

Cloud Models

According to **shared responsibility model**, the cloud offers a few new concepts, namely, Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS), and as a bonus, serverless, or Function as a Service (FaaS), a term used these days to describe fully serverless environments.

IaaS provides the most flexible cloud computing model and enables you to retain the most control over your infrastructure. Google Compute Engine is an example. With an IaaS solution, you can deploy virtual machine environments onto servers, giving you full ownership of the infrastructure end-to-end, without having to manage the physical servers yourself. IaaS does have limitations as well; for example, you need more overhead to manage your resources, since you retain a lot of control. That inherently means you have a lot more responsibility for the security of your environments.

The **PaaS** model offers a simple, cost-effective solution to developing and deploying applications on a scalable and highly available platform. PaaS typically offers development teams a lot more speed for application deployment, and because it typically ramps up and down based on usage, a PaaS solution can be more cost-effective. Google App Engine is an example; however, while developer expertise has evolved, App Engine is no longer the default model, and the current trend for consuming cloud computing by application developers is to use Google Kubernetes Engine (GKE), which falls somewhere in the middle between PaaS and IaaS. Some of the limitations of PaaS concern data security, since the cloud service provider controls the underlying infrastructure; vendor lock-in (although this is not an issue with GKE); operational limitations; and a lack of full developer flexibility.

The **SaaS** model is the most familiar to everyone, in which an application is delivered over the Internet through a web browser, without the need to download or install anything on the client side. SaaS solutions are advantageous for software that is designed to perform a general set of tasks and to disallow a developer end user to customize or modify the application. Examples of SaaS are Salesforce, Intuit QuickBooks, and Google Workspace. There are several limitations to SaaS: these solutions are designed to solve only certain use cases and are not designed as solutions that enable developers the freedom to build. Google Drive, for example, is a SaaS solution designed purely for file storage on the Internet. End users don't have much control here beyond the service catalog of options Google Drive provides. Major implications of SaaS are vendor lock-in, lack of interoperability, lack of control and customization, and concerns about data security.

Taken from [Google Cloud Certified Book](#)