MARYAM UNIVERSITY

Cloud Computing

SUPERVISED BY
Zahid Yaad

SUBMITTED BY
Hekmatullah Himmat

Roll No: 00,4824

Department of Information Technology

Maryam University

Session 2018-2022

APPROVAL SHEET

This is certifying that this thesis, which submitted by the partial of performance of the requirements for the award of BIT (Bachelor in Information Technology) degree has been, approved the supervisory committee.

Supervisor:

Mr. Zahid Yaad

Signature: ______________________________

External Examiner:

Signature: ______________________________

Chancellor:

Dr. Shahla Rasheed

Maryam University

Signature: ______________________________

ACKNOWLEDGEMENT

In the Name of Allah, the Most Gracious and the most Merciful, all praises to Allah for the Strengths and His blessing in completing this thesis. I have been privileged learn theories and concepts which would have been impossible if I had not comprehensively carried out the needed research. I am thankful to a number of people who helped to bring this research to fruition.

Special appreciation goes to my supervisor **Mr. Zahid Yaad** for his excellent guidance, caring, patience, and providing me with an excellent atmosphere for doing research. My supervisor was there throughout my preparation of the proposal and the conceptualization of its organization. I would not have been able to do the research and achieve learning in the same manner without his assistance and support. His recommendations and instructions have qualified me to assemble and finish the research effectively.

I would also like to thank all my teachers, who throughout my educational career have maintained and stimulated me to believe in my abilities.

Finally, my family has supported and helped me along the course of this dissertation by giving encouragement and providing the moral and emotional support I needed to complete my thesis. To them, I am everlastingly grateful.

Dedication

This document contains thesis for Cloud Computing project, the specializing part of the study Computer Science at **Maryam University** with a lot of effort, we have completed our thesis and we hope to receive our degrees.

The basic aim of this book is to provide the knowledge of the Cloud Computing this book provides detailed background knowledge about Cloud Computing.

First of all, I would like to thank Almighty Allah for giving me the courage and strength to accomplish my Four-year degree in the Computer Science which was the wish of my parents. Secondly I would like to thank all the members of my family who have given me reassurance and encouragement till the end of this long journey in Education.

Bundle of thanks to all our hard working teachers who have worked tirelessly for me since the beginning of the first grade up to Graduation.

A heartfelt thanks to my dear indeed brother, teacher and friend **Faidullah Rohi** for always doing his best to help me in my studies.

Lastly, I would like to thanks my Friends for their encouragement, who have always supported me.

Cloud Computing

Executive Summary

Cloud computing is a significant advancement in the delivery of information technology and services.

Cloud computing is considered as one of most accomplished and ubiquitous paradigms in 21st century, especially for the service related computing. It has been credited for the revolution, it has brought by abstracting the complex and expansive computing infrastructure underneath. Though, in service-oriented age, there are only three cloud paradigms that have attained significant popularity -- Infrastructure-as-a Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS).

Cloud computing now offers organizations more choices regarding how to run infrastructures, save costs, and delegate liabilities to third-party providers. It has become an integral part of technology and business models, and has forced businesses to adapt to new technology strategies.

Accordingly, the demand for cloud computing has forced the development of new market offerings, representing various cloud service and delivery models. These models significantly expand the range of available options, and task organizations with dilemmas over which cloud computing model to employ.

This thesis poses analysis of available cloud computing models and potential future cloud computing trends.

Table of Contents

Table of Figures

Chapter No: 1
Introduction to Cloud Computing

Cloud computing is a computing paradigm, where a large pool of systems are connected in private or public networks, to provide dynamically scalable infrastructure for application, data and file storage. With the advent of this technology, the cost of computation, application hosting, content storage and delivery is reduced significantly.

Cloud computing is a practical approach to experience direct cost benefits and it has the potential to transform a data center from a capital-intensive set up to a variable priced environment. The idea of cloud computing is based on a very fundamental principal of „reusability of IT capabilities'. The difference that cloud computing brings compared to traditional concepts of “grid computing”, “distributed computing”, “utility computing”, or “autonomic computing” is to broaden horizons across organizational boundaries.

Forrester defines cloud computing as: “A pool of abstracted, highly scalable, and managed compute infrastructure capable of hosting end customer applications and billed by consumption.”

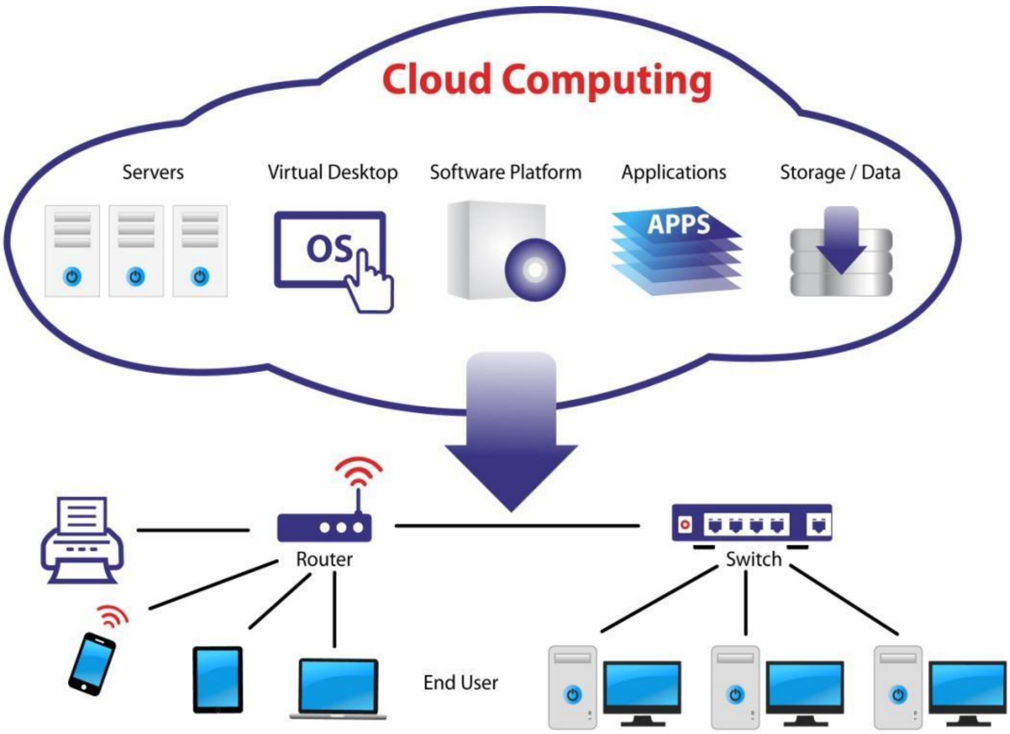


Figure1.1: Cloud Computing

1.1 What is the cloud?

Cloud computing is receiving a great deal of attention, both in publications and among users, from individuals at home to the U.S. government. Yet it is not always clearly defined. Cloud computing is a subscription-based service where you can obtain networked storage space and computer resources. One way to think of cloud computing is to consider your experience with email. Your email client, if it is Yahoo!, Gmail, Hotmail, and so on, takes care of housing all of the hardware and software necessary to support your personal email account.

When you want to access your email you open your web browser, go to the email client, and log in. The most important part of the equation is having internet access. Your email is not housed on your physical computer; you access it through an internet connection, and you can access it anywhere. If you are on a trip, at work, or down the street getting coffee, you can check your email as long as you have access to the internet.

Your email is different than software installed on your computer, such as a word processing program. When you create a document using word processing software, that document stays on the device you used to make it unless you physically move it. An email client is similar to how cloud computing works. Except instead of accessing just your email, you can choose what information you have access to within the cloud.

Computing is being transformed to a model consisting of services that are commoditized and delivered in a manner similar to utilities such as water, electricity, gas, and telephony. In such a model, users access services based on their requirements regardless of where they are hosted. Several computing paradigms such as Grid computing have promised to deliver this utility computing vision. Cloud computing is the most recent emerging paradigm promising to turn the vision of "computing utilities" into a reality.

Cloud computing is a technological advancement that focuses on the way in which we design computing systems, develop applications, and leverage existing services for building software. It is based on the concept of dynamic provisioning, which is applied not only to services, but also to compute capability, storage, networking, and Information Technology (IT) infrastructure in general. Resources are made available through the Internet and offered on a pay-per-use basis from Cloud computing vendors.

Today, anyone with a credit card can subscribe to Cloud services, and deploy and configure servers for an application in hours, growing and shrinking the infrastructure serving its application according to the demand, and paying only for the time these resources have been used.

This chapter provides a brief overview of the Cloud computing phenomenon, by presenting its vision, discussing its core features, and tracking the technological developments that have made it possible. The chapter also introduces some of its key technologies, as well as some insights into developments of Cloud computing environments.

1.2 The Vision of Cloud Computing

Cloud computing allows anyone having a credit card to provision virtual hardware, runtime environments, and services. These are used for as long as needed and no upfront commitments are required. The entire stack of a computing system is transformed into a collection of utilities, which can be provisioned and composed together to deploy systems in hours, rather than days, and with virtually no maintenance costs.

This opportunity, initially met with skepticism, has now become a practice across several. Application domains and business sectors. The demand has fast-tracked the technical development and enriched the set of services offered, which have also become more sophisticated and cheaper.

Despite its evolution, the usage of Cloud computing is often limited to a single service at time or, more commonly, a set of related services offered by the same vendor. The lack of effective standardization efforts made it difficult to move hosted services from one vendor to another.

The long term vision of Cloud computing is that IT services are traded as utilities in an open market without technological and legal barriers. In this Cloud marketplace, Cloud service providers and consumers, trading Cloud services as utilities, play a central role. Many of the technological elements contributing to this vision already exist.

Different stakeholders leverage Clouds for a variety of services. The need for ubiquitous storage and compute power on demand is the most common reason to consider Cloud computing.

A scalable runtime for applications is an attractive option for application and system developers that do not have infrastructure or cannot afford any further expansion of existing one. The capability of Web-based access to documents and their processing using sophisticated applications is one the appealing factors for end-users.

Figure1.2: global cloud

1.3 Defining a Cloud

Cloud computing has become a popular buzzword and it has been widely used to refer to different technologies, services, and concepts. It is often associated with virtualized infrastructure or hardware on demand, utility computing, IT outsourcing, platform and software as a service, and many other things that now are the focus of the IT industry. Figure 1.3 depicts the plethora of different notions one portrays when defining Cloud computing.

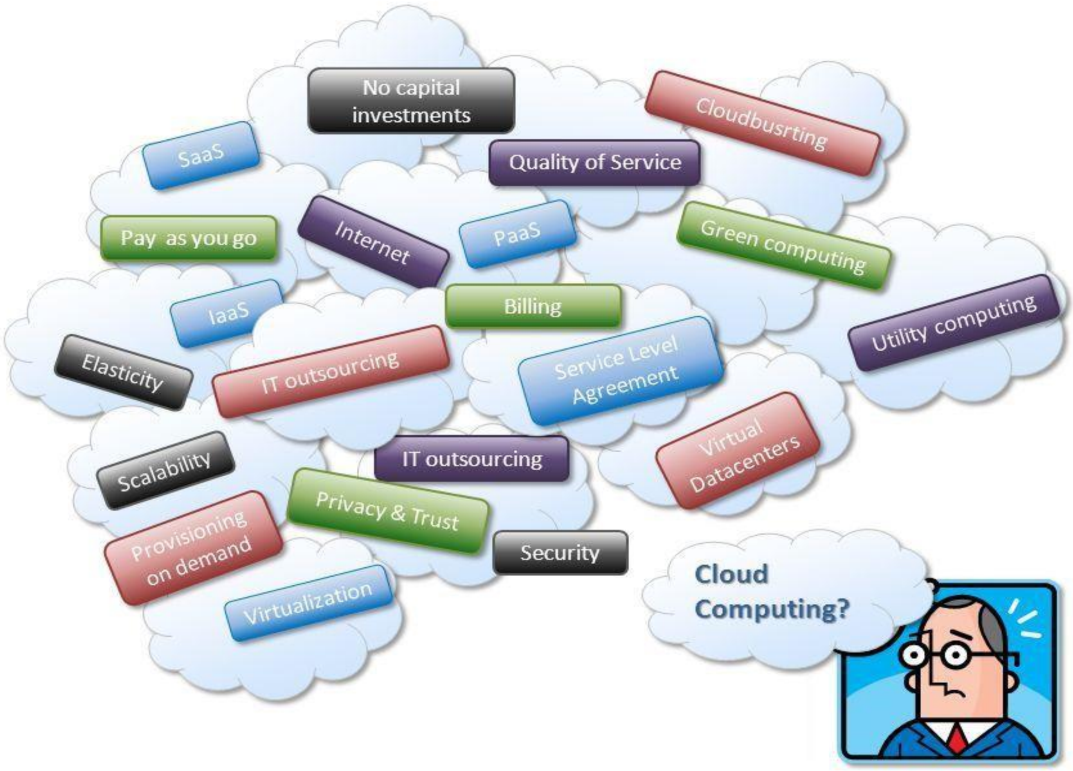


Figure1.3: Cloud

1.4 Cloud-Computing Reference Model

A fundamental characteristic of Cloud computing is the capability of delivering on demand a variety of IT services, which are quite diverse from each other. This variety creates a different perception of what Cloud computing is among users. Despite this, it is possible to classify Cloud computing services offerings into three major categories:

Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS). These categories are related to each other as described in Fig. 1.5, which provides an organic view of Cloud computing.

We refer to this diagram as "Cloud Computing Reference Model" and we will use it throughout the book to explain the technologies and introduce the relevant research on this phenomenon. The model organizes the wide range of Cloud computing services into a layered view that walks the computing stack from bottom to top.

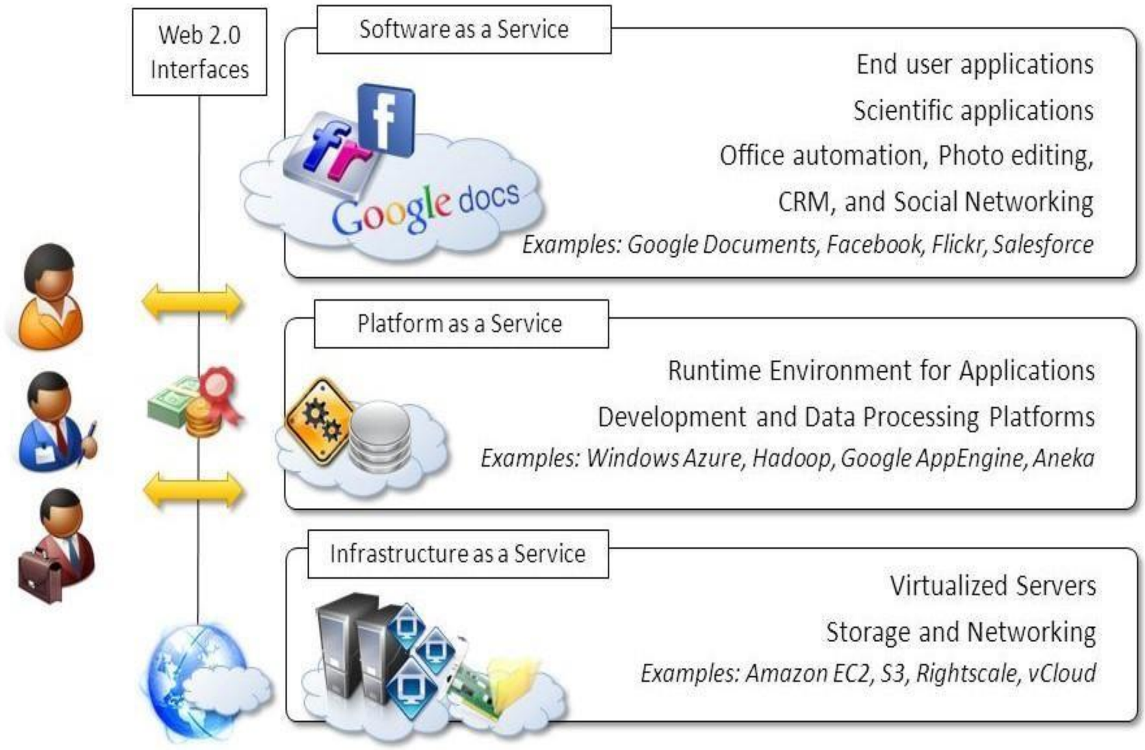


Figure1.4: Cloud Computing Reference

At the base of the stack, Infrastructure-as-a-Service solutions deliver infrastructure on demand in the form of virtual hardware, storage, and networking. Virtual hardware is utilized to provide compute on demand in the form of virtual machines instances. These are created on users' request on the provider's infrastructure, and users are given tools and interfaces to configure the software stack installed in the virtual machine. The pricing model is usually defined in terms of dollars per hours, where the hourly cost

Is influenced by the characteristics of the virtual hardware. Virtual storage is delivered in the form of raw disk space or object store. The former complements a virtual hardware offering that requires persistent storage. The latter is a more high-level abstraction for storing entities rather than fi les. Virtual networking identifies the collection of services that manage the networking among virtual instances and their connectivity towards the Internet or private networks.

Platform-as-a-Service solutions are the next step in the stack. They deliver scalable and elastic runtime environments on demand that host the execution of applications. These services are backed by a core middleware platform that is responsible for creating the abstract environment where applications are deployed and executed. It is the responsibility of the service provider to provide scalability and to manage fault-tolerance, while users are requested to focus on the logic of the application developed by leveraging the provider's APIs and libraries. This approach increases the level of abstraction at which Cloud computing is leveraged but also constrains the user in a more controlled environment.

At the top of the stack, Software-as-a-Service solutions provide applications and services on demand. Most of the common functionalities of desktop applications—such as office automation, document management, photo editing, and customer relationship management (CRM) software—are replicated on the provider's infrastructure, made more scalable, and accessible through a browser on demand. These applications are shared across multiple users, whose interaction is isolated from the other users. The SaaS layer is also the area of social networking Websites, which leverage Cloud-based infrastructures to sustain the load generated by their popularity.

Each layer provides a different service to users. IaaS solutions are sought by users that want to leverage Cloud computing from building dynamically scalable computing systems requiring a specific software stack. IaaS services are therefore used to develop scalable Web sites or for background processing. PaaS solutions provide scalable programming platforms for developing applications, and are more appropriate when new systems have to be developed.

IaaS solutions target mostly end users, who want to benefit from the elastic scalability of the Cloud without doing any software development, installation, configuration, and maintenance. This solution is appropriate when there are existing SaaS services that fi t user's needs (i.e., email, document management, CRM, etc.) and a minimum level of customization is needed.

1.5 Characteristics and Benefits

Cloud computing has some interesting characteristics that bring benefits to both Cloud Service Consumers (CSCs) and Cloud service providers (CSPs). They are

- No upfront commitments;
- On demand access;
- Nice pricing;
- Simplified application acceleration and scalability;
- Efficient resource allocation;

- Energy efficiency; and
- Seamless creation and the use of third-party services.

The most evident benefit from the use of Cloud computing systems and technologies is the increased economical return due to the reduced maintenance costs and operational costs related to IT software and infrastructure. This is mainly because IT assets, namely software and infrastructure, are turned into utility costs, which are paid for as long as they are used and not upfront. Capital costs are costs associated to assets that need to be paid in advance to start a business activity. Before Cloud computing, IT infrastructure and software generated capital costs, since they were paid upfront to afford a computing infrastructure enabling the business activities of an organization.

The revenue of the business is then utilized to compensate over time for these costs. Organizations always minimize capital costs, since they are often associated to depreciable values. This is the case of hardware: a server bought today for 1000 dollars will have a market value less than its original price when it will be replaced by a new hardware. In order to make profit, organizations have also to compensate this depreciation created by time, thus reducing the net gain obtained from revenue.

Minimizing capital costs is then fundamental. Cloud computing transforms IT infrastructure and software into utilities, thus significantly contributing in increasing the net gain.
Moreover, it also provides an opportunity for small organizations and start-ups: these do not need large investments to start their business but they can comfortably grow with it. Finally, maintenance costs are significantly reduced: by renting the infrastructure and the application services, organizations are not responsible anymore for their maintenance. This task is the responsibility of the Cloud service provider, who, thanks to the economies of scale, can bear maintenance costs.

Increased agility in defining and structuring software systems is another significant benefit. Since organizations rent IT services, they can more dynamically and flexibly compose their software systems, without being constrained by capital costs for IT assets. There is a reduced need for capacity planning, since Cloud computing allows to react to unplanned surges in demand quite rapidly.

For example, organizations can add more servers to process workload spikes, and dismiss them when there is no longer need. Ease of scalability is another advantage. By leveraging the potentially huge capacity of Cloud computing, organizations can extend their IT capability more easily.

Scalability can be leveraged across the entire computing stack. Infrastructure providers offer simple methods to provision customized hardware and integrate it into existing systems. Platform-as-a-Service providers offer run-time environment and programming models that are designed to scale applications. Software-as-a-Service offerings can be elastically sized on demand without requiring users to provision hardware, or to program application for scalability.

End users can benefit from Cloud computing by having their data and the capability of operating on it always available, from anywhere, at any time, and through multiple devices. Information and services stored in the Cloud are exposed to users by Web-based interfaces that make them accessible from portable devices as well as desktops at home.

Since the processing capabilities (i.e., office automation features, photo editing, information management, and so on) also reside in the Cloud, end users can perform the same tasks that previously were carried out with considerable software investments. The cost for such opportunities is generally very limited, since the Cloud service provider shares its costs across all the tenants that he is servicing.

Multi-tenancy allows for a better utilization of the shared infrastructure that is kept operational and fully active. The concentration of IT infrastructure and services into large datacenters also provides opportunity for considerable optimization in terms of resource allocation and energy efficiency, wch eventually can lead to a less impacting approach on the environment.

Finally, service orientation and on demand access create new opportunities for composing systems and applications with a flexibility not possible before Cloud computing. New service offerings can be created by aggregating together existing services and concentrating on added value. Since it is possible to provision on demand any component of the computing stack, it is easier to turn ideas into products, with limited costs and by concentrating the technical efforts on what matters: the added value.

1.6 Distributed Systems

Clouds are essentially large distributed computing facilities that make available their services to third parties on demand. As a reference, we consider the characterization of a distributed system proposed by Tanenbaum “A distributed system is a collection of independent computers that appears to its users as a single coherent system

This is a general definition, which includes a variety of computer systems but it evidences two very important elements characterizing a distributed system: the fact it is composed of multiple independent components and that these components are perceived as a single entity by users. This is particularly true in case of Cloud computing, where Clouds hide the complex architecture they rely on and provide a single interface to the users. The primary purpose of distributed systems is to share resources and to utilize them better.

This is true in the case of Cloud computing, where this concept is taken to the extreme and resources (infrastructure, runtime environments, and services) are rented to users. In fact, one of the driving factors for Cloud computing has been the availability of large computing facility of IT giants (Amazon, Google, etc.), who found that offering their computing capabilities as a service to be an opportunity for better utilization of their infrastructure. Distributed systems often exhibit other properties such as heterogeneity, openness, scalability, transparency, concurrency, continuous availability, and independent failures. To some extent, these also characterize Clouds, especially in the context of scalability, concurrency, and continuous availability.

Three major milestones have led to Cloud computing: mainframe computing, cluster computing, and Grid computing.

1.6.1 Mainframes

These were the first examples of large computational facilities leveraging multiple processing units. Mainframes were powerful, highly reliable computers specialized for large data movement and massive IO operations. They were mostly used by large organizations for bulk data processing such as online transactions, enterprise resource planning, and other operations involving the processing of significant amount of data. Even though mainframes cannot be considered distributed systems, they were offering large computational power by using multiple processors, which were presented as a single entity to users. One of the most attractive features of mainframes was the ability to be highly reliable computers that were “always on” and capable of tolerating failures transparently.

No system shut down was required to replace failed components, and the system could work without interruptions. Batch processing was the main application of mainframes. Now their popularity and deployments have reduced, but evolved versions of such systems are still in use for transaction processing (i.e., online banking, airline ticket booking, supermarket and telcos, and government services).

1.6.2 Clusters

Cluster computing started as a low-cost alternative to the use of mainframes and supercomputers. The technology advancement that created faster and more powerful mainframes and supercomputers has eventually generated an increased availability of cheap commodity machines as a side effect.

These machines could then be connected by a high-bandwidth network and controlled by specific software tools that manage them as a single system. By starting from the 1980s, clusters became the standard technology for parallel and high-performance computing. Being built by commodity machines, they were cheaper than mainframes, and made available high-performance computing to a large number of groups, including universities and small research labs. Cluster technology considerably contributed to the evolution of tools and framework for distributed computing, some of them include:

One of the attractive features of clusters was that the computational power of commodity machines could be leveraged to solve problems previously manageable only on expensive supercomputers. Moreover, clusters could be easily extended if more computational power was required.

1.6.3 Grids

Grid computing [8] appeared in the early 90^{s} as an evolution of cluster computing. In analogy with the power grid, Grid computing proposed a new approach to access large computational power, huge storage facilities, and a variety of services. Users can “consume” resources in the same way as they use other utilities such as power, gas, and water. Grids initially developed as aggregation of geographically dispersed clusters by means of Internet connection. These clusters belonged to different organizations and arrangements were made among them to share the computational power. Different from a “large cluster”, a computing grid was a dynamic aggregation of heterogeneous computing nodes, and its scale was nationwide or even worldwide.

Several reasons made possible the diffusion of computing grids: i) clusters were now resources quite common; ii) they were often under-utilized; iii) new problems were requiring computational power going beyond the capability of single clusters; iv) the improvements in networking and the diffusion of Internet made possible long distance high bandwidth connectivity. All these elements led to the development of grids, which now serve a multitude of users across the world.

Cloud computing is often considered as the successor of Grid computing. In reality, it embodies aspects of all of these three major technologies. Computing Clouds are deployed on large datacenters hosted by a single organization that provides services to others. Clouds are characterized by the fact of having virtually infinite capacity, being tolerant to failures, and always on as in the case of mainframes. In many cases, the computing nodes that form the infrastructure of computing Clouds are commodity machines as in the case of clusters. The services made available by a Cloud vendor are consumed on a pay-per-use basis and Clouds implement fully the utility vision introduced by Grid computing.

1.7 Virtualization

Virtualization is another core technology for Cloud computing. It encompasses a collection of solutions allowing the abstraction of some of the fundamental elements for computing such as: hardware, runtime environments, storage, and networking. Virtualization has been around for more than 40 years, but its application has always been limited by technologies that did not allow an efficient use of virtualization solutions. Today these limitations have been substantially overcome and virtualization has become a fundamental element of Cloud computing.

This is particularly true for solutions that provide IT infrastructure on demand. Virtualization confers that degree of customization and control that makes Cloud computing appealing for users and, at the same time, sustainable for Cloud services providers.

Virtualization is essentially a technology that allows creation of different computing environments. These environments are named as virtual, because they simulate the interface that is expected by a guest. The most common example of virtualization is hardware virtualization. This technology allows simulating the hardware interface expected by an operating system. Hardware virtualization allows the co-existence of different software stacks on top of the same hardware.

These stacks are contained inside virtual machine instances, which operate completely isolated from each other. High-performance server can host several virtual machine instances, thus creating the opportunity of having customized software stack on demand. This is the base technology that enables Cloud computing solutions delivering virtual server on demands, such as Amazon EC2, Right Scale, VMware VCloud, and others. Together with hardware virtualization, storage and network virtualization complete the range of technologies for the emulation of IT infrastructure.

Virtualization technologies are also used to replicate runtime environments for programs. In the case of process virtual machines, which include the foundation of technologies such as Java or .NET, where applications instead of being executed by the operating system are run by a specific program called virtual machine.

This technique allows isolating the execution of applications and providing a finer control on the resource they access. Process virtual machines offer a higher level of abstraction with respect to the hardware virtualization since the guest is only constituted by an application rather than a complete Software stack. This approach is used in Cloud computing in order to provide a platform for scaling applications on demand, such as Google AppEngine and Windows Azure.

Having isolated and customizable environments with minor impact on performance is what makes virtualization an attractive technology. Cloud computing is realized through platforms that leverage the basic concepts described above and provides on-demand virtualization services to a multitude of users across the globe.

1.8 Web

The Web is the primary interface through which Cloud computing deliver its services. At present time, it encompasses a set of technologies and services that facilitate interactive information sharing, collaboration, user-centered design, and application composition. This has transformed the Web into a rich platform for application development. Such evolution is known as “Web 2.0”. This term captures a new way in which developers architect applications, deliver services through the Internet, and provide a new user experience for their users.

Web 2.0 brings interactivity and flexibility into Web pages, which provide enhanced user experience by gaining Web-based access to all the functions that are normally found in desktop applications. These capabilities are obtained by integrating a collection of standards and technologies such as XML, Asynchronous JavaScript and XML (AJAX), Web Services, and others. These technologies allow building applications leveraging the contribution of users, who now become providers of content.

Also, the capillary diffusion of the Internet opens new opportunities and markets for the Web, whose services can now be accessed from a variety of devices: mobile phones, car dashboards, TV sets, and others. This new scenarios require an increased dynamism for applications, which is another key element of this technology. Web 2.0 applications are extremely dynamic: they improve continuously and new updates and features are integrated at a constant rate, by following the usage trend of the community. There is no need to deploy new software releases on the installed base at the client side.

Users can take advantage of the new software features simply by interacting with Cloud applications. Lightweight deployment and programming models are very important for effective support of such dynamism. Loose coupling is another fundamental property. New applications can be “synthesized” simply by composing existing services and integrating them together, thus providing added value. By doing this, it becomes easier to follow the interests of users. Finally, Web 2.0 applications aim to leverage the long tail of Internet users by making themselves available to everyone either in terms of media accessibility or cost.

Examples of Web 2.0 applications are Google Documents, Google Maps, Flickr, Facebook, Twitter, YouTube, delicious, Blogger, and Wikipedia. In particular, social networking Websites take the biggest advantage from Web 2.0. The level of interaction in Web sites like Facebook or Flickr would not have been possible without the support of AJAX, RSS, and other tools that make the user experience incredibly interactive. Moreover, community Websites harness the collective intelligence of the community which provides content to the applications themselves: Flickr provides advanced services for storing digital pictures and videos, Facebook is a social networking Website leveraging the user activity for providing content, and Blogger as any other blogging Website provides an online diary that is fed by the users.

This idea of the Web as a transport that enables and enhances interaction was introduced in 1999 by Darcy DiNucci and started to fully realize in 2004. Today, it is a mature platform for supporting the need of Cloud computing, which strongly leverages Web 2.0. Applications and frameworks for delivering Rich Internet Applications (RIAs) are fundamental for making Cloud services accessible to the wider public.

From a social perspective, Web 2.0 applications definitely contributed to make people more accustomed to the use of Internet in their everyday lives, and opened the path to the acceptance of Cloud computing as a paradigm where even the IT infrastructure is offered through a Web interface.

1.9 Service-Oriented Computing

Service orientation is the core reference model for Cloud computing systems. This approach adopts the concept of services as main building blocks of application and system development. Service-Oriented Computing (SOC) supports the development of rapid, lowcost, flexible, interoperable, and evolvable applications and systems.

A service is an abstraction representing a self-describing and platform agnostic component that can perform any function: this can be anything from a simple function to a complex business process. Virtually, any piece of code that performs a task can be turned into a service and expose its functionalities through a network accessible protocol. A service is supposed to be loosely coupled, reusable, programming language independent, and location transparent.

Loose coupling allows services to serve different scenarios more easily and makes them reusable. Independence from a specific platform increases services accessibility. Thus, a wider range of clients, which can look up services in global registries and consume them in location transparent manner, can be served. Services are composed and aggregated into a Service-Oriented Architecture (SOA) which is a logical way of organizing software systems to provide end users or other entities distributed over the network with services through published and discoverable interfaces.

Service-Oriented Computing introduces and diffuses two important concepts, which are also fundamental for Cloud computing: Quality of Service (QoS) and Software as a Service (SaaS).

- Quality of Service identifies a set of functional and non-functional attributes that can be used to evaluate the behavior of a service from different perspectives. These could be performance metrics such as response time, or security attributes, transactional integrity, reliability, scalability, and availability. QoS requirements are established between the client and the provider between a Service Level Agreement (SLA) that identifies the minimum values (or an acceptable range) for the QoS attributes that need to be satisfied upon service call.

The concept of Software as a Service introduces a new delivery model for applications. It has been inherited from the world of Application Service Providers (ASPs). These deliver software services-based solutions across the wide area network from a central data center and make them available on subscription or rental basis. The ASP is responsible for maintaining the infrastructure and making available the application, and the client is freed from maintenance cost and difficult upgrades.

This software delivery model is possible because economies of scale are reached by means of multi-tenancy. The SaaS approach reaches its full development with Service Oriented Computing, where loosely coupled software component can be exposed and priced singularly, rather than entire applications. This allows the delivery of complex business processes and transactions as a service, while allowing applications to be composed on the fl y and services to be reused from everywhere by anybody.

One of the most popular expressions of service orientation is represented by Web Services (WS) These introduce the concepts of SOC into the World Wide Web, by making it consumable by applications and not only humans. Web services are software components exposing functionalities accessible by using a method invocation pattern that goes over the HTTP protocol. The interface of a Web service can be programmatically inferred by metadata expressed through the Web Service Description Language (WSDL) this is an XML language that defi nes the characteristics of the service and all the methods, together with parameters descriptions and return type, exposed by the service. The interaction with Web services happens through Simple Object Access Protocol (SOAP) This is an XML language defi ning how to invoke a Web service method and collect the result. By using SOAP and WSDL over HTTP, Web services become platform independent and accessible as the World Wide Web. The standards and specifi cations concerning Web services are controlled by the W3C, while among the most popular architectures for developing Web services, we can note ASP.NET and Axis.

The development of systems in terms of distributed services that can be composed together is the major contribution given by SOC to the realization of Cloud computing. Web services technologies have provided the right tools to make such composition straightforward and integrated with the mainstream World Wide Web (WWW) environment easier.

1.10 Utility-Oriented Computing

Utility computing is a vision of computing, defining a service provisioning model for compute services in which resources such as storage, compute power, applications, and infrastructure are packaged and offered on a pay-per-use basis. The idea of providing computing as a utility like natural gas, water, power, and telephone connection has a long history but has become a reality today with the advent of Cloud computing. Among the earliest forerunners of this vision, we can include the American scientist, John McCarthy, who in a speech for the MIT centennial in 1961 observed:

"If computers of the kind I have advocated become the computers of the future, then computing may someday be organized as a public utility just as the telephone system is a public utility... The computer utility could become the basis of a new and important industry."

The first traces of this service provisioning model can be found in the mainframes era. IBM and other mainframe providers offered mainframe power to organizations such as banks and government agencies throughout their datacenters.

The business model introduced with utility computing brought new requirements and led to an improvement of mainframe technology: additional features such as operating systems, process control and user metering facilities. The idea of computing as utility remained and extended from the business domain to the academia with the advent of cluster computing. Not only businesses but also research institutes became acquainted with the idea of leveraging an external IT infrastructure on demand. Computational science, which was one of the major driving factors for building computing clusters, still required huge compute power for addressing Grand Challenge problems, and not all the institutions were able to satisfy their computing needs internally. Access to external clusters still remained a common practice.

The capillary diffusion of the Internet and the Web provided the technological means to realize utility computing at a world-wide scale and through simple interfaces. As already discussed before, computing grids-provided a planet-scale distributed computing infrastructure that was accessible on demand.

Computing grids brought the concept of utility computing to a new level: market orientation Being accessible on a wider scale, it is easier to provide a trading infrastructure where Grid products—storage, computation, and services—are bid for or sold. Moreover, e-Commerce technologies provided the infrastructure support for utility computing.

In the late nineties, a significant interest in buying online any kind of good spread in the wide public: food, clothes, multimedia products, and also online services such as storage space and Web hosting.

After the dot-com bubble, this interest reduced in size but the phenomenon made the wide public keener to buy online services. As a result, infrastructures for on-line payment through credit card become easily accessible and well proven.

From an application and system development perspective, service-oriented computing and Service Oriented Architectures (SOAs) introduced the idea of leveraging external services for performing a specific task within a software system. Applications were not only distributed, but started to be composed as a mesh of services provided by different entities.

These services, accessible through the Internet, were made available by charging according on usage. Service-oriented computing broadened the concept of what could have been accessed as a utility in a computer system. Not only compute power and storage but also services and application components could be utilized and integrated on demand. Together with this trend, Quality of Service became an important topic to investigate on.

All these factors contributed to the development of the concept of utility computing and offered important steps in the realization of Cloud computing, in which "computing utilities" vision comes to its full expression.

1.11 How can you use the cloud?

The cloud makes it possible for you to access your information from anywhere at any time. While a traditional computer setup requires you to be in the same location as your data storage device, the cloud takes away that step. The cloud removes the need for you to be in the same physical location as the hardware that stores your data. Your cloud provider can both own and house the hardware and software necessary to run your home or business applications.

This is especially helpful for businesses that cannot afford the same amount of hardware and storage space as a bigger company. Small companies can store their information in the cloud, removing the cost of purchasing and storing memory devices. Additionally, because you only need to buy the amount of storage space you will use, a business can purchase more space or reduce their subscription as their business grows or as they find they need less storage space.

One requirement is that you need to have an internet connection in order to access the cloud. This means that if you want to look at a specific document you have housed in the cloud, you must first establish an internet connection either through a wireless or wired internet or a mobile broadband connection. The benefit is that you can access that same document from wherever you are with any device that can access the internet. These devices could be a desktop, laptop, tablet, or phone. This can also help your business to function more smoothly because anyone who can connect to the internet and your cloud can work on documents, access software, and store data. Imagine picking up your smartphone and downloading a .pdf document to review instead of having to stop by the office to print it or upload it to your laptop. This is the freedom that the cloud can provide for you or your organization.

Chapter No: 2
Types of clouds

There are different types of clouds that you can subscribe to depending on your needs. As a home user or small business owner, you will most likely use public cloud services.

1. Public Cloud - A public cloud can be accessed by any subscriber with an internet connection and access to the cloud space.

2. Private Cloud - A private cloud is established for a specific group or organization and limits access to just that group.

3. Community Cloud - A community cloud is shared among two or more organizations that have similar cloud requirements.

4. Hybrid Cloud - A hybrid cloud is essentially a combination of at least two clouds, where the clouds included are a mixture of public, private, or community.

2.1 Cloud Computing Models

Cloud Providers offer services that can be grouped into three categories.

2.1.1: Software as a Service (SaaS)

In this model, a complete application is offered to the customer, as a service on demand. A single instance of the service runs on the cloud & multiple end users are serviced. On the customers‟ side, there is no need for upfront investment in servers or software licenses, while for the provider, the costs are lowered, since only a single application needs to be hosted & maintained. Today SaaS is offered by companies such as Google, Salesforce, Microsoft, Zoho, etc.

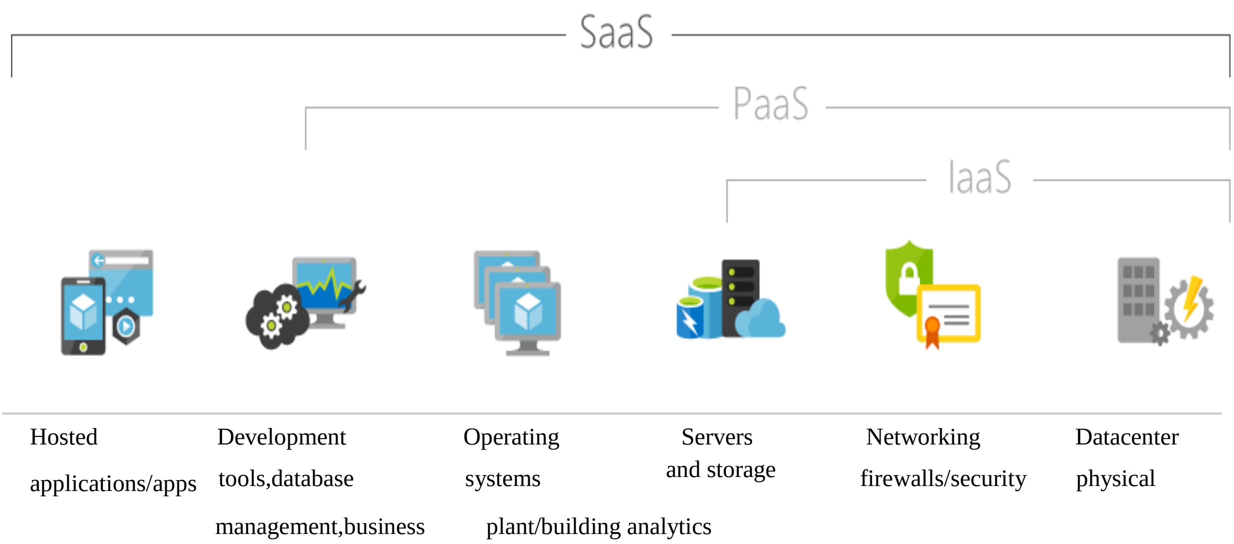


Figure2.1: Software as a Service (SaaS)

2.1.2: Platform as a Service (PaaS)

Here, a layer of software or development environment is encapsulated & offered as a service, upon which other higher levels of service can be built. The customer has the freedom to build his own applications, which run on the provider's infrastructure.

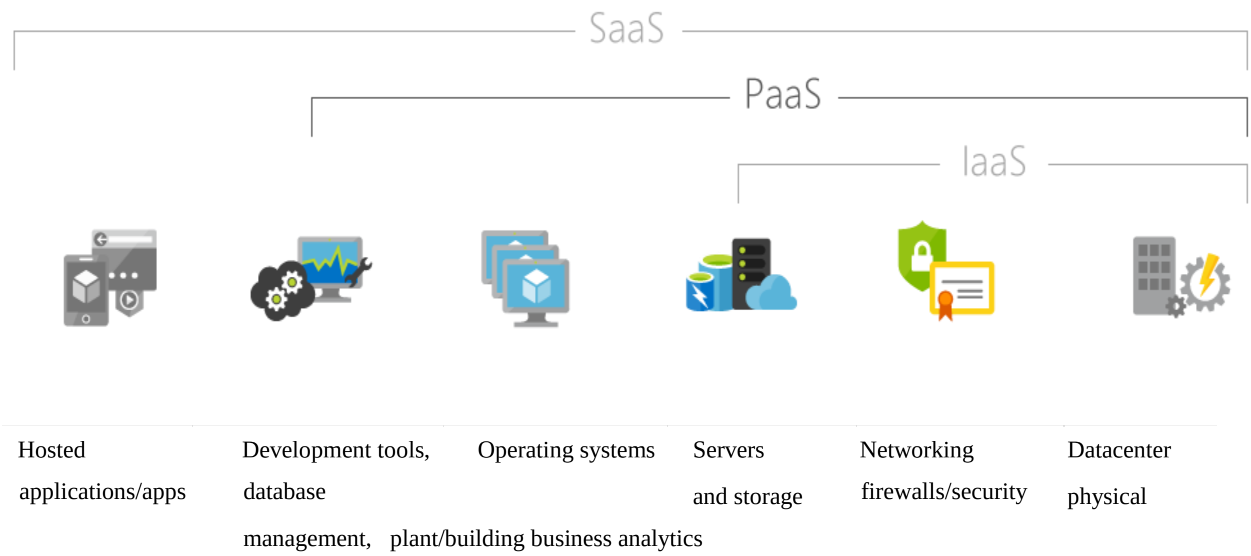
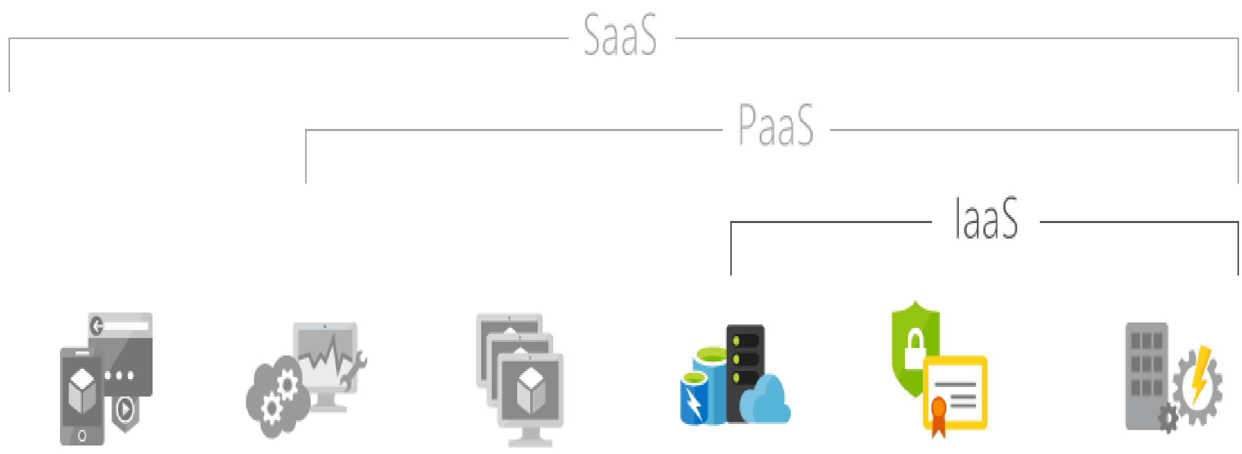


Figure2.2: Platform as a Service (PaaS)

To meet manageability and scalability requirements of the applications, PaaS providers offer a predefined combination of OS and application servers, such as LAMP platform (Linux, Apache, MySql and PHP), restricted J2EE, Ruby etc. Google's App Engine, Force.com, etc. are some of the popular PaaS examples.

2.1.3: Infrastructure as a Service (Iaas)

IaaS provides basic storage and computing capabilities as standardized services over the network. Servers, storage systems, networking equipment, data centre space etc. are pooled and made available to handle workloads. The customer would typically deploy his own software on the infrastructure. Some common examples are Amazon, GoGrid, 3 Tera, etc.



Hosted applications/apps | Development tools,database management, | Operating systems | Servers and storage | Networking firewalls/security | Datacenter physical plant/building business analytics

Figure2.3: Infrastructure as a Service (Iaas)

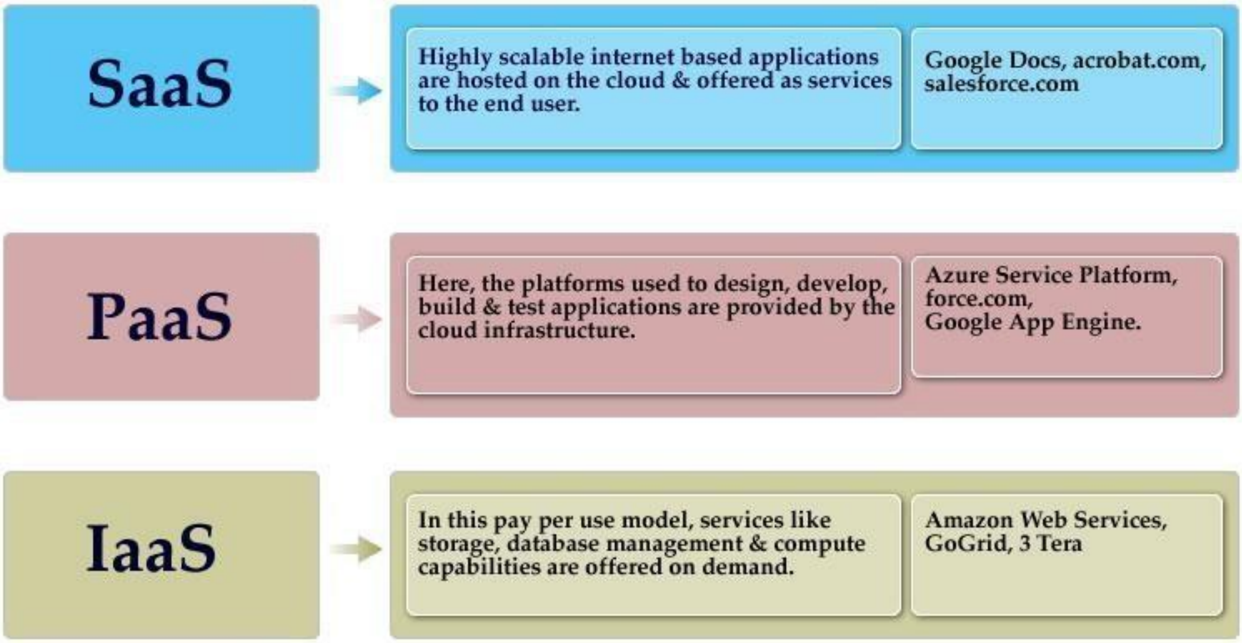


Figure2.4: Infrastructure as a Service(Iaas)

2.2 Understanding Public and Private Clouds

Enterprises can choose to deploy applications on Public, Private or Hybrid clouds. Cloud Integrators can play a vital part in determining the right cloud path for each organization.

2.3 Public Cloud

Public clouds are owned and operated by third parties; they deliver superior economies of scale to customers, as the infrastructure costs are spread among a mix of users, giving each individual client an attractive low-cost, “Pay-as-you-go” model.

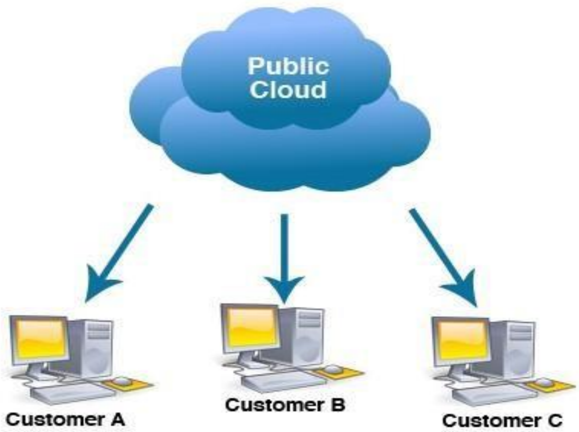


Figure2.5: Public Cloud

All customers share the same infrastructure pool with limited configuration, security protections, and availability variances. These are managed and supported by the cloud provider.

One of the advantages of a Public cloud is that they may be larger than an enterprises cloud, thus providing the ability to scale seamlessly, on demand.

2.4 Private Cloud

Private clouds are built exclusively for a single enterprise. They aim to address concerns on data security and offer greater control, which is typically lacking in a public cloud.

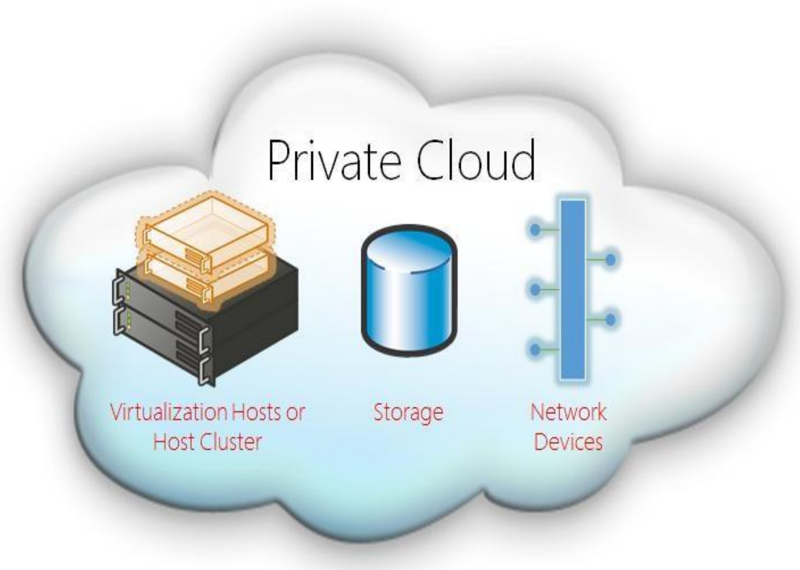


Figure2.6: Private Cloud

There are two variations to a private cloud:

2.4.1: On-premise Private Cloud

On-premise private clouds, also known as internal clouds are hosted within one's own data center. This model provides a more standardized process and protection, but is limited in aspects of size and scalability.

IT departments would also need to incur the capital and operational costs for the physical resources. This is best suited for applications which require complete control and configurability of the infrastructure and security.

2.4.2: Externally hosted Private Cloud

This type of private cloud is hosted externally with a cloud provider, where the provider facilitates an exclusive cloud environment with full guarantee of privacy. This is best suited for enterprises that don't prefer a public cloud due to sharing of physical resources.

2.5 Hybrid Cloud

Hybrid Clouds combine both public and private cloud models. With a Hybrid Cloud, service providers can utilize 3rd party Cloud Providers in a full or partial manner thus increasing the flexibility of computing. The Hybrid cloud environment is capable of providing on-demand, externally provisioned scale. The ability to augment a private cloud with the resources of a public cloud can be used to manage any unexpected surges in workload.

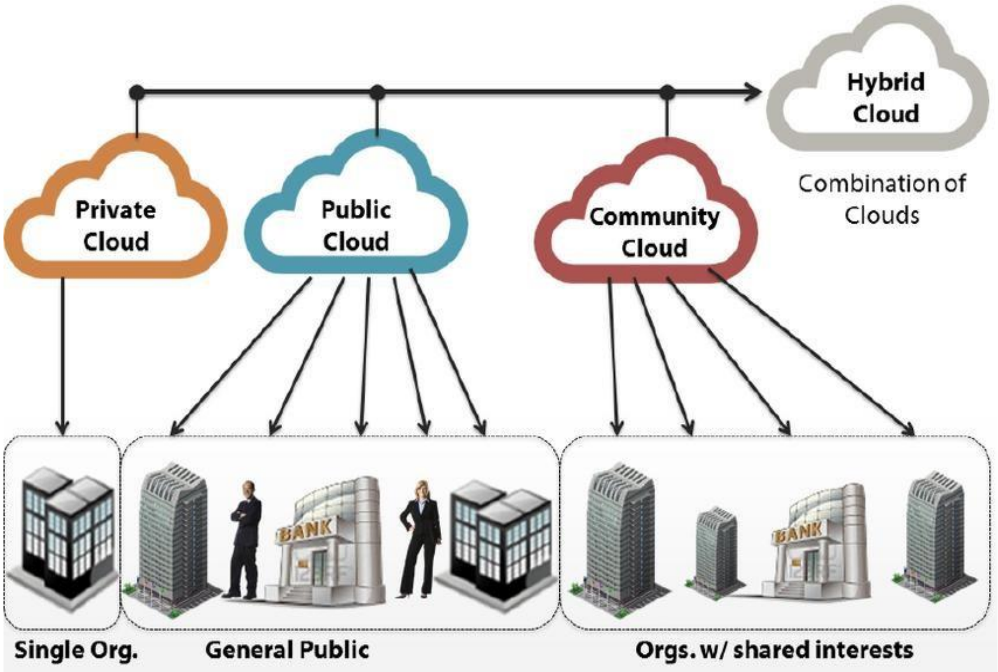


Figure2.7: Hybrid Cloud

2.6 Cloud Computing Benefits

Enterprises would need to align their applications, so as to exploit the architecture models that Cloud Computing offers. Some of the typical benefits are listed below:

2.6.1: Reduced Cost

There are a number of reasons to attribute Cloud technology with lower costs. The billing model is pay as per usage; the infrastructure is not purchased thus lowering maintenance.
Initial expense and recurring expenses are much lower than traditional computing.

2.6.2: Increased Storage

With the massive Infrastructure that is offered by Cloud providers today, storage & maintenance of large volumes of data is a reality. Sudden workload spikes are also managed effectively & efficiently, since the cloud can scale dynamically.

2.6.3: Flexibility

This is an extremely important characteristic. With enterprises having to adapt, even more rapidly, to changing business conditions, speed to deliver is critical. Cloud computing stresses on getting applications to market very quickly, by using the most appropriate building blocks necessary for deployment.

2.7 Cloud Computing Challenges

Despite its growing influence, concerns regarding cloud computing still remain. In our opinion, the benefits outweigh the drawbacks and the model is worth exploring. Some common challenges are:

2.7.1: Data Protection

Data Security is a crucial element that warrants scrutiny. Enterprises are reluctant to buy an assurance of business data security from vendors. They fear losing data to competition and the data confidentiality of consumers.

In many instances, the actual storage location is not disclosed, adding onto the security concerns of enterprises. In the existing models, firewalls across data centers (owned by enterprises) protect this sensitive information. In the cloud model, Service providers are responsible for maintaining data security and enterprises would have to rely on them.

2.7.2: Data Recovery and Availability

All business applications have Service level agreements that are stringently followed. Operational teams play a key role in management of service level agreements and runtime governance of applications. In production environments, operational teams support

- Appropriate clustering and Fail over
- Data Replication
- System monitoring (Transactions monitoring, logs monitoring and others)
- Maintenance (Runtime Governance)
- Disaster recovery
- Capacity and performance management If, any of the above mentioned services is under-served by a cloud provider, the damage & impact could be severe.

2.7.3: Management Capabilities

Despite there being multiple cloud providers, the management of platform and infrastructure is still in its infancy. Features like „Auto-scaling" for example, are a crucial requirement for many enterprises. There is huge potential to improve on the scalability and load balancing features provided today.

2.7.4: Regulatory and Compliance Restrictions

In some of the European countries, Government regulations do not allow customer's personal information and other sensitive information to be physically located outside the state or country. In order to meet such requirements, cloud providers need to setup a data center or a storage site exclusively within the country to comply with regulations. Having such an infrastructure may not always be feasible and is a big challenge for cloud providers.

With cloud computing, the action moves to the interface that is, to the interface between service suppliers and multiple groups of service consumers. Cloud services will demand expertise in distributed services, procurement, risk assessment and service negotiation areas that many enterprises are only modestly equipped to handle.

Chapter No: 3
Choosing a cloud provider

Each provider serves a specific function, giving users more or less control over their cloud depending on the type. When you choose a provider, compare your needs to the cloud services available. Your cloud needs will vary depending on how you intend to use the space and resources associated with the cloud. If it will be for personal home use, you will need a different cloud type and provider than if you will be using the cloud for business. Keep in mind that your cloud provider will be pay-as-you-go, meaning that if your technological needs change at any point you can purchase more storage space (or less for that matter) from your cloud provider.

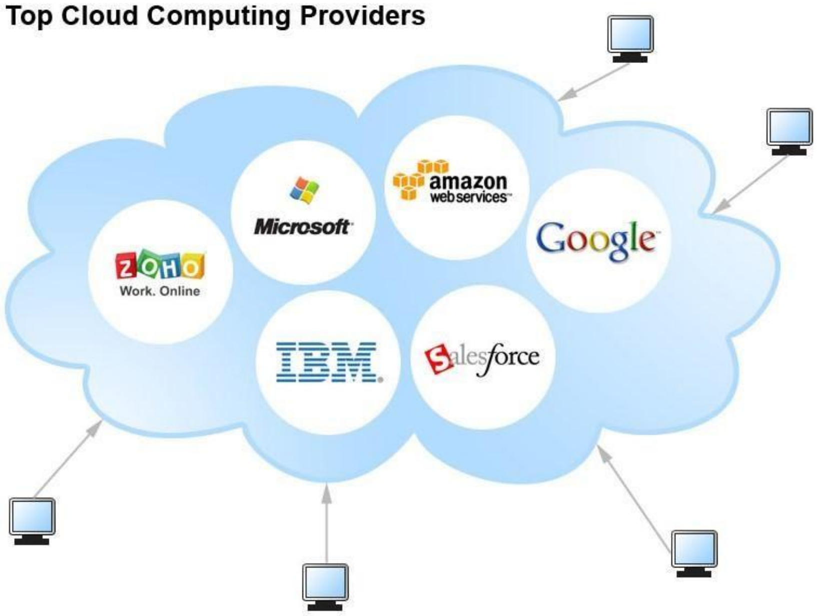


Figure3.1: Cloud Computing Provider

There are three types of cloud providers that you can subscribe to: Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS). These three types differ in the amount of control that you have over your information, and conversely, how much you can expect your provider to do for you. Briefly, here is what you can expect from each type.

3.1: Software as a Service - A SaaS provider gives subscribers access to both resources and applications. SaaS makes it unnecessary for you to have a physical copy of software to install on your devices. SaaS also makes it easier to have the same software on all of your devices at once by accessing it on the cloud. In a SaaS agreement, you have the least control over the cloud.

3.2: Platform as a Service - A PaaS system goes a level above the Software as a Service setup. A PaaS provider gives subscribers access to the components that they require to develop and operate applications over the internet.

3.3: Infrastructure as a Service - An IaaS agreement, as the name states, deals primarily with computational infrastructure. In an IaaS agreement, the subscriber completely outsources the storage and resources, such as hardware and software that they need.

As you go down the list from number one to number three, the subscriber gains more control over what they can do within the space of the cloud. The cloud provider has less control in an IaaS system than with an SaaS agreement.

What does this mean for the home user or business looking to start using the cloud? It means you can choose your level of control over your information and types of services that you want from a cloud provider. For example, imagine you are starting up your own small business.

You cannot afford to purchase and store all of the hardware and software necessary to stay on the cutting edge of your market. By subscribing to an Infrastructure as a Service cloud, you would be able to maintain your new business with just as much computational capability as a larger, more established company, while only paying for the storage space and bandwidth that you use.

However, this system may mean you have to spend more of your resources on the development and operation of applications. As you can see, you should evaluate your current computational resources, the level of control you want to have, your financial situation, and where you foresee your business going before signing up with a cloud provider.

If you are a home user, however, you will most likely be looking at free or low-cost cloud services (such as web-based email) and will not be as concerned with many of the more complex cloud offerings. After you have fully taken stock of where you are and where you want to be, research into each cloud provider will give you a better idea of whether they are right for you.

3.1 Security

The information housed on the cloud is often seen as valuable to individuals with malicious intent. There is a lot of personal information and potentially secure data that people store on their computers, and this information is now being transferred to the cloud. This makes it critical for you to understand the security measures that your cloud provider has in place, and it is equally important to take personal precautions to secure your data.

The first thing you must look into is the security measures that your cloud provider already has in place. These vary from provider to provider and among the various types of clouds. What encryption methods do the providers have in place? What methods of protection do they have in place for the actual hardware that your data will be stored on? Will they have backups of my data? Do they have firewalls set up?

If you have a community cloud, what barriers are in place to keep your information separate from other companies? Many cloud providers have standard terms and conditions that may answer these questions, but the home user will probably have little negotiation room in their cloud contract.

A small business user may have slightly more room to discuss the terms of their contract with the provider and will be able to ask these questions during that time. There are many questions that you can ask, but it is important to choose a cloud provider that considers the security of your data as a major concern.

No matter how careful you are with your personal data, by subscribing to the cloud you will be giving up some control to an external source. This distance between you and the physical location of your data creates a barrier. It may also create more space for a third party to access your information. However, to take advantage of the benefits of the cloud, you will have to knowingly give up direct control of your data. On the converse, keep in mind that most cloud providers will have a great deal of knowledge on how to keep your data safe. A provider likely has more resources and expertise than the average user to secure their computers and networks.

3.2 Essential Characteristics of the Cloud

We can summarize the essential characteristics of the Cloud as below:

3.2.1: On-demand self-service:

A service consumer can automatically make use of the computing capabilities, such as server processing time and network storage without requiring human interaction with each service's provider.

3.2.2: Broad network access:

Cloud capabilities (HW and SW) are available over the network and accessed through various platforms (e.g., mobile phones, laptops, and tablets).

3.2.3: Resource pooling:

The provider's computing resources (HW and SW) are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to users' demand. Multi-tenancy is the most important feature of the cloud-based application.

It is characterized by the location independence feature in which the customer has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state, or datacenter). Examples of resources include storage, processing, memory, network bandwidth, and virtual machines.

3.2.4: Rapid elasticity:

Capabilities can be rapidly and elastically provisioned; it can be quickly scaled out, and quickly scaled in. For the user, the capabilities available for provisioning appear to be unlimited and can be purchased in any quantity at any time.

3.2.5: Measured Service:

Cloud systems automatically control and optimize resources use by leveraging a metering capability in which resources' usage can be monitored, controlled, and reported, providing transparency for both the provider and consumer of the utilized service. The advantage here is that you are paying for exactly what you are using.

3.3 cloud computing models

The cloud services can be categorized into software services and infrastructure or hardware services. In terms of maturity, software in the cloud is much more evolved than hardware in the cloud.

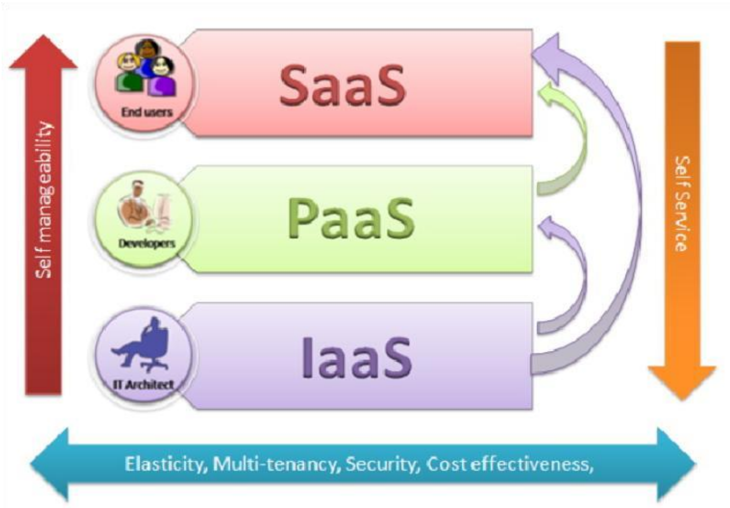


Figure3.2: Cloud Computing Models

3. **3.1: Cloud Software as a Service (SaaS):** is basically a term that refers to software in the cloud. It represents the capability provided to the consumer to use the provider's applications running on a cloud infrastructure. The applications are accessible from various client devices through an interface such as a web browser (e.g. web-based email like Gmail is a form of SaaS provided by Google). The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities.

3.4 SaaS systems have some defining characteristics:

3.4.1: Availability via web browser

SaaS software never requires the installation of software on your laptop or desktop. You access it through a web browser using open standards or a browser plug-in.

3.4.2: On-demand availability

You should not have to go through a sales process to gain access to SaaSbased software. Once you have access, you should be able to go back into the software any time, from anywhere.

3.4.3: Payment terms based on usage

SaaS does not need any infrastructure investment or complex setup, so you should not have to pay any massive setup fees. You should simply pay for the parts of the service you use as you use them. When you no longer need those services, you simply stop paying.

3.4.4: Minimal IT demands

SaaS systems don't require a high technical knowledge for their configuration.

3.4.5: Cloud Platform as a Service (PaaS):

The capability provided to the consumer is to deploy onto the cloud infrastructure consumer created or acquired applications created using programming languages and tools supported by the provider. The consumer does not manage or control the underlying cloud infrastructure, but has control over the deployed applications and possibly application hosting environment configurations.

3.4.6: Cloud Infrastructure as a Service (IaaS):

The capability provided to the consumer is to make use of 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, deployed applications.

3.5 Infrastructure as a Service (IaaS)

The focus of this tutorial is on the IaaS service model. Amazon Web Services (AWS) is one of the major players in this area. The AWS is based on pure virtualization, it owns all the hardware and controls the network infrastructure and you own everything from the guest operating system up. You request virtual instances on-demand and let them go when you are done.

AppNexus represents a different approach to this problem. As with AWS, AppNexus enables you to gain access to servers on demand. However, it provides dedicated servers with virtualization on top. You have the confidence in knowing that your applications are not fighting with anyone else for resources and that you can meet any requirements that demand full control over all physical server resources.

Hybrid computing takes advantage of both approaches, offering virtualization when appropriate and a dedicated hardware when appropriate. In addition, most hybrid vendors such as Rack space and Go Grid base their model on the idea that users still want a traditional data center and dedicated storage, but they just want it in the cloud. This tutorial focuses on the Amazon AWS and provides a practical example about using the Amazon EC2 IaaS solution.

3.6 Amazon Web Services (AWS)

AWS is Amazon’s umbrella description of all of their web-based technology services. It encompasses a wide variety of services, all of which fall into the concept of cloud computing like:

- Amazon Elastic Cloud Compute (Amazon EC2)
- Amazon Simple Storage Service (Amazon S3)
- Amazon Simple Queue Service (Amazon SQS)
- Amazon Cloud Front
- Amazon Simple DB

3.7 Amazon Elastic Cloud Computing (EC2)

Amazon EC2 is the heart of the Amazon cloud. It provides a web services API for provisioning, managing, and de-provisioning of virtual servers inside the Amazon cloud. In other words, any application anywhere on the Internet can launch a virtual server in the Amazon cloud with a single web services call.

At the time of this tutorial, Amazon’s EC2 U.S. footprint has three data centers on the east coast of the U.S. and two in western Europe You can sign up separately for an Amazon European data center account, but you cannot mix U.S. and European environments.

The servers in these environments run a highly customized version of tools that are using the virtualization concept and enable the dynamic provisioning and de-provisioning of servers. When you want to start up a virtual server in the Amazon environment, you launch a new instance based on a predefined Amazon machine image (AMI). The AMI includes your operating system and any other prebuilt software.

Most people start with a standard AMI based on their favorite operating system, customize it, create a new image based on their preferences, and then launch their servers based on their custom images. Many competitors to Amazon also provide persistent internal storage for nodes to make them operate more like a traditional data center.

In addition, servers in EC2 like any other server on the Internet can access Amazon S3 for cloud-based persistent storage. EC2 servers in particular achieve cost savings and greater efficiencies in accessing S3 .

When most people think of the Amazon cloud, they are thinking about Amazon EC2. EC2 represents your virtual network with all of the virtual servers running inside that network. When you use EC2, you will be able to use S3 to store your machine images and also for other storage needs.

3.8 Amazon EC2 Concepts

Amazon EC2 is a bit more complex than S3. Figure 2 shows all of the concepts that make up Amazon EC2 and how they relate to each other. The main concepts are:

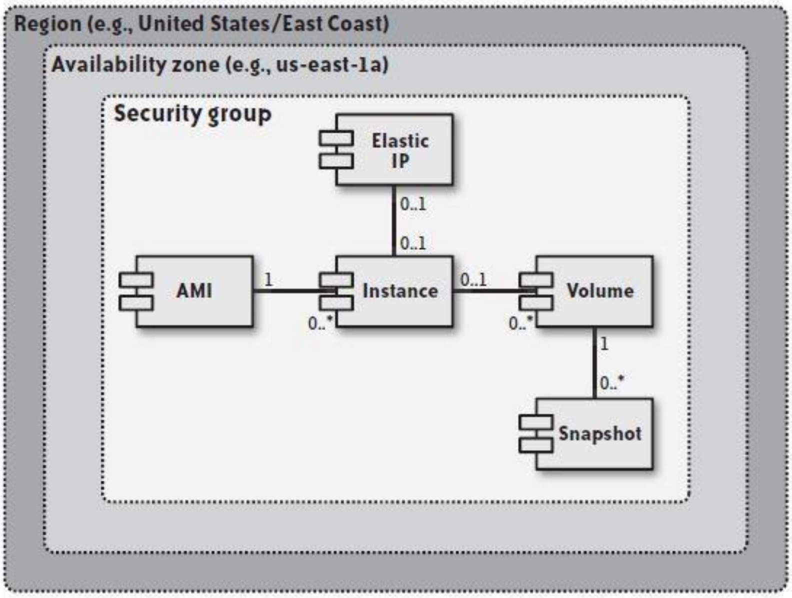


Figure3.3: Amazon ec2 concepts

3.8.1: Instance

An Amazon EC2 instance is a virtual server running your choice of guest operating system based on the machine image from which the instance was created.

3.8.2: Amazon Machine Image (AMI)

A copy of your server that you can use to launch any number of instances. If you are familiar with the concept of ghosting, the machine image represents your ghost image from which you can build any number of servers. Minimally, a machine image will have the core operating system plus common preinstalled tools. Amazon has some prebuilt AMIs to get you started easily.

3.8.3: Elastic IP address

This is simply a static IP address that is assigned to you. (The term “elastic” doesn’t mean a dynamic address, it is a static one) By default, each Amazon instance comes with a dynamically assigned IP address that may be reassigned to another user when your instance terminates. Elastic IP addresses are reserved to you and thus useful for instances that must always be accessible by the same static IP address.

3.9 PaaS: Development & QA

PaaS Development and QA services help companies to improve code quality and development team collaboration, and to accelerate software development and a continuous integration cycle. Many development teams prefer to use the agile software development method over the traditional waterfall model The agile development method assumes high level of collaboration and an intense continuous delivery cycle with frequent code releases, tests, and deployments. Modern development teams are often virtual (or otherwise geographically distributed) and therefore can benefit from PaaS tools that simplify and streamline team collaboration and create a faster time-to-market cycle.

As noted earlier, the software development process typically takes less time than subsequent software testing, maintenance, and support. This is a known problem, and a gap that many PaaS development services work to address. Most PaaS development services are oriented towards an agile development process; this is not to say that these PaaS services are not applicable to the waterfall method, but companies won't likely realize the same level benefits as with agile methodology. Software development and operations include seven fundamental phases plan, code, build, test, release, deploy, and operate. In the agile development process, phases from code to release are highly iterative because code releases are very frequent and testing is continuous.

Many companies use agile methodology and deploy new releases of code several times a week, or even per day: "Slide Share deploys several times a day (between 2 and 20 times… It feels risky at first, but it's actually much less risky than a big bang deployment. If something goes wrong, it's easy to triangulate the cause, and if something doesn't go wrong, it's easy to trust that code" – Director, LinkedIn (Slide Share).

Many frequent small releases can be more efficient than one large cumulative release:

- Faster response to customer feedback about the product
- Small releases are easier to troubleshoot and roll back
- Sustainable development cycle
- Continuous motivation and attention from the team
- Quick adaptation to new requirements and trends

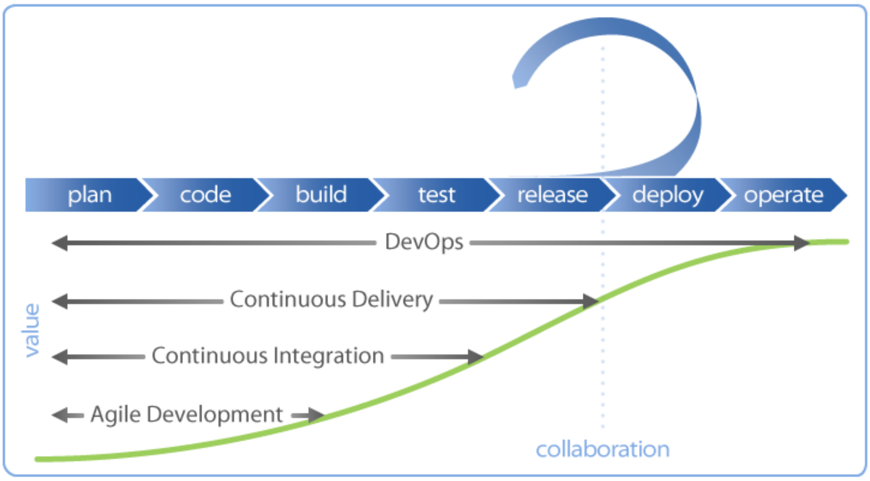


Figure3.4: PaaS Development & QA

In an enterprise organization, each of the development and operation processes has a separate employee role assigned – there are several employees with different functional roles normally involved in the software development and delivery process.

However, such functional segregation of duties often delays the software delivery process, and raises the chances of inadvertent bugs and errors. In highly-segmented teams, development employees don't always feel fully responsible for their actions, and tend to leave specific functions to others. For example, developers may feel that they are not fully responsible for the quality of the code, and focus on pushing their code into the hands of QA engineers in the hope that all bugs will be caught by the subsequent team. Needless to say, such a strategy doesn't work well for the end product. PaaS development platforms like Google AppEngine and Amazon Beanstalk take a continuous delivery approach to the next level by giving more power to developers through automating build, test, release, deployment, and operation processes. we outline continuous delivery functional roles and compare these roles between enterprise on-premise and PaaS development environments.

Delivery phase	Enterprise on premise	Enterprise PaaS	Phase description
Plan	Developer, QA, Operations	Developer	Software architecture and infrastructure design
Develop	Developer	Developer	Software development
Build	Build engineer	Automated by PaaS	Software packaging

Test	QA engineer	Partially automated by PaaS/ Developer	Software testing
Release	Build engineer	Automated by PaaS	Software release (version labeling, deployment preparation)
Deploy	Operations engineer	Automated by PaaS	Software package deployment to production infrastructure
Operate	Operations engineer	Automated by PaaS/Developer	Monitoring, Reporting, Troubleshooting, Failover

Table3.1: PaaS Development & QA

Some may argue that not all software delivery and operations processes can be fully automated, and it is difficult to disagree with that. However, PaaS simplifies these processes to a level that software developers can handle.

That is, you don't need to be an operating system expert or be closely familiar with network routing protocols in order to manage the operations of your infrastructure, because the cloud platform will do that for you. In addition to routine infrastructure operations, PaaS also autoscales infrastructure based on certain application performance conditions. It takes substantial effort and expertise to design such automated scalability internally, while a PaaS platform provides it as part of the service offering. Here, we review the two most popular development PaaS platforms – Google AppEngine and Amazon Beanstalk – and outline the differences between them.

3.10 Google App Engine

Google App Engine (GAE) is a pure PaaS platform that completely abstracts infrastructure services away from developers. Physically, GAE is represented by the web or application server, depending on whether the developer uses Python or Java. Google App Engine fully supports the Python, Java, and Go programming languages. GAE runs within a "sandbox," which isolates and secures operating system processes (see the nodes in Figure 23); so, the GAE web server doesn't have direct access to file system components like network sockets, system calls, and schedulers.

Google wants to take on all infrastructure-related tasks, and therefore doesn't provide access to lower-level OS services. Instead, the service offers a rich set of higher-level APIs, to cover a majority of typical development needs. APIs provide programmable access to a wide set of services, such as SQL and NoSQL databases, Mail, Map Reduce, and Log services.

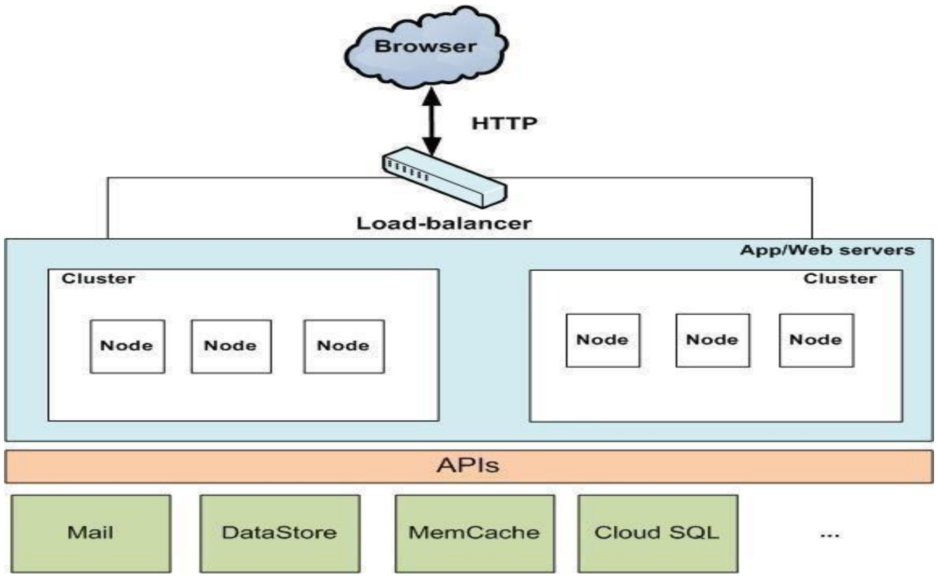


Figure3.5: browser

There is absolutely no system administration or other infrastructure tasks involved, as GAE handles all system routines. GAE continuously monitors application performance and autoscales the environment by adding new nodes to the application cluster. All application-related configurations is done using configuration descriptor files, which are packages within the application and can be written in XML or YAML formats. A GAE application descriptor can configure a variety of features and tasks, such as security, job scheduling, task queuing, URL handling, database indexing, and backend server instances.

Below is an example of a basic application configuration file: This YAML descriptor configures the "myapp" application, version 1, which runs in a Java container. It also defines a handler, who requires authentication to access/admin a URL, and defines that only an admin user can be authenticated.

3.11 Amazon Elastic Beanstalk

Amazon Elastic Beanstalk (AEB) is a PaaS platform built on top of Amazon's IaaS infrastructure. AEB runs applications within EC2 cloud virtual machines – which we discussed in the IaaS: Computation section.The difference between Amazon IaaS offerings and PaaS

AEB is that AEB automatically provisions the infrastructure and reduces the amount of time required for continuous integration tasks.

At the same time, AEB provides full access to the operating system and other lower-level infrastructure components. AEB supports the Java, .NET, PHP, and Python programming languages. To use AEB, you simply upload your application deployment file to the Amazon cloud using Amazon Web Services console or the Git version control repository, then specify which version of the application you want to deploy and which environment (OS, Application server, DB, etc.) you want to run this application against. Once the environment is launched, AEB automatically handles capacity provisioning, auto-scaling, health monitoring, and other infrastructure operations – customers needn't spend time on infrastructure operations. However, if customers *want* to customize the infrastructure configuration, they have full access to all system components.

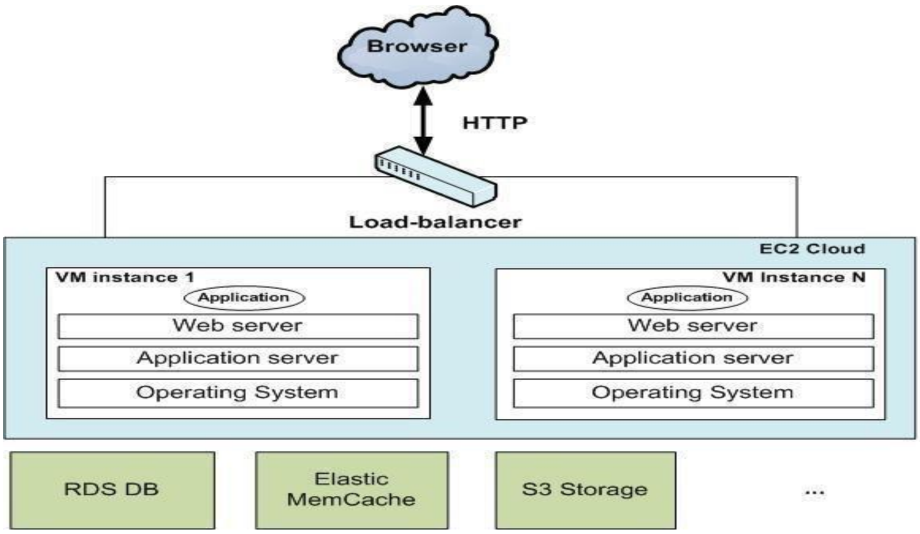


Figure3.6: Amazon Elastic Beanstalk

	Amazon Elastic Beanstalk	**Google App Engine**
Container type	Virtual Machine	Sandbox (process isolation)
Language support	Java, .NET, Python, PHP	Java, Python, GO, partial support for other languages
Customization	Fully customizable. Access to OS services, APIs, etc.	Limited customization. No access to OS services. No custom libraries, or APIs

Ease of use	Good. UI console, configuration descriptors, CLI. Easy to deploy.	Good. AWS UI console, CLI. Access to version control.
Features	All Amazon Web Services or any custom features are available	Only services/APIs included with App Engine, but there are many of them and these are mostly sufficient
Pricing	EC2 pricing: $/number of hours VM running	$/number of apps, number of instances and features
Free	One EC2 micro instance for one year	Generous free quota per day: 28 instance hours, 1GB data storage, 1GB in/out bandwidth
Auto-scaling	Yes	Yes
Monitoring	Yes, UI tools and alerts	Yes, UI tools and alerts
Integration	Integration with any external and internal services	Integration with any Google services

Table3.2: Amazon Beanstalk vs Google App Engine

3.12 Amazon Elastic Beanstalk vs. Google App Engine

Overall, both Amazon Elastic Beanstalk (AEB) and Google App Engine (GAE) provide solid PaaS offerings. These services have fundamentally different technical architectures: Amazon Beanstalk deploys applications within virtual machines, while Google App Engine runs applications within a sandboxed OS process. Both services are very easy to use, and customers can deploy applications within minutes.

From a configuration prospective, GAE might have a slight edge over AEB because it allows specifying all application and environment configuration details in application configuration descriptions. This approach is very common for developers, and requires no learning curve. The learning curve with AEB is potentially a little steeper because developers must understand at least the basics of Amazon Web Services in order to use AEB.

Beanstalk definitely wins in terms of customization and available features. Since it runs within a typical virtualization container, developers have full access to all services and applications as they would on any other server.

GAE instead allows access to internal services only through the provided APIs. GAE prevents customers from accessing the OS directly, and from adding their own API libraries. This may create problems for customers who want to use custom language frameworks or do lower-level programming like thread management or network sockets.

Although customization is definitely a limiting factor, Google nonetheless did a really good job of continuously expanding the list of available APIs and ensuring that these APIs cover most customer needs.

Integration is certainly an advantage of AEB. GAE can integrate only with Google services, but a customer seeking integration with a service that runs outside of Google will encounter challenges. If customers want to use GAE, they likely must move their entire web infrastructure to Google; however, as discussed previously, relying completely on a single vendor is considered a poor strategy because of lock-in potential.

GAE holds some advantage in pricing for small applications because of resource granularity – that is, customers can start very small and benefit from free GAE offerings. AEB is less flexible because it measures computational resources in virtual machines, and the smallest virtual machine costs approximately $40/month. For larger applications, GAE loses its price advantage.

3.13 When to use Google App Engine or Amazon Beanstalk?

I recommend using Google App Engine for small applications that don't have customization or integration requirements. Compared to Amazon Beanstalk, Google App Engine provides more granular pricing options because GAE charges customers per resource utilization, not per virtual machine as Amazon does.

GAE's free quota is generous, and may even allow running small applications for free. Before moving application to Google App Engine, customers must be sure that GAE satisfies all application dependencies.

Amazon Elastic Beanstalk is a better fit for large enterprise applications that typically require integration with multiple services and advanced customizations.

Amazon Elastic Beanstalk (AEB) is a PaaS platform built on top of Amazon's IaaS infrastructure. AEB runs applications within EC2 cloud virtual machines which we discussed in the IaaS Computation section.

The difference between Amazon IaaS offerings and PaaS AEB is that AEB automatically provisions the infrastructure and reduces the amount of time required for continuous integration tasks.

Chapter No:
4 SaaS services

Software as a service is a cloud services delivery model that offers an on-demand online software subscription. As with other cloud delivery models, SaaS offers companies the opportunity to reduce internal IT support costs and transfer maintenance liability to the SaaS provider. SaaS is by far the most widely-used cloud delivery model because almost every software vendor is looking to put its offering on the SaaS rails – there are SaaS offerings in every category of software products, and it would probably take days to list all SaaS software vendors in this paper. Therefore, in our SaaS taxonomy we list only selected groups of vendors in a few categories. In this section, we discuss SaaS trends using Sales force as an example. Sales force is the largest SaaS company, with $2.3 billion in revenue for 2012, representing approximately 15% of total SaaS market revenues “Gartner Says Worldwide Software-as-a-Service Revenue to Reach $14.5 Billion in 2012”).

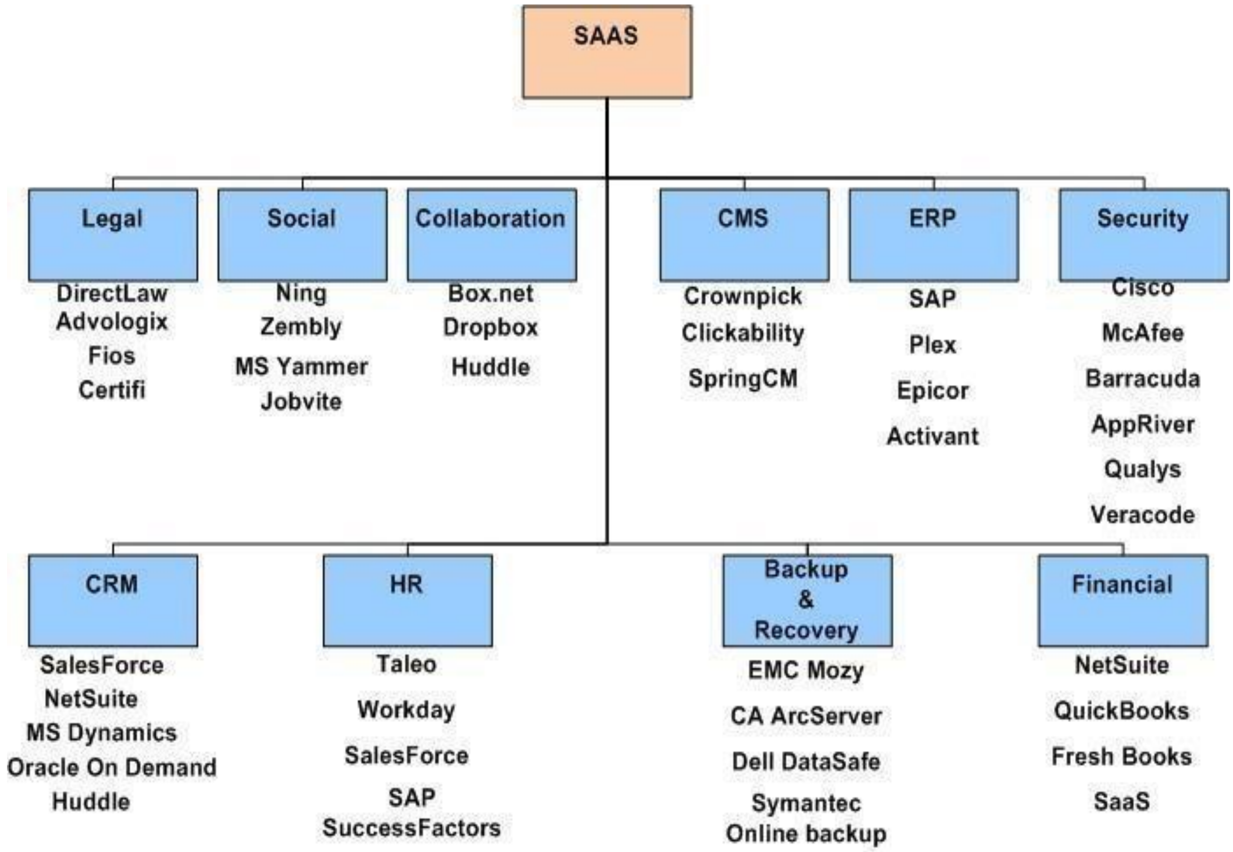


Figure4.1: SaaS services

SaaS is a large market with strong growth potential. Gartner Group estimates that by 2015 worldwide SaaS revenue will reach $22.5 billion A continuation of the ASP (application services provider) model, the mainstream SaaS services appeared at the end of the 1990s when companies like Sales force offered customers outsourced hosting and management of their software. The ASP model also offered centralized third-party application hosting, but was different from SaaS because hosting and operations still required manual involvement. Further, ASPs sometimes required the installation of a thick desktop client (locally running software which performs majority of the computational tasks on the user's computer, not on the server side), while SaaS applications are normally accessed through web browsers or mobile apps.

SaaS platforms employ multi-tenant architectures where the same platform hardware and software is shared among multiple customers. Multi-tenancy hosting and service automation allows SaaS providers to keep service costs low. Depending on the SaaS provider and type of application, some application customization is often available, but this is typically more limited than with PaaS and IaaS cloud solutions.

Initial SaaS ideas caused significant concern related to security, performance, and service availability. However, through the years SaaS technology and business models have significantly matured, and overcome many of these initial worries. Companies now actively use SaaS, and that's unlikely to change.

4.1 SaaS Business Challenge

The main competitive advantage fueling SaaS growth is the ability of SaaS providers to offer high-quality services at cheaper prices than on-premise software vendors. In addition, businesses are attracted to SaaS services because such expenses are treated as operational (OpEx) instead of capital (CapEx). As discussed in previous chapters, companies prefer OpEx over CapEx for the tax advantage.

The SaaS business model may seem perfect – it leaves all control over application management in the hands of SaaS provider. As such, the provider decides when to upgrade software, when to add new features, and when to raise prices.

There is no free lunch, however, and the cost advantage of SaaS business models comes at a price. Lock-in potential can be high with some popular SaaS services, such as CRM or ERP – those familiar with CRM or ERP systems know how difficult it is to switch vendors. Further, many SaaS companies – even those with thousands of customers – may not reach profitability for years after starting the service. The largest SaaS company, Salesforce, had negative net profit margins in the recent quarter (Q2, 2012). As a matter of fact, Sales force hasn't had a quarterly profit margin higher that 10% since 2006, even though company revenues have been steadily growing. By comparison, SAP profit margin in Q2, 2012 was around 17%.

Figure4.2: SaaS Business Challenge

So, why would an SaaS giant like Sales force with 2012 revenues of $2.3 billion per quarter have a negative profit margin? The reason is high customer acquisition costs (i.e., sales and marketing expenses).

The SaaS business model follows the economy-of-scale principle – it is a low-price, high-volume business. Sales force currently has roughly 90,000 business customers, a huge number. To maintain sustainable growth for their business, Sales force must keep customer churn rates low and continuously accelerate the customer acquisition pace. They must also keep their prices low in order to compete with on premise CRM vendors, other SaaS players, and new market entrants.

Let’s compare SAP and Sales force’s financial income statements from fiscal year 2012

	SAP (SAP)	Salesforce (CRM)

Revenues	14.3 billion (EUR)	2.26 billion (USD)
Cost of Goods (% of revenue)	28%	21%
Sales, marketing, and administration	26%	66%
R&D expenses	13%	13%
EBT(earnings before tax)	31%	- 1.5%
NET income	24%	-0.5%

Table4.1: SaaS Business Challenge

In, we can clearly see that "sales and marketing" represents a majority of Sales force's expenses. R&D expenses are approximately equal (in terms of % from revenue) between two companies. Cost of goods is lower for Sales force, which makes sense because their webbased solution is cheaper to develop and support compared to on-premise installed software.

Sales force also operates in a centralized, web-based, multi-tenant infrastructure that doesn't carry some of the expenses typical of on-premise software such as multi-platform support or integration with a variety of internal infrastructure systems.

Could Sales force have survived and become so successful without such a massive investment in sales and marketing?

SaaS and the cloud in general is a relatively new market – it takes time to educate customers and establish a level of trust. Competition is fierce, and although barriers to enter the SaaS market are higher than those of IaaS, they are still not high enough to keep competitors out; SaaS providers must keep prices low to stay competitive. They can minimize their product development and delivery expenses, but at the same time they don't have the same profit opportunities in consulting and professional services as on-premise software vendors do.

Large software vendors typically generate 30-40% of their revenues from services, while Sales force revenues from services make only about 7-8%.

Sales force will certainly have to reduce their investment in sales and marketing at some point when the SaaS market matures and consolidates.

4.2 Sales force Platform Overview

Salesforce.com started as a CRM application in 1999, but with time it grew into a powerful enterprise platform covering many enterprise business and social needs. Salesforce components are closely integrated with each other, and all of them offer API for automation and external integration. The Salesforce CRM system has an easy-to-use web interface for customers, and an API for developers.

Salseforce Force.com in particular is a PaaS development platform that natively supports several programming languages, including Java, PHP, .NET, and Ruby. It can be integrated with most popular cloud services, such as Amazon Web Services, Facebook, and Twitter.

The Salesforce platform (see Figure 28) consists of several components:

4.2.1: Sales cloud - A CRM (customer relationships management) system, which helps to organize customer information and manage sales processes. It helps to close deals faster, to research sales data, and to share all information with members of your team.

4.2.2: Service Cloud – customer service software that allows communication with customers through the contact center and social media

4.2.3: Radian6 – a social media marketing platform for customer engagement and marketing-lead generation

4.2.4: AppExchange – Salesforce's application store. It hosts applications developed by the Salesforce community.

4.2.5: Chatter – the company's private social network, which allows for quickly collaborating with colleagues, discovering resources, and sharing information and content.

4.2.6: Force.com – a PaaS development platform for creating and deploying Salesforce integrated applications to the Salesforce cloud

4.2.7: Heroku – a cloud PaaS hosting platform that can be used to host any application

4.2.8: Data.com – a business data provider and aggregator

4.2.9: Database.com – PaaS database technology with a set of included APIs and security features

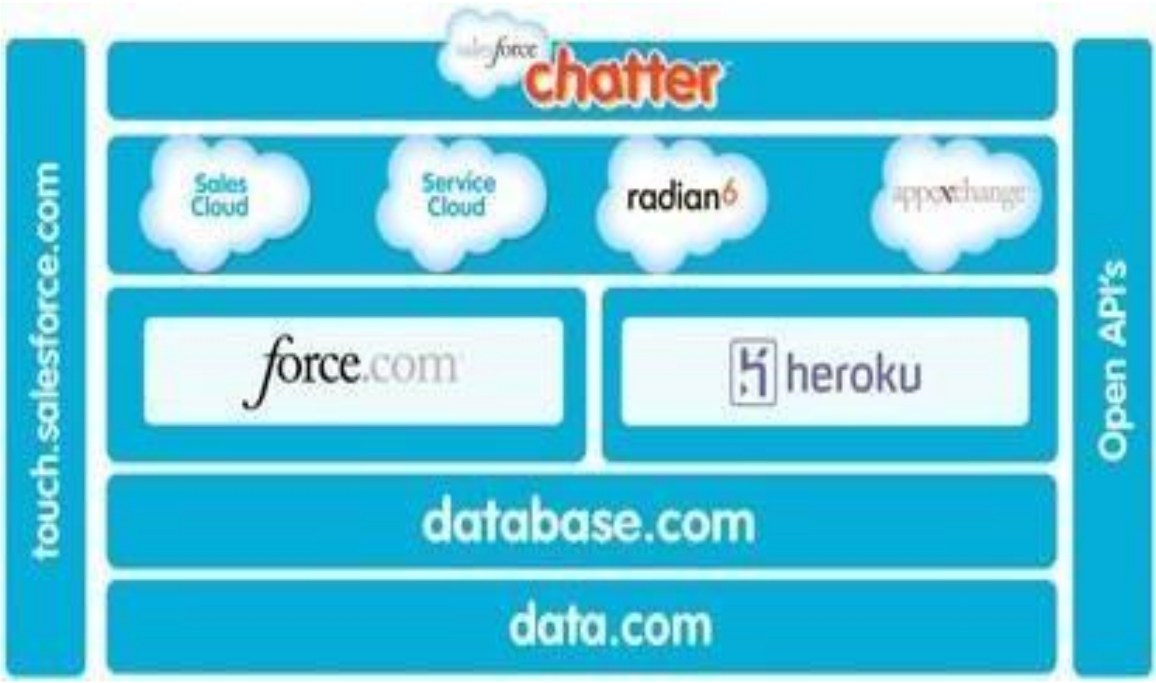


Figure4.3: Sales force Platform Overview

4.3 Is Sales force both SaaS and PaaS?

Sales force has both SaaS and PaaS components; however, as with many other technologies, the line between SaaS and PaaS is very thin. It depends not only on technical characteristics, but also on user perception. From the end-user’s perspective, Salesforce is certainly a SaaS because it has a nice user interface accessible through a browser or a mobile application. From development prospective, though, Salesforce is a PaaS because it provides programmable access through its API.

4.4 The Future of the Cloud Computing Market

Cloud Computing is a disruptive methodology that is rapidly changing how computing is done. When mass adoption of cloud computing services began in 2005 and 2006, several cloud providers achieved an early market lead.

Then the market started to boom around these cloud services, and many companies saw potential and entered the market. Currently, although a few major players lead the market, none of them holds a market lock in terms of technology standards and features.

According to James M. Utter back’s Mastering the Dynamics of Innovation, this is a typical market response to the introduction of a new innovative product. So, what should we expect to happen next in the cloud computing market?

Utter back presents evidence for his hypothesis that the peak in the total number of competing market players occurs shortly after the “dominant design” state of a product is reached. Following the emergence of the dominant design, we can expect a significant number of market exits and consolidations.

By Utter back’s definition, the dominant design of a product is “the one that wins the allegiance of the market place, the one that competitors and innovators must adhere to if they command significant market following” Once a dominant design is determined by market participants, companies tend to use it as a reference standard for their product implementations.

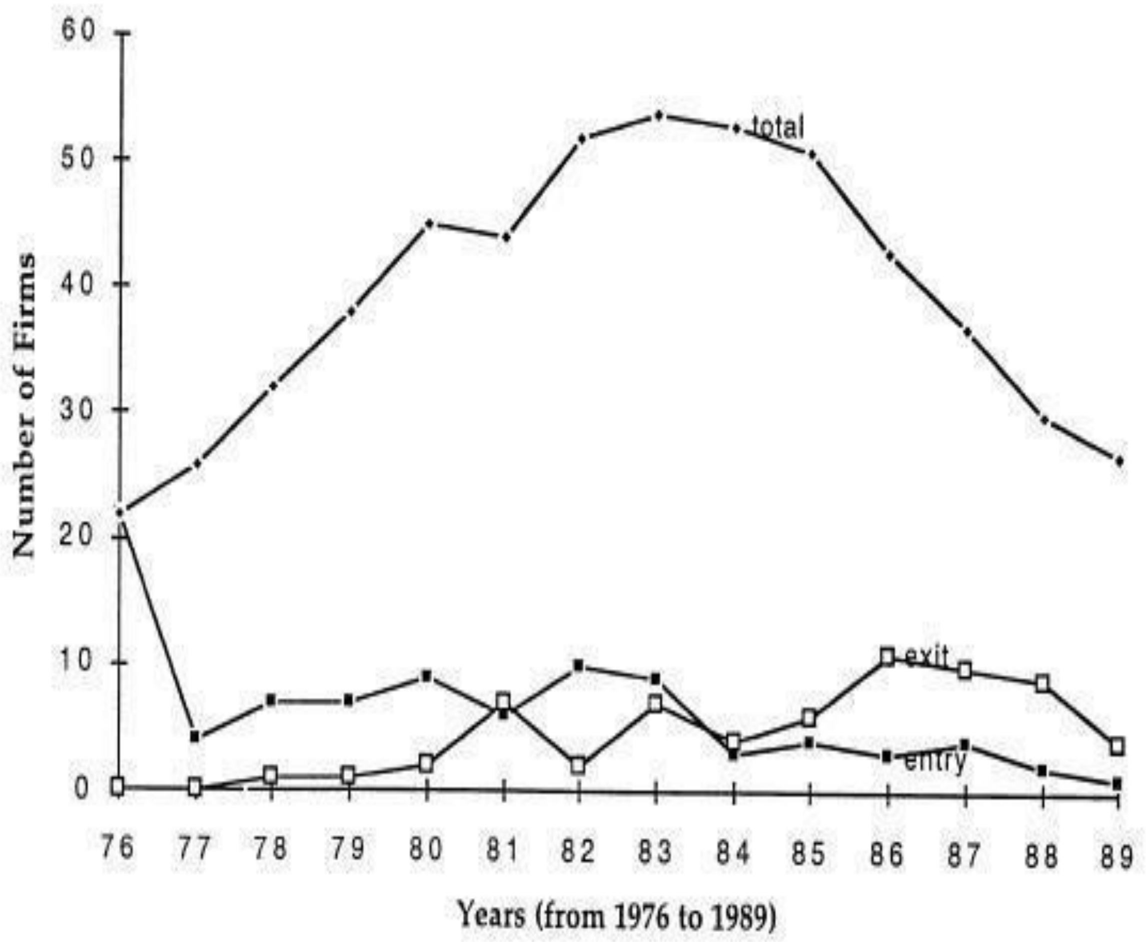
Although there are can be some edge cases, and the dominant design may not necessarily satisfy the entire population of customers, it nonetheless typically covers the needs of the majority. The establishment of a dominant design cannot always be predicted – it depends on a combination of certain technical and business market events in a given timeframe.

To analyze current the cloud computing technology market and hypothesize about future trends, we can draw a historical parallel to compare the disruption caused by cloud computing technology with the past market entrances of other disruptive products.

Introduced by IBM 1973, Winchester disk drives were a disruptive technology for the rest of the decade. As Clayton Christensen presented in The Innovator's Challenge: Understanding the Influence of Market Environment on Processes of Technology Development in the Rigid Disk

Drive Industry there were a total of five generations of Winchester drives: 14-inch, 8-inch, 5¼-inch, 3 ½-inch, and 2½-inch. In the period from 1976 to 1983, many companies followed IBM’s lead and entered the lucrative Winchester drive market.

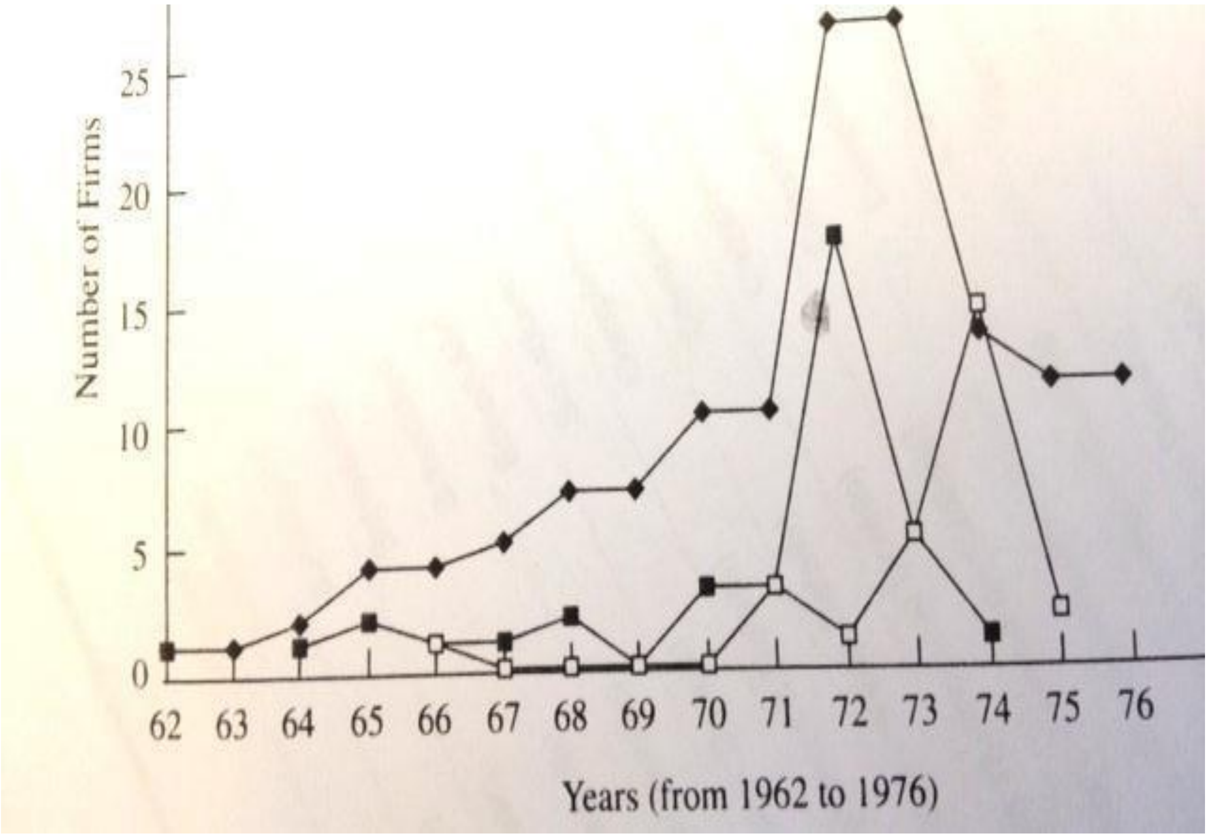
In the first few years, the number of market exits was minimal. The number of companies peaked around 1983, approximately when 3½-inch hard drives began to dominate the market. Once this dominant technology was determined, market exit events accelerated, followed by a number of consolidations and bankruptcies. By the 1990s, the market had standardized on 3½-inch drives, with only a few major vendors surviving the competition.



EXIT **ENTRY** **TOTAL**

Figure4.4: The Future of the Cloud Computing Market

The electronic calculator demonstrated similar market trends The market peaked in 19711972 with 25 participants. After that, exits started because a dominant design was introduced with the invention of the innovative calculator chip, which tremendously simplified the calculator-assembly process and minimized the number of parts.



□ **EXIT** ■ **ENTRY** ♦ **TOTAL**

Figure4.5: The Future of the Cloud Computing Market

Both the Winchester drive and electronic calculator demonstrated the same pattern: the initial introduction of a disruptive product attracted multiple players, but after a dominant product design was chosen by market participants companies that couldn't adopt this design either left the market or merged with other players.

It is worth noting that the dominant product design doesn't always represent the most sophisticated cutting-edge solution – it is simply a solution that fits the majority of customers and provides reasonable investment returns to vendors.

4.5 Public Cloud Domination

Based on market research we can hypothesize with a high level of confidence that the public cloud computing model is likely to become dominant for serving computing resources. According to the market survey in 2011 roughly 28% of surveyed companies employed public cloud services, and by 2014 that share of companies using a public cloud will grow to 51%. That said, this doesn't mean that these companies will stop using private clouds; but, in the long term, private and hybrid cloud market shares are likely to continuously drop, and these cloud models will likely be used only for specific business cases.

At this point, cloud computing competition is heating up – we can assume that the market hasn't yet peaked, but given the high number of market players it is obviously getting closer to that point. Following the market peak, we should expect a large number of consolidations and market exits. After that, the cloud computing market structure should stabilize; with only a few remaining large companies dominating the market.

The current state of cloud computing technology is certainly not final. Cloud technology and related business models will evolve further, and undergo several iterations of product innovation followed by an even longer period of process innovation. Cloud computing standards are still under development, and it will take some time to develop universal standards.

4.6 Open-Source vs. Proprietary Cloud Technologies

Many new cloud market entrants prefer to use open-source technology to enable their platforms. However, more established players like Amazon, Google, and Microsoft employ proprietary systems. The large companies that took an early cloud market lead developed sophisticated proprietary cloud technologies to gain competitive advantage.

It is not feasible for these companies to open-source their technology and give away their technical lead. New market entrants are in the different position of not having time to develop their own technologies to catch up with the market.

One of the major companies using open-source cloud platforms is Hewlett-Packard, which just recently entered the public cloud market in 2012. To quickly build their cloud platform and catch up with other market players, they use the OpenStack open-source cloud platform. In the long run, we expect to see both proprietary and open-source technologies competing in the cloud market.

Potentially, some of the proprietary technologies could be open-sourced later on, once the market has consolidated and stabilized. At the end-market stabilization period, companies will pay more attention to process innovation rather than product innovation.

(Product innovation is the creation of new products or services, or significant improvement of an existing product or service. Process innovation means minor changes or improvements in product or service capabilities, such as a new service delivery method, software feature, and so on.)

4.7 Cloud Delivery Models

For the foreseeable future, all three major delivery models (IaaS, PaaS, and SaaS) are here to stay. It is likely that IaaS market share is going to steadily drop as customers realize that there is more value and resource-savings from software and platform services rather than infrastructure. However, certain technical factors and dependencies will keep IaaS cloud services afloat for some time to come.

We will likely see more consolidations in the IaaS space because the IaaS cloud service market has a very low barrier to entry and minimal technological differentiation factor. As IaaS becomes a commodity, investment returns will drop and cloud providers will be forced to move to the more lucrative PaaS and SaaS segments. Smaller IaaS providers do not have the financial resources to develop sophisticated software to compete in the PaaS or SaaS space, and therefore consolidations and partnerships will be necessary for survival.

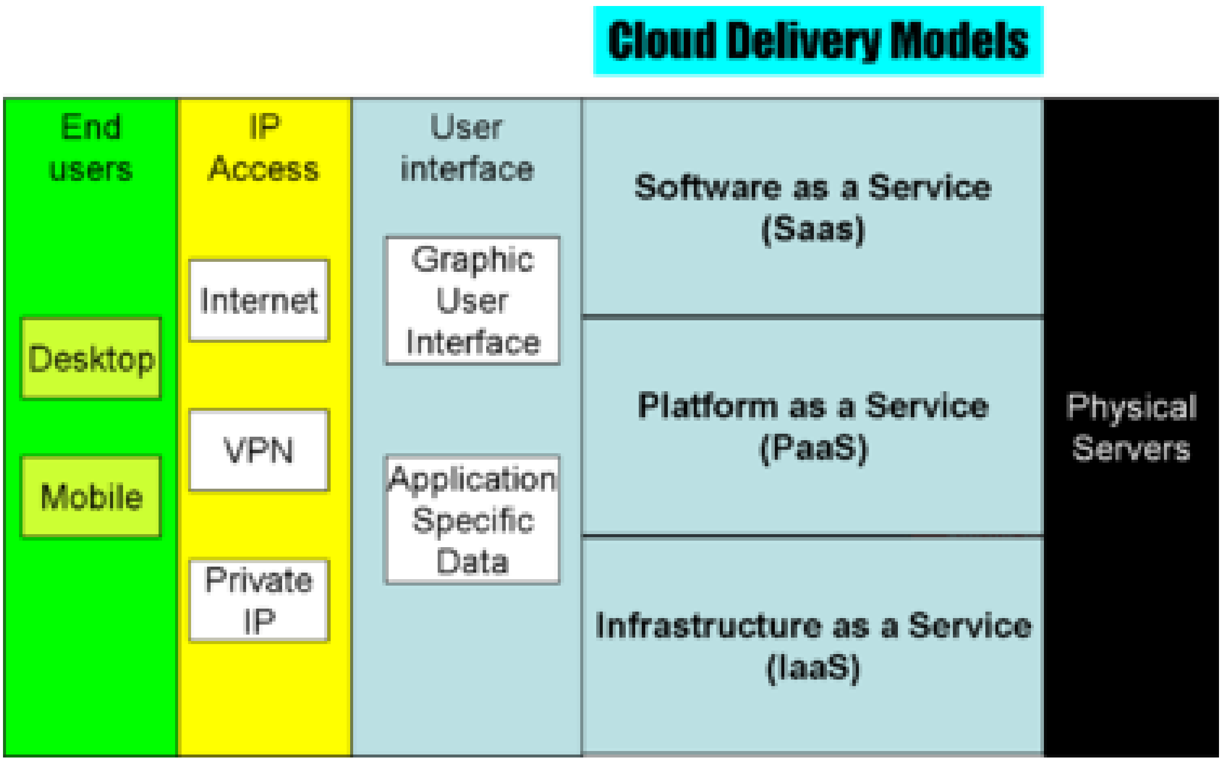


Figure4.6: Cloud Delivery Models

PaaS development and QA services help companies to improve code quality, foster development team collaboration, and accelerate software development and a continuous integration cycle. Many development teams prefer to use the agile software development method over the traditional waterfall model PaaS development platforms and collaboration tools provide companies with substantial time and money savings, and therefore we are likely to see more innovation in this segment.

Cloud business intelligence and data analytics remain one of the most untapped segments in the cloud services market. The primary cloud BI adoption challenge is security.

Several key factors are driving BI into the cloud:

- Higher time-to-value ratio
- OpEx tax structure
- Lack of skills and internal expertise
- Elasticity and pay-as-you-grow options
- Lower maintenance costs

I recommend considering public data analytics and BI offerings in following scenarios:

- Data sources from which the company plans to analyze data are already hosted in the cloud. For example, if you are looking to integrate your BI platform with information from Sales force's CRM system, then it makes sense to build a BI platform in the cloud because the data source is already there.
- The organization is willing to accept the limitations of a chosen BI cloud provider and can clearly set expectations in terms of needed features and potential cloud BI platform customizations.
- The company cannot predict its computational workloads, and therefore will strongly benefit from the cloud's elasticity.
- The company lacks the financial resources to build an internal BI platform, and therefore must rely on a public cloud provider that doesn't require an up-front investment.
- The cloud provider's SLA (service-level agreement) fits the company's policies and needs.
- The company lacks the internal IT resources to deliver a solid BI platform.
- The company knows that all required data sources can be integrated with a cloud BI platform.

4.8 The Thin Line between IaaS, PaaS, and SaaS

By definition, cloud delivery models should be classified by the following characteristics.

SaaS: software applications (not including ones with APIs)

PaaS: execution runtimes without direct access to the OS (databases, development platforms, and application servers)

IaaS: virtual machines, servers, storage, network gear, and any other hardware/OS

In some cases it is difficult to precisely classify a specific cloud computing service. For example, Amazon RDS database services can arguably be attributed to either IaaS or PaaS delivery models. As providers looking to offer better productivity tools for their customers, they are complementing IaaS with PaaS and SaaS services. Therefore, the line between cloud delivery models is becoming blurred.

Sales force offers both SaaS and PaaS components. From an end-user's perspective Salesforce is certainly a SaaS because it has nice user interface accessible through a browser or mobile application. However, from a development prospective Salesforce is a PaaS because it provides programmable access through an API. Similarly, in the case of cloud BI platforms Google's Big Query BI offering could be considered both a SaaS and PaaS solution – it is SaaS for the users who run queries against it, but PaaS for developers who use an API to program it.

As such, we can conclude that cloud delivery model classifications depend not only on technical characteristics, but also on user perception.

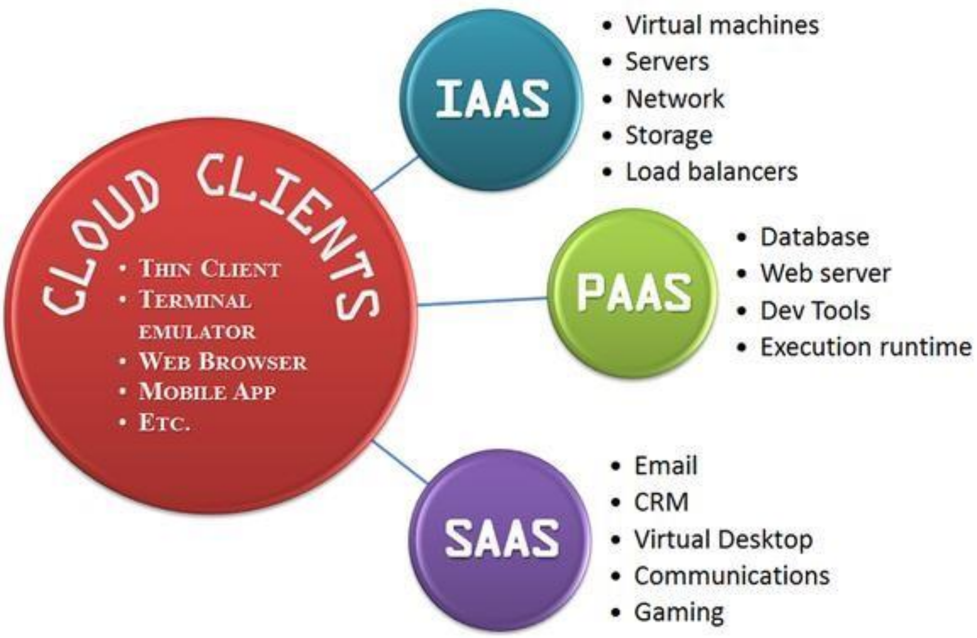


Figure4.7: The Thin Line between IaaS, PaaS, and SaaS

4.9 Cloud Adoption and Control Challenges

Cloud transformation is a lengthy process that involves both technical and organizational challenges. As with the introduction of other disruptive technologies, cloud computing has experienced considerable adoption resistance, and cloud proponents should be prepared to overcome resistance in their organizations. Complexity paired with uncertainty creates organizational cloud adoption barriers.

Security is by far the most serious barrier to the adoption of cloud computing, but there are other adoption challenges as well:

- Data security concerns
- Cost uncertainty
- Loss of control
- Reliability concerns
- Reduced data portability
- Reduced software compatibility
- Reduced performance
- Lock-in fears

Organizational challenges are hard to quantify, but such challenges can be critical in the decision-making process.

4.10 Cloud Services Pricing

Cloud market competition is becoming intense, and providers are pressed to aggressively reduce service prices. A competitive and uncertain market with many existing market players and continuous market entrances leads to high market volatility. In such a volatile market, long-term customer commitments are key for a company's success. Cloud providers offer lucrative deals to lock customers in because of these benefits:

- With long-term customers commitments, providers can better plan their cloud capacity for years ahead. The ability to forecast capacity helps providers to make correct capital expenditure (CapEx) decisions.

- There are low barriers to enter some cloud market segments (such as IaaS), and entrants periodically disrupt the market by offering new technologies and competitive processes. This creates a challenge for established cloud providers, and makes it more difficult to keep customers. Prepaid services ensure long-term customer commitment.

- Providers want to prevent customers from using the services of other providers.

Therefore, in the current market customers can secure much better prices if they commit to a provider (see Appendix A). In many cases, the price difference between pay-as-you-go and prepaid pricing can be very large.

The SaaS model follows the economy-of-scale principle – it is a low-price, high-volume business. The customer acquisition process is expensive. Salesforce and other SaaS providers have low profit margins because they must invest heavily in the sales and marketing of their products. They will have to reduce this investment in sales and marketing at some point when the SaaS market matures and consolidates.

To provide sustainable growth for their business, SaaS providers must keep customer churn rates low and continuously accelerate the customer acquisition pace. They must also keep their prices low to compete with on premise CRM vendors, other SaaS players, and new market entrants.

Conclusion:

To summarize, the cloud provides many options for the everyday computer user as well as large and small businesses. It opens up the world of computing to a broader range of uses and increases the ease of use by giving access through any internet connection. However, with this increased ease also come drawbacks. You have less control over who has access to your information and little to no knowledge of where it is stored. You also must be aware of the security risks of having data stored on the cloud. The cloud is a big target for malicious individuals and may have disadvantages because it can be accessed through an unsecured internet connection.

If you are considering using the cloud, be certain that you identify what information you will be putting out in the cloud, who will have access to that information, and what you will need to make sure it is protected. Additionally, know your options in terms of what type of cloud will be best for your needs, what type of provider will be most useful to you, and what the reputation and responsibilities of the providers you are considering are before you sign up.

References:

1. Ray j Rafael's. Cloud Computing: Finding the Silver Lining, Not the Silver Bullet.
2. Cloud computing from beginning to end (2015)
3. Ray j Rafael's. Basics about Cloud Computing.
4. A hand-on approach (2013).
5. By arshdeep bagha and vijay madisetti. Guidelines on Security and Privacy in Public Cloud Computing. National Institute of Standards and Technology.
6. Stroud, Harrison & Lewis, Grace. T-Check in System-of-Systems Technologies: Cloud Computing (CMU/SEI-2010-TN-009). Software Engineering Institute, Carnegie
7. Cloud computing concepts, technology & architchere and cloud computing design patterns 2013 June and 2015
8. By thonas Erl and others. The Challenge of the Computer Utility. Addison-Wesley Educational Publishers Inc., US, 1966.
9. HAMILTON, J. Internet-Scale Service Efficiency. In Large-Scale Distributed Systems and Middleware (LADIS) Workshop (September 2008).
10. above the Clouds: A Berkeley View of Cloud Computing. Michael Armbrust,
11. Architecting for the Cloud: Best Practices, Jinesh Varia , January 2010
12. "Cloud Computing Takes Off", Morgan Stanley, May 23, 2011
13. Amazon web services for dummies (2013)
14. By Bernard golden
15. One of the first guides and leading authorities about subject
16. Cloudonomics: the business value of cloud computing (2012)