

[New Staging Africa](#)[Journeys ▾](#)[Courses ▾](#)[Job Advisor](#)[Badges](#)[About ▾](#)[Help](#)[My Dashboard](#)[English ▾](#)

Cloud Computing V2

Cloud service models

The pizza analogy

The cloud has different service models. With platform, infrastructure, and software offered as services, the pizza analogy is an easy way to understand this approach.

The pizza analogy in cloud computing can be explained as:

- You build a pizza by preparing the dough, purchasing certain toppings, heating the oven, baking it, and then serving and eating the pizza along with drinks at home.
- *Infrastructure as a Service (IaaS)*

is like buying a pre-made pizza from the supermarket. You bake it in your oven, serve it with drinks, and eat the pizza at home.

Platform as a Service (PaaS) is like ordering a pizza from a pizza delivery restaurant. The pizza is prepared by the restaurant and delivered to your front door. You provide the drinks and eat it at home.

Software as a Service (SaaS) is like going to a restaurant and eating the pizza there while enjoying the company of others and sharing the atmosphere of the restaurant.

To build an application, you must provide the infrastructure, platforms, operating systems, networking components, and so on, which is like making a pizza at home.

With IaaS, you order hardware and an infrastructure. Often, this infrastructure is managed for you. You deploy only the middleware, runtime, and your application. The infrastructure is like the pizza that is pre-made, and you bake it to your liking.




PaaS is like getting a pizza delivered. The pizza is ready to be eaten, and need only to provide drinks to go with it. In the cloud, this means that the cloud provider offers access to the platform and runtime and you only need to push the application.

SaaS is using an application that is hosted at the cloud provider, which is similar to going to a restaurant and enjoying your pizza there.

Cloud Service Models

The following table outlines the different Cloud Service Models.

The following table outlines the different Cloud Service Models.

 <p>IaaS (infrastructure as a service):</p>	 <p>PaaS (platform as a service):</p>	
---	---	---

New Staging Africa

Journeys ▾

Courses ▾

Job Advisor

Badges

About ▾

Help

My Dashboard

networking, servers, and other
 computing resources in the cloud.

English ▾

environment in which users can
 build and deliver applications. The
 provider supplies and manages the
 underlying infrastructure.

internet that are ready to be
 consumed. Users subscribe to the
 software and access it through the
 web or vendor application
 programming interfaces (APIs).

>

>Infrastructure as a Service

IaaS is a cloud computing offering in which a vendor provides users access to computing resources, such as servers, storage, and networking.

IaaS offerings are built on top of a standardized, secure, and scalable infrastructure. The virtualization of the hardware is performed by a program that is known as a hypervisor. A hypervisor manages virtual machines (VMs) or virtual servers, which hosts multiple operating system instances that are running on a specific physical machine. Each operating system appears to have the host's processor, memory, and other resources all to itself, but in reality the hypervisor is controlling and provisioning access.

Organizations use their own platforms and applications within a service provider's infrastructure.

Key features:

- *Instead of purchasing hardware, users pay for IaaS on demand.*
- *Infrastructure is scalable depending on your processing and storage needs.*

- You avoid the cost of buying and maintaining your own hardware.

English ▾

>Platform as a Service

PaaS is a cloud-computing offering that provides users with a cloud environment in which they can develop, manage, and deliver applications.

In addition to instances and other computing resources, users can use a suite of prebuilt tools and run times to develop, customize, and test their own applications.

PaaS typically entails the developer uploading the application code, or pointing to it and letting the PaaS complete the following tasks:

- 1. Obtain the runtime binary files and dependencies for the application.*
- 2. Structure their application code into the correct directory tree for containerization.*
- 3. Provision a container (or set of containers) on which the application can run.*
- 4. Automatically generate a simple and basic networking configuration for access to the application.*
- 5. Provide automatic and built-in monitoring of the application.*
- 6. You can update and redeploy the application with zero downtime.*

PaaS typically involves sacrificing some level of fine-grained control over the application's environment to gain convenience, ease of use, and rapid deployment by using a predefined deployment process. PaaS also uses external services or APIs that allow rapid composition of applications by reusing pieces of infrastructure (for example, a database) that require little to no investment in setup and configuration.

PaaS also gives the developer an automatic method for scaling. For example, consider a situation where the developer

wants more hardware resources that are dedicated to an application (scaling up or vertical scaling) or more instances of

[New Staging Africa](#)[Journeys ▾](#)[Courses ▾](#)[Job Advisor](#)[Badges](#)[About ▾](#)[Help](#)[My Dashboard](#)

Key features:

English ▾

- PaaS provides a platform with tools to test, develop, and host applications in the same environment.
- Enables organizations to focus on software development without having to worry about the underlying infrastructure.
- Providers manage security, operating systems, server software, and backups.
- Facilitates collaborative work even if teams work remotely.

>Software as a Service

SaaS is a cloud-computing offering that provides users with access to a vendor's cloud-based software. Users do not install applications on their local devices.

Instead, the applications are on a remote cloud network that is accessed through the web or an API. Through the application, users can store and analyze data and collaborate on projects.

Key features:

- SaaS vendors provide users with software and applications through a subscription model.
- Users do not have to manage, install, or upgrade software; SaaS providers manage all of those items.
- Data is secure in the cloud; equipment failure does not result in loss of data.
- Applications are accessible from almost any internet-connected device from anywhere in the world.

This figure below shows examples of the services that are available for each model.

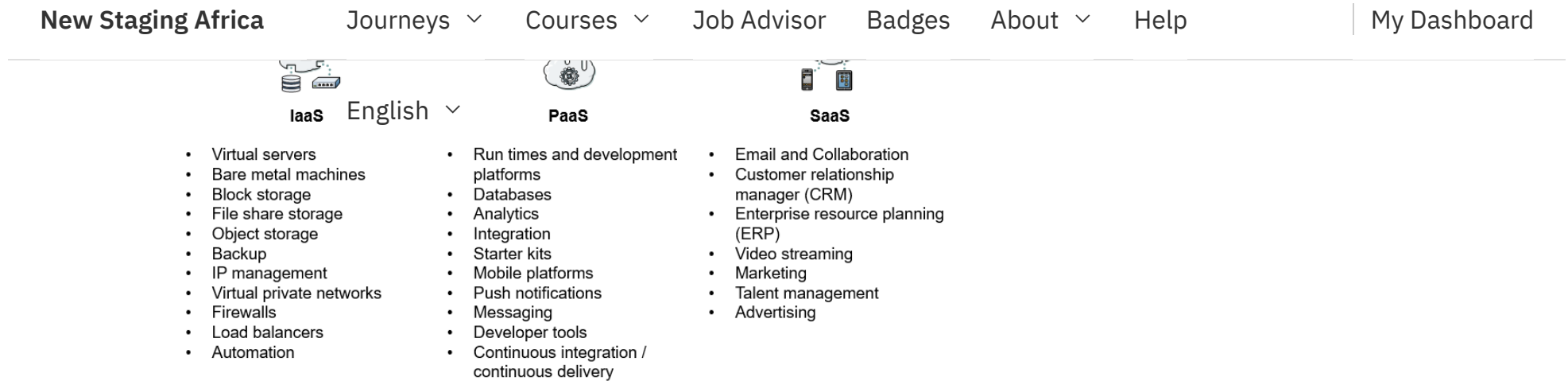


Figure 1: Examples of Cloud Services

Cloud provider and client responsibilities

This figure below shows the split between the provider and client responsibilities when dealing with on-premises or “as a service” scenarios.

Typically, the cost decreases as you move to the right in the scenarios that are shown in the figure; however, the flexibility also is reduced.

Organizations or departments within an organization make their own cost-based decisions about which delivery model to use for individual applications or projects.

Most enterprises end up using some combination of all the models that are shown in the figure.

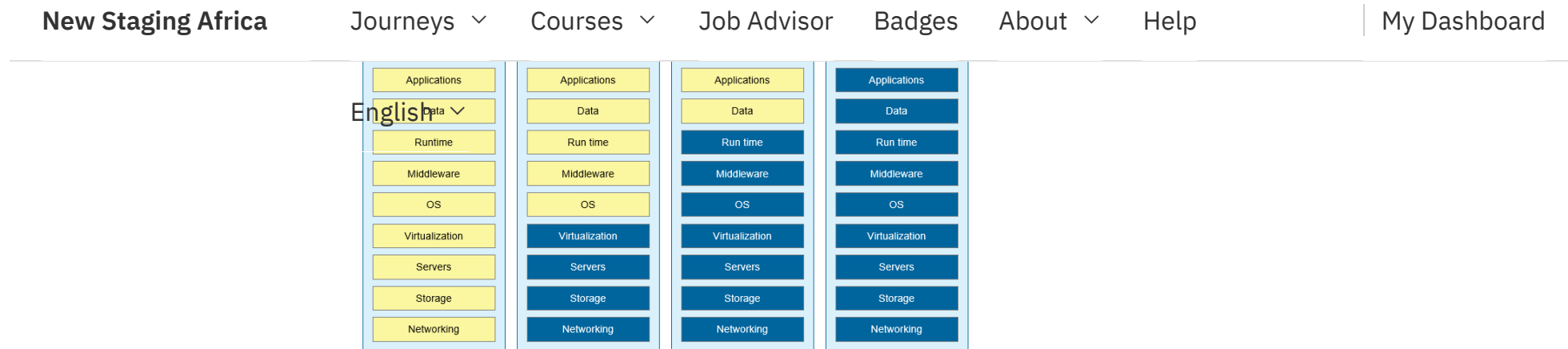


Figure 2: Cloud provider and client responsibilities

Choices when building cloud applications

When developing applications for the cloud, developers have many options to choose from in terms of platforms, frameworks, tools, and services:

1. Traditional development involves deploying applications to bare metal servers or VMs. Bare metal servers enable isolation and single tenancy, and VMs enable shared resources and multi-tenancy. When using traditional deployment, the developers are responsible for deploying and managing all the needed runtimes and dependencies for the application.
2. With containers, developers can package applications with runtimes by using a lightweight packaging medium that is called containers. Unlike VMs, which provide hardware virtualization, containers provide operating system-level virtualization.
3. PaaS solutions such as Cloud Foundry enable developers to focus on coding and pushing code to a platform that *takes* care of deployment and scaling. Developers do not need to worry about packaging or maintaining the host that runs the

applications.

New Staging Africa

Journeys ▾

Courses ▾

Job Advisor

Badges

About ▾

Help

My Dashboard

API call.

English ▾

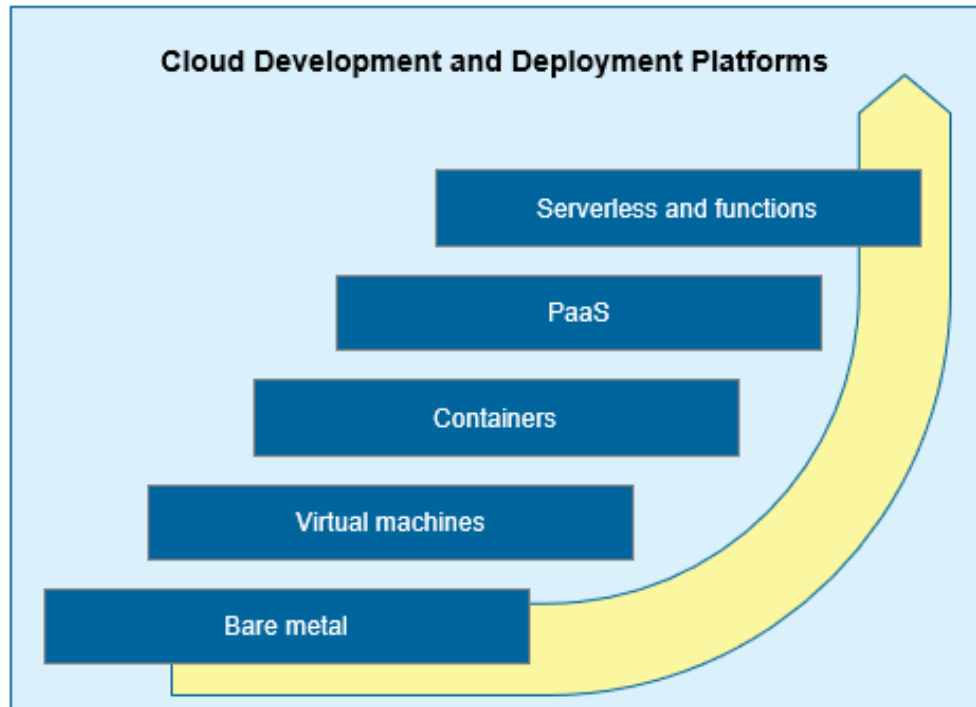


Figure 3: Cloud Development and Deployment Platforms

← Previous

Completed 17 of 21 Modules

Next →

[Contact](#) [Privacy](#) [Terms of use](#) [Accessibility](#) [Report Abuse](#) [Feedback](#) [Cookie preferences](#)

New Staging Africa

[Journeys](#) ▾

[Courses](#) ▾

[Job Advisor](#)

[Badges](#)

[About](#) ▾

[Help](#)

[My Dashboard](#)

[English](#) ▾