Describe the different categories of cloud services

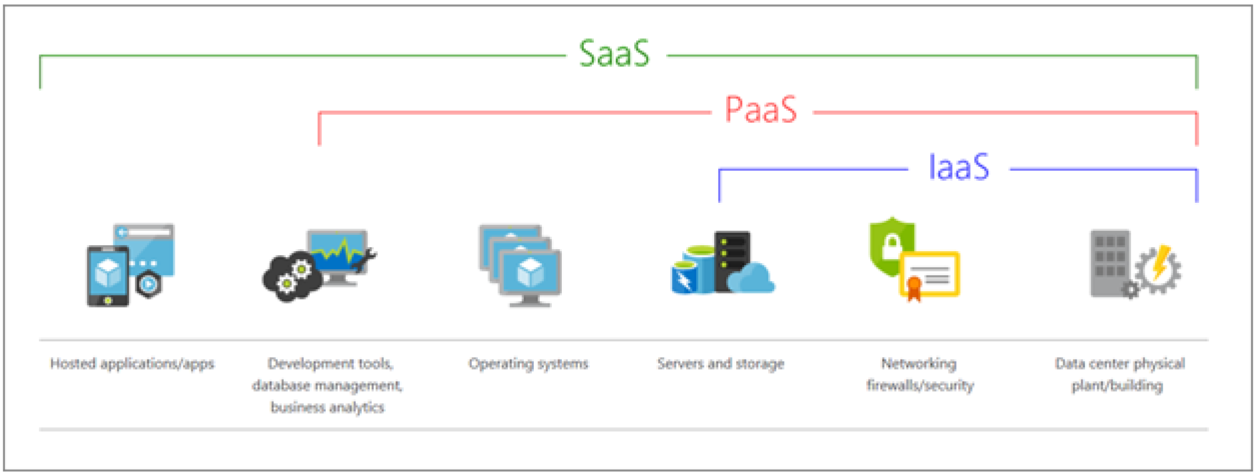
coursera.org/learn/microsoft-azure-cloud-services/supplement/GGewi/describe-the-different-categories-of-cloud-services

What are cloud service models?

If you've been around cloud computing for a while, you've probably seen the *PaaS*, *IaaS*, and *SaaS* acronyms for the different *cloud service models*. These models define the different levels of shared responsibility that a cloud provider and cloud tenant are responsible for.

Model	Definition	Description
IaaS	*Infrastructure-as-a-Service*	This cloud service model is the closest to managing physical servers; a cloud provider will keep the hardware up-to-date, but operating system maintenance and network configuration is up to you as the cloud tenant. For example, Azure virtual machines are fully operational virtual compute devices running in Microsoft datacenters. An advantage of this cloud service model is rapid deployment of new compute devices. Setting up a new virtual machine is considerably faster than procuring, installing, and configuring a physical server.
PaaS	*Platform-as-a-Service*	This cloud service model is a managed hosting environment. The cloud provider manages the virtual machines and networking resources, and the cloud tenant deploys their applications into the managed hosting environment. For example, Azure App Services provides a managed hosting environment where developers can upload their web applications, without having to worry about the physical hardware and software requirements.
SaaS	*Software-as-a-Service*	In this cloud service model, the cloud provider manages all aspects of the application environment, such as virtual machines, networking resources, data storage, and applications. The cloud tenant only needs to provide their data to the application managed by the cloud provider. For example, Microsoft Office 365 provides a fully working version of Microsoft Office that runs in the cloud. All you need to do is create your content, and Office 365 takes care of everything else.

The following illustration demonstrates the services that might run in each of the cloud service models:


SaaS
PaaS
IaaS
Hosted applications/apps
Development tools, database management, business analytics
Operating systems
Servers and storage
Networking firewalls/security
Data center physical plant/building

IaaS

IaaS is the most flexible category of cloud services. It aims to give you complete control over the hardware that runs your application. Instead of buying hardware, with IaaS, you rent it.

Advantages:

- **No CapEx**. Users have no up-front costs.
- **Agility**. Applications can be made accessible quickly and deprovisioned whenever needed.
- **Management**. The shared responsibility model applies; the user manages and maintains the services they have provisioned, and the cloud provider manages and maintains the cloud infrastructure.
- **Consumption-based model**. Organizations pay only for what they use and operate under an Operational Expenditure (OpEx) model.
- **Skills**. No deep technical skills are required to deploy, use, and gain the benefits of a public cloud. Organizations can use the skills and expertise of the cloud provider to ensure workloads are secure, safe, and highly available.
- **Cloud benefits**. Organizations can use the skills and expertise of the cloud provider to ensure workloads are made secure and highly available.
- **Flexibility**. IaaS is the most flexible cloud service because you have control to configure and manage the hardware running your application.

PaaS

Advantages:

- **No CapEx**. Users have no up-front costs.

- **Agility**. PaaS is more agile than IaaS, and users don't need to configure servers for running applications.
- **Consumption-based model**. Users pay only for what they use, and operate under an OpEx model.
- **Skills**. No deep technical skills are required to deploy, use, and gain the benefits of PaaS.
- **Cloud benefits**. Users can take advantage of the skills and expertise of the cloud provider to ensure that their workloads are made secure and highly available. In addition, users can gain access to more cutting-edge development tools. They can then apply these tools across an application's lifecycle.
- **Productivity**. Users can focus on application development only, because the cloud provider handles all platform management. Working with distributed teams as services is easier because the platform is accessed over the internet. You can make the platform available globally more easily.
- **Disadvantage**
- **Platform limitations**. There can be some limitations to a cloud platform that might affect how an application runs. When you're evaluating which PaaS platform is best suited for a workload, be sure to consider any limitations in this area.

SaaS

Advantages:

- **No CapEx**. Users have no up-front costs.
- **Agility**. Users can provide staff with access to the latest software quickly and easily.
- **Pay-as-you-go pricing model**. Users pay for the software they use on a subscription model, typically monthly or yearly, regardless of how much they use the software.
- **Skills**. No deep technical skills are required to deploy, use, and gain the benefits of SaaS.
- **Flexibility**. Users can access the same application data from anywhere.
- **Disadvantage**

- **Software limitations**. There can be some limitations to a software application that might affect how users work. Because you're using as-is software, you don't have direct control of features. When you're evaluating which SaaS platform is best suited for a workload, be sure to consider any business needs and software limitations.

What is serverless computing?

Like PaaS, *serverless computing* enables developers to build applications faster by eliminating the need for them to manage infrastructure. With serverless applications, the cloud service provider automatically provisions, scales, and manages the infrastructure required to run the code. Serverless architectures are highly scalable and event-driven, only using resources when a specific function or trigger occurs.

It's important to note that servers are still running the code. The "serverless" name comes from the fact that the tasks associated with infrastructure provisioning and management are invisible to the developer. This approach enables developers to increase their focus on the business logic, and deliver more value to the core of the business. Serverless computing helps teams increase their productivity and bring products to market faster, and it allows organizations to better optimize resources and stay focused on innovation.

✓

Completed