Subject Name: Cloud Computing (CS802[B])

UNIT I

Introduction to Service Oriented Architecture, Web Services, Basic Web Services Architecture, Introduction to SOAP, WSDL and UDDI; RESTful Services: Definition, Characteristics, Components, Types; Software as a Service, Plat form as a Service, Organizational Scenarios of Clouds, Administering & Monitoring Cloud Services, Benefits and Limitations, Study of a Hypervisor.

INTRODUCTION: CLOUD COMPUTING

Cloud computing is the use of various services, such as software development platforms, servers, storage and software, over the internet, often referred to as the "cloud."

In general, there are three cloud computing characteristics that are common among all cloud-computing vendors:

1. The back-end of the application (especially hardware) is completely managed by a cloud vendor.
2. A user only pays for services used (memory, processing time and bandwidth, etc.).
3. Services are scalable

Many cloud computing advancements are closely related to virtualization. The ability to pay on demand and scale quickly is largely a result of cloud computing vendors being able to pool resources that may be divided among multiple clients. It is common to categorize cloud computing services as infrastructure as a service (IaaS), platform as a service (PaaS) or software as a service (SaaS).

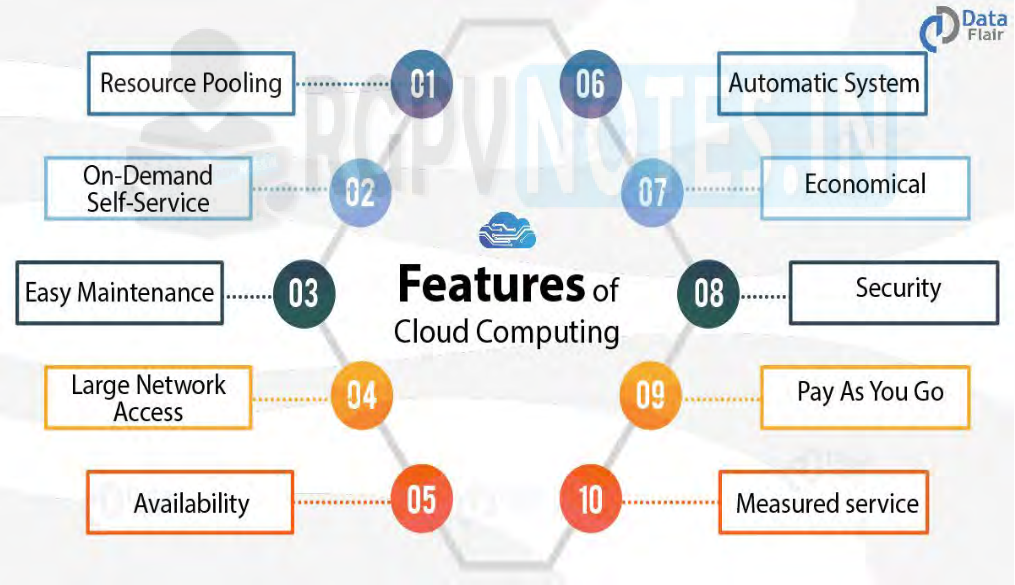

Figure 1.1: Characteristics of Cloud Computing

CHARACTERISTICS

Following are the characteristics of Cloud Computing:

1. Resources Pooling: It means that the Cloud provider pulled the computing resources to provide services to multiple customers with the help of a multi-tenant model. There are different physical and virtual resources assigned and reassigned which depends on the demand of the customer. The customer generally has no control or information over the location of the provided resources but is able to specify location at a higher level of abstraction.
2. On-Demand Self-Service: It is one of the important and valuable features of Cloud Computing as the user can continuously monitor the server uptime, capabilities, and allotted network storage. With this feature, the user can also monitor the computing capabilities.
3. Easy Maintenance: The servers are easily maintained and the downtime is very low and even in some cases, there is no downtime. Cloud Computing comes up with an update every time by gradually making it better. The updates are more compatible with the devices and perform

faster than older ones along with the bugs which are fixed.

4. Large Network Access: The user can access the data of the cloud or upload the data to the cloud from anywhere just with the help of a device and an internet connection. These capabilities are available all over the network and accessed with the help of internet.
5. Availability: The capabilities of the Cloud can be modified as per the use and can be extended a lot. It analyzes the storage usage and allows the user to buy extra Cloud storage if needed for a very small amount.
6. Automatic System: Cloud computing automatically analyzes the data needed and supports a metering capability at some level of services. We can monitor, control, and report the usage. It will provide transparency for the host as well as the customer.
7. Economical: It is the one-time investment as the company (host) has to buy the storage and a small part of it can be provided to the many companies which save the host from monthly or yearly costs. Only the amount which is spent is on the basic maintenance and a few more expenses which are very less.
8. Security: Cloud Security, is one of the best features of cloud computing. It creates a snapshot of the data stored so that the data may not get lost even if one of the servers gets damaged. The data is stored within the storage devices, which cannot be hacked and utilized by any other person. The storage service is quick and reliable.
9. Pay as you go: In cloud computing, the user has to pay only for the service or the space they have utilized. There is no hidden or extra charge which is to be paid. The service is economical and most of the time some space is allotted for free.
10. Measured Service: Cloud computing resources used to monitor and the company uses it for recording. This resource utilization is analyzed by supporting charge-per-use capabilities. This means that the resource usages which can be either virtual server instances that are running in the cloud are getting monitored measured and reported by the service provider. The model pay as you go is variable based on actual consumption of the manufacturing organization.

Service Oriented Architecture

Service-Oriented Architecture (SOA) is a style of software design where services are provided to the other components by application components, through a communication protocol over a network. Its principles are independent of vendors and other technologies. In service-oriented architecture, a number of services communicate with each other, in one of two ways: through passing data or through two or more services coordinating an activity. This is just one definition of Service-Oriented Architecture.

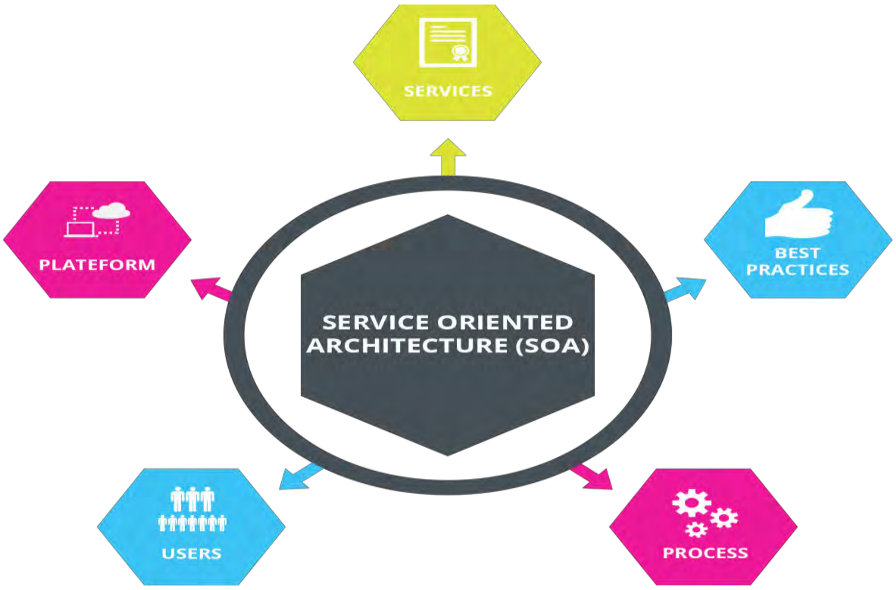

Figure 1.2: Service Oriented Architecture in Cloud Computing

1. Services - The services are the logical entities defined by one or more published interfaces.
2. Service Provider - It is a software entity that implements a service specification.
3. Service Consumer - It can be called as a requestor or client that calls a service provider. A

service consumer can be another service or an end-user application.

4. Service Locator - It is a service provider that acts as a registry. It is responsible for examining service provider interfaces and service locations.
5. Service Broker - It is a service provider that passes service requests to one or more additional service providers.

WEB SERVICES

Web service is a standardized medium to propagate communication between the client and server applications on the WWW (World Wide Web). A web service is a software module that is designed to perform a certain set of tasks.

1. Web services in cloud computing can be searched for over the network and can also be invoked accordingly.
2. When invoked, the web service would be able to provide the functionality to the client, which invokes that web service.

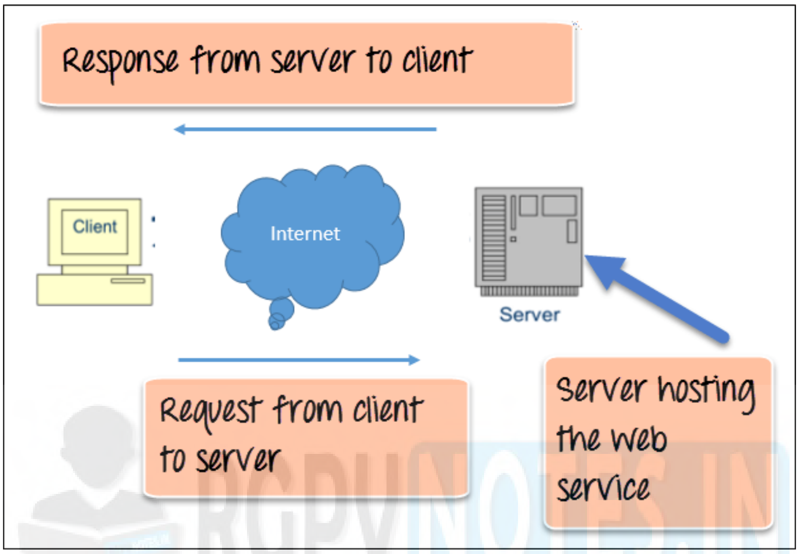
Response from server to client
Client
Internet
Server
Request from client to server
Server hosting the web service

Figure 1.3: Web Services

BASIC WEB SERVICE ARCHITECTURE

Web Services are a mode of communication between the client and the server applications on the World Wide Web.

The architecture of web service consists of three roles: service provider, service requester, and service registry. The interaction consists of three operations: publish, find, and bind. These operations and roles act upon the artifacts of web services. The web service artifacts are the software modules of web service and their description.

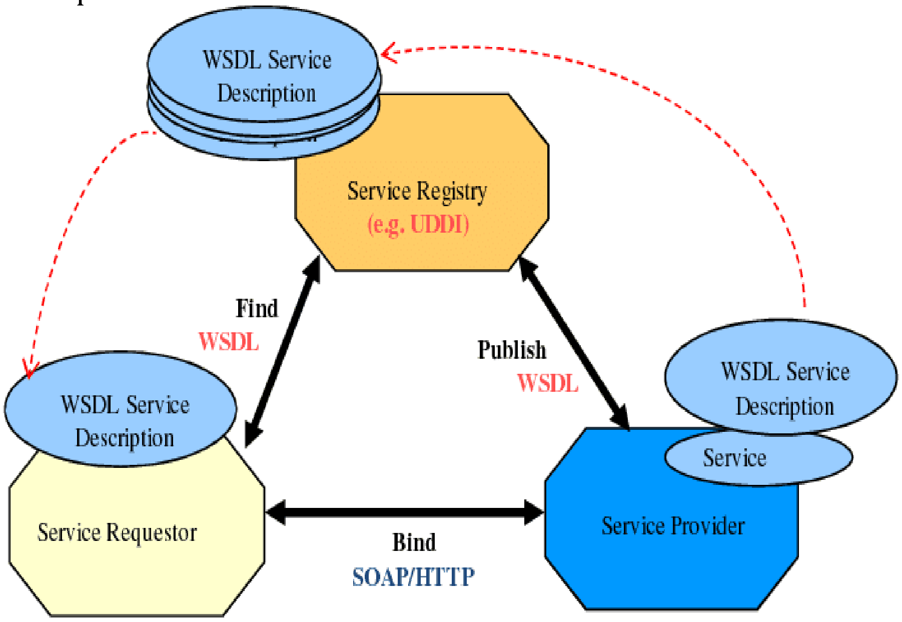

Figure 1.4: Web Services Architecture

The service provider also has a network-associable module that defines a service description for the web service and then publishes it to the service requestor or service registry. The service requestor uses the find operation to retrieve the service description either locally or from the service registry. It also uses the service description to bind with the service provider and invoke the web service implementation.

There are three roles of web service architecture:

1. Service Provider: It is the platform that hosts the services. It creates web service and makes it available to client applications who want to use it.
2. Service Requestor: It is the application that is looking for and invoking or initiating an interaction with a service. Here, the browser plays the requester role, driven by a consumer or a program without a user interface. In short, the client application that needs to contact a web service is Service Requestor. The client application can be a .Net application, a Java application, or any other language-based application that looks for some sort of functionality.
3. Service Registry: Service requestors find the service and obtain binding information for services during development. It is the application that provides access to the UDDI. The UDDI enables the client application to locate the web service.

WEB SERVICES - COMPONENTS

XML-RPC: This is the simplest XML-based protocol for exchanging information between computers.

1. XML-RPC is a simple protocol that uses XML messages to perform RPCs.
2. Requests are encoded in XML and sent via HTTP POST.
3. XML responses are embedded in the body of the HTTP response.
4. XML-RPC is platform-independent.
5. XML-RPC allows diverse applications to communicate.
6. A Java client can speak XML-RPC to a Perl server.
7. XML-RPC is the easiest way to get started with web services.

SOAP: SOAP is an XML-based protocol for exchanging information between computers.

1. SOAP is a communication protocol.
2. SOAP is for communication between applications.
3. SOAP is a format for sending messages.
4. SOAP is designed to communicate via Internet.
5. SOAP is platform independent.
6. SOAP is language independent.
7. SOAP is simple and extensible.
8. SOAP allows you to get around firewalls.
9. SOAP will be developed as a W3C standard.

WSDL: WSDL is an XML-based language for describing web services and how to access them.

1. WSDL stands for Web Services Description Language.
2. WSDL was developed jointly by Microsoft and IBM.
3. WSDL is an XML based protocol for information exchange in decentralized and distributed environments.
4. WSDL is the standard format for describing a web service.
5. WSDL definition describes how to access a web service and what operations it will perform.
6. WSDL is a language for describing how to interface with XML-based services.
7. WSDL is an integral part of UDDI, an XML-based worldwide business registry.
8. WSDL is the language that UDDI uses.
9. WSDL is pronounced as 'wiz-dull' and spelled out as 'W-S-D-L'.

UDDI: UDDI is an XML-based standard for describing, publishing, and finding web services.

1. UDDI stands for Universal Description, Discovery, and Integration.
2. UDDI is a specification for a distributed registry of web services.
3. UDDI is platform independent, open framework.

4. UDDI can communicate via SOAP, CORBA, and Java RMI Protocol.
5. UDDI uses WSDL to describe interfaces to web services.
6. UDDI is seen with SOAP and WSDL as one of the three foundation standards of web services.
7. UDDI is an open industry initiative enabling businesses to discover each other and define how they interact over the Internet.

RESTful WEB SERVICES

Restful Web Services is a lightweight, maintainable, and scalable service that is built on the REST architecture. Restful Web Service, expose API from your application in a secure, uniform, stateless manner to the calling client. The calling client can perform predefined operations using the Restful service. The underlying protocol for REST is HTTP. REST stands for REpresentational State Transfer.
RESTful web services are loosely coupled, lightweight web services that are particularly well suited for creating APIs for clients spread out across the internet. Representational State Transfer (REST) is an architectural style of client-server application cantered on the transfer of representations of resources through requests and responses. In the REST architectural style, data and functionality are considered resources and are accessed using Uniform Resource Identifiers (URIs), typically links on the Web. The resources are represented by documents and are acted upon by using a set of simple, well-defined operations.
For example, a REST resource might be the current weather conditions for a city. The representation of that resource might be an XML document, an image file, or an HTML page. A client might retrieve a particular representation, modify the resource by updating its data, or delete the resource entirely.
An application or architecture considered RESTful or REST-style has the following characteristics

1. State and functionality are divided into distributed resources – This means that every resource should be accessible via the normal HTTP commands of GET, POST, PUT, or DELETE. So if someone wanted to get a file from a server, they should be able to issue the GET request and get the file. If they want to put a file on the server, they should be able to either issue the POST or PUT request. And finally, if they wanted to delete a file from the server, they can issue the DELETE request.
2. The architecture is client/server, stateless, layered, and supports caching
   - Client-server is the typical architecture where the server can be the web server hosting the application, and the client can be as simple as the web browser.
   - Stateless means that the state of the application is not maintained in REST. For example, if you delete a resource from a server using the DELETE command, you cannot expect that delete information to be passed to the next request.

RESTful WEB SERVICES: COMPONENTS

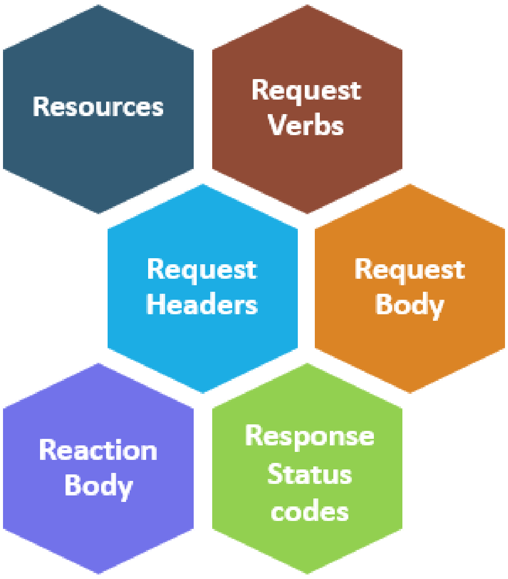

Figure 1.5: RESTful Web Services: Components

1. Resources: The main key component is simply an asset. Let expect that a web application on a

server has records of a few representatives.

2. Request Verbs: These portray what you need to do with the asset. A program gives a GET action word to train the endpoint it needs to get information. Be that as it may, there are numerous different action words accessible including things like POST, PUT, and DELETE.
3. Request Headers: These are extra guidelines sent with the solicitation. These might characterize the kind of reaction required or the approval subtleties.
4. Request Body: Data is sent with the solicitation. Information is regularly sent in the solicitation when a POST demand is made to the REST web administration. In a POST call, the customer really tells the web administration that it needs to add an asset to the server. Henceforth, the solicitation body would have the subtleties of the asset which is required to be added to the server.
5. Reaction Body: This is the fundamental body of the reaction.
6. Response Status Codes: These codes are the general codes that are returned alongside the reaction from the webserver. A model is the code 200 which is typically returned if there is no mistake while restoring a reaction to the customer.

CLOUD MODELS

Cloud models come in three types: SaaS (Software as a Service), IaaS (Infrastructure as a Service) and PaaS (Platform as a Service). Each of the cloud models has their own set of benefits that could serve the needs of various businesses.

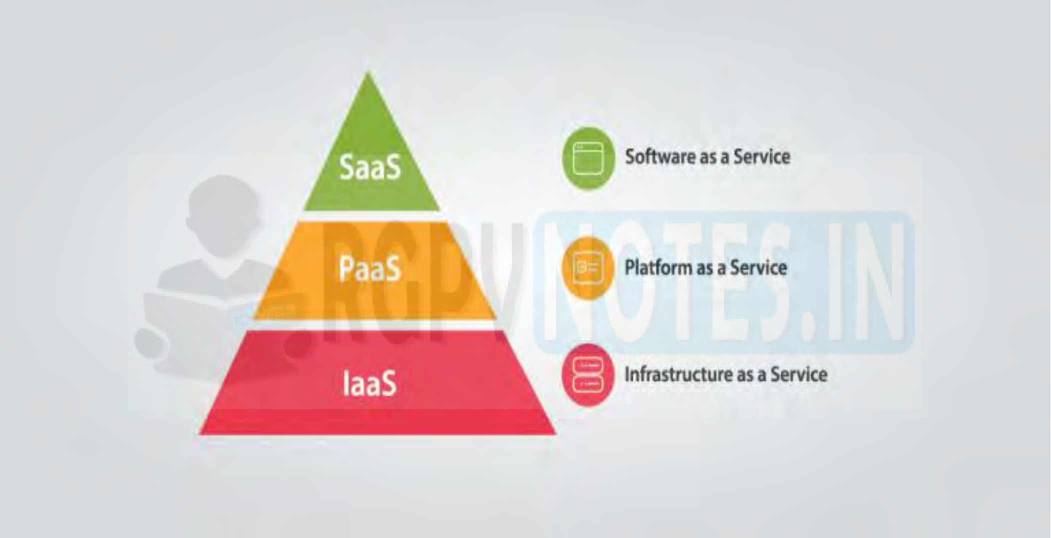

Figure 1.6: Cloud Models

SaaS: SaaS or Software as a Service is a model that gives quick access to cloud-based web applications. The vendor controls the entire computing stack, which you can access using a web browser. These applications run on the cloud and you can use them by a paid licensed subscription or for free with limited access.

SaaS does not require any installations or downloads in your existing computing infrastructure. This eliminates the need for installing applications on each of your computers with the maintenance and support taken over by the vendor. Some known examples of SaaS include Google G Suite, Microsoft Office 365, and Dropbox etc.

IaaS: IaaS or Infrastructure as a Service is basically a virtual provision of computing resources over the cloud. An IaaS cloud provider can give you the entire range of computing infrastructures such as storage, servers, networking hardware alongside maintenance and support.

Businesses can opt for computing resources of their requirement without the need to install hardware on their premises. Amazon Web Services, Microsoft Azure, and Google Compute Engine are some of the leading IaaS cloud service providers.

PaaS: Platform as a Service or PaaS is essentially a cloud base where you can develop, test and organize the different applications for your business. Implementing PaaS simplifies the process of enterprise software development. The virtual runtime environment provided by PaaS gives a

favourable space for developing and testing applications.
The entire resources offered in the form of servers, storage and networking are manageable either by the company or a platform provider. Google App Engine and AWS Elastic Beanstalk are two typical examples of PaaS. PaaS is also subscription based that gives you flexible pricing options depending on your business requirements.

ORGANIZATIONAL SCENARIOS OF CLOUDS

Public Cloud: Public cloud services are characterized as being available to clients from a third party service provider via the Internet. Public cloud vendors typically provide an access control mechanism for their users. Public clouds provide an elastic, cost effective means to deploy solutions.

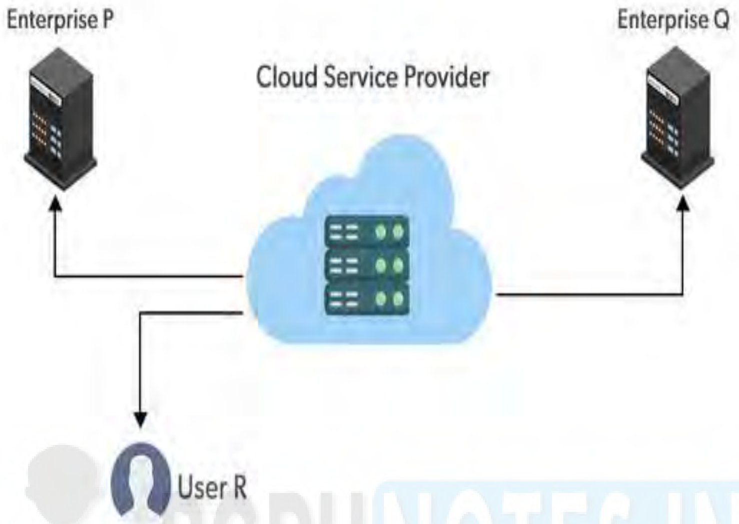

Figure 1.7: Public Cloud

Private Cloud: A private cloud is that in which, data and processes are managed within the organization without the restrictions of network bandwidth, security exposures and legal requirements that using public cloud services might entail. In addition, private cloud services offer the provider and the user greater control of the cloud infrastructure, improving security and resiliency because user access and the networks used are restricted and designated.

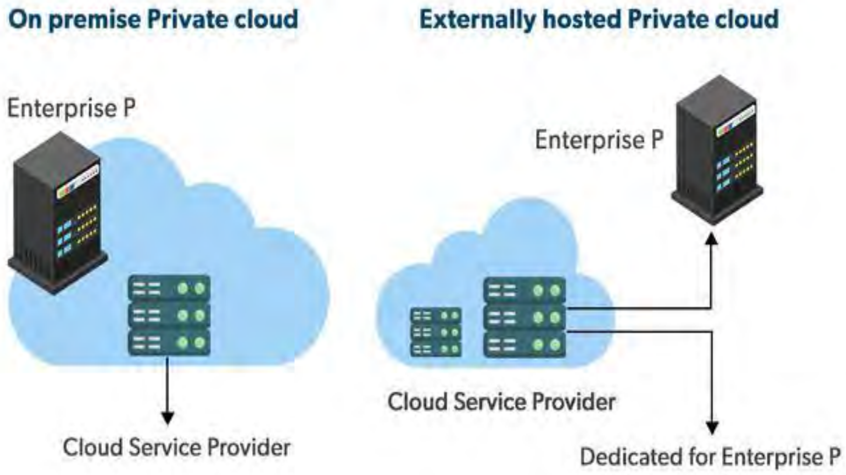

Figure 1.8: Private Cloud

Community Cloud: A community cloud is controlled and used by a group of organizations that have shared interests, such as specific security requirements or a common mission. The members of the community share access to the data and applications in the cloud.

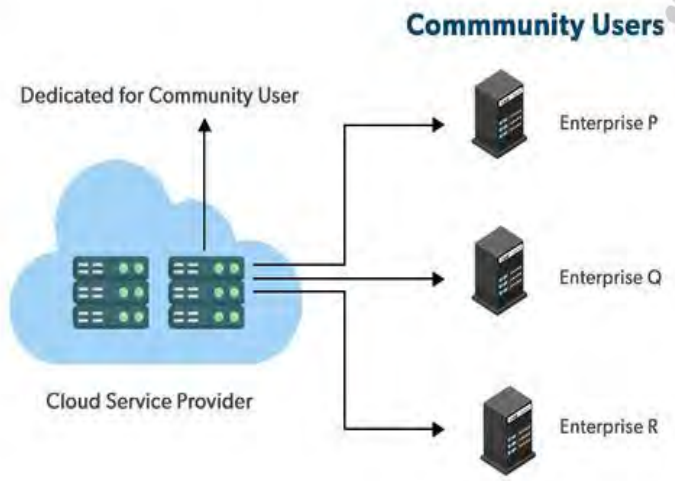

Figure 1.9: Private Cloud

Hybrid Cloud: A hybrid cloud is a combination of a public and private cloud that interoperates. In this model users typically outsource no business-critical information and processing to the public cloud, while keeping business-critical services and data in their control.

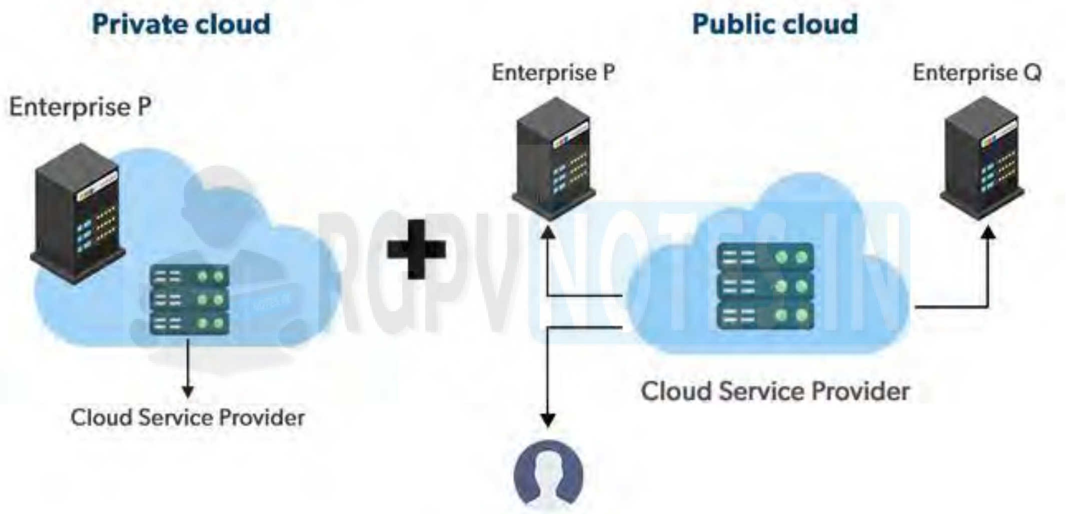

Figure 1.10: Hybrid Cloud

PUBLIC VERSES PRIVATE CLOUDS

S. No.	Public Cloud	Private Cloud
1.	Services owned and provided by third party provider	Service used by single organization
2.	The maintenance cost borne by service provider	Higher security as the resources are not shared
3.	Pay as you go Model. Thus setting and operating cost is less	Greater flexibility to control the cloud environment.
4.	Shared responsibility for security-provider and consumer	Opportunity to control entire cloud infra stack
5.	All resources are hosted on cloud providers infra.	Complex than using a public cloud.

Table 1.1: Difference between Public and Private Cloud

ADMINISTERING CLOUD COMPUTING SERVICES

Administering Cloud Computing services is an important process when you have hosted your business data on the cloud. The business owners need to know whether the performance is at the right level and whether the deleted data is permanently gone.

Investigating the reliability and viability of a cloud provider should be handled well when hosting data on the cloud. Business owners need to enable administration in terms of monitoring every dimension of the service they are getting.

Cloud Service provider can definitely build and provide stable services that are cost effective and efficient. However, there can be a serious gap between the actual service and the promised services.

BENEFITS OF CLOUD COMPUTING

1. Back-up and restore data: Once the data is stored in the cloud, it is easier to get back-up and restore that data using the cloud.
2. Improved collaboration: Cloud applications improve collaboration by allowing groups of people to quickly and easily share information in the cloud via shared storage.
3. Excellent accessibility: Cloud allows us to quickly and easily access store information anywhere, anytime in the whole world, using an internet connection. An internet cloud infrastructure increases organization productivity and efficiency by ensuring that our data is always accessible.
4. Low maintenance cost: Cloud computing reduces both hardware and software maintenance costs for organizations.
5. Mobility: Cloud computing allows us to easily access all cloud data via mobile.
6. IServices in the pay-per-use model: Cloud computing offers Application Programming Interfaces (APIs) to the users for access services on the cloud and pays the charges as per the usage of service.
7. Unlimited storage capacity: Cloud offers us a huge amount of storing capacity for storing our important data such as documents, images, audio, video, etc. in one place.
8. Data security: Data security is one of the biggest advantages of cloud computing. Cloud offers many advanced features related to security and ensures that data is securely stored and handled.

LIMITATION OF CLOUD COMPUTING

1. Internet Connectivity: As you know, in cloud computing, every data (image, audio, video, etc.) is stored on the cloud, and we access these data through the cloud by using the internet connection. If you do not have good internet connectivity, you cannot access these data. However, we have no any other way to access data from the cloud.
2. Vendor lock-in: Vendor lock-in is the biggest disadvantage of cloud computing. Organizations may face problems when transferring their services from one vendor to another. As different vendors provide different platforms, that can cause difficulty moving from one cloud to another.
3. Limited Control: As we know, cloud infrastructure is completely owned, managed, and monitored by the service provider, so the cloud users have less control over the function and execution of services within a cloud infrastructure.
4. Security: Although cloud service providers implement the best security standards to store important information. But, before adopting cloud technology, you should be aware that you will be sending all your organization's sensitive information to a third party, i.e., a cloud computing service provider. While sending the data on the cloud, there may be a chance that your organization's information is hacked by Hackers.

STUDY OF A HYPERVISOR

A hypervisor is a form of virtualization software used in Cloud hosting to divide and allocate the resources on various pieces of hardware. The program which provides partitioning, isolation or abstraction is called virtualization hypervisor. The hypervisor is a hardware virtualization technique that allows multiple guest operating systems (OS) to run on a single host system at the same time. A hypervisor is sometimes also called a virtual machine manager (VMM).

TYPE-1 Hypervisor: The hypervisor runs directly on the underlying host system. It is also known as "Native Hypervisor" or "Bare metal hypervisor". It does not require any base server operating system. It has direct access to hardware resources. Examples of Type 1 hypervisors include VMware ESXi, Citrix XenServer and Microsoft Hyper-V hypervisor.

Pros & Cons of Type-1 Hypervisor:

Pros: Such kind of hypervisors are very efficient because they have direct access to the physical hardware resources (like Cpu, Memory, Network, Physical storage). This causes the empowerment the security because there is nothing any kind of the third-party resource so that attacker couldn't compromise with anything.

Cons: One problem with Type-1 hypervisor is that they usually need a dedicated separate machine to perform its operation and to instruct different VMs and control the host hardware resources.

TYPE-2 Hypervisor: A Host operating system runs on the underlying host system. It is also known as 'Hosted Hypervisor". Such kind of hypervisors doesn't run directly over the underlying hardware rather they run as an application in a Host system (physical machine). Basically, software installed on an operating system. Hypervisor asks the operating system to make hardware calls. Example of Type 2 hypervisor includes VMware Player or Parallels Desktop. Hosted hypervisors are often found on endpoints like PCs. The type-2 hypervisor is are very useful for engineers, security analyst (for checking malware, or malicious source code and newly developed applications).

Pros & Cons of Type-2 Hypervisor:

Pros: Such kind of hypervisors allows quick and easy access to a guest Operating System alongside the host machine running. These hypervisors usually come with additional useful features for guest machine. Such tools enhance the coordination between the host machine and guest machine.

Cons: Here there is no direct access to the physical hardware resources so the efficiency of these hypervisors lags in performance as compared to the type-1 hypervisors, and potential security risks are also there an attacker can compromise the security weakness if there is access to the host operating system so he can also access the guest operating system.