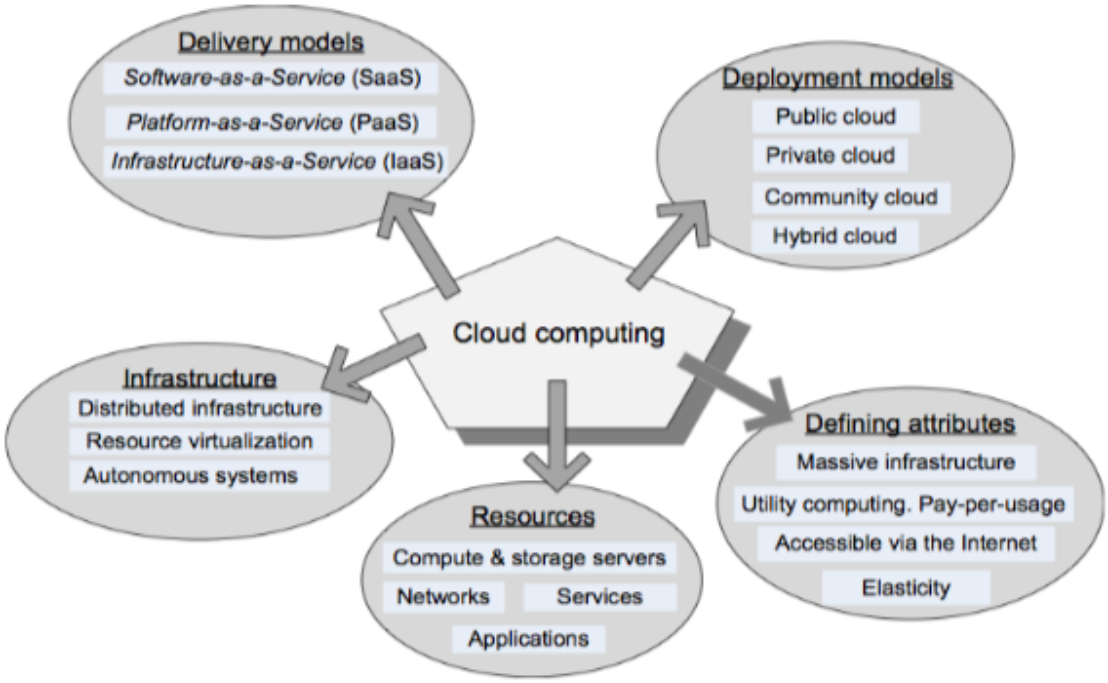
e-PG Pathshala
Subject: Computer Science
Paper: Cloud Computing
Module 4: Delivery Models
Module No: CS/CC/4
Quadrant 1 —e-text

1. Introduction

Cloud computing is a service oriented architecture. The term service in the world of software indicates endpoint of a connection. And, the connection should be offered in an environment supported by the underlying computing system models. In our case, the underlying system is our 'pool of resources', which is an essential characteristic of cloud computing, as we learnt in the last module. Also, as part of the characteristics, we learnt that Internet is an essential component of cloud. Now, the 'connection' here is to be offered through the Internet and the endpoint of this connection is the service. Hence, in cloud, all and any resource that is provided for the customer is a service. Thusstorage, processing, bandwidth, and active user accounts etc. are all offered as a service.Cloud is an environment through which these services (i.e., the pooled resources)are offered in a pay-as-you-go model accessible over the Internet. It is important to understand how these services are delivered to the consumers. In cloud, the mechanism of delivery of a serviceis dependent on the type of the service being delivered. Thus, there can be as many delivery models as there are different service offerings in the cloud. However, NIST (National Institute of Standards and Technology) originally had definedthree delivery models and we will study these here.

Figure 1 below shows the overview of cloud. In earlier modules we learnt the types of resources, Infrastructure, defining attributes, and deployment models. In this module, we'll discuss about the delivery models in cloud. This will complete the discussion on the various components or parts of the overview of cloud computing components.



Courtesy: Dan C. Marinescue [1]

Figure 1: Overview of Cloud Computing

2. Learning Outcome

A service in general is a transaction between a provider, who will offer the service and a consumer, who will use the service. Since cloud is about software service, the delivery models are typically described from the perspectives of the service providers and the service consumers. In case of public cloud the service providers are the cloud owners, who understandably have built the infrastructures in the cloud stack and are offering the facilities of these through one of the many delivery models possible. In case of private, hybrid or even community cloud, these are the people who are helping the organization(s) to create and offer the services. In either case, there are service providers and service consumers. In this module, we will discuss the benefits of the delivery models from the perspective of both the providers as well as the consumers.

At the end of this module, students will be able to:

1. Understand and appreciate what is Software-as-a-Service.
2. Understand and appreciate what is Platform-as-a-Service.
3. Understand and appreciate what is Infrastructure-as-a-Service.

3. Cloud Stack

Figure 2 below shows the cloud stack, which contains all the layers available in cloud. Delivery models allow one or many of these layers to be controlled by a consumer.

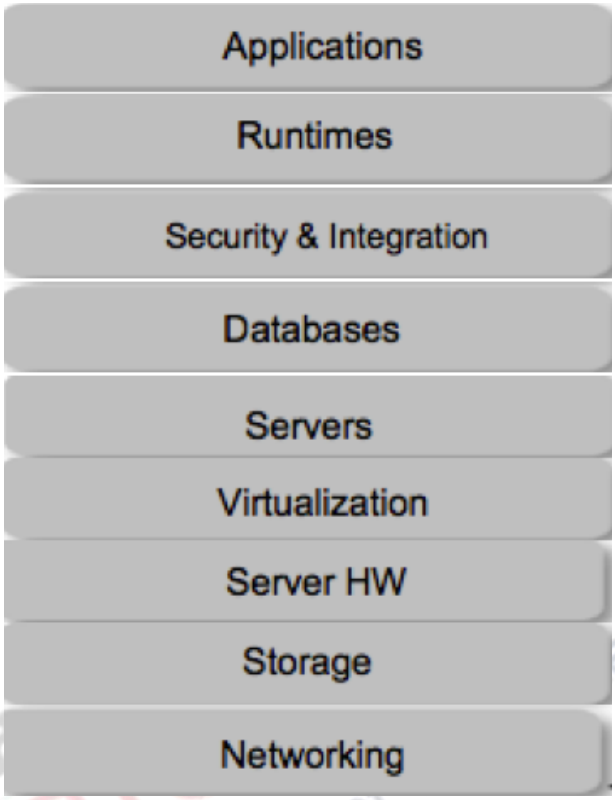


Figure 2: Cloud Stack

As is expected, we find that the application is the uppermost layer and, in many cases, this is the primary concern of the users. Runtime is the environment where the application is deployed. Next comes the security and integration. Databases and the servers are next where the data will be stored and the necessary processes will run, while the hardware components of these two software layers, viz., server hardware and storage, are under the virtualization layer that keeps the hardware hidden away from the users.

Details of all the layers of the stack need not be understood at this point in time, which we'll learn in later modules. it is enough to understand only a few concepts at this point of the course.The application is always run at the top and the whole stack is dependent on the underlying server and storage hardware and these are connected using the networking. These are the major assets of a cloud owner. Delivery models will be explained using the stack.

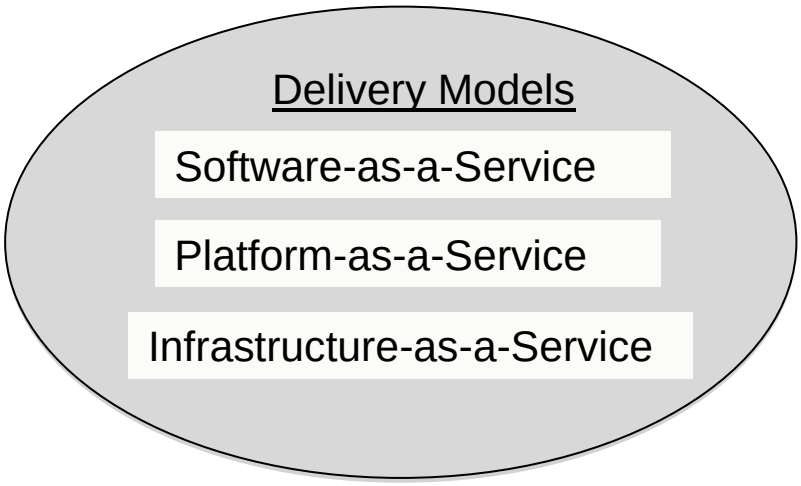
4. Delivery Models

As per NIST, there are three possible delivery models. These are:

1. Software-as-a-Service or SaaS
2. Platform-as-a-Service or PaaS

3. Infrastructure-as-a-Serviceor IaaS

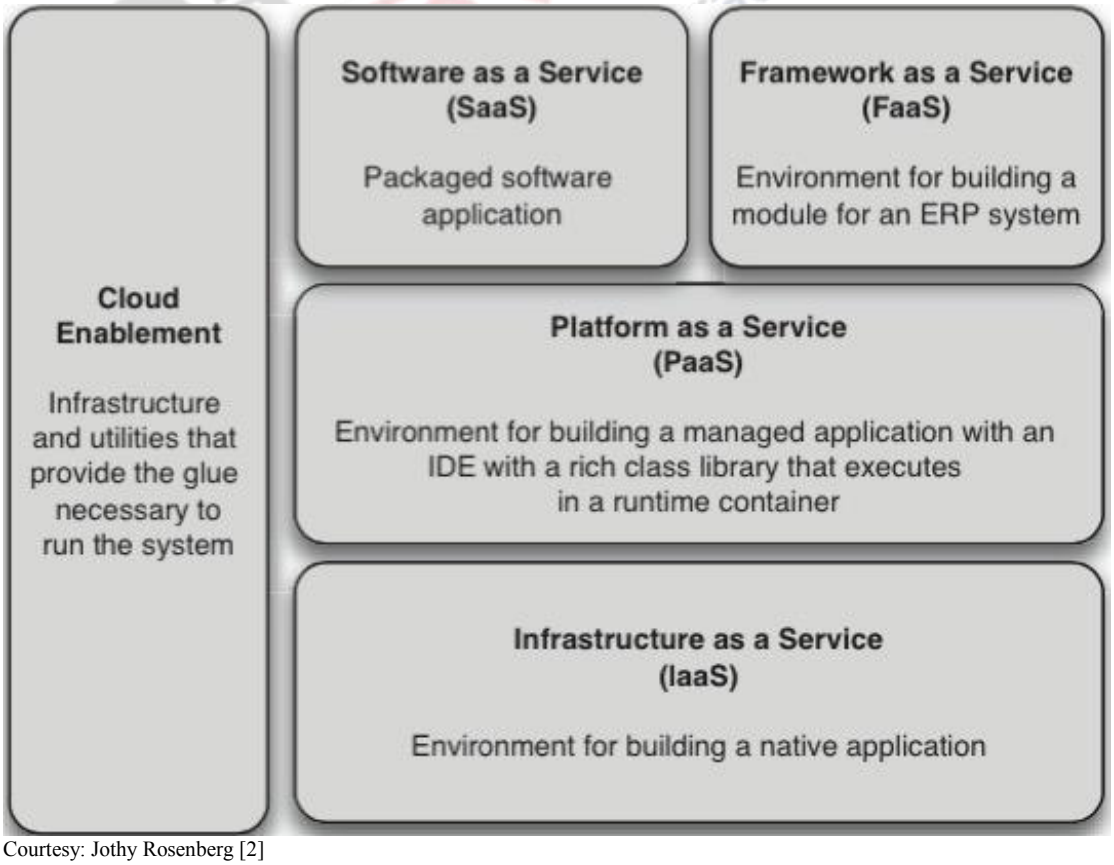
This is shown in Figure 3 below.



Courtesy: Dan C. Marinescue [1]

Figure 3: Delivery Models

Figure 4 below shows the layers of the delivery models. This explains the abstraction of the different delivery models from the perspective of what is available in which model.



Courtesy: Jothy Rosenberg [2]

Figure 4: Models in Layered Architecture

Software-as-a-Service is at the top along with Framework-as-a-Service, while Infrastructure-as-a-Service is at the bottom and Platform-as-a-Service is sandwiched in the middle. This explains the pre-packaging that is offered at each delivery model. The flexibility of the users using the models increases as we go down the layers, with maximum flexibility provided to the Infrastructure-as-a-Service users and minimum flexibility offered to the Software-as-a-Service and the Framework-as-a-Service users. Now, let us take a look at the details of these three models.

4.1. Software-as-a-Service or SaaS

According to NIST, Software-as-a-Service (SaaS) is defined as follows:

> "The capability provided to the consumer is to use the provider's applications running on a cloud infrastructure. The applications are accessible from various client devices through either a thin client interface, such as a web browser (e.g., web-based email), or a program interface. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited userspecific application configuration settings."

Software-as-a-Service (SaaS) refers to services and applications that are available on an on-demand basis. Itenables the consumer with the capability to use the provider's applications running on a cloud infrastructure. Consumers can easily access the applications running in the cloud using any device at their ends. Providers may provide a web browser based API (e.g., web-based email). Alternatively, a program interface is provided by the provider using which the consumer can access the offered applications. SaaS is a rather old concept, it being in existence long before the concept of cloud came. However, SaaS gained momentum when it was offered through cloud.

SaaS offers software as it is and does not have an opportunity of customization based on the users requirement. In this perspective, Framework-as-a-Service (FaaS) is an environment that is very closely related to SaaS bringing in the required flexibility in SaaS offering. Users using FaaSare allowed to extend existing functionalities of the SaaS applications. For example, the services provided by the Salesforce.com are SaaS and those provided by Force.com areFaaS that extends the Salesforce.com SaaS offerings. It is, sometimes, not sufficient to use specific application in the given form, at the same time it may be too time-consuming to write the whole application. It is in such a scenario that FaaS works. It is possible to enhance the capabilities of the base SaaS system using FaaS offering. Customized and specialized applications may be created out of the general-purpose SaaS applications suitable specifically for a specific organizationthat can be made available to any SaaS customer. However, the restrictions on specific languages and APIs as provided by the FaaS environment must be followed.

SaaSmay be considered to be the most advanced version of all the services provided by cloud service providers in the sense that the specific operating system, the software required and of course the needed applications, all have to be supplied and maintained by the service provider, along with the versions and editions of the applications etc.Applications are hosted in the servers of the service provider. On the other hand, in SaaS, the responsibility of the consumer is the least.

Let us consider the example of an organization ABC that uses Microsoft Excel on a regular basis. ABC has offices in various cities in India and the employees of the organization often need to exchange information to collaborate on projects that are handled by various employees located in different cities. When the office located in Kolkata decided to upgrade the version of MS Excel, if all the other offices also do not do the same, there will be a versioning problem when next time the employees in Mumbai office tries to merge an Excel file with another sent by a colleague in Kolkata office. However, if all the employees in all the offices would use a cloud service offering the service over the Internet, all the employees would use the same version and hence no problem would be there.

The above example shows that the traditional desktop applications such as word processing or spreadsheet, when delivered to customers over the Internet by a third party, reduces the burden of maintaining and upgrading the software, freeing the consumer to spend the time on more important aspect of the business.

From the perspective of the consumer, the SaaS model enjoys the following benefits:

- No infrastructure cost
- No upfront cost of buying any software
- Administration is simple and easy
- All the updates are taken care of by the service provider
- Performance guarantee by the provider
- In case multiple users are collaborating, versions would not be a problem hence, collaboration is easy
- Since it is hosted in cloud, it is accessible from anywhere through the Internet.
- Rapid implementation
- Standalone and configurable applications
- Subscription and pay-as-you-go (PAYG) pricing

However, the service providers are burdened with responsibilities. The providers' concerns are the following:

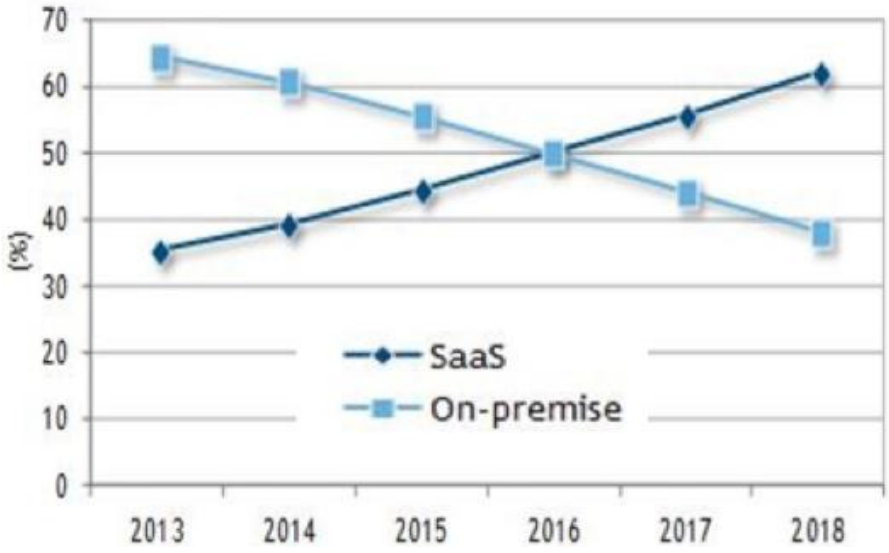
- Installation, configuration and maintenance of all servers (web server, mail server, database server etc.),
- Purchase and maintenance of all the required software in the correct version.
- Integration of the software (if required),
- Patch management, since often software need to be patched due to bugs or problems found in them,
- Monitoring usage and billing,
- Ensuring security,
- And many more.

However, in spite of the ease of use from the point of view of users, all situations are not suitable for SaaS. Typically, there are a few types of applications that are suitable candidate to be used as SaaS model. Suitability of SaaS is limited to the followings:

- When many users use the same product, such as email.
- For applications where significant peak in demand comes only periodically, such as billing and payroll.
- If there is a need for Web or mobile access, such as mobile sales management software.

- The overall need for the application is only short-term, such as collaborative software for a project.

The most important contribution of SaaS is that it allows users to be free of the worries of buying, installing and maintaining the required software, along with the need to keep track of any new version or a patch that may be I the market.



Courtesy: IDC [3]

Figure 5: SaaS versus On-premise Software

Thus, having on-premise software can be eliminated or reduced drastically by SaaS. More and more customers are understanding this advantage and using cloud based SaaS for their requirements. Forbes reports that IDC has predicted that the growth of SaaS would overtake the on-premise software acquisition, as shown in Figure 5 above.

Now, we turn our attention to the next delivery model, Platform-as-a-Service or PaaS.

4.2. Platform-as-a-Service or PaaS

According to NIST, Platform-as-a-Service (PaaS) is as follows:

> "The capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages, libraries, services, and tools supported by the provider.3 The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly configuration settings for the application-hosting environment."

PaaS provides the users with an environment where an application can be developed and deployed, saving developers from the complexities of the infrastructure side. This allows the developers concentrate only on the development of the application thereby improving the speed of development, and allowing the consumer to focus on the application itself. In a PaaS model, the infrastructure such

as the servers, databases, networking, operating system etc. are taken care of by the PaaS provider, freeing the application developers from many worries. However, the language or the platform for the application development must be the ones supported by the service provider and offered through predefined interfaces. Therefore, if the environment required is different from the ones being offered by a certain vendor, the developer is forced to move out and look for other vendors who can provide the required environment. The consumer would consume and pay for the CPU, bandwidth, storage and any such items that may be used during the development and deployment of the application. However, in this model there is no scope for any direct interaction or managing of the underlying infrastructure, which is still managed by the service provider. Issues such as how and where to deploy the applications in terms of the underlying infrastructure, issues related to the operating system, how to provision the underlying resources (both hardware as well as software) and how or when to configure the supporting technologies such as load balancers and databases are all under the strict control of the service providers. Developer's job is to design, implement and deploy the application through the use of the interfaces offered as part of PaaS model in the form of a Web-based interface or in the form of programming APIs and libraries.

PaaS consumers are not to directly interact with or administer the virtual environment deployed in the cloud. Instead, they are to concentrate specifically on writing the application. While this brings the benefit of simplification of management and handling of various resources, it also comes at the cost of less flexibility. Further cause of concern maybe from the fact that it is required for a PaaS consumer to necessarily use the languages/platforms supported by the service provider.

Having said that, in a PaaS model, installation and configuring of the servers, operating systems, and databases are all done by the service providers. They are also responsible for obtaining the required licenses for any software being used for the development of the application, althoughunlike in a SaaS model, the consumer here bears some burden. These include the setting up of the applications and operations needed for the development and deployment.

Using PaaS is highly beneficial for the consumer since they have a reasonably flexible and scalable service while being free from the trouble of acquiring all the necessary resources. Flexibility also comes from the fact that, while being under the strict supervision of the service provider, the platform can be adjusted according to the needs of the consumer. For example, if a company using the PaaS model of a service provider suddenly observessomething that could cause a peak load on its client website, it can easily and quickly increase the technological resources (since cloud is elastic, there is no problem to acquire more resources). This can be done seamlessly, without any interruption or impact to its customers.

According to authors of the book "Cloud Computing, A Practical Approach" [5], the following are the characteristics of PaaS:

- *Runtime framework.* This is environment for coding and is the most fundamental offering of PaaS model. Since consumers' requirement is to develop and deploy, the development environment is of utmost importance and this is provided through runtime framework. The runtime framework executes end-user code followingto the policies set by the user and the provider.
- *Abstraction*. The level of abstraction is a way to recognize the involvement of the consumer in the cloud. While PaaS models have lesser levels of

abstraction than its counterpart in SaaS, it enjoys more abstraction when compared to the other model, IaaS since it has more interactions and involvement in the system than SaaS and less in comparison to IaaS.

- Automation. Calculating the percentage of the infrastructural resources needed by the developed application to be appropriately deployed and running in cloud is an automated process in PaaS model. This means that the developer need not worry about the requirement of the infrastructure for the deployment of the software being developed using PaaS model. The amount of requirement is judged automatically and the provisioning for resources are done. This, of course, is at a cost to the consumer and are all part of the service level agreement between the provider and the consumer before the process of consuming starts.
- Cloud services. This again is an essential part of the whole mechanism where everything is being offered as a service. These services generally include specific components for developing applications, advanced services for application monitoring, management, and reporting etc.
-

Perhaps Google AppEngine is the most popular product in PaaS category. It provides a scalable runtime based on different programming languages supported by it. It also offers additional APIs and components provided to the developer that offer further scalability.

The following are the benefits that a PaaS consumer enjoys:

- Lower upfront and operational costs
- No investment in terms of hardware and software maintenance
- Greater and faster scalability
- Improved performance due to access to better systems in cloud
- Integration is seamless
- Development is faster due to easy access to improved environment
- Server-side scripting environment
- Database management system
- Access to any amount of storage

4.3.Infrastructure-as-a-Service or IaaS

According to NIST, Infrastructure-as-a-Service (IaaS) is defined as follows:

> "The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, and deployed applications; and possibly limited control of select networking components (e.g., host firewalls)."

IaaS provides great flexibility and control over the cloud resources being consumed,

but typically more work is required of the developer to operate effectively in the environment.

The 'infrastructure' in the term infrastructure-as-a-Service (IaaS) is the use of infrastructure or virtual machines on demand. An IaaS provider supplies virtual machine images of different operating system flavors. The developers can run any custom or packaged applications by tailoring these images. The applications can run natively on the chosen OS. The service provider is responsible for the network elements, servers and other hardware required. Here the level of abstraction is far less when compared with the earlier two models, viz., SaaS and PaaS, since the consumer is responsible for the installation and operation of its operating system and all the required applications. Thus, IaaS delivers customizable infrastructure on demand. As part of IaaS offering, it is possible to obtain servers as well as network devices, load balancers, and database and Web servers.Storage and bandwidth are also consumable commodities in an IaaS environment, with storage typically charged per gigabyte per month and bandwidth charged for transit into and out of the system. Use of these is typically metered and charged in hour-long increments.

IaaS is good for workload partitioning, application isolation, sandboxing, and hardware tuning. The providers of IaaS model allow consumers to deploy and access virtual machines (VM) with requisite capabilities and use these VMs as required. Therefore, it is an improved utilization of the IT infrastructure.IaaS provides a secured environment where third party applications can be safely executed and cannot cause harm to the rest of the environment. From the perspective of the customer using IaaS allows them to take advantage of the full customization offered by the cloud to deploy their infrastructure without incurring the cost of installation and maintenance of all the underlying hardware.

While PaaS models offer a way to deploy and manage applications on the cloud, IaaS offers a similar scenario in virtual machines on top of which the IT infrastructure is built and configured.Also, in PaaS, the service provider allocates more resources in case the deployed application needs it and the agreement between the provider and consumer supports such an action. However, the same is not true for a IaaS solution. Here, the consumer has to make an estimate of the CPU, memory etc. for the VMs and these only will be accessible. IaaS solutions, only provide ways to provision more resources.

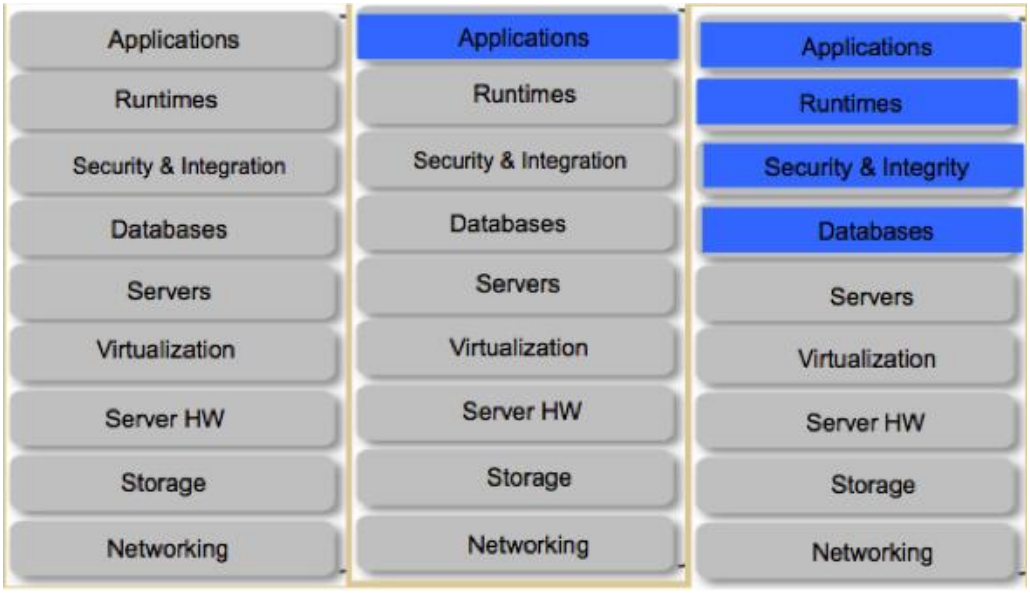
Benefits of the IaaS model include:

- No initial hardware investment
- Highly Scalability
- Pay per Use
- Location independence
- Physical security of data center locations for servers

5. Comparison of the Delivery Models

In SaaS users possess the capability using applications provided by the cloud. However, SaaS does not allowthe consumer any level of control over the deployed applications. Nor is there any control of the user over other resources being used such as the platform or the infrastructure. *PaaS*, on the other hand,*enables users* to develop their own applications. PaaS also provides any necessary tools or language

support enabling users to deploy the applications in the cloud infrastructure. The most flexibility is however provided by *IaaS.*It not only allows the user use the virtualized resources, users are able to run new operating system or any arbitrary software. Figure 6 below shows the ownership of the three delivery models. 6(a) represents the SaaS solution, (b) represents the PaaS solution and (c) represents the IaaS solution.



(a) **(b)** **(c)**

Figure 6: Relative ownership of the cloud stack

The blue blocks in the figures 6(a), 6(b) and 6(c) indicate the control of the consumer while the gray ones indicate the control of the providers. As can be seen from the figure, In SaaS, all the control lie with the service provider and nothing of the stack is controlled by the consumer. As shown in (b) above, in PaaS, while provider controls all the components right upto and including the Runtime, the consumer owns the applications. This scenario changes quite a bit in the IaaS model as shown in the (c) part of the figure above. Here, the consumer controls upto Databases, while the provider controls the rest of the stack.

A user of IaaS is operating at the lowest level of granularity available and with the least amount of prepackaged functionality.A PaaS user operates at a higher level compared to the IaaS user and SaaS user operates at the top-most level where everything comes prepackaged.

Table 1 below summarizes the various aspects of the three models:

Category	Characteristics	Product Type	Vendors/Products
SaaS	Customers are provided with applications that are accessible anytime and from anywhere.	Web applications and services (Web 2.0)	SalesForce.com (CRM) Clarizon.com (project management)

			Google Apps
PaaS	Customers are provided with a platform for developing applications hosted in the cloud.	Programming APIs and frameworks Deployment systems	Google AppEngine Microsoft Azure Manjrasoft Aneka Data Synapse
IaaS/ HaaS	Customers are provided with virtualized hardware and storage on top of which they can build their infrastructure.	Virtual machine management infrastructure Storage management Network management	Amazon EC2 and S3 GoGrid Nirvanix

Now, let us now take a look at the growth rate of these three delivery models. Figure 7below shows how the user ultimately uses the various services. They have used the measure CAGR or the **compound annual growth rate.** It is a useful measure of growth over multiple time periods. It is a way of measuring the growth rate using compounding mechanism over the time period.From the figure it is clear that initially in 2013, IaaS has the majority workload share, but by 2015 SaaS workloads take the majority share, and by 2018 will have 59 percent share of all cloud workloads. PaaS will have the second-fastest growth, although it will lose the share of total cloud workloads from 15 percent in 2013 to 13 percent by 2018.

Courtesey: Cisco Global Cloud Index 2013-2018[4]

Figure 7: Deployment percentage of the three models

6. Summary

In this module, we have looked at the three major service delivery models used to deploy services in cloud. These are SaaS or Software as a Service, PaaS or Platform as a Service and IaaS or Infrastructure as a Service. While SaaS is used for users that wish to use an application in a provided version, PaaS is used for the development of an application in a customized manner and deploy. IaaS, on the other hand allows the use of the infrastructure of the service provider and create platforms as well as applications in a customized environment. IaaS users enjoy the most from the perspective of flexibility and control, with PaaS users coming as the second while the SaaS users do not enjoy any. However, SaaS users need to be responsible only for their data and need to pay for the rest, while PaaS and definitely the IaaS users shoulder greater responsibility in the cloud environment.

References

1. Dan C. Marinescu, “Cloud Computing, Theory and Practice”, 1st Edition, Morgan Kaufmann, 2013.
2. Jothy Rosenberg and Arthur Mateos, “The Cloud at Your Service*The When, How, and Why of Enterprise Cloud Computing*”, Manning Publication, 2010.
3. http://www.forbes.com/sites/louiscolumbus/2014/12/20/idc-predicts-saas-enterprise-applications-will-be-a-50-8b-market-by-2018/
4. http://www.cisco.com/c/en/us/solutions/collateral/service-provider/global-cloud-index-gci/Cloud_Index_White_Paper.pdf
5. Toby Velte, AnthoniVelte and R. C. Elsenpeter, “Cloud Computing, A Practical Approach”, McGraw Hill Publication, 2009.
6. RajkumarBuyya, James Broberg, A. M. Goscinski, “Cloud Computing: Principles and Paradigms”, Wiley Publications, 2011.
7. RajkumarBuyya, Christian Vecchiola, S. ThamaraiSelvi, “Mastering Cloud Computing Foundations and Applications Programming”, Elsevier Morgan Kaufmann, 2013.