

The Three Service Models of Cloud Computing

NIST defines three service models for cloud computing:

1. Infrastructure as a Service (IaaS)
2. Platform as a Service (PaaS)
3. Software as a Service (SaaS)

The basic thing that separates these three service models is “who’s responsible for what?”. As you can see in Figure 1, the further “up” the stack you go, the greater the responsibility assumed by the cloud service provider.

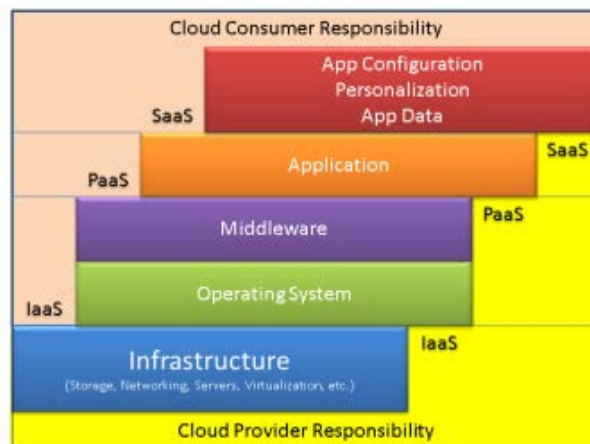


Figure 1: Cloud Service Models

What this Figure 1 tells us is that for:

- ♦ **Infrastructure as a Service (IaaS)**, the Cloud Provider is responsible for everything up to, but not including, the operating system and the Cloud Consumer is responsible for everything from the operating system up through the rest of the stack.
- ♦ **Platform as a Service (PaaS)**, the Cloud Provider is responsible for everything up through the middleware & development environment and the Cloud Consumer is responsible for everything from the application up through the rest of the stack.

- ♦ **Software as a Service (SaaS)**, the Cloud Provider is responsible for everything up through the application and the Cloud Consumer is responsible for the application configuration, personalization, and application data.

One thing to notice here is that, in all cases, the Cloud Provider is responsible for the physical infrastructure (which will likely include a hypervisor to support the creation of virtual machines), and the Cloud Consumer is responsible for application configuration, personalization, and data. You'll also notice in the diagram that, from the Cloud Provider perspective, SaaS "sits on" PaaS which sits on IaaS. I want to note that this is certainly a viable configuration (i.e. a PaaS Cloud Provider could deploy their PaaS capability onto their own – or someone else's – IaaS infrastructure & SaaS can run on PaaS), but it's not the way it has to be. In fact, that's not usually the optimal way to run PaaS or SaaS. The reason for that is because an IaaS infrastructure is optimized to run a wide range of generic workloads. If you know in advance what the characteristics of your workload are (which, if you're deploying PaaS or SaaS, you have a pretty good idea), you can tune your infrastructure to best meet the demands of that workload. For example, if you were offering a SaaS logging service, you would want to optimize your storage infrastructure for streaming writes, whereas if you were offering a database as a part of your PaaS offering, you would want to optimize your storage based on the I/O patterns specific to the database system you were using. In any event, simply understand that the models *can* but don't *have to* be stacked on top of one another.

Another way to conceptualize the three models is to look at the likely consumers of each. They really are targeted at very different users!

- ♦ **Infrastructure as a Service (IaaS)**, the typical consumer for an IaaS offering is a system administrator. If you refer back to Figure 1, you'll see that this makes sense. With IaaS, the *consumer* is responsible for the operating system and everything above it. This is very much the way that many data centers operate in a "traditional" IT shop, with the "data center" team providing ping, power, and pipe to the system administrator. This is also one of the reasons that I view IaaS pretty much as a better way of doing what we've been doing for the last 20 years!
- ♦ **Platform as a Service (PaaS)**, with PaaS, the target audience is either the application developer or the application hoster. Platform as a Service implies that the Cloud Provider is responsible for everything up to the application environment. This includes middleware (such as JBOSS, Spring, STRUTS, the .Net framework) and all the app developer has to do is start writing code – the hoster simply installs the app and turns around and offers it to his customers as PaaS. Since the Cloud Provider manages all the underlying components, the developer doesn't have to worry about whether everything is configured correctly – it was built using the same automated system the other instances in the cloud were built with.

- Software as a Service (SaaS)**, the Cloud Provider is responsible for everything up through the application and the Cloud Consumer is responsible for the application configuration, personalization, and application data. SaaS is the “holy grail” of cloud computing because it allows the business consumer to focus on the application, which is where their business processes and everything that makes them unique lives. It also allows the Cloud Provider to focus on the infrastructure and compete with other Cloud Providers based on price and performance.
- The target consumer for each service model, along with the Cloud Provider responsibilities is summarized in

SERVICE	PROVIDER	TARGET CONSUMER
IaaS	Physical Infrastructure Hypervisor	System Administrator
PaaS	Physical Infrastructure Hypervisor Operating System Middleware / Dev Stack	Application Developer / Host
SaaS	Physical Infrastructure Hypervisor Operating System Middleware Application	Application Administrator / End User

Table 1: Target Consumer by Service Model

That pretty much wraps up the NIST view of the three different service models. And now, just when you begin to think you’re getting a handle on all this, Gartner has to go throw a wrench in the works. Apparently, they aren’t content with the NIST definitions, so they came up with the following “additions” to the IaaS category:

- Self-managed IaaS, for cost-effective agile replacement of traditional data center infrastructure.*
- Lightly managed IaaS, for customers who wish to primarily self-manage but want the provider to be responsible for routine operations tasks.*
- Complex managed hosting, for customers who want to outsource operational responsibility for the infrastructure underlying Web content and applications.*

**Gartner Magic Quadrant for Cloud Infrastructure as a Service and Web Hosting

According to these definitions, the NIST incarnation of IaaS is what Gartner refers to as “Self-managed IaaS”. The “Complex managed hosting” sounds an awfully lot like the NIST version of PaaS.

Source : <http://kensvirtualreality.wordpress.com/2012/10/09/cloud-101-the-three-service-models/>