Cloud Computing

UNIT-I

Introduction to Cloud Computing:

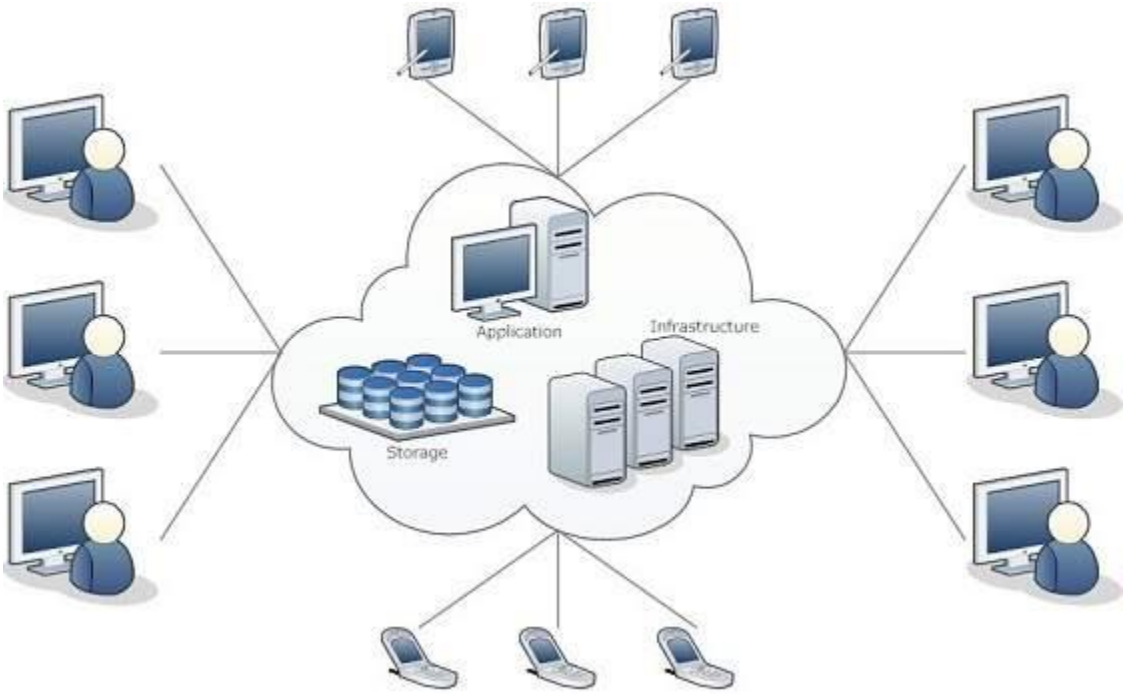
1. Cloud Computing in a Nutshell

The term **Cloud** refers to a **Network** or **Internet.** In other words, we can say that Cloud is something, which is present at remote location. Cloud can provide services over public and private networks, i.e., WAN, LAN or VPN.

Applications such as e-mail, web conferencing, customer relationship management (CRM) execute on cloud.

What is Cloud Computing?

Cloud Computing refers to **manipulating, configuring,** and **accessing** the hardware and software resources remotely. It offers online data storage, infrastructure, and application.



Cloud computing offers **platform independency,** as the software is not required to be installed locally on the PC. Hence, the Cloud Computing is making our business applications **mobile** and **collaborative.**

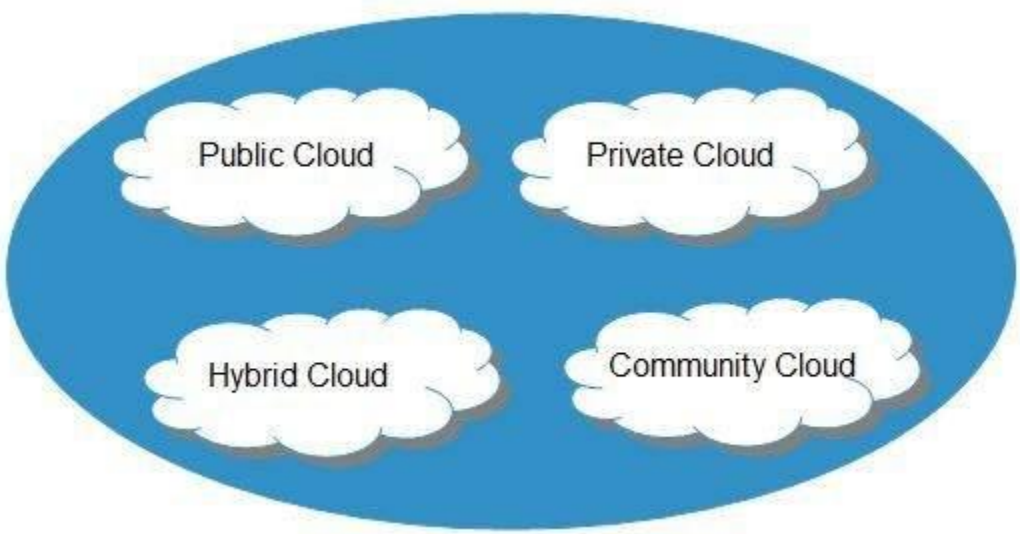
Basic Concepts

There are certain services and models working behind the scene making the cloud computing feasible and accessible to end users. Following are the working models for cloud computing:

- Deployment Models
- Service Models

Deployment Models

Deployment models define the type of access to the cloud, i.e., how the cloud is located? Cloud can have any of the four types of access: Public, Private, Hybrid, and Community.



Public Cloud- The **public cloud** allows systems and services to be easily accessible to the general public. Public cloud may be less secure because of its openness.

Private Cloud- The **private cloud** allows systems and services to be accessible within an organization. It is more secured because of its private nature.

Community Cloud- The **community cloud** allows systems and services to be accessible by a group of organizations.

Hybrid Cloud- The **hybrid cloud** is a mixture of public and private cloud, in which the critical activities are performed using private cloud while the non-critical activities are performed using public cloud.

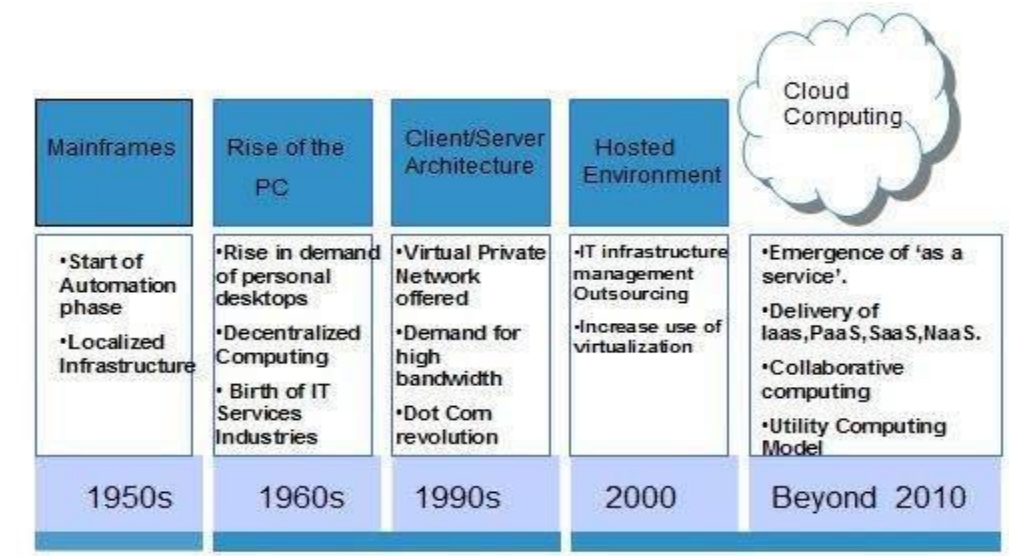
Service Models

Cloud computing is based on service models. These are categorized into three basic service models which are -

- Infrastructure-as–a-Service (IaaS)
- Platform-as-a-Service (PaaS)
- Software-as-a-Service (SaaS)

History of Cloud Computing

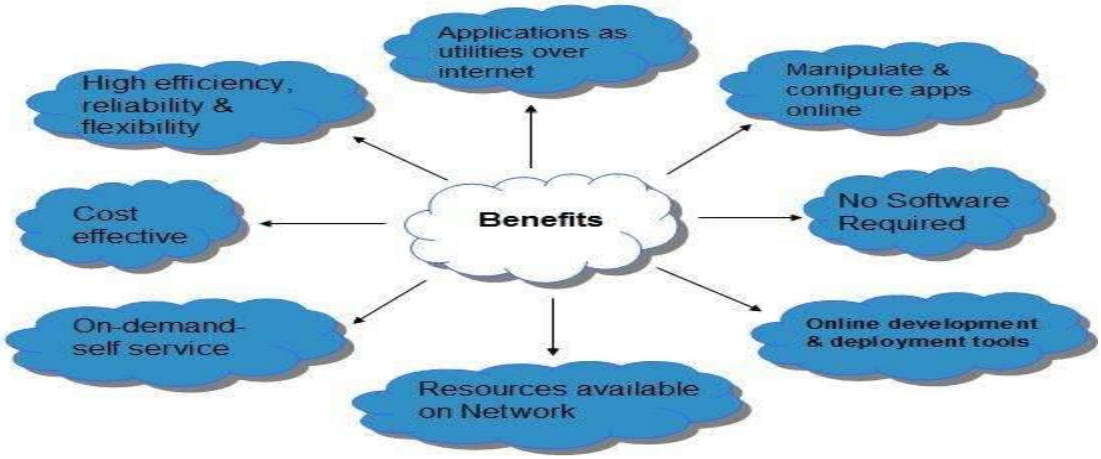
The concept of **Cloud Computing** came into existence in the year 1950 with implementation of mainframe computers, accessible via **thin/static clients.** Since then, cloud computing has been evolved from static clients to dynamic ones and from software to services. The following diagram explains the evolution of cloud computing:



Benefits

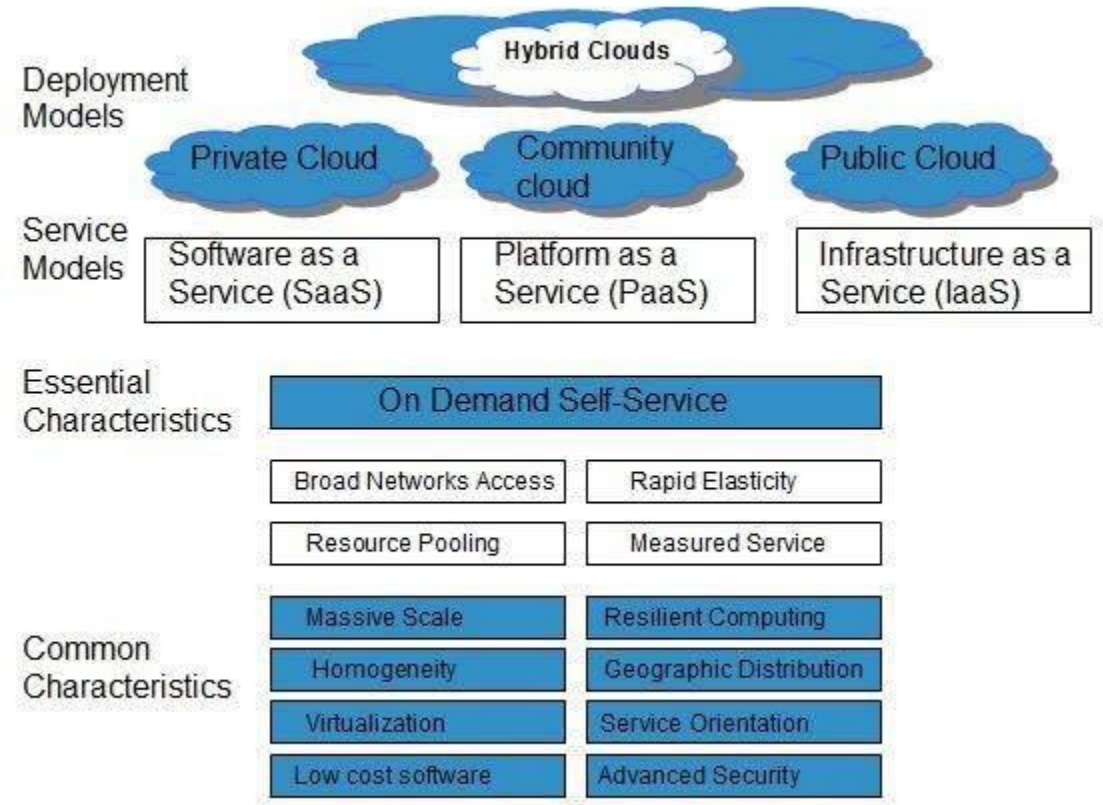
Cloud Computing has numerous advantages. Some of them are listed below -

- One can access applications as utilities, over the Internet.
- One can manipulate and configure the applications online at any time.
- It does not require to install a software to access or manipulate cloud application.
- Cloud Computing offers online development and deployment tools, programming runtime environment through **PaaS model.**
- Cloud resources are available over the network in a manner that provide platform independent access to any type of clients.
- Cloud Computing offers **on-demand self-service.** The resources can be used without interaction with cloud service provider.
- Cloud Computing is highly cost effective because it operates at high efficiency with optimum utilization. It just requires an Internet connection
- Cloud Computing offers load balancing that makes it more reliable.



Characteristics of Cloud Computing

There are four key characteristics of cloud computing. They are shown in the following diagram:



On Demand Self Service- Cloud Computing allows the users to use web services and resources on demand. One can logon to a website at any time and use them.

Broad Network Access- Since cloud computing is completely web based, it can be accessed from anywhere and at any time.

Resource Pooling- Cloud computing allows multiple tenants to share a pool of resources. One can share single physical instance of hardware, database and basic infrastructure.

Rapid Elasticity- It is very easy to scale the resources vertically or horizontally at any time. Scaling of resources means the ability of resources to deal with increasing or decreasing demand. The resources being used by customers at any given point of time are automatically monitored.

Measured Service- In this service cloud provider controls and monitors all the aspects of cloud service. Resource optimization, billing, and capacity planning etc. depend on it.

2. Roots of Cloud Computing

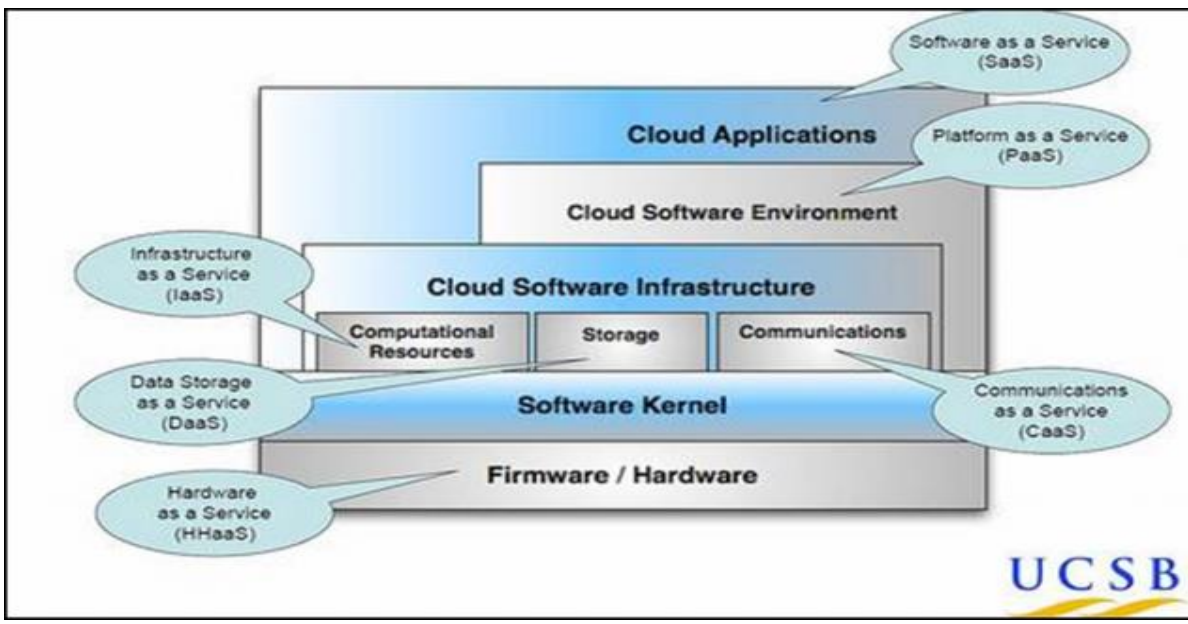
The roots of clouds computing can be stated by observing the advancement of several technologies, especially in hardware (virtualization, multi-core chips), Internet technologies (Web services, service-oriented architectures, Web 2.0), distributed computing (clusters, grids), and systems management (autonomic computing, data center automation).

Some of these technologies have been tagged as hype in their early stages of development; however, they later received significant attention from academia and were sanctioned by major industry players. Consequently, a specification and standardization process followed, leading to maturity and wide adoption. The emergence of cloud computing itself is closely linked to the maturity of such technologies. We present a closer look at the technologies that form the base of cloud computing, with the aim of providing a clearer picture of the cloud ecosystem.

Computing delivered as a utility can be defined as “on demand delivery of infrastructure, applications, and business processes in a security-rich, shared, scalable, and based computer environment over the Internet for a fee”.

3. Layers and Types of Cloud

As more functionality moves to the internet cloud every provider and user is developing their own definition. Industry experts and researchers are struggling to formulate a standard set of terms to describe all the different functions. This graphic developed by Lamia Youseff, University of California, Santa Barbara and Maria Butrico, Dilma Da Silva, IBM T.J. Watson Research Center depicts as five layers, with three constituents to the cloud infrastructure layer. The figure represents the inter-dependency between the different layers in the cloud.



They define the five layers as follows:

A. Cloud Application Layer –The most visible layer to the end-users of the cloud. Normally, the users access the services provided by this layer through web-portals, and are sometimes required to pay fees to use them.

B. Cloud Software Environment Layer – The second layer in our proposed cloud ontology is the cloud software environment layer (also dubbed the software platform layer). The users of this layer are cloud applications' developers, implementing their applications for and deploying them on the cloud.

C. Cloud Software Infrastructure Layer – The cloud software infrastructure layer provides fundamental resources to other higher-level layers. Cloud services offered in this layer can be categorized into: computational resources, data storage, and communications.

D. Software Kernel – This cloud layer provides the basic software management for the physical servers that compose the cloud. Software kernels at this level can be implemented as an OS kernel, hypervisor, and virtual machine monitor and/or clustering middleware.

E. Hardware and Firmware – The bottom layer of the cloud stack in our proposed ontology is the actual physical hardware and switches that form the backbone of the cloud. In this regard, users of this layer of the cloud are normally big enterprises with huge IT requirements in need of subleasing Hardware as a Service (HaaS).

Types of Cloud

Cloud computing is usually described in one of two ways. Either based on the deployment model, or on the service that the cloud is offering.

Based on a deployment model, we can classify cloud as:

- public,
- private,
- hybrid
- community cloud

Based on a service the cloud model is offering, we are speaking of either:

- IaaS (Infrastructure-as-a-Service)
- PaaS (Platform-as-a-Service)
- SaaS (Software-as-a-Service)
- or, Storage, Database, Information, Process, Application, Integration, Security, Management, Testing-as-a-service

4. Features of a cloud

There are numerous features of cloud services that are beneficial to businesses of all sizes.

On demand self-service: Resources can automatically be provisioned without the need of human interaction as and when needed.

Compatibility: Cloud services allows access to the data from any location and on any device. Employees can work from anywhere.

Better team work: Access from any location facilitates better collaboration with persons working on field being able to update their reports

Elasticity: Cloud services are scalable. Consumers can add resources they need and discard resources they do not want.

Reliability: Cloud runs on multiple servers and are automated to run even if one server fails. Resources are drawn from the other servers to ensure continuity without any interruption.

Nil investment: consumers need not make any upfront investment on hardware. All services can be drawn on subscription basis and users pay for the resources they consume.

Disaster recovery: With replication and storing across multiple servers, cloud allows easy cost-effective solutions at times of data loss due to some theft or calamity.

Updates: Consumers need not worry about software updates and technical issues. The cloud servers are in different places away from your business premises. The providers do all the updates and patches.

Security: Cloud services offers enhanced security. You can access your data from any system even if you lose your personal device.

5. Software-as-a-Service

Software as a service (SaaS) is a software distribution model in which a third-party provider hosts applications and makes them available to customers over the Internet. SaaS is one of three main categories of cloud computing, alongside infrastructure as a service (IaaS) and platform as a service (PaaS).

SaaS removes the need for organizations to install and run applications on their own computers or in their own data centers. This eliminates the expense of hardware acquisition, provisioning and maintenance, as well as software licensing, installation and support. Other benefits of the SaaS model include:

Flexible payments: Rather than purchasing software to install, or additional hardware to support it, customers subscribe to a SaaS offering. Generally, they pay for this service on a monthly basis using a pay-as-you-go model. Transitioning costs to a recurring operating expense allows many businesses to exercise better and more predictable budgeting. Users can also terminate SaaS offerings at any time to stop those recurring costs.

Scalable usage: Cloud services like SaaS offer high scalability, which gives customers the option to access more, or fewer, services or features on-demand.

Automatic updates: Rather than purchasing new software, customers can rely on a SaaS provider to automatically perform updates and patch management. This further reduces the burden on in-house IT staff.

Accessibility and persistence: Since SaaS applications are delivered over the Internet, users can access them from any Internet-enabled device and location.

But SaaS also poses some potential disadvantages. Businesses must rely on outside vendors to provide the software, keep that software up and running, track and report accurate billing and facilitate a secure environment for the business' data. Providers that experience service disruptions, impose unwanted changes to service offerings, experience a security breach or any other issue can have a profound effect on the customers' ability to use those SaaS offerings. As a result, users should understand their SaaS provider's service-level agreement, and make sure it is enforced.

SaaS is closely related to the ASP (application service provider) and on demand computing software delivery models. The *hosted application management* model of SaaS is similar to ASP: the provider hosts the customer's software and delivers it to approved end users over the internet. In the *software on demand* SaaS model, the provider gives customers network-based access to a single copy of an application that the provider created specifically for SaaS distribution. The application's source code is the same for all customers and when new features are functionalities are rolled out, they are rolled out to all customers. Depending upon the service level agreement (SLA), the customer's data for each model may be stored locally, in the cloud or both locally and in the cloud.

There are SaaS applications for fundamental business technologies, such as email, sales management, customer relationship management (CRM), financial management, human resource management, billing and collaboration. Leading SaaS providers include Salesforce, Oracle, SAP, Intuit and Microsoft.

6. Platform-as-a-Service

Platform as a service (PaaS) is a cloud computing model in which a third-party provider delivers hardware and software tools -- usually those needed for application development -- to users over the internet. A PaaS provider hosts the hardware and software on its own infrastructure. As a result, PaaS frees users from having to install in-house hardware and software to develop or run a new application.

PaaS does not typically replace a business's entire IT infrastructure. Instead, a business relies on PaaS providers for key services, such as application hosting or Java development.

A PaaS provider builds and supplies a resilient and optimized environment on which users can install applications and data sets. Users can focus on creating and running applications rather than constructing and maintaining the underlying infrastructure and services.

Many PaaS products are geared toward software development. These platforms offer compute and storage infrastructure, as well as text editing, version management, compiling and testing services that help developers

create new software more quickly and efficiently. A PaaS product can also enable development teams to collaborate and work together, regardless of their physical location.

PaaS pros and cons

The principal benefit of PaaS is simplicity and convenience for users -- the PaaS provider supplies much of the infrastructure and other IT services, which users can access anywhere via a web browser. PaaS providers then charge for that access on a per-use basis -- a model that many enterprises prefer, as it eliminates the capital expenses they traditionally have for on-premises hardware and software. Some PaaS providers charge a flat monthly fee to access their service, as well as the apps hosted within it.

Service availability or resilience, however, can be a concern with PaaS. If a provider experiences a service outage or other infrastructure disruption, this can adversely affect customers and result in costly lapses of productivity. Provider lock-in is another common concern, since users cannot easily migrate many of the services and much of the data produced through one PaaS product to another competing product. Users must evaluate the business risks of service downtime and lock-in before they commit to a PaaS provider.

Internal changes to a PaaS product are also a potential issue. For example, if a PaaS provider stops supporting a certain programming language or opts to use a different set of development tools, the impact on users can be difficult and disruptive. Users must follow the PaaS provider's service roadmap to understand how the provider's plans will affect its environment and capabilities.

7. **Infrastructure-as-a-Service**

Infrastructure as a service (IaaS) is a form of cloud computing that provides virtualized computing resources over the internet. IaaS is one of the three main categories of cloud computing services, alongside software as a service (SaaS) and platform as a service (PaaS).

In an IaaS model, a cloud provider hosts the infrastructure components traditionally present in an on-premises data center, including servers, storage and networking hardware, as well as the virtualization or hypervisor layer.

The IaaS provider also supplies a range of services to accompany those infrastructure components. These can include detailed billing, monitoring, log access, security, load balancing and clustering, as well as storage resiliency, such as backup, replication and recovery. These services are increasingly policy-driven, enabling IaaS users to implement greater levels of automation and orchestration for important infrastructure tasks. For example, a user can implement policies to drive load balancing to maintain application availability and performance.

IaaS customers access resources and services through a wide area network (WAN), such as the internet, and can use the cloud provider's services to install the remaining elements of an application stack. For example, the user can log in to the IaaS platform to create virtual machines (VMs); install operating systems in each VM; deploy middleware, such as databases; create storage buckets for workloads and backups; and install the enterprise workload into that VM. Customers can then use the provider's services to track costs, monitor performance, balance network traffic, troubleshoot application issues, manage disaster recovery and more.

Any cloud computing model requires the participation of a provider. The provider is often a third-party organization that specializes in selling IaaS. Amazon Web Services (AWS) and Google Cloud Platform (GCP) are examples of independent IaaS providers. A business might also opt to deploy a private cloud, becoming its own provider of infrastructure services.

IaaS pros and cons

Organizations choose IaaS because it is often easier, faster and more cost-efficient to operate a workload without having to buy, manage and support the underlying infrastructure. With IaaS, a business can simply rent or lease that infrastructure from another business.

IaaS is an effective model for workloads that are temporary, experimental or that change unexpectedly. For example, if a business is developing a new software product, it might be more cost-effective to host and test the application using an IaaS provider. Once the new software is tested and refined, the business can remove it from the IaaS environment for a more traditional, in-house deployment. Conversely, the business could commit that piece of software to a long-term IaaS deployment, where the costs of a long-term commitment may be less.

In general, IaaS customers pay on a per use basis, typically by the hour, week or month. Some IaaS providers also charge customers based on the amount of virtual machine space they use. This pay-as-you-go model eliminates the capital expense of deploying in-house hardware and software.

8. Challenges and Risks.

Cloud Migration

Cloud migration is the process of moving data, applications, and other important information of an organization from its on-premises either desktops or servers to the cloud infrastructure, and this can also involve in moving data between different cloud setups.

Cloud migration enables all the computing capabilities those were performed earlier by devices installed on-premises. Cloud migration is a big challenge as many companies when they require to migrate from on-premises to cloud or from one cloud to another, they partner with experienced cloud service provider.

Incompatibility:

During moving workloads from on-premises to the cloud, the common issue the incompatibility between on-premises infrastructure and the services which are companies going to buy from the public cloud providers. In last current years, most CSPs tried to create “connectors of sort” to make practices more standardize and homogenous.

Data security:

CSPs are responsible to provide clouds' security, but they're not responsible for your apps, servers, and security of data. As per CDW 2013 State of the Cloud Report, "46 percent of respondents face security of data or applications as a significant challenge."

When your CSP ensure you about the complete compliance and regulation, don't consider it as 100% compliant and yielding. You still require to encrypt and secure your own data and should invest in buying suite of tools from your CSP to protect your data from cyber-attacks.

Lack of expertise:

With the quick advancements and improvements in cloud technologies, more and more organizations are clouds to place their workloads. However, they face difficulties to keep up with the tools which require particular expertise. Organizations can deal with this challenge by providing cloud technologies training to their sys admins along with development staff.

By adding cloud specialists to IT teams may be costly too for small and medium businesses (SMBs). Luckily, various routine activities that specialists perform can be automated using automated tools. Now, many organizations are also moving to DevOps tools, like Puppet and Chef due to their multi-tasking and automation capabilities such as monitoring resource usage, automating backups etc. These automating tools considerably contribute to cloud optimization for cost, security, and governance.

Bandwidth Cost:

Though organizations and businesses can save money on hardware using cloud, but they have to pay extra for the bandwidth they use to access their workloads. However, it doesn't charge much for smaller apps, but data-intensive apps need more bandwidth which can costs higher.

Cloud computing platforms:

9. Infrastructure as service: Amazon EC2

"Amazon Elastic Compute Cloud (Amazon EC2) is an Amazon Web Service (AWS) you can use to access servers, software, and storage resources across the Internet in a self-service manner "

- Provides scalable, pay as-you-go compute capacity

- Elastic - scales in both direction

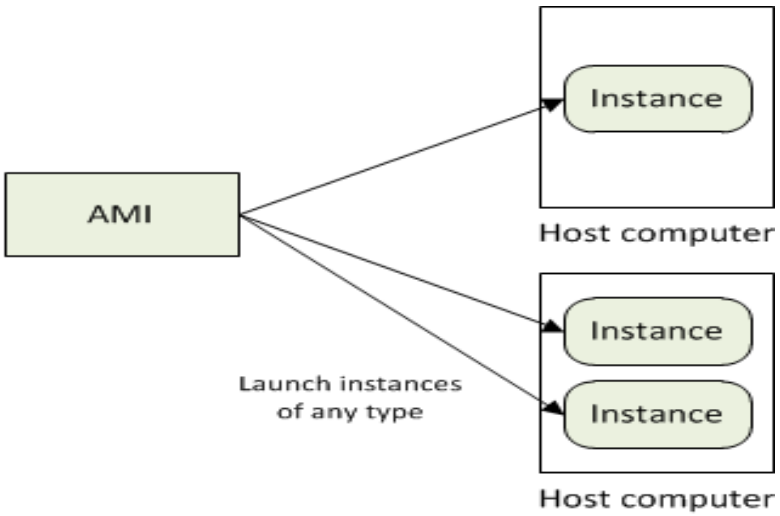
EC2 Concepts

- AMI & Instance

  Amazon Machine Image (AMI) is a template for software configuration (Operating System, Application Server, and Applications)

  Instance is a AMI running on virtual servers in the cloud

  Each *instance type* offers different compute and memory facilities



- Region & Zones

  Amazon have data centers in different region across the globe

  An instance can be launched in different regions depending on the need.

  - Closer to specific customer
  - To meet legal or other requirements

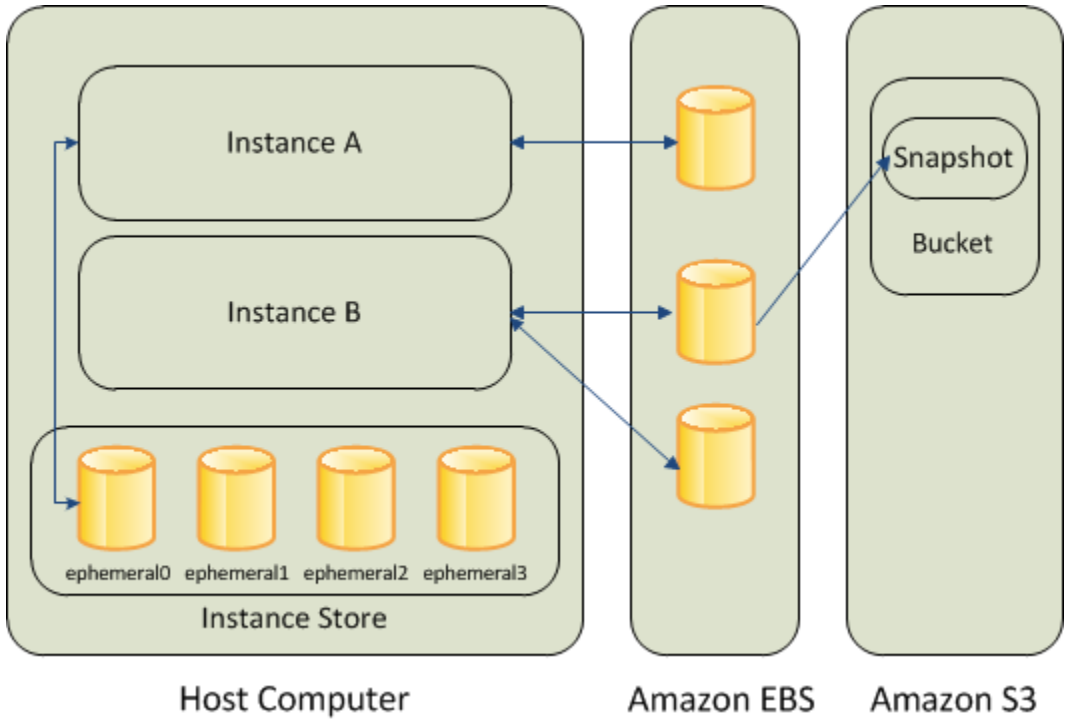
  Each region has set of zones

  - Zones are isolated from failure in other zones
  - Inexpensive, low latency connectivity between zones in same region

- Storage

Amazon EC2 provides three type of storage option

- Amazon EBS
- Amazon S3
- Instance Storage



- Networking and Security

  Instances can be launched on one of the two platforms

  - EC2-Classic
  - EC2-VPC

  Instance IP address is dynamic.

  - new IP address is assigned every time instance is launched

  Static IP address – Elastic IP address

  - Remap the Elastic IP to new instance to mask failure
  - Separate pool for EC2-Classic and VPC

  Security Groups to access control to instance

- Monitoring, Auto Scaling and Load Balancer

  Monitor statistics of instances and EBS

  - CloudWatch

  Automatically scales amazon EC2 capacity up and down based on rules

  - Add and remove compute resource based on demand
  - Suitable for businesses experiencing variability in usage

  Distribute incoming traffic across multiple instances

  - Elastic Load Balancing

10. Platform as Service: Google App Engine, Microsoft Azure

GOOGLE APP ENGINE

Google App Engine is a cloud computing technology offered by Google Cloud Platform for hosting web applications in Google-managed data centers. Google App Engine is a Platform as a Service (PaaS) offering for Python, Java, Go and PHP.

App Engine application can run in two environments, the **standard** environment & the **flexible** environment. Both these environments can be used in your application simultaneously if you structure your application using the micro-services architecture.

WHY GOOGLE APP ENGINE

Build Apps

Build robust and highly scalable web applications & mobile backends with built-in services & APIs like NoSQL datastores, memcache, and a user authentication API.

Scales Automatically

Google App Engine will scale your application automatically based on need, from zero to millions of users in response to the amount of traffic it receives.

Start Quickly, Build Faster

Deploying web and mobile applications becomes much faster for you with built-in services like load balancing, health checks, and application logging.

GOOGLE APP ENGINE FEATURES

Google Cloud SQL

A fully-managed web service that enables you to create, configure, and use relational databases that resides in Google's cloud.

NoSQL Datastore

A schemaless object datastore, including a rich data modeling API, scalable storage & SQL-like query language.

Popular Languages

Build your application in Python, Java, PHP or Go.

User Authentication

Allows applications to sign in users with the Google Accounts and addresses these users with unique identifiers.

Search

Perform Google-like searches over structured data such as atom, HTML, plain text, numbers, dates, and geographic locations.

Memcache

A distributed, in-memory data cache that can be extensively used to enhance & improve application performance.

Security Scanner

Security Scanner by default scans the common web application security for vulnerabilities, like mixed content & XSS.

Traffic Splitting

Route incoming requests to different app versions, run A/B tests & do incremental feature roll-outs.

Azure Platform As A Service

Platform as a service (PaaS) is a deployment and development environment within the cloud that delivers simple cloud-based apps to complex, cloud-enabled applications. PaaS is designed to support the complete web application lifecycle of building, testing, deploying, managing, and updating.

PaaS includes a complete infrastructure of servers, storages, networking, and middleware development tools like business intelligence services (BI), database management systems, etc. A complete platform is offered in PaaS in which the client can host their applications without the need to worry about the maintenance of the servers and its operating systems. However, the user of the PaaS service should look after the implementation of the developed application to decide whether to scale it up or down depending on the traffic that the application receives.

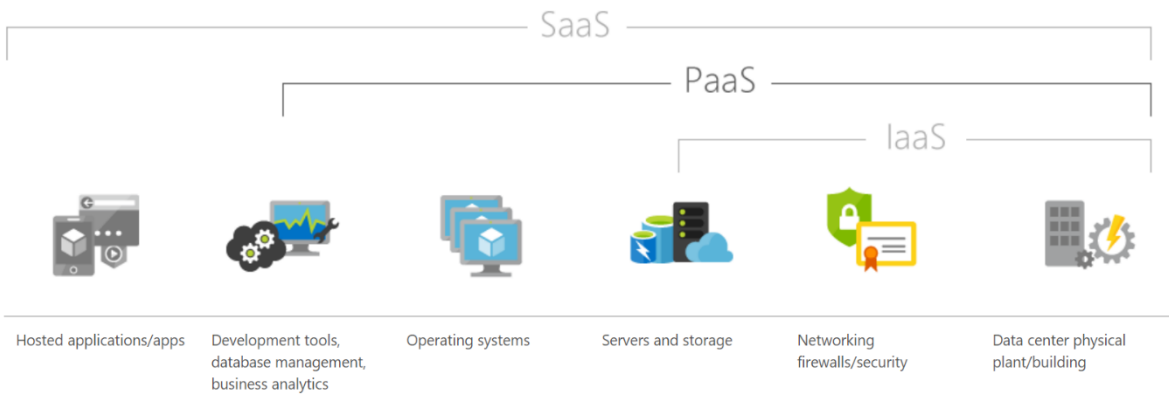


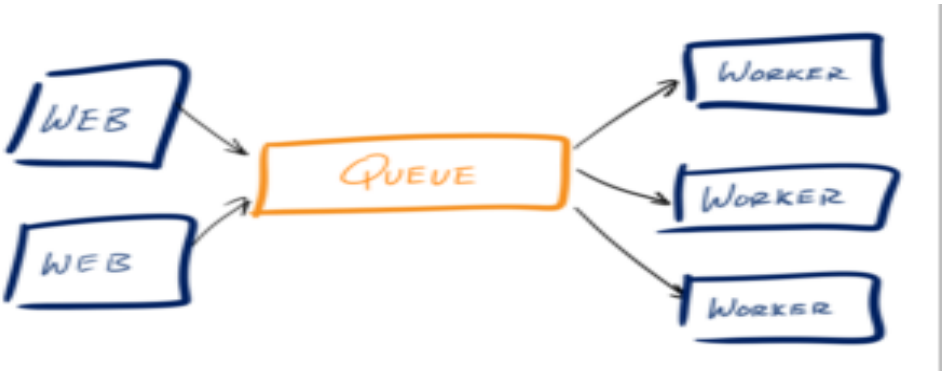
Figure 2 Source: Microsoft

The PaaS backbone utilizes virtualization techniques, where the virtual machine is independent of the actual hardware that hosts it.

Azure Cloud Services has two main components; the application files such as the source code, DLL, etc. and the configuration file. Together these two will spin up a combination of Worker Roles and Web Roles. On the cloud services, Azure handles all the hard work of the

operating systems on your behalf, so that the full focus is to build a quality application for the end users.

The Web Role is an Azure VM that is preconfigured as a web server running IIS (Internet Information Service) which automatically loads the developed application when the Virtual machine boots up. This results in the creation of the public endpoint for the application which is usually in the form of a website but could be an API or similar.



The Worker Role runs alongside with the Web Role and performs the computing functions needed for the smooth operation of your application. The Web Role will accept the user's input and will queue up for an action to process later by the Work Role. Subsequently, this enables the Web Role to be more productive and responsive.

Azure PaaS Services

Azure offers five main services of Platform as a Service in which multiple service types host a custom application or a business logic for specific use cases:

1. Web Apps

These are an abstraction of a Web Server such as IIS and Tomcat that run applications written in mostly in Java, Python,.NET, PHP, Node.js, etc. These are simple to set up and provide a variety of benefits, available 99.9% of the time which is a key benefit.

2. Mobile Apps

The back ends of mobile apps can be hosted on the Azure PaaS easily using the SDKs available for all major mobile operating systems of iOS, Android, Windows, etc. It enables the unique ability of offline sync so the user can use the app even if they are offline and sync the data back when they are back online. Another major benefit is the ability to push notifications allowing sending of custom notifications for all targeted application users.

3. Logic Apps

No apps are hosted, but there is an orchestrated business logic app to automate a business process. These are initiated by a trigger when a predefined business condition is met.

4. Functions

Functional apps can perform multiple tasks within the same application. These functional apps host smaller applications such as microservices and background jobs that only run for short periods.

5. Web Jobs

These are a part of a service that runs within an app service on web apps or mobile apps. They are similar to Functions but do not require any coding to set it up.

11. Utility Computing

Utility computing is a service provisioning model in which a service provider makes computing resources and infrastructure management available to the customer as needed, and charges them for specific usage rather than a flat rate. Like other types of on-demand computing (such as grid computing), the utility model seeks to maximize the efficient use of resources and/or minimize associated costs.

The word utility is used to make an analogy to other services, such as electrical power, that seek to meet fluctuating customer needs, and charge for the resources based on usage rather than on a flat-rate basis. This approach, sometimes known as pay-per-use or metered services is becoming increasingly common in enterprise computing and is

sometimes used for the consumer market as well, for Internet service, Web site access, file sharing, and other applications.

Another version of utility computing is carried out within an enterprise. In a shared pool utility model, an enterprise centralizes its computing resources to serve a larger number of users without unnecessary redundancy.

12. Elastic Computing.

Nowadays cloud computing are well known phenomenon for everyone. Most of the small and large business have switched their data to cloud storage. Moreover, organization also prefer to have elastic computing.
But before proceeding to know more about elastic computing let's have a quick outline of cloud computing. **Cloud Computing or Cloud** is defined as using various services such as software development platforms, servers, storage, over the Internet.
So, **what does Elastic Cloud Computing mean?**
Elastic computing is nothing but a concept in cloud computing in which computing resources can be scaled up and down easily by the cloud service provider. Cloud service provider gives you provision to flexible computing power when and wherever required. The elasticity of these resources depends upon the following factors such as processing power, storage, bandwidth, etc.
Types of Elastic Cloud Computing
Rather than various types, elastic computing have only one type i.e. **Elasticity, or fully-automated scalability** which removes manual labor for increasing or decreasing resources as everything is controlled by triggers by the system monitoring tools.
Elasticity refers the ability to fit the resources needed to cope with loads, so that when load increase you scale up by adding more resources and when demand diminishes you shrink back and remove unneeded resources. Elasticity is mostly important in Cloud environment where you pay-per-used resources only.
Benefits/Pros of Elastic Cloud Computing
Elastic Cloud Computing has numerous advantages. Some of them are as follow:-
1. Cost Efficiency: - Cloud is available at much cheaper rates than traditional approaches and can significantly lower the overall IT expenses. By using cloud solution companies can save licensing fees as well as eliminate overhead charges such as the cost of data storage, software updates, management etc.
2. Convenience and continuous availability: - Cloud makes easier access of shared documents and files with view and modify choice. Public clouds also offer services that are available wherever the end user might be located. Moreover it guaranteed continuous availability of resources and In case of system failure; alternative instances are automatically spawned on other machines.
3. Backup and Recovery: - The process of backing up and recovering data is easy as information is residing on cloud simplified and not on a physical device. The various cloud providers offer reliable and flexible backup/recovery solutions.
4. Cloud is environmentally friendly:-The cloud is more efficient than the typical IT infrastructure and it takes fewer resources to compute, thus saving energy.

5. Scalability and Performance: - Scalability is a built-in feature for cloud deployments. Cloud instances are deployed automatically only when needed and as a result enhance performance with excellent speed of computations.
6. Increased Storage Capacity:-The cloud can accommodate and store much more data compared to a personal computer and in a way offers almost unlimited storage capacity.

Disadvantages/Cons of Elastic Cloud Computing:-
1. **Security and Privacy in the Cloud**: - Security is the biggest concern in cloud computing. Companies essentially hide their private data and information over cloud as remote based cloud infrastructure is used, it is then up to the cloud service provider to manage, protect and retain data confidential.

2. **Limited Control**: - Since the applications and services are running remotely companies, users and third party virtual environments have limited control over the function and execution of the hardware and software.

3. **Dependency and vendor lock-in: -** One of the major drawbacks of cloud computing is the implicit dependency on the provider. It is also called “vendor lock-in”. As it becomes difficult to migrate vast data from old provider to new. So, it is advisable to select vendor very carefully.

4. **Increased Vulnerability**: - Cloud based solutions are exposed on the public internet therefore are more vulnerable target for malicious users and hackers. As we know nothing is completely secure over Internet even the biggest organizations also suffer from serious attacks and security breaches.

UNIT-II

1. The promise of the cloud

“Cloud computing” promises myriad benefits — including cost savings on technology infrastructure and faster software upgrades — for users ranging from small startups to large corporations. That’s an auspicious future considering that not everyone agrees on exactly what cloud computing is or what it can do.

Despite the ethereal name, in its broadest terms, the concept of cloud computing is simple. Rather than running software on its own computers — “on premises” as the terminology goes — a company buys access to software on computers operated by a third party. Typically, the software is accessed over the Internet using only a web browser. If the software performs properly, it doesn’t matter where the systems that run it are located. They are “out there somewhere” — in “the cloud” of the Internet. Since companies tend to purchase access to this remote software on a subscription basis, cloud computing is also often termed “software as a service.”

These days, no computer user is an island. A recent study determined that 80% of the data used by business comes from outside the company.

In between these explanations of cloud computing lies a variety of products and services, all of which claim to offer several advantages — lowered investment in hardware, more efficient use of computing systems in existing data centers, easier scale-up of the applications and services. These approaches are now possible due to faster and more pervasive communications. As bandwidth has become cheap and readily available, and transmission speed is no longer an impediment, it's possible to store data and run software anywhere for users to access from wherever they want.

2. The Cloud service offerings and Deployment model

Cloud computing refers to the use of network of remote servers that are hosted over the Internet, and there are many cloud deployment and service models.

One of the most unique characteristics of cloud computing is that the services from data storage to creation of software applications can be availed on pay-per-use basis.

Cloud Deployment Models

The cloud deployment models mentioned below are based on the National Institute of Standards and Technologies. There are four basic cloud deployment models, which are:

1) **Private cloud model**

In this system, the cloud infrastructure is set up on the premise for the exclusive use of an organization and its customers. In terms of cost efficiency, this deployment model doesn't bring many benefits. However, many large enterprises choose it because of the security it offers.

2) **Public cloud model**

Public cloud is hosted on the premise of the service provider. The service provider than provides cloud services to all of its customers. This deployment is generally adopted by many small to mid-sized organizations for their non-core and some of their core functions.

3) **Community cloud**

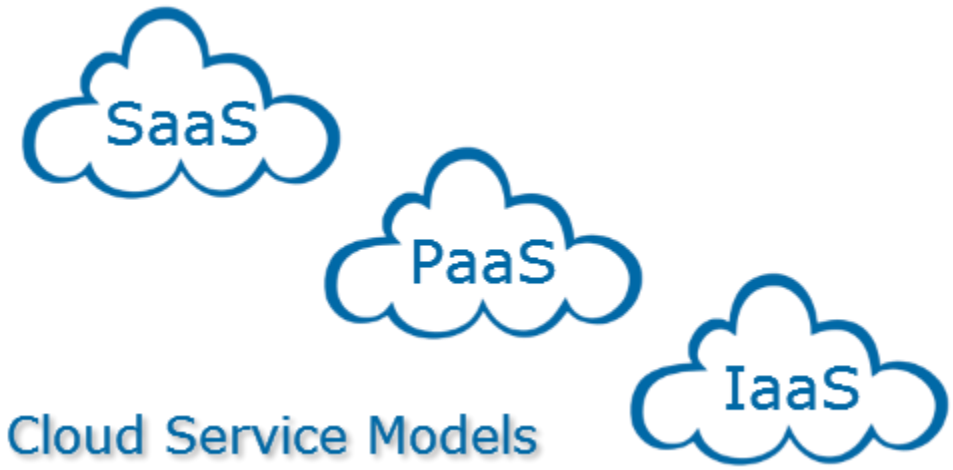
Community cloud model is a cloud infrastructure shared by a group of organizations of similar industries and backgrounds with similar requirements i.e. mission, security, compliance and IT policies. It may exist on or off premise and can be managed by a community of these organizations.

4) **Hybrid cloud model**

Hybrid cloud is a combination of two or more models, private cloud, public cloud or community cloud. Though these models maintain their separate entities they are amalgamated through a standard technology that enables the portability of data and applications.

Cloud Service Models

Cloud service models can be broadly defined in three categories – SaaS (Software as a Service), PaaS (Platform as a Service) and IaaS (Infrastructure as a Service).



1) **SaaS**

SaaS is a software service provided over the internet and requires no prior installation. These services can be availed from any part of the world at a minimal per month fee.

2) **PaaS**

Platform as a Service runs on physical servers, database software and web servers. All these are basically known as platforms, and cloud computing firms provide platforms as a service allowing you to use the service without having to build it from the scratch.

3) **IaaS**

Infrastructure as a Service includes infrastructure such as servers, private networks, disk drives, long term storage solutions, email servers, domain name servers etc. IaaS on demand allows organizations to make use of operating systems and associated software without having to pay for hefty license fees.

Some of the services available for cloud customers in the Cloud Service Models mentioned above are:

SaaS	PaaS	IaaS
CRM	Business Intelligence	Platform Hosting
Sales	Application Development	Content Delivery Network
Content Management	Development and Testing	Storage
Human Resources	Database	Compute
ERP	Integration	Backup and Recovery

3. Challenges in the cloud

Cloud computing challenges are numerous and thorny, to be sure. These days, everyone is in the cloud — but that doesn't mean that they've figured out how to overcome all the challenges of cloud computing.

In the RightScale 2018 State of the Cloud Report, 96 percent of IT professionals surveyed said their companies were using cloud computing services, and 92 percent were using the public cloud. On average, organizations are running about 40 percent of their workloads in the cloud, and that percentage is growing.

As companies move more applications to the cloud, the cloud market is booming. According to Gartner, the public cloud market will likely be worth $186.4 billion in 2018, up 21.4 percent over last year. The infrastructure as a service (IaaS) market is growing particularly fast. This segment alone could grow 35.9 percent this year to total $40.8 billion.

However, numerous surveys are finding that organizations still have concerns about cloud computing. While IT leaders are embracing the cloud because of the benefits it offers, they continue to face very significant cloud computing challenges, including the following:

Cloud Computing Challenge 1: Security

Since the advent of the public cloud, enterprises have worried about potential security risks, and that hasn't changed. In the RightScale survey, it was the number one challenge cited by respondents, with 77 percent saying that cloud security is a challenge, including 29 percent who called it a significant challenge.

Cybersecurity experts are even more concerned about cloud security than other IT staffers are. A 2018 Crowd Research Partners survey found that 90 percent of security professionals are concerned about cloud security. More specifically, they have fears about data loss and leakage (67 percent), data privacy (61 percent) and breaches of confidentiality (53 percent).

Interestingly, though, security concerns appear to be waning as time passes, particularly among companies that have been using the cloud longer. The RightScale report noted, "As companies become more experienced with cloud, the top challenge shifts. Security is the largest issue among cloud beginners, while cost becomes a bigger challenge for intermediate and advanced users."

And in a cloud analytics survey conducted by vendor Teradata, 46 percent of those surveyed pointed to increased security as a potential *benefit* rather than a challenge of cloud computing.

Cloud Computing Challenge 2: Managing Cloud Spending

As previously mentioned, the RightScale report found that for some organizations managing cloud spending has overtaken security as the top cloud computing challenge. By their own estimates, companies are wasting about 30 percent of the money they spend on the cloud.

Organizations make a number of mistakes that can help drive up their costs. Often, developers or other IT workers spin up a cloud instance meant to be used for a short period of time and forget to turn it back off. And many organizations find themselves stymied by the inscrutable cloud pricing

schemes that offer multiple opportunities for discounts that organizations might not be utilizing.

Multiple technological solutions can help companies with cloud cost management challenges. For example cloud cost management solutions, automation, containers, serverless services, autoscaling features and the many management tools offered by the cloud vendors may help reduce the scope of the problem. Some organizations have also found success by creating a central cloud team to manage usage and expenses.

Cloud Computing Challenge 3: Lack of Resources/Expertise

Lack of resources and expertise ranked just behind security and cost management among the top cloud implementation challenges in the RightScale survey. Nearly three-quarters (73 percent) of respondent listed it as a challenge with 27 percent saying it was a significant challenge.

While many IT workers have been taking steps to boost their cloud computing expertise, employers continue to find it difficult to find workers with the skills they need. And that trend seems likely to continue. The Robert Half Technology 2018 Salary Guide noted, "Technology workers with knowledge of the latest developments in cloud, open source, mobile, big data, security and other technologies will only become more valuable to businesses in the years ahead.

Cloud Computing Challenge 4: Governance

Governance and control were fourth in the list of cloud computing challenges in the RightScale survey with 71 percent of respondents calling it a challenge, including 25 percent who see it as a significant challenge.

In this case, one of the greatest benefits of cloud computing — the speed and ease of deploying new computing resources — can become a potential downfall. Many organizations lack visibility into the "shadow IT" used by their employees, and governance becomes particularly challenging in hybrid cloud and multi-cloud environments.

Cloud Computing Challenge 5: Compliance

The recent flurry of activity surrounding the EU General Data Protection Regulation (GDPR) has returned compliance to the forefront for many enterprise IT teams. Among those surveyed by RightScale, 68 percent cited

compliance as a top cloud computing challenge, and 21 percent called it a significant challenge.

Interestingly, one aspect of the GDPR law may make compliance easier in the future. The law requires many organizations to appoint a data protection officer who oversees data privacy and security. Assuming these individuals are well-versed in the compliance needs for the organizations where they work, centralizing responsibility for compliance should help companies meet any legal or statutory obligations.

Cloud Computing Challenge 6: Managing Multi-Cloud Environments

Most organizations aren't using just one cloud. According to the RightScale findings, 81 percent of enterprises are pursuing a multi-cloud strategy, and 51 percent have a hybrid cloud strategy (public and private clouds integrated together). In fact, on average, companies are using 4.8 different public and private clouds.

Multi-cloud environments add to the complexity faced by the IT team. To overcome this challenge, experts recommend best practices like doing research, training employees, actively managing vendor relationships and re-thinking processes and tooling.

Cloud Computing Challenge 7: Migration

While launching a new application in the cloud is a fairly straightforward process, moving an existing application to a cloud computing environment is far more difficult. A Dimensional Research study sponsored by Velostrata found that 62 percent of those surveyed said their cloud migration projects were more difficult than expected. In addition, 64 percent of migration projects took longer than expected, and 55 percent exceeded their budgets.

Cloud Computing Challenge 8: Vendor Lock-In

Currently, a few vendors, namely Amazon Web Services, Microsoft Azure, Google Cloud Platform and IBM Cloud, dominate the public cloud market. For both analysts and enterprise IT leaders, this raises the specter of vendor lock-in.

In a Stratoscale Hybrid Cloud Survey, more than 80 percent of those surveyed expressed moderate to high levels of concern about the problem.

"The increasing dominance of the hyperscale IaaS providers creates both enormous opportunities and challenges for end users and other market participants," said Sid Nag, research director at Gartner.

Cloud Computing Challenge 9: Immature Technology

Many cloud computing services are on the cutting edge of technologies like artificial intelligence, machine learning, augmented reality, virtual reality and advanced big data analytics. The potential downside to access to this new and exciting technology is that the services don't always live up to enterprise expectations in terms of performance, usability and reliability.

In the Teradata survey, 83 percent of the large enterprises surveyed said that the cloud was the best place to run analytics, but 91 percent said analytics workloads weren't moving to the cloud as quickly as they should. Part of the problem, cited by 49 percent of respondents, was immature or low-performing technology.

Cloud Computing Challenge 10: Integration

Lastly, many organizations, particularly those with hybrid cloud environments report challenges related to getting their public cloud and on-premise tools and applications to work together. In the Teradata survey, 30 percent of respondents said connecting legacy systems with cloud applications was a barrier to adoption.

Similarly, in a Software One report on cloud spending, 39 percent of those surveyed said connecting legacy systems was one of their biggest concerns when using the cloud.

Broad Approaches to Migrating into Cloud:

4. Why Migrate?

There are many problems that moving to the cloud can solve. Here are some typical scenarios that will benefit from cloud migration.

1. Flexibility

Cloud-based services are ideal for businesses with growing or fluctuating bandwidth demands. If your needs increase it’s easy to scale up your cloud

capacity, drawing on the service's remote servers. Likewise, if you need to scale down again, the flexibility is baked into the service. This level of agility can give businesses using cloud computing a real advantage over competitors – it's not surprising that CIOs and IT Directors rank 'operational agility' as a top driver for cloud adoption.

2. Disaster recovery

Businesses of all sizes should be investing in robust disaster recovery, but for smaller businesses that lack the required cash and expertise, this is often more an ideal than the reality. Cloud is now helping more organisations buck that trend. According to Aberdeen Group, small businesses are twice as likely as larger companies to have implemented cloud-based backup and recovery solutions that save time, avoid large up-front investment and roll up third-party expertise as part of the deal.

3. Automatic software updates

The beauty of cloud computing is that the servers are off-premise, out of sight and out of your hair. Suppliers take care of them for you and roll out regular software updates – including security updates – so you don't have to worry about wasting time maintaining the system yourself. Leaving you free to focus on the things that matter, like growing your business.

4. Capital-expenditure Free

Cloud computing cuts out the high cost of hardware. You simply pay as you go and enjoy a subscription-based model that's kind to your cash flow. Add to that the ease of setup and management and suddenly your scary, hairy IT project looks at lot friendlier. It's never been easier to take the first step to cloud adoption.

5. Increased collaboration

When your teams can access, edit and share documents anytime, from anywhere, they're able to do more together, and do it better. Cloud-based workflow and file sharing apps help them make updates in real time and gives them full visibility of their collaborations.

6. Work from anywhere

With cloud computing, if you've got an internet connection you can be at work. And with most serious cloud services offering mobile apps, you're not restricted by which device you've got to hand.

The result? Businesses can offer more flexible working perks to employees so they can enjoy the work-life balance that suits them – without productivity taking a hit. One study reported that 42% of workers would swap a portion of their pay for the ability to telecommute. On average they'd be willing to take a 6% pay cut.

7. Document control

The more employees and partners collaborate on documents, the greater the need for watertight document control. Before the cloud, workers had to send files back and forth as email attachments to be worked on by one user at a time. Sooner or later – usually sooner – you end up with a mess of conflicting file content, formats and titles.

And as even the smallest companies become more global, the scope for complication rises. According to one study, "73% of knowledge workers collaborate with people in different time zones and regions at least monthly".

When you make the move to cloud computing, all files are stored centrally and everyone sees one version of the truth. Greater visibility means improved collaboration, which ultimately means better work and a healthier bottom line. If you're still relying on the old way, it could be time to try something a little more streamlined.

8. Security

Lost laptops are a billion dollar business problem. And potentially greater than the loss of an expensive piece of kit is the loss of the sensitive data inside it. Cloud computing gives you greater security when this happens. Because your data is stored in the cloud, you can access it no matter what happens to your machine. And you can even remotely wipe data from lost laptops so it doesn't get into the wrong hands.

9. Competitiveness

Wish there was a simple step you could take to become more competitive? Moving to the cloud gives access to enterprise-class technology, for

everyone. It also allows smaller businesses to act faster than big, established competitors. Pay-as-you-go service and cloud business applications mean small outfits can run with the big boys, and disrupt the market, while remaining lean and nimble. David now packs a Goliath-sized punch.

10. Environmentally friendly

While the above points spell out the benefits of cloud computing for your business, moving to the cloud isn't an entirely selfish act. The environment gets a little love too. When your cloud needs fluctuate, your server capacity scales up and down to fit. So you only use the energy you need and you don't leave oversized carbon footprints. This is something close to our hearts at Salesforce, where we try our best to create sustainable solutions with minimal environmental impact.

5. Deciding on cloud migration

The usage of cloud storage is more profitable, but before migrating the complete storage under the cloud the basics things to checkout are-

1. Why Are You Thinking About Moving to the Cloud? It's important for your organization to understand why you're considering or moving to the cloud. The primary goal of moving to the cloud, If you don't know, you won't know if you're hitting the target or not. The cloud is no different from any other project in that you need clear goals, objectives, and projects leads, as well as executive sponsorship.

2. Which Cloud Should You Move To? To decide between the three main public cloud providers—Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP). The consideration will base upon three key areas: cost-performance optimization, security, and your business needs.

3. Which Applications Should You Migrate? Some applications will be more suitable for the cloud than others, especially applications with variable usage patterns. This leads to creates waste on-premises that you can eliminate by migrating to the cloud and taking advantage of auto-scaling. Also, technical consideration to keep in mind is whether you can identify suitable compute, storage, and network options for your application.

4. What deployment model should you use? Each of the cloud service providers offers three different deployment models: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). Each presents various levels of control, flexibility, and management. Where IaaS brings a high level of flexibility and cloud management control, it's more resource intensive. PasS requires fewer IT resources and offers a faster speed to market, but also has restrictions on technical functionality and compute resources. Although SaaS frees up IT resources and offers rapid deployment, it gives you minimal control of your applications and little to no flexibility. Determining which deployment model to use will be much easier.

5. What Organizational Changes Do You Need to Consider? Once you've decided it makes sense to move to the cloud, you'll need to determine how roles, culture, and policies will need to change. You'll need new security policies to manage sensitive data that can now be accessed from any device, anywhere. You'll need new skills to manage your infrastructure and to decide whether to train your existing team on how to be successful in their new cloud role or if you will hire new talent with cloud experience. Perhaps most importantly, you'll need to adopt a new organizational mindset based on active management. Active management of your cloud deployment is an essential.

6. The seven-step model of Migration into Cloud

Migrating an application to the cloud is not an easy task. It is important to strictly adhere to the seven step model to ensure that the process is robust and error free. The seven quick stages of migration into the cloud are outlined below.

1. Assessment

Migration starts with an assessment of the issues relating to migration, at the application, code, design, and architecture levels. Moreover, assessments are also required for tools being used, functionality, test cases, and configuration of the application. The proof of concepts for migration and the corresponding pricing details will help to assess these issues properly.

2. Isolate

The second step is the isolation of all the environmental and systemic dependencies of the enterprise application within the captive data center. These include library, application, and architectural dependencies. This step results in a better understanding of the complexity of the migration.

3. Map

A mapping construct is generated to separate the components that should reside in the captive data center from the ones that will go into the cloud.

4. Re-architect

It is likely that a substantial part of the application has to be re-architected and implemented in the cloud. This can affect the functionalities of the application and some of these might be lost. It is possible to approximate lost functionality using cloud runtime support API.

5. Augment

The features of cloud computing service are used to augment the application.

6. Test

Once the augmentation is done, the application needs to be validated and tested. This is to be done using a test suite for the applications on the cloud.

New test cases due to augmentation and proof-of-concepts are also tested at this stage.

7. Optimise

The test results from the last step can be mixed and so require iteration and optimization. It may take several optimizing iterations for the migration to be successful. It is best to iterate through this seven step model as this will ensure the migration to be robust and comprehensive.

7. Migration Risks and Mitigation

In the Seven-Step Model of Migration into the Cloud, the process step of testing and validating includes efforts to identify the key migration risks. In the optimization step, we address various approaches to mitigate the identified migration risks.
Migration risks for migrating into the cloud fall under two broad categories: the general migration risks and the security-related migration risks.

In the former case: we address several issues including performance monitoring and tuning—essentially identifying all possible production level deviants; the business continuity and disaster recovery in the world of cloud computing service; the compliance with standards and governance issues; the IP and licensing issues; the quality of service (QoS) parameters as well as the corresponding SLAs committed to; the ownership, transfer, and storage of data in the application; the portability and interoperability issues which could help mitigate potential vendor lock-ins.

On the security front, the cloud migration risks are plenty. Issues include security at various levels of the enterprise application as applicable on the cloud in addition to issues of trust and issues of privacy.

There are several legal compliances that a migration strategy and implementation has to fulfill, including obtaining the right execution logs as well as retaining the rights to all audit trails at a detailed level—which currently may not be fully available. On matters of governance, there are several shortcomings in the current cloud computing service vendors.

Matters of multi-tenancy and the impact of IT data leakage in the cloud computing environments is acknowledged; however, the robustness of the

solutions to prevent it is not fully validated. Key aspects of vulnerability management and incident responses quality are yet to be supported in a substantial way by the cloud service vendors.

8. Managing Cloud Services

Performance management is all about how your software services run effectively inside your own environment and through the cloud.

If you start to connect software that runs in your own data center directly to software that runs in the cloud, you create a potential bottleneck at the point of connection.

Services connected between the cloud and your computing environment can impact performance if they aren't well planned. This is especially likely to be the case if there are data translations or specific protocols to adhere to at the cloud gateway.

As a customer, your ability to directly control the resources will be much lower in the cloud. Therefore,

- The connection points between various services must be monitored in real time. A breakdown may impact your ability to provide a business process to your customers.
- There must be expanded bandwidth at connection points.

Provisioning of cloud computing services

With Software as a Service (SaaS), a customer expects provisioning (to request a resource for immediate use) of extra services to be immediate, automatic, and effortless. The cloud service provider is responsible for maintaining an agreed-on level of service and provisions resources accordingly.

The normal situation in a data center is that software workloads vary throughout the day, week, month, and year. So the data center has to be built for the maximum possible workload, with a little bit of extra capacity thrown in to cover unexpectedly high peaks.

Cloud computing service management

Service management in this context covers all the data center operations activities. This broad discipline considers the necessary techniques and tools for managing services by both cloud providers and the internal data center managers across these physical, IT and virtual environments.

Service management encompasses many different disciplines, including

- Configuration management
- Asset management
- Network management
- Capacity planning
- Service desk
- Root cause analysis
- Workload management
- Patch and update management

Administering Cloud Services:

9. Service Level Agreements (SLA) and Monitoring Support

Service Level Agreements (SLAs) in the Cloud. Most people require a blueprint for architects and contractors to start building a new home and similarly would expect a new car to come with a warranty. An SLA serves as both the blueprint and warranty for cloud computing.

In order to survive in today's world, one must be able to expect the unexpected as there are always new, unanticipated challenges. The only way to consistently overcome these challenges is to create a strong initial set of ground rules, and plan for exceptions from the start. Challenges can come from many fronts, such as networks, security, storage, processing power, database/software availability or even legislation or regulatory changes. As cloud customers, we operate in an environment that can spans geographies, networks, and systems. It only makes sense to agree on the desired service level for your customers and measure the real results. It only makes sense to set out a plan for when things go badly, so that a minimum level of service is maintained. Businesses depend on computing systems to survive.

In some sense, the SLA sets expectations for both parties and acts as the roadmap for change in the cloud service – both expected changes and surprises. Just as any IT project would have a roadmap with clearly defined deliverables, an SLA is equally critical for working with cloud infrastructure. That raises the next question in the journey: what should be in the SLA?

In order to consistently develop an effective SLA, a list of important criteria needs to be established. Let's start with an initial list:

- Availability (e.g. 99.99% during work days, 99.9% for nights/weekends)
- Performance (e.g. maximum response times)
- Security / privacy of the data (e.g. encrypting all stored and transmitted data)
- Disaster Recovery expectations (e.g. worse case recovery commitment)
- Location of the data (e.g. consistent with local legislation)
- Access to the data (e.g. data retrievable from provider in readable format)
- Portability of the data (e.g. ability to move data to a different provider)
- Process to identify problems and resolution expectations (e.g. call center)
- Change Management process (e.g. changes – updates or new services)
- Dispute mediation process (e.g. escalation process, consequences)
- Exit Strategy with expectations on the provider to ensure smooth transition

With a core set of criteria established, the next step is to evaluate the criticality of the cloud service and associated data. Nearly any computing system can be made extremely reliable, but the costs may be too high. Not every system needs the

same degree of reliability as NASA designed for the space shuttles, and few could afford the costs.

For example, providing a read-only catalogue for customers is simple. While the catalogue may be very high value, it is easy to restore from backup with minimal customer impact. However, if the same service has an online shopping with financial transactions and customer data, then the risk level and importance to the business just increased. The nature of the service is integral to determining the right SLA.

The SLA should act as a guide for handling potential problems. We need to look at the SLA as a tool for protecting the stability of the service, protecting the assets of the company and minimizing the expense should drastic actions be required. As an example, changing service providers and undoing the contracts in place, should be a last resort; it's a very expensive and painful solution. Nonetheless, it needs to be covered in the SLA so that both parties can disengage a lawsuit.

10. Managing Cloud Resources

Cloud management is the management of cloud computing products and services.

Public clouds are managed by public cloud service providers, which include the public cloud environment's servers, storage, networking and data center operations.[1] Users may also opt to manage their public cloud services with a third-party cloud management tool.

Users of public cloud services can generally select from three basic cloud provisioning categories:

- User self-provisioning: Customers purchase cloud services directly from the provider, typically through a web form or console interface. The customer pays on a per-transaction basis.
- Advanced provisioning: Customers contract in advance a predetermined amount of resources, which are prepared in advance of service. The customer pays a flat fee or a monthly fee.
- Dynamic provisioning: The provider allocates resources when the customer needs them, then decommissions them when they are no longer needed. The customer is charged on a pay-per-use basis.

Managing a private cloud requires software tools to help create a virtualized pool of compute resources, provide a self-service portal for end users and handle security, resource allocation, tracking and billing.[2] Management tools for private clouds tend to be service driven, as opposed to resource driven, because cloud environments are typically highly virtualized and organized in terms of portable workloads.[3]

In hybrid cloud environments, compute, network and storage resources must be managed across multiple domains, so a good management strategy should start by defining what needs to be managed, and where and how to do it.[4] Policies to help govern these domains should include configuration and installation of images, access control, and budgeting and reporting.[4] Access control often includes the use of Single sign-on (SSO), in which a user logs in once and gains access to all systems without being prompted to log in again at each of them.

A cloud management system combines software and technologies in a design for managing cloud environments.[5] Software developers have responded to the management challenges of cloud computing with cloud management systems.[6]

At a minimum, a cloud-management system should have the ability to:[7]

- manage a pool of heterogeneous compute-resources
- provide access to end users for consuming IT Resources with a multi-tenant approach
- monitor availability
- audit logs
- resource allocation
- manage resource allocation
- tracking spending
- automating cloud consumption

UNIT – 3

Web Service-

A web service is a collection of open protocols and standards used for exchanging data between applications or systems. Software applications are written in

various programming languages and are running on various platforms by the use web services to exchange data over computer networks like the Internet in a manner like inter-process communication on a single computer. This interoperability is due to the use of open standards.

1. **SOAP/WSDL Web Services-**

SOAP Web Services-

SOAP stands for Simple Object Access Protocol. It is an XML-based protocol used for accessing web services.

SOAP is a recommendation for communication between two applications.

SOAP is an XML based protocol. It is platform independent and language independent. By using SOAP, the possibility for interacting with other programming language applications is possible.

Advantages of Soap Web Services

i. **WS Security**: SOAP defines its own security known as WS Security.
ii. **Language and Platform independent**: SOAP web services can be written in any programming language and executed in any platform.

Disadvantages of Soap Web Services

i. **Slow**: SOAP uses XML format that must be parsed to be read. It defines many standards that must be followed while developing the SOAP applications. So it is slow and consumes more bandwidth and resource.
ii. **WSDL dependent**: SOAP uses WSDL and doesn't have any other mechanism to discover the service.

WSDL

WSDL is an acronym for Web Services Description Language.

WSDL is an xml document containing information about web services such as method name, method parameter and how to access it.

WSDL acts as an interface between web service applications.

WSDL document describes a web service. It specifies the location of the service, and the methods of the service, using major elements such as-

i. <types>
ii. <message>
iii. <portType>
iv. <binding>

The structure of a WSDL document is:

```
<definitions>
<types>
  data type definitions.....
</types>
<message>
  definition of the data being communicated....
</message>
<portType>
  set of operations......
</portType>
<binding>
  protocol and data format specification....
</binding>
</definitions>
```

WSDL Example

```
<message name="getTermRequest">
  <part name="term" type="xs:string"/>
</message>
```

```
<message name="getTermResponse">
  <part name="value" type="xs:string"/>
</message>
<portType name="glossaryTerms">
  <operation name="getTerm">
    <input message="getTermRequest"/>
    <output message="getTermResponse"/>
  </operation>
</portType>
```

The <portType> element defines **a web service**, the **operations** that can be performed, and the **messages** that are involved.

WSDL defines four types of responses. They are-

i. One-way
ii. Request-response
iii. Solicit-response
iv. Notification

The request-response type is the most common operation type.

WSDL One-Way Operation

A one-way operation example:

```
<message name="newTermValues">
  <part name="term" type="xs:string"/>
  <part name="value" type="xs:string"/>
</message>
<portType name="glossaryTerms">
  <operation name="setTerm">
    <input name="newTerm" message="newTermValues"/>
  </operation>
</portType >
```

In the example, the portType "glossaryTerms" defines a one-way operation called "setTerm".

The "setTerm" operation allows input of new glossary terms messages using a "newTermValues" message with the input parameters "term" and "value". However, no output is defined for the operation.

WSDL Request-Response Operation

A request-response operation example:

```
<message name="getTermRequest">
  <part name="term" type="xs:string"/>
</message>
<message name="getTermResponse">
  <part name="value" type="xs:string"/>
</message>
<portType name="glossaryTerms">
  <operation name="getTerm">
    <input message="getTermRequest"/>
    <output message="getTermResponse"/>
  </operation>
</portType>
```

In the example, the portType "glossaryTerms" defines a request-response operation called "getTerm".

The "getTerm" operation requires an input message called "getTermRequest" with a parameter called "term" and will return an output message called "getTermResponse" with a parameter called "value".

WSDL Binding to SOAP

WSDL bindings defines the message format and protocol details for a web service.

A request-response operation example:

```
<message name="getTermRequest">
  <part name="term" type="xs:string"/>
</message>
```

```
<message name="getTermResponse">
  <part name="value" type="xs:string"/>
</message>

<portType name="glossaryTerms">
  <operation name="getTerm">
    <input message="getTermRequest"/>
    <output message="getTermResponse"/>
  </operation>
</portType>

<binding type="glossaryTerms" name="b1">
   <soap:binding style="document"
   transport="http://schemas.xmlsoap.org/soap/http" />
   <operation>
     <soap:operation soapAction="http://example.com/getTerm"/>
     <input><soap:body use="literal"/></input>
     <output><soap:body use="literal"/></output>
  </operation>
</binding>
```

The **binding** element has two attributes - name and type.

The name attribute (you can use any name you want) defines the name of the binding, and the type attribute points to the port for the binding, in this case the "glossaryTerms" port.

The **soap:binding** element has two attributes - style and transport.

The style attribute can be "rpc" or "document". In this case we use document. The transport attribute defines the SOAP protocol to use. In this case we use HTTP.

2. REST Web Services

REST stands for REpresentational State Transfer. REST is web standard based architecture and uses HTTP Protocol. It revolves around resource where every component is a resource and a resources are accessed by a common interface using HTTP standard methods.

REST was first introduced by Roy Fielding in 2000.

In REST architecture, a REST Server simply provides access to resources and REST client accesses and modifies the resources. Here each resource is identified by URIs or global IDs. REST uses various representations to represent a resource like text or XML.

HTTP methods

There are four HTTP methods that are commonly used in REST based architecture. They are-

- **GET** − Provides a read only access to a resource.
- **POST** − Used to create a new resource.
- **DELETE** − Used to remove a resource.
- **PUT** − Used to update a existing resource or create a new resource.

Web services based on REST Architecture are known as RESTful web services. These webservices uses HTTP methods to implement the concept of REST architecture. A RESTful web service usually defines a URI, Uniform Resource Identifier a service, provides resource representation such as set of HTTP Methods.

Advantages of RESTful Web Services

i. **Fast**: RESTful Web Services are fast because there is no strict specification like SOAP. It consumes less bandwidth and resource.
ii. **Language and Platform independent**: RESTful web services can be written in any programming language and executed in any platform.
iii. **Can use SOAP**: RESTful web services can use SOAP web services as the implementation.

3. SOAP v/s REST

There are many differences between SOAP and REST web services. They are-

No.	SOAP	REST
1)	SOAP is a **protocol**.	REST is an **architectural style**.
2)	SOAP stands for **Simple Object Access Protocol**.	REST stands for **REpresentational State Transfer**.
3)	SOAP **can't use REST** because it is a protocol.	REST **can use SOAP** web services because it is a concept and can use any protocol like HTTP, SOAP.
4)	SOAP **uses services interfaces to expose the business logic**.	REST **uses URI to expose business logic**.
5)	**JAX-WS** is the java API for SOAP web services.	**JAX-RS** is the java API for RESTful web services.
6)	SOAP **defines standards** to be strictly followed.	REST does not define too much standards like SOAP.
7)	SOAP **requires more bandwidth** and resource than REST.	REST **requires less bandwidth** and resource than SOAP.

8)	SOAP **defines its own security**.	RESTful web services **inherits security measures** from the underlying transport.
9)	SOAP **permits XML** data format only.	REST **permits different** data format such as Plain text, HTML, XML, JSON etc.
10)	SOAP is **less preferred** than REST.	REST **more preferred** than SOAP.

AJAX

4. Asynchronous 'rich' interfaces

Rich user interfaces are those, which can be achieved by using a combination of dynamic HTML elements such as HTML and JavaScript. However, the scope of such an interface is limited to client-side behavior and has minimal functional implications due to the lack of server-side interactions.

The power of AJAX is in its capability to provide even richer interface by supplementing its dynamic user interface with powerful functionality through seamless server-side invocation power.

AJAX allows individual user interface components to communicate with the server and exchange data without the need for refreshing the whole screen. This is achieved using a process called Web Remoting.

Web remoting, or the process of communicating between a browser and a server, can be performed in multiple ways. The popular approaches that are supported by browsers are IFrames and XMLHttpRequest. Dynamic HTML can be complemented with either of these methods to generate AJAX functionality.

Asynchronous AJAX

Asynchronous communication between the client and the server forms the backbone of AJAX. Although an asynchronous request-response method can provide significant value in the development of rich functionality by itself, the results are lot more pronounced when used in conjunction with other functional standards such as CSS, DOM, JavaScript, and so on.

Client-server communication can be achieved either by using IFrames, or by using the supported JavaScript function call XMLHttpRequest().

Due to certain limitations of IFrames, XMLHttpRequest has gained a lot more acceptance. While IFrame can also be an effective option for implementing AJAX-based solutions.

The primary advantage of using AJAX-based interfaces is that the update of content occurs without the page refresh.

A typical AJAX implementation using XMLHttpRequest happens as described in the following steps:

1. An action on the client side, whether this is a mouse click or a timed refresh, triggers a client event
2. An XMLHttpRequest object is created and confi gured
3. The XMLHttpRequest object makes a call
4. The request is processed by a server-side component
5. The component returns an XML (or an equivalent) document containing the result
6. The XMLHttpRequest object calls the callback() function and processes the result
7. The HTML DOM is updated with any resulting values

Mashups

5. User interface services

Mashup

A mashup is a technique by which a website or a Web application uses data, presentation or functionality from two or more sources to create a new service.

Mashups are made possible via Web services or public APIs that allows for the free access.

Most mashups are visual and interactive in nature.

To a user, a mashup should provide a richer and more interactive experience. A mashup is also beneficial to the developers because it requires less code, allowing for a quicker development cycle.

It is also the way through which a user interacts with an application or a website. The growing dependence of many companies on web applications and mobile applications has led many companies to place increased priority on user interface as an effort to improve the user's overall experience.

User interface is an interactive platform that is used to intact the user and an application together. The increasing dependence of many companies on web applications and mobile applications have led many enterprises to increase the priority on user interfaces, to improve the overall user application. The first user interface ever developed was the command line interface, which was nearly a blank screen with a line for user input. This evolution of user interface technology has led to development of graphical user interface which made many of the system designers to include icons, images, text, visuals and multimedia, which interacted with the user, creating a sense of human to machine conversation.

Three market trends provide the main drivers for the change of user interface and interaction technologies. They are namely visual info-gratification, the user as the new interface, and smarter devices, emerging user interfaces for an immersive user experience.

The major factors that drive the growth of user interface service include growing digital technology, increasing penetration of smartphones and tablets, and growing demand for data monitoring and controlling applications, in an automotive industry.

In recent, times, the emerging popularity of mobile applications has also affected the user interface. Mobile user interfaces are generally concerned with creating an usable, interactive interfaces for smaller screens like mobile phones and tablets.

The increasing consumer demand is the major factor that drives the growth of the market. Many enterprises such as Microsoft, designed a dual interface for their Windows 8 operating system, which created a sense of confusion and not user-friendly. The interface was different for touch controls and different for general input control, which was not liked by the users across the globe.

Cloud Technologies

6. Study of Hypervisor

A **hypervisor** or **virtual machine monitor** (**VMM**) is computer software, firmware or hardware that creates and runs virtual machines. A computer on which a hypervisor runs one or more virtual machines is called a *host machine*, and each virtual machine is called a *guest machine*.

The hypervisor presents the guest operating systems with a virtual operating platform and manages the execution of the guest operating systems.

Multiple instances of a variety of operating systems may share the virtualized hardware resources: for example, Linux, Windows, and macOS instances can all run on a single physical x86 machine.

This contrasts with operating-system-level virtualization, where all instances must share a single kernel, though the guest operating systems can differ in user space, such as different Linux distributions with the same kernel.

In their 1974 article, *Formal Requirements for Virtualizable Third Generation Architectures*, Gerald J. Popek and Robert P. Goldberg classified two types of hypervisors-

Type-1 Native or Bare-Metal Hypervisors

These hypervisors run directly on the host's hardware to control the hardware and to manage guest operating systems. For this reason, they are sometimes called bare metal hypervisors. The first hypervisors, which IBM developed in the 1960s, were the native hypervisors. These included the test software SIMMON and the CP/CMS operating system.

Type-2 Hosted Hypervisors

These hypervisors run on a conventional operating system just as other computer programs do. A guest operating system runs as a process on the host. These hypervisors abstract guest operating systems from the host operating system. VMware Workstation, VMware Player, VirtualBox, Parallels Desktop for Mac and QEMU are examples of type-2 hypervisors.

The distinction between these two types is not always clear. For instance, Linux's Kernel-based Virtual Machine (KVM) and FreeBSD's are kernel modules that effectively convert the host operating system to a type-1 hypervisor. At the same time, since Linux distributions and FreeBSD are still general-purpose operating systems, with applications competing with each other for VM resources can also be categorized as type-2 hypervisors.

Several factors led to a resurgence around 2005 in the use of virtualization technology among Unix, Linux, and other Unix-like operating systems:

- Expanding hardware capabilities, allowing each single machine to do more simultaneous work
- Efforts to control costs and to simplify management through consolidation of servers
- The improved security, reliability, and device independence possible from hypervisor architectures

Visualization Technology

7. Virtual Machine Technology

Virtualization-

Virtualization is the process of creating a software-based, or virtual, representation of something, such as virtual applications, servers, storage and networks. It is the single most effective way to reduce IT expenses while boosting efficiency and agility for all size businesses.

Benefits of Virtualization

Virtualization can increase IT agility, flexibility and scalability while creating significant cost savings. Greater workload mobility, increased performance and availability of resources, automated operations and the other additional benefits include:

- Reduced capital and operating costs.
- Minimized or eliminated downtime.
- Increased IT productivity, efficiency, agility and responsiveness.
- Faster provisioning of applications and resources.
- Greater business continuity and disaster recovery.
- Simplified data center management.

Virtual Machines Explained

A virtual computer system is known as a “virtual machine” (VM): a tightly isolated software container with an operating system and application inside. Each self-contained VM is completely independent. Putting multiple VMs on a single computer enables several operating systems and applications to run on just one physical server, or “host.”

A thin layer of software called a “hypervisor” decouples the virtual machines from the host and dynamically allocates computing resources to each virtual machine as needed.

Key Properties of Virtual Machines

VMs have the following characteristics, which offer several benefits.

Partitioning

- Run multiple operating systems on one physical machine.
- Divide system resources between virtual machines.

Isolation

- Provide fault and security isolation at the hardware level.

- Preserve performance with advanced resource controls.

Encapsulation

- Save the entire state of a virtual machine to files.
- Move and copy virtual machines as easily as moving and copying files.

Hardware Independence

- Provision or migrate any virtual machine to any physical server.

Types of Virtualization

Server Virtualization

Server virtualization enables multiple operating systems to run on a single physical server as highly efficient virtual machines. Key benefits include:

- Greater IT efficiencies
- Reduced operating costs
- Faster workload deployment
- Increased application performance
- Higher server availability
- Eliminated server sprawl and complexity

Network Virtualization

By completely reproducing a physical network, network virtualization allows applications to run on a virtual network as if they were running on a physical network — but with greater operational benefits and all the hardware independencies of virtualization. (Network virtualization presents logical networking devices and services — logical ports, switches, routers, firewalls, load balancers, VPNs and more — to connected workloads.)

Desktop Virtualization

Deploying desktops as a managed service enables IT organizations to respond faster to changing workplace needs and emerging opportunities. Virtualized desktops and applications can also be quickly and easily

delivered to branch offices, outsourced and offshore employees, and mobile workers using iPad and Android tablets.

8. Virtualization Applications in Enterprises

Any discussion of server virtualization software typically ends in clicking glasses and high fives, heated discussions or slap fights, but it almost always begins with VMware, as does this list. These top ten virtualization vendors deliver the best virtualization software solutions on the market today.

When it comes to server virtualization software technology offerings, you might not require every bit and byte of programming they're composed of, but you'll rejoice at the components of their feature sets when you need them.

These solutions scale from a few virtual machines that host a handful of Web sites, virtual desktops or intranet services all the way up to tens of thousands of virtual machines serving millions of Internet users. If you don't know all the virtualization software names on this list, it's time for an introduction.

1. VMware

The major data center anywhere in the world is the VMware. VMware dominates the server virtualization market. Its domination doesn't stop with its commercial product, VMware vSphere. VMware also dominates the desktop-level virtualization market and perhaps even the free server virtualization market with its VMware Server product. VMware remains in the dominant spot due to its innovations, strategic partnerships and rock-solid products.

2. Microsoft

Microsoft came up with the only non-Linux hypervisor, Hyper-V, to compete in a tight server virtualization market that VMware currently dominates. Not easily outdone in the data center space, Microsoft offers attractive licensing for its Hyper-V product and the operating systems that live on it.

For all Microsoft shops, Hyper-V is a viable solution that has only gotten more competitive in the virtualization space with each new Windows Server release. Microsoft has also been steadily gaining traction with enterprises looking to leverage the company's Azure cloud services as well as those interested in managing both on-premises Hyper-V services and Azure services.

3. Red Hat

For the past 15 years, everyone has recognized Red Hat as an industry leader and open source champion. Hailed as the most successful open source company, Red Hat entered the world of virtualization in 2008 when it purchased Qumranet and with it, its own virtual solution: KVM and SPICE (Simple Protocol for Independent Computing Environment). Red Hat released the SPICE protocol as open source in December 2009.

The company's renowned Red Hat Enterprise Virtualization (RHEV) desktop and server virtualization platform is based on the KVM hypervisor and Red Hat's Enterprise Linux (RHEL) server operating system. RHEV is based on open standards and works with Linux and Windows, as well as enterprise applications like SAP, SAS and Oracle.

4. Oracle

If Oracle's world domination of the enterprise database server market doesn't impress you, its acquisition of Sun Microsystems has made it an impressive virtualization player. Additionally, Oracle owns an operating system (Sun Solaris), multiple virtualization software solutions and server hardware.

5. Amazon

Amazon's Elastic Compute Cloud (EC2) is the industry-standard virtualization platform. Ubuntu's Cloud Server supports seamless integration with Amazon's EC2 services. EngineYard's Ruby application services leverage Amazon's cloud as well.

6. Google

When you think of Google, virtualization might not make the top of the list of things that come to mind, but its Google Apps, AppEngine and extensive Business Services list demonstrates how it has embraced cloud-oriented services.

The company's open source Google Ganeti cluster virtual server management software tool is built on top of existing virtualization technologies like Xen or KVM and essentially serves as a wrapper around these hypervisors to help system admins set up clusters.

9. Pitfalls of Virtualization

Pitfalls

Mismatching Servers

This aspect is commonly overlooked especially by smaller companies that don't invest enough funds in their IT infrastructure and prefer to build it from several bits and pieces. This usually leads to simultaneous virtualization of servers that come with different chip technology.

Frequently, migration of virtual machines between them won't be possible and server restarts will be the only solution. This is a major hindrance and means losing the benefits of live migration and virtualization.

Creating Too many Virtual Machines per Server

One of the great things about virtual machines is that they can be easily created and migrated from server to server according to needs. However, this can also create problems sometimes because IT staff members may get carried away and deploy more Virtual Machines than a server can handle.

This will lead to a loss of performance that can be quite difficult to spot. A practical way to work around this is to have some policies in place regarding VM limitations and to make sure that the employees adhere to them.

Misplacing Applications

A virtualized infrastructure is a more complex than a traditional one and with several applications deployed, losing track of applications is a distinct possibility. Within a physical server infrastructure keeping track of all the apps and the machines running them isn't a difficult task. However, once you add a significant number of virtual machines to the equation, things can get messy and App patching, software licensing and updating can turn into painfully long processes.

Multitenant Software

Multi-tenancy is an architecture in which a single instance of a software application serves multiple customers. Each customer is called a tenant. Tenants may be given the ability to customize some parts of the application, such as color of the user interface or business rules, but they cannot customize the application's code.

Multi-tenancy can be economical because software development and maintenance costs are shared. It can be contrasted with single-tenancy, an architecture in which each customer has their own software instance and may be given access to code. With a multi-tenancy architecture, the provider only must make updates once. With a single-tenancy architecture, the provider must touch multiple instances of the software to make updates.

In cloud computing, the meaning of multi-tenancy architecture has broadened because of new service models that take advantage of virtualization and remote access. A software-as-a-service provider, for example, can run one instance of its application on one instance of a database and provide web access to multiple customers. In such a scenario, each tenant's data is isolated and remains invisible to other tenants.

10. Multi-entity support

Multi-entity provides the capability to create and manage multiple entities within a single tenant. An entity represents a legal entity or an independently operated business unit. Each entity can manage its own

business objects, settings, transactions, and features Multi-entity allows you to:

- Log in as a corporate user to run reports and view data across entities
- Model your enterprise organization in a multi-entity hierarchy
- Keep the business operations isolated among entities
- Limit data access within different entities
- Grant user access across entities
- Define business objects centrally in a single entity, and share with other entities
- Share certain settings and business objects across entities

Multi-entity Hierarchy

Multi-entity hierarchy shows the structure of your entities. You can manage your multi-entity hierarchy to structure your global enterprise organization within a single tenant. New entities can be added or edited as company structure changes.

Business Objects Sharing Across Entities

It offers sharing and central management options to make management across entities easier. You can define accounting periods centrally in a single entity and share with other entities. You can also share products within a multi-entity hierarchy, which allows entities to cross-sell products from other entities.

User Access Across Entities

Each entity is completely isolated from other entities. A user by default can only access to the entity in which the user is created. To access the other entities, the user must be granted permission to access them. Only the entity administrators have permission to grant or deny user accesses.

Entity Switcher

The entity switcher lists all the provisioned entities in the multi-entity hierarchy. After you logged in from the Zuora UI, you can switch to the entities that you have permission to access from the entity switcher. The entity switcher is in the upper right of the page and next to your login user name. When you click the current entity display name, the multi-entity

hierarchy is displayed. You can click an entity display name from the entity switcher to access the entity. Also, you can search for an entity display name in the entity search field. The entities that you do not have permission to access are grayed out.

Multi-entity Reporting

If your organization uses Insights Analysis, you can report across multiple entities in Insights Analysis. An entity selector in the report builder enables you to specify which entities to report on, and an Entity object in each data source enables you to include entity information in reports.

11. Multi tenancy using cloud data stores

Whether an IT organization is going with public or private clouds, it's important to understand the nuances of multi-tenant architecture. For

public clouds, IT managers need to understand the degree of multi-tenancy supported by whichever vendor they are looking at. For private clouds, the entire responsibility of designing a multi-tenant architecture rests with the IT managers.

Enterprise cloud adoption has gone beyond the levels of intellectual pursuits and casual experimentation. An analysis by IDC shows that $17 billion of the $359 billion of worldwide IT spending for 2009 could be attributed to cloud computing. Two-thirds of *Baseline* magazine's survey participants plan to expand their use of public clouds.

None of that is to say that there aren't nagging issues, including but not limited to how different enterprise workloads match up against different types of clouds and responsible ways to plan and implement the necessary migrations.

Based on the characteristics of the workload, cloud adoption will swing between public and private clouds. Large enterprises have requirements that will force them to strike a balance between the two clouds for their workloads. This is different for small-to-medium businesses (SMBs) and start-ups, which might have a strong business case for wanting to use public clouds for almost all their workloads. But in the end, their respective preferences will not be as much about the size of the organization as they are about the nature of their IT workloads.

Besides appropriate workload distribution, architectural considerations are also key. Multi-tenancy is one such architectural consideration, and understanding multi-tenancy is a critical first step towards broader IT cloud adoption.

12. Data access control for enterprise applications

Enterprise Application Access is a unique cloud architecture that closes all inbound firewall ports, while ensuring that only authorized users and devices have access to the internal applications they need, and not the entire network. No one can access applications directly because they are hidden from the Internet and public exposure. Enterprise Application Access integrates data path protection, single sign-on, identity access, application security, and management visibility and control into a single service.

It can be deployed in minutes through a unified portal with a single point of control, in any network environment, and at a fraction of the cost of traditional solutions. The result is a secure-access delivery model that enables a zero CapEx, low OpEx model for critical workloads deployed in any environment.

UNIT- 4

Cloud Security Fundamentals

1. **Vulnerability assessment tool for cloud-**

When it comes to patching and mitigating vulnerabilities, knowing is half the battle. Unfortunately, most organizations don't have an effective vulnerability assessment program in place, and the businesses that need it most lack the skills and resources to manage it effectively.
Many new cloud-based vulnerabilities assessment tools are being introduced.

The tools simplify the vulnerability management by allowing organizations to scan public-facing assets to identify open vulnerabilities. It is particularly effective in helping organizations to achieve and maintain compliance with regulatory and industry compliance mandates such as PCI-DSS (Payment Card Industry Data Security Standards), or HIPAA (Health Insurance Portability and Accountability Act).

The approach of these tools also is to is to provide an attackers-eye view, by scanning the Internet-connected servers and applications from the same perspective as an attacker has when scanning the network for holes to exploit. While it's important to know where vulnerabilities exist both inside and outside the network, it's crucial to address flaws that can be exploited directly from outside the network.

The heart of tools being used is the Retina Network Security Scanner which is one of the most well-known, and widely used vulnerability scanning platforms in the industry. This enables to run manual or schedule automated vulnerability scans, and create reports that can be accessed and viewed from a Web browser.

With the release of Assessment tools, a trusted hands-free approach to vulnerability management and regulatory compliance with more accuracy and a lower annual cost is initiated.

One of the advantages of these tools is that it is entirely cloud-based, and there is no hardware or software to deploy. The platform integrates with Microsoft Live credentials, or Active Directory, and can be managed from any Web browser.

2. Privacy and Security in cloud

The security principal concerns with the entrusting of an organization's critical information to geographically dispersed cloud platforms that are not under the direct control of the organization. In addition to the conventional IT information system security procedures, designing security into cloud software during the software development life cycle can greatly reduce the cloud attack surface.

Developing secure software is based on applying the secure software design principles that form the fundamental basis for software assurance. Software assurance has been given many

definitions, and it is important to understand the concept. The Software Security Assurance Report defines software assurance as "the basis for gaining justifiable confidence that software will consistently exhibit all properties required to ensure that the software, in operation, will continue to operate dependably despite the presence of sponsored faults.

The Data and Analysis Center for Software requires that software must exhibit the following three properties to be considered secure:
Dependability — Software that executes predictably and operates correctly
under a variety of conditions, including when under attack or running on a malicious host
Trustworthiness — Software that contains a minimum number of vulnerabilities or no vulnerabilities or weaknesses that could sabotage
the software's dependability. It must also be resistant to malicious logic.

- **Integrity**

The concept of cloud information *integrity* requires that the following three
principles are met:
Modifications are not made to data by unauthorized personnel or processes.
Unauthorized modifications are not made to data by authorized personnel
or processes.
The data is internally and externally consistent — in other words, the
internal information is consistent both among all sub-entities and with
the real-world, external situation.

The cloud provides security services such as-

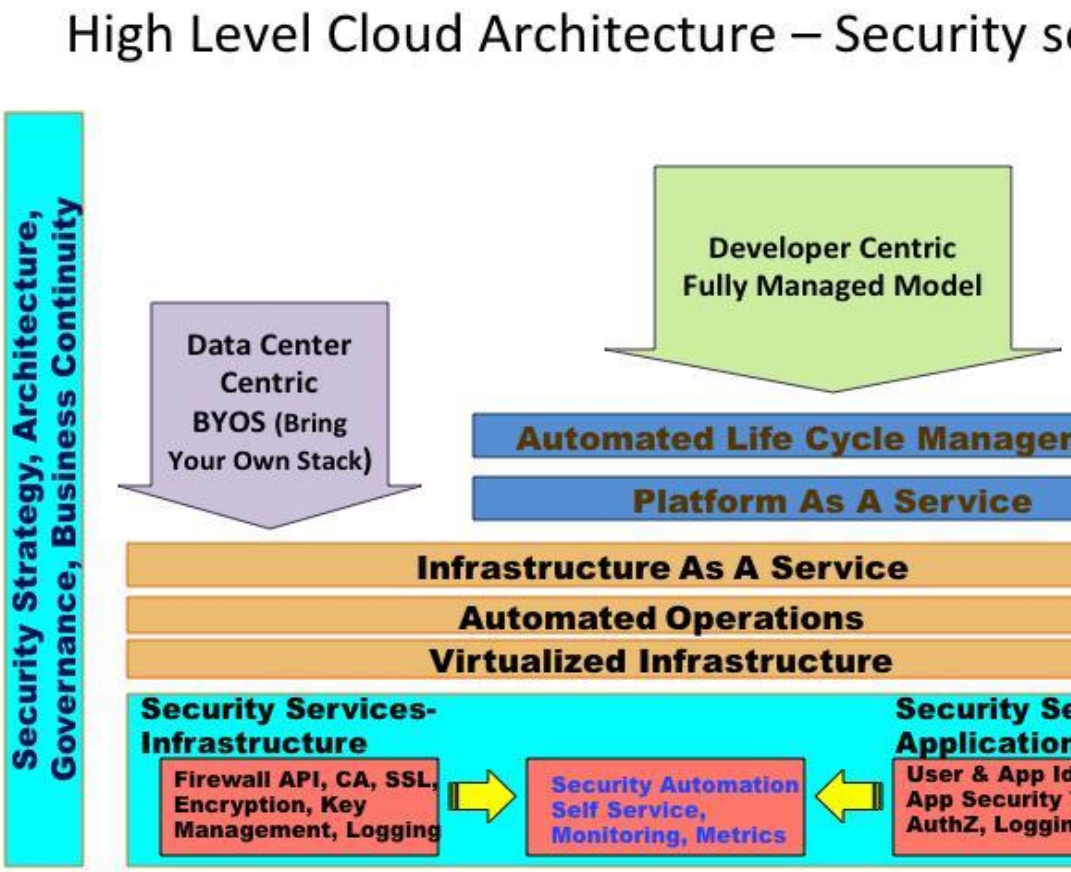
i. **Authentication**
ii. **Authorization**
iii. **Auditing**
iv. **Accountability**

The privacy conditions that are to be maintained in cloud are-

- **Handling data** — Some data is more sensitive and requires special handling.

Code practices — Care must be taken not to expose too much information
to a would-be attacker.
Language options — Consider the strengths and weakness of the language
used.
Input validation and content injection — Data (content) entered by a user
should never have direct access to a command or a query.
Physical security of the system — Physical access to the cloud servers should be restricted.

3. **Cloud computing security architecture:**



Explain each term in own manner..

i) Architectural Considerations-

a) General Issues

A variety of topics influence and directly affect the cloud security architecture.
They include such factors as compliance, security management, administrative
issues, controls, and security awareness.
Compliance with legal regulations should be supported by the cloud security
architecture. As a corollary, the cloud security policy should address classification of information, what entities can potentially access information, under what conditions the access must be provided, the geographical jurisdiction of
the stored data, and whether the access is appropriate. Proper controls
should be determined and verified with assurance methods, and appropriate
personnel awareness education should be put in place.

i. **Compliance**
ii. **Security management**
iii. **Information classification**
iv. **Employee termination**

b) Trusted Cloud computing

Trusted cloud computing can be viewed as a computer security architecture
that is designed to protect cloud systems from malicious intrusions and attacks,
and ensure that computing resources will act in a specifi c, predictable manner
as intended. A trusted cloud computing system will protect data in use by
hypervisors and applications protect against unauthorized access to information,
provide for strong authentication, apply encryption to protect sensitive
data that resides on stolen or lost devices, and support compliance through
hardware and software mechanisms.

c) Secure Execution Environments and Communications

Secure Execution Environment

Configuring computing platforms for secure execution is a complex task; and in many instances it is not performed properly because of the large number of parameters that are involved. This provides opportunities for malware to exploit vulnerabilities, such as downloading code embedded in data and having the code executed at a high privilege level.

In cloud computing, the major burden of establishing a secure execution environment is transferred from the client to the cloud provider. However, protected data transfers must be established through strong authentication mechanisms, and the client must have practices in place to address the privacy and confidentiality of information that is exchanged with the cloud. In fact, the client's port to the cloud might provide an attack path if not properly provisioned with security measures. Therefore, the client needs assurance that computations and data exchanges are conducted in a secure environment. This assurance is affected by trust enabled by cryptographic methods. Also, research into areas such as compiler-based virtual machines promises a more secure execution environment for operating systems.

Another major concern in secure execution of code is the widespread use of "unsafe" programming languages such as C and C++ instead of more secure languages such as object-oriented Java and structured, object-oriented C#.

Secure Communications

As opposed to having managed, secure communications among the computing resources internal to an organization, movement of applications to the cloud requires a reevaluation of communications security. These communications apply to both data in motion and data at rest.

Secure cloud communications involve the structures, transmission methods, transport formats, and security measures that provide confidentiality, integrity, availability, and authentication for transmissions over private and public communications networks. Secure cloud computing communications should ensure the following:

- **Confidentiality**
- **Integrity**
- **Availability**

d) Micro-architectures

The term *computer architecture* refers to the organization of the fundamental elements composing the computer. From another perspective, it refers to the view a programmer has of the computing system when viewed through its instruction set. The main hardware components of a digital computer are the central processing unit (CPU), memory, and input/output devices. A basic CPU of a general-purpose digital computer consists of an arithmetic logic unit (ALU), control logic, one or more accumulators, multiple general-purpose registers, an instruction registers, a program counter, and some on-chip local memory. The ALU performs arithmetic and logical operations on the binary words of the computer. The design elements of the microprocessor hardware and firmware that provide for the implementation of the higher-level architecture are referred to as *microarchitecture*. As an example, a microarchitecture design might incorporate the following:

- **Pipelining** — Increases the performance of a computer by overlapping the steps of different instructions. For example, if the instruction cycle is divided into three parts — fetch, decode, and execute — instructions can

be overlapped (as shown in Figure 6-8) to increase the execution speed
of the instructions.

Superscalar processor — A processor that enables the concurrent execution
of multiple instructions in both the same pipeline stage as well as different pipeline stages.

- **Very-long instruction word (VLIW) processor** — A processor in which
a single instruction specifi es more than one concurrent operation. For**Multi-programming** — Executes two or more programs simultaneously
on a single processor (CPU) by alternating execution among the programs.
- **Multi-tasking** — Executes two or more subprograms or tasks at the same
time on a single processor (CPU) by alternating execution among the tasks.
- **Multi-processing** — Executes two or more programs at the same time on
multiple processors.

4. Identity Management and Access control Identity management

Identity management and access control are fundamental functions required for secure cloud computing. The simplest form of identity management is logging on to a computer system with a user ID and password. However, identity management, such as is required for cloud computing, requires more robust authentication, authorization, and access control. It should determine what resources are authorized to be accessed by a user or process by using technology such as biometrics or smart cards, and determine when a resource has been accessed by unauthorized entities.

Identity Management

Identification and authentication are the keystones of most access control systems. Identification is the act of a user professing an identity to a system, usually in the form of a username or user logon ID to the system. Identification establishes user accountability for the actions on the system. User IDs should be unique and not shared among different individuals. In many large organizations,

user IDs follow set standards, such as first initial followed by last name,
and so on. To enhance security and reduce the amount of information
available to an attacker, an ID should not reflect the user's job title or function.
Authentication is verification that the user's claimed identity is valid, and it
is usually implemented through a user password at logon. Authentication is
based on the following three factor types:
Type 1 — Something you know, such as a personal identification number
(PIN) or password
Type 2 — Something you have, such as an ATM card or smart card
Type 3 — Something you are (physically), such as a fingerprint or retina
scan
Sometimes a fourth factor, something you do, is added to this list. Something
you do might be typing your name or other phrases on a keyboard. Conversely,
something you do can be considered something you are.
Two-factor authentication requires two of the three factors to be used in the
authentication process. For example, withdrawing funds from an ATM machine
requires two-factor authentication in the form of the ATM card (something you
have) and a PIN number (something you know).

Passwords

Because passwords can be compromised, they must be protected. In the ideal
case, a password should be used only once. This "one-time password," or OTP,
provides maximum security because a new password is required for each new
logon. A password that is the same for each logon is called a *static password*. A

password that changes with each logon is termed a *dynamic password*. The changing
of passwords can also fall between these two extremes. Passwords can be required to change monthly, quarterly, or at other intervals, depending on the
criticality of the information needing protection and the password's frequency
of use. Obviously, the more times a password is used, the more chance there
is of it being compromised. A *passphrase* is a sequence of characters that is usually
longer than the allotted number for a password. The passphrase is converted
into a virtual password by the system.
In all these schemes, a front-end authentication device or a back-end authentication
server, which services multiple workstations or the host, can perform
the authentication.
Passwords can be provided by several devices, including tokens, memory
cards, and smart cards.

Memory Cards

Memory cards provide nonvolatile storage of information, but they do not have
any processing capability. A memory card stores encrypted passwords and other
related identifying information. A telephone calling card and an ATM card are
examples of memory cards.

Smart Cards

Smart cards provide even more capability than memory cards by incorporating
additional processing power on the cards. These credit-card-size devices
comprise microprocessor and memory and are used to store digital signatures,
private keys, passwords, and other personal information.

Biometrics

An alternative to using passwords for authentication in logical or technical

access control is *biometrics*. Biometrics is based on the Type 3 authentication mechanism — something you are. Biometrics is defined as an automated means of identifying or authenticating the identity of a living person based on physiological or behavioral characteristics. In biometrics, identification is a one-to-many search of an individual's characteristics from a database of stored images. Authentication is a one-to-one search to verify a claim to an identity made by a person. Biometrics is used for identification in physical controls and for authentication in logical controls.

Implementing Identity Management

Realizing effective identity management requires a high-level corporate commitment and dedication of sufficient resources to accomplish the task. Typical undertakings in putting identity management in place include the following:

- Establishing a database of identities and credentials
- Managing users' access rights
- Enforcing security policy
- Developing the capability to create and modify accounts
- Setting up monitoring of resource accesses
- Installing a procedure for removing access rights
- Providing training in proper procedures

5. Access control

Access control is intrinsically tied to identity management and is necessary to
preserve the confidentiality, integrity, and availability of cloud data.
These and other related objectives flow from the organizational security
policy. This policy is a high-level statement of management intent regarding
the control of access to information and the personnel who are authorized to
receive that information.
Three things that must be considered for the planning and implementation of
access control mechanisms are threats to the system, the system's vulnerability
to these threats, and the risk that the threats might materialize. These concepts
are defined as follows:
Threat — An event or activity that has the potential to cause harm to the
information systems or networks
Vulnerability — A weakness or lack of a safeguard that can be exploited
by a threat, causing harm to the information systems or networks
Risk — The potential for harm or loss to an information system or network;
the probability that a threat will materialize
Controls
Controls are implemented to mitigate risk and reduce the potential for loss.
Two important control concepts are *separation of duties* and the principle of *least*
privilege. Separation of duties requires an activity or process to be performed by

two or more entities for successful completion. Thus, the only way that a security
policy can be violated is if there is collusion among the entities. For example, in
a financial environment, the person requesting that a check be issued for payment
should not also be the person who has authority to sign the check. Least
privilege means that the entity that has a task to perform should be provided
with the minimum resources and privileges required to complete the task for
the minimum necessary period.

Models for Controlling Access

Controlling access by a subject to an object involves setting up access rules. These
rules can be classified into three categories or models.

Mandatory Access Control

The authorization of a subject's access to an object depends upon labels, which
indicate the subject's *clearance*, and the *classification or sensitivity* of the object. For
example, the military classifies documents as unclassified, confidential, secret,
and top secret. Similarly, an individual can receive a clearance of confidential,
secret, or top secret and can have access to documents classified at or below his
or her specified clearance level. Thus, an individual with a clearance of "secret"
can have access to secret and confidential documents with a restriction. This
restriction is that the individual must have a *need to know* relative to the classified documents involved. Therefore, the documents must be necessary for
that individual to complete an assigned task. Even if the individual is cleared
for a classification level of information, the individual should not access the
information unless there is a need to know.

Discretionary Access Control

With discretionary access control, the subject has authority, within certain limitations, to specify what objects are accessible. For example, access control lists (ACLs) can be used. An access control list is a list denoting which users have what privileges to a resource. For example, a *tabular listing* would show the subjects or users who have access to the object, e.g., fi le X, and what privileges they have with respect to that fi le.

An *access control triple* consists of the user, program, and fi le, with the corresponding access privileges noted for each user. This type of access control is used in local, dynamic situations in which the subjects must have the discretion to specify what resources certain users are permitted to access. When a user within certain limitations has the right to alter the access control to certain objects, this is termed a *user-directed discretionary access control*.

Nondiscretionary Access Control

A central authority determines which subjects can have access to certain objects based on the organizational security policy. The access controls might be based on the individual's role in the organization (role-based) or the subject's responsibilities and duties (task-based). In an organization with frequent personnel changes, nondiscretionary access control is useful because the access controls are based on the individual's role or title within the organization. Therefore, these access controls don't need to be changed whenever a new person assumes that role.

Access control can also be characterized as *context-dependent* or *content-dependent*.

Context-dependent access control is a function of factors such as location, time of day, and previous access history. It is concerned with the environment or context of the data. In content-dependent access control, access is determined by the information contained in the item being accessed.

6. Autonomic Security

Autonomic computing refers to a self-managing computing model in which computer systems reconfigure themselves in response to changing conditions and are self-healing. The promise of autonomic computing will take a number of years to fully materialize, but it offers capabilities that can improve the security of information systems and cloud computing. The ability of autonomic systems to collect and interpret data and recommend or implement solutions can go a long way toward enhancing security and providing for

recovery from harmful events.

Autonomic Systems

Autonomic systems are based on the human autonomic nervous system, which
is self-managing, monitors change that affect the body, and maintains internal
balances. Therefore, an autonomic computing system has the goal of performing
self-management to maintain correct operations despite perturbations to
the system. Such a system requires sensory inputs, decision-making capability,
and the ability to implement remedial activities to maintain an equilibrium
state of normal operation. Examples of events that would have to be handled Malicious attacks

- Hardware or software faults
- Excessive CPU utilization
- Power failures
- Organizational policies
- Inadvertent operator errors
- Interaction with other systems
- Software updates

IBM introduced the concept of autonomic computing and its eight defining
characteristics5 as follows:

- **Self-awareness** — An autonomic application/system "knows itself" and
is aware of its state and its behaviors.
- **Self-configuring** — An autonomic application/system should be able configure and reconfigure itself under varying and unpredictable conditions.
- **Self-optimizing** — An autonomic application/system should be able to
detect sub-optimal behaviors and optimize itself to improve its execution.
- **Self-healing** — An autonomic application/system should be able to detect
and recover from potential problems and continue to function smoothly.

Self-protecting — An autonomic application/system should be capable
of detecting and protecting its resources from both internal and external
attack and maintaining overall system security and integrity.
Context-aware — An autonomic application/system should be aware of its
execution environment and be able to react to changes in the environment.
Open — An autonomic application/system must function in a heterogeneous
world and should be portable across multiple hardware
and software architectures. Consequently, it must be built on standard
and open protocols and interfaces.

Autonomic Protection

Autonomic self-protection involves detecting a harmful situation and taking
actions that will mitigate the situation. These systems will also be designed
to predict problems from analysis of sensory inputs and initiate corrective
measures.
An autonomous system security response is based on network knowledge,
capabilities of connected resources, information aggregation, the complexity of
the situation, and the impact on affected applications.
The decision-making element of autonomic computing, considering
the current security posture and security context of the system to be protected,
can take actions such as changing the strength of required authentications or
modifying encryption keys. The security context is derived from information
acquired from network and system supervising elements and then collected
into a higher-level representation of the system security status.

An oft overlooked aspect of autonomic systems is that security vulnerabilities
can be introduced by configuration changes and additional autonomous activities
that are intended to address other computational areas.

Autonomous protection systems should, therefore, adhere to the following
guidelines:
- Minimize overhead requirements.
- Be consistent with security policies.
- Optimize security-related parameters.
- Minimize impact on performance.
- Minimize potential for introducing new vulnerabilities.
- Conduct regression analysis and return to previous software versions if
problems are introduced by changes.

7. Cloud computing security challenges:

i. Virtualization security management virtual threats
ii. VM Security Recommendations
iii. VM-Specific Security techniques.

(not found)

PART VI

Enterprise cloud computing

Unit 5 (TEXTBOOK)

The ecosystem of technologies related to the enterprise adoption of cloud computing is constantly evolving. In addition to the three major cloud providers, new ones are emerging from amongst those already in the data center hosting business. Apart from cloud providers, there are also tools to manage combinations of in-house and cloud resources. Similarly, there are frameworks to assist enterprises in creating 'private' clouds within their own data centers. As cloud computing matures, many of the concerns surrounding its use for enterprise applications are likely to be addressed. In the meantime, there are a few quick-wins that can result in immediate benefits by leveraging available cloud platforms. In the longer term, cloud computing will itself evolve in hereto unknown directions, and we speculate on a few of these: In particular, the convergence of public and private clouds, and the emergence of 'cloud-services.'

CHAPTER 17

Enterprise cloud computing ecosystem

So far we have covered a number of cloud computing technologies as well as explored their impact on the software needs of enterprises. In the process we have limited our discussion to the major cloud computing providers, viz. Amazon, Google and Microsoft, with most of our examples taken from the first two, given that Microsoft's offering is still in its nascent stages at the time of writing.

However, the cloud computing ecosystem includes other cloud providers, albeit smaller than the three major ones. Additionally, there are a range of emerging technologies that complement public clouds, enable interoperability between private data centers and public clouds, or facilitate the creation of private clouds within enterprises.

Figure 17.1 depicts our classification of the cloud ecosystem from an enterprise perspective, also indicating the organizations involved in creating and bringing these technologies to market. Needless to say this list of organizations is incomplete and evolving; moreover, given the rate of innovation in the cloud space, it is possible that additional technology categories may emerge in the future. (Note: there are other similar classifications, such as the OpenCrowd taxonomy[1], which includes a far broader range of technologies and applications.)

[1] www.opencrowd.com/views/cloud.php

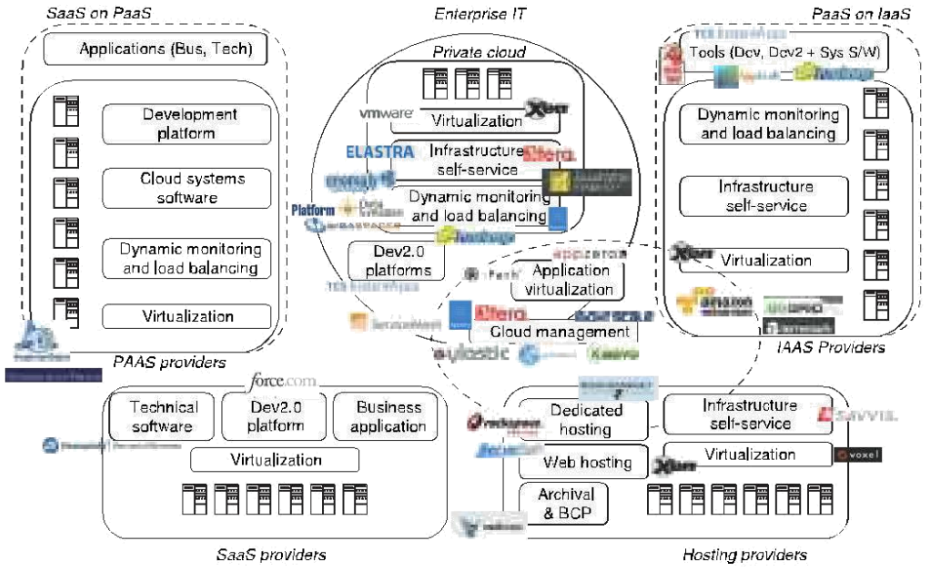


FIGURE 17.1. Enterprise cloud computing ecosystem

The enterprise cloud ecosystem comprises of three main categories; cloud service providers, tools for using and managing cloud deployments and tools for building private clouds. Each of these categories include a number of underlying technologies as illustrated in Figure 17.1 (some are shaded to indicate that they are directly available to enterprise while others are used internally by public cloud providers). We cover each of these categories in the next few sections. We also examine in detail some open source technologies that mimic public cloud implementations, which can therefore be used by enterprises embarking on creating private clouds.

17.1 PUBLIC CLOUD PROVIDERS

Let us recall the key elements of a public cloud platform, such as Amazon EC2, Google App Engine or Microsoft Azure. An infrastructure as a service (IaaS) cloud (such as Amazon EC2) offers self-service infrastructure (both compute and storage) provisioning built on an underlying large-scale data center based on virtualization. Dynamic monitoring and load balancing services are the latest addition to the Amazon cloud, i.e. CloudWatch, Auto Scaling and Elastic Load Balancing. Platform as a service (PaaS) clouds, such

as Google App Engine and Microsoft Azure, offer a software development and deployment platform while hiding much of the underlying virtualization and dynamic resource management layers. The common feature of these offerings is the *ability of users to pay only for the resources they actually consume, at a very fine granularity*. **From a business perspective this is the essence of classifying any offering as a public cloud service.**

Data center hosting services are not new to the IT industry and have been around for decades. The traditional service offering from such providers has been 'dedicated server' hosting, where a set of physical servers and storage were provisioned and dedicated for use by a particular enterprise customer. In recent years, these providers have also begun to offer virtual servers, based initially on process virtual machines such as user-mode Linux and virtual private servers (see Chapter 8), but now almost universally based on native hardware virtualization, typically using the open source Xen VMM. Similarly, web-hosting services have been mainstream for a number of years as well. Many websites today, especially of small and medium enterprises, are hosted and managed by web-hosting providers. These providers have offered hosted storage and database space as part of their web-hosting services, and are now offering additional services based on virtual servers.

Hosting services (dedicated server as well as web) have traditionally charged on a monthly basis depending on the resources, dedicated or virtual, deployed for a user. Many traditional hosting providers are now offering additional features similar to cloud providers, such as self-service provisioning, while continuing to offer their traditional hosting offerings: Rackspace, Savvis, Voxel, ServerVault and ServePath fall into this category, as shown in Figure 17.1. However, only some, such as Terremark and GoGrid (which is operated by ServePath) have also begun to offer true cloud pricing, i.e. charging only for resources that users actually consume on an *hourly* basis. Therefore we have classified GoGrid and Terremark as an IaaS cloud along with Amazon EC2, but not the other hosting providers even though they are also 'cloud providers' in the more general sense of the term. It is likely, however, that many of these will enter the IaaS space in the near future, resulting in a rapid expansion of the number of public IaaS clouds. Still others may evolve towards niche infrastructure services with cloud pricing models, such as Vaultscape for storage backup.

The platform as a service (PaaS) space, as compared to IaaS, is still sparse, with very few entrants apart from Google and Microsoft. An example of a niche PaaS offering is Engine Yard, which provides a development platform using Ruby on Rails, based on underlying infrastructure running on Amazon

EC2; i.e. an example of 'PaaS over IaaS.' In a similar vein AppScale is an open source implementation of the Google App Engine interface on Amazon EC2. AppScale also leverages the Hadoop project's open source versions of Google's BigTable and GFS (covered in Chapter 10). We describe AppScale in more detail in Section 17.3.2 below.

Finally, recall our extensive treatment of the Dev 2.0 paradigm (end-user-driven application development tools) exemplified by Salesforce.com's Force.com platform, TCS's InstantApps, as well many others as mentioned in Chapter 12. We classify most Dev 2.0 platforms, including Force.com, in the software as a service category, since they limit application features as compared to a PaaS platform where access to a full programming language is provided. InstantApps on Amazon EC2, however, can be considered to be a PaaS (on IaaS) offering, since this tool allows users to write native client and server-side code (in JavaScript as well as Java), thereby in principle allowing any web application feature to be developed on it, just as for a full PaaS platform such as Google App Engine or Azure. (Additionally, InstantApps is also deployable on-premise as a traditional software development tool.)

17.2 CLOUD MANAGEMENT PLATFORMS AND TOOLS

Configuring and managing a small set of servers on an IaaS cloud can be accomplished easily using an IaaS offering's own infrastructure self-service APIs and tools. Managing a larger and more complex deployment requires more tools support and automation, just as it does within an enterprise data center. Cloud management platforms such as 3tera, RightScale, Kaavo, EnStratus and Ylastic provide web-based graphical tools to configure and manage complex configurations of servers deployed in the cloud. Some of these tools work only with Amazon EC2, while others, such as RightScale, enable management of multi-cloud deployments; for example, spanning Amazon EC2 and GoGrid. Further, all these cloud management platforms are themselves deployed in the cloud, either on an IaaS platform such as Amazon EC2 (e.g. RightScale) or in partnership with smaller, hosting providers (e.g. 3tera).

In addition to graphical self-service infrastructure management, some cloud management tools also offer dynamic monitoring and load balancing. These capabilities were crucial in the initial stages of IaaS before Amazon EC2 itself introduced Elastic Load Balancing, CloudWatch and Auto Scaling. Though no longer required if one is only using Amazon EC2, in a multi-cloud scenario they may become increasingly important when the number of IaaS clouds grows.

From an enterprise IT perspective though, at least for the foreseeable future private data centers (including those managed by dedicated hosting providers) will remain part and parcel of enterprise IT. Cloud deployments will only complement this infrastructure. Therefore, the ability to manage complex IT environments that span local infrastructure as well as cloud deployments will be required. Traditional infrastructure management tools, such as from CA or IBM, will certainly respond to this need. In the meanwhile, some cloud management tools, such as 3tera, Appistry or ServiceMesh can also be deployed within the enterprise and used to manage a combination of local and public cloud infrastructure.

Recall how application software is made available on an IaaS cloud: Amazon EC2 packages a virtual server along with its software configuration as an AMI; GoGrid has its own similar format called GSI. However, creating such virtual images from scratch, say from a deployment of an enterprise application, is not simple, further the images so created are large. Niche, 'application virtualization' technology from rPath or AppZero makes such tasks simpler and also optimizes the application images in size. Such tools also promise to enable portability of application images across different clouds and private data centers.

Finally, there is another emerging trend related to infrastructure management: Hosted technical software for email security and virus scanning, such as MessageLabs (recently acquired by Symantec). We foresee that even more technical software, such as middleware for managing web services, workflows, or identity, moving to hosted models. Leading in this arena are the .NET services included in Microsoft's cloud platform, as we mentioned briefly in Chapter 5.

17.3 Tools for building private clouds

Automation of data center operations within the enterprise using technologies such as virtualization promises to improve efficiencies by increasing server utilization, as we have seen in Chapter 8. Self-service infrastructure provisioning together with dynamic monitoring and load balancing are equally critical elements in achieving such higher utilization levels as well as in reducing manpower requirements and costs of infrastructure management.

There is considerable interest amongst enterprises, especially within corporate IT, in creating 'private clouds.' For the most part, such private cloud projects involve a combination of virtualization, self-service infrastructure and dynamic monitoring and load balancing. Many are re-branded versions

of already on-going data center automation projects. Others have been driven by the emergence of public clouds as an attractive option for enterprise application deployments, motivating efforts to achieve similar efficiencies within the enterprise.

Tools for self-service infrastructure as well as some dynamic resource control capabilities are now being provided by products from virtualization tool vendors such as VMware. Cloud management tools that are deployed within the enterprise such as 3tera and Appistry (which are also available in the cloud), as well as others such as Elastra and Enomaly also provide such features. In high-performance computing environments, such as trading floors of investment banks or for scientific computing, Grid computing technologies such as from GigaSpaces, Platform Computing and DataSynapse (now acquired by TIBCO) offer similar features tuned for applications exhibiting large real-time demand fluctuations.

The question remains as to what extent data center automation using such technologies constitutes a 'private cloud,' and if so whether it can achieve efficiencies nearing those of public clouds. We shall return to this question in the next chapter. In the meantime, we examine the design of two open source projects that implement systems resembling Amazon EC2's IaaS platform and Google App Engine. One of these, Eucalyptus, is already available as a commercial open source offering: While data center automation technology aims to improve the efficiency of the entire data center, including a variety of platforms, legacy systems and hardware configurations, these open source projects (as well as some cloud management tools such as Enomaly) can be used to create private-cloud-like islands within enterprise data centers, but without promising to include all legacy platforms in their ambit. Deploying applications on these platforms involves similar steps as deploying on a public cloud, only that the hardware resources may be located in-house. It is also possible that hosting providers may leverage such platforms in the future to offer cloud-like services that mimic the larger cloud platforms, and in the process also offering interoperability and competing on price, location and personalized service. For this reason both these platforms, Eucalyptus and AppScale, merit a more detailed study.

17.3.1 IaaS using Eucalyptus

Eucalyptus [41] is an open source framework (developed at the University of California, Santa Barbara) that implements infrastructure as a service (IaaS) on a collection of server clusters. Eucalyptus Systems is a commercial offering

based on this open source project, targeted at enterprises interested in building private clouds. The design of Eucalyptus also provides insights into the issues that need to be handled while creating an IaaS cloud, and serves as a platform for research in this emerging area. Since Eucalyptus implements external APIs identical to Amazon EC2, it also provides clues as to the possible internal architectures of such public clouds. For the same reason, Eucalyptus deployments can also be controlled by cloud management tools, such as RightScale (which in fact offers management of the experimental Eucalyptus cloud at UCSB).

Figure 17.2 illustrates the Eucalyptus architecture. Eucalyptus can run on a collection of one or more server clusters. Servers within each cluster are connected via a fast local Ethernet, while clusters can be connected to each other via possibly slower networks, such as private wide area networks or even the internet. Each server node runs the Xen hypervisor on which user virtual machines are provisioned on demand. Each cluster node also runs a Eucalyptus Instance Manager (IM) process on the XenLinux host operating system (provisioned automatically when Xen boots). One node in each cluster

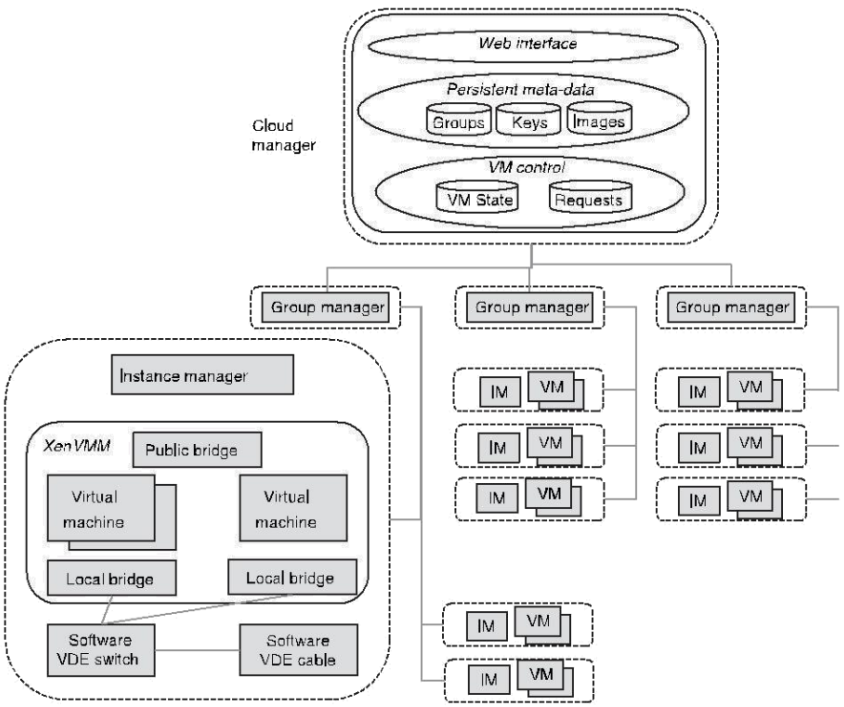


FIGURE 17.2. Eucalyptus IaaS framework

runs a Eucalyptus group manager (GM) process, and a single cloud manager (CM) process server is responsible for overall control of the Eucalyptus cloud, i.e. all clusters. The group managers and cloud manager run natively on specially designated servers rather than as virtual machines.

The cloud manager provides a web interface, including REST APIs identical to that of Amazon EC2, using which users can request for virtual servers and monitor their status. The cloud manager also maintains persistent meta-data describing the cloud, such as groups, their addresses, access keys given to users to connect to their assigned servers, as well as virtual machine images using which user virtual servers ma be provisioned. The cloud manager is responsible for controlling the entire cloud, and therefore also needs to track the dynamic state of all virtual machines in the cloud, the load, performance and utilizations of each cluster, as well as the status of all user requests.

Each instance manager on a cluster node is responsible for provisioning virtual machines as requested by its group manager, as well as communicating the status of the node, i.e. resource utilization and available capacity, to its group manager on request. Each group manager in turn monitors the status of its entire cluster and communicates that back to the cloud manager on request. The group manager also makes intra-cluster scheduling decisions in response to the requests it receives from the cloud manager.

Eucalyptus allows virtual servers allocated to a single cloud user (a virtual cluster) to span across groups; this feature ensures scalability as well as enable the creation of virtual clusters where some servers reside within an enterprise and others on an external cloud (akin to Amazon's virtual private cloud). Virtual machine instances in such virtual clusters need to be connected to each other, but must not be able to access any VMs belonging to another user. Vanilla virtualization does not cater to such isolation needs, i.e. a Xen VM that has access to the physical network of its server can in principle send packets to any server on that network. Therefore, to implement network isolation between users as well as provide transparent network connectivity within a user's virtual cluster, even if it spans across groups, Eucalyptus implements a virtual distributed Ethernet (VDE) [2]. Such a VDE is implemented using software switches on each server and software 'cables' between servers that hide the actual network topology, effectively providing a VLAN for each virtual cluster. Routing intra virtual cluster communication between groups via the VDE is therefore an additional responsibility of

[2] http://vde.sourceforge.net

Eucalyptus group managers. Finally, any virtual machine can additionally be connected to the 'public' network (via the local Ethernet), and publish a public IP address (via network address translation). Note that the 'public' network in this context could be the internet, a corporate wide area network or even a local LAN. At least one VM in a virtual cluster needs to be connected in this manner so as to enable users to access the virtual cluster from the public network.

17.3.2 PaaS on IaaS: AppScale

Recall (from Chapter 5) that the Google App Engine platform provides developers with an open source Python-based web-server (`dev_appserver.py`) that allows deployment of GAE applications on desktops, using which applications can be tested and debugged while being developed before they are uploaded to the Google PaaS cloud. The AppScale [11] open source project (also developed at the University of California, Santa Barbara) mimics the GAE platform through distributed deployment of the GAE development web-server on a cluster of virtual machines. Using AppScale, a GAE-like PaaS environment can be implemented in a scalable manner on an IaaS platform, such as EC2 or Eucalyptus.

Figure 17.3 illustrates the AppScale architecture. Instances of the GAE development web-server, `dev_appserver.py` are deployed on multiple virtual machines as an AppScale application server (AS) component. Since the datastore included with `dev_appserver.py` is a local file-based implementation that emulates the actual Google datastore APIs, this component is modified in AppScale so as to connect to a Hadoop HBase+HDFS deployment. (Recall that HBase and HDFS are open source implementations of Google's

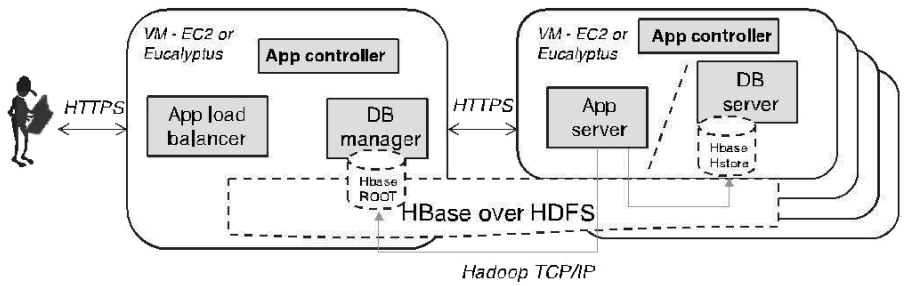


FIGURE 17.3. AppScale PaaS on IaaS architecture

BigTable and GFS architectures.) Therefore, AppScale also includes a collection of database server (DBS) components on a number of virtual machines on which HBase and HDFS run (these virtual machines can be the same ones on which AS components are deployed). Each AppScale node also runs an application controller (AC) that controls the AppScale components deployed on it.

An AppScale cluster includes one head node that implements a database manager (DBM) component that functions as the master node for the HBase and HDFS deployment. When an AS node needs to access the datastore, they communicate (via HTTPS) with the DBM to retrieve HBase meta-data following which it can communicate with the files storing HBase tables directly. A user interacts with the AC on the AppScale head node upload application code, which is also stored in the distributed AppScale datastore. Through this AC users also configure a set of virtual machines running AS servers; these VMs are dedicated to serve that user. The head node also includes an application load balancer (ALB) component which is responsible for directing incoming HTTP requests to one of the AS nodes assigned to a user.

It is important to note that the user's view of AppScale differs from GAE in an important respect: In GAE users are oblivious of which web-servers are used to serve their applications, how many servers are running, and how many different applications (belonging to different users) are served by each web-server. On the other hand, while AppScale enables execution of GAE applications, the level of control given to users as regards the VMs dedicated to serve their applications more closely resembles Microsoft Azure. For example, by combining AppScale and Amazon's dynamic monitoring and load balancing tools, it should be possible, in principle, to allow the set of virtual machines actually serving a user's application to vary automatically within pre-defined limits, just as in Azure.

CHAPTER 18

Roadmap for enterprise cloud computing

We are nearing the conclusion of our exploration of cloud computing; along the way we have also covered many aspects of enterprise architecture. Certain aspects of enterprise architecture may need to adapt to or be replaced by emerging paradigms, such as PaaS or Dev 2.0. At the same time many enterprise applications can, technically, be easily deployed in an IaaS cloud. We have also examined the question of cloud economics and shown that in principle public cloud can be cheaper, due to elasticity, and also faster to use, due to automated provisioning. We have also noted the additional cost advantages of PaaS platforms that enable public facing applications to be made available at no cost, with charges accruing only once load increases.

So, where could, and should, an enterprise begin leveraging the cloud? There are valid concerns with using public clouds from an enterprise perspective, especially with respect to (a) data confidentiality, lock-in, and auditability as well as (b) software licensing. Enterprises are still wary of placing production or sensitive data in the cloud, since current cloud offerings are essentially public networks and hence exposed to more attacks. While there are no fundamental obstacles to making a cloud environment as secure as an in-house data center, this requires careful planning using encrypted storage, virtual LANs and network middleware. Some cloud providers have begun to take the first steps towards these levels of security, such as Amazon's VPC.

When it comes to software licensing, while a few software vendors have begun making tools available in the cloud as bundled cloud-based AMIs, most have yet to transition to a usage-based model. High up-front software licensing costs can obviate any advantages of usage-based pricing at the infrastructure level, and to that extent limits the number of enterprise applications that can advantageously leverage IaaS public clouds.

Finally, when it comes to PaaS or Dev 2.0 platforms, these usually involve application migration or fresh development, primarily because of the novel development paradigms involved or non-relational data models. To what extent this will change with Microsoft Azure's support for standard relational SQL remains to be seen.

18.1 QUICK WINS USING PUBLIC CLOUDS

While keeping the above concerns in mind, we believe that the following areas, some of which have been briefly mentioned earlier in Section 4.4.3, represent opportunities for leveraging public clouds in the near term, without compromising on security or data risk. Further, each of these use cases specifically exploits the *elasticity* properties of public clouds.

18.1.1 Development and testing

The infrastructure needs for developing and testing enterprise applications are different from those of a production environment, for example the requirements regarding data security are lower. At the same time, variability and volatility is high, with servers being required for each new project, many of which can be released once the application is rolled out. Further, the time for provisioning and configuring a development environment can often become a significant overhead in many large organizations due to procurement and infrastructure management procedures. Leveraging cloud services for development-and-testing servers is therefore a cost-effective and low-risk option, which can also improve business agility in terms of how rapidly new applications can be developed.

Performance testing of new applications on a production capacity hardware configuration is difficult, especially early in the development cycle, simply because of non-availability of such an environment. Using the cloud a production-class infrastructure can be provisioned on demand and disbanded once the performance testing is complete.

18.1.2 Analytics in the cloud

We have already discussed the MapReduce-based cloud programming paradigm that enables massively parallel computations while automatically compensating for inevitable hardware and software failures. Such analytical tasks need the cloud and would be next to impossible in traditional data centers. On the other hand, normal enterprise analytics may not share such scale, but can benefit greatly from elasticity. Often enterprises need to run regular analytics on customers, supply chains or manufacturing operations, say on a daily basis. Such jobs may run for a few hours on dedicated hardware, and occasionally require even larger capacity, thus leading to over provisioning of infrastructure. Using the cloud, the required infrastructure can be provisioned when needed and disbanded thereafter. (Note that especially in the case of analytics, large volumes of data may be involved; it is important to recognize that one can circumvent this constraint by physically shipping data and transferring only small volumes over the network.)

18.1.3 Disaster planning in the cloud

Maintaining a disaster-recovery site that can be rapidly brought into production when needed to ensure business continuity requires replicating hardware infrastructure at least partially, which in normal circumstances may remain unutilized. Instead, it is possible to store a virtual image of the production environment in the cloud so that actual backup servers can be provisioned only when required. Similarly production data backups can be physically shipped to a location near the cloud provider on a regular basis and loaded into the cloud only when needed. Alternatively, updates can be replicated regularly over the network and exported to disk remotely rather than locally. Such cloud-based disaster-recovery mechanisms can be orders of magnitude cheaper than replicating infrastructure, while offering similar levels of protection and business continuity.

18.1.4 Low/Variable volume 24×7 portals

As a common scenario in the case of small or medium enterprises, consider a web-based portal or application that needs to be made available 24×7, but it is not clear how much traffic will flow to this site. Using a PaaS platform such as GAE such an application can be deployed without incurring *any* running

costs, while also ensuring that the site will scale automatically if load increases. In fact, this model can be combined with IaaS processing, by using the PaaS platform to queue requests that are actually processed by more traditional applications that run on an IaaS cloud. Virtual resources on the IaaS cloud can be provisioned *on demand* when queues build up.

18.1.5 Enterprise mashup portals

A number of useful applications and data are available on the web in the form of 'mashups,' such as the Google Map mashup, that run as JavaScript code within the browser on a user's desktop. Allowing such mashups to be deployed within the user interface of an enterprise application is a potential security risk, since it requires the user's browser to allow 'cross-site scripting,' thereby allowing the browser to *simultaneously* connect to a server on the internet as well as the application server on the corporate network.

Instead, those pieces (i.e., pages) of enterprise applications that include public mashup applications can be hosted on servers deployed in a public cloud. Access to application services within the corporate network can be re-directed through secure web services instead of direct access from the user's browser, which is likely to be safer due to the additional layer of security introduced within the cloud-based server.

Thus, just as web-server technology was first used to create enterprise 'portal' architectures so that users could experience a single entry point to different enterprise applications, cloud platforms can play a similar role by integrating publicly available mashups with enterprise applications at the user interface level.

18.1.6 Mobile enterprise applications

Users now expect access to enterprise applications from mobile devices. Providing a rich mobile experience requires a return to 'fatter' client applications, as well as supporting disconnected operation via intelligent asynchronous data replication. Moreover, the fact that mobile devices are *personal*, rather than enterprise owned and controlled, introduces the need for an added layer of security. Cloud-based applications serving mobile clients could potentially provide such a secure intermediate layer, in a manner similar to that described

above for mashups: Mobile clients could connect to specific subsets of application functionality deployed in cloud-based servers. Support for asynchronous data replication as well as secure access to web services published by applications within the corporate network would be provided within the cloud-based server.

18.1.7 Situational applications using Dev 2.0

As we have seen in Chapters 12 and 14, it is possible to achieve order-of-magnitude improvements in software development productivity using Dev 2.0 platforms, such as Force.com or TCS InstantApps. If one were to take an inventory of all applications in a large enterprise, we would typically find a small number of complex, highly loaded, mission-critical applications, a moderate number of departmental applications of medium complexity and usage, and finally a large 'long tail' of small, lightly loaded, 'situational' applications. Examples includes custom workflow and mashup applications assembled to drive simple internal automation or pilot new business processes within individual business units. Dev 2.0 platforms deployed in public clouds are ideal for situational applications, since business units can rapidly provision, configure, use and then discard such applications.

18.2 Future of enterprise cloud computing

Moving beyond the immediate, let us now consider what technological and industry trends are likely to drive the cloud computing ecosystem in the future. In the process we may unearth some clues as to how cloud computing may eventually come to impact enterprise IT.

As has been well elucidated in the popular book *The Big Switch* [8], the evolution of industrial use of electricity from private generating plants to a public electricity grid can serve as an illuminating analogy for the possible evolution of enterprise IT and cloud computing. In such an analogy, privately run enterprise data centers are analogous to private electric plants whereas the public electricity grid illustrates a possible model towards which the public clouds of today may evolve.

As another analogy, let us consider data communications: In the initial days of digital networks, corporations owned their own data communication

lines. Today all data communication lines are owned by operators who lease them out, not only to end-users, but also to each other. The physical resource (bandwidth) has become a commodity, and it is only in the mix of value added services where higher profits are to be made.

18.2.1 Commoditization of the data center

We are already seeing trends towards commoditization of computation and storage hardware. It is precisely by utilizing commodity hardware efficiently that the large cloud providers have been able to achieve their operational efficiencies of scale. The next stage of evolution is for larger collections of hardware to become standardized, starting with racks of servers, and eventually the data center itself. We are already seeing evidence of this in dedicated hosting providers who are now striving to move 'up' the value chain into cloud computing, as their core business comes under margin pressure. Eventually, it is possible that the highly virtualized, power efficient data center, offering on-demand resource provisioning, also becomes a commodity product, much like servers and storage today, as illustrated in Figure 18.1.

Apart from the natural process of standardization and commoditization, there are additional drivers leading data center commoditization: An

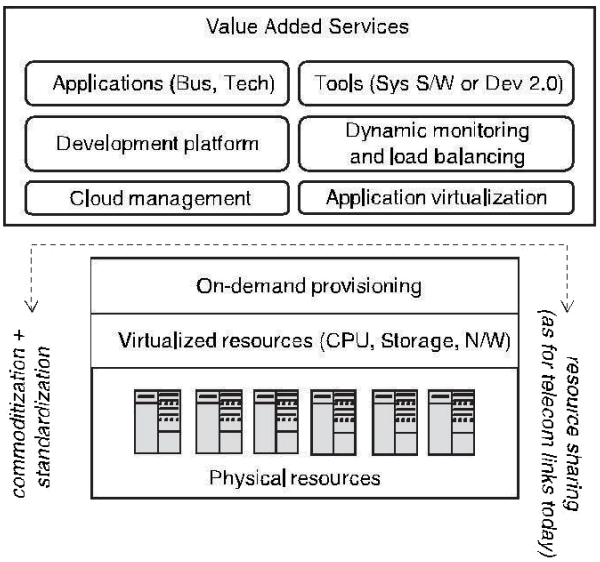


FIGURE 18.1. Commoditization of the data center

important concern for enterprise adoption of cloud computing is the physical location of data centers on which clouds operate. In certain industries, such as financial services and health-care, many governments regulate the location of physical storage of data to be within a specified geographical area, such as a country or continent. As a result, cloud providers such as Amazon maintain data centers in different geographical regions, and allow users to choose which 'regions' their virtual resources and data will be provisioned in. At the same time, a large number of enterprises already rely on managed hosting providers instead of housing their servers in-house, and as we have seen, many of these are already beginning to exploit virtualization and automated provisioning technologies. These data centers also need to be physically located so as to conform to the same regulations. The resulting possibility for evolution of the cloud ecosystem is outlined below:

18.2.2 Inter-operating Virtualized Data Centers

So far, data centers managed by different providers operate in isolation in the sense that while end-users can often provision resources on-demand in these facilities, trading of capacity between such facilities, such as is the case for electricity or even data communications bandwidth, does not take place.

The key technology elements that could enable on-demand exchange of capacity are already in place: It is possible to programmatically provision and access resources using web services, as demonstrated by IaaS providers and frameworks such as Eucalyptus. What is missing is standardization of such APIs so that business models and trading can be based on a non-proprietary mechanism of making requests and ensuring service levels. However, we believe this is not far away. What could become possible if and when such standardization does happen?

Recall that the larger data centers (such as maintained by cloud providers) are almost always located near cheap power sources, thereby significantly lowering their running costs. Now, we speculate whether a natural evolution of such an ecosystem might not see data center owners, be they providers of dedicated hosting or cloud computing, begin leasing data center resources to *each other* and not only to end-users. From an enterprise perspective this could, for example, enable a managed hosting provider to ensure that a customer's applications are run on a mix of servers, some physically nearby, while others are leased from a larger-scale provider who reaps economies of scale, while also ensuring that any geographical constraints on data storage are maintained.

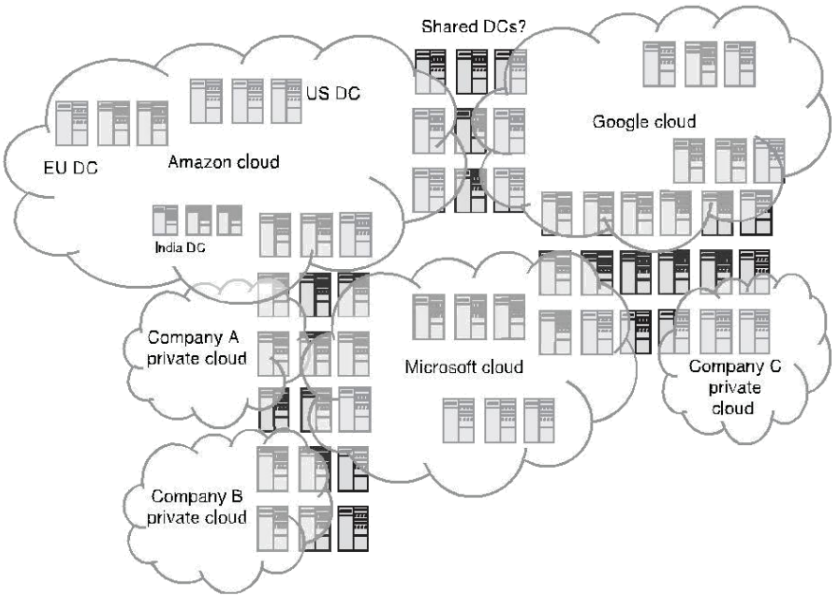


FIGURE 18.2. Future of enterprise cloud computing?

Further, with standardization of the virtualized data center and capacity exchange APIs, it should also become possible for different cloud providers to either co-locate or cross-lease their data centers, so that, for example a portal running on Google App Engine can communicate with a database running on an Amazon EC2 virtual server without having to traverse the public internet: In other words, while these services are managed by different providers, the servers they happen to use for a particular user reside in data centers that are 'near' each other from a network perspective, i.e. on a high-speed LAN rather than a WAN. Speculating even further, even some of the servers that an enterprise considers 'in-house' may also actually reside in the same or 'nearby' data centers, as part of a managed hosting service. Finally, of course, users connect to applications using VPNs over the internet, or through dedicated leased lines for better performance. Such a scenario is illustrated in Figure 18.2.

18.2.3 Convergence of private and public clouds

As we have mentioned earlier in Chapter 6, all but the largest enterprises are unlikely to enjoy economies of scale similar to public clouds, and those who do may end up becoming public cloud providers themselves in the future.

Further, the virtualized data center with on-demand provisioning may end up being a commodity that one can buy off the shelf, much like a basic server today. As we have discussed above, it may become the case that enterprises will use servers in managed data centers, be they based on dedicated hosting or clouds, with many of these servers naturally being co-located or at least 'near' each other from a network perspective. Moreover, whether or not servers are dedicated to an enterprise, or are virtual servers on shared resources, they can all be connected on the same virtual private network so that from both a performance as well as network address space perspective, they are essentially indistinguishable.

What this means is that the distinction between public and private clouds becomes blurred, just as it is today for communications: Users are essentially oblivious as to exactly which physical cables their data traffic travels on, even on an *internal* corporate WAN, and whether or not at the lowest levels it is multiplexed with traffic from other users.

So, what technologies should an enterprise focus on when exploring how their data centers will be managed? Virtualization and on-demand provisioning, we believe, will become available off the shelf. The areas to focus on when exploring private clouds are the higher layers of Figure 18.1, i.e. cloud management, dynamic load balancing, application virtualization and software tools. These are more complex features, where there is likely to be more competition and value addition, and where standardization is unlikely, at least in the near future. At the same time, it is also precisely these areas where the cloud ecosystem will provide many alternative solutions to choose from. Finally, the cloud ecosystem itself will become far more usable once such services deployed 'in the cloud' *appear*, for all practical purposes, to be 'inside' the enterprise, from either a network address or performance perspective.

18.2.4 Generalized 'cloud' services

As we conclude our exploration of cloud computing, it is natural to ask whether the efficiencies promised by the cloud computing model can, in any way, be generalized to other arenas of knowledge-based work. Recall the key elements of cloud computing, as we outlined at the outset of our journey in Section 1.1:

- Computing resources packaged as a **commodity** and made available over the internet.
- **Rapid provisioning** of resources by end-users.

- A **usage-based pricing** model that charges consumers only for those cloud resources they actually use.

In Chapter 12 and 14 we introduced and described Dev 2.0 platforms. In a sense, Dev 2.0 platforms offer similar 'cloud-like' features, but in the domain of software development:

- Application functionality packaged as re-usable templates; a **commodity** of sorts.
- **Rapid provisioning** of new applications using multi-tenant platforms.
- **Usage-based pricing**, on the basis of the number of users, transaction volume or application complexity.

Can such 'cloud-like' features be achieved in other arenas? We consider here the case of *services*, such as those often outsourced to a software development provider, a call-center or an insurance claims processor; more generally, any other knowledge-based task that can be outsourced using information technology.

The traditional outsourcing model has been that of 'time-and-materials' (T&M) billing. Using cloud vs. in-house resources as an analogy, T&M is the equivalent of deploying an 'in-house' data center, where the onus of making efficient use of the resources deployed lies entirely with the customer rather than the provider.

Another outsourcing model is that of fixed-price projects, where detailed specifications are drawn up by the customer and executed in by the services provider according to a pre-determined price and schedule. A fixed-price project is analogous to a data center managed by a dedicated infrastructure provider at a fixed price while adhering to stringent SLAs[1]. In the fixed-price model, all the risk lies with the supplier, be it a data-center provider or services contractor. Naturally, providers of fixed-price projects account for their risks within their pricing models. Further, the onus of providing detailed specifications or defining service levels falls on the customer. Thus, even the fixed-price model has inefficiencies built into it.

Is there something better, i.e., a model for outsourcing services that exhibits some of the 'cloud-like' features described above? For example, how can we define services as a composition of commodity tasks? How can end-users easily request for and 'provision' services once having broken up their project into such tasks? And finally, how can the price of such tasks be objectively

[1] Service-level agreements.

estimated, as unambiguously as, say, the price per cpu-hour in the domain of cloud-based resources?

Hopefully such questions can and will be answered. Perhaps the model-based abstractions used to architect Dev 2.0 platforms may also have a role to play in the process. Only time will tell how far and how rapidly the transition to 'cloud-like' infrastructure, development tools and services will take place in the complex world of enterprise IT.