# Cloud computing

* Amazon EC2/S3
* IBM’s Blue Cloud
* Sun Network.com
* Microsoft Azure Services Platform
* Numbus
* OpenNEbula
* Eucalyptus
* OpenQRM
* RESERVOIR
* Salesforce
* RightScale
* Microsoft SharePoint Services
* Microsoft Window Live Service
* Apple Mobile Me

<http://www.webopedia.com/quick_ref/cloud_computing.asp>

* Cloud computing is taking «cloud services» and moving them outside an organizations firewall on shared systems
* Applications and services are accessed via the Web, instead of your hard drive
* A private cloud is the same platform (as public cloud) but is implemented within the corporate firewall
* Describes too many technologies – grid computing, utility computing, software-as-a-service, internet-based applications, autonomic computing, peer-to-peer computing and remote processing

<http://www.webopedia.com/quick_ref/cloud_computing_terms.asp>

* XaaS – Anything as a service
* Apache CloudStack – open source cloud computing and IaaS platform
* Desktop-as-a-service – virtual desktop infrastructure
* Eucalyptus – open source cloud computing and IaaS platform – private clouds
* OpenStack Grizzly – open source clout computing platform
* Red Hat – IaaS that builds upon a collection of more than 60 open source projects

<http://home.cse.ust.hk/~weiwa/teaching/Fall15-COMP6611B/reading_list/AboveTheClouds.pdf>

* The illusion of infinite computing resources available on demand, thereby eliminating the need for users to plan far ahead for provisioning
* The elimination of an up-front commitment by users, thereby allowing companies to start small and increase hardware resources only when there is an increase in their needs
* The ability to pay for use of computing resources on a short-term basis as needed and release them as needed, thereby rewarding conservation by letting machines and storage go when no longer useful
* Amazon EC2 – users can control nearly the entire software stack; Google AppEngine – traditional web applications; Microsoft Azure – in between
* Cloud computing is for software what foundries was for hardware – enabled “fab-less” semiconductor companies
* Cloud Computing allows deploying SaaS—and scaling on demand—without building or provisioning a datacenter
* Real time information services benefit from cloud computing as they require large datasets and must be highly available

<http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6203493>

* Internet of Services (IoS)
  + Service elasticity resulting from clouds’ ability to automatically scale services and infrastructures
  + Cost reduction when adapted to service demands
  + Pay-per-use models allows paying for actual resource consumption
  + Improved time-to-market for services
  + Increased service availability
  + Cloud interoperability – services on multiple clouds providing unlimited scalability
* SaaS – on-demand access to any application
* PaaS – platform for building and delivering applications
* IaaS – on-demand compute/network/storage infrastructures
* Service auto-scaling for varying workloads
* Payment per service vs. payment per user
* *Distributed service – facilitates communication among devices, logic and applications in the IoT.* <http://www.dglogik.com/component/tags/tag/15-distributed-services-architecture>
* *Runs on multiple computers within a network at the same time and can be stored on servers or with cloud computing* <http://searchitoperations.techtarget.com/definition/distributed-applications-distributed-apps>
* Cloud bursting/hybrid clouds on premise infrastructures combined with public clouds to provide extra capacity during peak hours
* Cloud brokers help users choose the most suitable cloud
* Cloud providers should unify APIs on the future - the Open Cloud Computing Interface (OCCI; http://occi-wg.org)
* Improving availability, reliability and resiliency
* Energy efficiency

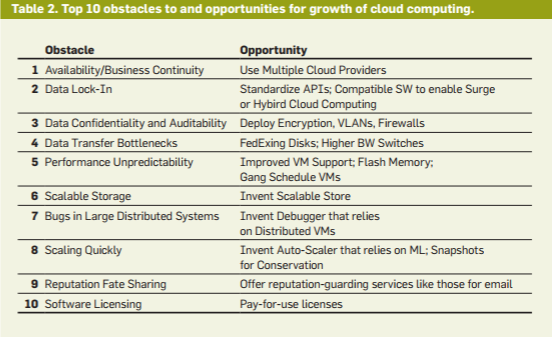
<http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6735443>

* Predicting activity pattern via SPEED and representing actions in a finite-order Markov model and partial matching to improve prediction accuracy
* Cloud computing provides scalable computing and storage power for developing, maintaining, and running home services. In addition, using cloud computing allows the user to access (monitor and/or control) home devices anytime and anywhere
* Database/Data Store: stores data from microcontroller-enabled sensors and Cloud services for data analysis and visualization, and serves as command queue being sent to actuators as well.
* Server/API layer between the back end and the front end: facilitates processing the data received from the sensors and storing the data in database. It also receives commands from the web application client to control the actuators and stores the commands in database. The actuators make requests to consume the commands in the database through the server.
* Web application serving as Cloud services: enable to measure and visualize sensor data, and control devices using a mobile device (e.g., smart phone).
* Google App Engine platform – simple to use, scalable to service requests, and has built-in data store and a flexible interface
* Our Web applications is categorized into two main parts: a front-end and a back-end. The front-end serves as a Web client interacting with the user. The back-end serves as computing services for logic processing or storage services for data storing

<http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6704815>

* Cloud – a metaphor for the Web as a space where computing has been preinstalled and exists as a service. OS, applications, storage, data and processing capacity all exist on the web, ready to be shared among users.
* Main objective of cloud computing – make better use of distributed resources and solve large-scale computing problems. Distributed system that offers computing services via a computer communication network (Internet)
* Resources in the cloud are transparent to the users, and the users need not know the exact location(s) of the resources. They can be shared among a large number of users, who should be able to access applications and data from anywhere at any time.
* IaaS – delivery of huge computing resources such as capacity of storage, processing and network; availability to remotely access computing resources; pay per use, security, reliability; aka Hardware-as-a-Service
* PaaS – supports a set of application programs interface to cloud applications
* SaaS – provides a service that is directly consumable by the end user; pay-as-you-go; seeks to replace apps running on a PC
* Biggest challenges are privacy and security, latency, lack of standards

<http://delivery.acm.org/10.1145/1730000/1721672/p50-armbrust.pdf?ip=129.241.222.226&id=1721672&acc=OPEN&key=CDADA77FFDD8BE08%2E5386D6A7D247483C%2E4D4702B0C3E38B35%2E6D218144511F3437&CFID=808242841&CFTOKEN=84218007&__acm__=1505208361_2dd4f654325c0a39e9f1b3b55ae4acd7>

* Grid computing – protocols to offer shared computation and storage over long distances
* Public clouds provide utility computing as a service
* Private cloud is an internal data center of a business/organization not available to the public
* New aspects of cloud computing
  + The appearance of infinite computing resources available on demand – no need to plan far ahead
  + The elimination of an up-front commitment by cloud users – ability to increase hardware resources when needed
  + The ability to pay for use of computing resources on a short-term basis
* construction and operation of extremely large-scale, commodity-computer data centers at low-cost locations was the key necessary enabler of cloud computing, for they uncovered the factors of 5 to 7 decrease in cost of electricity, network bandwidth, operations, software, and hardware available
* Any application needs a model of computation, storage and communication
* Amazon EC2 looks much like physical hardware and lets users control nearly the entire software stack from kernel upward – difficult to offer automatic scalability and failover because state management and semantics associated with replication are application-dependent – hardware virtual machines
* Google AppEngine is a domain-specific platform which is targeted exclusively at traditional Web applications, enforcing an application structure of clean separation between a stateless computation tier and a stateful storage tier – automatic scaling and high-availability mechanisms – application framework
* Applications for Microsoft’s Azure are written using the .NET libraries, and compiled to the Common Language Runtime, a language-independent managed environment. The framework is significantly more flexible than AppEngine’s, but still constrains the user’s choice of storage model and application structure
* Section about economics
* 
* Applications software needs to both scale down rapidly as well as scale up, which is a new requirement. Such software also needs a pay-for-use licensing model to match needs of cloud computing.
* Infrastructure software must be aware that it is no longer running on bare metal but on VMs. Moreover, metering and billing need to be built in from the start
* Hardware systems should be designed at the scale of a container (at least a dozen racks), which will be the minimum purchase size. Cost of operation will match performance and cost of purchase in importance, rewarding energy proportionality5 by putting idle portions of the memory, disk, and network into low-power mode. Processors should work well with VMs and flash memory should be added to the memory hierarchy, and LAN switches and WAN routers must improve in bandwidth and cost

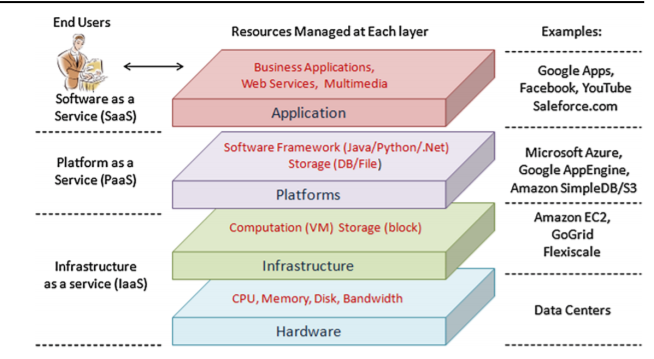
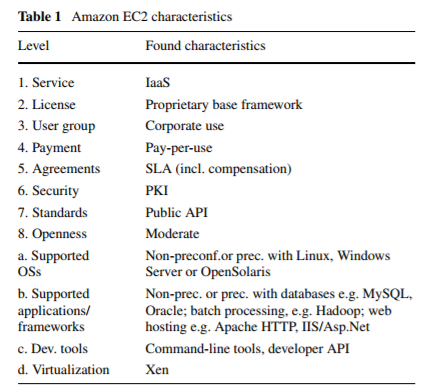
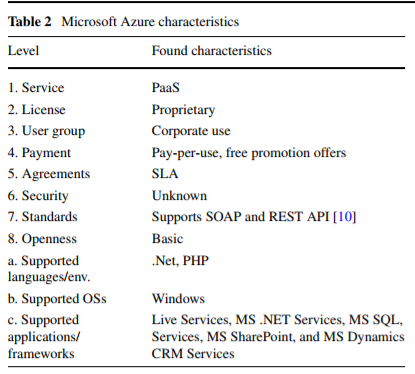
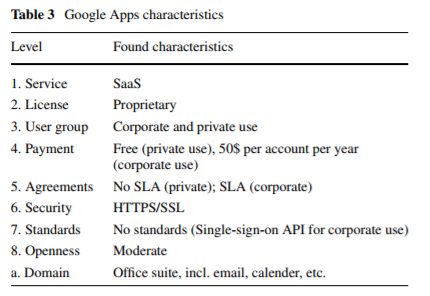
<http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5474674>

* Cloud computing is a model for enabling convenient, ondemand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction - P. Mell and T. Grance, "Draft nist working definition of cloud computing - v15," 21. Aug 2009, 2009
  + On-demand self-service
  + Broad network access
  + Resource pooling
  + Rapid elasticity
  + Measured Service
* Software as a Service (SaaS). Cloud consumers release their applications on a hosting environment, which can be accessed through networks from various clients (e.g. web browser, PDA, etc.) by application users. Cloud consumers do not have control over the Cloud infrastructure that often employs a multi-tenancy system architecture, namely, different cloud consumers' applications are organized in a single logical environment on the SaaS cloud to achieve economies of scale and optimization in terms of speed, security, availability, disaster recovery, and maintenance. Examples of SaaS include SalesForce.com, Google Mail, Google Docs, and so forth.
* Platform as a Service (PaaS). PaaS is a development platform supporting the full "Software Lifecycle" which allows cloud consumers to develop cloud services and applications (e.g. SaaS) directly on the PaaS cloud. Hence the difference between SaaS and PaaS is that SaaS only hosts completed cloud applications whereas PaaS offers a development platform that hosts both completed and in-progress cloud applications. This requires PaaS, in addition to supporting application hosting environment, to possess development infrastructure including programming environment, tools, configuration management, and so forth. An example of PaaS is Google AppEngine.
* Infrastructure as a Service (IaaS). Cloud consumers directly use IT infrastructures (processing, storage, networks, and other fundamental computing resources) provided in the IaaS cloud. Virtualization is extensively used in IaaS cloud in order to integrate/decompose physical resources in an ad-hoc manner to meet growing or shrinking resource demand from cloud consumers. The basic strategy of virtualization is to set up independent virtual machines (VM) that are isolated from both the underlying hardware and other VMs. Notice that this strategy is different from the multi-tenancy model, which aims to transform the application software architecture so that multiple instances (from multiple cloud consumers) can run on a single application (i.e. the same logic machine). An example of IaaS is Amazon's EC2
* Data storage as a Service (DaaS). The delivery of virtualized storage on demand becomes a separate Cloud service - data storage service. Notice that DaaS could be seen as a special type IaaS. The motivation is that on-premise enterprise database systems are often tied in a prohibitive upfront cost in dedicated server, software license, post-delivery services, and in-house IT maintenance. DaaS allows consumers to pay for what they are actually using rather than the site license for the entire database. In addition to traditional storage interfaces such as RDBMS and file systems, some DaaS offerings provide table-style abstractions that are designed to scale out to store and retrieve a huge amount of data within a very compressed timeframe, often too large, too expensive or too slow for most commercial RDBMS to cope with. Examples of this kind of DaaS include Amazon S3, Google BigTable, and Apache HBase, etc.
* Private/community/public/hybrid cloud
* Amazon Web Services launced Virtual Private Cloud (VPC) – seamless bridge between an organizations existing IT infrastructure and the Amazon public cloud. – private because secure connection between IT legacy and the cloud and AWS dedicates “isolated” resources for the VPC

<https://s3.amazonaws.com/academia.edu.documents/7299777/cloud%20computing%20a%20study%20of.pdf?AWSAccessKeyId=AKIAIWOWYYGZ2Y53UL3A&Expires=1505211696&Signature=g77BxNPv2nhEKyd1QyiULsBALz8%3D&response-content-disposition=inline%3B%20filename%3DCloud_computing_A_study_of_infrastructur.pdf>

* Typical cloud computing providers deliver common business applications online which are accessed from another web service or software like a web browser, while the software and data are stored on servers.
* computing capability that provides an abstraction between the computing resource and its underlying technical architecture (e.g., servers, storage, networks), enabling convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction."
* List of benefits
* SaaS - This is the idea that someone can offer you a hosted set of software (running on a platform and infrastructure) that you don't own but pay for some element of utilization - by the user, or some other kind of consumption basis. Here you don't have to do any development or programming, but you may need to come in and configure the (very flexible, configurable and sometimes customizable) software. You don't have to purchase anything. You just pay for
* PaaS - This is the idea that someone can provide the hardware (as in IaaS) plus a certain amount of application software - such as integration into a common set of programming functions or databases as a foundation upon which you can build your application. Platform as a Service (PaaS) is an application development and deployment platform delivered as a service to developers over the Web
* IaaS - is the delivery of hardware (server, storage and network), and associated software (operating systems virtualization technology, file system), as a service. Unlike PaaS services, the IaaS provider does very little management other than keep the data center operational and users must deploy and manage the software services themselves just the way they would in their own data center
  + Utility computing service and billing model
  + Automation of administrative tasks
  + Dynamic scaling
  + Desktop virtualization
  + Policy-based services
  + Internet connectivity
* Main disadvantages: security, service management to ensure SLA, control over access and policies, pricing

<https://link.springer.com/content/pdf/10.1007%2Fs13174-011-0027-x.pdf>

* Lack of standardization of cloud computing services, and each cloud service provider uses different technologies, protocols, and formats. Further, most clouds are very vague about the actual internal workings. All this makes interoperability when working with multiple services or migrating to new services difficult.
* The cloud itself consists “of a collection of interconnected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resource(s)”
* Underlying infrastructure and software is abstracted and offered as a service. Build on a scalable and flexible infrastructure. Offers on-demand service provisioning and quality of service (QoS) guarantees. Pay for use of computing resources without up-front commitment by cloud users. Shared and multitenant. Accessible over the Internet by any device
* 
* Three service models are currently being differentiated—Software-as-a-Service (SaaS), i.e., online applications, such as web-based email, Platform-as-a-Service (PaaS), which allows customers to deploy their own applications, and Infrastructure-as-a-Service (IaaS), which provides, for example, processing power or storage
* A pioneer in virtualization and computing power offerings is Amazon. The Amazon Elastic Compute Cloud (EC2) is one of the most widely used infrastructure platforms. Further popular virtualization services include ServePath’s GoGrid and the Rackspace Cloud. Other services are the IBM Smart Business cloud solutions, Oracle Cloud Computing, GigaSpaces, RightScale, and Nimbus
* Amazon also provides special database solutions, such as the Amazon SimpleDB
* PaaS providers offer a managed higher-level software infrastructure, where customers can build and deploy particular classes of applications and services using the tools, environments, and programming languages supported by the provider. The offers include the use of the underlying infrastructure, such as servers, network, storage, or operating systems, over which the customers have no control, as it is abstracted away below the platform
* Platform services are mostly aimed at specific domains, such as the development of web applications, and are dependent on the programming language. Customers get a separated environment to test and develop or to permanently deploy their applications. Google’s App Engine is targeted at traditional web applications offering a Java or Python environment
* Cloud software offerings typically provide specific, already-created applications running on a cloud infrastructure. A very well-known SaaS is the web-based e-mail. Most software cloud computing services are web-based applications, which can be accessed from various client devices through a thin client interface, such as a web browser. The customers of these services do not manage or control the underlying infrastructure and application platform; only limited userspecific configurations are possible. Features in standard nonremote software applications providing Internet-based storage are also often considered to be part of SaaS offerings.
* The cloud’s internal security considers the mechanisms used to protect the cloud customers’ virtual instances and data within the cloud. Due to the low-level of IaaS, the customer has most control over the security compared with PaaS and SaaS. When using PaaS, the customers may be able to craft their own authentication system or adapt other parts of the system. However, below the application level, security is dealt by the provider, who often gives little or no information about their practices. When using SaaS, the user must rely even more on the provider to implement sufficient security mechanisms
* 
* 
* 

[https://aws.amazon.com](https://aws.amazon.com/documentation/?nc2=h_ql_d)

* Amazon Web Services
* Cloud computing provides a simple way to access servers, storage, databases and a broad set of application services over the Internet. A Cloud services platform such as Amazon Web Services owns and maintains the network-connected hardware required for these application services, while you provision and use what you need via a web application.
* Benefits:
  + Trade capital expense for variable expense – only pay for resources you consume
  + Benefit from massive economies of scale – more users, lower prices
  + Stop guessing capacity – scale up and down as required
  + Increase speed and agility – reduced time it takes to get resources available
  + Stop spending money on running and maintaining data centers
  + Go global in minutes
* AWS offers a broad set of global compute, storage, database, analytics, application, and deployment services that help organizations move faster, lower IT costs, and scale applications – over 50 services are available
  + Compute: EC2 – virtual servers in the Cloud, among others
  + Storage: S3 – scalable storage in the Cloud, among others
  + Database: Aurora – High Performance Managed Relational Database; DynamoDB – Managed NoSQL Database; RDS – Managed Relational Database Service for MySQL etc., among others
  + Internet of Things: AWS IoT Platform, Greengrass, IoT Button – connect Devices to the Cloud
    - Although things, Internet and connectivity are the three core components of IoT, the value is in closing the gap between the physical and digital world in self-reinforcing and self-improving systems
    - With AWS IoT, you can filter, transform, and act upon device data on the fly. You can update your rules to implement new device and application features at any time.
  + Other products: migration, networking and content delivery, developer tools, management tools, security identity & compliance, analytics, artificial intelligence, mobile services, application services, messaging, business productivity, desktop & app streaming, contact center, game development
* AWS Free Tier (12 months) – EC2, S3, RDS, IoT
* IaaS – contains the basic building blocks for cloud OT and typically provide access to networking features, computers, and data storage space. Provides the highest level of flexibility and management control.
* PaaS – remove the need for organizations to manage the underlying infrastructure and allow you to focus on the deployment and management of your applications. Increases efficiency.
* SaaS – provides you with a completed product that is run and managed by the service provider. Usually refers to end-user applications. No need to think about how the service is maintained or how the underlying infrastructure is managed. (web-based email)
* Used by companies like Spotify, Airbnb, yelp, Shazam
* Security in the cloud is recognized as better than on-premises.

<https://azure.microsoft.com>

* 42 regions – more than any cloud provider
* Popular products – total 75 products
  + Virtual Machines
  + App Service
  + SQL Database
  + Storage
* Cloud Platform consists of a set of physical assets, such as computers and hard disk drives, and virtual resources, such as virtual machines, that are contained in Google’s data centers around the globe.
  + This distribution of resources provides several benefits, including redundancy and reduced latency by locating resources closer to clients.
* In cloud computing, what you might be used to thinking of as software and hardware products, become services. These services provide access to the underlying resources.
* Cloud Platform gives you options for computing and hosting. You can choose to work with a managed application platform, leverage container technologies to gain lots of flexibility, or build your own cloud-based infrastructure to have the most control and flexibility.
  + Managed Application Platform (App Engine) – PaaS – Google handles most of the management of resources, automatically scales system to provide need resources.
  + Cluster management (Container Engine) – focus on application code, instead of on deployments and integration into hosting environments. Gives you the flexibility of on-premises or hybrid clouds.
  + Unmanaged infrastructure (Compute Engine) – IaaS – Virtual machines. System provides a robust computing infrastructure, but you must choose and configure the platform components that you want to use. Google ensures that resources are available, reliable and ready to use, but it’s the user’s responsibility to configure, administer, and monitor the systems. User has complete control of the systems and unlimited flexibility.

<https://www.ibm.com/cloud-computing>

* SaaS – Cloud-based applications run on distant computers “in the cloud” that are owned and operated by others and that connect to users’ computers via the internet and, usually, a web browser.
  + Benefits: sign up at rapidly start using innovative business apps; apps and data are accessible from any connected computer; no data is lost if your computer breaks; the service can dynamically scale to usage needs
* PaaS – provides a cloud-based environment with everything required to support the complete lifecycle of building and delivering web-based (cloud) applications – without the cost and complexity of buying and managing the underlying hardware, software, provisioning and hosting
  + Benefits: develop applications and get to market faster; deploy new web applications to the cloud in minutes; reduce complexity with middleware as a service
* IaaS – provides companies with computing resources including servers, networking, storage, and data center space on a pay-per-use basis
  + Benefits: No need to invest in your own hardware; infrastructure scales on demand to support dynamic workloads; flexible, innovative services available on demand
* Public cloud – are owned and operated by companies that offer rapid access over a public network to affordable computing resources. With public cloud services, users don’t need to purchase hardware, software, or supporting infrastructure, which is owned and managed by providers.
  + Innovative SaaS business apps; flexible, scalable IaaS for storage and compute services on a moment’s notice; Powerful PaaS for cloud-based applications development and deployment environments
* Private cloud – infrastructure operated solely for a single organization, whether managed internally or by a third party, and hosted either internally or externally. Private clouds can take advantage of cloud’s efficiencies, while providing more control of resources and steering clear of multi-tenancy
  + Self-service interface controls services, allowing IT staff to quickly provision, allocate and deliver on-demand IT resources; highly automated management of resource pools for everything from compute capability to storage, analytics, and middleware; sophisticated security and governance designed for a company’s specific requirements
* Hybrid cloud – uses a private cloud foundation combined with the strategic integration and use of public cloud services. The reality is a private cloud can’t exist in isolation from the rest of a company’s IT resources and the public cloud. Most companies with private clouds will evolve to manage workloads across data centers, private clouds, and public clouds—thereby creating hybrid clouds.
  + Allow companies to keep the critical applications and sensitive data in a traditional data center environment or private cloud; enables taking advantage of public cloud resources like SaaS, for the latest applications, and IaaS, for elastic virtual resources; facilitates portability of data, apps and services and more choices for deployment models
* The IBM Cloud architecture is built so that you can easily apply AI and machine learning capabilities to your enterprise workloads. Using these integrated cognitive capabilities, you can set yourself apart from the competition with deeper understanding and virtually unmatched intelligence

<https://www.rackspace.com/cloud/cloud-computing/difference> The Difference between private and public cloud, RACKSPACE US, INC

* Public clouds - These are based on shared physical hardware which is owned and operated by a third-party provider. Public clouds are ideal for small and medium sized businesses or businesses that have fluctuating demands. The primary benefits of the public cloud include the speed with which you can deploy IT resources and the ability to pay only of the server resources you use. By spreading infrastructure costs across a number of users, each can take advantage of a low-cost, pay-as-you-go approach to IT provisioning. And, due to the sheer size of public clouds, you can scale compute power up and down as business demands, within a matter of minutes.
* Private clouds - A private cloud is infrastructure dedicated entirely to your business that’s hosted either on-site or in a service provider’s data center. The private cloud delivers all the agility, scalability and efficiency of the public cloud, but also provides greater levels of control and security, making it ideal for larger businesses or those with strict data, regulation and governance obligations. Another key benefit of private cloud is the ability to customize the compute, storage and networking components to best suit your specific IT requirements — something that cannot be achieved so easily in the public cloud environment.
* Hybrid cloud - Getting the right cloud solution no longer needs to be a choice of public or private. With the hybrid cloud, you get the best of both. The hybrid cloud allows you to combine public cloud with private cloud or dedicated hosting and leverage the best of what each has to offer to meet your needs. Use the public cloud for non-sensitive operations, the private cloud for business-critical operations, and incorporate any existing dedicated resources to achieve a highly flexible, highly agile and highly cost-effective solution.

<https://search.proquest.com/docview/1624016976?pq-origsite=gscholar>

* IaaS is the pillar of which a cloud computing architecture is built. It is a highly efficient platform to construct SaaS and PaaS layer on top of it.

<https://link.springer.com/content/pdf/10.1007%2F978-3-642-20917-8.pdf>

* Amazon offers virtualized resources for generic use while the clouds provided by Google and Microsoft allow application hosting. With every provider striving for a competitive edge, all current offerings are indeed proprietary and there are no standards so that, in general, a quick provider change is not an easy task.
* Cloud computing uses virtualization and the modern Web to dynamically provide resources of various kinds as services which are provisioned electronically. These services should be available in a reliable and scalable way so that multiple consumers can use them either explicitly upon request or simply as and when required. From the cloud provider’s perspective, this usually implies a multi-tenant architecture and a usage-based billing model. Thus, we can define the term cloud computing as follows: By using virtualized computing and storage resources and modern Web technologies, cloud computing provides scalable, network-centric, abstracted IT infrastructures, platforms, and applications as on-demand services. These services are billed on a usage basis
* One of the appealing aspects of cloud computing is that it hides the complexity of IT technology from users and developers. No need to know details of how a service is generated – it is the service provider’s job to provide a corresponding abstraction layer
* Resource virtualization is at the heart of most cloud architectures. The concept of virtualization allows an abstract, logical view on the physical resources and includes servers, data stores, networks, and software. The basic idea is to pool physical resources and manage them as a whole. Individual requests can then be served as required from these resource pools.
* Advantages of virtualization: resource usage, management, consolidation, energy consumption, less space required, emergency planning
* Benefits of virtualization for customer: dynamic behavior, availability, access
* Virtualization possible for OS, platform, storage, network, application
* Besides virtualization, service-oriented architectures and Web services are to be considered as the fundamental prerequisites for cloud computing. Service-oriented architectures (SOA) are architectures whose components are implemented as independent services. They can be flexibly tied together and orchestrated and they can communicate via messages in a loosely coupled configuration. With cloud computing, virtualized IT infrastructures, platforms, and entire applications are implemented
* Properties of SOA:
  + It consists of distributed components, i.e. the services.
  + Heterogeneous service consumers and service providers are interoperable across platforms; different programming languages and platforms can be used to implement individual services.
  + Services are loosely coupled and will be bound dynamically at runtime. An SOA consequently allows dynamic adjustments, which have a local (but no systemwide) effect.
* Organizational view corresponds to the deployment model and the technical view to the service models as specified in the NIST definition
* A public cloud comprises all cloud offerings where the providers and the potential users do not belong to the same organizational unit. The providers make their cloud accessible to the public and usually offer a self-service Web portal where the users can specify their desired scope of services. For this purpose, no overall framework agreement is necessary, but the contractual obligations are entered within the scope of the performance specifications. The services are billed based on the resources actually used in the corresponding period. In contrast to this model, the providers and users of a so-called private cloud belong to the same organizational unit. The main reason why a private cloud would be preferred over a public cloud is usually security: In the private cloud, control over the data remains with the users or their organization. The objective is both to make sure that the tools which are available in the public cloud can also be used in the private cloud, and to keep the door open to be able to scale the applications that were first developed for the private cloud for later use in the public cloud. Scenarios where services from the public cloud and from the private cloud are brought together, are referred to as hybrid cloud. With a hybrid cloud, certain functionalities or load peaks are usually transferred to the public cloud, while normal operation relies on the organization’s private resources.
* The cloud services provided on the PaaS layer are usually not targeted to end users, but rather to developers. These are programming environments (PE) and execution environments (EE) where proprietary software written in a specific programming language can be executed
* Cloud software applications that directly address the end user belong to the SaaS layer. This model frees the customers from the need to install the software locally and thus to provide the required resources themselves. Seen from the cloud architecture perspective, the SaaS offering can be developed and operated by the provider on the basis of a PaaS or IaaS offering. (Adobe Photoshop, Google Docs, Google Maps, Windows Live)
* Amazon EC2, S3, EBS, SQS, SimpleDB, Relational Database Service
* Google App Engine [85] is a PaaS that includes a programming environment, tool support, and an execution environment. This ‘instrumentation’ can be used to develop web applications for the scalable Google infrastructure. Google App Engine virtually frees web application designers from any tasks involving server administration so that they can focus on developing the required application functionality
* Google App Engine provides environments for the Python and Java programming languages. For these languages, a local execution environment is available where designers can test and trial their web applications during the development phase. Once the application is ready to go live, it can be moved to the Google infrastructure and run there.
* Google Storage, Cloud Print(allows any application to print to any output device on the Internet)
* The Windows Azure platform comprises a compute service for running applications, a storage service for storing data, and an SQL service for providing highly available relational databases in the cloud
  + With the Azure Service Platform, software products can be installed as cloud services on the Internet, or alternatively as applications in the in-house data center. The two methods can also be combined to implement a flexibly scalable hybrid cloud
* A service level agreement is an agreement between a service provider and a service consumer related to the service level (quality of service).
* The SLA implies a mutual agreement with respect to security, priorities, responsibilities, guarantees, and billing modalities. In addition, the SLA specifies metrics such as availability, throughput, response times, and others. By nature, SLAs always consider the output side, i.e. they are drafted from the service consumer’s perspective. A provider can stand out by delivering a service in a superior quality or in a particularly innovative way.
* Security is not only related to safely accessing resources, but also covers data privacy issues.
  + IaaS: Highest flexibility, the customer is responsible for security
  + PaaS: Medium flexibility, both the customer and the provider are responsible for security
  + SaaS: Lowest flexibility, the provider is responsible for security
* Open Source Cloud Stack - A number of solutions suitable for creating a cloud architecture are already available. Thus, it is possible to design an open source cloud stack
* Cloud infrastructures from commercial providers, such as Amazon EC2, and S3, and PaaS offerings, such as Google App Engine, boast a high degree of usability and can be used at low cost (or even for free). In some cases, however, it is desirable to build a private cloud infrastructure. Situations where a private cloud would be preferred over a public cloud might be characterized by special security requirements or the need to store critical company data

<http://web.b.ebscohost.com/ehost/ebookviewer/ebook/bmxlYmtfXzQwODc1OF9fQU41?sid=d513a3fe-acf8-4a73-958c-276353cc426b@sessionmgr4008&vid=0&format=EB&rid=1>

* Security concerns the confidentiality, availability and integrity of data or information. Security may also include authentication and non-repudiation.
* Privacy concerns the expression of or adherence to various legal and non-legal norms regarding the right to private life. In the European context this is often understood as compliance with European data protection regulations. Although it would be highly complex to map cloud issues onto the full panoply of privacy and personal data protection regulatory architectures, the globally accepted privacy principles give a useful frame: consent, purpose restriction, legitimacy, transparency, data security and data subject participation.
* Trust revolves around ‘assurance’ and confidence that people, data, entities, information or processes will function or behave in expected ways. Trust may be human to human, machine to machine (eg, handshake protocols negotiated within certain protocols), human to machine (eg, when a consumer reviews a digital signature advisory notice on a website) or machine to human (eg, when a system relies on user input and instructions without extensive verification). At a deeper level, trust might be regarded as a consequence of progress towards security or privacy objectives.

<https://ebookcentral.proquest.com/lib/ntnu/reader.action?docID=1115176> Rountree, Derrick, and Ileana Castrillo. The Basics of Cloud Computing : Understanding the Fundamentals of Cloud Computing in Theory and Practice, Elsevier Science, 2012. ProQuest Ebook Central, .

* With virtualization, you are able to host multiple virtual systems on one physical system. This has cut down implementation costs. You don’t need to have separate physical systems for each customer. In addition, virtualization allows for resource pooling and increased utilization of a physical system.
* One of the most pressing issues that have kept people from moving to the cloud is a lack of understanding of what the cloud is and what it offers. This lack of understanding causes fear. Usually the fear is around potential hidden costs, lack of control, integration issues, and security concerns. However, all the issues can be mitigated if you have a good understanding of what to look for in a cloud provider and what to expect from one.
* Even though some people consider cloud implementations to be more secure in certain aspects than traditional deployments, other aspects are often considered less secure and more of a risk. The risk mainly comes from the fact that you will not have direct control over the systems and the data. You have to trust what the service provider is doing
* What happens if the service provider goes out of business? How do you get access to your data? Does the company that takes over ownership of the systems then own your data? Is that company obligated to give it to you? What happens if there is a dispute and you don’t pay your bill? Can your data be held hostage? These are questions that you must ask when you’re considering a service provider. Different service providers will give different answers, so you must be aware of what you can expect from your provider.
* Privacy is a big concern when it comes to cloud implementation. The cloud provider will have direct access to your organization’s data. If this data is meant to be private, you have to worry about what measures are being taken to keep it private. In certain situations, you may be violating privacy standards simply by storing the data with an external provider.
* With multitenancy, you have very little control over or even knowledge of who may be sharing the same systems as you. You may unknowingly have competitors using those same systems. If your competitors were able to exploit some security flaw on the host system, they might be able to access your environment. The same thing goes for hackers. Hackers buy cloud space too. Their main goal may be to find and exploit areas that they can use to gain access to other environments on the same host.
* Public cloud
  + Advantages: deployments can offer increased availability over what is achievable internally. Saves the company hardware, software, training and staffing costs. Offers scalability without having to build out your own infrastructure, in contrast to private clouds. Offers accessibility on several devices. Cost savings both in amount and type of saving. No upfront costs for initial hardware and software deployments.
  + Drawbacks: Having your data housed externally can cause problems when you’re doing reporting or trying to move to on-premises systems. If you need to run reports or do business intelligence (BI) analytics against the data, you could end up having to transmit the data through the Internet. This can raise performance concerns as well as security issues. Reduced flexibility due to providers upgrade schedule and forced downtime due to offline maintenance. Public cloud providers raise a real issue over data security. There is a question of data ownership. Since the service provider owns the systems where your data resides, the provider could potentially be considered the true owner of the data. There is also an issue with data access. Theoretically, anyone who works at the service provider could potentially have access to your data.
* Private cloud
  + Advantages: Easier to troubleshoot as you have direct access to all systems. Monitor system as needed – flexibility to increase ability to service customer’s needs.
  + Drawbacks: Need to implement infrastructure is costly, need to estimate future needs as well as peak times. Need expertise in all applications and systems you deploy – expensive training and education.
* Hybrid clouds – connects multiple separate cloud environments. Offers the freedom to implement whatever is necessary to meet your organization’s needs, but hybrid clouds can be complex and expensive to implement
  + Advantages: allows you leave data internal until you can be assured that it is safe to move it to a public cloud environment
  + Drawbacks: Complex to implement – different sets of rules and procedures for each environment. May lead to duplication of data thus potentially altering only one data set and not the other.
* With SaaS implementations, the service provider usually controls virtually everything about the application. In many cases, this will limit any customization that can be done. Customization is costly.

<https://stackify.com/azure-vs-aws-comparison/>

* One of the biggest advantages of cloud computing is the simplicity of deploying an application. As a developer, I want to deploy my app to multiple servers without having to deal with the actual servers. Being able to take advantage of PaaS features like SQL databases, caching, queueing, NoSQL and other technologies are also a big deal.