run on VMs. Small sites use one VM, while large sites use multiple VMs to handle high traffic.

Running Business Applications: Companies use VMs for software like HR and accounting. Each app runs separately, so problems in one do not affect others.

Development and Testing: Developers use VMs to test software. They can quickly create, delete, and try apps on different operating systems.

Data Processing: For large data tasks, multiple VMs work together. Each handles part of the data, making processing faster and cost-effective.

Practical Activity 3.1. Research Cloud VM Pricing

Aim: To compare the **virtual machine (VM) pricing** offered by major cloud providers.

Materials Required

- Computer or smartphone with internet access
- Web browser
- Notebook or worksheet

Websites to Visit

- Amazon Web Services
- Microsoft Azure
- Google Cloud Platform

Procedure

Step 1. Open a web browser and visit the official pricing pages of the three cloud providers.

Step 2. Look for their Virtual Machine (VM) or Compute service pricing pages.

Step 3. Select a similar VM configuration on each platform, for example:

- 2 vCPUs
- 4 GB RAM
- Linux operating system

Step 4. Note the hourly price listed for that configuration.

Step 5. Record the prices in the observation table.

Step 6. Compare the prices and identify which provider offers the best value for this configuration.

Step 7. Write a short summary explaining your findings.

Observation Table

Cloud Provider	Example VM Type	vCPU	RAM	Operating System	Hourly Price (Approx.)
Amazon Web Services	t3.medium	2	4 GB	Linux	~$0.056/hr
Microsoft Azure	B-series VM	2	4 GB	Linux	~$0.080/hr
Google Cloud Platform	e2-standard-2 (approx.)	2	4 GB	Linux	~$0.067/hr

Result

Students compare VM pricing across different cloud providers.

Practical Activity 3.2. Design a VM Architecture for an Online Bookstore

Aim: To understand how virtual machines (VMs) can be used to design the architecture of a simple cloud-based application.

Materials Required

- Notebook or chart paper
- Pencil and ruler
- Computer or smartphone for reference (optional)

Procedure

Step 1. Imagine that you are building an online bookstore application.

Step 2. The application must allow users to browse books, place orders, and store customer information.

Step 3. Identify the main components required for the system:

Web Server – handles user requests and displays the website.

Database Server – stores book details and customer orders.

Step 4. Decide how many Virtual Machines (VMs) are needed to run these components.

Step 5. Design a VM architecture by assigning a specific role to each VM.

Step 6. Draw a diagram showing the interaction between users, web servers, and the database server.

Step 7. Think about how the system can handle increased traffic during festival seasons.

Step 8. Write a short explanation describing your architecture and how it manages user requests.

Example Architecture

VM 1 – Web Server

- Runs the web application.
- Handles requests from users.
- Sends data requests to the database server.

VM 2 – Database Server

- Stores information about books, customers, and orders.
- Processes data queries from the web server.

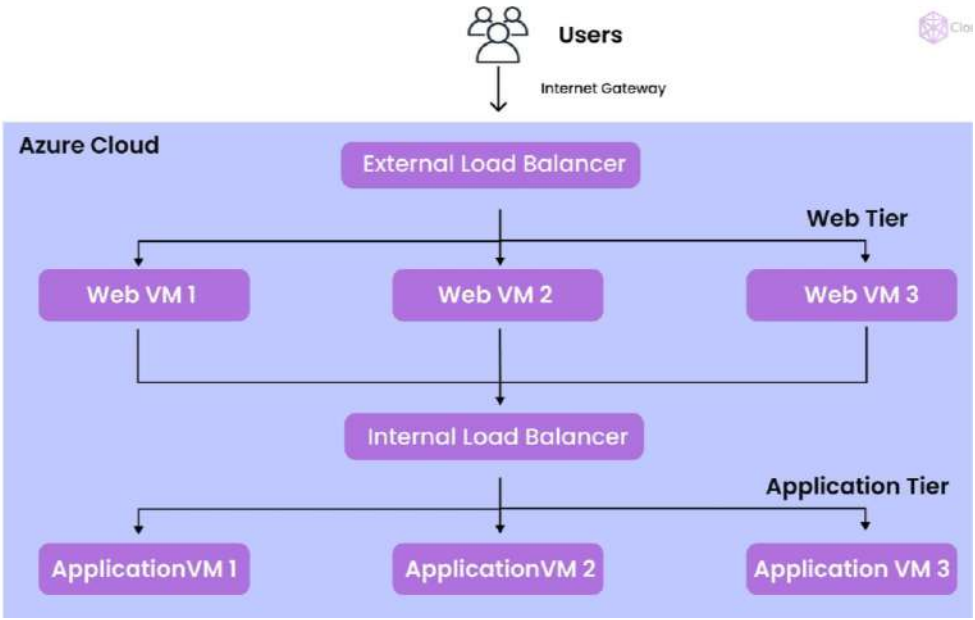
VM 3 – Backup / Load Balancer (Optional)

- Distributes incoming user requests among multiple web servers.

- Helps maintain performance during high traffic.

Observation

Students create a VM architecture diagram showing how different virtual machines work together to run an online application.



Result

The activity demonstrates how applications can be built using multiple virtual machines with different roles.

Practical Activity 3.3. Virtualization Timeline Research

Aim: To study the history and development of virtualization and understand how it led to the growth of cloud computing.

Materials Required

- Computer or smartphone with internet access
- Notebook or worksheet
- Chart paper and markers (optional for drawing the timeline)

Procedure

Step 1. Research the history of virtualization using reliable online sources or textbooks.

Step 2. Find out when virtualization was first introduced and which organizations developed the early technology.

Step 3. Identify some early virtualization systems and products.

Step 4. Study how virtualization technology gradually evolved and became the foundation of cloud computing.

Step 5. List the important milestones in chronological order.

Step 6. Draw a timeline diagram showing the major developments.

Step 7. Write a short explanation describing how virtualization evolved into modern cloud services.

Observation Table

Year / Period	Key Development	Description
1960s	Early Virtualization Concept	Mainframe computers allowed multiple users to share resources.
1970s	IBM Virtual Machines	IBM developed virtualization technology for mainframe systems.
1999	VMware Virtualization	VMware introduced virtualization for x86 servers.
2006	Launch of Cloud Services	Amazon Web Services introduced scalable cloud infrastructure.
2010s	Growth of Cloud Platforms	Companies like Microsoft Azure and Google Cloud expanded cloud services globally.

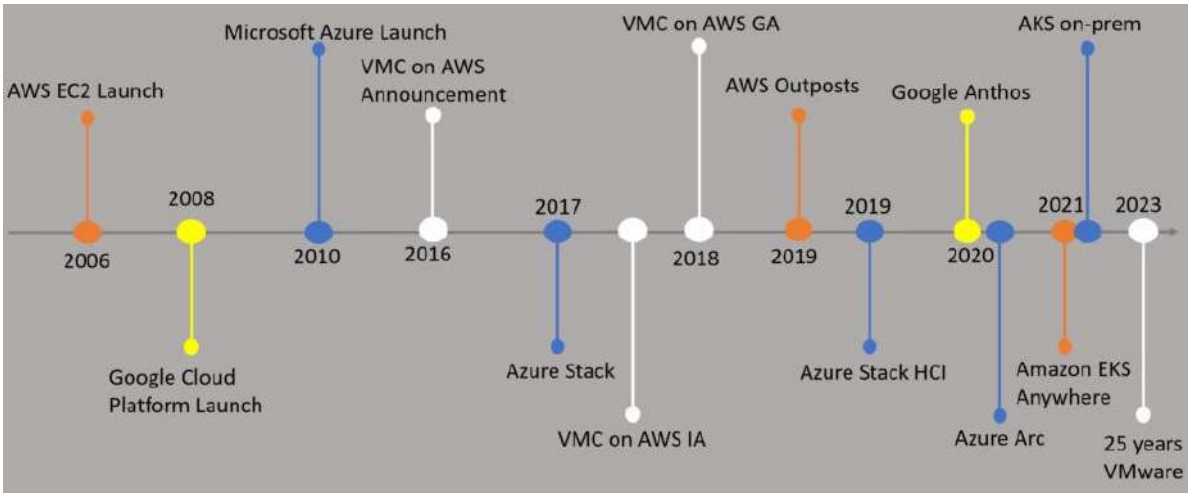


Fig. Timeline Diagram

Practical Activity 3.4. Group Discussion – Life Without Virtualization

Aim: To understand the importance of virtualization technology and its impact on modern computing and everyday life.

Materials Required

- Notebook or worksheet
- Pen or pencil

Group Formation

Procedure

Step 1. Form groups of 4–5 students and discuss the question: "How would the world be different if virtualization had never been invented?"

Step 2. Think about how virtualization supports modern technologies such as cloud computing, data centers, and online services.

Step 3. Discuss how different areas of life might be affected, including:

- Business and e-commerce
- Education and online learning
- Entertainment and streaming services
- Communication and social media

Step 4. List examples of technologies or services that depend on virtualization.

Step 5. Record your group's ideas in the observation table.

Step 6. Summarize your discussion and prepare a short conclusion.

Step 7. Present your group's findings to the class.

Observation Table

Area of Life	Technology / Service	Possible Impact Without Virtualization
Business	Amazon online stores	Higher cost of servers and slower services
Education	Google Classroom	Limited access to online classes
Entertainment	YouTube	Difficulty streaming videos at scale
Communication	WhatsApp	Messaging services may be slower or limited
Navigation	Google Maps	Real-time navigation services may not work efficiently

Result

Students identify how virtualization supports modern digital services and infrastructure.

Summary

In this session, students learned about virtualization and how it helps provide cloud computing power using Virtual Machines (VMs). They understood the limitations of physical servers, such as high cost, underutilization, and difficulty in scaling. Students learned that virtualization allows one physical server to run multiple virtual machines using a hypervisor, which manages resources like CPU, memory, and storage. They also explored the components of a VM, such as vCPU, vRAM, virtual disk, and network. Additionally, students learned how cloud providers like Amazon Web Services, Microsoft Azure, and Google Cloud Platform offer virtual machines as services. Overall, the session helped them understand how virtualization improves efficiency, scalability, and reliability in modern cloud computing.

Check Your Progress

A. Multiple Choice Questions (MCQs)

1. What is virtualization in computing? (a) Running only one program on a computer (b) Creating virtual versions of computers or servers (c) Removing operating systems from computers (d) Storing files on external drives
2. Which software manages and runs virtual machines? (a) Compiler (b) Hypervisor (c) Browser (d) Antivirus
3. Which type of hypervisor runs directly on hardware? (a) Type 1 Hypervisor (b) Type 2 Hypervisor (c) Application Hypervisor (d) Network Hypervisor

4. What is the main advantage of virtualization? (a) Increased hardware cost (b) Better use of computing resources (c) Reduced internet speed (d) Limited system usage
5. Virtualization forms the foundation of which modern technology? (a) Cloud Computing (b) Word Processing (c) Image Editing (d) Offline Software

B. Fill in the Blanks

1. _____ allows multiple virtual machines to run on a single physical computer.
2. A _____ acts like a real computer with its own operating system.
3. The software that creates and manages virtual machines is called a _____.
4. A _____ hypervisor runs directly on the physical hardware.
5. Virtualization is the basis of modern _____ computing.

C. True or False

1. Virtualization allows multiple systems to run on a single physical machine.
2. A virtual machine cannot run its own operating system.
3. Hypervisors manage and control virtual machines.
4. Virtualization increases the need for more physical servers.
5. Cloud computing relies on virtualization technology.

D. Short Answer Questions

1. What is virtualization?
2. What is a virtual machine (VM)?
3. What is the role of a hypervisor?
4. Name the two types of hypervisors.
5. Why is virtualization important in cloud computing?

Module 2. Cloud Security & Data Protection

This module on *Cloud Security and Data Protection* introduces the fundamental concepts of securing data and managing access in cloud computing environments. It aims to develop an understanding of how digital information is protected from unauthorized access, misuse, and cyber threats. In Session 1, you will learn about cloud security, the shared responsibility model, and the concept of authentication. They explore different authentication factors, understand the limitations of passwords, and learn how Multi-Factor Authentication (MFA) enhances security. The session also enables students to create strong passwords and adopt safe security practices. In Session 2, you are introduced to access control and the Principle of Least Privilege. They learn about Identity and Access Management (IAM), including its key components such as users, groups, roles, and policies, and understand how permissions are assigned and managed to ensure secure access to cloud resources. In Session 3, you will study encryption as a method of protecting data. They learn about plaintext and ciphertext, types of encryption (symmetric and asymmetric), and the role of encryption keys. The session also covers encryption at rest and in transit, along with secure cloud storage practices and key management services. Overall, the module equips students with essential knowledge and skills to ensure data security, protect digital identities, and apply safe practices while using cloud technologies.

Session 1. Authentication & Multi-Factor Authentication for Secure Cloud Access

Diya logged in a school cloud system, using her username and password to access assignments and records. One day, she experienced that someone else could try to access her account. Her password is accidentally exposed through a phishing email. To prevent such incidents, the school enables Multi-Factor Authentication (MFA). Now, after entering her password, Riya must also enter a one-time code generated on her mobile authenticator app or use her fingerprint. Even if someone knows her password, they cannot log in without her phone or biometric verification.

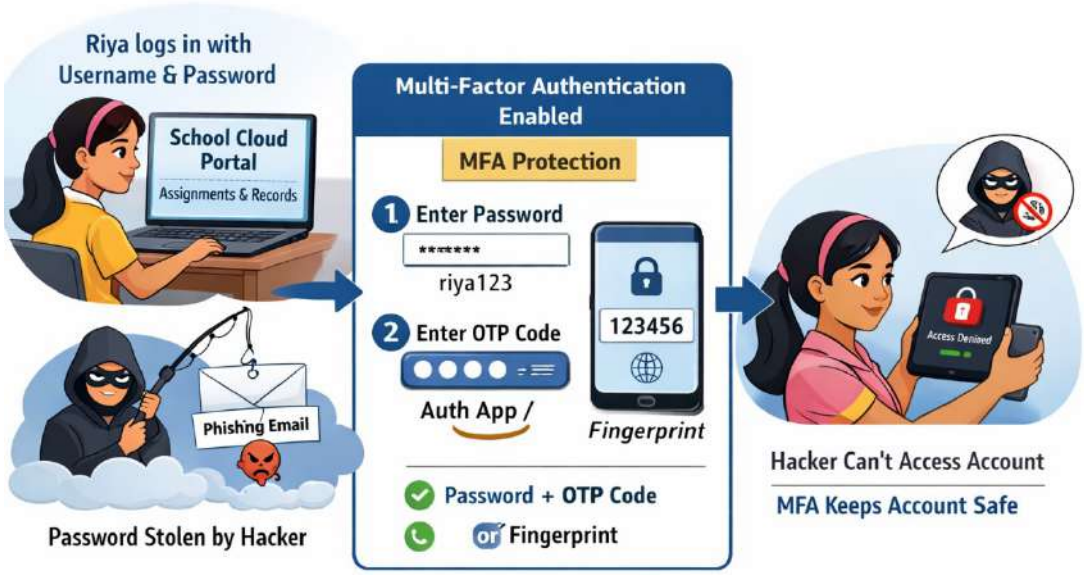


Fig. 1.1: Riya's MFA Protection

1.1. Cloud Security

Security in the cloud is about protecting your data and your applications from unauthorized access, theft, damage, or disruption. Cloud computing allows users to store data and run applications on the internet instead of personal computers. This means data is stored on remote servers. Because data travels through the internet and is stored online, it can be targeted by hackers if proper security is not used.

There are several reasons for cloud security. It helps to:

- Protect personal and organizational data
- Prevent unauthorized access
- Keep services running safely
- Maintain privacy
- Avoid data loss

Without security, sensitive information like passwords, financial data, and documents can be misused.

1.1.1 The Shared Responsibility Model

In cloud computing, security responsibility is shared between cloud provider and the cloud user.

Cloud Provider:

The company providing cloud services is responsible for:

- Physical security of data centers
- Network infrastructure
- Hardware maintenance
- Basic platform security

Cloud User:

The user is responsible for:

1. Protecting account login details
2. Managing access permissions
3. Keeping passwords secure
4. Enabling security features like MFA

This concept is called the Shared Responsibility Model because both provider and user must work together for security. Figure 1.2 shows Shared Responsibility Model.

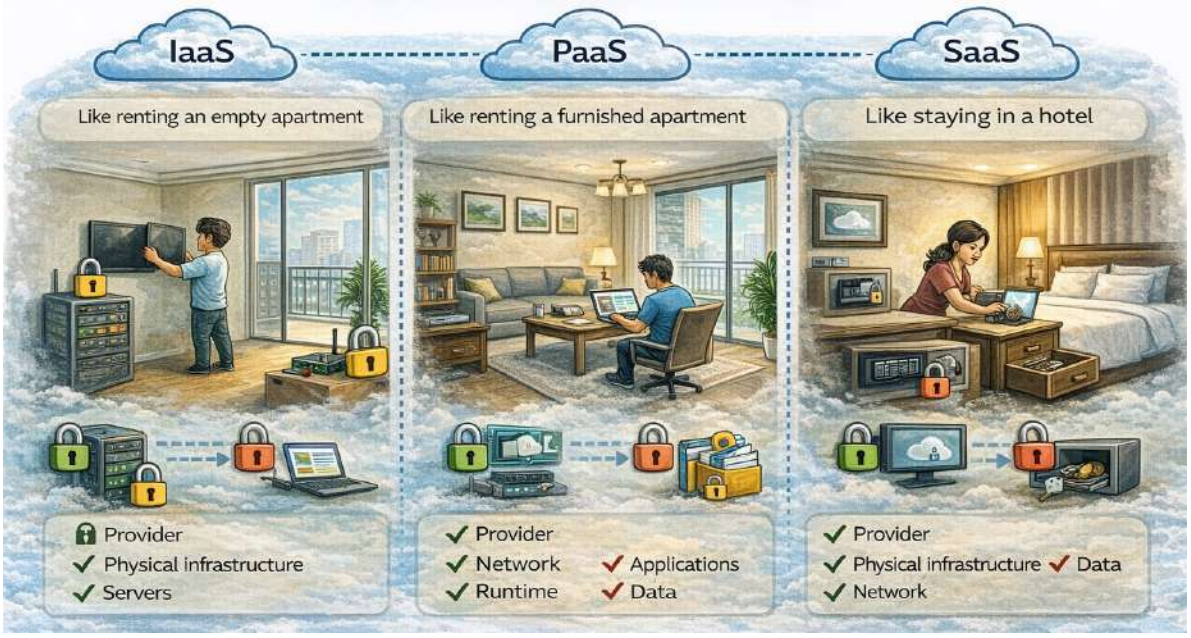


Fig. 1.2: Shared Responsibility Model

1.1.2 Responsibility by Service Type

The responsibility depends on the type of cloud service. There are three main service models:

Infrastructure as a Service (IaaS) — the provider secures the physical infrastructure, but the user is responsible for the operating system, applications, data, and network configuration. This is like renting an empty apartment where you have to install everything yourself.

Platform as a Service (PaaS) — the provider also secures the operating system and runtime environment. The user is responsible for applications and data. This is like renting a furnished apartment where the furniture is provided but you still need to lock your doors.

Software as a Service (SaaS) — the provider secures almost everything. The user is responsible for data managing user access. This is like staying in a hotel where the hotel handles security, but you are still responsible for not leaving your valuables in plain sight.

1.2. Authentication

Authentication is the process of verifying the identity of a user before giving access to a system.

Example: When you log into email using username and password, the system checks whether you are the correct user.

Authentication ensures:

- Only authorized users can access accounts.
- Data remains protected.
- Systems stay secure.

1.2.1 Factors of Authentication

Authentication works using three main factors. Figure 1.3 shows Factors of Authentication.

Fig. 1.3: Factors of Authentication

1. Something You Know: Information only you know. *Examples:* Password, PIN, Security answers

2. Something You Have: A physical item you possess.

Examples: Mobile phone, Smart card, Security token

3. Something You Are: Your biological features.

Examples: Fingerprint, Face recognition, Iris scan

1.2.2 Username and Password: The Most Common Method

You have probably used your username and password while login in your email account, social media and many other websites. The most common authentication method in use today is the combination of a username and password.

Username identifies the user and password confirms identity. This method is simple and widely used but not always secure.

1.2.3 Weaknesses of Passwords

Passwords alone can be risky because:

- Users create weak passwords
- Passwords can be guessed
- Passwords can be stolen through phishing
- Same password reused on many sites
- Password leaks from data breaches

Because of these risks, stronger security is needed.

Examples of Weak Passwords

Types of Passwords	Examples of Passwords
1. Very Common Passwords: There are some common passwords, that users normally used and these types of passwords initial tried by the hackers.	• 123456 • password • 12345678 • qwerty • abc123
2. Personal Information Based Passwords: People also use some kind of password based on their personal information. Hackers can guess these from social media.	• riya123 • Rahul2008 • amit@123 • Schoolname123 • india2024
3. Short Passwords: There are some short passwords used by many users. Short passwords are very easy to crack.	• 1234 • abcd • pass • admin
4. Repeated or Pattern Passwords: Some of the users have the habit of using the password as repeated numbers. These patterns are quite predictable.	• 111111 • 000000 • aaaaaa • 123123
5. Dictionary Words: Some of the users use the single word for password. Using such single word as password are easy for hacking tools.	• sunshine • football • welcome • computer

1.3 Multi-Factor Authentication (MFA)

Multi-Factor Authentication (MFA) means using two or more authentication factors to verify identity. Figure 1.4 shows illustration of MFA. MFA is sometimes called two-factor authentication or 2FA when exactly two factors are used. But the principle applies to any combination of multiple factors. *Example:* Using the Password + OTP on phone is the most common method of MFA used nowadays.

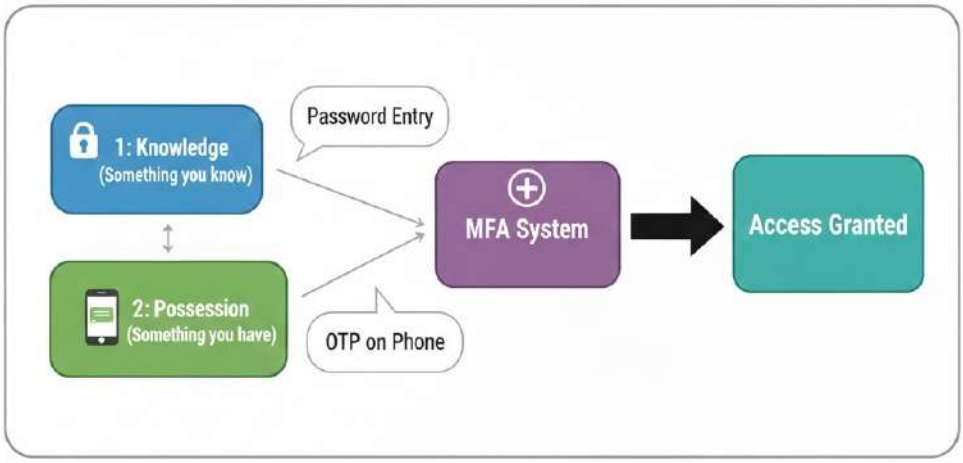


Fig. 1.4: Multi-Factor Authentication (MFA)

1.3.1 Security with MFA

MFA is safer because even if a password is stolen, attackers cannot access the account without the second factor. Microsoft research shows that MFA can block **99.9%** of automated account hacks. It is the single most important thing you can do for your cyber safety. It is one of the most effective security measures.

MFA provides, (i) Extra security layer, (ii) Reduces hacking risk, (iii) Protects cloud accounts, (iv) Prevents unauthorized access

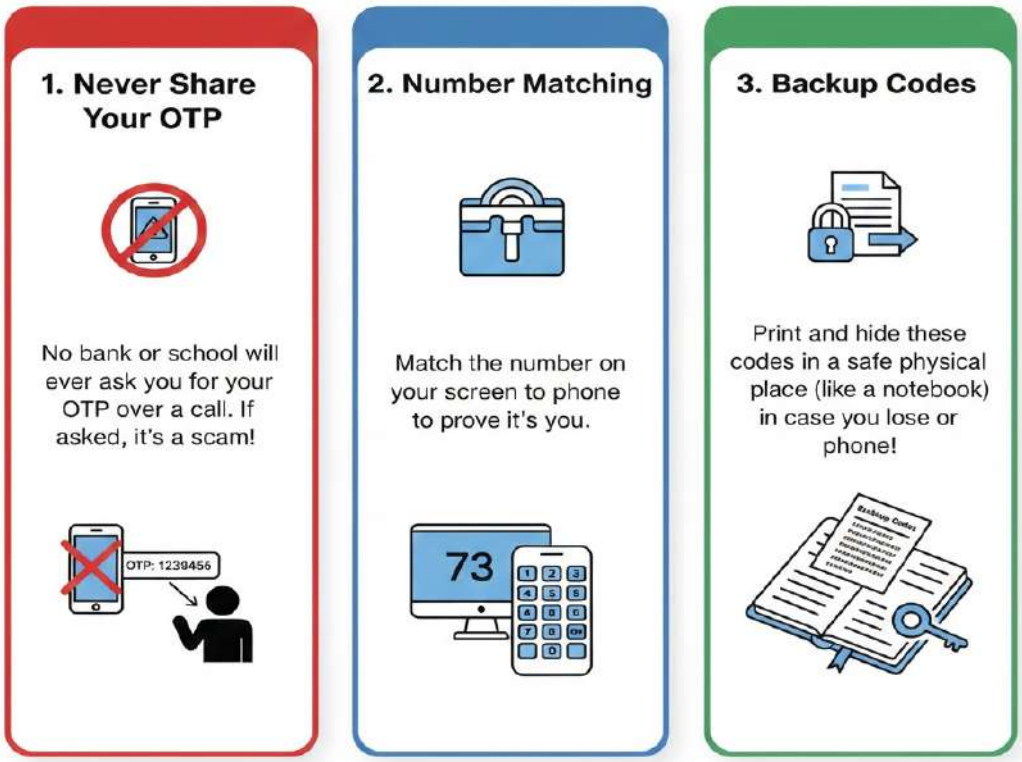


Fig. 1.5: Security with MFA

1.3.2 Key benefits of Multi-Factor Authentication

key benefits of Multi-Factor Authentication (MFA) with real-world examples:

- **Neutralizes Stolen Passwords:** If a hacker guesses your Gmail password, they are still blocked because they don't have the OTP sent to your phone.
- **Defeats Phishing Scams:** Even if you accidentally enter your login details on a fake Instagram site, the hacker cannot enter without your biometric (fingerprint) scan.
- **Acts as a Security Alarm:** You know someone is trying to hack you immediately when you receive a push notification on your phone while you aren't even logged in.
- **Protects Financial Transactions:** When using UPI (like PhonePe or GPay), the second factor (your secret PIN) ensures that even if someone has your phone, they can't spend your money.
- **Secures Personal Identity:** Using MFA on your Aadhaar account ensures that only you can download your digital ID by entering the code sent to your registered mobile number.
- **Prevents "Credential Stuffing":** If your gaming account password is leaked, MFA stops hackers from using that same password to break into your **school email**.

- **Ensures Device-Specific Access:** By using a Physical Security Key (like a USB), you ensure that your computer can only be unlocked when that specific device is plugged in.

1.3.3 Real-World Examples of MFA

MFA is being used in many real-life applications. Here are several real-life examples of Multi-Factor Authentication (MFA), some of them are illustrated in Figure 1.6.

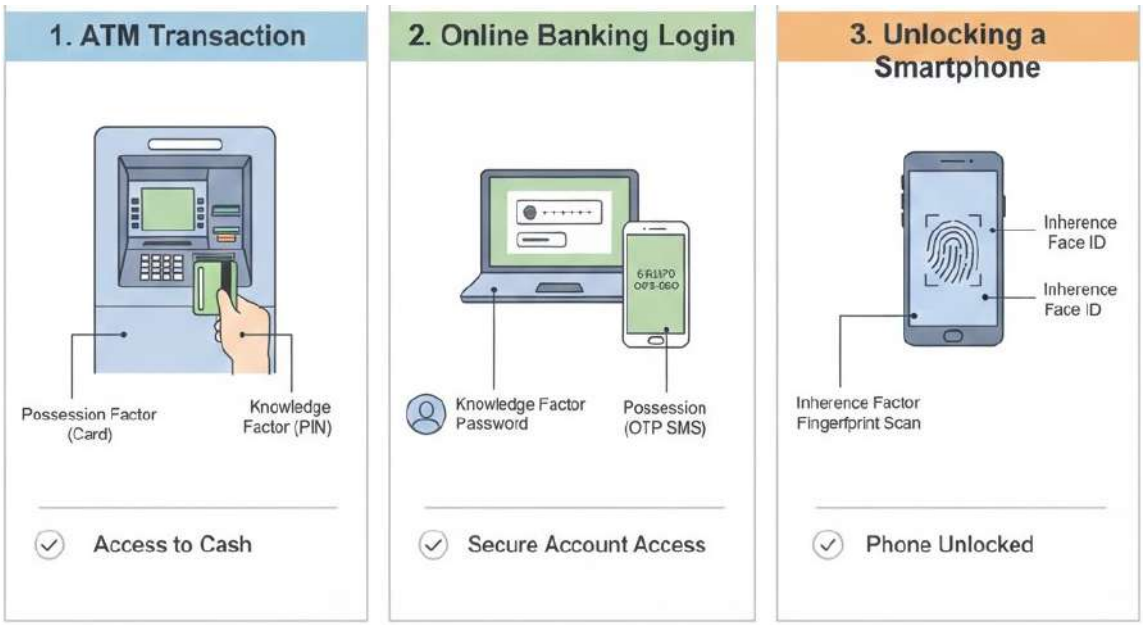


Fig. 1.6 : Real-World Examples of MFA

1. **ATM Cash Withdrawal:** Requires your physical Debit Card (something you have) and your **secret PIN** (something you know).
2. **UPI Payments (Gpay/PhonePe):** Requires your **registered smartphone** (something you have) and your UPI PIN (something you know).
3. **Gmail Login:** Requires your account password (something you know) and a Google Prompt or SMS code on your phone (something you have).
4. **WhatsApp Web:** Requires your phone's physical presence to scan a QR code using your biometric fingerprint or Face ID.
5. **Aadhaar Verification:** Requires your 12-digit Aadhaar number (something you know) and an OTP sent to your linked mobile or a fingerprint scan (something you are).
6. **Instagram Login:** Requires your username/password (something you know) and a 6-digit code from an authenticator app (something you have).
7. **Smartphone Unlock:** Requires a screen pattern/PIN (something you know) and your fingerprint or Face ID (something you are).

8. **Online Exam Portals:** Require your student ID/password (something you know) and a webcam face scan to verify it is really you (something you are).
9. **Amazon/Flipkart Shopping:** Requires your saved card details (something you know) and an OTP from your bank (something you have) to finish the payment.

1.4 Types of MFA

Multi-Factor Authentication (MFA) is categorized based on the different "factors" used to prove who you are. To be true MFA, a system must use at least two different types from this list.

- **Knowledge Factor (Something you know):** Entering a password or a secret **PIN** to unlock your personal email or social media account.
- **Possession Factor (Something you have):** Receiving a 6-digit OTP (One-Time Password) via SMS on your phone to authorize an online purchase.
- **Inherence Factor (Something you are):** Using your fingerprint or **Face ID** to quickly unlock your smartphone or a banking app.
- **Location Factor (Somewhere you are):** Being allowed to log into a school computer only when your GPS location shows you are on the **school campus**.
- **Hardware-based Factor (A physical gadget):** Plugging a USB Security Key (like a YubiKey) into a laptop and tapping it to prove you are physically present.

Fig. 1.5 : Types of MFA

Real-Life Example

Type	The Simple Rule	Real-Life Example
Knowledge	What you *know*	Your ATM PIN
Possession	What you *have*	An OTP sent to your phone
Inherence	What you *are*	Scanning your Fingerprint
Location	Where you *are*	Being at your Home or School
Hardware	What you *hold*	Using a USB Security Key

1.5. Setting Up MFA on Cloud Accounts

Multi-Factor Authentication (MFA) on cloud accounts is a security method that requires users to provide two or more proofs of identity before accessing cloud services. Instead of relying only on a password, MFA adds extra verification such as a one-time code sent to a mobile phone, an authenticator app, a hardware token, or biometric identification like fingerprint or face recognition. This additional layer of protection makes cloud accounts much safer because even if a password is stolen, unauthorized users cannot log in without the second verification factor. MFA is therefore an essential practice for protecting cloud data, applications, and user identities.

1.5.1 Enabling MFA on AWS

Steps to enable MFA on AWS:

Step 1. Sign in to the AWS Management Console as the root user or as an administrator.

Step 2. Navigate to the IAM service. IAM stands for Identity and Access Management. This is where you manage users and their security settings.

Step 3. Choose “Users” from the left navigation menu and select the user you want to enable MFA for.

Step 4. Select the “Security credentials” tab and then click “Assign MFA device” or “Manage” next to MFA.

Step 5. Choose the type of MFA device. AWS offers several options:

1. Virtual MFA device (like Google Authenticator or Authy)
2. U2F security key (like YubiKey)
3. Hardware TOTP token (a hardware device that displays codes)

Step 6. If you choose a virtual MFA device, AWS will display a QR code. Open your authenticator app on your phone and scan the QR code. The app will start generating codes.

Step 7. Enter two consecutive codes from your authenticator app into the AWS console to verify that the setup worked correctly.

Step 8. Click “Assign MFA” to complete the process.

1.5.2 Enabling MFA on Azure

Steps to enable MFA on Azure:

Step 1. Log into Azure portal

Step 2. Open Security settings

Step 3. Select Multi-Factor Authentication

Step 4. Choose verification method

Step 5. Register phone/app

Step 6. Verify code

1.5.3 Enabling MFA on Google Cloud

Steps to enable MFA on Google Cloud:

Step 1. Open Google Account security

Step 2. Go to 2-Step Verification

Step 3. Add phone or authenticator app

Step 4. Enter verification code

Step 5. Activate MFA

1.6. Best Practices for Account Security

1.6.1 Use Strong, Unique Passwords

Strong password is not easy to guess. It can be a combination of letters, number and symbols. For each account there should be different password. Some of the examples of strong password can be as:

Password	**Reason for Strong Password**
R!yaStudy#2026	This password contains uppercase and lowercase letters, includes numbers Uses special characters and has more than 10 characters.
Cl0ud@Secure99	This password is hard to guess. It is mix of letters, numbers, symbols and it is not a simple word.
My#Laptop$2025	This password is a long password, it includes symbols. It is easy for user but hard for hacker.
T3ch!Learn#Cloud	It is a combination of words, number replacing letters and special symbols.
G00d#Day@Study	It uses substitutions, contains symbols and it is not predictable.

Strong Passwords

These passwords are strong and cannot be easily hacked, because, these are:

- Are long (8–12+ characters)
- Mix letters, numbers, symbols
- Avoid personal details
- Avoid dictionary words alone
- Are unique for each account

Easy Trick for Students

Use a passphrase method:

Take a sentence:

"**I love cloud learning in 2026!**"

Password becomes:

ILvCl0ud!2026

It is easy to remember but strong.

1.6.2 Always Enable MFA

MFA is essential for cloud accounts, especially those with administrative privileges. This is the single most effective step you can take to protect your accounts. MFA provides extra protection and should always be turned on. When choosing an MFA method, prefer authenticator apps over SMS where possible. Authenticator apps provide better security and work even without cellular service. Enable MFA on all your important accounts: email, social media, banking, cloud services, and any other service that contains sensitive information or could be used to access other accounts.

1.6.3 Never Share Credentials

The login credentials are personal and private. Never share your password or MFA codes with anyone, for any reason. Legitimate organizations will never ask for your password. If someone asks, it is a scam.

Also be careful about phishing, fake emails or websites that look legitimate but are designed to steal your credentials. Always check the website address before entering your password.

1.6.4 Regularly Review Account Activity

Most cloud providers and online services offer ways to review your account activity. Get in the habit of checking these regularly. If you notice anything suspicious, take action immediately: Change your password, remove unknown devices, Update passwords regularly, revoke any unfamiliar sessions, and report the incident to the service provider.

Set up alerts if the service offers them. Many services can notify you by email or SMS when certain events occur, such as sign-ins from new devices or changes to security settings.

Practical Activity 1.1. Identify Strong and Weak Passwords

Objective

To understand the difference between strong and weak passwords and learn how to create secure passwords to protect digital information.

Material Required

- Notebook and pen

Procedure

Step 1. Write down or observe a list of sample passwords such as:

- 123456
- password
- Riya123
- A@7kL!9pQ
- India@2024

Step 2. Analyze each password based on the following criteria:

- Length (at least 8 characters)
- Use of uppercase and lowercase letters
- Inclusion of numbers
- Use of special characters (e.g., @, #, $, !)
- Avoidance of common words or personal information

Step 3. Classify each password as **strong** or **weak** based on the above criteria.

Step 4. Create your own password following all the rules of a strong password.

Step 5. Share and discuss (without revealing real passwords) why certain passwords are strong or we

Observation / Result

Passwords such as 123456 and password are weak, Riya123 is moderately weak due to personal information, while A@7kL!9pQ and India@2024 are strong as they use a combination of characters and are difficult to guess.

Conclusion:

A strong password should be long, complex, and unique, using a combination of letters, numbers, and special characters, and should not contain easily guessable information.

Practical Activity 1.2. Creating a Strong Password

Objective

To create secure passwords using password rules.

Material Required

- Notebook and pen
- Computer or smartphone (optional)
- List of password rules/guidelines

Procedure

Step 1. Begin by discussing the importance of passwords in protecting online accounts and personal data.

Step 2. Note down the key features of a strong password:

- At least 8–12 characters long
- Combination of uppercase and lowercase letters
- Inclusion of numbers
- Use of special characters (e.g., @, #, $, %)
- Should not contain personal information like name or date of birth

Step 3. Ask students to think of a simple phrase or sentence (e.g., "I love playing cricket").

- Convert the phrase into a password by:
- Using initials (e.g., *Ilpc*)
- Adding numbers and special characters (e.g., *Ilpc@2026!*)
- Mixing uppercase and lowercase letters

Step 4. Write down the created password in the notebook (for practice only, not real passwords).

- Check the strength of the password using the rules discussed.

Observation

Students observe that simple passwords are easy but insecure, and by applying proper rules they can create strong and secure passwords.

Conclusion:

A strong password is created by combining letters, numbers, and special characters in a unique way, making it difficult for others to guess or hack.

Practical Activity 1.3. Demonstration of Multi-Factor Authentication

Objective

To understand the concept of Multi-Factor Authentication (MFA) and learn how it enhances security by using more than one method of verification.

Material Required

- Computer or smartphone with internet access
- A sample online account (e.g., email or school account for demonstration)
- Mobile phone for receiving OTP (One-Time Password)
- Notebook and pen

Procedure

Step 1. Begin with a discussion on authentication and why passwords alone may not be सुरक्षित enough.

Step 2. Explain the concept of Multi-Factor Authentication, which uses at least two of the following:

- Something you know (password)
- Something you have (OTP on phone)
- Something you are (fingerprint/face recognition)

Step 3. Open a demo account (or simulate the process) on a device.

Step 4. Enter the username and password (first factor).

Step 5. Enable or demonstrate the second factor authentication (such as OTP): Enter the OTP received on the registered mobile number or email

Step 6. (Optional) Show a third factor like fingerprint or face unlock if available on the device.

Step 7. Ask students to note down the steps and observe the difference between single-factor and multi-factor authentication.

Observation/ Result

Students observe that MFA uses multiple verification steps and provides stronger security by preventing access even if the password is known.

Conclusion

Multi-Factor Authentication makes online accounts more secure by requiring multiple forms of verification, reducing the chances of unauthorized access.

Practical Activity 1.4. Identifying Authentication Factors

Objective

To develop an understanding of authentication and its importance in ensuring secure access to digital systems by identifying and differentiating various authentication factors.

Material Required

- Chart paper / A4 sheets
- Pens / markers
- Sample list of authentication methods (printed or written on board)
- Sticky notes (optional)

Procedure

Step 1. Begin with a brief explanation of authentication as the process of verifying the identity of a user.

Step 2. Introduce the three main types of authentication factors:

- **Knowledge Factor (Something you know):** Passwords, PINs
- **Possession Factor (Something you have):** ATM card, OTP on mobile
- **Inherence Factor (Something you are):** Fingerprint, face recognition

Step 3. Divide the class into small groups (3–5 students each).

Step 4. Provide each group with a list of examples such as:

- Password
- OTP received on phone
- Fingerprint scan
- Security question
- Smart card
- Face unlock

Step 5. Ask each group to:

- Identify the type of authentication factor for each example.
- Create a table or chart showing classification.

Step 6. Each group presents their classification and reasoning to the class.

Step 7. Teacher reviews the answers and clarifies any misconceptions.

Observation

Students are able to classify authentication factors correctly and understand that using multiple authentication methods improves security and promotes safe digital practices.

Practical Activity 1.5. Case Study Discussion — Why MFA is Needed

Objective

To understand the real-life importance of Multi-Factor Authentication (MFA) in enhancing security and preventing unauthorized access.

Material Required

- Printed case study handouts or board-written scenario
- Chart paper / A4 sheets
- Pens / markers
- Sticky notes (optional)

Procedure

Step 1. Begin with a brief explanation of Multi-Factor Authentication (MFA) as a security method that uses two or more authentication factors.

Step 2. Present a simple case study to the class, for example:
A student shares their password with a friend or uses a weak password, and someone else gains unauthorized access to their account.

Step 3. Divide the class into small groups (3–5 students each).

Step 4. Ask each group to discuss and analyze the case study based on the following points:

- What went wrong in the given situation?
- What risks or consequences occurred?
- How could the problem be prevented?
- How would using MFA (e.g., password + OTP or fingerprint) improve security?

Step 5. Students note their discussion points on chart paper.

Step 6. Each group presents their analysis and suggestions to the class.

Step 7. Conclude with a teacher-led discussion highlighting the importance of using multiple authentication factors for better security.

Observation

Password-only security is vulnerable, while Multi-Factor Authentication (MFA) adds extra layers of protection and improves overall account security.

Summary

In this session, students learned about cloud security and how it protects data and applications from unauthorized access. They understood the shared responsibility model, where both users and cloud providers work together to ensure security. Students learned about authentication, especially the use of usernames and passwords, and why passwords alone are not secure. They

explored Multi-Factor Authentication (MFA), which adds extra security using multiple factors like password, OTP, and biometric verification. They also learned that MFA can be used on platforms like Amazon Web Services, Microsoft Azure, and Google Cloud Platform. Overall, the session helped students understand the importance of strong passwords, MFA, and safe practices to protect cloud accounts.

Check your progress

A. Multiple Choice Questions (MCQs)

1. What is the main purpose of authentication? (a) To delete data (b) To verify user identity (c) To speed up internet (d) To store files
2. Which of the following is an example of "Something you have"? (a) Password (b) PIN (c) OTP on mobile (d) Security question
3. Which service model gives maximum security responsibility to the cloud provider? (a) IaaS (b) PaaS (c) SaaS (d) LAN
4. Which of the following is a weak password? (a) R!ya#2026 (b) Cl0ud@Secure99 (c) 123456 (d) T3ch!Learn#Cloud
5. MFA stands for: (a) Multi-File Access (b) Multi-Factor Authentication (c) Managed Firewall Access (d) Multiple Form Access

B. Fill in the Blanks

1. Authentication verifies the _____ of a user.
2. Cloud security follows the _____ Responsibility Model.
3. OTP is an example of something you _____.
4. MFA adds an extra layer of _____.
5. Fingerprint is an example of something you _____.

C. True or False

1. Passwords alone are always fully secure.
2. MFA uses two or more authentication factors.
3. Cloud users are responsible for protecting their login credentials.
4. "1234" is considered a strong password.
5. MFA helps prevent unauthorized access.

D. Short Answer Questions

1. What is authentication?
2. Why are passwords considered weak?
3. Define Multi-Factor Authentication (MFA).
4. Name any three authentication factors.
5. Write two benefits of enabling MFA.

Session 2. Managing Access in Cloud Computing using IAM (Users, Roles and Permissions)

At TechLearn Company, the cloud system is like a smart digital office building with many rooms. Rahul the developer needs access to the "Server Room," Meena from operations needs entry to the "Monitoring Room," and Arjun from finance only needs to enter the "Billing Room." Instead of giving everyone a master key, the cloud administrator uses IAM to give each person a special access card that opens only the rooms required for their job. This follows the Principle of Least Privilege — no extra access, no unnecessary risk. Even the company's website application gets its own temporary access pass (IAM role) to upload files to one specific storage area, and the pass expires automatically. Because everyone has limited and proper access, the company's cloud remains safe, organized, and protected from mistakes or misuse. Figure 2.1 illustrates TechLearn's cloud office. In this session we are going to discuss IAM Roles and Permissions.

Fig. 2.1: TechLearn's Cloud Office

2.1. Introduction to Access Control

Access control is a security method used to decide who can enter a system, what they can see and what actions they can perform.

In cloud computing, not every user should have full access to all resources. For example, a student should not access financial records or a developer should not change billing settings.

Access control protects data by allowing **only authorized users** to perform specific tasks. Figure 2.2 shows different access controls.

Fig. 2.2: Different Access Controls

Principle of Least Privilege

The Principle of Least Privilege means; A user should get only the minimum access required to do their work.

For example:

- A student can view assignments but cannot delete server data.
- A finance officer can view billing but cannot change application code.

This reduces risk because:

- Mistakes are minimized
- Unauthorized changes are prevented
- Security is improved

2.2 Identity and Access Management (IAM)

IAM stands for Identity and Access Management. It is a system in cloud computing that manages users (identity) and controls their permissions (access).

2.2.1 IAM as a Service in Cloud Platforms

Most cloud platforms provide IAM as a built-in service.

Cloud Provider	**IAM Service Name**
Amazon Web Services (AWS)	AWS Identity and Access Management (IAM)
Microsoft Azure	Microsoft Entra ID (formerly Azure Active Directory)

Google Cloud	Cloud Identity and Access Management (Cloud IAM)

IAM service allows cloud administrators to:

- *Manage users:* IAM helps administrators create, update, or remove user accounts in the cloud system.
- *Set permissions:* IAM allows administrators to decide what actions users can perform, such as read, write, or delete.
- *Control resource access:* IAM ensures users can access only the specific cloud resources they are authorized to use.
- *Monitor activities:* IAM enables administrators to track user actions and login activities for better security.

Each component helps control access properly.

2.2.2 Core Components of IAM

IAM mainly includes:

- *Users:* Individual accounts created for people or services to access cloud resources.
- *Groups:* Collections of users that share the same permissions for easier management.
- *Roles:* Temporary permission sets assigned to users or applications to perform specific tasks.
- *Policies:* Rules that define what actions are allowed or denied on particular cloud resources. Figure 2.3 shows IAM components.

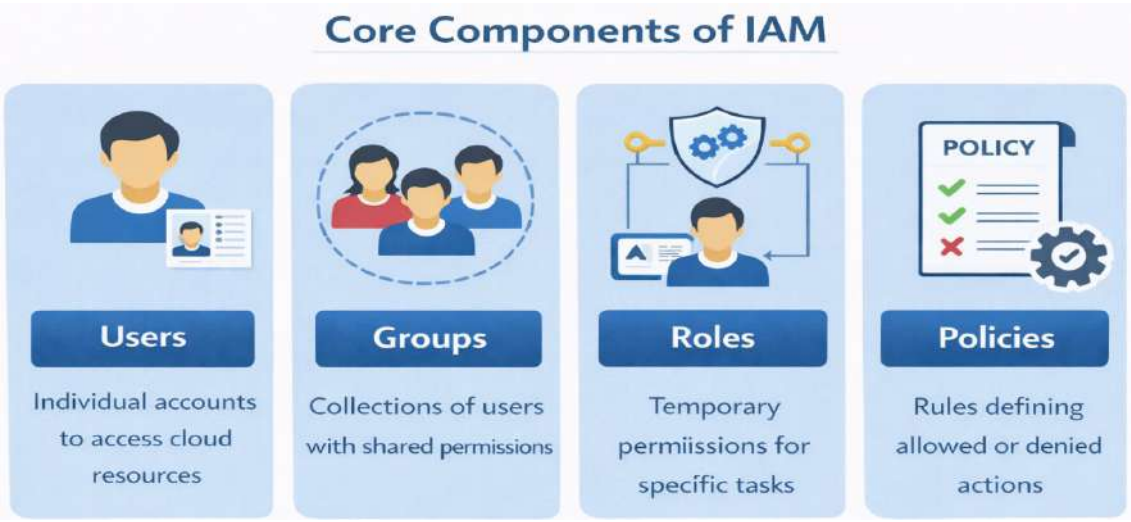


Fig. 2.3: Core Components of IAM

2.2.3 Core Functions of IAM

- *Create user accounts:* IAM allows administrators to add new users to the cloud system with secure login details.
- *Assign roles:* IAM enables administrators to give specific responsibilities and permissions based on a user's job.
- *Control who can access cloud resources:* IAM ensures that only authorized users can view, modify, or manage specific cloud services and data. Figure 2.4 shows the core functions of IAM.

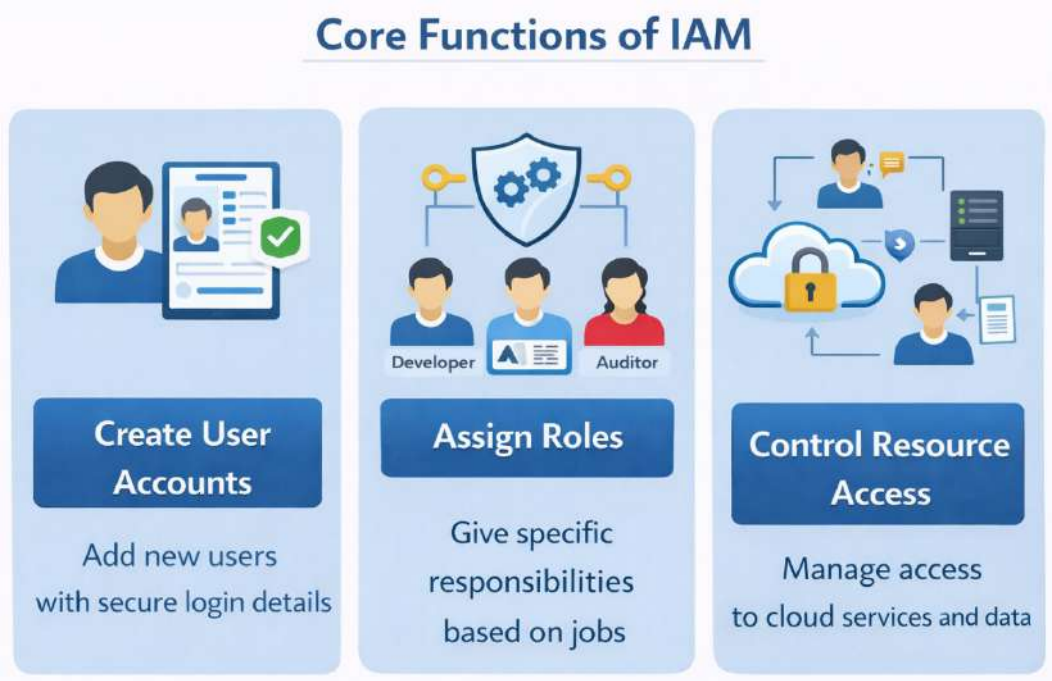


Fig. 2.4: Core Functions of IAM

2.3. IAM Users

An IAM user is a digital identity for a person or application that needs to access cloud resources. It has an account in the cloud platform.

Examples:

- Teacher
- Developer
- System administrator
- Application

Each user has Username, Password and Permissions.

2.3.1 Creating and Managing Users

When a new person joins a company, the IAM team creates a user for them. Cloud administrators can:

- *Create new users:* Administrators add new user accounts with access type, so individuals can access cloud services securely.

- ***Console access:*** They can log in through a website (needs password)
- ***Programmatic access:*** They can use APIs and command line (needs access keys)
- *Delete users:* Administrators remove user accounts when access is no longer required.
- *Reset passwords:* Administrators change or recover passwords if users forget them or for security reasons.
- *Assign permissions:* Administrators decide what actions users are allowed to perform in the cloud system.

Figure 2.5 shows cloud user management.

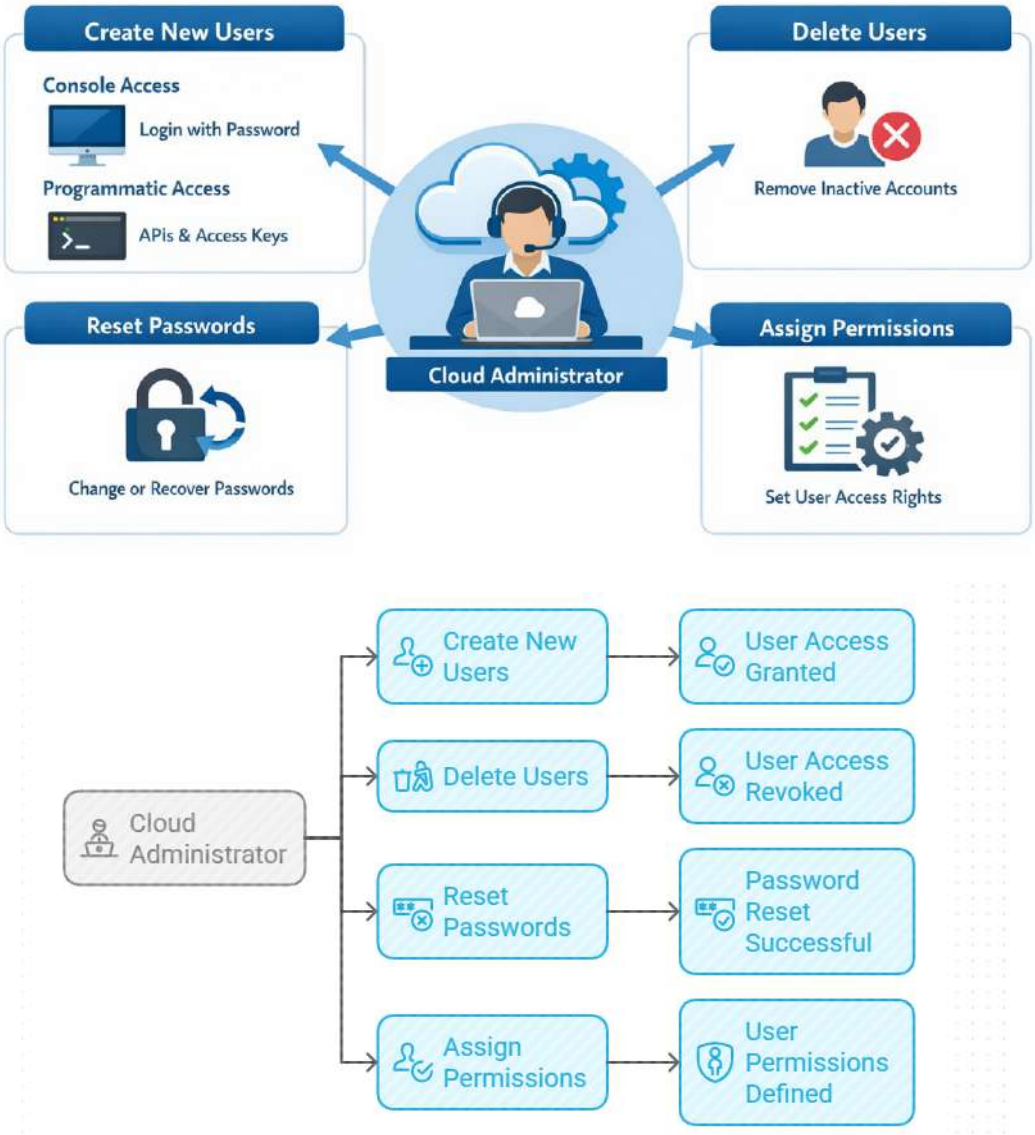


Fig. 2.5: Cloud User Management

Users are managed carefully to avoid misuse.

2.3.3 User Credentials

Credentials are the keys to log in to your account. There are two types of credentials:

1. Passwords: Used for logging into the web console (the website where you manage cloud resources). Just like your email password, it should be strong and secret.

2. Access Keys: Access keys are secret codes used by software to connect to cloud services securely. It is used for programmatic access – when a program or script needs to access cloud services without a human typing a password.

An access key has two parts:

- **Access Key ID:** Like a username (not secret)
- **Secret Access Key:** Like a password (MUST be kept secret)

2.4. IAM Group

An IAM group is a collection of users. Instead of assigning permissions to each user separately, we assign them to a group.

Think of a group of Class 10 students. They have the same schedule. Instead of giving each student a separate timetable, they have been given a single time table. In the same way, in IAM, instead of giving permissions to each user one by one, they all put in a group to assign the permission to all the members of group.

For example:
Group Name: Developers
Members: All developer users

2.4.1 Organizing Users into Groups

Users can be grouped based on job roles as shown in table 2.1 to make management easier.

Table 2.1: Users in groups

Group Name	Who is in it?	What they need
Developers	People who build applications	Create and manage servers, deploy code
Operations	People who keep systems running	Monitor systems, check logs, respond to problems
Finance	People who track costs	View bills and spending, but NOT change anything
Auditors	People who check security	View everything but change nothing

2.4.2 Assigning Permissions to Groups

Permissions are given to the group. When you attach a policy (rules) to a group, all members automatically get those permissions. *Example:* If Developers group has access to compute services, all developers get access.

This saves time and reduces errors.

2.5. IAM Roles

An IAM role is a temporary identity that can be assumed by anyone who needs it. It's not tied to one specific person. It is a set of permissions that can be temporarily assigned to:

- Users
- Applications
- Cloud services

Roles do not have permanent login credentials.

2.5.1 Roles vs. Users

IAM User	IAM Role
Has username & password	No permanent credentials
Permanent identity	Temporary permissions
Used by people	Used by services/applications

2.5.2 When to Use Roles

Roles are used in many situations, and these roles can be temporarily assigned. *For example,* an application needs to access cloud storage. Instead of storing passwords, it uses a role.

Roles are used for:

1. Applications Need Access: Imagine an app running on a server needs to read files from storage. Instead of putting a password in the app, create a role with read permissions. The server gets temporary credentials automatically.

2. One Service Needs to Talk to Another: In the cloud, different services need to work together. A role gives them permission to do this safely.

3. Cross-Account Access: A big company might have separate cloud accounts for different teams. A developer in the "Development" account might need temporary access to a database in the "Production" account. A role makes this possible without creating a permanent user in the production account.

4. Temporary Credentials: Roles provide temporary access credentials. These are more secure, automatically expire and reduce risk of misuse. Sometimes a normal user needs extra power for a specific task (like fixing an urgent problem). Instead of giving them permanent admin access, create a role with those permissions and assign them for specific time.

2.6. IAM Policies

A policy is a document that defines what actions are allowed or denied. It's like a written rule, that defines:

- What actions are allowed
- On which resources
- For which users or roles

Policies control permissions.

2.6.1 JSON Format of Policies

Policies are written in JSON (JavaScript Object Notation) format.

2.6.2 Elements of a Policy

A policy mainly includes:

Effect: This says whether you are Allowing or Denying something.

- "Allow" – yes, you can do this
- "Deny" – no, you cannot do this

If there is a conflict, deny always wins. Even if another policy says Allow, a Deny will override it.

Examples:

- "Effect": "Allow" → User is allowed to perform the action.
- "Effect": "Deny" → User is not allowed to perform the action.

Action: This says What operation can be done such as Read, Write or Delete.

Examples:

- "Action": "s3:Read" → User can read files from storage.
- "Action": "ec2:StartInstances" → User can start virtual servers.
- "Action": "dynamodb:DeleteItem" → User can delete database records.

Resource: This says Which cloud resource such as Storage bucket, Server or Database.

Examples:

"Resource": "ProjectFilesBucket" → Applies to a specific storage bucket.

"Resource": "Server123" → Applies to a specific virtual machine.

"Resource": "BillingDashboard" → Applies to billing service.

Simple Combined Example

```
{
  "Effect": "Allow",
  "Action": "s3:Read",
  "Resource": "ProjectFilesBucket"
}
```

Meaning:

The user is allowed to read files from the ProjectFilesBucket.

2.6.4 Managed Policies vs Inline Policies

There are two ways to create policies:

Managed Policy

Pre-created by cloud provider or reused across many users.

These are pre-created by cloud provider that you can attach to many users, groups, or roles. They are reusable.

- **AWS Managed Policies** – Created by AWS for common situations (like "ReadOnlyAccess")
- **Customer Managed Policies** – Created by you for your specific needs

Benefits: Change the policy once, and it updates everywhere it's attached.

Inline Policy

These are policies embedded directly into one specific user, group, or role. They cannot be reused.

Use when: You have a very special permission that only one specific person needs.

Best Practice: Use managed policies most of the time. They are easier to manage.

Managed policies are easier to manage.

Comparison: Managed Policies vs Inline Policies

Feature	Managed Policies	Inline Policies
Definition	Pre-created or reusable policies provided by cloud platform or created once and reused.	Custom policies created and attached directly to a single user, group, or role.
Reusability	Can be attached to multiple users, groups, or roles.	Attached to only one specific user, group, or role.
Management	Easy to manage and update in one place.	Must be edited separately for each entity.
Maintenance	Changes apply to all entities using the policy.	Changes affect only the specific attached entity.
Best For	Common permissions used across many users.	Special permissions for one particular user or role.

Feature	Managed Policies	Inline Policies
Flexibility	Less flexible for unique cases.	Highly customized for specific needs.
Example	“ReadOnlyAccess” policy used for many finance users.	A custom policy giving one user access to a specific storage bucket.

2.7. Real-World Example – IAM for a Company

Imagine a company using cloud services. Imagine a company called **Tech Innovate** that builds software using cloud services. They have different teams with different needs.

2.7.1 Developers Group: Full Access to Compute Services

Who: The developers who write code and build applications.

What they need:

- Create new servers
- Start and stop servers
- Deploy applications
- Delete servers when no longer needed

Solution: Create a "Developers" group. Attach policies that give full access to compute services.

When a new developer joins, they are added to this group. They immediately have all the permissions they need. No extra work!

2.7.2 Operations Group: Full Access to Monitoring and Logs

Who: The operations team who keeps systems running smoothly.

What they need:

- View system health
- See logs to find problems
- Set up alerts
- BUT NOT change the applications or create new servers

Solution: Create an "Operations" group. Give them permissions to monitoring tools and logs, but NOT to compute services.

2.7.3 Finance Group: Read-Only Access to Billing

Who: The finance team who tracks spending.

What they need:

- View bills and costs
- See usage reports

- BUT NOT change any technical resources

Solution: Create a "Finance" group. Give them read-only access to billing. They can see costs but cannot accidentally delete anything.

2.7.4 Application Role: Limited Access to Specific S3 Bucket

Who: An application that runs on servers.

What it needs:

- Read configuration files from one specific bucket
- Write log files to another specific bucket
- NOT access any other buckets or services

Solution: Create a role called "AppServiceRole" with exactly these permissions. Attach this role to the servers running the application. The servers get temporary credentials automatically. No passwords stored anywhere!

This follows the Principle of Least Privilege.

Summary Table: TechInnovate's IAM Setup

Group/Role	Who/What	Permissions
Developers Group	All developers	Full access to compute services
Operations Group	Operations staff	Full access to monitoring and logs
Finance Group	Finance team	Read-only access to billing
AppServiceRole	Application servers	Read from config bucket, write to logs bucket

Practical Activity 2.1. Understanding IAM Users, Groups & Roles

Objective

To understand how IAM controls access using Users, Groups, Roles, and Policies.

Material Required

- Chart paper / A4 sheets
- Pens / markers
- Sample scenario (written on board or printed)
- Sticky notes (optional)

Procedure

Step 1. Begin with a brief explanation of IAM as a system used to manage access to digital resources.

Step 2. Introduce the key components:

- **Users:** Individual persons (e.g., student, teacher)
- **Groups:** Collection of users with similar access (e.g., students group)
- **Roles:** Permissions assigned based on responsibilities (e.g., admin, viewer)

Step 3. Present a simple scenario, for example:
A school uses a digital system where students can view content, teachers can upload content, and the principal can manage all activities.

Step 4. Divide the class into small groups (3–5 students each).

Step 5. Ask each group to:

- Identify different users in the scenario.
- Create groups based on similar roles.
- Assign appropriate roles and permissions to each group.

Step 6. Students prepare a chart showing:

- Users → Groups → Roles → Permissions

Step 7. Each group presents their model to the class.

Step 8. Teacher reviews and explains the correct mapping and importance of IAM in security.

Observation

Students observe that users, groups, and roles have distinct functions, with groups simplifying access control and roles ensuring secure and appropriate permissions.

Practical Activity 2.2. Permission Matching Worksheet

Objective

To match correct IAM component with real-life scenario.

Material Required

- Printed permission matching worksheets
- Pens / pencils
- Chart paper (optional)
- Blackboard / whiteboard

Procedure

Step 1. Briefly explain IAM components: **Users, Groups, Roles, and Permissions** with simple examples.

Step 2. Distribute worksheets containing:

- Column A: Real-life scenarios (e.g., A student can only view study material, A teacher uploads assignments, An admin manages all users).
- Column B: IAM components (User, Group, Role, Permission).

Step 3. Instruct students to match each scenario with the correct IAM component.

Step 4. Ask students to justify their answers by explaining:

- Who is involved (User/Group)
- What responsibility is assigned (Role)
- What action is allowed (Permission)

Step 5. Students complete the worksheet individually or in pairs.

Step 6. Conduct a class discussion to review answers and clarify concepts.

Observation

Students correctly match IAM components with real-life scenarios and understand how users, roles, and permissions work together to manage access control.

Practical Activity 2.3. Simple Policy Understanding

Objective

To develop an understanding of access control policies and their role in defining permissions for users in a digital system.

Material Required

- Printed sample policies or board-written examples
- Chart paper / A4 sheets
- Pens / markers
- Sticky notes (optional)

Procedure

Step 1. Introduce the concept of a **policy** as a set of rules that defines what actions are allowed or denied for users.

Step 2. Provide simple examples of policies, such as:

- Students can view and download study material but cannot delete it.
- Teachers can upload and edit content.
- Admin has full access to manage users and data.

Step 3. Divide the class into small groups (3–5 students each).

Step 4. Assign each group a set of policy statements.

Step 5. Ask students to:

- Identify the user/group involved.
- Identify the action/permission (view, edit, delete, upload).
- Determine whether the action is allowed or denied.

Step 6. Students prepare a table showing:

- User/Group → Action → Allowed/Denied

Step 7. Each group presents their understanding to the class.

Step 8. Teacher summarizes and explains how policies help in maintaining security and proper access control.

Observation

Students understand that policies define allowed and denied actions and can identify users, actions, and permissions, recognizing their importance in ensuring secure system access.

Summary

In this session, students learned about access control and how it manages who can access cloud resources and what actions they can perform. They understood the Principle of Least Privilege, which ensures users get only the minimum permissions required for their tasks. Students explored Identity and Access Management (IAM) and its key components such as users, groups, roles, and policies. They learned how IAM users represent individual identities, groups help manage multiple users, roles provide temporary access, and policies define permissions using rules. They also understood the difference between managed and inline policies. Additionally, they learned that major cloud platforms like Amazon Web Services, Microsoft Azure, and Google Cloud Platform provide IAM services. Overall, the session helped students understand how proper access control improves security and protects cloud resources.

Check Your Progress

A. Multiple Choice Questions (MCQs)

1. What is the main purpose of IAM in cloud computing?
   (a) Increase internet speed (b) Manage identity and control access
   (c) Store multimedia files (d) Design websites
2. The Principle of Least Privilege means:
   (a) Give all users full access (b) Give users temporary admin rights
   (c) Give users minimum required access (d) Deny access to everyone
3. Which IAM component provides temporary permissions?
   (a) User (b) Group (c) Role (d) Password

4. Which document defines allowed or denied actions on resources?
   (a) Policy (b) Password (c) Access Key (d) Console
5. Which IAM component is best for assigning the same permissions to many users?
   (a) Role (b) Group (c) Access Key (d) Secret Key

B. Fill in the Blanks

1. IAM stands for Identity and _____ Management.
2. A collection of users with similar permissions is called a _____.
3. Roles provide _____ credentials.
4. Policies are written in _____ format.
5. In case of conflict, _____ always overrides Allow.

C. True or False

1. IAM Users have permanent login credentials.
2. Roles are mainly used by applications or services.
3. Inline policies can be reused across many users.
4. Groups make permission management easier.
5. The Principle of Least Privilege increases security risk.

D. Short Answer Questions

1. What is IAM?
2. Define the Principle of Least Privilege.
3. Differentiate between IAM User and IAM Role (one point).
4. What is the purpose of an IAM Group?
5. Name any two elements of a policy.

Session 3. Encryption Basics and Secure Data Storage in Cloud Computing

A city hospital uses a cloud-based healthcare application to store and manage patient records. When a patient's medical report is saved in the cloud database, it is automatically encrypted, which means the information is converted into a secret code (ciphertext) so no unauthorized person can read it — this is called encryption at rest.

When a doctor opens the report on a tablet, the data travels securely from the cloud server to the device using HTTPS and TLS encryption, which protects it during transmission — this is encryption in transit. Even if a hacker tries to intercept the data, they will only see unreadable coded information. The hospital also uses a Key Management Service (KMS) to securely manage encryption keys. By using encryption at rest, encryption in transit, and proper key management, the hospital keeps patient data safe and follows data protection laws.

Figure 3.1 shows securing patient data. In this session we are going to discuss about encryption basics & secure storage.

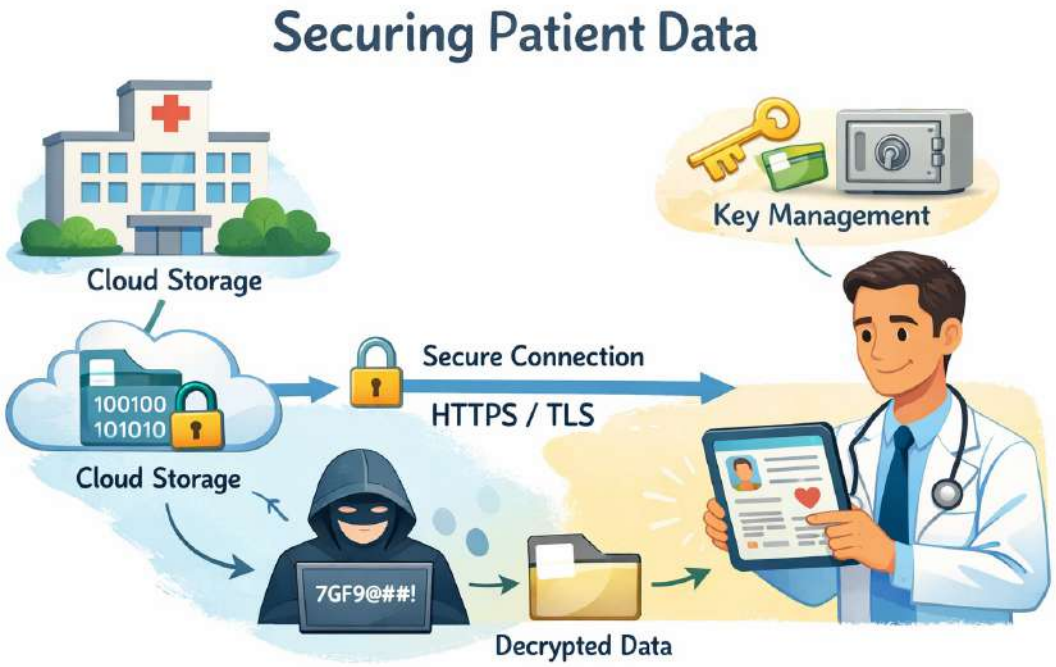


Fig. 3.1: Securing Patient Data

3.1. Encryption

Encryption is the process of converting normal data into a secret code so that unauthorized people cannot read it. Figure 3.2 shows encryption cycle. It protects:

- Passwords
- Bank details
- Medical records
- Personal messages

Only authorized users with the correct key can read the data.

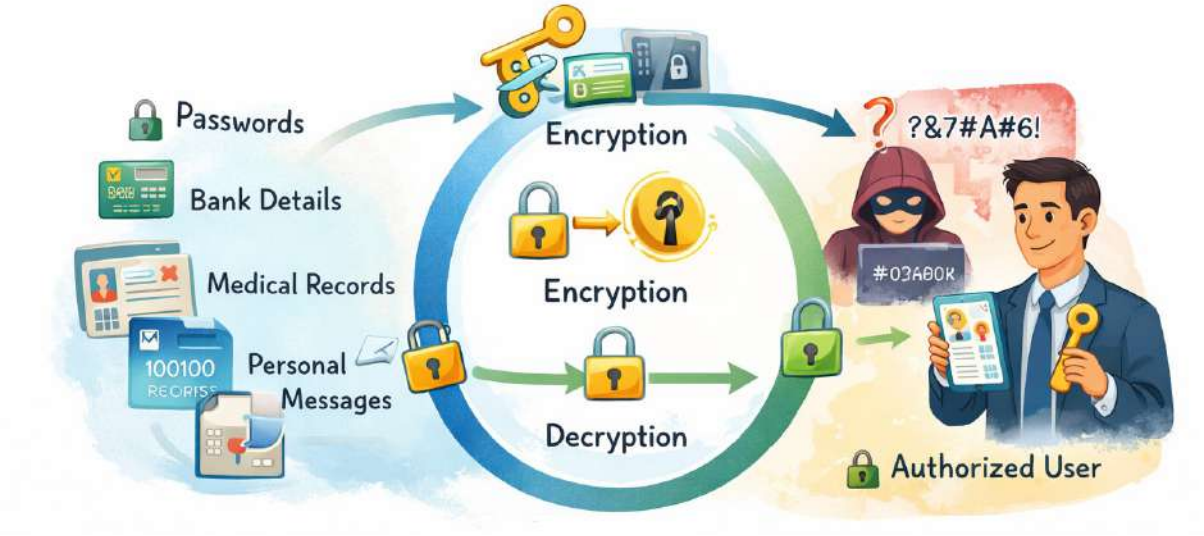


Fig. 3.2: Encryption Cycle

3.1.2 Plaintext vs Ciphertext

Plaintext is normal readable information, while ciphertext is scrambled coded information created to protect data from unauthorized access.

Plaintext → Original readable data

Example: "My password is 1234"

Ciphertext → Encrypted unreadable data

Example: "XyT!89@#Lm"

Encryption changes plaintext into ciphertext. Figure 3.3 shows comparison between plaintext and ciphertext.

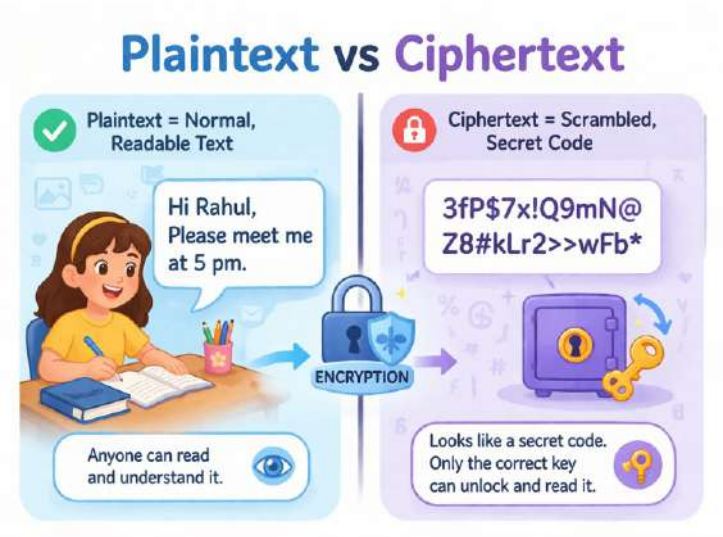


Fig. 3.3: Comparison between Plaintext and Ciphertext

More Examples

Example of Plaintext	**Example of Ciphertext**
Hello Student	K#9Lp@2xQ
My ATM PIN is 4567	Xz!78@LmP0
Patient ID: 1023	Qw$8rT@9Lm
Email Password: Study@2026	9Lm#K2@pQr!
Transfer Rs 5000	T$7yP!xZ92

3.1.3 Role of Keys in Encryption

A key is a special code used to encrypt data, that is, lock it and decrypt data, that is, unlock it.

Without the correct key, encrypted data cannot be read.

Think of it like:

Lock → Encryption

Key → Secret password to open it

Examples of Role of Keys in Encryption

Encryption keys are secret codes used to lock (encrypt) and unlock (decrypt) data.

Example 1: Password Protection

- Plaintext: My Password
- Encryption Key: Key123
- Ciphertext: Scrambled unreadable data

Only someone with Key123 can convert ciphertext back to plaintext.

Role of Key: Controls who can read the password.

Example 2: WhatsApp Message Security

When you send a message:

- Message is encrypted using a secret key.
- Only the receiver's key can decrypt it.

Role of Key: Ensures only sender and receiver can read messages.

Example 3: Cloud Storage Files

A company stores files in cloud storage:

- Files are encrypted using a key.
- Without that key, files appear unreadable.

Role of Key: Protects stored files from hackers.

Example 4: Online Banking

- When logging into banking:
- Website uses encryption keys to secure login data.
- Browser and server exchange keys safely.

Role of Key: Protects login credentials during transmission.

Example 5: Digital Locker Documents
Government digital locker stores certificates:

- Documents encrypted using encryption keys.
- Only authorized users with correct keys can open them.

Role of Key: Maintains privacy of documents.

3.2. Types of Encryptions

There are two main ways to encrypt data. They use keys differently.

3.2.1 Symmetric Encryption

Symmetric encryption uses the same key for encrypting and decrypting.
Example: If one key locks the data, the same key unlocks it.
Symmetric encryption is used for:

- Encrypting stored data
- Large files
- Databases
- Advantage of symmetric encryption is that it is fast and efficient.

3.2.2 Asymmetric Encryption

Asymmetric encryption uses two different keys:

- Public Key → Used to encrypt data
- Private Key → Used to decrypt data

The public key can be shared, but the private key must be kept secret.
Asymmetric encryption is used for:

- Secure communication
- HTTPS websites
- Digital signatures

Figure 3.4 shows symmetric and asymmetric encryption process.

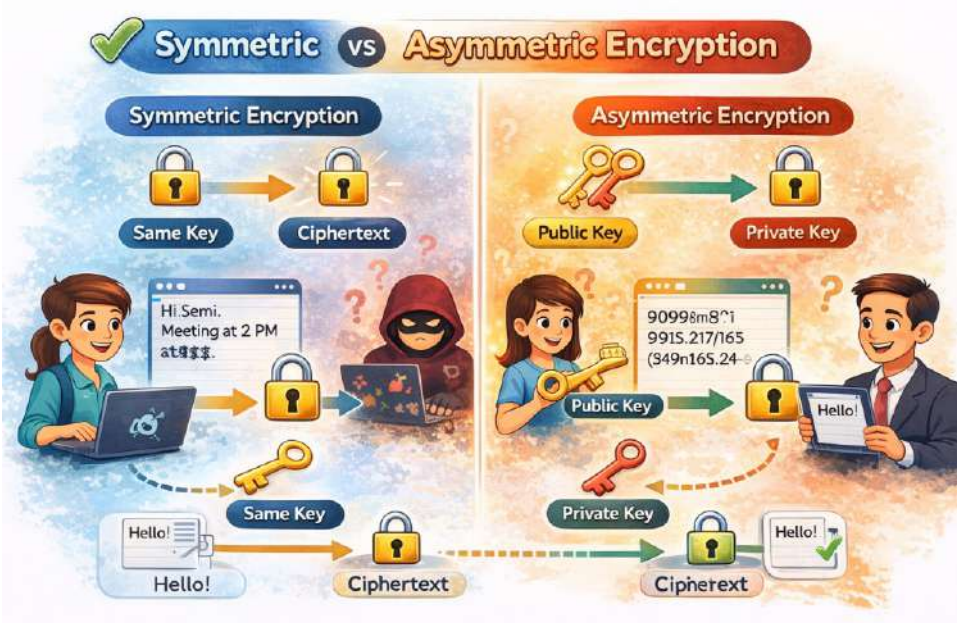


Fig. 3.4: Symmetric and Asymmetric Encryption

3.2.3 Comparison and Use Cases

Feature	Symmetric Encryption	Asymmetric Encryption
Keys	One key for everything	Two keys: public and private
Speed	Very fast	Much slower
Key Sharing	Hard – must share secretly	Easy – public key can be shared openly
Best For	Encrypting large amounts of data	Sharing keys, digital signatures
Example	AES	RSA

Both types are often used together in cloud systems.

When you visit a secure website (HTTPS), your browser uses asymmetric encryption to safely share a temporary symmetric key. Then it uses symmetric encryption (which is faster) for the actual data. Thus, it makes use of best of both.

3.3. Encryption at Rest

3.3.1 Data at Rest

Data at rest means data that is stored and not moving. Moving data, also called data in transit, is information that is being transferred from one place to another through a network or the internet.

Examples of data at rest:

- Files saved in cloud storage
- Database records
- Backups

3.3.2 Encrypting Stored Data

Cloud systems encrypt:

- Storage buckets
- Databases
- Backup files

This ensures that even if storage is accessed illegally, data remains unreadable.

3.3.3 How Cloud Providers Implement Encryption at Rest

Cloud providers automatically encrypt stored data using strong encryption methods.

This includes:

- Disk-level encryption
- Database encryption
- Backup encryption

Most cloud services enable encryption by default. Let us see how it works:

Simple Version:

- You enable encryption on your storage (often just a checkbox)
- When you upload a file, the cloud automatically encrypts it before saving
- When you download the file, the cloud automatically decrypts it
- You don't even notice it happening!

Behind the Scenes (Simplified):

- Each file gets its own unique encryption key
- That key is itself encrypted with a master key
- The master key is stored safely in a special service

This is called **envelope encryption** – like putting a letter in an envelope, then putting that envelope in another envelope!

3.3.4 Customer-Managed Keys vs Provider-Managed Keys

You have a choice about who controls the keys:

Provider-Managed Keys

- Cloud Company controls and manages encryption keys.
- Easy to use.

Customer-Managed Keys

- Customer create and manage your own keys
- Customer control their own encryption keys.
- More secure but requires proper management.
- Required by some organizations for extra security

3.4. Encryption in Transit

3.4.1 Data in Transit

Data in transit means data that is moving between systems over a network.

Examples:

- Data sent from browser to cloud server
- Data transferred between cloud services
- Online payments and shopping
- Sending a message on WhatsApp
- Logging into Social Media
- Watching a YouTube video
- Uploading a photo to Google Drive

3.4.2 Risks of Unencrypted Transmission

If data is not encrypted then hackers can intercept it. Sensitive information can be stolen. An identity theft may occur.

What they could steal:

- Your passwords
- Your credit card numbers
- Your private messages
- Your photos

Tampering

Someone could change your data while it's traveling. They could:

- Change the amount in a bank transfer
- Modify a message you sent
- Inject viruses into files you download

Impersonation

Someone could pretend to be a website you trust and trick you into giving them information.

This is why encryption in transit is essential.

3.4.3 TLS/SSL Protocols

The technology that encrypts data in transit is called TLS (Transport Layer Security) and SSL (Secure Sockets Layer). These are security protocols used to encrypt data during transmission.

When you see lock symbol in browser, HTTPS in website address, then it means TLS/SSL encryption is active. Figure 3.5 shows SSL/TSL symbol.

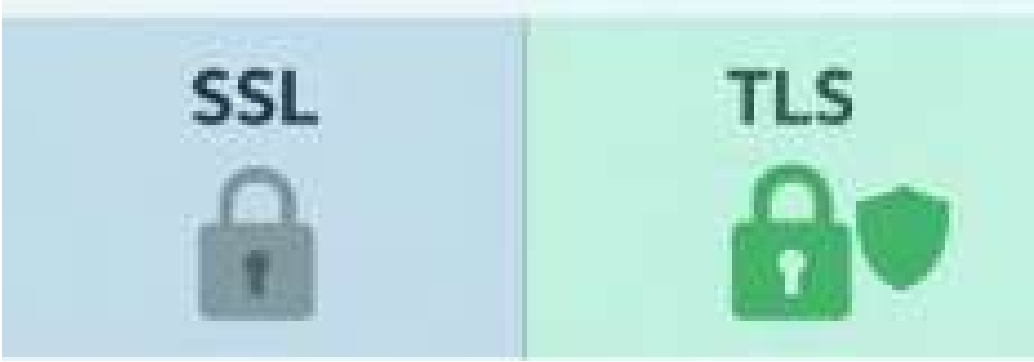


Fig. 3.5: SSL/TSL

How it works?

Look at your browser's address bar. If you see:

- **https://** at the beginning of the web address
- A small **padlock icon** 🔒

Your connection is encrypted!

What TLS does:

- ***Encryption:*** Scrambles your data so no one can read it
- ***Authentication:*** Confirms you are talking to the real website, not a fake one
- *Integrity:* Makes sure your data wasn't changed during transmission

3.4.4 How Encryption in Transit Protects Data

Encryption in transit:

- Scrambles data before sending
- Prevents hacking
- Protects login details

- Secures online payments

Even if intercepted, the data cannot be read.

Let's follow your data on a secure connection:

Step 1. Enter the URL https://www.amazon.in in your browser

Step 2. The browser and Amazon's server do a "handshake" – they agree on how to encrypt and exchange keys safely

Step 3. From this point on, EVERYTHING is encrypted:

- Your login credentials
- Your search queries
- Your credit card information
- The pages you view

Step 4. If anyone tries to intercept your data, they see only random gibberish

Result: Your information stays private and safe!

Look at your browser and check whether the website address starts with http:// or https://? . Do you see the padlock?

3.5. Secure Storage in the Cloud

Cloud providers offer many services to keep your data safe. Let's look at some of them.

3.5.1 Encrypted Cloud Storage Services

(a) AWS – S3 Server-Side Encryption

It allows to store files in the cloud. When you upload a file to Amazon's cloud storage service called Simple Storage Service (S3), the system automatically encrypts the file before saving it on the server. This encryption converts your data into a secret code so that unauthorized users cannot read it.

You do not need to manually encrypt the file — AWS handles the encryption process in the background and securely stores the encryption keys. When an authorized user downloads the file, it is automatically decrypted and made readable again. This helps protect important data such as documents, images, backups, and project files stored in the cloud.

You can enable encryption with one click:

SSE-S3: Amazon manages the keys for you (simplest)

SSE-KMS: You manage keys using AWS Key Management Service (more control)

SSE-C: You provide your own keys (most control)

(b) Azure – Storage Service Encryption

Azure – Storage Service Encryption (SSE) is a security feature in Microsoft Azure that automatically encrypts your data before it is saved in cloud storage. This means that when you upload files, images, documents, or database data to Azure Storage, the system converts the information into a secure coded format (ciphertext) so that unauthorized users cannot read it.

The encryption and decryption happen automatically in the background, so users do not need to perform any extra steps. When an authorized user accesses the data, Azure decrypts it safely. This ensures that stored data remains protected at rest in the cloud.

Azure automatically encrypts all data written to its storage services.

- Uses 256-bit AES encryption
- Microsoft manages keys unless you choose otherwise

(c) Google Cloud – Default Encryption at Rest

Most modern cloud storage uses default encryption. Google Cloud automatically encrypts all customer data at rest before it's written to disk. Also enabled by default!

- Uses AES-256 encryption
- Google manages keys by default
- You can use your own keys if you prefer

With all these services, encryption happens automatically. The data stored in the cloud remains safe.

3.5.2 Encrypted Database Services

Cloud databases encrypt:

- Customer data
- Transaction records
- Account information

This prevents unauthorized reading of stored data. The following cloud providers offer the database encryption service.

Provider	**Database Encryption Service**
AWS	Amazon RDS Encryption – encrypts databases, backups, and snapshots
Azure	Azure SQL Database Transparent Data Encryption (TDE) – automatic
Google Cloud	Cloud SQL Encryption – automatic encryption for all data

3.5.3 Key Management Services (KMS)

KMS is a service used to:

- Create encryption keys
- Store keys securely
- Control access to keys

It helps manage encryption securely in cloud systems. The key management service provided by the cloud providers are:

Provider	**Key Management Service**
AWS	AWS Key Management Service (KMS)

Azure	Azure Key Vault
Google Cloud	Google Cloud Key Management

What these services do:

- Create and store encryption keys securely
- Rotate keys automatically (create new ones regularly)
- Control who can use which keys
- Keep logs of when keys are used
- Protect keys using special hardware (HSMs)

3.6. Real-World Example – Healthcare Application

Imagine a hospital using a cloud-based medical app. A real-world example of cloud security can be seen in a healthcare application used by a hospital to store patient records. When a patient's medical reports, prescriptions, and test results are saved in the cloud, the data is encrypted so that it cannot be read by unauthorized people. This is called encryption at rest. Figure 3.6 shows cloud security in healthcare.

Fig. 3.6: Cloud Security in Healthcare

When a doctor accesses the records from a computer or tablet, the data travels securely over the internet using encrypted connections (HTTPS/TLS), which protects it during transmission — this is encryption in transit. Only authorized doctors and staff with proper login credentials can view or update the records. By using encryption and secure access control, the hospital protects patient privacy and follows data protection laws.

3.6.1 Encrypting Patient Records at Rest

The patient data contains:

- Patient names, addresses, phone numbers
- Medical histories
- Test results

- Doctor's notes
- X-ray images and reports

All patient records stored in the cloud are encrypted. If someone accesses the database illegally, they see unreadable data.

Database Encryption: All patient information in the database is encrypted using transparent data encryption. Even if someone steals the database file, they cannot read it.

Storage Encryption: All medical images are stored in encrypted cloud storage. Each image is encrypted with its own key.

Backup Encryption: All backups are also encrypted. Even backup tapes stored off-site are safe.

Key Management: The encryption keys are stored in a Key Management Service, with strict controls on who can use them.

Result: If an attacker breaks into the storage system, they find only encrypted garbage. Useless without the keys!

3.6.2 Encrypting Data in Transit

When a doctor views patient records, data travels:

- From database to application server
- From application server to doctor's browser
- From doctor's browser back to application server (when updating records)

Browser to App: The website uses HTTPS (TLS). The doctor sees the padlock icon. All communication is encrypted.

App to Database: Even though this traffic stays within the cloud provider's network, it's still encrypted.

Result: If anyone intercepts network traffic at any point, they see only encrypted data. No patient information is exposed.

3.6.3 Compliance with Data Protection Laws

Healthcare data is protected by strict laws. In India, this includes the Digital Personal Data Protection Act and healthcare regulations.

Encryption helps the hospital:

- Protect patient privacy
- Follow government regulations
- Avoid legal penalties

Practical Activity 3.1. Understanding Plaintext and Ciphertext (Manual Encryption Activity)

Objective

To understand how encryption converts readable data into coded form.

Materials Required

Notebook, Pen, Blackboard

Procedure

Step 1. Write a simple message in your notebook:
"Cloud Data is Safe"

Step 2. Use a simple rule: Replace each letter with the next letter in the alphabet

A → B

B → C

C → D

Step 3. Encrypt the message using the rule.

Example:
C → D
L → M

Step 4. Write the encrypted version (Ciphertext).

Step 5. Exchange your ciphertext with your partner.

Step 6. Partner must decrypt using the reverse rule.

Observation

1. Original message = Plaintext
2. Encrypted message = Ciphertext
3. Rule used = Encryption Key

Practical Activity 3.2. Identifying Encryption in Daily Life

Objective

To identify where encryption is used in real life.

Materials Required

- Chart paper / A4 sheets
- Pens / markers
- Internet-enabled device (optional)
- List of common digital services (written on board or printed)

Procedure

Step 1. Begin with a brief explanation of **encryption** as a method of converting data into a secure form to prevent unauthorized access.

Step 2. Provide examples of daily life applications, such as:

- Secure websites (HTTPS)

- Online banking and payments
- Messaging apps
- Email services

Step 3. Divide the class into small groups (3–5 students each).

Step 4. Assign each group the task to identify situations in daily life where encryption is used.

Step 5. Ask students to:

- List at least 4–5 examples of encrypted activities.
- Explain how encryption is useful in each case (e.g., protecting passwords, securing transactions).

Step 6. Students prepare their findings on chart paper.

Step 7. Each group presents their examples and explanations to the class.

Step 8. Teacher summarizes key points and highlights the importance of encryption in ensuring privacy and security.

Observation / Result

Students recognize that encryption is widely used in daily life to protect sensitive information and ensure secure communication.

Practical Activity 3.3. Demonstration to Understand Encryption at Rest in Cloud Storage

Objective

To understand encryption at rest in cloud storage.

Materials Required

- Computer/mobile with internet access
- Cloud storage account (e.g., Google Drive or similar)
- Sample file (document/image)
- Notebook and pen

Procedure

Step 1. Log in to a cloud storage account using valid credentials.

Step 2. Upload a sample file (document or image) to the cloud storage.

Step 3. Open the uploaded file to verify successful storage.

Step 4. Log out of the account and attempt to access the file without logging in (to observe access restriction).

Step 5. Log in again and access the file using correct credentials.

Step 6. Explain that the stored file is protected using **encryption at rest**, which secures the data even while it is stored on servers.

Step 7. Discuss how only authorized users with proper access can view or modify the file.

Observation / Result

Students observe that encryption at rest secures stored data and prevents unauthorized access.

Practical Activity 3.4. Differentiating Symmetric and Asymmetric Encryption

Objective

To differentiate symmetric and asymmetric encryption.

Material Required

- Chart paper / A4 sheets
- Pens / markers
- Blackboard / whiteboard
- Slips of paper for message writing

Procedure

Step 1. Briefly explain encryption and introduce two types: **symmetric** and **asymmetric encryption**.

Symmetric Encryption Activity:

- Teacher provides a common secret key (e.g., shift each letter by +2).
- Students write a message and encrypt it using the given key.
- Exchange messages with a partner and decrypt using the same key.

Asymmetric Encryption Activity:

- One student creates a simple "public rule" (e.g., replace vowels with numbers).
- Other students use this rule to encrypt messages.
- Only the original student explains the reverse rule ("private key") to decrypt the message.

Step 2. Students perform both activities and note:

- Number of keys used
- Ease of encryption/decryption
- Level of security

Step 3. Students prepare a comparison table highlighting differences between symmetric and asymmetric encryption.

Step 4. Teacher reviews and summarizes key differences.

Observation / Result

Students observe that symmetric encryption uses one key while asymmetric encryption uses two keys, making them different in method and security.

Summary

In this session, students learned the importance of encryption in protecting digital data by converting plaintext into ciphertext to prevent unauthorized access. They understood the role of encryption keys and explored symmetric and asymmetric encryption methods. Students also learned about encryption at rest and in transit, how HTTPS and TLS/SSL ensure secure communication, and the role of Key Management Services (KMS) in managing and protecting encryption keys.

Check Your Progress

A. Multiple Choice Questions (MCQs)

1. A company stores customer data in cloud storage. Even if someone accesses the storage illegally, the data appears unreadable. Which concept is applied here?
   (a) Encryption in transit (b) Encryption at rest (c) Firewall
   (d) Data compression
2. When you see “https://” and a padlock in the browser, it means:
   (a) Website is free (b) Website is fast (c) Data is encrypted during transmission (d) Website is government approved
3. A messaging app uses a public key to encrypt a message and a private key to decrypt it. This is:
   (a) Symmetric encryption (b) Asymmetric encryption (c) Hashing
   (d) Backup encryption
4. Which encryption type is faster and suitable for encrypting large databases?
   (a) Asymmetric encryption (b) Symmetric encryption (c) SSL
   (d) Digital signature
5. In cloud systems, encryption keys are securely created and stored using:
   (a) Antivirus software (b) Key Management Service (KMS) (c) Web browser (d) Operating system

B. Fill in the Blanks

1. Readable original data is called _____.
2. Encrypted unreadable data is called _____.

3. Encryption that protects stored files is called encryption at _____.
4. TLS/SSL protocols protect data in _____.
5. In asymmetric encryption, the _____ key is kept secret.

C. True or False

1. The same key is used for encryption and decryption in symmetric encryption.
2. Data in transit refers to data stored in cloud databases.
3. Without the correct encryption key, ciphertext cannot be read.
4. HTTPS indicates that data is encrypted during transmission.
5. Encryption removes the need for passwords.

D. Short Answer Questions

1. Differentiate between plaintext and ciphertext with one example.
2. Why is encryption in transit important during online banking?
3. Explain the difference between symmetric and asymmetric encryption.
4. What is the role of Key Management Service (KMS) in cloud security?
5. In a hospital cloud system, where would encryption at rest and encryption in transit be used?

Session 4. Secure Web Communication using HTTPS and SSL/TLS

Rohan, a Class X student, wanted to buy a science reference book online using his father's debit card. When he opened the shopping website, he first checked the address bar and saw a padlock and "https://" before the website name. This means the site was secure and using SSL/TLS encryption.

When Rohan entered the card number and password, the information was converted into encrypted code before traveling through the internet. Even if a hacker tried to intercept the data, they would only see unreadable ciphertext. The website's SSL certificate also proved that it was the real shopping site and not a fake one. Because of HTTPS, Rohan's payment was completed safely without exposing any sensitive information. Figure 4.1 shows access to secure website.

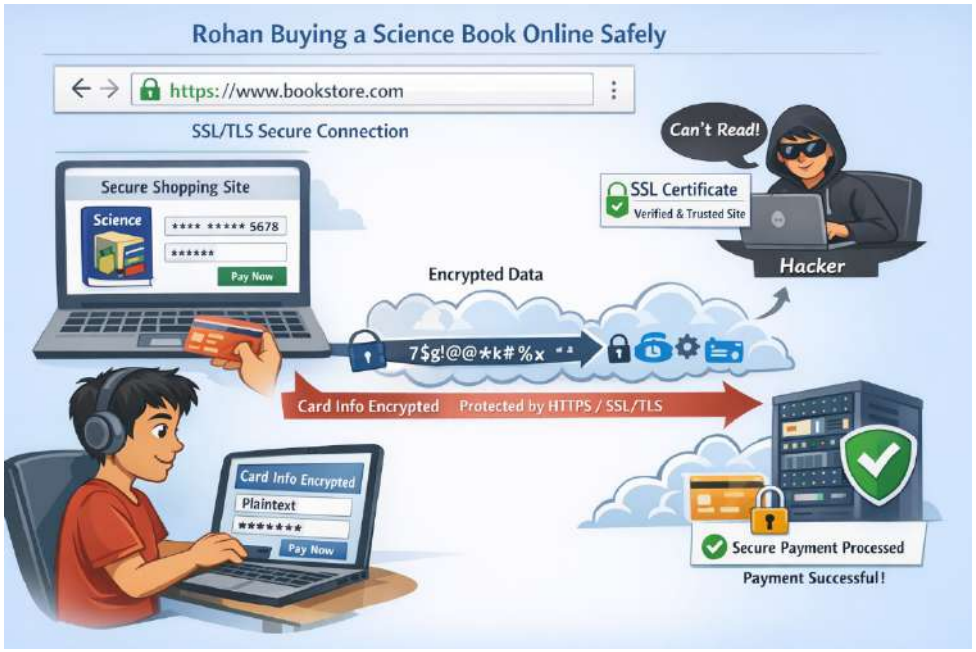


Fig. 4.1: Access to Secure Website

In this session we are going to discuss about HTTPS and SSL concept.

4.1. Hypertext Transfer Protocol (HTTP)

4.1.1 Definition

HTTP stands for Hypertext Transfer Protocol. It is a communication rule that allows your browser such as, Chrome, Edge, or Firefox to talk to a web server and load websites.

Whenever you open a website, HTTP helps send and receive data between your device (client) and website server.

Think of HTTP like a set of rules for a conversation. If you speak Hindi and someone speaks Tamil, you can't understand each other. But if you both agree to speak English, you can communicate. HTTP is the "English" that browsers and servers agree to speak.

What HTTP Does:

- When you enter a website address, your browser sends an HTTP request to the server
- The server understands the request and sends back an HTTP response
- Your browser receives the response and displays the webpage

This happens for EVERY website you visit, for EVERY image you see, for EVERY link you click.

Figure 4.2 shows HTTP communication flow.

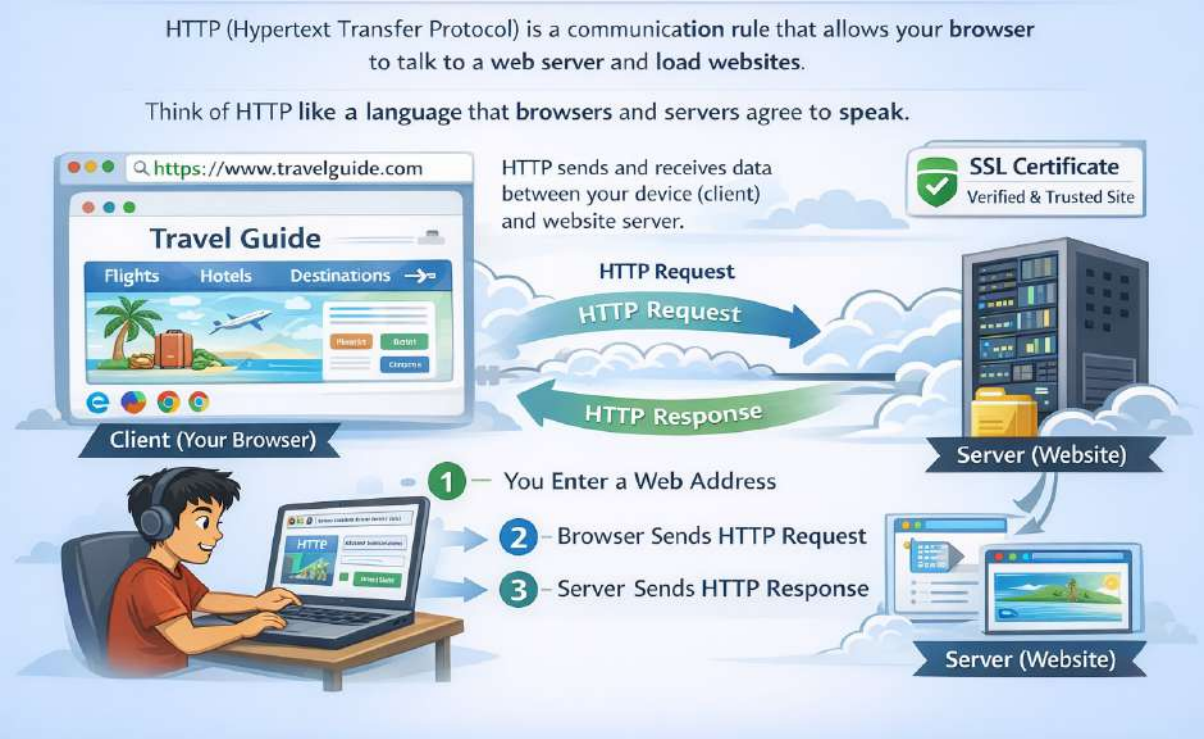


Fig. 4.2: HTTP Communication Flow

4.1.2 How HTTP Transmits Data

When you type a website address:

2. Your browser sends a request to the server.
3. The server sends back website data.
4. Your browser displays the webpage.

Example: You type: http://example.com

Your browser requests the page → Server sends page → You see website.

This process happens within seconds.

Opening the webpage with HTTP is just like sending a postcard. Anyone who handles it can read your message.

The major problem with HTTP is that HTTP sends data in plain text. Plain text means anyone intercepting the data can read it. Also, hackers on public Wi-Fi can steal information.

Example: If you enter your password on an HTTP website, it can be captured easily. HTTP does NOT provide encryption, security and identity verification.

4.2. Hypertext Transfer Protocol Secure (HTTPS)

HTTPS stands for Hypertext Transfer Protocol Secure. It is the secure version of HTTP. It protects data using encryption.

- HTTP + Encryption = HTTPS
- HTTPS is created by combining:
- HTTP + SSL/TLS encryption.

This means:

- Data is encrypted before sending.

- Hackers cannot read it.
- Communication becomes secure.

4.2.1 The Padlock Icon in Browser

When a website uses HTTPS, A padlock symbol appears in the browser address bar. The website address begins with: **https://**

The padlock means it is a secure connection, encrypted communication and valid SSL certificate.

If a website is not secure, browsers show: ⚠ "Not Secure" warning. Fig 4.3 shows secure and not secure websites.

Fig. 4.3: Secure and not secure websites

4.3. SSL/TLS?

SSL/TLS are security technologies that protect data while it travels over the internet.

4.3.1 SSL – Secure Sockets Layer

SSL (Secure Sockets Layer) is a security protocol used to protect data sent over the internet. It encrypts the information exchanged between a user's browser and a website, so that hackers cannot read or steal sensitive details such as passwords, credit card numbers, or personal information.

When a website uses SSL, the communication becomes secure and private. Although SSL is an older protocol and has mostly been replaced by the more secure TLS (Transport Layer Security), people still commonly refer to website security as "SSL." SSL helps create a safe and trusted connection between users and websites.

4.3.2 TLS – Transport Layer Security

TLS (Transport Layer Security) is a modern security protocol that protects data sent over the internet by encrypting the communication between a user's device and a web server. It is the improved and more secure version of SSL.

When you see "https://" and a padlock symbol in the browser, it means TLS is working in the background to keep the connection safe.

TLS ensures that sensitive information such as passwords, banking details, and personal data is encrypted, preventing hackers from reading or modifying it during transmission. It provides confidentiality, integrity, and secure authentication for online communication.

Versions of TLS:

Version	Year	Status
TLS 1.0	1999	Deprecated (not safe)
TLS 1.1	2006	Deprecated (not safe)
TLS 1.2	2008	Still widely used
TLS 1.3	2018	Latest and most secure

4.3.3 How SSL/TLS Works

Step 1. Handshake Process

When you open a secure website, your browser and server perform a "handshake".

This means that they agree on encryption method and they verify identity.

Step 2. Certificate Exchange

The website sends its SSL certificate to your browser.

Your browser checks if the certificate valid and if it issued by trusted authority.

Step 3. Encryption Key Establishment

Your browser and server create a secret encryption key. This key will be used for secure communication.

Step 4. Secure Data Transmission

After key creation, all data is encrypted. Passwords, card details, messages are protected. Hackers cannot read the information. Figure 4.4 shows SSL security cycle.

Fig. 4.4: SSL Security Cycle

4.4. SSL Certificate

SSL Certificate (Secure Sockets Layer Certificate) is a digital certificate that verifies the identity of a website and enables an encrypted connection between a user's browser and the website server.
It ensures that the website you are visiting is genuine and that the information you send (such as passwords, card details, or personal data) is **s**ecure and protected from hackers.

4.4.1 Contents of a Certificate

Fig. 4.5: A typical SSL Certificate

An SSL certificate contains:
1. Domain Name: A domain name is the unique and easy-to-remember address of a website on the internet. It helps users access websites without remembering complex numerical IP addresses.
For example, instead of typing a long number like 192.168.1.1, users can type a simple name like www.example.com. The domain name directs the browser to the correct web server where the website is hosted www.schoolportal.com

- *2. Issuer:* The company that issued the certificate.
- *3. Public Key:* This key is used for encryption.
- *4. Expiration Date:* Certificate is valid only for limited time.

4.4.2 Certificate Authorities (CAs)

Certificate Authorities (CAs) are trusted organizations that issue SSL certificates.
Examples of well know Cas are:

- *DigiCert:* A trusted Certificate Authority that provides paid SSL/TLS certificates for secure websites and businesses.

- *GlobalSign:* An international Certificate Authority that issues digital certificates to verify website identity and enable HTTPS security.
- *Let's Encrypt:* A free, automated Certificate Authority that provides SSL/TLS certificates to help websites enable HTTPS easily.

Browsers trust certificates are issued by approved CAs. Figure 4.6 shows CAs.

Fig. 4.6: Certificate Authorities and SSL Certificates

4.5. Why HTTPS is Essential

It is essential to search on the secured websites using HTTPS protocol because of various reasons. Some of the important reasons are mentioned below.

4.5.1 Authentication

HTTPS confirms that you are talking to the real website. It prevents phishing attacks. The SSL certificate proves the website's identity. When the browser checks the certificate and see it's valid and matches the domain, you can be sure you're on the real site.

4.5.2 Encryption

Without encryption, anyone can see your data as it travels across the internet. On public Wi-Fi, this is especially dangerous.

HTTPS encrypts: Passwords, Bank details, Personal data.

Even if someone intercepts data, it remains unreadable.

4.5.3 Integrity

Integrity means data cannot be changed during transmission. If a hacker tries to modify it, the system detects it.

4.5.4 SEO Benefits and Browser Warnings

SEO Benefits and Browser Warnings are important reasons to use HTTPS for websites. Search engines like Google give higher ranking to secure websites

that use HTTPS, which improves their visibility in search results — this is known as an SEO (Search Engine Optimization) benefit.

Secure websites also build trust among users. On the other hand, if a website uses only HTTP and is not secure, modern browsers display a “Not Secure” warning in the address bar. This warning may discourage users from entering personal information and can reduce website traffic. Therefore, using HTTPS improves both security and website credibility.

Search engines like Google prefer HTTPS websites, because of the following benefits.

- Better ranking
- More trust
- No “Not Secure” warning

Browsers block or warn users about HTTP sites. Figure 4.7 shows HTTPS benefits.

Fig. 4.7: HTTPS Benefits

4.6. HTTPS in the Cloud

HTTPS in the Cloud means using secure, encrypted connections for websites and applications hosted on cloud platforms. When a website is hosted on cloud services like AWS, Azure, or Google Cloud, HTTPS can be enabled by installing an SSL/TLS certificate.

This ensures that all communication between users and the cloud server is encrypted and protected from hackers. Cloud platforms also provide certificate management services that automatically issue and renew certificates, making security easier to manage. By enabling HTTPS in the cloud, organizations protect user data, improve trust, and ensure safe online transactions. Cloud platforms make it easy to enable HTTPS.

4.6.1 Load Balancers with SSL/TLS Certificates

In the cloud, HTTPS is often handled by load balancers. In cloud systems load balancers distribute traffic across servers. Every server has to handle encryption and decryption. Without load balancer, it uses the computing power that could be used for actual application.

The load balancer can Store SSL certificates, encrypt incoming traffic, Protect websites automatically.

With Load Balancer:

- Users connect to the load balancer using HTTPS
- The load balancer handles all the encryption/decryption work
- It forwards requests to your application servers using plain HTTP (inside the secure cloud network)
- Your servers don't waste power on encryption

4.6.2 Cloud Certificate Management Services

Cloud providers offer certificate management tools. Some popular tools are as given below.

AWS Certificate Manager: Automatically provides and renews SSL certificates.

Azure App Service Certificates: Provides secure certificates for Azure websites.

Google Cloud Certificate Manager: Manages certificates for secure web applications.

These services issue certificates, renew certificates automatically and reduce manual effort.

4.6.3 Enabling HTTPS for a Website Hosted on Cloud

Steps to enable HTTPS:

Step 1. Register domain name.

Step 2. Request SSL certificate.

Step 3. Attach certificate to web server or load balancer.

Step 4. Force redirect from HTTP to HTTPS.

Step 5. Test the secure connection.

After this, website becomes secure. Padlock appears and data is encrypted.

Practical Activity 4.1. Identifying Secure and Non-Secure Websites

Objective

To help students identify secure and non-secure websites based on browser indicators..

Materials Required

- Computer or laptop with internet connection
- Web browser (Chrome/Firefox/Edge)
- List of sample website URLs (both HTTP and HTTPS)
- Notebook and pen

Step 1. Open a web browser on your computer.

Step 2. Enter different website URLs (e.g., one starting with *http* and another with *https*).

Step 3. Observe the address bar carefully for security indicators.

Step 4. Look for the padlock icon and check if the URL begins with *https://*.

Step 5. Click on the padlock icon to view security details (certificate information).

Step 6. Compare it with a non-secure website (http://) and note the absence of security indicators.

Step 7. Repeat the process for 3–4 websites and record your findings.

Observation Table

Website	**HTTP or HTTPS?**	**Padlock Visible?**	**Secure Warning?**
Website 1			
Website 2			
Website 3			
Website 4			

Observation

Secure websites use HTTPS with a padlock icon and valid certificates for encrypted communication, while non-secure websites use HTTP and may show warnings, making data vulnerable.

Practical Activity 4.2. Explore SSL Certificate Details of a Website

Objective

To examine and understand the SSL certificate details of a website to verify its security.

Materials Required

- Computer or laptop with internet connection
- Web browser (such as Google Chrome / Mozilla Firefox)
- Access to any secure website (https://)

Procedure

Step 1. Open a web browser like Google Chrome.

Step 2. Visit a secure website (for example: https://www.google.com).

Step 3. Observe the padlock icon in the address bar.

Step 4. Click on the padlock icon to view security information.

Step 5. Select the option like "Connection is secure" or "Certificate is valid."

Step 6. Click on "Certificate" or "More Information" to open certificate details.

Step 7. Note the following details:

- Issued to (website name)
- Issued by (Certificate Authority)
- Validity period (start and expiry date)

Step 8. Repeat the activity for 1–2 different websites and compare results.

Observation

Websites with a valid SSL certificate show a padlock icon, use HTTPS, and provide secure encrypted communication.

Practical Activity 4.3. Demonstrate How Data Travels Securely Through HTTPS

Objective

To understand how data is securely transmitted over HTTPS using encryption.

Materials Required:

- Computer or laptop with internet connection
- Web browser (such as Google Chrome / Mozilla Firefox)
- Access to websites using HTTP and HTTPS

Procedure:

Step 1. Open a web browser like Google Chrome.

Step 2. Visit a website starting with **http://** (if available).

Step 3. Observe that there is no padlock icon in the address bar.

Step 4. Now visit a website starting with **https://** (for example: https://www.google.com).

Step 5. Observe the padlock icon in the address bar.

Step 6. Click on the padlock icon to view security details.

Step 7. Enter some sample text in a search box (do not enter personal information).

Step 8. Explain that in HTTPS, the data is encrypted before being sent and decrypted at the destination.

Step 9. Compare the difference between HTTP (not secure) and HTTPS (secure).

Observation

Data transmitted through HTTPS is encrypted and secure, while HTTP sends data in an unencrypted and less secure form.

Practical Activity 4.4. Demonstrate How HTTPS is Enabled in Cloud-Hosted Websites

Objective

To understand how HTTPS is enabled in cloud-hosted websites using SSL certificates.

Materials Required

- Computer or laptop with internet connection
- Web browser (such as Google Chrome / Mozilla Firefox)
- Access to a cloud-hosted website (e.g., hosted on Amazon Web Services, Microsoft Azure, or Google Cloud Platform)

Procedure

Step 1. Open a web browser like Google Chrome.

Step 2. Visit a cloud-hosted website (for example, a site deployed on Amazon Web Services).

Step 3. Observe the URL and check if it starts with **https://**.

Step 4. Look for the padlock icon in the address bar.

Step 5. Click on the padlock icon to view SSL certificate details.

Step 6. Note the issuing authority and validity period of the certificate.

Step 7. Discuss how cloud platforms automatically or manually enable HTTPS using SSL/TLS certificates.

Step 1. Compare with a non-secure HTTP website (if available).

Observation

Cloud-hosted websites use SSL certificates to enable HTTPS, ensuring secure and encrypted communication between the user and the server.

Practical Activity 4.5. Identify Signs of Unsafe Websites

Objective

To recognize common signs that indicate a website may be unsafe or insecure.

Materials Required:

- Computer or laptop with internet connection
- Web browser (such as Google Chrome / Mozilla Firefox)

Procedure

Step 1. Open a web browser like Google Chrome.

Step 2. Visit different websites and observe their URLs.

Step 3. Check whether the website uses **https://** or **http://**.

Step 4. Look for the padlock icon in the address bar.

Step 5. Click on the padlock (if available) to view security details.

Step 6. Identify warning signs such as:

- "Not Secure" label in the address bar
- Missing padlock icon
- Suspicious or misspelled domain names
- Too many pop-up ads or redirects

Step 7. Compare these features with a secure website.

Step 8. Note the differences between safe and unsafe websites.

Observation

Unsafe websites often lack HTTPS, show security warnings, and display suspicious features like unusual URLs or excessive pop-ups.

Summary

In this session, you have learned that HTTP (Hypertext Transfer Protocol) is used for communication between browsers and servers but transfers data in plain text, making it less secure, while HTTPS (Hypertext Transfer Protocol Secure) provides a safer alternative by encrypting data using SSL/TLS protocols. They understood the role of SSL certificates in verifying website identity and enabling secure connections, as well as how trusted Certificate Authorities issue these certificates. Students also explored the SSL/TLS handshake process, where a secure encryption key is established between the browser and server, ensuring that data remains confidential, protected, and securely transmitted over the internet.

Check Your Progress

A. Multiple Choice Questions (MCQs)

1. When a student visits a website with **https://**, it means the website is using: (a) File compression (b) Secure encrypted communication (c) Offline browsing (d) Data backup

2. The padlock symbol in the browser address bar mainly indicates: (a) Website is paid (b) Website is secure (c) Website is fast (d) Website is government approved
3. Which organization issues SSL certificates to websites? (a) Internet Provider (b) Certificate Authority (c) Search Engine (d) Web Browser
4. If a website uses only HTTP, the data sent between browser and server is: (a) Encrypted (b) Hidden (c) Plain text (d) Deleted
5. The first process when a browser connects to a secure website is called: (a) Login process (b) Handshake process (c) Download process (d) Installation process

B. Fill in the Blanks

1. HTTP stands for _____ Transfer Protocol.
2. The secure version of HTTP is called _____.
3. SSL certificates help create a _____ connection between browser and server.
4. Trusted organizations that issue digital certificates are called _____.
5. The process where browser and server agree on encryption is called__.

C. True or False

1. HTTPS encrypts data sent over the internet.
2. SSL certificates verify the identity of websites.
3. HTTP is more secure than HTTPS.
4. A padlock symbol indicates secure communication.
5. Certificate Authorities issue SSL certificates.

D. Short Answer Questions

1. Why is HTTPS important while making online payments?
2. What is the role of an SSL certificate on a website?
3. What is meant by the SSL/TLS handshake process?
4. What is the difference between HTTP and HTTPS?
5. Name two Certificate Authorities that issue SSL certificates.

Module 3. Modern Cloud Applications & Services

This module on Modern Cloud Applications and Services introduces how real-world digital applications are built and function using cloud technologies. In Session 1, you will learn how modern applications use cloud computing services such as IaaS, PaaS, and SaaS, along with compute services, databases, CDNs, and AI to deliver fast and efficient user experiences. In Session 2, you will understand the working of e-commerce systems, including product databases, shopping carts, order processing, and secure digital payment methods using payment gateways, HTTPS, and tokenization. In Session 3, you will explore the concept of the Internet of Things (IoT), its architecture, and the role of edge computing in enabling real-time decision-making and automation in systems like smart classrooms and smart cities. In Session 4, you will learn how real-time location and navigation systems work using GPS, cloud servers, and Maps APIs to provide accurate tracking and route information. Overall, the module enables students to understand the integration of cloud technologies in everyday applications and develop foundational knowledge of modern digital systems.

Session 1. Building Modern Applications Using Cloud Services

Riya, a Class 9 student in Jaipur, opens Instagram and instantly sees new posts, likes, and friend suggestions. Later, she plays songs on Spotify, and the music starts immediately—even though the company is based far away in Sweden.

She wonders how everything works so quickly. The next day, her teacher explains that all these photos, notifications, and songs are managed through integrated cloud services, which store and deliver data instantly over the internet, as shown in Figure 1.1.

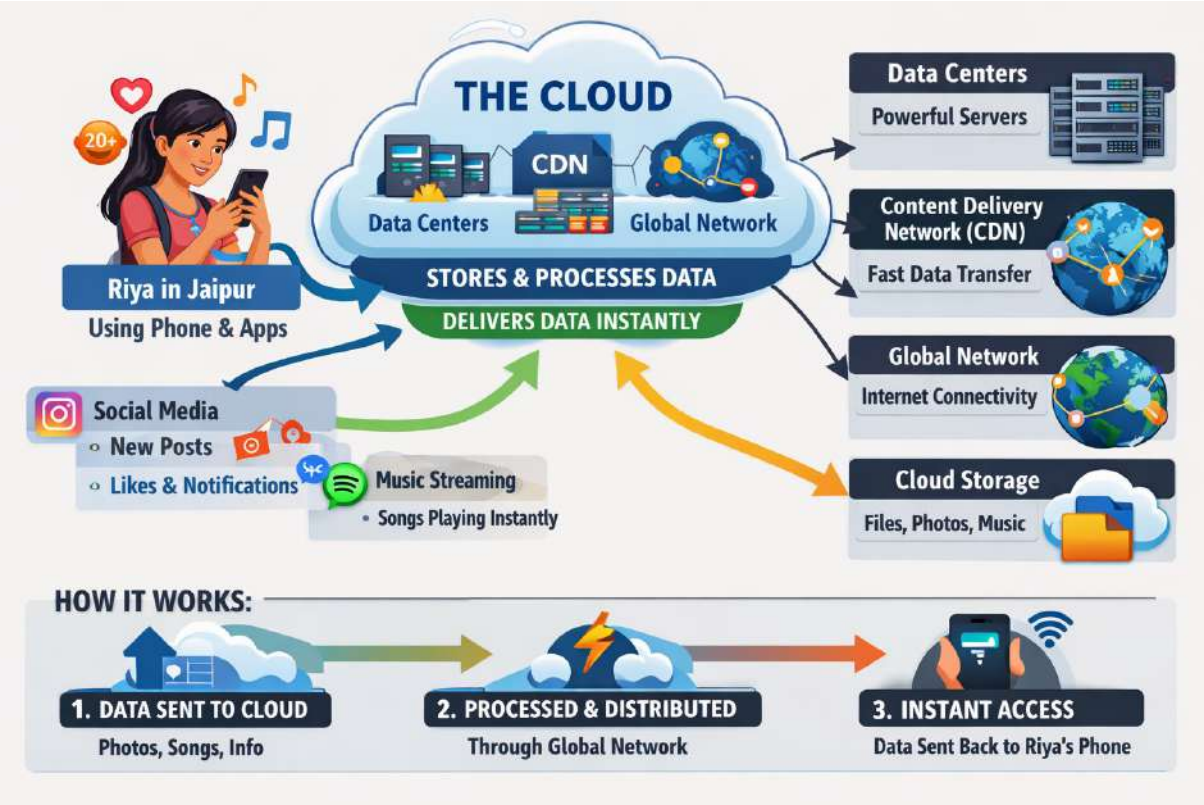


Fig. 1.1: Riya's Cloud Powered Digital Journey

1.1 How Modern Applications Are Built

1.1.1 From Scratch to Pre-built Services

Building an app today is similar to building a house using ready-made materials. Instead of creating everything from the beginning, developers use pre-built tools and services.

In the past, companies had to buy their own servers, maintain computer rooms, install software, and hire experts to manage the systems. This required a lot of time and money.

Today, developers can use ready-made services from cloud providers like Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform

(GCP). These platforms provide servers, storage, databases, and other tools online. Developers simply select the services they need and pay only for their usage.

1.1.2 Cloud Computing

Cloud computing means providing computing services such as servers, storage, databases, software, and AI through the internet.

Instead of owning physical computers, users connect to cloud providers and use their resources when needed. They pay only for the amount of service they use.

This works like electricity: people do not build their own power plants at home. They connect to the power supply and pay for the electricity they consume. Cloud computing follows the same idea for computing resources. This idea is explained in Figure 1.2.

Fig. 1.2: The Electricity Grid Analogy for Cloud Computing

1.1.3 Cloud Service Models (IaaS, PaaS, SaaS)

Cloud services come in three main models. Table 1.1 summarizes all three using a food analogy that makes them easy to remember.

Table 1.1: Cloud Service Models

Model	**Food Analogy**	**What You Manage**	**What Cloud Manages**	**Example**
IaaS (Infrastructure as a Service)	Renting a commercial kitchen — you cook everything	OS, software, data, security	Physical servers, power, network	Amazon EC2, Azure VMs
PaaS (Platform as a Service)	Meal kit delivery — ingredients	Your app code only	OS, hardware, runtime, scaling	Google App Engine, Azure App Service

	provided, you cook			
SaaS (Software as a Service)	Dining at a restaurant — just eat and enjoy	Nothing	Everything end-to-end	Gmail, Google Classroom, Netflix

This session focuses on IaaS and PaaS, because these are the building blocks developers use to create applications like the ones Riya uses every day.

1.2 Compute Services

Every time you tap a button in an app — "Like," "Search," "Buy" — your phone sends a request to a cloud server. That server must receive the request, run some logic ("Who are Riya's friends? What posts should she see?"), and send back an answer. The service that does this processing is called a Compute Service.

Compute Service: a cloud resource that runs code, processes requests, and performs the logical operations of an application.

If an app were a person, the compute service would be its brain. There are three main types, as Figure 1.3 shows.

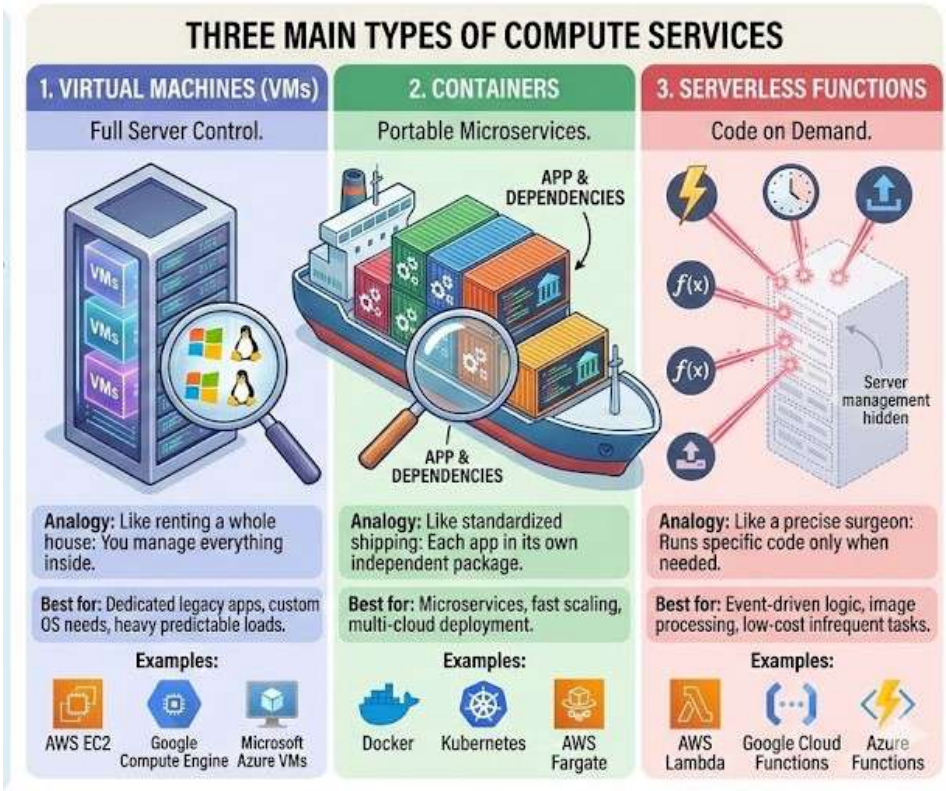


Fig. 1.3: Three Types of Compute Services

1.2.1 Virtual Machines (IaaS)

A Virtual Machine (VM) is a virtual version of a computer that runs inside a physical server. It has its own operating system, memory, CPU, and storage, just like a real computer.

A single physical server can run many VMs using special software called a hypervisor. Each VM works independently, so users can install any operating system or software they want. However, the user is responsible for managing updates, security, and maintenance. Examples include Amazon EC2, Azure Virtual Machines, and Google Compute Engine.

1.2.2 Containers (PaaS)

A container is a small package that includes an application and everything it needs to run, such as libraries and settings. This ensures the app works the same on any system.

Containers are lightweight and efficient because many of them can share the same operating system. Docker is commonly used to create containers, and Kubernetes helps manage large numbers of them. Examples of cloud container services include Amazon EKS, Azure AKS, and Google GKE.

1.2.3 Serverless (PaaS/FaaS)

Server-less computing allows developers to run code without managing servers. Developers upload small pieces of code called functions, and the cloud runs them automatically when a specific event occurs.

The cloud provider manages the infrastructure, and users pay only when the code actually runs. Examples include AWS Lambda, Azure Functions, and Google Cloud Functions.

1.2.4 Comparison of Compute Services

Table 1.2 compares all three compute types so you can quickly decide which is best for a given situation.

Table 1.2: Comparison of Compute Services

Type	Model	You Manage	Cloud Manages	Best For
Virtual Machine	IaaS	OS, software, patches, scaling	Physical server, power, network	Full control, legacy apps
Containers	PaaS	App code, container images	OS, engine, hardware	Portable, microservices apps
Serverless	PaaS/FaaS	Just your code	Servers, OS, runtime, auto-scaling	Event-driven, short tasks, APIs

1.3 Database Services — The Memory of the Application

When Riya creates her Instagram account, where does her username, password, and profile photo go? When she posts a photo, where is it recorded? When she searches for a friend, where does the app look? The answer to all three questions is: the database. The database is the memory of the application.

Database: An organized collection of data that can be stored, retrieved, and updated efficiently.

The compute service is the brain — it thinks. The database is the memory — it remembers. As shown in Figure 1.4, every request the brain processes usually require fetching or saving something to memory.

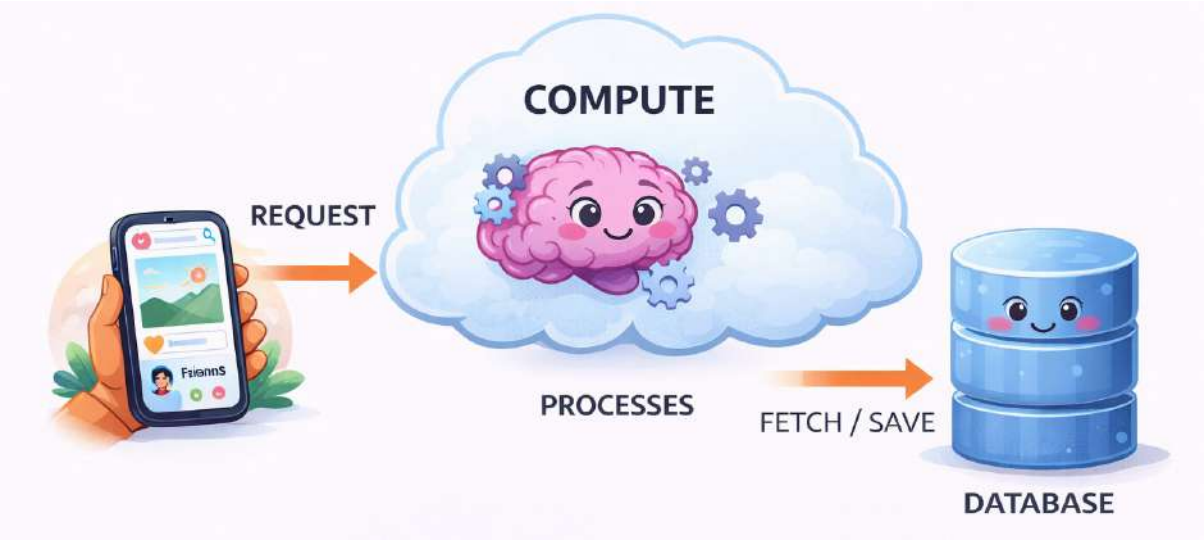


Fig. 1.4: Compute and Database Working Together

1.3.1 Structured Data: Tables, Rows, and Columns

A relational database stores data in a structured, table-based format — exactly like a spreadsheet. Table 1.3 shows a simple example.

Table 1.3: Example of Structured Data — Students Table

StudentID	FirstName	LastName	Grade
101	Riya	Sharma	10
102	Arjun	Mehta	9
103	Meera	Gupta	10

The entire grid is a Table — a collection of related records. Each horizontal line is a Row, representing one student. Each vertical category — StudentID, FirstName, LastName, Grade — is a Column, representing one type of information. This structure makes it fast to search, sort, and retrieve any specific data.

1.3.2 Primary Keys and Foreign Keys

The real power of a relational database is the ability to link tables. Figure 1.5 shows how two tables — Students and Enrolments — are linked.

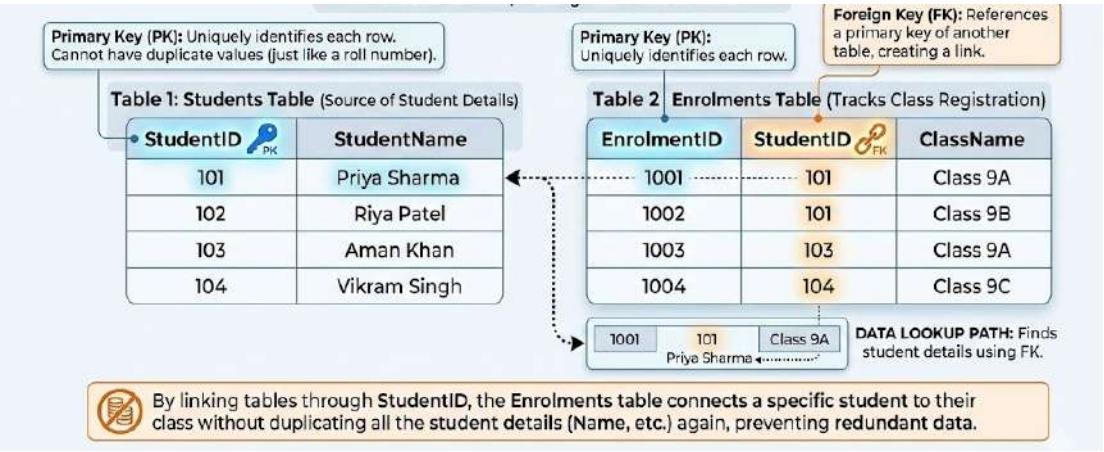


Fig. 1.5: Primary and Foreign Keys Linking Two Tables

Primary Key (PK): a column in a table that uniquely identifies each row — no two rows can share the same primary key value, just like no two students have the same roll number.

Foreign Key (FK): a column in one table that references the primary key of another table, creating a link between them.

In Figure 1.5, StudentID is the primary key in the students table and a foreign key in the Enrolments table. This link tells us exactly which student is enrolled in which class, without duplicating all the student details again.

1.3.3 Cloud Database Services

Traditionally, setting up a database required buying a server, installing software, and managing backups manually. Cloud database services handle all of this automatically. Table 1.4 shows popular databases and their managed cloud equivalents.

Table 1.4: Popular Databases and Their Managed Cloud Services

Database Engine	**Type**	**AWS Service**	**Azure Service**	**GCP Service**
MySQL	Open-source Relational	Amazon RDS for MySQL	Azure Database for MySQL	Cloud SQL for MySQL
PostgreSQL	Open-source Relational	Amazon RDS for PostgreSQL	Azure Database for PostgreSQL	Cloud SQL for PostgreSQL
SQL Server	Commercial Relational	Amazon RDS for SQL Server	Azure SQL Database	Cloud SQL for SQL Server

1.4 CDN Services — Delivering Content Fast

1.4.1 The Problem CDNs Solve

When an app's main server is located far from the user, data takes more time to reach them. This delay becomes noticeable when loading large files like photos or videos.

A CDN (Content Delivery Network) solves this problem by storing copies of content on many servers located in different places. When a user requests data, it is delivered from the nearest server, which makes websites and apps load faster.

1.4.2 What CDNs Deliver

CDNs mainly deliver static content, which does not change often for different users. Examples include images, videos, CSS files, and JavaScript files used in websites and apps.

Common CDN services include Amazon CloudFront, Azure CDN, and Google Cloud CDN. Other widely used CDN providers are Akamai and Cloudflare.

1.5 AI Services — Intelligence in Modern Applications

1.5.1 AI-as-a-Service

In the past, adding artificial intelligence to an application required experts, large datasets, and expensive computers.

Today, cloud providers offer AI-as-a-Service, which gives developers ready-to-use AI tools through simple APIs. Developers can send data such as images, audio, or text to the service and quickly receive useful results without building their own AI models.

Table 1.5 summarizes the four main types.

Table 1.5: Types of AI-as-a-Service and Real-World Examples

AI Service Type	What It Does	Real-World Example in India
Vision (Image Recognition)	Identifies objects, faces, text, and unsafe content in images/videos.	Google Photos lets you search 'Diwali 2023' and finds all photos from that occasion. Instagram auto-tags friends.
Speech	Converts speech to text (STT) and text to lifelike speech (TTS).	Google Assistant and Alexa understand Hindi voice commands. YouTube auto-generates captions.
Language (NLP)	Understands, translates, and	Google Translate between Hindi and English. Flipkart analyses

	analyses human language text.	product reviews to detect sentiment.
Recommendation	Analyses past behaviour to predict what a user might like next.	Netflix 'Continue Watching,' Amazon 'Customers also bought,' and Spotify 'Discover Weekly.'

Practical Activity 1.1. Mapping Cloud Services to a Real App (Google Classroom)

Objective

To identify and categorize the different cloud services used in Google Classroom by observing its features.

Materials Required

- Smartphone / Computer with internet access
- Google Classroom account (student login)
- Notebook and pen

Procedure

Step 1. Open Google Classroom on your device.

Step 2. Join or access any available class.

Step 3. Explore different features such as:

- Posting announcements
- Uploading assignments
- Viewing class materials
- Submitting homework

Step 4. Observe how data is stored, shared, and accessed.

Step 5. Identify the cloud services used in each feature:

- Storage (saving files)
- Computing (processing tasks)
- Communication (sharing messages)

Step 6. Create a table and map each feature to its corresponding cloud service type (IaaS, PaaS, SaaS).

Step 7. Discuss your observations with classmates or teacher.

Observation:

Students observe that Google Classroom uses cloud services like storage

for saving assignments, computing for processing data, and communication tools for interaction, mainly categorized under Software as a Service (SaaS).

Conclusion:
Real-world applications like Google Classroom depend on multiple cloud services (IaaS, PaaS, SaaS) to provide smooth, accessible, and collaborative learning experiences.

Summary

In this session, students learned how modern applications use cloud computing to deliver services over the internet. They understood cloud service models (IaaS, PaaS, SaaS) and compute services like Virtual Machines, Containers, and Serverless. Students also learned about databases, CDNs for fast content delivery, and AI services. Through an activity using Google Classroom, they identified how real-world apps use cloud services.

Check Your Progress

A. Multiple Choice Questions

1. A startup wants full control over OS and software while using cloud. Which model should they choose?
   (a) SaaS (b) PaaS (c) IaaS (d) FaaS
2. Which compute service is best for running small event-based tasks like sending OTPs?
   (a) Virtual Machine (b) Container (c) Serverless (d) Database
3. A student accesses Google Classroom without installing any software. This is an example of:
   (a) IaaS (b) PaaS (c) SaaS (d) CDN
4. Which service helps a video load faster by using the nearest server?
   (a) Database (b) CDN (c) Compute Engine (d) AI Service
5. Identifying faces in photos is an example of:
   (a) Language AI (b) Speech AI (c) Vision AI (d) Compute Service

B. Fill in the Blanks

1. Cloud computing provides services over the _____.
2. A virtual version of a computer is called a _____.
3. The _____ key uniquely identifies each record in a table.
4. CDN stores data on multiple _____ servers.
5. Pre-built intelligent tools provided by cloud are called _____ services.

C. True or False

1. Serverless computing means no servers exist at all. _____
2. Containers include application code and dependencies. _____
3. CDN mainly delivers static content like images and videos. _____
4. SaaS requires users to manage hardware and OS. _____
5. A Foreign Key links two tables in a database. _____

D. Short Answer Questions

1. Why do apps like Instagram load content quickly across countries?
2. Differentiate between Primary Key and Foreign Key.
3. Why is Serverless suitable for apps used occasionally?
4. What is the role of a database in an application?
5. Identify the cloud service used in Google Classroom.

Session 2. E-Commerce Systems and Secure Digital Payments

It was Meera's birthday, and she received ₹2,000 to spend from her parents. She opened Flipkart, searched for wireless earphones under ₹2,000, and found a pair costing ₹1,800 with ₹200 delivery charges. She placed the order and paid through UPI. Within minutes, she received a confirmation message for delivery.

Her cousin Kabir wondered how Flipkart knew the item was in stock, how UPI transferred money securely, and how the transaction stayed safe. The answer is that many cloud services work together in the background—checking inventory, creating the order, and processing the secure payment in just a few seconds.

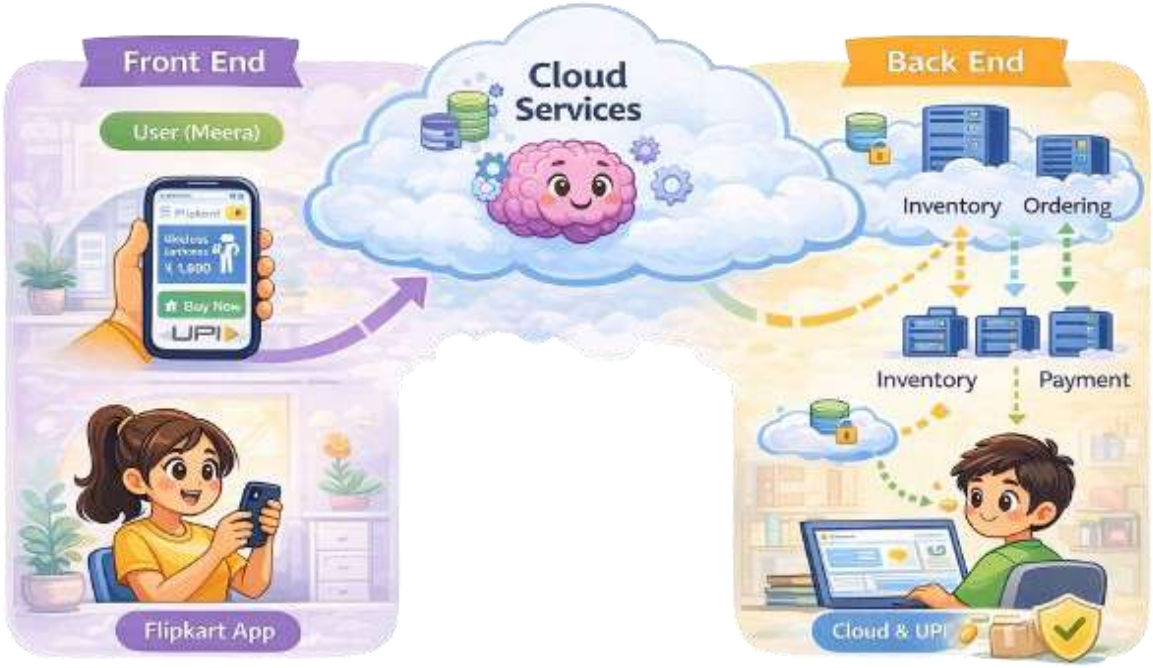


Fig. 2.1: The Two Sides of an E-Commerce Transaction

2.1 Introduction to E-Commerce Systems

An E-Commerce system is an online platform where people can buy or sell products and services through the internet. It includes a website or app for customers and a cloud-based system that manages products, orders, and payments.

It works like a digital marketplace. Customers browse items online, add them to a cart, make payments using options like UPI or cards, and the products are delivered to their homes.

Popular e-commerce platforms in India include Flipkart, Amazon India, Myntra, Big Basket, Zomato, Swiggy, and MakeMyTrip. These platforms use cloud technology to run their services.

2.2 Product Database

2.2.1 Database Stores

Every item listed on a platform like Flipkart lives as a row in a product database. Table 2.1 shows the kind of information stored for a single product.

Table 2.1: Information Stored for a Product in an E-Commerce Database

Information Type	**Description**	**Example (Wireless Earphones)**
ProductID (PK)	A unique identifier for the product.	P-5021
ProductName	The title visible on the product page.	Boat Rockerz 255 Pro Wireless Earphones
Description	Detailed features text.	Bluetooth 5.0, 40-hour battery, IPX5 water resistant…
Category	Department and sub-department.	Electronics > Audio > Wireless Earphones
Price	Current selling price.	₹1,799
StockQuantity	Units available in warehouse.	312
SellerID (FK)	Which seller offers this product.	S-441 (TechMart India)
ImageURL	Link to product photo in cloud storage.	https://cdn.flipkart.com/img/p5021.jpg
Rating	Average customer rating.	4.3 stars

2.2.2 Database Schema — Linked Tables

A real product database does not store all information in one giant table. Storing the full seller's name and category name in every product row would repeat the same data thousands of times. Instead, the product table stores only IDs (foreign keys) that point to separate Sellers and Categories tables. This design is called a database schema. Figure 2.2 shows this structure.

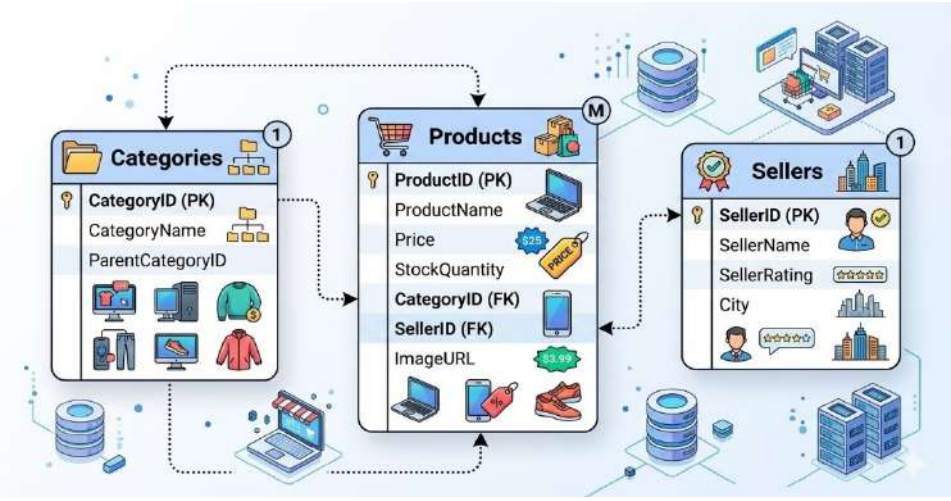


Fig. 2.2: Simplified E-Commerce Product Database Schema

Table 2.2 shows the Categories table with a hierarchical structure — each category can have a parent category, allowing nesting like Electronics > Audio > Wireless Earphones.

Table 2.2: Categories Table — Hierarchical Structure

CategoryID (PK)	CategoryName	ParentCategoryID
101	Electronics	NULL (Top Level)
102	Audio	101
103	Wireless Earphones	102
104	Mobile Phones	101

If a seller changes their name, it only needs to be updated in one place — the Sellers table — and the change reflects across all their products automatically. This is the power of relational design.

2.2.3 Searching and Filtering with SQL

When you type 'wireless earphones under ₹2000' in Flipkart's search bar and select '4 stars & above,' the system converts your inputs into a database query. Figure 2.3 illustrates this process.

Fig. 2.3: How Search and Filter Convert to a Database Query

The database scans millions of rows, applies the conditions, and returns matching results — usually within milliseconds. Every filter you apply (brand, price range, delivery time) adds one more condition to the query.

2.2.4 Shopping Cart

When you click Add to Cart on Flipkart, that item is not yet purchased. It sits in a temporary holding area — the shopping cart — until you decide to checkout. The cart works exactly like the wire basket you carry around a supermarket: it holds your selected items until you reach the billing counter.

2.2.5 Order Processing Workflow

When a customer clicks Checkout, the e-commerce system starts a step-by-step process in the cloud. Each step must complete successfully for the order to continue.

Step 1. Checkout: The system collects all items from the customer's cart.

Step 2. Create Order Record: A new entry is created in the Orders database with the status Pending.

Step 3. Check Inventory: The system verifies whether all items are available in stock. If any item is unavailable, the order stops. If available, the stock is temporarily reserved.

Step 4. Calculate Total: The system calculates the final amount by adding item prices, taxes, and delivery charges, and subtracting any discounts or coupons.

Step 5. Start Payment: The final amount and order ID are sent to a payment gateway to complete the payment.

2.2.6 Payment Workflow

Payment Gateway: a third-party cloud service that securely transmits payment details between the customer, the e-commerce platform, and the

customer's bank, authorizes the transaction, and returns a success or failure response.

Riya ordered a book online and paid using UPI. The payment gateway securely sent her payment request to the bank. The bank verified the transaction and approved it. Within seconds, the website displayed “Payment Successful,” and her order was confirmed.

Popular payment gateways in India include Razorpay, Paytm Payment Gateway, CCAvenue, and BillDesk. All major UPI apps (PhonePe, Google Pay, Paytm) also work through gateway infrastructure.

2.2.7 Payment Confirmation and Order Update

After a customer completes payment on the gateway page, the e-commerce platform receives confirmation in two ways. First, the browser returns to the website with a payment success or failure message. Second, the gateway sends a secure webhook message directly to the platform’s server to confirm the transaction.

If the payment is successful, the order status changes from Pending to Confirmed, and the payment ID is saved with the order. If the payment fails, the order status becomes Failed, and the reserved stock is released.

For Meera's order, a PaymentID from Razorpay is stored alongside the confirmed status — this links the order to the specific payment transaction for any future disputes or refunds. For Kabir's failed order, the status is marked Failed and PaymentID is NULL.

Finally, the platform sends Meera an on-screen confirmation, an email with order details and estimated delivery, and an SMS notification.

2.3 Payment Security Basics

2.3.1 HTTPS — Encrypting the Connection

HTTPS (Hypertext Transfer Protocol Secure): the secure version of HTTP that encrypts all data transmitted between your browser and the website, preventing unauthorized interception.

Imagine you are sending a letter. HTTP sends it in a transparent envelope — anyone who intercepts the letter can read it. HTTPS sends it inside a locked steel box — even if intercepted, the contents are unreadable. Figure 2.4 illustrates this difference.

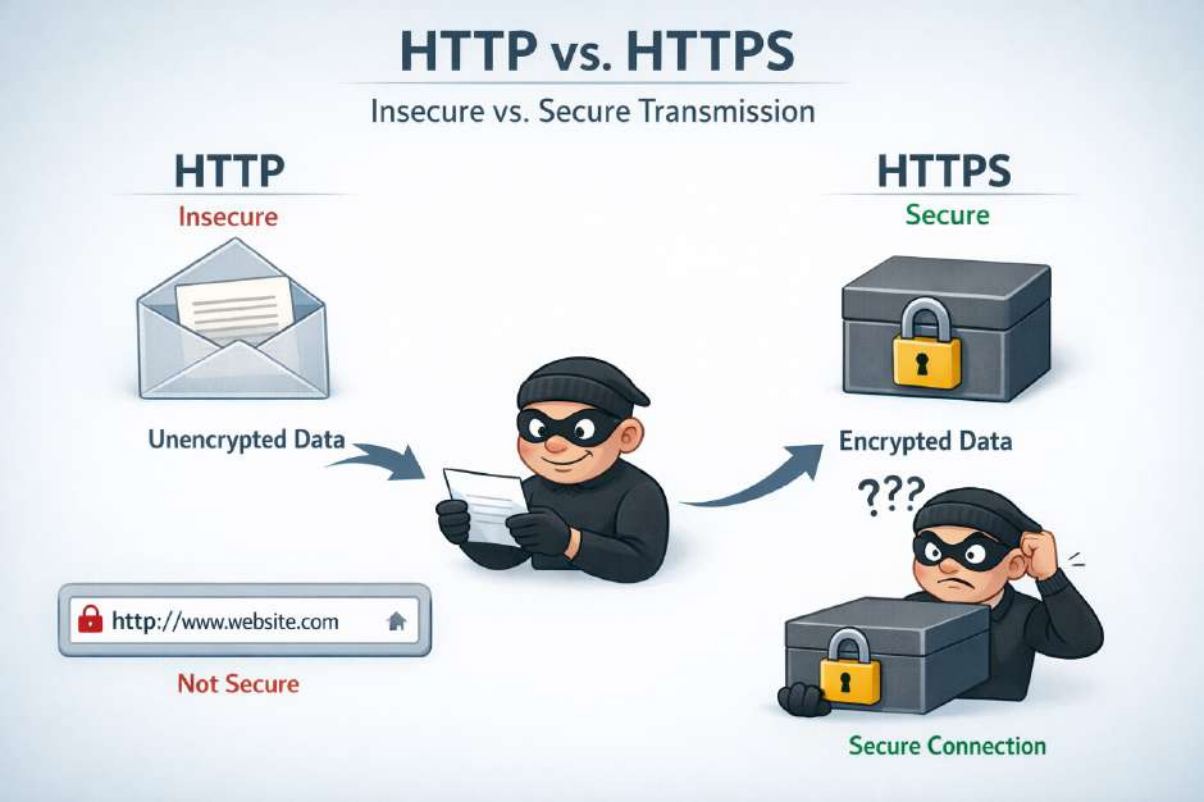


Fig. 2.4: HTTP vs. HTTPS — Insecure vs. Secure Transmission ***(Redraw)***

Always check for the padlock icon in your browser's address bar before entering any payment or personal information. Any payment page without HTTPS is unsafe.

2.3.2 Never Store Raw Payment Data

A trustworthy e-commerce platform never stores your full card number, CVV, or UPI PIN on its own servers. Why? Because no matter how strong their security is, a determined hacker might break in. If the database contains no actual card data, there is nothing of value to steal. This is why payment details go directly to the payment gateway — the e-commerce site only learns whether the payment succeeded or failed.

2.3.3 Tokenization

Tokenization is a security technique that replaces sensitive payment data (like a card number) with a non-sensitive substitute called a token, which is meaningless to anyone except the payment gateway that issued it.

When you tick 'Save card for future purchases' on Flipkart, Flipkart does not save your card number. Instead, Razorpay gives Flipkart a unique code like tok_X9Y2Z — the token. The next time you buy, Flipkart sends the token to Razorpay, which recognises it and charges your card. Even if a hacker steals Flipkart's database, they find only useless tokens. Figure 2.4 explains this with a two-step illustration.

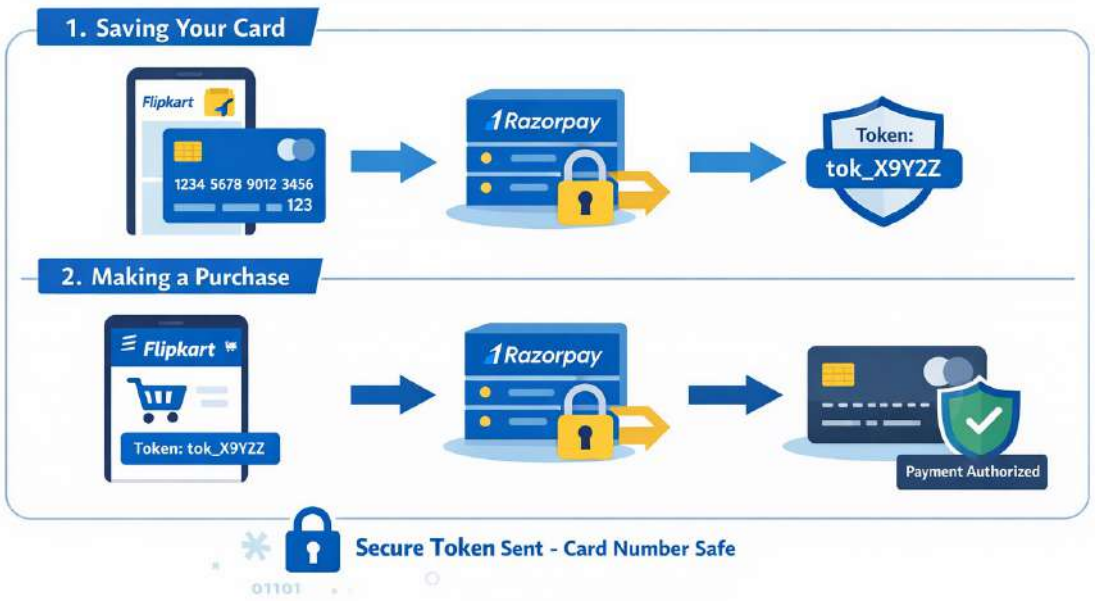


Fig. 2.4: How Tokenization Protects Your Card Details

Practical Activity 2.1. Mapping the Checkout Workflow of a Real E-Commerce App

Objective

To identify and map the step-by-step checkout workflow of a real e-commerce app and understand how different cloud services support each step.

Procedure

Step 1. Open any e-commerce app like Amazon or Flipkart.

Step 2. Search for a product (e.g., book, headphones, or shoes).

Step 3. Add the product to the cart.

Step 4. Proceed to the checkout page.

Step 5. Observe each step in the checkout process:

- Viewing cart items
- Entering delivery address
- Selecting payment method
- Placing the order

Step 6. Note how the app stores, processes, and displays information at each step.

Step 7. Create a flowchart showing the checkout workflow.

Step 8. Identify the cloud services used in each step:

- Database (storing user and order details)
- Compute (processing order and payment)

- CDN (loading product images quickly)
- Discuss your findings with classmates or teacher.

Observation:
Students observe that the checkout process involves multiple steps where cloud services like databases store user data, compute services process orders, and CDNs ensure fast loading of images and pages.

Conclusion:
E-commerce apps like Amazon and Flipkart use integrated cloud services to provide a smooth, fast, and secure checkout experience.

Summary

In this session, students learned how an e-commerce system works by connecting a user interface with cloud-based services. They understood how product data is stored in databases using tables, primary keys, and foreign keys. Students explored the complete checkout workflow, including adding items to the cart, order creation, inventory checking, and payment processing through secure payment gateways. They also learned about payment security concepts like HTTPS and tokenization.

Check Your Progress

A. Multiple Choice Questions

1. When a customer adds a product to cart but does not buy it, where is it stored?
   (a) Payment Gateway (b) Shopping Cart (c) CDN (d) AI Service
2. Which step happens first during checkout?
   (a) Payment (b) Inventory Check (c) Order Creation (d) Checkout
3. Which service securely processes online payments?
   (a) CDN (b) Database (c) Payment Gateway (d) Compute Engine
4. Why is HTTPS important in e-commerce?
   (a) Stores data (b) Encrypts communication (c) Loads images (d) Manages cart
5. Saving card details as a code instead of real number is called:
   (a) Encryption (b) Tokenization (c) Compression (d) Indexing

B. Fill in the Blanks

1. The unique identifier of a product is called _____.
2. The shopping cart holds items _____ before purchase.
3. A _____ verifies and processes online payments.
4. _____ ensures secure data transfer between user and website.
5. _____ replaces sensitive card details with a safe code.

C. True or False

1. Payment gateways store full card details permanently. _____
2. HTTPS protects data from being intercepted. _____
3. Inventory is checked after payment is completed. _____
4. Foreign Key connects two related tables. _____
5. Failed payments release reserved stock. _____

D. Short Answer Questions

1. What is the role of a shopping cart in e-commerce?
2. Why is inventory check important before payment?
3. What is a payment gateway?
4. Why should users avoid HTTP websites for payments?
5. How does tokenization improve payment security?

Session 3. Introduction to IoT and Edge Computing

One rainy morning near Nashik, a sensor fixed on a dam detected that the water level had reached 98% capacity. Within seconds, a siren sounded at the dam, officials in Pune received an alert on their monitoring dashboard, and nearby farmers got warning SMS messages — all automatically.

At school, Harsh told his teacher about this. The teacher explained that this system uses Internet of Things (IoT) and edge computing together. The sensor (IoT device) collects real-world data, while a local edge processor quickly makes decisions, such as triggering the siren. Instead of sending the data to a distant cloud server first, the decision happens near the source, allowing the response to occur in milliseconds. Figure 3.5.1 illustrates this process: the sensor detects the water level, a gateway collects the data, the edge processor triggers the siren locally, and the cloud records the event and sends alerts to authorities and farmers.

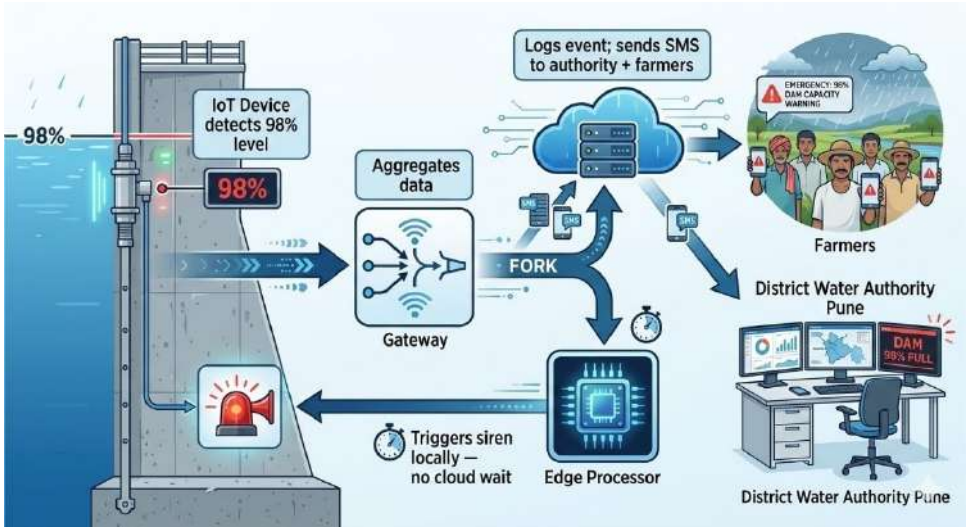


Fig. 3.1: IoT Alert at the Dam — Sensor to Siren in Three Seconds

3.1 IoT Architecture — Four Building Blocks

IoT (the Internet of Things) is a system that connects physical objects to digital networks so they can collect environmental data, transmit it automatically, and trigger actions — without any human involvement at the point of collection. Every IoT system, regardless of its application, is built from four components: sensors and devices, gateways, a cloud backend, and end applications. These four layers always work in the same sequence.

IoT (Internet of Things): A network of physical objects embedded with sensors and network connectivity that detect environmental data and transmit it automatically — without human intervention — to a digital processing system.

3.1.1 Sensors and Devices (Data Collection Layer)

A sensor measures something from the real world such as temperature, light, motion, pressure, or moisture and converts it into digital data.

An IoT device includes the sensor along with small electronics that process and send this data.

Examples

- A temperature sensor inside a food delivery bag checks if food is still warm.
- A soil moisture sensor in a farm check how wet the soil is.
- A GPS device in a train continuously sends its location.

These devices collect data automatically and send it forward without human involvement.

3.1.2 Gateway (Data Collection and Forwarding Layer)

A gateway collects data from many nearby sensors and sends it to the next system. Different sensors may use different communication methods (Bluetooth, Zigbee, radio, etc.). The gateway converts all data into a common format and forwards it to the cloud or edge processor.

Example: Like a post office sorting center that collects letters from many places, organizes them, and sends them to the correct destination.

3.1.3 Cloud Backend (Storage and Analysis Layer)

The cloud backend stores all the data coming from IoT devices and analyzes it.

It can:

- Save data for a long time
- Detect patterns
- Generate dashboards
- Send alerts when something unusual happens

Example: A factory may have hundreds of temperature sensors. The cloud compares today's readings with past data and can detect if a machine is slowly overheating before it fails.

3.1.4 End Applications (User Layer)

End applications are the apps or dashboards where people see IoT data and take action. They may be: Mobile apps, Web dashboards, SMS alerts, Automated control systems.

Examples

- A farmer checks soil moisture in a mobile app.
- A school monitors electricity usage through a dashboard.
- Farmers receive SMS alerts when a dam water level rises.

3.2. Edge vs. Cloud

Edge computing and cloud computing are not alternatives — they are complementary layers designed to handle different parts of the same problem. Edge handles what must happen immediately and locally. Cloud handles what benefits from long-term storage, large-scale analysis, and central oversight. Table 3.5.1 compares the two across the decision criteria that matter when building an IoT system.

Table 3.1: Edge Computing vs. Cloud Computing — Design Decision Criteria

Criteria	**Edge Computing**	**Cloud Computing**
Response time needed	Milliseconds — no network delay; decision made on-site.	Seconds are acceptable — trend reports, batch analysis.
Works without internet?	Yes — local processing continues if network drops.	No — requires stable connectivity to function.
Data handled	Processes locally; sends only summaries or important alerts upstream.	Stores all incoming data long-term; analyses patterns across months.
Processing capacity	Limited — built for specific, fast, repetitive tasks.	Effectively unlimited — handles complex AI models and large data sets.
Best suited for	Safety alarms, real-time physical control, offline-capable systems.	Trend dashboards, machine-learning training, central reporting.

3.3 IoT in India

India uses IoT in sectors like cities, farming, and transportation. These systems work through four layers: devices, gateways, cloud services, and applications. Under the Smart Cities Mission, cities like Pune, Surat, and Bhopal use IoT. For example, smart streetlights adjust brightness automatically, and flood sensors give early warnings.

In agriculture, companies like BSNL and Reliance Jio provide IoT services. Soil sensors measure moisture and help farmers decide when to irrigate.

In transportation, NTES shows live train locations. GPS devices send data to cloud servers, which display it on passengers' phones.

As you can see in Figure 3.2

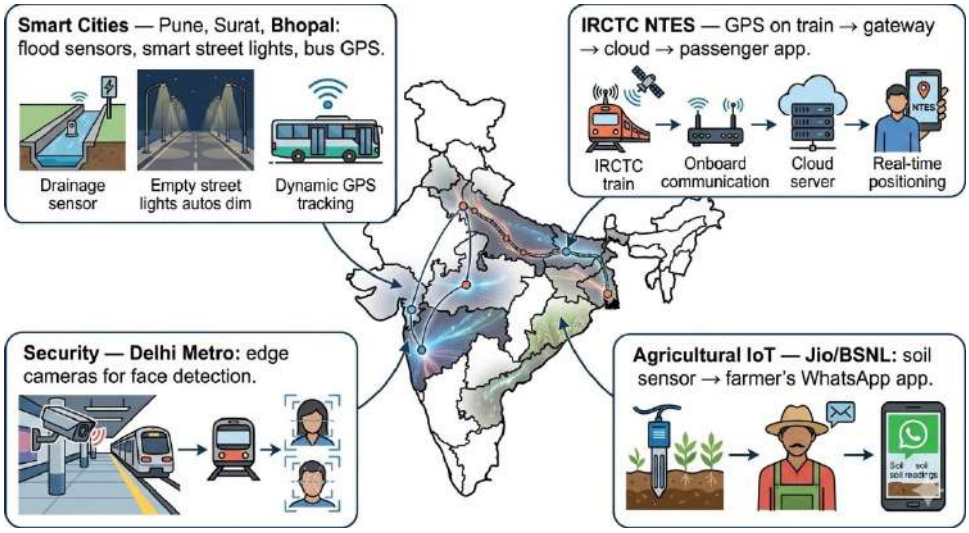


Fig. 3.2: IoT in India — Four Sectors, Same Architecture

Practical Activity 3.1. Building a Smart Classroom IoT Architecture

Objective

To design and understand a Smart Classroom system using IoT components and identify how data is collected, processed, and used.

Materials Required

- Notebook and pen
- Chart paper / A4 sheets
- Internet-enabled device (optional)
- Pencil and colors for diagram

Procedure

Step 1. Discuss the concept of a Smart Classroom (automatic lights, smart attendance, temperature control, etc.).

Step 2. Identify common devices used in a classroom:

- Lights and fans
- Projector / smart board
- Attendance system
- Temperature sensors

Step 3. List IoT components involved:

- Sensors (temperature, motion, light)
- Devices/Actuators (fans, lights)
- Internet/Wi-Fi connection
- Cloud platform (data storage and processing)

Step 4. Draw a diagram of Smart Classroom IoT Architecture showing:

- Sensors collecting data
- Data sent via internet
- Cloud processing
- Output actions (automatic control)

Step 5. Show the flow of data using arrows (Input → Processing → Output).

Step 6. Label each component clearly (sensor, gateway, cloud, user interface).

Step 7. Explain how the system works step-by-step in your notebook.

Step 8. Discuss how IoT improves efficiency, energy saving, and learning experience.

Observation

Students observe that sensors collect real-time data, which is sent to the cloud for processing, and appropriate actions (like switching lights or displaying data) are performed automatically.

Conclusion

A Smart Classroom uses IoT architecture where connected devices, sensors, and cloud services work together to automate tasks, improve efficiency, and enhance the learning environment.

Summary

In this session, students learned about the Internet of Things (IoT) and how it connects physical devices to digital systems. They understood the four main components of IoT architecture—sensors, gateways, cloud backend, and end applications. Students also learned the difference between edge computing and cloud computing, where edge provides fast local decisions and cloud handles storage and analysis. Through a practical activity, they designed a Smart Classroom IoT system and understood how data is collected, processed, and used to automate tasks and improve efficiency.

Check Your Progress

A. Multiple Choice Questions

1. Which IoT component collects data from the real world?
   (a) Gateway (b) Cloud (c) Sensor (d) Application
2. A system that must respond instantly (like a fire alarm) should use:
   (a) Cloud Computing (b) Edge Computing (c) Database (d) CDN
3. What is the main role of a gateway in IoT?
   (a) Store data (b) Display data (c) Convert and forward data (d) Run apps

4. Which layer stores and analyzes large amounts of IoT data?
   (a) Sensor Layer (b) Gateway Layer (c) Cloud Backend (d) Device Layer
5. Which of the following is an example of an IoT application?
   (a) Calculator (b) Smart irrigation system (c) Text editor (d) Paint

B. Fill in the Blanks

1. _____ collect real-world data like temperature and motion.
2. A _____ connects sensors and sends data forward.
3. _____ computing processes data near the device.
4. The _____ stores and analyses IoT data.
5. _____ show the results to users.

C. True or False

1. IoT devices require human input to collect data. _____
2. Edge computing can work even without internet connectivity. _____
3. Cloud computing is best for instant decision-making. _____
4. Gateways can connect devices using different communication methods. _____
5. IoT systems have only two layers. _____

D. Short Answer Questions

1. What is IoT in simple terms?
2. Why is edge computing important in real-time systems?
3. What is the function of a gateway in IoT?
4. Name the four layers of IoT architecture.
5. How does IoT improve efficiency in a smart classroom?

Session 4. Real-Time Location Services and Navigation Systems

Nandini ordered lunch on Swiggy from her office in Chennai. Soon after placing the order, she saw a map on the app showing the delivery partner's moving location. The pin moved along Anna Salai, stopped briefly at a traffic signal, and reached her building almost exactly at the predicted time of 18 minutes.

What looked like a simple moving pin was actually several technologies working together. GPS satellites provided the delivery partner's location, the phone sent these coordinates to a cloud server, a Maps API displayed the location on a digital map, and a routing system calculated the estimated delivery time using real-time traffic data. Figure 3.6.1 shows what was happening behind her screen.

Fig. 4.1: Swiggy Tracking — Four Systems Behind One Moving Pin

4.1 Global Positioning System (GPS)

GPS (Global Positioning System) is a satellite navigation system that helps a device find its exact location on Earth. It gives coordinates called latitude (north–south position) and longitude (east–west position).

GPS works using signals from a network of satellites orbiting the Earth. A GPS receiver in a phone or device calculates its position by measuring the distance from several satellites. This method is called trilateration. GPS itself does not require internet; internet is only needed when the location is sent to an app or server.

4.2 Maps APIs — From Coordinates to Maps

A Maps API is an online service that converts location data (coordinates or addresses) into visual maps and navigation information. It can show roads, buildings, routes, travel time, and nearby places.

Instead of creating their own map database, apps use map services such as Google Maps Platform, Microsoft Azure Maps, MapMyIndia, and OLA Maps to display maps and provide navigation features. Figure 4.2 shows how an app, a Maps API, and the user's screen interact.

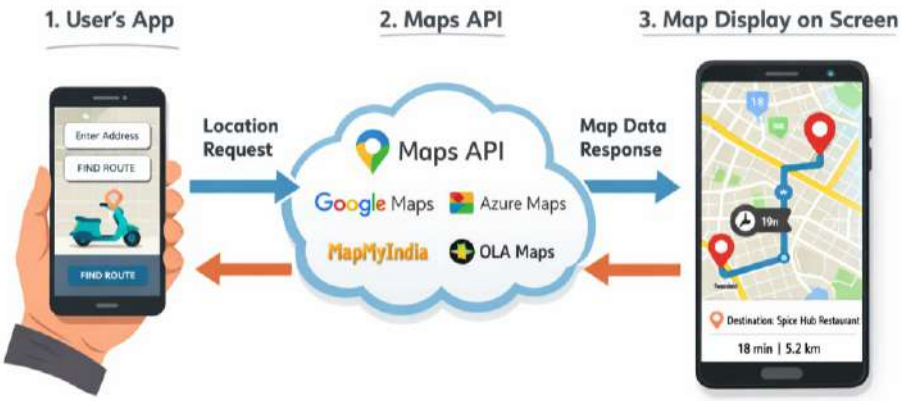


Fig. 4.2: Maps API — The Request-Response Cycle Between App and Map Data

4.3 Real-Time Location Tracking

Real-time tracking services such as delivery tracking, ride navigation, or train tracking follow a similar system architecture.

GPS Device: The phone's GPS chip receives satellite signals and calculates the device's latitude and longitude. This step does not require internet.

Location Upload: The app sends the location to a cloud server every few seconds using the internet. This small data update is called a location ping.

Cloud Storage: The cloud stores the latest location of the device so it can be quickly accessed when needed.

Maps Service: A Maps API such as Google Maps Platform, MapmyIndia, or OLA Maps places the coordinates on a digital map and calculates the route or ETA.

User Display: The app shows the map on the user's screen with a moving pin that updates every few seconds.

The map background (roads and buildings) is usually a stored image. Only the location pin moves as new coordinates are received.

Practical Activity 4.1. Tracing the Architecture of a Real-Time Location Service

Objective

To identify and trace how a real-time location service works by analyzing its architecture and data flow.

Materials Required

- Smartphone with internet and GPS enabled
- Any location-based app such as Google Maps
- Notebook and pen

Procedure

Step 1. Open Google Maps on your device.

Step 2. Turn on location (GPS) services.

Step 3. Search for a nearby place (e.g., school, hospital, or market).

Step 4. Start navigation to the selected location.

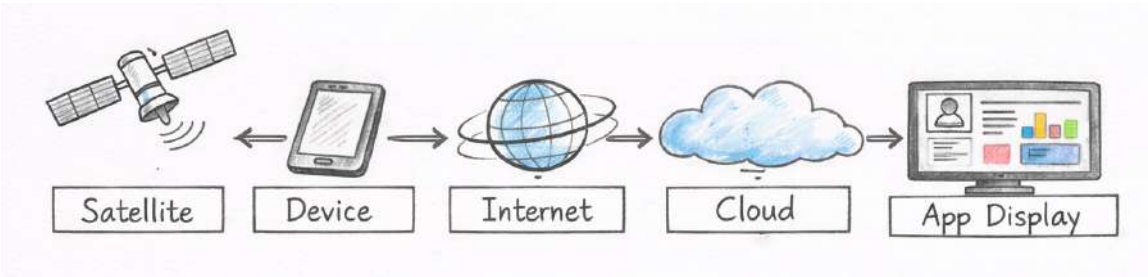
Step 5. Observe how your position updates in real-time on the map.

Step 6. Identify the components involved in the system:

1. GPS satellites (collect location data)
2. Mobile device (receives signals)
3. Internet connection (sends/receives data)
4. Cloud servers (process routes and traffic data)

Step 7. Draw a diagram showing the architecture:

Satellite → Device → Internet → Cloud → App display



Step 8. Trace the flow of data step-by-step in your notebook.

Step 9. Discuss how real-time updates (traffic, route changes) are provided.

Observation

Students observe that real-time location services continuously collect data from GPS, send it to cloud servers for processing, and display updated results instantly on the app.

Conclusion

Real-time location services use a combination of GPS, internet, and cloud computing to provide accurate and continuously updated navigation information.

Summary

In this session, students learned how location-based services work using GPS, maps, and cloud technologies. They understood how GPS satellites provide location coordinates without internet and how these coordinates are sent to cloud servers. Students explored the role of Maps APIs in converting location data into visual maps and routes. They also learned how real-time tracking works through continuous location updates (pings).

Check Your Progress

A. Multiple Choice Questions

1. GPS determines location using signals from:
   (a) Wi-Fi routers (b) Satellites (c) Cloud servers (d) Mobile towers
2. Which component converts coordinates into a visual map?
   (a) GPS chip (b) Cloud storage (c) Maps API (d) Sensor
3. What is sent to the cloud repeatedly for real-time tracking?
   (a) Images (b) Location pings (c) Videos (d) Text messages
4. Which technology calculates routes and estimated time?
   (a) GPS (b) Maps API (c) Database (d) Gateway
5. Which of the following does NOT require internet?
   (a) GPS location calculation (b) Map display
   (c) Cloud processing (d) Real-time updates

B. Fill in the Blanks

1. GPS provides _____ and _____ coordinates.
2. The method used by GPS to find location is called _____.
3. A _____ converts coordinates into map visuals.
4. Small updates sent to the server are called location _____.
5. The _____ stores and processes location data.

C. True or False

1. GPS requires internet to calculate location. _____
2. Maps APIs help apps display maps and routes. _____
3. Location pings are sent only once during tracking. _____
4. Cloud servers store and process location data. _____
5. Map tiles show the background of maps. _____

D. Short Answer Questions

1. What is GPS and how does it work?
2. Why is internet needed after GPS calculates location?
3. What is the role of a Maps API?
4. Explain the term “location ping.”
5. How does real-time tracking work in apps?

Module 4. Cloud Deployment & Operations

This module on Cloud Deployment and Operations introduces the processes involved in deploying, managing, and maintaining applications in cloud environments. In Session 1, you will learn the application deployment workflow, including stages such as develop, build, test, deploy, and monitor, along with concepts like CI/CD and Infrastructure as Code for efficient and automated deployment. In Session 2, you will understand website hosting by differentiating between static and dynamic websites, exploring object storage, custom domains, DNS, and Content Delivery Networks for improved website performance. In Session 3, you will learn the importance of data protection through backup and restore planning, including concepts like RPO, RTO, types of backups, and the 3-2-1 backup rule to ensure data safety. In Session 4, you are introduced to cloud project development and collaboration, where they learn about project planning, Software Development Life Cycle (SDLC), project charter, requirements, and wireframing. Overall, the module enables students to understand how cloud-based applications are deployed, hosted, secured, and managed effectively in real-world scenarios.

Session 1. Application Deployment Workflow

Seema had been working on her school project for three weeks. Her weather-tracking web app was finally ready. She called her cousin Rohit, who works at a software company, and said, "I built it. Now what?" Rohit laughed. "Building it was step one. Now comes the part that actually matters — getting it out into the world." He opened his laptop and showed Seema a diagram of how applications travel from a developer's machine to a live server. That diagram looked something like Figure 1.1.

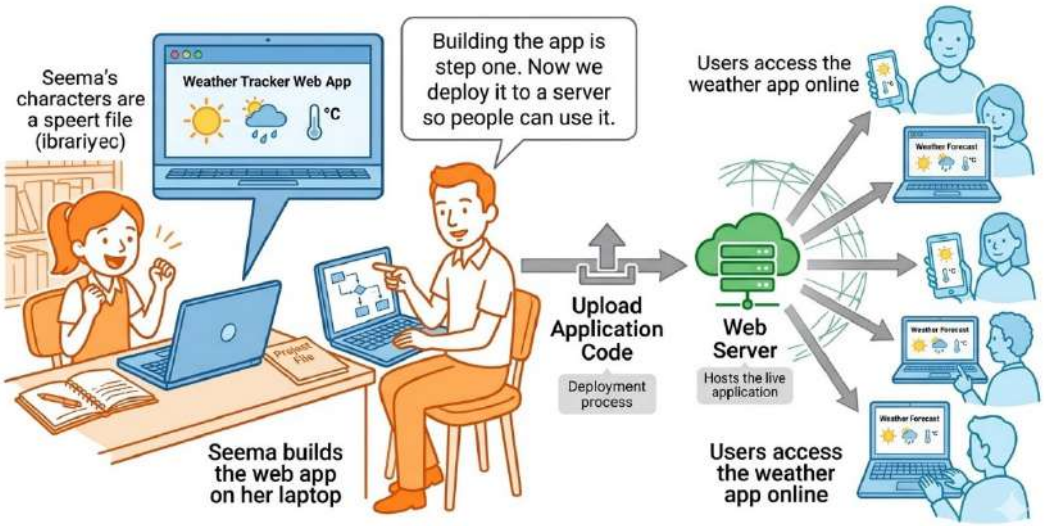


Fig. 1.1: The Five-Stage Cloud Deployment Pipeline

1.1 The Five-Stage Deployment Workflow

Seema continues improving her weather-tracking web app after Rohit shows her how cloud deployment works. One afternoon she asks, “If many people are working on the same app, how do they manage everything without making mistakes?” Rohit smiles and explains that real software companies follow a clear process. It works like a farming routine—each step happens in the correct order so the final crop grows well. In technology, this organized process is called a CI/CD pipeline, which helps teams automatically build, test, and release updates to an application.

Let’s see how the five stages work.

A. Develop: Developers write and update code and save it in a shared repository (like GitHub).

B. Build: Code is prepared so it can run properly (like packing everything needed).

C. Test: The app is checked to ensure it works correctly and has no errors.

D. Deploy: The app is released on the internet for users.

E. Monitor: The app is observed to fix issues and improve performance.

1.2 CI/CD: The Automation Engine

Seema continues improving her weather-tracking web app after Rohit explains how cloud deployment works. Rohit tells her that in real software companies many developers work on the same project at the same time. Without a proper system, their changes could conflict or break the application.

To solve this, teams use CI/CD pipelines—automated systems that build, test, and release updates whenever developers improve the program.

A. Continuous Integration (CI)

Continuous Integration (CI) means developers upload small code changes regularly to a shared repository such as GitHub or GitLab.

Each time new code is uploaded, the system automatically:

- builds the application,
- runs tests, and
- checks for errors.

This helps detect problems early. For example, when Seema adds a humidity feature to her weather app, automated tests confirm that the website still works correctly.

B. Continuous Delivery / Continuous Deployment (CD)

After the code passes all tests, the next stage is Continuous Delivery or Continuous Deployment (CD).

- **Continuous Delivery:** the system prepares the update, but a person clicks the final publish button.
- **Continuous Deployment:** the update is released automatically once tests pass.

This automation helps teams release updates quickly and safely.

C. The CI/CD Feedback Loop

As shown in Figure 1.2, CI and CD together create a continuous cycle:

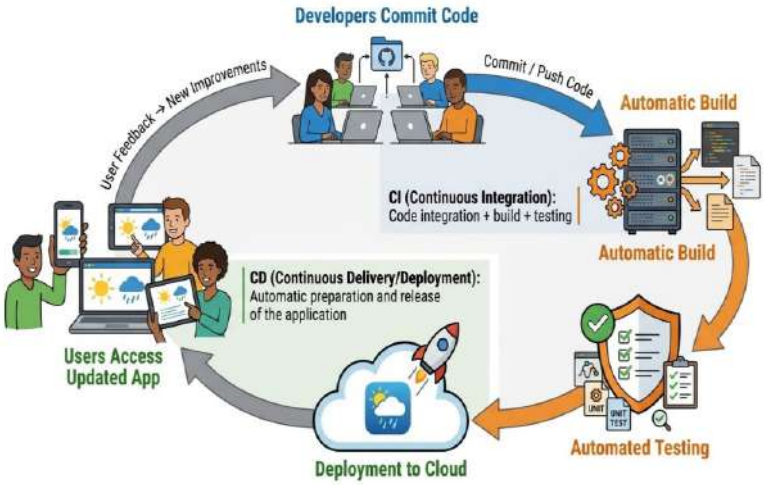


Fig. 1.2: The CI/CD Feedback Loop

Developers observe how the app performs and then improve it again. This repeating loop keeps the software stable and regularly updated.

Even small projects benefit from automation. For example, services like Indian Railway Catering and Tourism Corporation (IRCTC) release backend updates frequently so millions of passengers can book tickets smoothly without service interruptions.

1.3 Infrastructure as Code (IaC)

Earlier, setting up servers was slow and done manually. Infrastructure as Code (IaC) allows us to set up servers using a simple text file.

The file gives instructions like:

- Create virtual machines
- Install software
- Set network settings

All tasks are done automatically in minutes.

Table 1.1: Manual Server Setup vs. Infrastructure as Code

What Changes	Manual Setup	With IaC
Time to build environment	Several hours to days, depending on team	Minutes — run the script
Consistency across servers	Varies. Each person may configure slightly differently	Every environment is identical — same file, same result
Tracking changes	Difficult. Notes or memory required	Stored in Git; every change logged with author and reason

Recovering from failure	Rebuild manually — same slow process	Re-run the script; fresh environment in minutes

Version control is the quiet benefit most standard textbooks skip over entirely. Because your infrastructure file lives in the same repository as your code, you can see who changed what, when, and why. If a new server configuration breaks something, rolling back takes seconds.

Common IaC tools include Terraform and AWS CloudFormation. In India, the Government's GI Cloud initiative uses automated provisioning scripts so that new departments can go live without waiting weeks for manual server configuration.

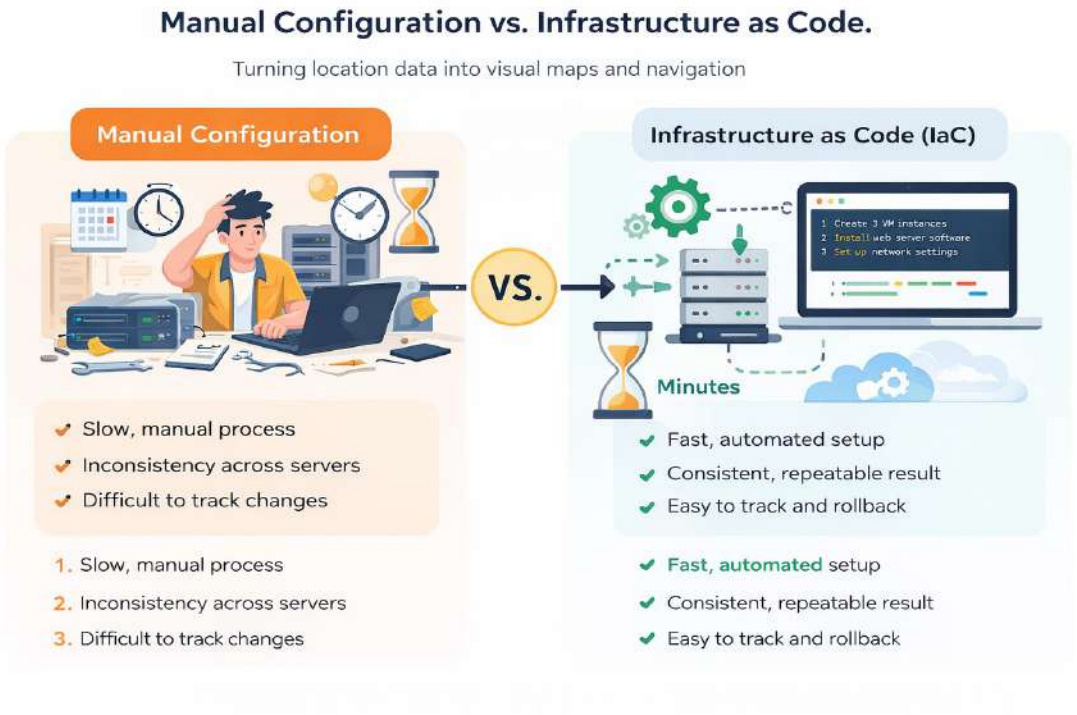


Fig. 1.3: Manual Configuration vs. Infrastructure as Code

1.4 Deployment Strategies

The deploy stage has more than one approach. Choosing the right one avoids downtime for users during an update.

A. Blue-Green Deployment

Run two identical environments at all times. One is live, called Blue; the other runs the new version, called Green. Once Green is tested and ready, traffic switches instantly. If something goes wrong, you switch back to Blue. Users experience no downtime.

B. Rolling Deployment

Update servers one at a time. At any moment, some servers run the old version and some run the new one. The update rolls gradually across all servers. This is simpler than blue-green but means two versions are briefly live at once.

C. Canary Release

Send the new version to a small group of users first, say five percent of traffic. Watch for errors. If everything looks fine, gradually increase that percentage. If something breaks, only a small fraction of users was affected. The name comes from the old mining practice of sending a canary into a tunnel before people entered. Table 4.1.2 summarizes these strategies.

Table 1.2: Deployment Strategy Comparison

Strategy	How It Works	Best When
Blue-Green	Two full environments; instant traffic switch	You need zero downtime and can afford double the resources
Rolling	Servers update one at a time; gradual changeover	Resources are limited and brief overlap is acceptable
Canary	New version reaches a small user group first	You want real-user testing before a full rollout

Practical Activity 1.1. Mapping a Five-Stage Deployment Workflow Using a Free Online Tool

Objective

To design and map a complete five-stage deployment workflow using a free online diagramming tool.

Materials Required

- Computer / Smartphone with internet access
- Free online tool such as Canva
- Notebook and pen

Procedure

Step 1. Open a free diagram tool like Canva.

Step 2. Create a new blank diagram or flowchart.

Step 3. Identify the five stages of deployment workflow:

- Development
- Testing
- Build

- Deployment
- Monitoring

Step 4. Add five boxes in sequence to represent each stage.

Step 5. Connect the boxes using arrows to show the flow of process.

Step 6. Add short descriptions inside each box:

- Development: Writing code
- Testing: Checking errors
- Build: Preparing application
- Deployment: Releasing to users
- Monitoring: Tracking performance

Step 7. Use colors, icons, or shapes to make the diagram clear and attractive.

Step 8. Save or download your workflow diagram.

Step 9. Present and explain your workflow to the class.

Observation

Students observe that application deployment follows a structured step-by-step workflow where each stage is important for successful software delivery.

Conclusion

A five-stage deployment workflow helps in organizing the process of developing, testing, and delivering applications efficiently using cloud-based tools.

Summary

In this session, students learned about the application deployment workflow and how software is released to users. They understood the five main stages—Develop, Build, Test, Deploy, and Monitor—and their importance in delivering reliable applications. Students also learned about CI/CD pipelines that automate building, testing, and deployment processes. They explored the concept of Infrastructure as Code (IaC) for faster and consistent server setup, and different deployment strategies like Blue-Green, Rolling, and Canary.

Check Your Progress

A. Multiple Choice Questions (MCQs)

5. Which stage involves writing and updating code?
   (a) Build (b) Develop (c) Test (d) Deploy
6. Which stage ensures the application works correctly before release?
   (a) Monitor (b) Build (c) Test (d) Deploy

7. What does CI/CD help in?
   (a) Designing UI (b) Automating build and deployment
   (c) Storing data (d) Creating hardware
8. Which deployment strategy releases updates to a small group first?
   (a) Rolling (b) Blue-Green (c) Canary (d) Manual
9. Infrastructure as Code (IaC) helps to:
   (a) Write application code (b) Automate server setup
   (c) Design websites (d) Manage users

B. Fill in the Blanks

1. The process of releasing an app to users is called _____.
2. CI stands for Continuous _____.
3. CD stands for Continuous _____.
4. The _____ stage tracks performance after deployment.
5. IaC uses _____ files to configure infrastructure.

C. True or False

1. CI/CD pipelines reduce manual work in deployment. _____
2. Testing is done after deployment. _____
3. Blue-Green deployment uses two environments. _____
4. Monitoring is not required after deployment. _____
5. IaC improves consistency in server setup. _____

D. Short Answer Questions

1. What is the purpose of the deployment stage?
2. Why is testing important in the workflow?
3. What is CI/CD in simple terms?
4. Name any one deployment strategy and explain it briefly.
5. How does IaC help in managing infrastructure?

Session 2. Introduction to Website Hosting and Cloud Deployment

Nandini wanted to put her resume online. Her classmate Kabir wanted to build an online store where customers could log in, browse products, and place orders. Both needed a website. Both needed hosting. But here is the thing: the type of hosting that works for Nandini would completely fail for Kabir. Mr. Deepak, their school's computer teacher, drew two boxes on the board and explained why their needs were fundamentally different. That drawing looked something like Figure 2.1.

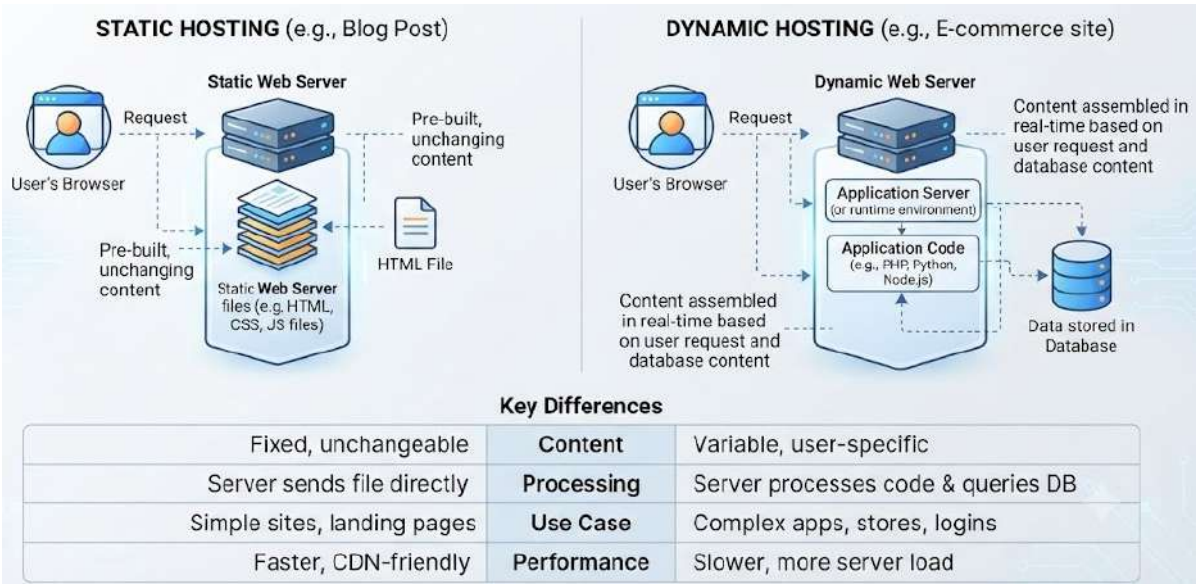


Fig. 2.1: How Static and Dynamic Hosting Differ

2.1 Static and Dynamic Hosting

A static website shows the same content to all users. The server simply sends fixed files like HTML, CSS, and images.

A dynamic website creates content each time it is opened. It can show different information based on the user, data, or situation.

Table 2.1 compares these two approaches across the factors that matter most when choosing a hosting solution.

Table 2.1: Static Hosting vs. Dynamic Hosting

Factor	Static Hosting	**Dynamic Hosting**
What the server does	Sends a pre-saved file as-is	Runs code, queries a database, builds the page on the spot
Speed	Very fast — no processing needed	Slower due to server-side computation and database calls

Cost	Very low; often free for small sites	Noticeably higher; requires more powerful servers
Examples	Portfolio sites, company brochures, documentation	Online banking, e-commerce, social media platforms
Where things go wrong	Content cannot change without a re-upload	Database failures or slow queries can make the entire page fail

Think of it this way: a printed notice board is static. Everyone reads the same thing. A WhatsApp group feed is dynamic. Each person sees messages addressed to them, in their language, with their preferences applied.

2.2 Object Storage for Static Site Hosting

A static website can be stored on cloud services called Object Storage.

Files are saved with unique links and can be accessed directly on the internet.

This is similar to storing photos or documents on Google Drive or Google Photos.

A. Hosting a Static Website Using Object Storage

Steps to Host a Static Website:

- Create storage on a cloud platform
- Upload website files (HTML, CSS, JavaScript)
- Enable website hosting option
- Use the given link to access the website

The overall process is illustrated in Fig. 4.2.2, which shows the four basic steps required to publish a static website using object storage.

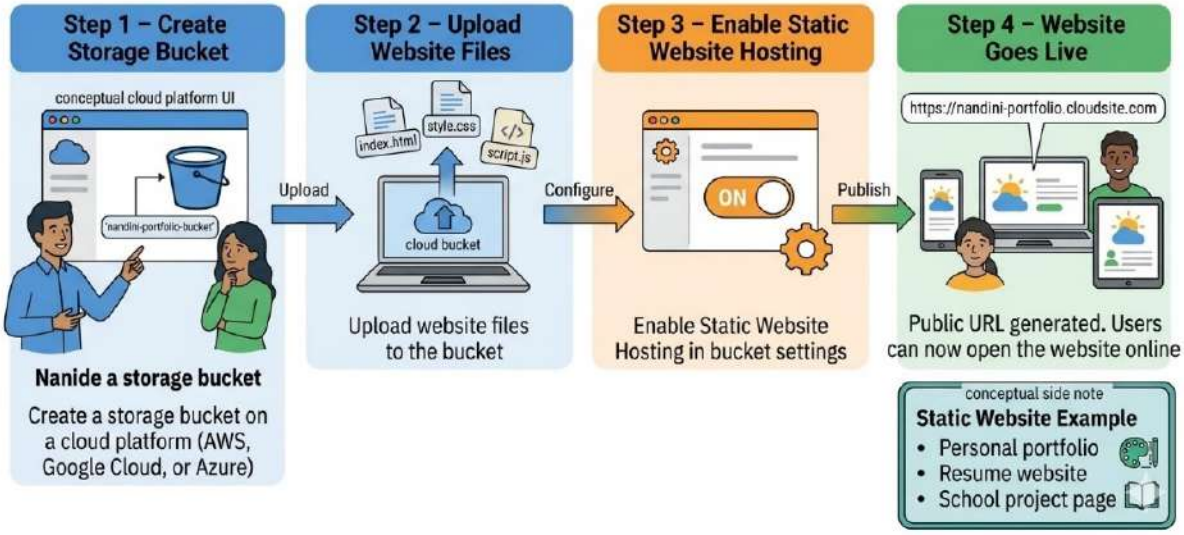


Fig. 2.2: Four Steps to Host a Static Site on Object Storage

B. Reliability and Cloud Advantage

One major benefit of object storage is automatic data replication. Cloud systems copy the stored files to multiple data centres. If one location faces a technical issue, another copy is used automatically.

This built-in reliability allows even small websites to remain available to users without the need for expensive servers or complex maintenance. For students and beginners, object storage therefore provides a simple and dependable way to host static websites on the cloud.

2.3 Custom Domains and DNS Records

A. Custom Domain in Professional Websites

When a website is hosted on cloud storage, the platform usually provides a long technical web address. Such addresses are difficult to remember and not suitable for professional sharing. A custom domain provides a short and meaningful name, such as www.nandinis-portfolio.in.

For vocational projects like student portfolios, school portals, or small business websites, a custom domain improves professionalism and makes the website easier for visitors to access.

B. Domain Name System (DNS) in Website Access

The Domain Name System (DNS) helps connect website names with the correct internet servers. It converts a human-readable domain name into a numerical IP address so that computers can locate the website.

Because of DNS, users can access websites using simple names instead of long number-based addresses.

C. Linking a Custom Domain to Cloud Hosting

To show a website using a domain name, it must be linked using DNS settings. A CNAME record connects the website name (like www) to its cloud location. After setup, users visiting the domain are automatically taken to the website.

Example: Different DNS records help manage website access and other services. Table 2.2 shows a simple example of DNS records used for a static website.

Table 2.2: Example DNS Records for a Website

Record Type	Purpose	Example
A Record	Connects the main domain to a server IP address	example.com → 192.0.2.1
CNAME Record	Redirects a subdomain to another domain	www.example.com → example.com

MX Record	Specifies the mail server for domain email	example.com → mail.example.com
TXT Record	Stores verification or security information	Domain verification string

A. Custom Domain

When a website is hosted on cloud storage, the hosting platform usually provides a long technical address. Such addresses are difficult to remember and not suitable for professional use. A custom domain gives the website a simple and meaningful identity, for example www.nandinis-portfolio.in.

For vocational projects such as online portfolios, school portals, or small business websites, a custom domain improves credibility and makes the site easier for users to find and share.

B. Domain Name System (DNS) in Website Access

The Domain Name System (DNS) acts like a directory of the internet. It converts human-readable domain names into machine-readable IP addresses so computers can locate the correct server. Without DNS, users would need to remember long numerical addresses instead of simple website names.

DNS responses are stored temporarily in a system called TTL (Time to Live). This caching process reduces repeated lookups and helps websites load faster. The main stages of this lookup process are illustrated in Fig. 4.2.3, which shows how a browser request travels through DNS servers before reaching the website server.

C. Connecting a Custom Domain to Cloud Hosting

To connect a domain to a cloud website, a **CNAME record** is used. It links a subdomain (like *www*) to the cloud website address.

Steps:

Step 1. Log in to domain account

Step 2. Open DNS settings

Step 3. Add a CNAME record

Step 4. Enter cloud website link

After setup, users visiting the domain will reach the website.

D. Common DNS Records Used in Web Hosting

Different DNS records perform specific roles in connecting a domain name with web servers, email services, and verification systems. Table 2.3 presents common DNS records using examples from Nandini's portfolio website, showing how each record helps the website function correctly on the internet.

DNS Records

Table 2.3: Important DNS Records

Record	Purpose
A	Connects domain to server (IPv4)
AAAA	Connects domain to server (IPv6)
CNAME	Links subdomain (www) to website
MX	Sets email server
TXT	Verifies domain ownership

These DNS configurations allow cloud-hosted websites to function like professional internet services while keeping the hosting system simple and reliable for learners and small projects.

2.4 Content Delivery Networks

A CDN (Content Delivery Network) is a network of servers in different locations. It sends website data from the nearest server, so the website loads faster.

Example: Like Mobile network gives fast internet using nearby towers, CDN uses nearby servers.

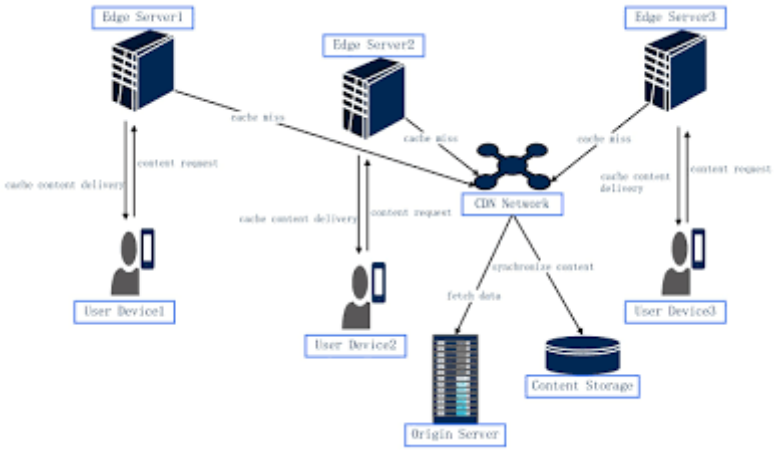


Fig. 2.3: A CDN Serving Users from Nearby Edge Servers

Practical Activity 2.1. Creating and Uploading a Static Webpage (index.html)

Objective

To create a simple static webpage using HTML and upload it to cloud storage for access and sharing.

Materials Required:

- Computer / Laptop with internet access
- Text editor (Notepad / VS Code)
- Web browser (Chrome/Firefox)
- Google Drive account

Procedure

Part A: Create a Static Webpage

Step 1. Open a text editor like Notepad or VS Code.

Step 2. Type the following basic HTML code:

```
<!DOCTYPE html>
<html>
<head>
    <title>My First Webpage</title>
</head>
<body>
    <h1>Welcome to My Webpage</h1>
    <p>This is my first static webpage created using
HTML.</p>
</body>
</html>
```

Step 3. Click **File** → **Save As**.

Step 4. Save the file as index.html.

Step 5. Open the file in a web browser to view your webpage.

Part B: Upload to Cloud Storage

Step 6. Open Google Drive in your browser.

Step 7. Click on New → File Upload.

Step 8. Select and upload the index.html file.

Step 9. After uploading, right-click the file and select Get Link.

Step 10. Set permission to Anyone with the link can view.

Step 11. Copy and share the link.

Observation

Students observe that HTML code creates a webpage structure and cloud storage allows easy uploading and sharing of files online.

Conclusion

A static webpage can be created using simple HTML and shared easily using cloud platforms like Google Drive, making content accessible from anywhere.

Summary

In this session, students learned the difference between static and dynamic hosting, where static websites display the same content to all users while dynamic websites generate content based on user interaction and data. They understood the advantages of static hosting such as speed, low cost, and simplicity, and explored object storage as a reliable method for hosting static websites on the cloud. Students also gained knowledge about custom domains and the Domain Name System (DNS), including various DNS records and how they connect domain names to websites. Additionally, they learned how Content Delivery Networks (CDNs) improve website performance by delivering content from nearby servers.

Check Your Progress

A. Multiple Choice Questions

1. A student wants to create a simple portfolio website with fixed content. Which hosting type is most suitable?
   (a) Dynamic hosting (b) Static hosting (c) Cloud database hosting (d) Virtual machine hosting
2. Which feature makes dynamic websites different from static websites?
   (a) Use of HTML files (b) Fixed content for all users (c) Content generation using server-side processing (d) No internet requirement
3. Which cloud service is best suited for hosting a static website at low cost?
   (a) Virtual Machines (b) Object Storage (c) Load Balancer (d) Database Server
4. What is the main role of DNS in website access?
   (a) Store website files (b) Convert domain names into IP addresses (c) Design web pages (d) Increase website security
5. Which technology improves website loading speed by using nearby servers?
   (a) DNS (b) CDN (c) FTP (d) HTTP

B. Fill in the Blanks

1. A _____ website shows the same content to every user.
2. Dynamic websites use _____ and databases to generate content.
3. A _____ domain makes a website easy to remember and professional.
4. A _____ record connects a domain name to another domain name.
5. A CDN delivers website data from the nearest _____ to the user.

C. True or False

1. Static websites require database processing for each request.
2. Dynamic hosting is generally more expensive than static hosting.
3. Object storage can store website files with unique links.
4. DNS allows users to access websites using easy-to-remember names.
5. CDN slows down website performance by adding extra steps.

D. Short Answer Questions

1. Why is static hosting suitable for a personal resume website?
2. Give one example of a dynamic website and explain why it is dynamic.
3. What is the role of object storage in static hosting?
4. How does a CNAME record help in website access?
5. How does a CDN improve user experience?

Session 3. Backup and Restore Planning in Cloud Computing

Kabir had spent two weeks building a database of every book in his school library. One afternoon, a classmate accidentally deleted the entire spreadsheet. No recycle bin. No undo. Kabir sat staring at a blank screen. The teacher, walked past and said: "This is why we keep copies." She explained how data is stored in multiple locations, so no single failure can destroy anything. What she described maps to a diagram something like Figure 3.1.

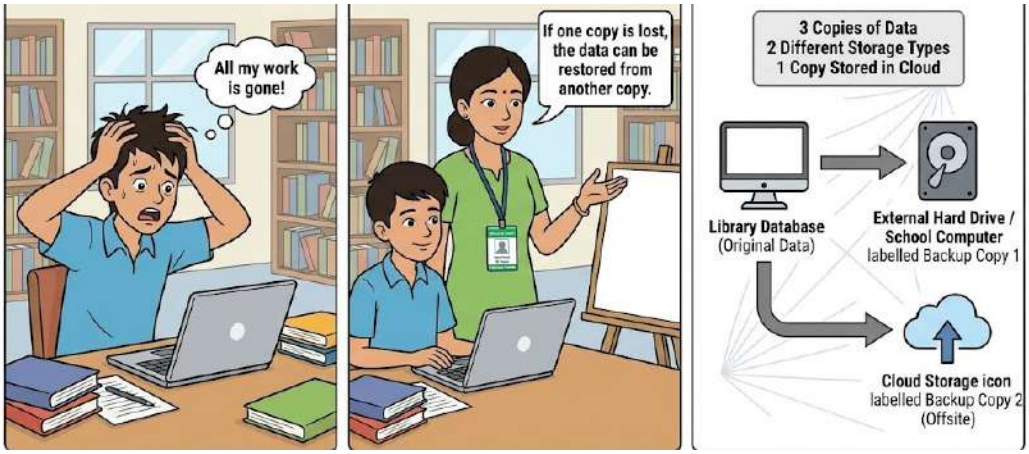


Fig. 3.1: The 3-2-1 Backup Structure

3.1 RPO and RTO

Data recovery planning helps organizations restore their data and services after a system failure or accidental data loss. Clear recovery objectives guide how frequently backups should be taken and how quickly systems must be restored to normal operation.

A. Recovery Point Objective (RPO)

RPO represents the maximum amount of recent data that can be lost during a system failure. The value is measured in time and depends on how frequently backups are created.

B. Recovery Time Objective (RTO)

RTO represents the maximum acceptable time required to restore a system after a failure. The restoration process must complete within this time so that services remain usable.

C. Application-Based Recovery Needs

Different applications require different recovery limits depending on their importance. Typical requirements are illustrated in Table 3.1.

Table 3.1: RPO and RTO Requirements Across Applications

Application	Acceptable RPO	Acceptable RTO	Reason
Online banking systems	Few seconds	Under 5 minutes	Financial transactions must remain accurate

E-commerce platforms	15–30 minutes	Under 1 hour	Orders and sales depend on continuous service
School library database	1 day	Few hours	Data can be re-entered if necessary
Personal photo storage	1 week	1–2 days	Temporary loss has limited operational impact

Before choosing any backup method, you need two numbers. Without them, your backup plan is just guesswork.

4.5.2 Backup Types

Three methods exist for capturing backup copies of data. Each balances storage space, backup speed, and restore complexity differently.

A. Full Backup

A full backup copies all files in the system every time the backup runs. This method simplifies restoration because the entire dataset exists in one backup copy.

B. Incremental Backup

Incremental backup stores only the files changed since the previous backup. This approach saves storage space and reduces backup time but requires multiple backup files during restoration.

C. Differential Backup

Differential backup stores all changes made since the last full backup. Restoration requires the full backup and the latest differential copy. The comparison of these methods appears in Table 3.2, while their daily data growth is illustrated in Figure 3.2.

Table 3.2: Backup Type Comparison

Method	Data Copied	Backup Speed	Restore Complexity
Full	Entire dataset	Slowest	Simple restoration
Incremental	Changes since last backup	Fastest	Multiple files required
Differential	Changes since last full backup	Medium	Full backup plus latest differential

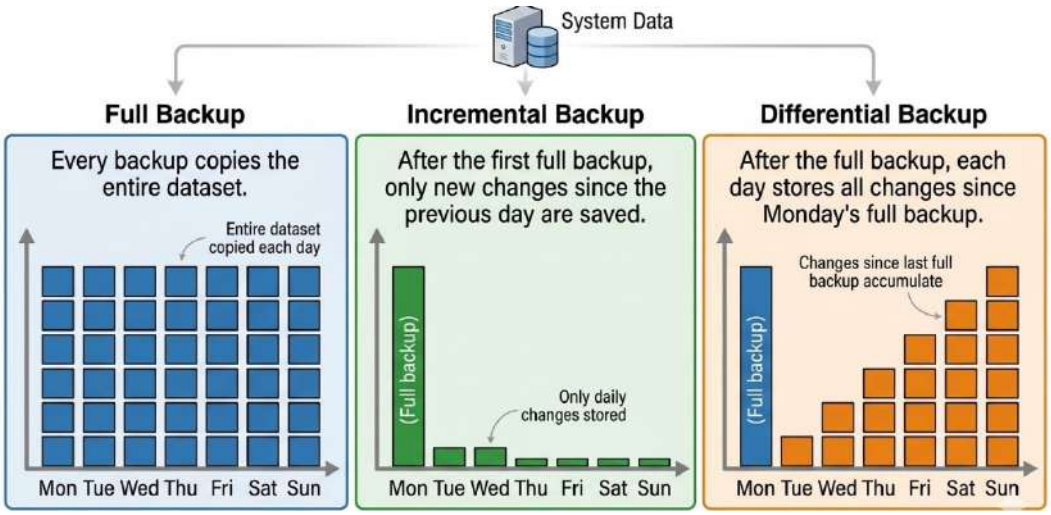


Fig. 3.2: Daily Data Volume for Each Backup Type Over One Week

3.3 The 3-2-1 Backup Rule

This is arguably the part most people overlook when planning data protection. The 3-2-1 rule is a simple, memorable framework that professionals use to ensure no single failure can destroy all copies of data.

A. Multiple Copies of Data

Professional backup planning keeps three copies of important data to avoid permanent loss from a single failure.

B. Different Storage Media

Copies are stored on at least two different storage systems such as local disks and cloud storage.

C. Offsite Backup Location

One copy remains at a separate location or cloud region so that local disasters cannot affect every copy simultaneously.

This strategy is widely used in cloud data protection systems and large organizational data centers.

Google Photos works exactly this way. When you enable backup on your phone, Google stores multiple copies across geographically separate data centers.

3.4 Automated Backup Tools

Manual backups are unreliable. People forget. They postpone. They assume the last backup is more recent than it actually is.

Cloud providers solve this with automated backup services. AWS Backup, Azure Backup, and Google Cloud's scheduled snapshot features let you define a backup policy: back up every six hours, keep the last seven daily backups, and keep one backup per week for a month. The system runs automatically with no human input required, and it sends alerts if a backup fails.

In India, many small businesses using cloud-hosted Tally or accounting software rely on automated nightly backups to Google Drive or Dropbox. The

backup runs at 2 AM without anyone being awake, and the business owner can restore to the previous night's data in under ten minutes if something goes wrong.

Practical Activity 3.1. Setting Up an Automated Backup Using Google Drive

Objective

To understand how to set up an automated backup of files using Google Drive to ensure data safety and easy access.

Materials Required

- Computer / Laptop with internet access
- Google account
- Web browser (Chrome/Firefox)
- Google Drive application (optional for desktop/mobile)

Procedure

Step 1. Access Google Drive

- Open a web browser and go to Google Drive.
- Sign in using your Google account credentials.

Step 2. Install Google Drive for Desktop (for Automation)

- Download and install Google Drive for Desktop from the official website.
- Sign in with your Google account.

Step 3. Set Up Automatic Backup

- Open Google Drive settings on your computer.
- Select the option “Folders from your computer” or “Add folder”.
- Choose the folder (e.g., Documents, Pictures, or Desktop) to back up.
- Select “Sync with Google Drive” or “Back up to Google Photos” (if images).
- Click “Save” to enable automatic backup.

Step 4. Verify Backup

- Open Google Drive in your browser.
- Check if the selected folder and files are visible.
- Make a change (add/edit a file) in the folder and verify if it updates automatically in Drive.

Observation / Result

Files stored in the selected folder are automatically backed up and synchronized to Google Drive, ensuring secure and updated data storage.

Conclusion

Automated backup using Google Drive helps in protecting important files from data loss and allows easy access from anywhere through cloud storage.

Summary

In this session, students learned the importance of backup and restore planning to protect data from loss. They understood key concepts such as Recovery Point Objective (RPO) and Recovery Time Objective (RTO), which define acceptable data loss and system recovery time. Students explored different backup types—full, incremental, and differential—and compared their speed, storage, and restoration processes. They also learned the 3-2-1 backup rule for ensuring data safety through multiple copies stored on different media and locations. Additionally, students gained knowledge about automated backup tools used in cloud computing and performed a practical activity to set up automatic backup using Google Drive, ensuring secure and continuous data protection.

Check Your Progress

A. Multiple Choice Questions

1. RPO represents which of the following in a backup plan?
   (a) Time to restore system (b) Maximum acceptable data loss (c) Number of backups (d) Storage capacity
2. Which backup type copies only the data changed since the last backup?
   (a) Full (b) Incremental (c) Differential (d) Mirror
3. The 3-2-1 backup rule recommends storing one copy of data:
   (a) On the same device (b) On paper (c) At a different location (d) In RAM
4. RTO defines:
   (a) Data storage limit (b) Backup frequency (c) Maximum recovery time after failure (d) Internet speed
5. Which backup type requires only one file to restore all data?
   (a) Incremental (b) Differential (c) Full (d) Snapshot

B. Fill in the Blanks

1. _____ is the maximum acceptable time to restore a system after failure.
2. A _____ backup copies all files every time it runs.

3. The 3-2-1 rule requires _____ copies of data.
4. Incremental backup stores changes since the last _____.
5. Automated backups reduce the need for _____ intervention.

C. True or False

1. RPO is measured in terms of acceptable data loss time.
2. Incremental backups take more storage than full backups.
3. The 3-2-1 rule ensures data safety from single point failure.
4. Differential backups store changes since the last full backup.
5. Manual backups are always more reliable than automated backups.

D. Short Answer Questions

1. Why is RTO important for online services?
2. Give one advantage of incremental backup.
3. What is the purpose of the 3-2-1 backup rule?
4. Why are automated backups preferred over manual backups?
5. Differentiate between full and differential backup (one point).

Session 4. Cloud Project Development & Collaboration

In the previous modules, you learned what cloud computing is, how cloud services are organized, and how organizations use cloud tools. In Module 5, you will apply all of that knowledge to build a real project — a **School Events Portal** — from planning to publishing, using completely free tools available on the Internet.

This module takes you through the complete journey: planning a project, creating a database using Google Sheets, building a website using Google Sites, and making it available to the world — all without writing a single line of code, spending any money, entering a mobile OTP, or providing a credit card. Only a Google account is needed.

4.1 Project Planning & Requirement Analysis

Rahul is a student at Greenfield Public School. Every Monday morning, students miss important school events because there is no single place to check upcoming activities. The notice board is outdated and classroom announcements are quickly forgotten. Rahul wonders: "Can we build something on the Internet to keep everyone informed?"

His teacher, Mrs. Sharma, says. "Yes, Rahul. That is exactly what a School Events Portal does. Before we build it, we must plan it carefully. Let us start with a Project Charter." Figure 4.1 shows how a planned project flows from idea to a live website.

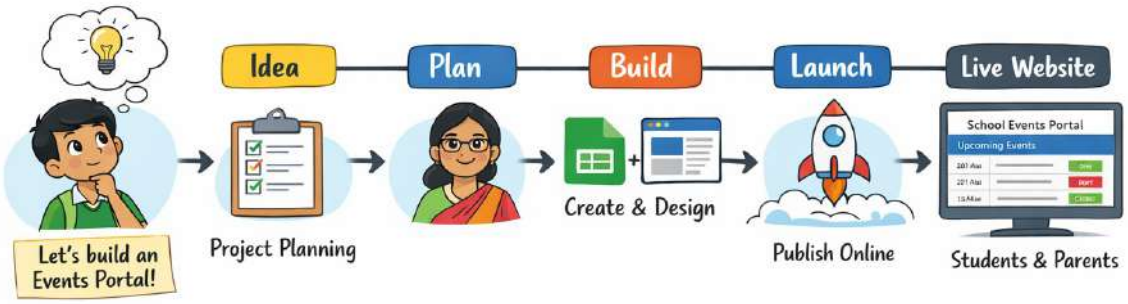


Fig. 4.1: Project lifecycle — from idea to a live cloud-hosted website

4.1.1 What is a Project?

A project is a set of planned tasks carried out to achieve a specific goal within a fixed time. In cloud computing, a project usually means creating something useful — a website, an app, or an information system — using cloud tools.

Example: Building a School Events Portal that shows upcoming school events to all students and parents, using Google Sheets and Google Sites.

4.1.2 Software Development Life Cycle (SDLC)

Before any real cloud project is built, developers follow a standard process called the Software Development Life Cycle (SDLC). It ensures the project is well-planned, properly built, tested, and maintained. Table 4.1 shows the four main SDLC phases.

Table 4.1: SDLC Phases for the School Events Portal

Sn	Phase	What Happens	Our Project
1	Plan	Define the problem, users, and requirements	Write the Project Charter and requirements list
2	Build	Create the database and build the website	Create Google Sheet and Google Site
3	Deploy	Publish and make it available on the Internet	Publish the Google Site for public access
4	Maintain	Update content and fix problems	Update the Google Sheet with new events

4.1.3 Project Charter

A Project Charter is a short document written before building anything. It defines the problem you are solving, who will use your solution, and what it must do. Writing a charter prevents wasted effort and keeps the team focused.

Table 4.2: Project Charter — School Events Portal

Field	Details
Project Name	School Events Portal
Problem Statement	Students and parents miss school events because event information is scattered across notice boards and verbal announcements.
Target Users	Students, Parents, and Teachers
Main Feature	Display upcoming school events with dates, names, and registration status in one place on the Internet
Tools Required	Google Sheets (event database) + Google Sites (website builder)
Cost	Completely free — only a Google account is needed. No OTP, no credit card.

4.1.4 Requirements

Requirements describe exactly what the system must do and how it must perform. They are divided into two types:

Functional Requirements describe what the system does — its features and functions.

Display a list of upcoming school events

Show event date, event name, and registration status for each event

Be accessible to anyone on the Internet without needing to log in

Non-Functional Requirements describe how well the system performs — quality factors.

The website must work on both mobile phones and computers

The page must load quickly without delays

Event data must be easy to update by the teacher without technical knowledge

4.1.5 Wireframe

A wireframe is a simple sketch of a website's layout — like a floor plan drawn before constructing a building. It shows where each element will be placed: the title, event list, and footer. Figure 4.2 shows the wireframe for our School Events Portal.

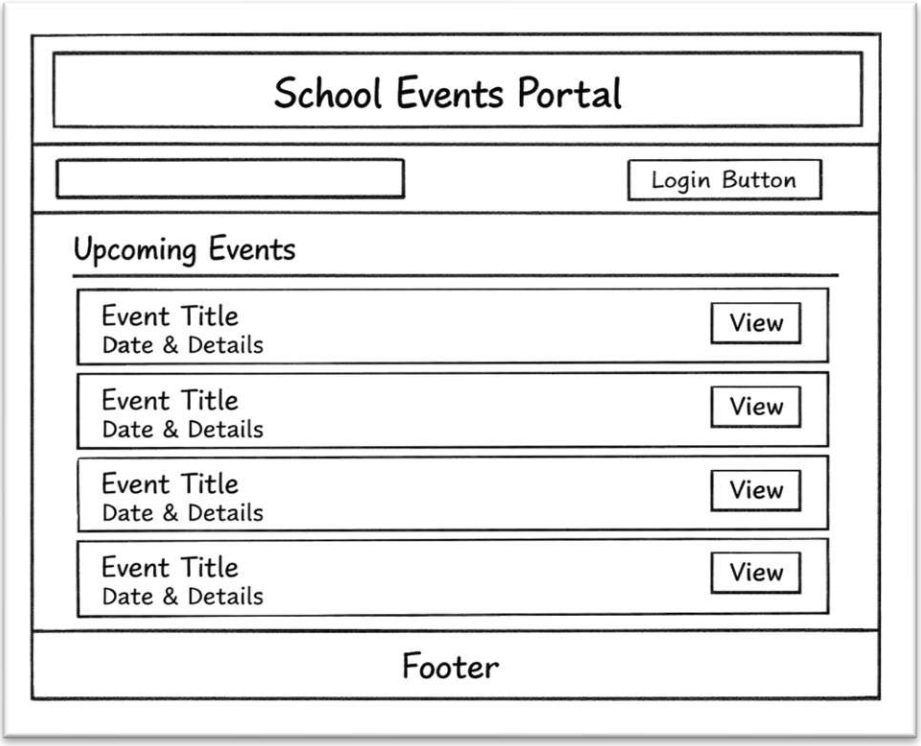


Fig. 4.2: Wireframe of the School Events Portal (Redraw)

Note: You can draw your wireframe on paper or use Google Slides. A wireframe does not need to look perfect — its purpose is to plan the layout before building.

Summary

In this session, students learned the importance of planning and collaboration in developing a cloud-based project. They understood the concept of a project and the role of the Software Development Life Cycle (SDLC), including the phases of planning, building, deployment, and maintenance. Students explored how a Project Charter helps define the problem, target users, goals, and tools required before starting development. They also learned to differentiate between functional and non-functional requirements and understood how a wireframe helps in visualizing the structure and layout of a website before implementation. Overall, students developed an understanding of systematic project development using cloud tools.

Check Your Progress

A. Multiple Choice Questions

1. What is the first phase of the Software Development Life Cycle (SDLC)?
   (a) Testing (b) Deployment (c) Planning / Requirement Analysis (d) Maintenance
2. A Project Charter is prepared to:
   (a) Design the website layout (b) Define project goals and requirements
   (c) Write program code (d) Test the system
3. Which of the following is a functional requirement?
   (a) Website should load quickly (b) Website must display event details
   (c) Website should be mobile-friendly (d) Website should be secure
4. A wireframe is used to:
   (a) Store data (b) Test software (c) Show layout of a website (d) Run code
5. Which SDLC phase involves publishing the project online?
   (a) Planning (b) Building (c) Deployment (d) Maintenance

B. Fill in the Blanks

1. The process used to develop software step-by-step is called the _____.
2. A _____ defines the project problem, users, and goals.
3. Requirements that describe system features are called _____ requirements.
4. A _____ is a simple sketch of a website layout.
5. The phase in which the project is made available online is called _____.

C. True or False

1. Planning is done after building the project.
2. A Project Charter helps avoid confusion during development.

3. Functional requirements describe how well a system performs.
4. A wireframe helps in visualizing the structure of a website.
5. Maintenance involves updating and improving the system after deployment.

D. Short Answer Questions

1. Why is a Project Charter important before starting a project?
2. Give one example of a functional and one non-functional requirement.
3. What is the role of the deployment phase in SDLC?
4. Why is a wireframe useful in website development?
5. List any two phases of SDLC and their purpose.

Answer Key

Module 1. Cloud Platforms & Architecture

Session 1. Global Cloud Providers

A. Multiple Choice Questions (MCQs)

1. (b) 2. (c) 3. (c) 4. (b) 5. (c)

B. Fill in the Blanks

1. Pay-as-you-go 2. Broad network access 3. Availability Zones 4. Traditional 5. Measured

C. True or False:

1. False 2. True 3. False 4. True 5. False

Session 2. Cloud Application Workflow

A. Multiple Choice Questions (MCQs)

1. (b), 2. (c), 3. (b), 4. (c), 5. (b)

B. Fill in the Blanks

1. Client, 2. Server, 3. Packets, 4. Protocols, 5. Database

C. True or False

1. True, 2. False, 3. True, 4. False, 5. True

Session 3. Cloud Compute Power in Virtual Machines

A. Multiple Choice Questions (MCQs)

1. (b), 2. (b), 3. (a), 4. (b), 5. (a)

B. Fill in the Blanks

1. Virtualization, 2. Virtual Machine (VM), 3. Hypervisor, 4. Type 1, 5. Cloud

True or False

1. True, 2. False, 3. True, 4. False, 5. True

Module 2. Cloud Security & Data Protection

Session 1. Authentication & Multi-Factor Authentication for Secure Cloud Access

A. Multiple Choice Questions (MCQs)

1. (b), 2. (c), 3. (c), 4. (c), 5. (b)

B. Fill in the Blanks

1. identity, 2. Shared, 3. have, 4. security, 5. are

C. True or False

1. False, 2. True, 3. True, 4. False, 5. True

Session 2. IAM Roles & Permissions Concept

A. Multiple Choice Questions (MCQs)

1. (b), 2. (c), 3. (c), 4. (a), 5. (b)

B. Fill in the Blanks

1. Access, 2. Group, 3. temporary, 4. JSON, 5. Deny

C. True or False

1. True, 2. True, 3. False, 4. True, 5. False

Session 3. Encryption Basics and Secure Data Storage in Cloud Computing

A. Multiple Choice Questions (MCQs)

(1) b, (2) c, (3) b, (4) b, (5) b;

B. Fill in the Blanks

1. Plaintext, 2. Ciphertext, 3. rest, 4. transit, 5. private;

C. True or False

1. True, 2. False, 3. True, 4. True, 5. False.

Session 4. Secure Web Communication using HTTPS and SSL/TLS

A. Multiple Choice Questions (MCQs)

(1) b, (2) b, (3) b, (4) c, (5) b

B. Fill in the blanks

1. Hypertext, 2. HTTPS, 3. secure (encrypted), 4. Certificate Authorities (CAs), SSL/TLS 5. handshake;

C. True or False

1. True, 2. True, 3. False, 4. True, 5. True.

Module 3. Modern Cloud Applications & Services

Session 1. Building Modern Applications Using Cloud Services

A. Multiple Choice Questions

1. (c), 2. (c), 3. (c), 4. (b), 5. (c)

B. Fill in the Blanks

1. Internet 2. Virtual Machine 3. Primary 4. Distributed / Global 5. AI

C. True or False

1. False 2. True 3. True 4. False 5. True

Session 2. E-Commerce Systems and Secure Digital Payments

A. Multiple Choice Questions

1. (b), 2. (d), 3. (c), 4. (b), 5. (b)

B. Fill in the Blanks

1. ProductID 2. Temporarily 3. Payment Gateway 4. HTTPS 5. Tokenization

C. True or False

1. False 2. True 3. False 4. True 5. True

Session 3. Introduction to IoT and Edge Computing

A. Multiple Choice Questions

1. (c), 2. (b), 3. (c), 4. (c), 5. (b)

B. Fill in the Blanks

1. Sensors 2. Gateway 3. Edge 4. Cloud Backend 5. Applications

C. True or False

1. False 2. True 3. False 4. True 5. False

Session 4. Real-Time Location Services and Navigation Systems

A. Multiple Choice Questions

1. (b), 2. (c), 3. (b), 4. (b), 5. (a)

B. Fill in the Blanks

1. Latitude, Longitude 2. Trilateration 3. Maps API 4. Pings 5. Cloud

C. True or False

1. False 2. True 3. False 4. True 5. True

Module 4. Cloud Deployment & Operations

Session 1. Application Deployment Workflow

A. Multiple Choice Questions (MCQs)

1. (b), 2. (c), 3. (b), 4. (c), 5. (b)

B. Fill in the Blanks

1. Deployment 2. Integration 3. Delivery / Deployment 4. Monitoring 5. Configuration / Script

C. True or False

1. True 2. False 3. True 4. False 5. True

Session 2. Introduction to Website Hosting and Cloud Deployment

A. Multiple Choice Questions

(b) 2. (c) 3. (b) 4. (b) 5. (b)

B. Fill in the Blanks

1. static 2. server-side code 3. custom 4. CNAME 5. server

C. True or False

1. False 2. True 3. True 4. True 5. False

Session 3. Backup and Restore Planning in Cloud Computing

A. Multiple Choice Questions

1. (b) 2. (b) 3. (c) 4. (c) 5. (c)

B. Fill in the Blanks

1. RTO 2. full 3. three 4. backup 5. human

C. True or False

1. True 2. False 3. True 4. True 5. False

Session 4. Cloud Project Development and Collaboration

A. Multiple Choice Questions

1. (c) 2. (b) 3. (b) 4. (c) 5. (c)

B. Fill in the Blanks

1. SDLC 2. Project Charter 3. functional 4. wireframe 5. deployment

C. True or False

1. False 2. True 3. False 4. True 5. True