



**ANGAZA CENTER**

# **CLOUD SERVICE MODELS**

Lilian Thairu, Angaza Center ADL

[www.angazacenter.org](http://www.angazacenter.org)



# Lesson Overview

- **Objectives:**

- Understand what Cloud Service Models are
- Explore PaaS
- Learn the advantages of PaaS
- Understand the Shared Responsibility Model in PaaS



# By Definition

**The Cloud Service Models** refers to how the computing resources are being offered to users over the internet.





**Cloud Service  
Models**

**Software as a  
Service  
(SaaS)**

# SaaS

SaaS is like going to a restaurant and getting a glass of water served to your table, or buying bottled water. You don't worry about where the water came from, how it was filtered, or whether the glass/bottle was cleaned. Everything is handled for you, and you simply consume the service.

In the cloud model, SaaS provides **fully functional software applications** over the internet. Users don't need to install or maintain anything—they simply access the software through a browser or app. The provider handles everything: servers, storage, data, updates, and security.

**Ideal for:** end users who need ready-to-use tools.

Examples include **Google Workspace (Gmail, Docs, etc.)**, **Microsoft 365**, and **Salesforce**.

# Advantages of SaaS

- **Ready-to-Use:** No installation or setup—users can start using the software immediately.
- **Automatic Updates:** The provider handles all updates, patches, and upgrades.
- **Accessibility:** Accessible from anywhere via a web browser or mobile app, making remote work easy.
- **Cost-Effective:** No need to purchase licenses or infrastructure—subscription-based pricing lowers upfront costs.
- **Scalable and Flexible:** Easily add or remove users, features, or storage as business needs change.

# Disadvantages of SaaS

- **No control over features:** You can't change how the software works beyond what's provided.
- **Internet dependency:** You need a stable connection to use it.
- **Data privacy concerns:** Your data lives on someone else's servers.
- **Subscription costs:** You keep paying for as long as you use it, and prices can rise.





**Cloud Service  
Models**

**Platform as a  
Service  
(PaaS)**



# PaaS

PaaS is like receiving filtered and heated water through a tap. You don't need to worry about plumbing, boilers, or treatment systems. You simply turn on the faucet and use the water for cooking or cleaning. The supplier handles water quality, temperature, and delivery; you just use it for your intended task.

In cloud computing, PaaS delivers a **ready-to-use development and deployment environment**. The provider manages infrastructure, operating systems, and middleware, allowing developers to focus solely on building and running applications. It simplifies the development lifecycle and speeds up time to market.

**Ideal for:** developers who don't want to manage the underlying infrastructure.

Examples include: **Google App Engine, Microsoft Azure App Services, and Heroku.**

# Advantages of PaaS

- **Faster Development and Deployment:** Developers can focus on coding and app logic instead of managing infrastructure.
- **Built-in Tools and Services:** Includes databases, application frameworks, monitoring tools, and more to accelerate development.
- **Lower Maintenance Overhead:** The provider manages software updates, security patches, and infrastructure performance.
- **Collaboration-Friendly:** Multiple developers can work on the same project with streamlined testing and deployment workflows.
- **Cost Savings for Dev Teams:** Reduces need for infrastructure specialists and shortens development cycles.



# Disadvantages of PaaS

- **Less control:** You can't fully customize the underlying infrastructure.
- **Vendor lock-in:** Moving your apps to another provider can be hard.
- **Limited compatibility:** Some platforms may not support all programming languages or tools you want.
- **Cost for scaling:** Higher usage can lead to higher fees.



**Cloud Service  
Models**

# **Infrastructure as a Service (IaaS)**



# IaaS

Think of IaaS like receiving raw, untreated water through a pipe directly into your property. You are responsible for filtering, heating, or treating the water as needed and for deciding how and when to use it. In this case, the water supplier only provides the infrastructure (pipes, pumps, water delivery), while you handle everything else.

In cloud terms, IaaS provides the **basic computing infrastructure**: virtual machines, storage, networks, and servers. You have complete control over the operating systems and deployed applications, but the cloud provider manages the hardware.

**Ideal for:** IT administrators who want flexibility to customize environments and install any software they need.

Examples: include **Amazon EC2**, **Microsoft Azure VMs**, and **Google Compute Engine**.

# Advantages of IaaS

- **Full Control Over Infrastructure:** Users can customize hardware, networks, and storage to fit their exact needs.
- **Scalability:** Resources can be scaled up or down quickly to meet changing demands.
- **Cost Efficiency:** Pay only for what you use; avoids capital expenditure on physical hardware.
- **Flexibility:** Supports any operating system or application, giving IT teams freedom to configure systems as needed.
- **Disaster Recovery and Backup:** Many IaaS providers include automated backup and disaster recovery features.



# Disadvantages of IaaS

- **You manage more:** You still need to handle operating systems, security patches, and configurations.
- **Costs can grow fast:** If you don't manage resources well, bills can become unexpectedly high.
- **Security responsibility:** The provider secures the hardware, but you must secure the software and data.
- **Complex setup:** Requires technical skills to configure and maintain.



**Cloud Service  
Models**

**GitHub  
(SaaS/PaaS)  
Functionality**



# By Definition

GitHub is a website where people can store and share their code so others can see it, use it, or help improve it.

Think of it like Google Drive, but for code and with extra tools that let many people work on the same project without messing up each other's work.

# How to set up an account

## 1. Go to the GitHub website

Open your browser and go to <https://github.com>.

## 2. Click “Sign up”

You’ll see it in the top right corner.

Alternatively, click on Continue with Google to sign up using social login.

## 3. Enter your details

- **Username:** choose something unique; this will be part of your GitHub profile link.
- **Email address:** use one you can access.
- **Password:** make it strong and memorable.

# How to set up an account

## **4. Verify your account**

GitHub might ask you to solve a puzzle or click images to prove you're human.

## **5. Choose your plan**

**Free plan** is enough for most beginners.

## **6. Confirm your email**

GitHub will send you a confirmation link. Open your email, click the link, and your account is active.





**ANGAZA CENTER**

# Thank You!



[www.angazacenter.org](http://www.angazacenter.org)



+254 11 3630966



[ADL@angazacenter.org](mailto:ADL@angazacenter.org)