Cloud computing

What is cloud computing

* Cloud computing is the delivery of different services through the Internet, including data storage, servers, databases, networking, and software.
* Cloud computing means storing and accessing the data and programs on remote servers that are hosted on the internet instead of the computer’s hard drive or local server.
* Cloud computing is also referred to as Internet-based computing, it is a technology where the resource is provided as a service through the Internet to the user.

Why Cloud Computing?

* Reduces cost
* More storage
* Reliability
* Scalability

Distributed cloud computing

Distributed cloud computing splits one task across multiple computers at different locations—all of which are networked. Each computer will complete an aspect of the task, allowing for the task to be completed faster.

Cloud deployment models

* **Private Cloud:** Resource managed and used by the organization.
* **Public Cloud:** Resource available for the general public under the Pay as you go model.
* **Community Cloud:** Resource shared by several organizations, usually in the same industry.
* **Hybrid Cloud:**This cloud deployment model is partly managed by the service provided and partly by the organization **.its a combination of public and private cloud**

## Cloud Service Models

SaaS :-

* Software as a Service (SaaS) is a web-based deployment model that makes the software accessible through a web browser.
* SaaS software users don’t need to care where the software is hosted, which operating system it uses, or even which programming language it is written in.
* The SaaS software is accessible from any device with an internet connection.
* This cloud service model ensures that consumers always use the most current version of the software.
* The SaaS provider handles maintenance and support.
* In the SaaS model, users don’t control the infrastructure, such as storage, processing power, etc.

## Platform as a Service (PaaS)

* Platform-as-a-Service (PaaS) provides a cloud computing framework for software application creation and deployment.
* It is a platform for the deployment and management of software apps. This flexible cloud computing model scales up automatically on demand.
* It also manages the servers, storage, and networking, while the developers manage only the application part.
* It offers a runtime environment for application development and deployment tools.
* This Model provides all the facilities required to support the complex life cycle of building and delivering web applications and services entirely for the Internet.
* This cloud computing model enables developers to rapidly develop, run, and manage their apps without building and maintaining the infrastructure or platform.

## Infrastructure as a Service

* Infrastructure-as-a-Service (IaaS) is a cloud computing service offering on-demand computing, storage, and networking resources.
* It usually works on a pay-as-you-go basis.
* Organizations can purchase resources on-demand and as needed instead of buying the hardware outright.
* The IaaS cloud vendor hosts the infrastructure components, including the on-premises data center, servers, storage, networking hardware, and the hypervisor (virtualization layer).
* This Model contains the basic building blocks for your web application. It provides complete control over the hardware that runs your application (storage, servers, VMs, networks & operating systems).
* IaaS model gives you the best flexibility and management control over your IT resources.

**DRaaS**– Disaster Recovery as a service is a cloud computing model that provides safeguards from natural (or artificial) catastrophes.

Database as a service (DBaaS) is a cloud computing managed service offering that provides access to a database without requiring the setup of physical hardware, the installation of software or the need to configure the database. Most database administration and maintenance tasks are handled by the service provider, enabling users to quickly benefit from the database service.

Virtualization

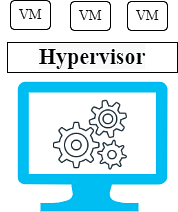
The process of virtualization is a way to create multiple virtually simulated instances over the computer hardware to utilize your system's underlying resources fully.

The software hypervisor allows the elements of the system, such as storage, memory, processor, etc, to be distributed among multiple separate and secure virtual computers, created using the hypervisor, which is termed as [virtual machines(VMs)](https://www.simplilearn.com/setting-up-virtual-machines-for-your-microsoft-azure-cloud-platform-article), where every virtual machine has its dedicated operating system which uses a part of the system's hardware resources for operation.

Virtualization is a foundational [element of cloud computing](https://www.simplilearn.com/tutorials/cloud-computing-tutorial/cloud-computing-architecture) and enables multiple users to share a single physical instance of a resource at a time.

the virtual machine is the emulation or a virtual representation of a physical device that can execute multiple operating systems on the same computer with a different operating system.

## How Does Virtualisation Work?



* A hypervisor is a software that allows you to create a virtual layer over the hardware system that manages the interaction between the virtual machines and the system's hardware resources.
* They are installed as any other software application in the system and perform the task of virtualization.
* It acts as a connection between the physical system and virtual machines to ensure the proper access of the hardware resources
* It also manages so that the virtual machines don't interfere with each other's memory and computing resources.
* The hypervisor also manages the Virtual machines and is known as the virtual machine monitor (VMM).

Types of hypervisor

### Type 1 Hypervisors

* A Type 1 hypervisor runs directly on a physical host.
* That’s why it’s also known as a bare metal hypervisor.
* Basically, you would install a Type 1 hypervisor before anything else on a physical host, so it sort of acts like that host’s operating system.
* Consequently, a Type 1 hypervisor has direct access to the underlying physical host’s resources—e.g., CPU, RAM, storage, and network interface

### Type 2 Hypervisors

* A Type 2 hypervisor runs on top of a host OS. For this reason, it’s also known as a hosted hypervisor.
* So, you would have to install a host OS on your physical host before you can install a Type 2 hypervisor.
* When a Type 2 hypervisor needs to communicate with the underlying hardware or access hardware resources, it must go through the host OS first.
* Type 2 hypervisors are usually easier to set up and use.

Type of virtualization

Full virtualization and para virtualization

1. Full virtualization enables the Guest operating system to run independently. In contrast, paravirtualization enables the Guest OS to interact with the hypervisor.
2. Full virtualization supports all the Guest OS without any change. On the other hand, the Guest OS has to be modified in paravirtualization and only a few OS support it.
3. The Guest OS will issue hardware calls in full virtualization. In contrast, the guest OS will interface directly with the hypervisor via drivers in paravirtualization.
4. Guest OS does not know that it built upon hypervisor whereas Guest OS knows that it built upon hypervisor
5. Full virtualization performance is slow. In contrast, paravirtualization performance is high than full virtualization.

Implementation levels of virtualization

* Virtualization is not that easy to implement. A computer runs an OS that is configured to that particular hardware.
* Running a different OS on the same hardware is not exactly feasible.
* To tackle this, there exists a hypervisor.
* What hypervisor does is, it acts as a bridge between virtual OS and hardware to enable its smooth functioning of the instance.
* There are five levels of virtualizations available that are most commonly used in the industry.

These are as follows:

### Instruction Set Architecture Level (ISA)

* In ISA, virtualization works through an ISA emulation.
* This is helpful to run heaps of legacy code which was originally written for different hardware configurations.
* These codes can be run on the virtual machine through an ISA.
* A binary code that might need additional layers to run can now run on an x86 machine or with some tweaking, even on x64 machines.
* ISA helps make this a hardware-agnostic virtual machine.
* The basic emulation, though, requires an interpreter.
* This interpreter interprets the source code and converts it to a hardware readable format for processing.

### Hardware Abstraction Level (HAL)

* As the name suggests, this level helps perform virtualization at the hardware level.
* It uses a bare hypervisor for its functioning.
* This level helps form the virtual machine and manages the hardware through virtualization.
* It enables virtualization of each hardware component such as I/O devices, processors, memory, etc.
* This way multiple users can use the same hardware with numerous instances of virtualization at the same time.

### Operating System Level

* At the operating system level, the virtualization model creates an abstract layer between the applications and the OS.
* It is like an isolated container on the physical server and operating system that utilizes hardware and software.
* Each of these containers functions like servers.
* When the number of users is high, and no one is willing to share hardware, this level of virtualization comes in handy.
* Here, every user gets their own virtual environment with dedicated virtual hardware resources. This way, no conflicts arise.

### Library Level

* OS system calls are lengthy and cumbersome.
* Which is why applications opt for APIs from user-level libraries.
* Most of the APIs provided by systems are rather well documented. Hence, library level virtualization is preferred in such scenarios.
* Library interfacing virtualization is made possible by API hooks.
* These API hooks control the communication link from the system to the applications.

### Application Level

* Application-level virtualization comes handy when you wish to virtualize only an application.
* It does not virtualize an entire platform or environment.
* On an operating system, applications work as one process.
* Hence it is also known as process-level virtualization.
* It is generally useful when running virtual machines with high-level languages.
* Here, the application sits on top of the virtualization layer, which is above the application program.
* The application program is, in turn, residing in the operating system.
* Programs written in high-level languages and compiled for an application-level virtual machine can run fluently here.

Cloud Scalability

* Cloud scalability is the ability of a [cloud computing](https://www.simplilearn.com/tutorials/cloud-computing-tutorial/what-is-cloud-computing) system to adapt to changing computing requirements by either increasing or decreasing its resources, such as computing power, storage, or network capacity on demand.
* It allows the system to adjust its resources to the workload to meet the required performance levels.
* This scalability often involves increasing or decreasing the number of servers, storage, or other computing resources.

## Vertical Scaling

* Vertical scaling in [cloud computing](https://www.simplilearn.com/pgp-cloud-computing-certification-training-course?source=GhPreviewCoursepages) is adding resources to an existing instance or server to increase its capacity or capabilities.
* It automatically enables the system to allocate more or fewer resources to meet changing requirements.
* Vertical scaling is usually done by increasing the server's computing power, adding more RAM or CPU cores, or adding storage capacities, such as hard disks or solid-state drives.
* Vertical scaling enables your applications to run faster and handle more load without purchasing a new server or instance.
* And vertical scaling is a popular choice for cloud computing because it is relatively easy to do and does not require any changes to the existing infrastructure.

## Horizontal Scaling

* Horizontal scaling in cloud computing refers to the ability to scale out a system by adding more nodes, or servers, to the system.
* This scaling is often used to improve the cluster's processing power, allowing applications and services to handle more concurrent requests or to process more significant amounts of [data](https://www.simplilearn.com/what-is-data-article).
* In cloud computing, horizontal scaling is usually achieved by adding additional virtual machines (VMs), containers, or other resources to an existing cluster.
* This type of scaling is often used to improve performance or to handle increased traffic.
* When done correctly, horizontal scaling can be a very effective way to enhance the performance of a system.

## Diagonal Scaling

* In cloud computing, diagonal scaling is a scaling in which the system is scaled vertically and horizontally, allowing for the addition of new [nodes](https://www.simplilearn.com/tutorials/nodejs-tutorial/what-is-nodejs) (machines) to both the columns and rows of cloud infrastructure simultaneously.
* This type of scaling is often used to improve performance and expand the system's capacity.
* Diagonal scaling can be used in a cloud system to add more servers, storage, and networking resources.
* This type of scaling can also improve the system's performance by adding more resources.
* Additionally, diagonal scaling can improve the fail-over capability of cloud infrastructure by increasing the number of nodes that can be used in a distributed architecture.