

SOFTWARE ENGINEERING

UNIT - 5

TOPIC – 3

INTRODUCTION TO CLOUD AND ITS SERVICES

Introduction to Cloud Computing

Cloud computing is a way to use computer resources like storage, applications, and processing power over the internet instead of owning and managing physical hardware. It lets you access these resources whenever you need them, and you only pay for what you use.

What is Cloud Computing?

Cloud computing means you can:

- **Store files** like photos, videos, and documents on the internet, so you don't need a big hard drive at home.
- **Run programs** that require a lot of computing power without buying expensive computers.
- **Use software** like email or document editors directly through the internet without downloading or installing them.

Imagine renting a car instead of buying one. You use the car only when needed and pay for it based on how long you use it. Similarly, cloud computing gives you the resources you need without the hassle of owning them.

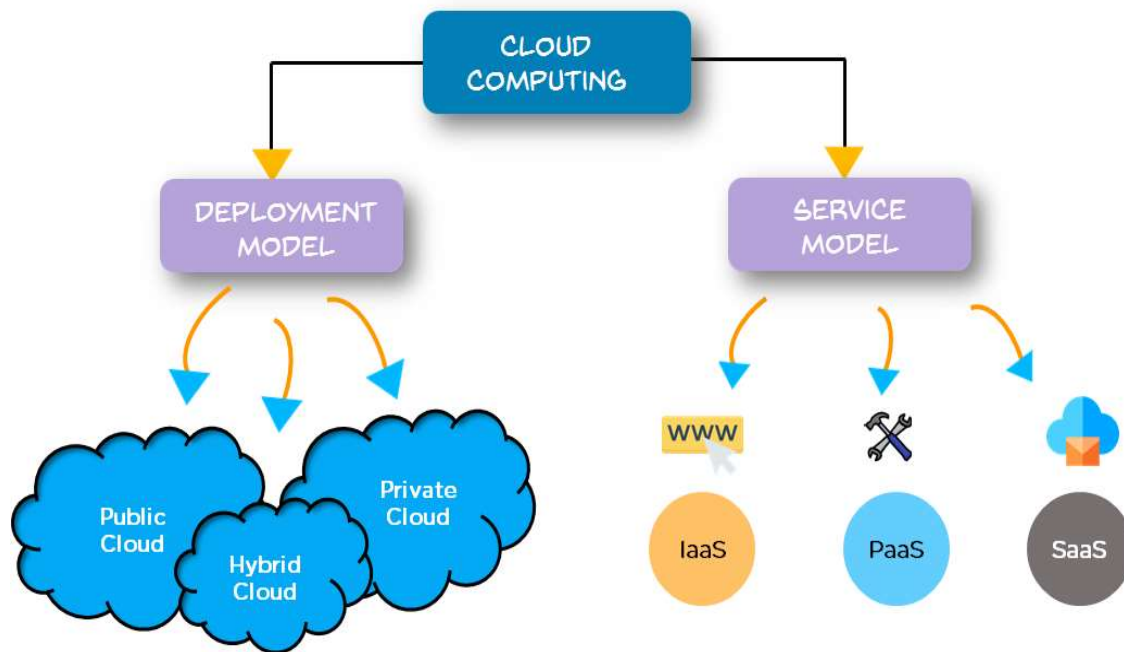
Features of Cloud Computing

1. **Availability:** Your data and applications are always accessible. Cloud providers work hard to make sure there is little to no downtime.
 - *Example:* Like water from a tap—always ready whenever you need it.
2. **Scalability:** You can adjust the number of resources based on your needs. If you need more storage or computing power, you can get it instantly, and when you don't need as much, you can scale back down.

- *Example:* Like a balloon you can inflate or deflate depending on how much air you need.
- 3. **Pay-as-you-go:** You only pay for what you use, saving money by not having to invest in expensive hardware upfront.
 - *Example:* Like a phone bill, where you pay for the minutes and data you use.

Types of Cloud Computing

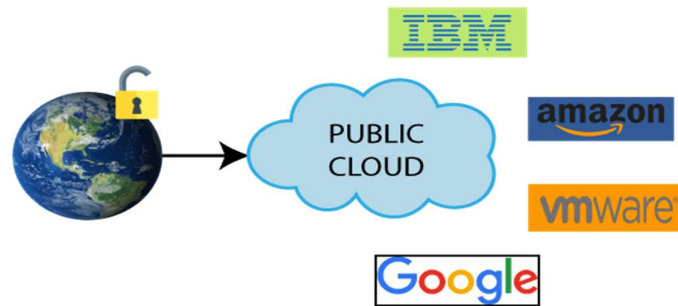
Cloud computing can be categorized based on **how it is deployed** and **the services it offers**.



Cloud Deployments

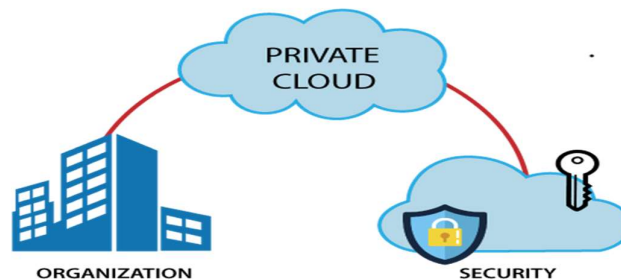
Deployment refers to how and where cloud services are made available. The three types of deployments are:

1. **Public Cloud:**
 - Resources like servers and storage are shared by multiple users and managed by companies like Amazon, Microsoft, and Google.
 - *Example:* A public library where anyone can borrow books.



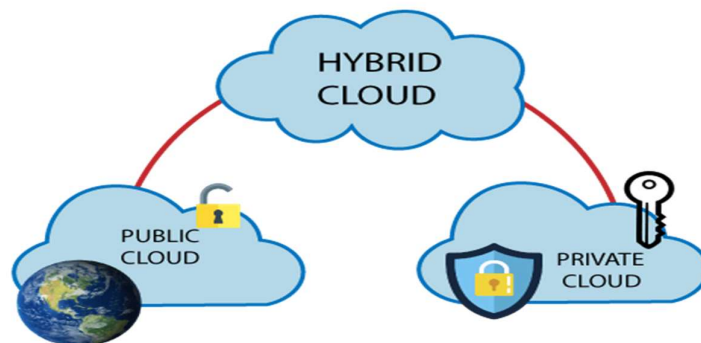
2. Private Cloud:

- Resources are used by a single organization, offering more control and security. It can be managed by the organization or a third-party provider.
- Example:* A private library owned by a school, only accessible to students and staff.



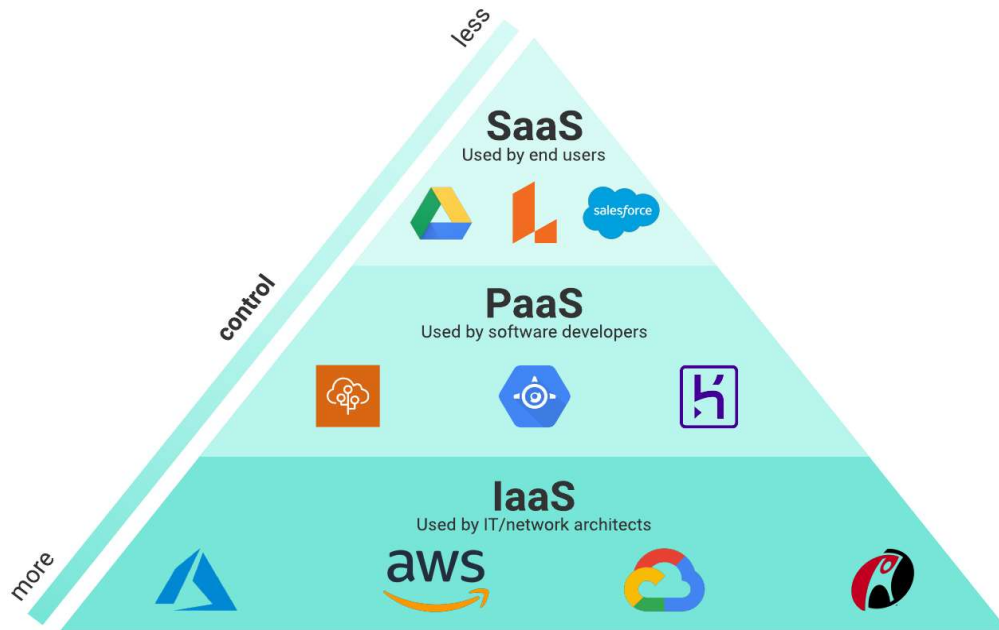
3. Hybrid Cloud:

- A combination of public and private clouds. Organizations can use private clouds for sensitive tasks and public clouds for less critical activities.
- Example:* A bakery with a private kitchen for secret recipes and a public shop to sell the goods.



Cloud Services

Cloud services are categorized based on the level of control they provide:



1. Infrastructure as a Service (IaaS):

- Provides basic resources like virtual machines and storage. You are responsible for managing everything else.
- *Example:* Renting an empty plot where you can build anything you want.

2. Platform as a Service (PaaS):

- Offers tools and platforms for developers to build applications without worrying about the underlying hardware.
- *Example:* A fully equipped kitchen where you can cook without buying the appliances.

3. Software as a Service (SaaS):

- Provides ready-made software that you can use directly over the internet.
- *Example:* Using Gmail or Microsoft Word online without installing anything on your computer.

How AWS Helps with Cloud Computing

AWS (Amazon Web Services) is a platform that offers cloud services for building, managing, and scaling applications easily. It removes the need to own physical servers, allowing you to focus on your projects.

Key AWS Services

1. Amazon EC2:

- Let's you rent virtual servers to run applications.
- *Example:* Like booking a hotel room based on your needs—a single room or a suite.

2. Amazon S3:

- Provides storage space to save and manage your files securely.
- *Example:* A giant online locker where you can keep your files.

3. Amazon RDS:

- Offers managed databases, so you don't have to worry about setting them up or maintaining them.
- *Example:* Like hiring an assistant to organize and maintain your files.

4. Amazon VPC:

- Allows you to create a private and secure section of the cloud for your resources.
- *Example:* A private office within a shared building where only authorized people can enter.

5. AWS Lambda:

- Runs your code automatically when needed without requiring servers.
- *Example:* A light sensor that turns on a light only when it detects motion.

Steps to Create Cloud Infrastructure Using AWS

Here is how you can set up infrastructure on AWS step by step:

1. Sign Up:

- Go to the AWS website and register for an account. Fill in your details and set up a payment method.

2. **Log In:**
 - Use your account credentials to access the AWS dashboard.
3. **Explore Services:**
 - Familiarize yourself with AWS services like EC2 for virtual servers, S3 for storage, and VPC for networking.
4. **Launch a Virtual Server:**
 - Go to EC2, choose an instance type, select an operating system, and launch your virtual machine.
5. **Set Up Storage:**
 - Use S3 to create a storage bucket where you can upload and manage your files.
6. **Configure Networking:**
 - Use VPC to design a secure network for your application.
7. **Monitor Resources:**
 - Use CloudWatch to track and optimize your usage, ensuring efficient performance.
8. **Secure Access:**
 - Set up user roles and permissions using AWS IAM to control who can access your resources.

Deploying a Web Application

To deploy a static web application using Docker and Nginx, follow these steps:

Steps to Deploy

1. **Push Code to GitHub:**
 - Store your application's code on GitHub. Initialize a Git repository, commit the files, and push them to a GitHub repository.
2. **Create a Virtual Machine:**
 - Launch an EC2 instance with Ubuntu or another operating system. Configure storage and security groups for access.
3. **Clone the Code and Install Docker:**
 - Use Git to clone your application's code from GitHub onto the virtual machine.

- Install Docker on the virtual machine. Docker is a tool that helps package and run your application in containers.

4. Write and Build a Dockerfile:

- Write a Dockerfile to define how your application will run. Build the Docker image using the **docker build** command.

5. Run the Application:

- Use the **docker run** command to create a container from the Docker image. Map it to port 80 so it can be accessed from the internet.
- Use the public IP address of your EC2 instance to access your web application.