

# **Overview**

- **Introduction to Cloud Computing**
- **History of Cloud Computing**
- **Distributed Computing**
- **Cluster Computing**

# **Introduction to Cloud Computing**

## **Definition**

Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model is composed of five essential characteristics, three service models, and four deployment models.

# Introduction to Cloud Computing

## Characteristics

- On-demand self-service
  - Unilateral provision of computing capabilities
  - Automatic access without human interaction
- Broad network access
  - Available over the network
  - Access through standard mechanisms
  - Heterogeneous thin or thick client platforms
- Resource pooling
  - Multi-tenant model
  - Assigns and reassigns physical and virtual resources based on demand preserving privacy and security
  - Physical location of resources is not known to the customers except special circumstances

# Introduction to Cloud Computing

## Characteristics

- Rapid elasticity
  - Cloud services can be elastically provisioned and released automatically to facilitate inward and outward scalability based on demand
- Measured service
  - Optimization of resource usage
  - Measured service for storage, processing, bandwidth, and users
  - Monitoring, controlling, and reporting resource use improves transparency

# Introduction to Cloud Computing

## Service Models

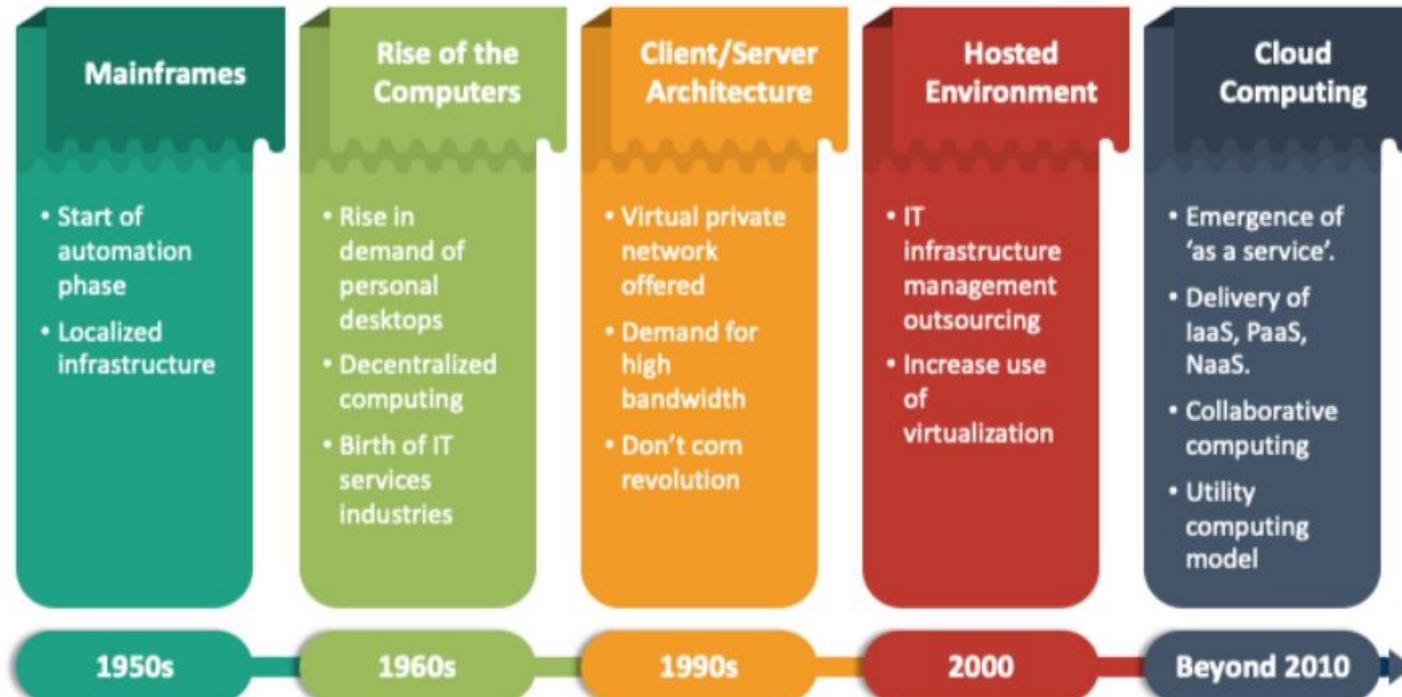
- Software as a Service (SaaS)
  - Use the provider's applications running on a cloud infrastructure
  - The applications are accessible from various client devices
  - The consumer does not manage or control the underlying cloud infrastructure
  - Example : Slack, Zoom etc
- Platform as a Service (PaaS)
  - Consumers deploy consumer-created or acquired applications onto the cloud infrastructure
  - Has control over the deployed applications
  - Configuration settings for the application-hosting environment
  - Example : Heroku, Google App Engine

# Introduction to Cloud Computing

## Service Models

- Infrastructure as a Service (IaaS)
  - Consumer use processing, storage, networks, and other fundamental computing resources to deploy and run arbitrary software
  - Example : DigitalOcean, Amazon Web Services

# History to Cloud Computing



# Distributed Computing

## Definition

Distributed computing is the method of making multiple computers work together to solve a common problem. It makes a computer network appear as a powerful single computer that provides large-scale resources to deal with complex challenges.

<https://aws.amazon.com/what-is/distributed-computing/>

# **Cluster Computing**

## **Definition**

Cluster computing is a collection of tightly or loosely connected computers that work together so that they act as a single entity. The connected computers execute operations all together thus creating the idea of a single system. The clusters are generally connected through fast local area networks (LANs)

# Distributed Computing vs Cluster Computing

- Distributed, in a narrow sense, is similar to a cluster, but its organization is relatively loose
- Each node in the distributed network completes different services. When one node is smashed, the service is inaccessible
- Distributed refers to splitting a business into different sub-services and distributing them on different machines
- Cluster refers to a group of servers that are grouped together to achieve the same business and can be considered as one computer

<https://medium.com/@mena.meseha/difference-between-distributed-and-cluster-a-ca9d50c2c44>

# **Overview**

- **Distributed Computing**
- **Cluster Computing**
- **Grid Computing**
- **Mobile Computing**

# Distributed Computing

## Definition

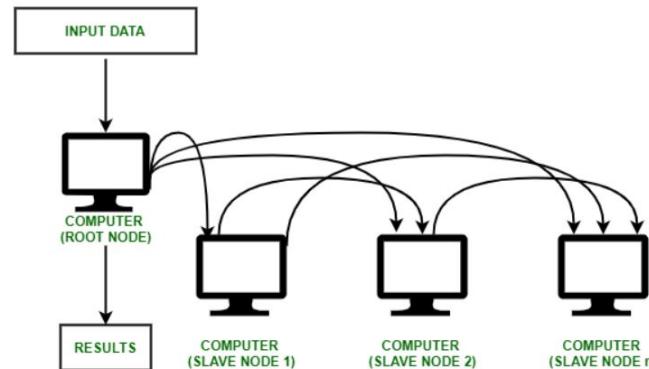
Distributed computing is the method of making multiple computers work together to solve a common problem. It makes a computer network appear as a powerful single computer that provides large-scale resources to deal with complex challenges.

<https://aws.amazon.com/what-is/distributed-computing/>

# Cluster Computing

## Definition

Cluster computing describes a network system comprised of homogeneous computers. Homogeneous computers have the same hardware and software. You can connect them to a high-speed local network to create a computer cluster that runs similar tasks. A centralized server controls and coordinates the machines.



<https://aws.amazon.com/what-is/grid-computing/>

# **Cluster Computing**

## **Advantages**

- High Performance
- Easy to manage
- Scalable
- Expandability
- Availability
- Flexibility

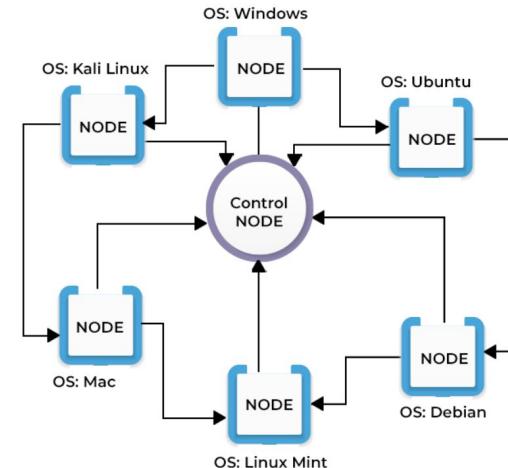
## **Disadvantages**

- High cost
- Problem in finding fault
- More space is needed

# Grid Computing

## Definition

Grid computing is a computing infrastructure that combines computer resources spread over different geographical locations to achieve a common goal. All unused resources on multiple computers are pooled together and made available for a single task. Organizations use grid computing to perform large tasks or solve complex problems that are difficult to do on a single computer.



<https://aws.amazon.com/what-is/grid-computing/>

# **Grid Computing**

## **Advantages**

- Easier to collaborate with other organizations.
- Make better use of existing hardware.
- It works as a super computer so that we do not have to buy such an expensive computer.
- There is no risk of failure.
- Jobs can be executed in parallel speeding performance
- Can solve larger, more complex problems in a shorter time

## **Disadvantages**

- Security concern because there is more opportunity to attack in it.
- Complex software required
- Reluctance in collaboration
- Still evolving
- Licensing across many servers may make it prohibitive for some applications

# Cluster Computing vs Grid Computing

Nodes or computers are homogeneous. Network are also should be homogeneous

All the computers in a cluster are dedicated to a single task

Computers of Cluster computing are co-located and are connected by high speed network bus cables

Cluster computing network is prepared using a centralized network topology

A centralized server controls the scheduling of tasks in cluster computing.

A dedicated centralized resource manager, managing the resources of all the nodes connected.

Nodes or computers are heterogeneous. Grid computers can have homogeneous or heterogeneous network

Computers of Grid Computing can leverage the unused computing resources to do other tasks.

Computers of Grid Computing can be present at different locations and are usually connected by the Internet or a low speed network bus.

Grid computing network is distributed and have a decentralized network topology.

Each node behaves independently without the need of any centralized scheduling server.

In Grid Computing, each node independently manages its own resources.

# Mobile Computing

Mobile Computing is a technology that provides an environment that enables users to transmit data from one device to another device without the use of any physical link or cables. In mobile computing, a computer is expected to be moving during normal usage.

Key concepts of mobile computing :

- Mobile Communication
- Mobile Hardware
- Mobile software

# **Mobile Computing**

## **Advantages**

- Seamless and reliable communication
- Increased Productivity
- Entertainment
- Portability
- Cloud Computing

## **Disadvantages**

- The obstacle to battery consumption.
- The transmission bandwidth is inefficient.
- Over the whole network, there are link losses.
- Fluctuation in the stability of the network.
- Small screen sizes.
- The issue of interoperability

# **Overview**

- **Service Oriented Architecture (SOA)**
- **Utility Computing**
- **Web 2.0**
- **Parallel Computing**

# Service Oriented Architecture (SOA)

SOA, or service-oriented architecture, defines a way to make software components reusable and interoperable via service interfaces. Services use common interface standards and an architectural pattern so they can be rapidly incorporated into new applications.

- Removes tasks from the application developer who
  - Previously redeveloped or duplicated existing functionality
  - Had to know how to connect or provide interoperability with existing functions
- Example of Services
  - Checking a customer's credit
  - Calculating a monthly loan payment
  - Processing a mortgage application

# Benefits of Service Oriented Architecture (SOA)

- **Faster time to market**
  - Developers reuse services across different business processes to save time and costs.
  - They can assemble applications much faster with SOA than by writing code and performing integrations from scratch.
- **Efficient maintenance**
  - It's easier to create, update, and debug small services than large code blocks in monolithic applications.
  - Modifying any service in SOA does not impact the overall functionality of the business process.
- **Greater adaptability**
  - SOA is more adaptable to advances in technology.
  - You can modernize your applications efficiently and cost effectively.

<https://aws.amazon.com/what-is/service-oriented-architecture/>

# **Principles of Service Oriented Architecture (SOA)**

- **Interoperability**
  - Any client system can run a service, regardless of the underlying platform or programming language
- **Loose coupling**
  - Little dependency on external resources such as data models or information systems
  - They should also be stateless without retaining any information from past sessions or transactions.
- **Abstraction**
  - Services should appear like a black box.
- **Granularity**
  - Services in SOA should have an appropriate size and scope, ideally packing one discrete business function per service.

# **Components of Service Oriented Architecture (SOA)**

- **Service**
  - **Service implementation** : The service implementation is the code that builds the logic for performing the specific service function
  - **Service contract** : The nature of the service and its associated terms and conditions
  - **Service interface** : How can invoke the service or exchange data
- **Service provider**
  - Creates, maintains, and provides one or more services
- **Service consumer**
  - Requests the service provider to run a specific service
  - Can be an entire system, application, or other service.
- **Service registry**
  - It is a network-accessible directory of available services. It stores service description documents from service providers.

# Service Oriented Architecture (SOA)

- **Type of Service**
  - **Private** : Available only to internal users of an organization
  - **Public** : Accessible over the internet to all
- **Service Level Agreement**
  - A service-level agreement (SLA) is a commitment between a service provider and a customer. Particular aspects of the service – quality, availability, responsibilities – are agreed between the service provider and the service user.

# 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.
- The utility model seeks to maximize the efficient use of resources and/or minimize associated costs.
- Utility is the packaging of system resources, such as computation, storage and services, as a metered service.
- This model has the advantage of a low or no initial cost to acquire computer resources; instead, resources are essentially rented.

[https://en.wikipedia.org/wiki/Utility\\_computing](https://en.wikipedia.org/wiki/Utility_computing)

# Grid computing vs Utility Computing

A process architecture that combines different computing resources from multiple locations to achieve desired and common goal

Distributes workload across multiple systems and allow computers to contribute their individual resources to common goal

Makes better use of existing resources, address rapid fluctuations in customer demands, improve computational capabilities, provide flexibility, etc

It mainly focuses on sharing computing resources.

Its characteristics include resource coordination, transparent access, dependable access, etc.

A process architecture that provide on-demand computing resources and infrastructure on basis of pay per use method

Allows organization to allocate and segregate computing resources and infrastructure to various users on basis of their requirements

It simply reduces IT costs, easier to manage, provide greater flexibility, compatibility, provide more convenience, etc.

It mainly focuses on acquiring computing resources.

Its characteristics include scalability, demand pricing, standardized utility computing services, automation, etc.

# Web 2.0

- Web 2.0 (also known as participative (or participatory) web and social web) refers to websites that emphasize user-generated content, ease of use, participatory culture and interoperability (i.e., compatibility with other products, systems, and devices) for end users.
- A Web 2.0 website allows users to interact and collaborate with each other through social media dialogue as creators of user-generated content in a virtual community.
- This contrasts the first generation of Web 1.0-era websites where people were limited to viewing content in a passive manner.

# **Web 2.0 : Features**

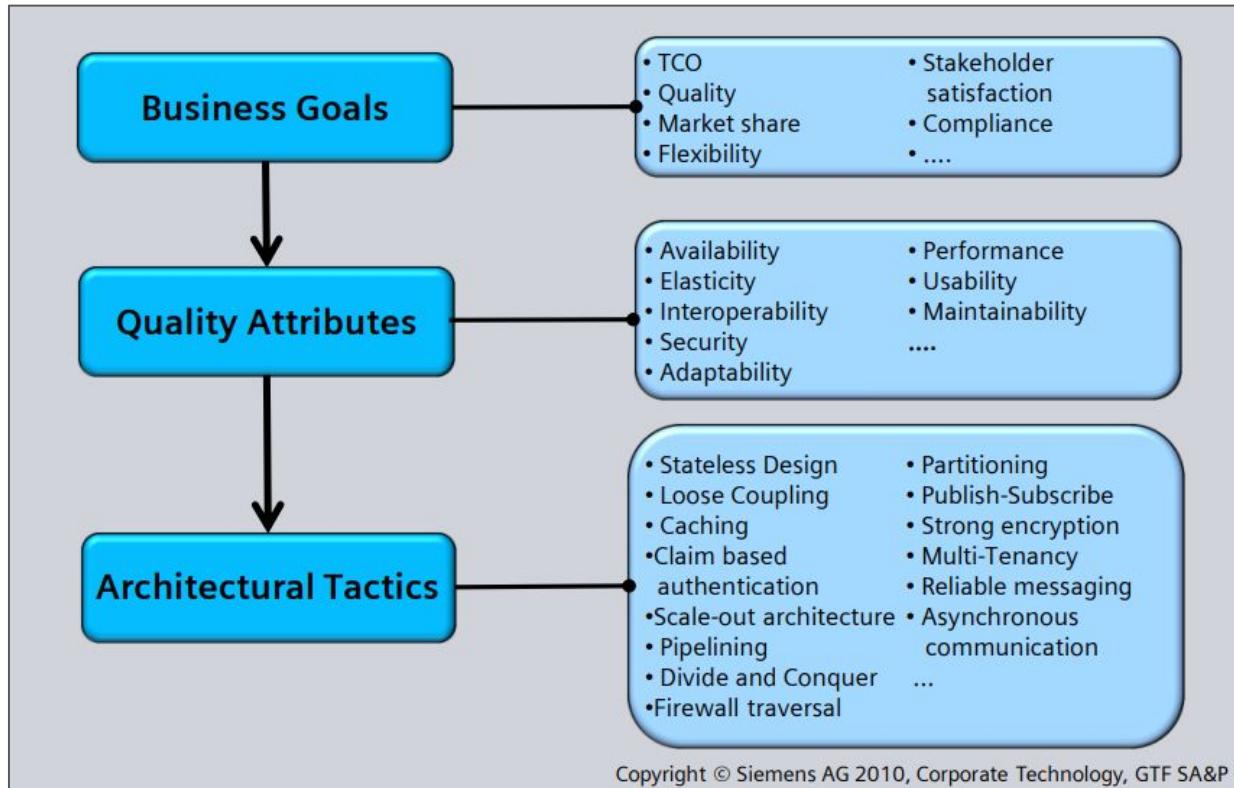
- Easy to access
- User interaction and participation
- Rich customization features
- Easy communication through video chatting, instant messaging facilities, etc.
- User-friendly writing tools and applications
- Data management and analysis
- Multimedia supporting tools
- Web application and hosting

# Parallel Computing

- Parallel computing simultaneously uses various computing resources for solving a computational problem.
- It is based on the principle that a single large problem is divided into small parts and parallelly runs different parts on different machines.
- Parallel computing supports applications that require processing of a large problem in a sophisticated way.
- Some of the examples are Big data problem, Data mining, Search engines, Medical diagnosis, Virtual reality, Multimedia

# Cloud Computing Architecture

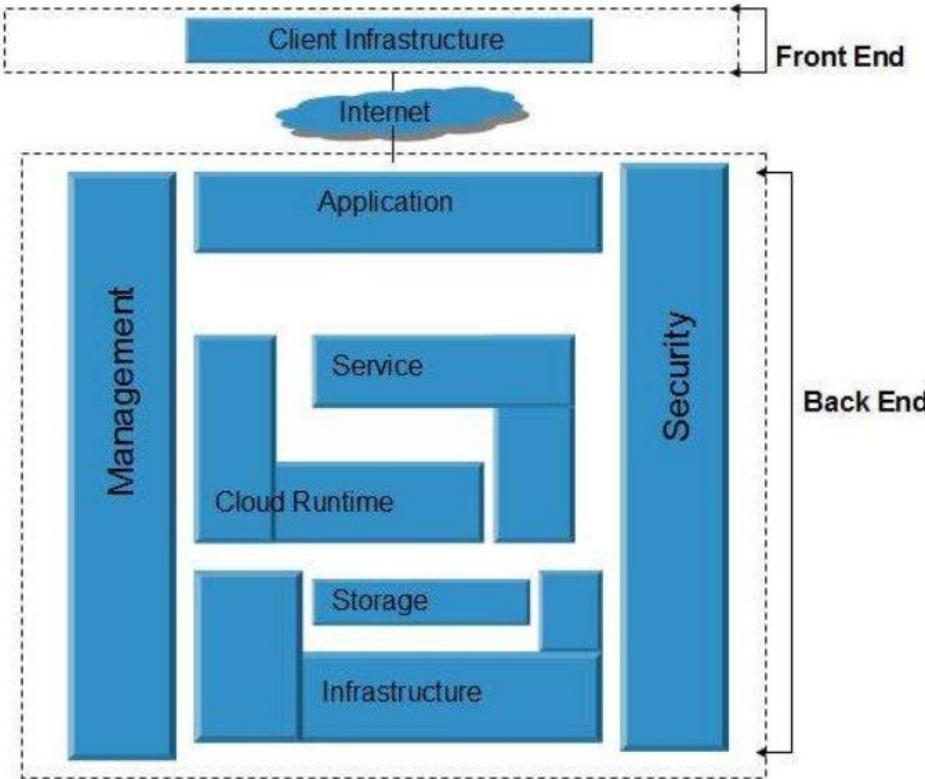
# High-level Architectural Approach



# Major building blocks of Cloud Computing Architecture

- **Reference Architecture**
  - Basis for documentation, project communication
  - Stakeholder and team communication
  - Payment, contract, and cost models
- **Technical Architecture**
  - Structuring according to XaaS Stack
  - Adopting Cloud Platform paradigms
  - Structuring cloud services and cloud components
  - Showing relationships and external endpoints
  - Middleware and communication
  - Management and security
- **Deployment Operation Architecture**
  - Geo-location check (Legal issues, export control)
  - Operation and monitoring

# Cloud Computing Architecture



# Cloud Computing Architecture

- **Cloud Computing Architecture**
  - **Frontend**
    - The client side of cloud computing system
    - **Client Infrastructure** : All the user interfaces and applications that are required to access the cloud platform
  - **Backend**
    - **Application :**
      - Software or platform to which client accesses
      - Provides the service in backend as per the client requirement
    - **Services :**
      - **SaaS**
      - **PaaS**
      - **IaaS**

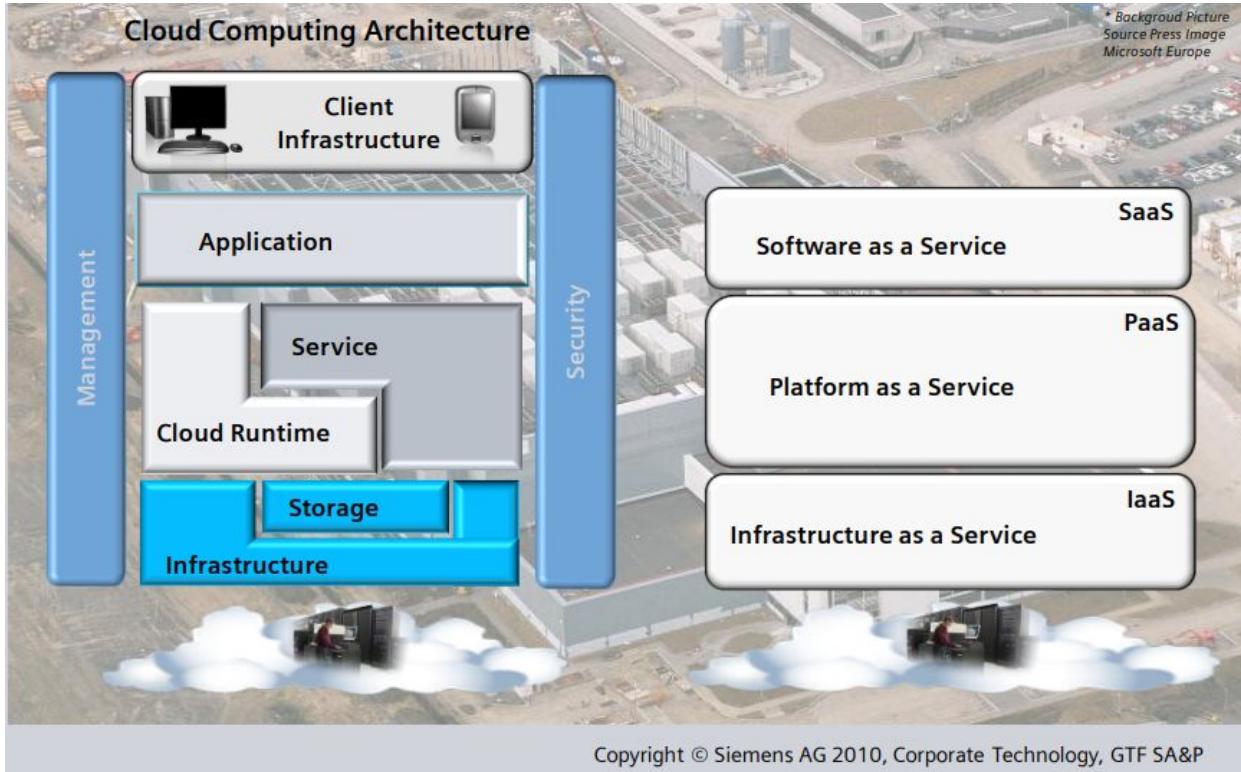
# Cloud Computing Architecture

- **Cloud Computing Architecture**
  - **Backend**
    - **Cloud Runtime :**
      - Provides the execution and Runtime platform/environment to the Virtual machine
    - **Storage :**
      - Provides flexible and scalable storage service and management of stored data
    - **Infrastructure :**
      - The hardware and software components of cloud
      - Includes servers, storage, network devices, virtualization software etc

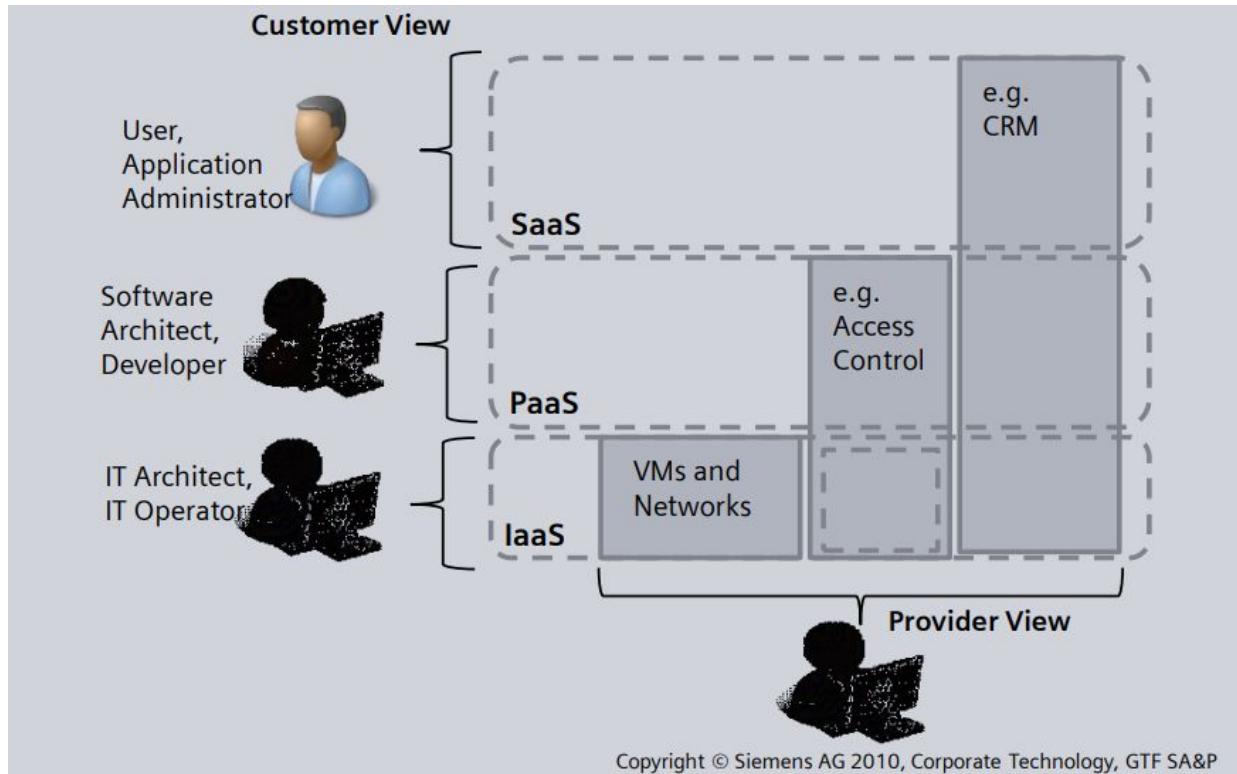
# Cloud Computing Architecture

- **Cloud Computing Architecture**
  - **Backend**
    - **Management :**
      - Management of backend components of every layer
    - **Security :**
      - Implementation of different security mechanisms in the backend
    - **Internet :**
      - Medium or a bridge between frontend and backend

# Cloud Computing Architecture vs XaaS

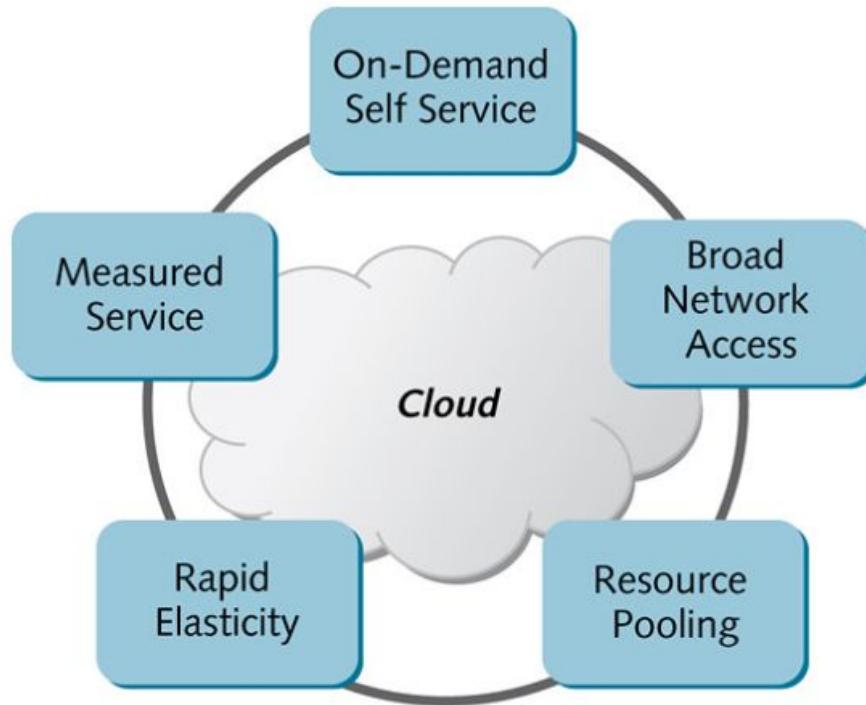


# Cloud Computing XaaS Stack Views



# Characteristics of Cloud Computing

# Cloud Computing Characteristics



# On-Demand Self Service

**Definition :** A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service provider.

## Benefits :

- Users can immediately deploy or remove resources to satisfy their needs without a **human intermediary** after signing up
- Help a business achieve its digital transformation objectives and be more responsive to **customer needs**.

# Broad Network Access

**Definition :** Capabilities are available over the network and accessed through **standard mechanisms** that promote use by **heterogeneous** thin or thick client platforms (e.g., mobile phones, tablets, laptops, and workstations).

The network can be :

- Intranet (For private cloud)
- Internet (For public cloud)

[https://csrc.nist.gov/glossary/term/broad\\_network\\_access](https://csrc.nist.gov/glossary/term/broad_network_access)

# Rapid Elasticity

**Definition** : Capabilities can be elastically provisioned and released, in some cases **automatically**, to scale rapidly outward and inward commensurate with demand. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be appropriated in any quantity at any time.

[https://csrc.nist.gov/glossary/term/rapid\\_elasticity](https://csrc.nist.gov/glossary/term/rapid_elasticity)

# Resource pooling

**Definition** : The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and re-assigned according to consumer demand. There is a sense of **location independence** in that the customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state, or datacenter).

Examples of resources include storage, processing, memory, and network bandwidth.

[https://csrc.nist.gov/glossary/term/resource\\_pooling](https://csrc.nist.gov/glossary/term/resource_pooling)

# Elasticity vs Scalability

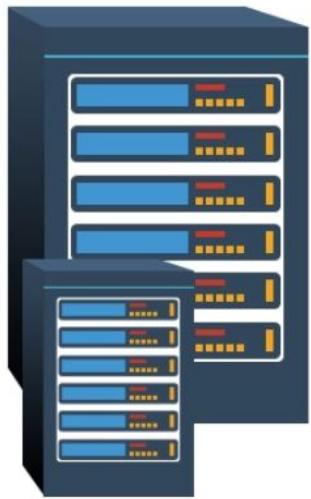
Cloud Elasticity	Cloud Scalability
1 Elasticity is used just to meet the sudden up and down in the workload for a small period of time.	Scalability is used to meet the static increase in the workload.
2 Elasticity is used to meet dynamic changes, where the resources need can increase or decrease.	Scalability is always used to address the increase in workload in an organization.
3 Elasticity is commonly used by small companies whose workload and demand increases only for a specific period of time.	Scalability is used by giant companies whose customer circle persistently grows in order to do the operations efficiently.
4 It is a short term planning and adopted just to deal with an unexpected increase in demand or seasonal demands.	Scalability is a long term planning and adopted just to deal with an expected increase in demand.

# Types Of Elasticity In Cloud Computing

You can take advantage of cloud elasticity in four forms; scaling out or in and scaling up or down.

- Vertical Scale Up/down :
  - Add more resources to a single computation unit i.e.  
Buy a bigger box
  - Move a workload to a computation unit with more  
resources
- Horizontal Scale Out/in
  - Adding additional computation units and having them  
act in concert
  - Splitting workload across multiple computation units

# Types Of Elasticity In Cloud Computing



Vertical Scaling  
(Scaling up)



Horizontal Scaling  
(Scaling out)

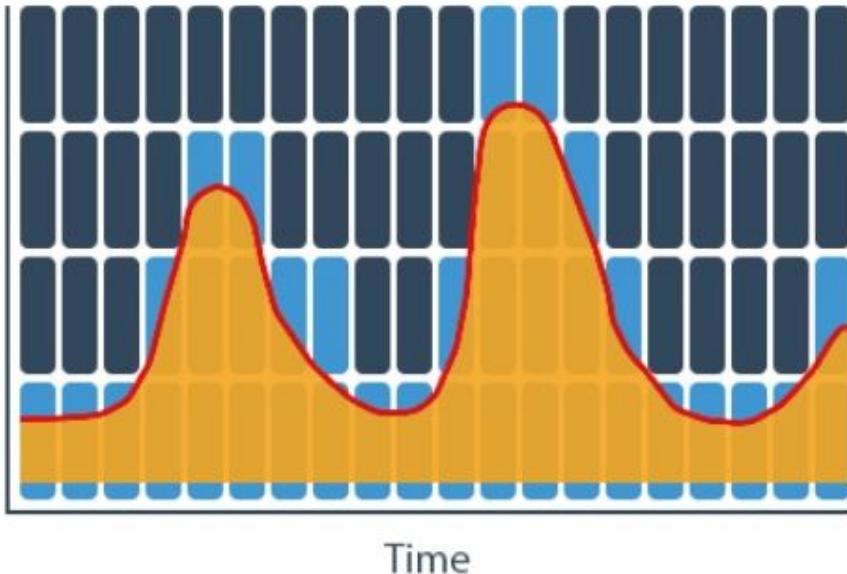
# What Is The Purpose Of Cloud Elasticity?

Cloud elasticity helps users prevent over-provisioning or under-provisioning system resources.

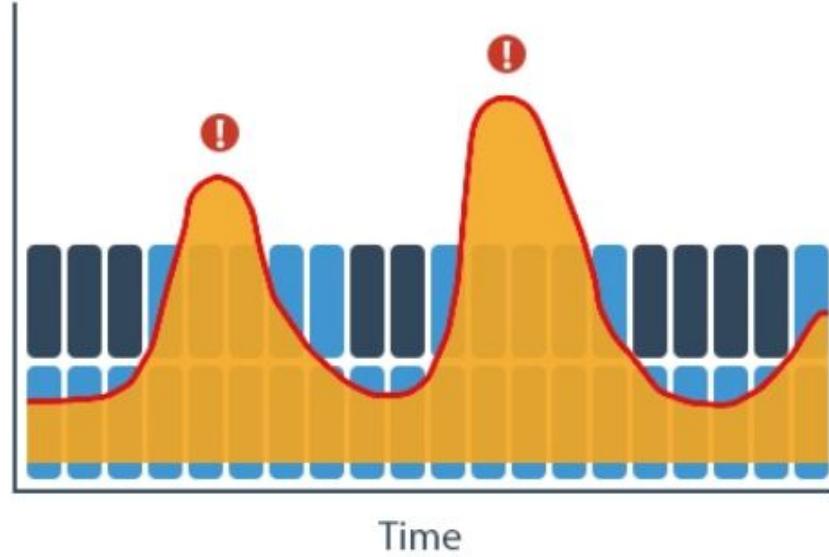
- Over-provisioning refers to a scenario where you buy more capacity than you need.
- Under-provisioning refers to allocating fewer resources than you use.

# What Is The Purpose Of Cloud Elasticity?

Overprovisioning

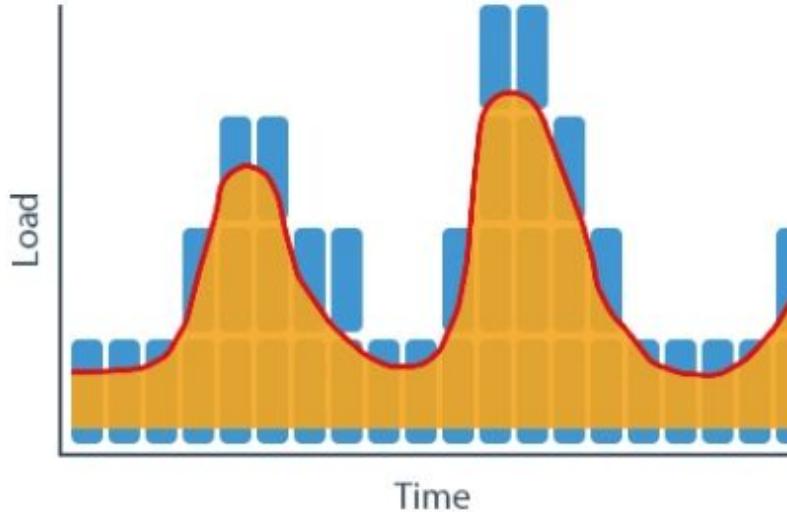
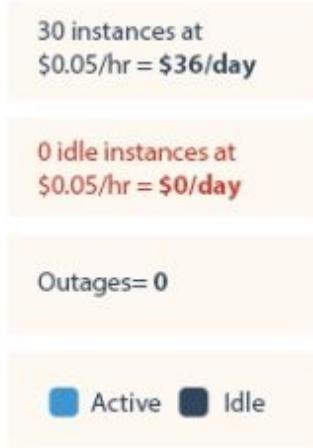


Underprovisioning



# Rapid Elasticity

## Scaling With Elasticity



# Measured Service

**Definition** : Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be monitored, controlled, and reported, providing transparency for both the provider and consumer of the utilized service.

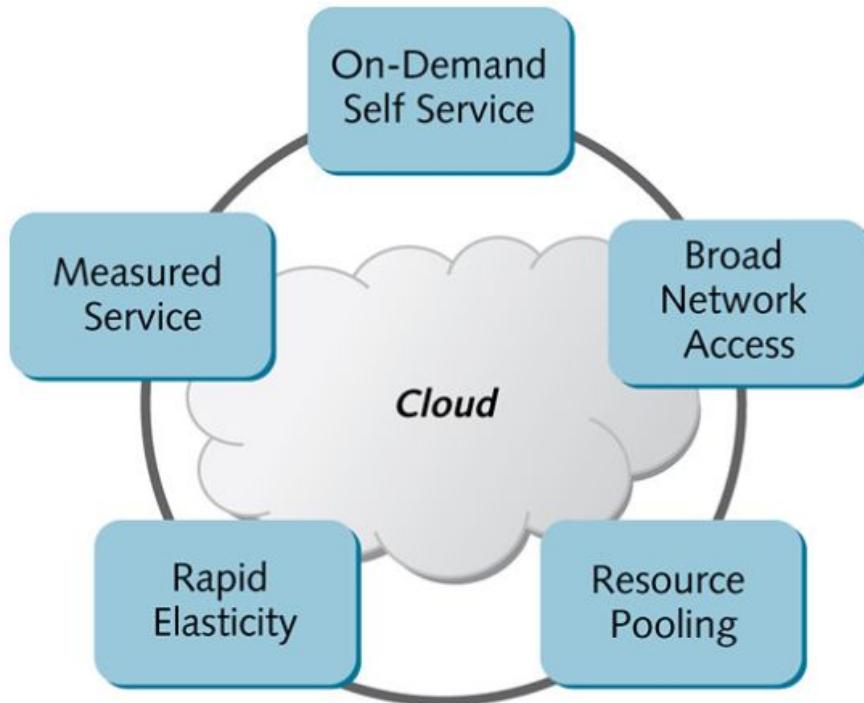
[https://csrc.nist.gov/glossary/term/measured\\_service](https://csrc.nist.gov/glossary/term/measured_service)

# Measured Service

- Cloud computing resource usage can be measured, controlled, and reported providing transparency for both the provider and consumer of the utilized service.
- Cloud computing services use a metering capability which enables to control and optimize resource use. This implies that just like air time, electricity or municipality water IT services are charged per usage metrics – pay per use. The more you utilize the higher the bill
- Cloud services generally charge users per hour of resource usage, or based on the number of certain kinds of transactions that have occurred, amount of storage in use, and the amount of data transferred over a network. All usage is measured.

# Characteristics of Cloud Computing

# Cloud Computing Characteristics



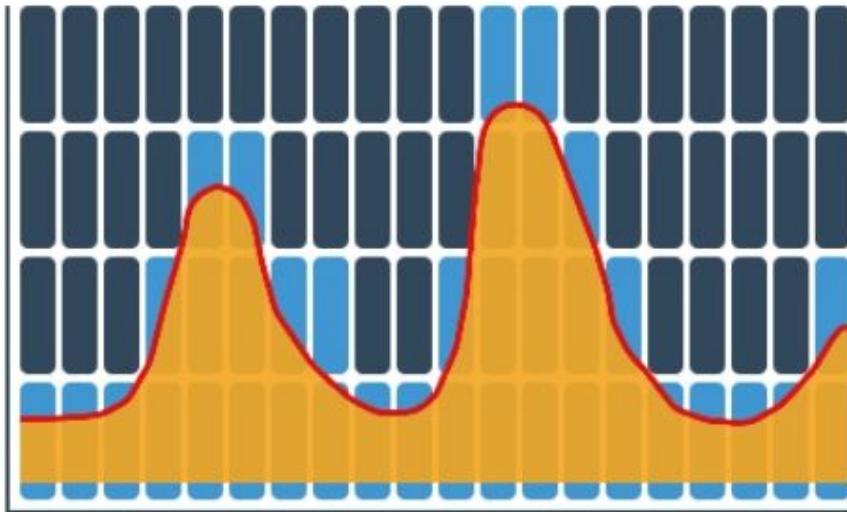
# What Is The Purpose Of Cloud Elasticity?

Cloud elasticity helps users prevent over-provisioning or under-provisioning system resources.

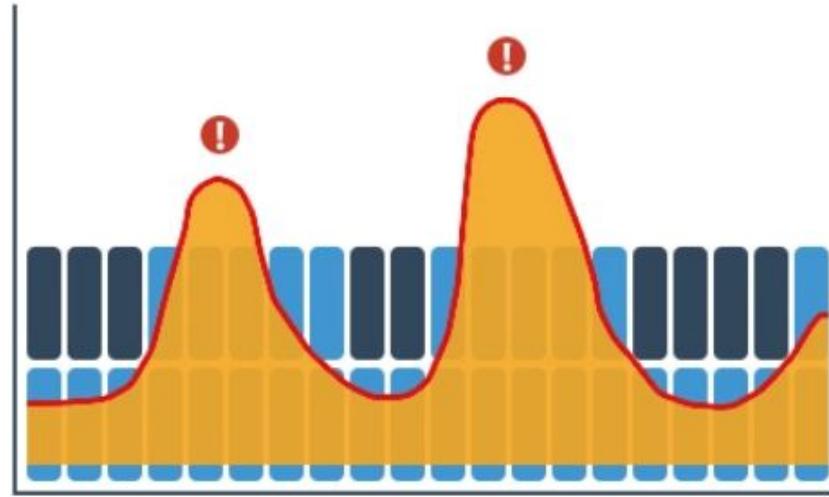
- Over-provisioning refers to a scenario where you buy more capacity than you need.
- Under-provisioning refers to allocating fewer resources than you use.

# What Is The Purpose Of Cloud Elasticity?

Overprovisioning

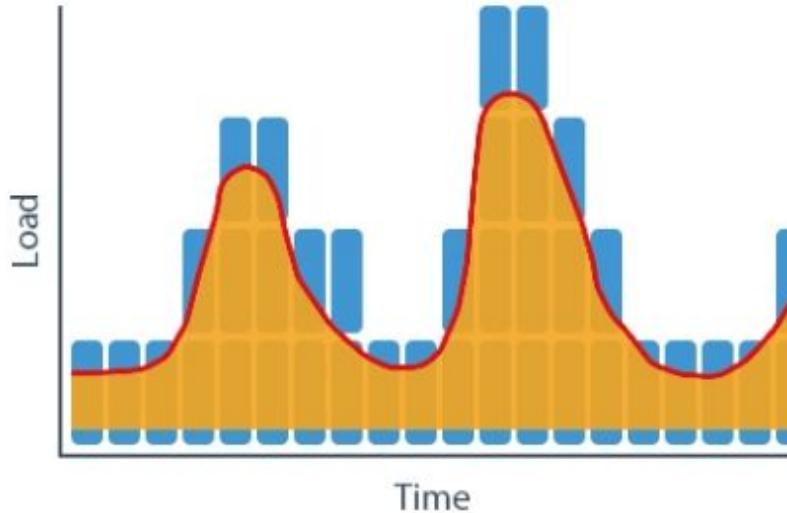


Underprovisioning



# Rapid Elasticity

## Scaling With Elasticity



# Measured Service

**Definition** : Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be monitored, controlled, and reported, providing transparency for both the provider and consumer of the utilized service.

[https://csrc.nist.gov/glossary/term/measured\\_service](https://csrc.nist.gov/glossary/term/measured_service)

# Measured Service

- Cloud computing resource usage can be measured, controlled, and reported providing transparency for both the provider and consumer of the utilized service.
- Cloud computing services use a metering capability which enables to control and optimize resource use. This implies that just like air time, electricity or municipality water IT services are charged per usage metrics – pay per use. The more you utilize the higher the bill
- Cloud services generally charge users per hour of resource usage, or based on the number of certain kinds of transactions that have occurred, amount of storage in use, and the amount of data transferred over a network. All usage is measured.

# Need For Cloud Computing

- **Flexibility :**
  - Add or remove capacity as needed, which is ideal for businesses that experience seasonal spikes in demand
  - Moving resources to meet demands
  - Getting to market faster
- **Benefits of Gaining Flexibility**
  - Optimized investments
  - Improved agility
  - Increased productivity
  - Improved security
  - Lower carbon footprint

<https://www.liquidweb.com/blog/gain-cloud-flexibility/>

# Need For Cloud Computing

- **Disaster recovery**
  - The strategies and services enterprises apply for the purpose of backing up applications, resources, and data into a cloud environment.
  - Helps protect corporate resources and ensure business continuity.
  - Recovery from backed up versions to either on-premise or cloud environments.
- **Why is Cloud Disaster Recovery Important?**
  - Outage due to power failures
  - Business continuity during network or power outages, system failures, natural disasters, accidents, cyber attacks, and during software updates.

<https://cloudian.com/guides/disaster-recovery/understanding-disaster-recovery-in-the-cloud/>

# Need For Cloud Computing

- **Automatic software updates**
  - Software updates can take hours! However, cloud computing suppliers do all server maintenance – including security updates –themselves, freeing up their customers' time and resources for other tasks.
- **Increased collaboration**
  - All the information about the assets is in the cloud and entire team of an organization access it by working together on a collaborative platform.
  - 
  - An integrated solution with a collaborative platform and an excellent user interface can save time and create a more creative environment.

# Need For Cloud Computing

- **Work from anywhere**
  - With cloud computing, if anyone got an internet connection then he/she can be at work.
  - Advantage of flexible working environment and working hour
  - Better work-life balance for employees
- **Capital-expenditure Free**
  - Cloud computing services are typically pay as you go, so there's no need for capital expenditure at all. And because cloud computing is much faster to deploy, businesses have minimal project start-up costs and predictable ongoing operating expenses.

# Need For Cloud Computing

- **Document Control**
  - Sharing files is so easy! You can easily collaborate with a co-worker, or anyone, across the room or across the globe, anytime, anywhere.
  - All files are in one central location, and everyone works off of one central copy.
  - Without the cloud, team members can only work on the document one at a time, and have to constantly send files back and forth via email, making them less efficient.

# Need For Cloud Computing

- **Security**
  - Every day thousands of devices are stolen – laptops, notebooks, cell phones –all with critical data. Given that your devices are password protected, you are likely only looking at a monetary loss of the device because all of your data and documents are still readily available in the cloud

# Need For Cloud Computing

- **Competitiveness**

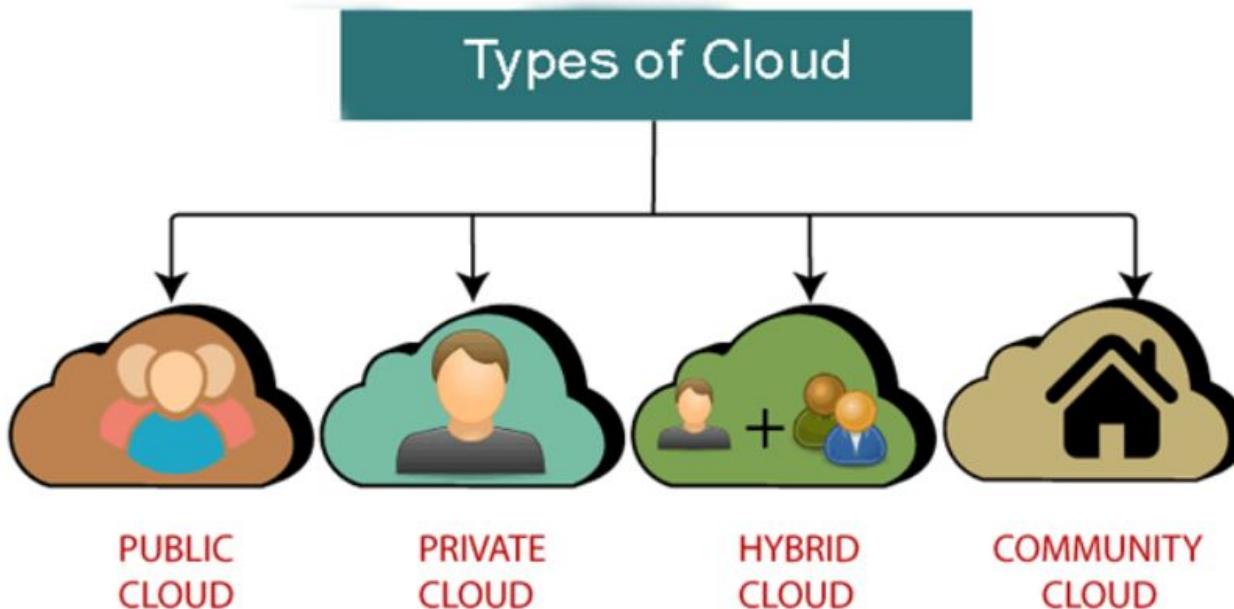
- Many larger companies have secondary data centers they can use for data backup and recovery, whereas smaller companies just back up to tape, usually.
- With the cloud, smaller companies can back up data or replicate servers to a remote site, and then fail-over the servers and network to the remote site in the event of a disaster.
- This gives smaller companies a competitive edge with larger companies

# Need For Cloud Computing

- **Environmental friendly**
  - Optimized resource utilization
  - Businesses using cloud computing only use the server space they need, which decreases their carbon footprint.

# Cloud Deployment Models

# Cloud Computing Types - Based on Deployment Models



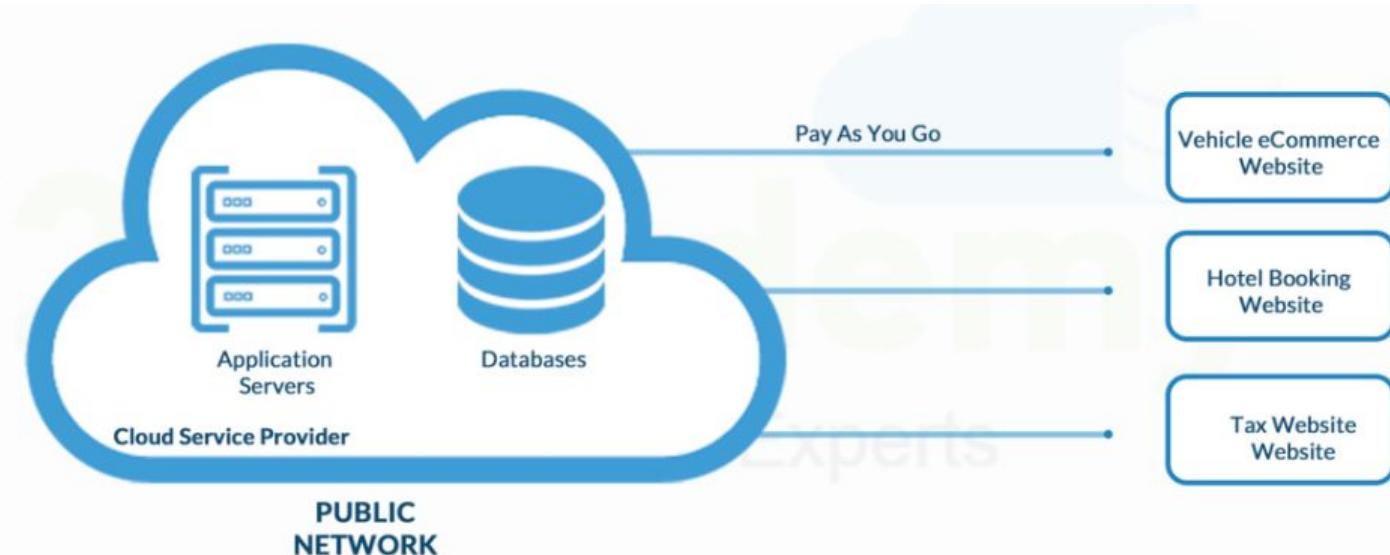
# **Cloud Computing Types - Based on Deployment Models**

- NIST defines four cloud deployment models:
  - Public clouds
  - Private clouds
  - Community clouds
  - Hybrid clouds
- A cloud deployment model is defined according to where the infrastructure for the deployment resides and who has control over that infrastructure.

# **Cloud Computing Types - Based on Deployment Models**

- Deciding which deployment model you will go with is one of the most important cloud deployment decisions you will make.
- Each cloud deployment model satisfies different organizational needs, so it's important that you choose a model that will satisfy the needs of your organization.

# Public Cloud



# Public Cloud

- Public cloud is open to all to store and access information via the Internet using the pay-per-usage method.
- In public cloud, computing resources are managed and operated by the Cloud Service Provider (CSP).
- **Example :**
  - Amazon elastic compute cloud (EC2)
  - IBM SmartCloud Enterprise
  - Microsoft
  - Google App Engine
  - Windows Azure Services Platform.

# Advantages of Public Cloud

- **Low Cost** : Public cloud has a lower cost than private, or hybrid cloud, as it shares the same resources with a large number of consumers.
- **Location Independent** : Public cloud is location independent because its services are offered through the internet.
- **Save Time** : In Public cloud, the cloud service provider is responsible for the manage and maintain data centers in which data is stored, so the cloud user can save their time to establish connectivity, deploying new products, release product updates, configure, and assemble servers.

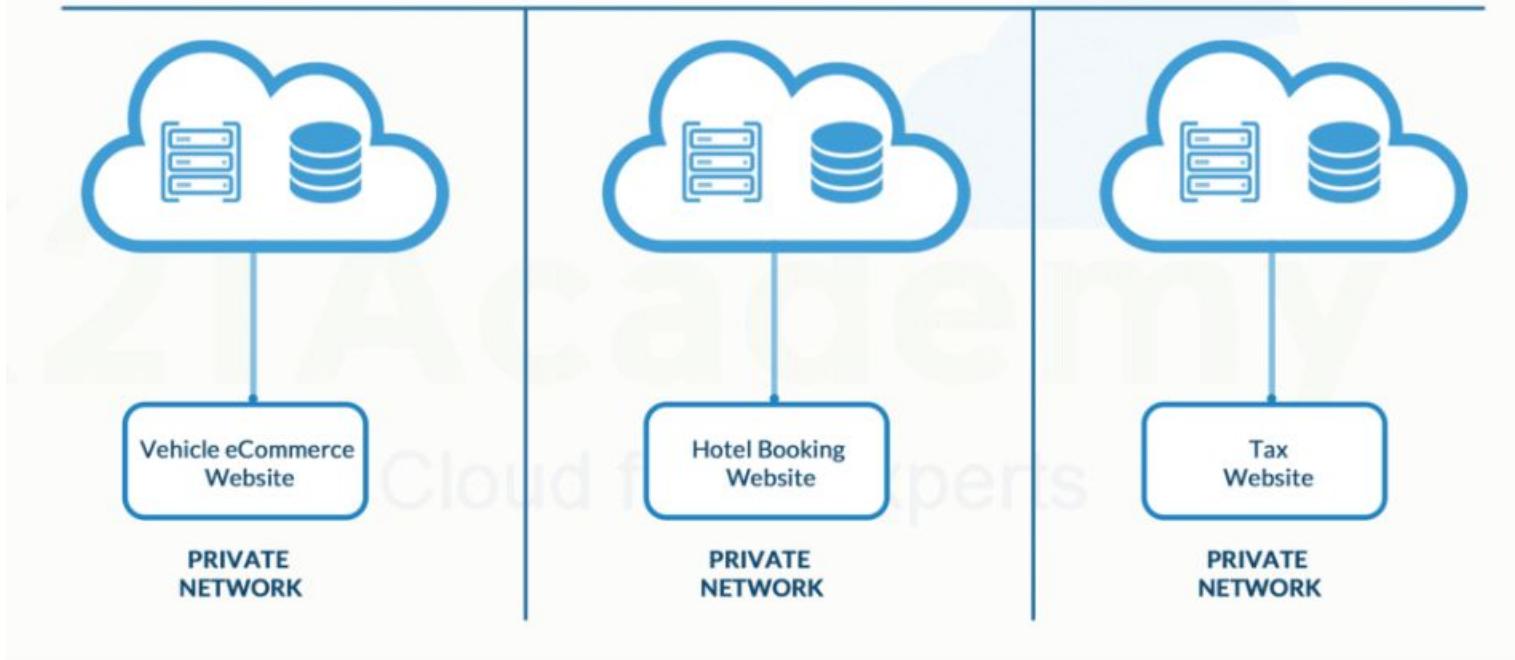
# **Advantages of Public Cloud**

- **Quickly and easily set up** : Organizations can easily buy public cloud on the internet and deploy and configure it remotely through the cloud service provider within a few hours.
- **Business Agility** : Public cloud provides an ability to elastically re-size computer resources based on the organization's requirements.
- **Scalability and reliability** : Public cloud offers scalable and reliable (24\*7 available) services to the users at an affordable cost.

# **Disadvantages of Public Cloud**

- **Low Security** : Public Cloud is less secure because resources are shared publicly.
- **Performance** : In the public cloud, performance depends upon the speed of internet connectivity.
- **Less customizable** : Public cloud is less customizable than the private cloud.

# Private Cloud



# Private Cloud

- Private cloud is also known as an **internal cloud** or **corporate cloud**.
- Private cloud provides computing services to a **private internal network (within the organization)** and **selected users** instead of the general public.
- Private cloud provides a **high level of security** and **privacy** to data through firewalls and internal hosting. It also ensures that operational and sensitive data are not accessible to third-party providers.

# Private Cloud

- **Example :**
  - HP Data Centers
  - Microsoft
  - Elastra-private cloud
  - Ubuntu

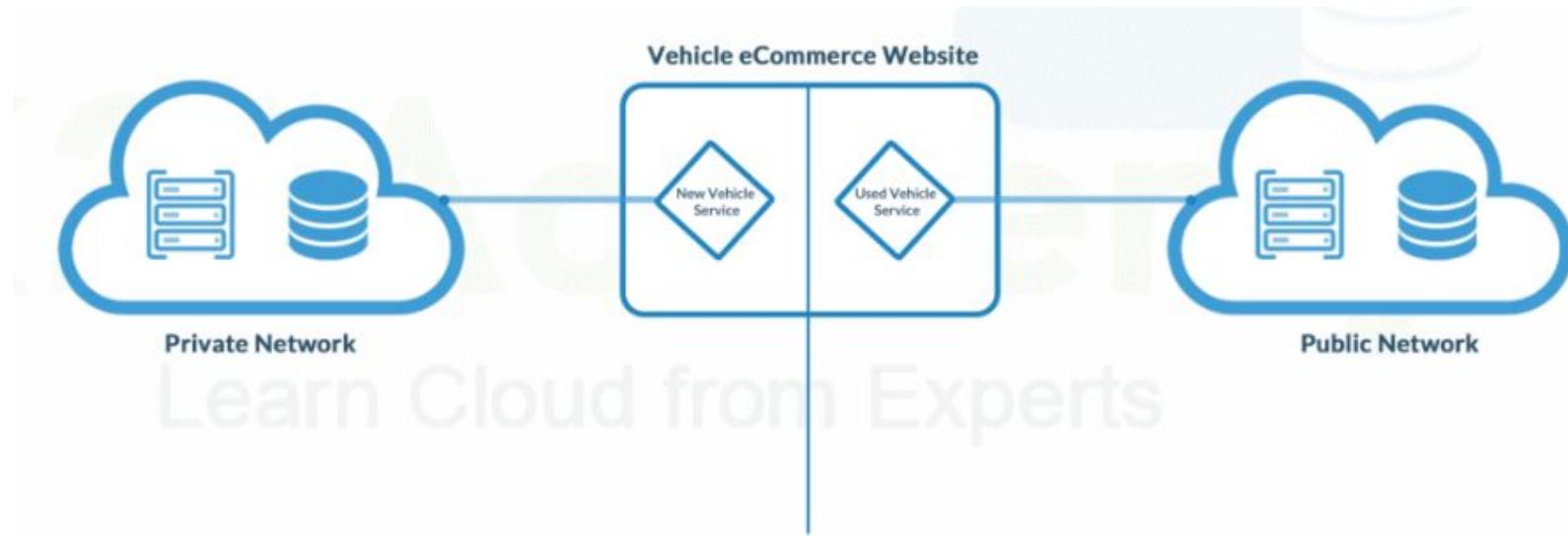
# Advantages of Private Cloud

- **More Control** : Private clouds have more control over their resources and hardware than public clouds because it is only accessed by selected users.
- **Security & privacy** : Security & privacy are one of the big advantages of cloud computing. Private cloud improved the security level as compared to the public cloud.
- **Improved performance** : Private cloud offers better performance with improved speed and space capacity.

# **Disadvantages of Private Cloud**

- **High cost** : The cost is higher than a public cloud because set up and maintain hardware resources are costly.
- **Restricted area of operations** : As we know, private cloud is accessible within the organization, so the area of operations is limited.
- **Limited scalability** : Private clouds are scaled only within the capacity of internal hosted resources.
- **Skilled people** : Skilled people are required to manage and operate cloud services.

# Hybrid Cloud



# Hybrid Cloud

- Hybrid cloud = public cloud + private cloud
- The main aim to combine these cloud (Public and Private) is to create a unified, automated, and well-managed computing environment.
- In the Hybrid cloud, **non-critical activities** are performed by the **public cloud** and **critical activities** are performed by the **private cloud**.

# Hybrid Cloud

- Mainly, a hybrid cloud is used in finance, healthcare, and Universities.
- **Example :**
  - Amazon
  - Microsoft
  - Google
  - Cisco
  - NetApp

# Advantages of Hybrid Cloud

- **Flexible and secure :** It provides flexible resources because of the public cloud and secure resources because of the private cloud.
- **Cost effective :**
  - Hybrid cloud costs less than the private cloud. It helps organizations to save costs for both infrastructure and application support.
  - It offers the features of both the public as well as the private cloud. A hybrid cloud is capable of adapting to the demands that each company needs for space, memory, and system.

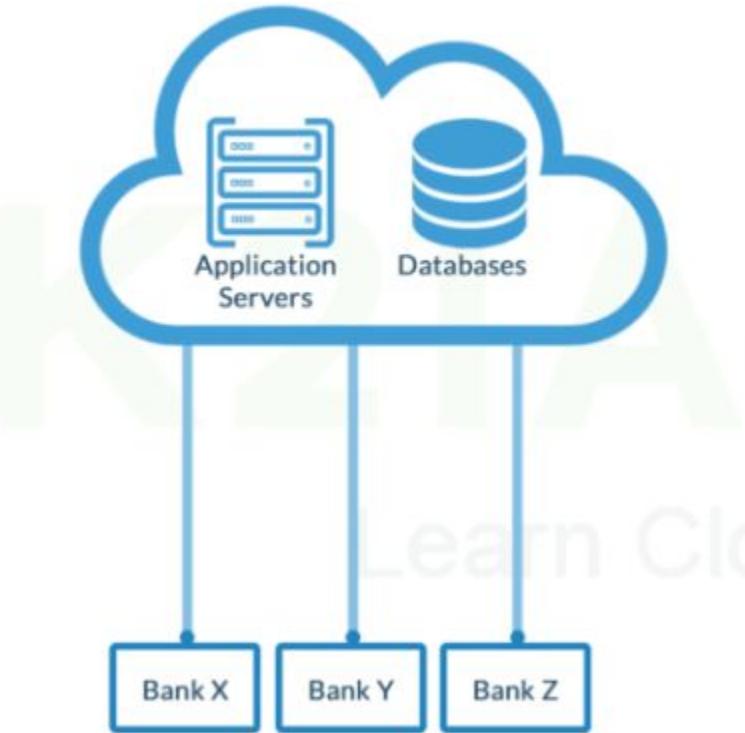
# **Advantages of Hybrid Cloud**

- **Security** : Hybrid cloud is secure because critical activities are performed by the private cloud.
- **Risk Management** : Hybrid cloud provides an excellent way for companies to manage the risk.

# **Disadvantages of Hybrid Cloud**

- **Networking issues** : In the Hybrid Cloud, networking becomes complex because of the private and the public cloud.
- **Infrastructure Compatibility** : Infrastructure compatibility is the major issue in a hybrid cloud. With dual-levels of infrastructure, a private cloud controls the company, and a public cloud does not, so there is a possibility that they are running in separate stacks.
- **Reliability** : The reliability of the services depends on cloud service providers.

# Community Cloud



<https://k21academy.com/cloud-blogs/cloud-computing-deployment-models/>

# Community Cloud

- Community cloud is a cloud infrastructure that allows systems and services to be accessible by a group of several organizations to share the information. It is owned, managed, and operated by one or more organizations in the community, a third party, or a combination of them.
- **Example :** Our government organization within India may share computing infrastructure in the cloud to manage data.

# **Advantages of Community Cloud**

- **Cost effective** : Community cloud is cost effective because the whole cloud is shared between several organizations or a community.
- **Flexible and Scalable** : The community cloud is flexible and scalable because it is compatible with every user. It allows the users to modify the documents as per their needs and requirement.
- **Security** : Community cloud is more secure than the public cloud but less secure than the private cloud.

# **Advantages of Community Cloud**

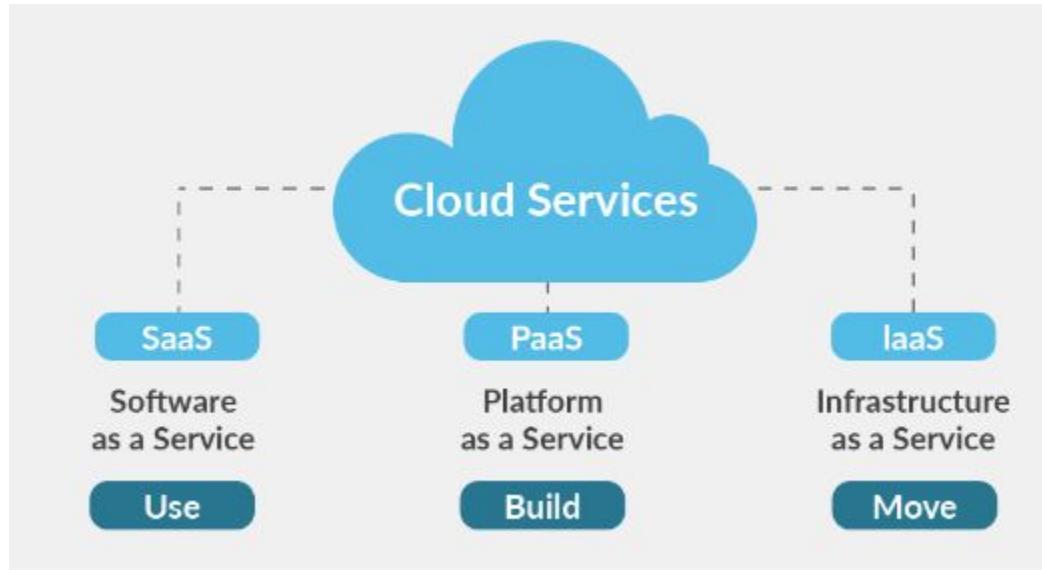
- **Sharing infrastructure** : Community cloud allows us to share cloud resources, infrastructure, and other capabilities among various organization

# Disadvantages of Community Cloud

- Community cloud is not a good choice for every organization.
- Slow adoption to data
- The fixed amount of data storage and bandwidth is shared among all community members.
- Community Cloud is costly than the public cloud.
- Sharing responsibilities among organizations is difficult.

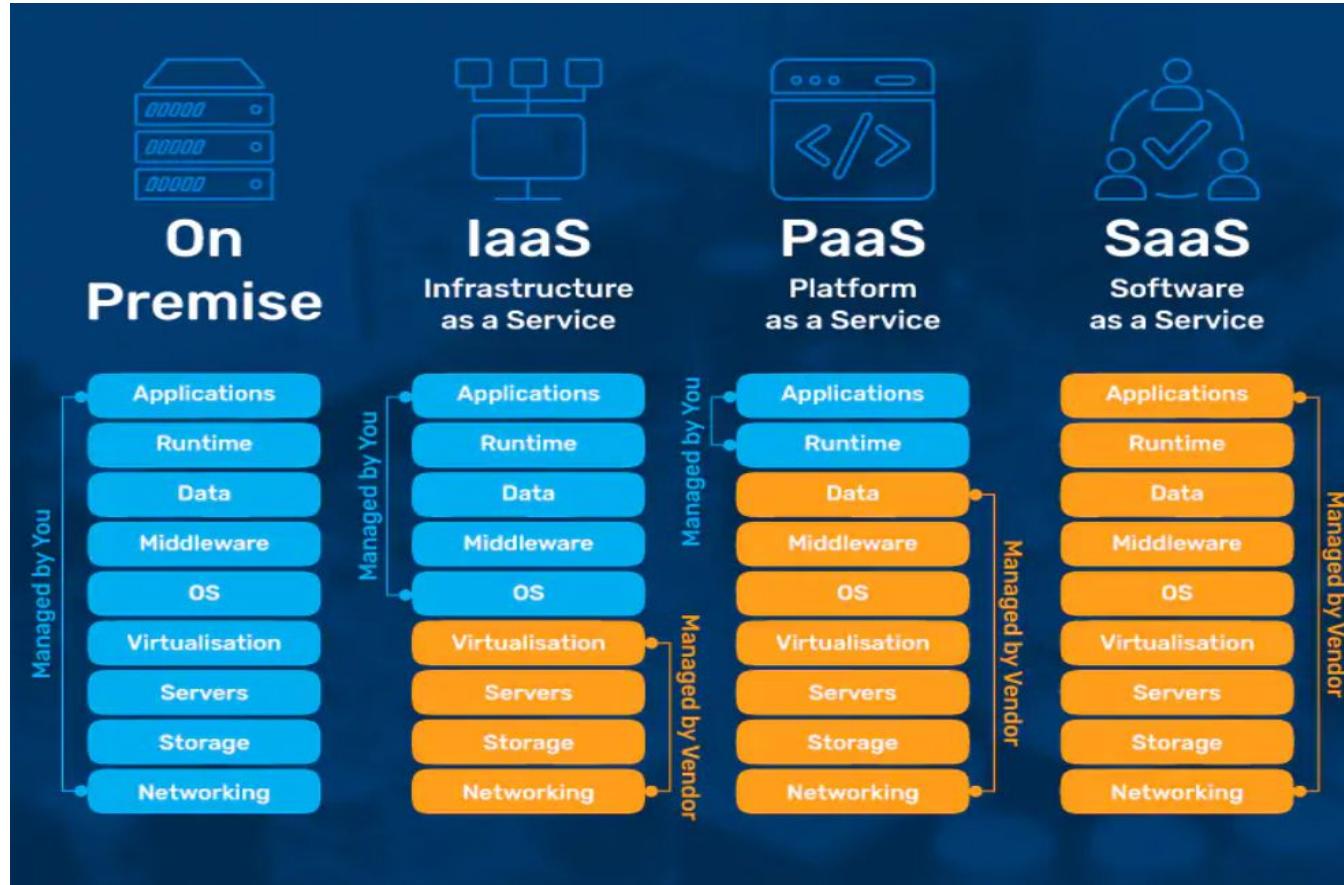
# Cloud Service Models

# Cloud Computing Types - Based on Service Models



<https://www.plesk.com/blog/various/iaas-vs-paas-vs-saas-various-cloud-service-models-compared/>

# Cloud Computing Types - Based on Service Models



# **Infrastructure as a Service**

The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, and deployed applications; and possibly limited control of select networking components (e.g., host firewalls).

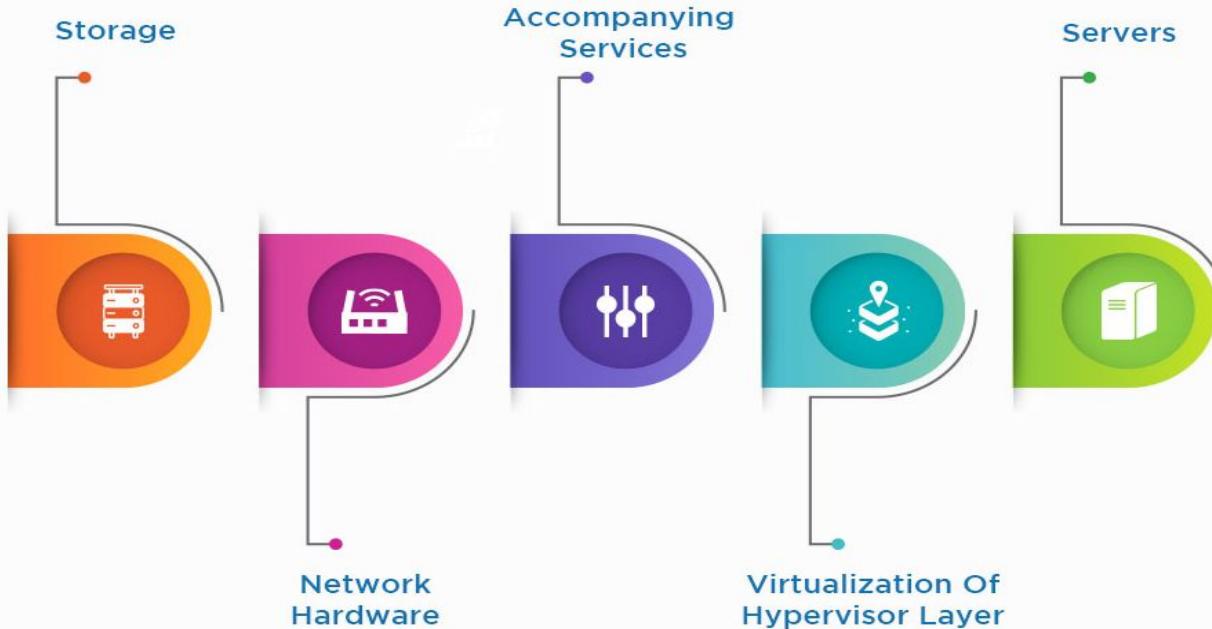
[https://csrc.nist.gov/glossary/term/infrastructure\\_as\\_a\\_service](https://csrc.nist.gov/glossary/term/infrastructure_as_a_service)

# Infrastructure as a Service



# Infrastructure as a Service

## IAAS INFRASTRUCTURE PRIMARY COMPONENTS



# Why Infrastructure as a Service Important ?

- Reducing your IT expenditure.
- Heavy up-front investment to maintain on-premises that handle only occasionally high workloads.

<https://aws.amazon.com/what-is/iaas/>

# Infrastructure as a Service

IaaS provides access to the following fundamental resources via **server virtualization** such as:

- Physical machines,
- Virtual machines,
- Virtual storage,
- Rent processing, storage,
- Network capacity and computing resources,
- Firewall, load balance, etc.
- Load Balancer
- IP Addresses
- Software Bundles
- Data Center Space
- Network Component

# **Infrastructure as a Service**

## IaaS Examples

- Amazon Web Services (AWS)
- Digital Ocean
- Microsoft Azure
- Google Compute Engine

# Infrastructure as a Service

IaaS provider responsibilities include:

- Creation, maintenance, and management of data center infrastructure
- Offer storage space and computational power to the user
- Offering maintenance-free databases, servers, and network structures
- Establishment of a continuous **virtualized** environment for consumer use
- Creating probable solution platforms to provide consumers with easy access, administration, and control over their individual component-rich infrastructure.

<https://www.filecloud.com/blog/2020/03/what-is-iaas-infrastructure/#.Y-sASHZBxPY>

# Infrastructure as a Service

The responsibilities of the user include:

- Choosing and giving a structure to strategize the virtualized environment
- Install, set up, manage, and update the application software and OS according to personalized IT components.
- Firewall configuration
- IaaS network operation
- Security integration to protect OS and installed applications
- Data security measures including data encryption to prevent data theft
- Implementing access and identity controls along with authentication strategies

<https://www.filecloud.com/blog/2020/03/what-is-iaas-infrastructure/#.Y-sASHZBxPY>

# **Infrastructure as a Service : Benefits**

- **Pay-per-use and cost-efficiency :**
  - IaaS service can be used on-demand and its components can be broken down apiece to serve unique functions for a business.
  - This reduces capital investment and induces cost efficiency.
  - It also reduces the maintenance budget since it is a high-functioning borrowed service.
  - Organizations can hence concentrate more on their business strategies and less on sustainable IT infrastructure.

<https://www.filecloud.com/blog/2020/03/what-is-iaas-infrastructure/#.Y-sASHZBxPY>

# Infrastructure as a Service : Benefits

- **Scalability :**
  - IaaS service infrastructure works to ensure authorized accessibility is always available from everywhere; evolving and updating with the need of the consumer.
  - Hence, delays are less often in capacity expansion scenarios and wastage of unused capacity is minimized.
  - Accommodating the needs of the client is of the highest priority for IaaS providers.

<https://www.filecloud.com/blog/2020/03/what-is-iaas-infrastructure/#.Y-sASHZBxPY>

# **Infrastructure as a Service : Benefits**

- **Time saving :**
  - A user can save a lot of time by choosing IaaS services because any upgrade or maintenance to the virtual or physical framework is automated through the service.
  - Saving time increases productivity, which increases profitability.
- **Rapid Innovation :**
  - A user can make quick decisions regarding new launches.
  - A new product, initiative, or program always has the necessary backup in the background for a quick start.
  - Instead of taking days to months, the computing infrastructure can be built in minutes to hours.

<https://www.filecloud.com/blog/2020/03/what-is-iaas-infrastructure/#.Y-sASHZBxPY>

# **Infrastructure as a Service : Benefits**

- **Independence of location :**
  - Authorized consumers can access the IaaS environment from anywhere which increases productivity.
  - Any urgent update can be made from anywhere in the world through the internet.
- **Quicker time to market :**
  - IaaS infrastructure can give businesses an edge over their competitors by providing faster time to market advantages with the help of flexibility and scalability.

<https://www.filecloud.com/blog/2020/03/what-is-iaas-infrastructure/#.Y-sASHZBxPY>

# **Infrastructure as a Service : Benefits**

- **Sturdy service :**
  - If any of the hardware resources fail to perform, the service will remain unaffected. Therefore, there is no single point of failure.
- **Enhanced security :**
  - Appropriate accessibility permission, service agreement, and integrated encryption ensure that the IaaS service provides a better security platform than in-house security measures.

# Infrastructure as a Service : Disadvantages

- **Control over security :**
  - While enhanced security features are part of the IaaS infrastructure, the user does not have any control over cloud security.
  - Thus reviewing the cloud service provider's service level agreement or SLA is of utmost importance.
- **Lack of flexibility :**
  - While service providers manage and maintain the software, they do not always upgrade the software.
  - This can cause abrupt crashes and productivity loss.
  - Therefore, the creation of a clear agreement with the service provider is important.

<https://www.filecloud.com/blog/2020/03/what-is-iaas-infrastructure/#.Y-sASHZBxPY>

# Infrastructure as a Service : Disadvantages

- **Technical problems :**
  - Constant dependency on the IaaS means constant dependency on the service provider.
  - Downtime, however little, can be devastating for such a business.
  - Any website can crash at any given moment, and since the user has no control over the service, there is no way to manually fix it.
- **Over Dependency :**
  - Full dependency on the provider or third party for your data

<https://www.filecloud.com/blog/2020/03/what-is-iaas-infrastructure/#.Y-sASHZBxPY>

# **Infrastructure as a Service : Disadvantages**

- **Upgrade & Maintenance :**
  - The organization is solely responsible for any upgrades of software and maintenance of tools or data system.
- **Others :**
  - Changing the provider is very much complicated
  - Possible privacy issues due to the provider's server locations

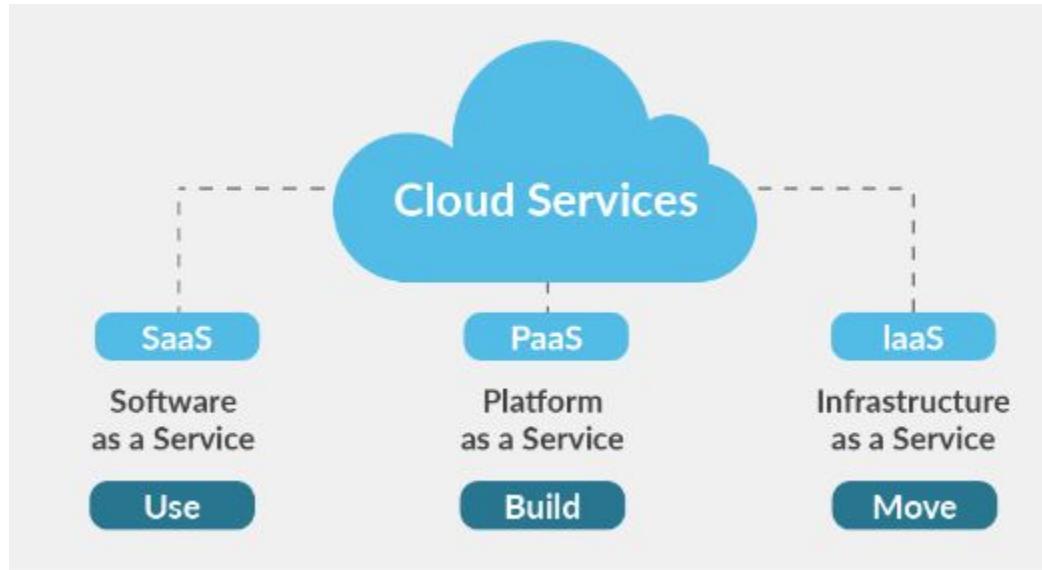
# Infrastructure as a Service : Disadvantages

- Examples of enterprises that use IaaS infrastructure
  - Netflix
  - HP
  - Ericsson
  - Xerox

<https://www.filecloud.com/blog/2020/03/what-is-iaas-infrastructure/#.Y-sASHZBxPY>

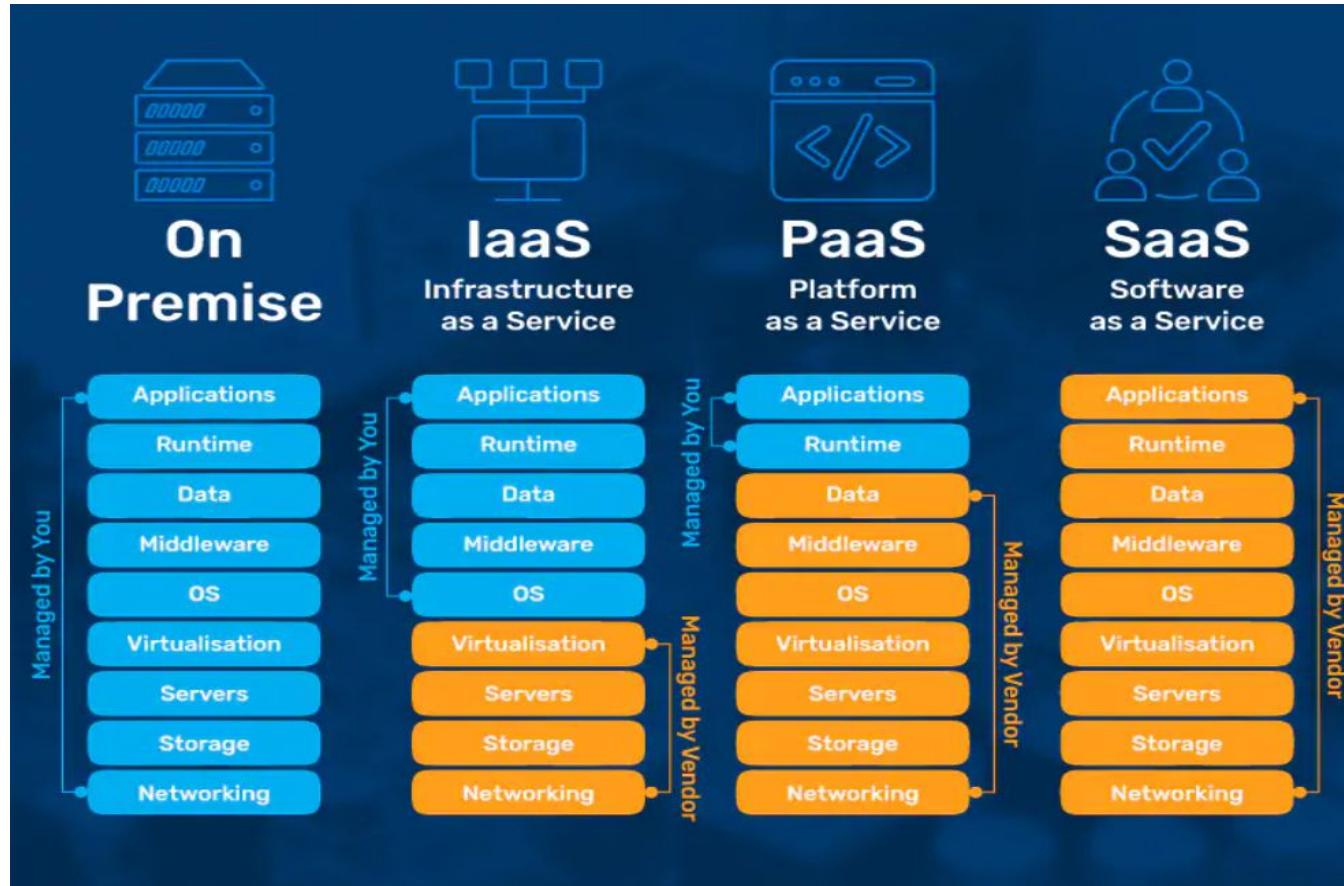
# Cloud Service Models

# Cloud Computing Types - Based on Service Models



<https://www.plesk.com/blog/various/iaas-vs-paas-vs-saas-various-cloud-service-models-compared/>

# Cloud Computing Types - Based on Service Models



# **Platform as a Service**

The capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages, libraries, services, and tools supported by the provider. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly configuration settings for the application-hosting environment.

# Platform as a Service



<https://blog.iron.io/what-is-platform-as-a-service/>

# Why Platform as a Service?

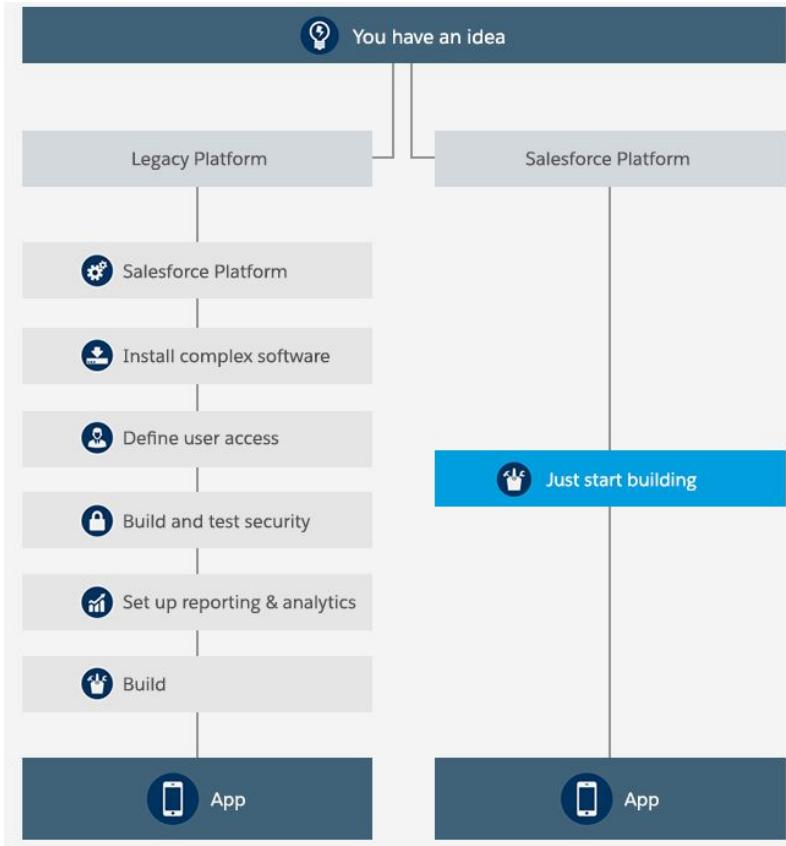
Platform as a Service eliminates the expense and complexity of :

- Evaluating
- Buying
- Configuring
- Managing

all the hardware and software needed for custom-built applications.

<https://www.salesforce.com/in/learning-centre/tech/paas/>

# Why Platform as a Service?



<https://www.salesforce.com/in/learning-centre/tech/paas/>

# Technical Advantages of Platform as a Service

- **Speed :**
  - App development is faster because your IT and developers are no longer responsible for the hardware and software used to build, maintain and protect your application development platform.
  - Once you have signed up to PaaS, you can start using the system straight away – no setup time lag.
  - For developers, being able to access tools, templates, code libraries and build packs can also reduce the time to release.

<https://www.salesforce.com/in/learning-centre/tech/paas/>

# Technical Advantages of Platform as a Service

- **Cost :**
  - Upfront costs are reduced, since there's no need to build anything before you begin developing.
  - You don't have to keep reinventing the wheel each time you build a new app, which cuts your development costs.
  - Capital costs - money tied up in costly and fast depreciating IT assets - are removed in favour of operational costs that are offset against business ROI.

# **Technical Advantages of Platform as a Service**

- **Scale :**
  - Designing apps for millions of connected devices creates potential scalability and security challenges.
- **Develop for multiple platforms—including mobile—more easily :**
  - Some service providers give you development options for multiple platforms, such as computers, mobile devices, and browsers making cross-platform apps quicker and easier to develop.
- **Use sophisticated tools affordably.**
- **Support geographically distributed development teams.**
- **Efficiently manage the application lifecycle.**

<https://www.salesforce.com/in/learning-centre/tech/paas/>

<https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-paas/>

# Business Advantages of Platform as a Service

- **Easy integration with legacy systems :**
  - You can use apps that incorporate data from your existing systems like SAP, Oracle and Microsoft.
  - It can help you unlock and modernise back-office systems with point-and-click simplicity.
- **Real-time information :**
  - You can create apps that deliver real-time data and updates to employees and managers, allowing them to make better business decisions.
  - You can create apps to help with workflow and approval processes too.

<https://www.salesforce.com/in/learning-centre/tech/paas/>

# Business Advantages of Platform as a Service

- **Easier IT maintenance :**
  - The vendor looks after the platform, so you just have to look after the apps themselves – reducing your IT overheads
- **Shared insights :**
  - With thousands of businesses using the platform, large-scale PaaS providers are quick to respond to user needs and solve common issues swiftly.
  - That means you can quickly benefit from tried and trusted solutions.

<https://www.salesforce.com/in/learning-centre/tech/paas/>

# Disadvantages of Platform as a Service

- **Vendor Dependency:** Very dependent upon the vendor's capabilities
- **Risk of Lock-In:** Customers may get locked into a language, interface or program they no longer need
- **Compatibility:** Difficulties may arise if PaaS is used in conjunction with existing development platforms
- **Security Risks:** While PaaS providers secure the infrastructure and platform, businesses are responsible for security of the applications they build

<https://www.comptia.org/content/articles/what-is-paas>

# Common Platform as a Service Scenarios

- **Development framework :**
  - PaaS provides a framework that developers can build upon to develop or customize cloud-based applications.
  - PaaS lets developers create applications using built-in software components.
  - Cloud features such as scalability, high-availability, and multi-tenant capability are included, reducing the amount of coding that developers must do.

# Common Platform as a Service Scenarios

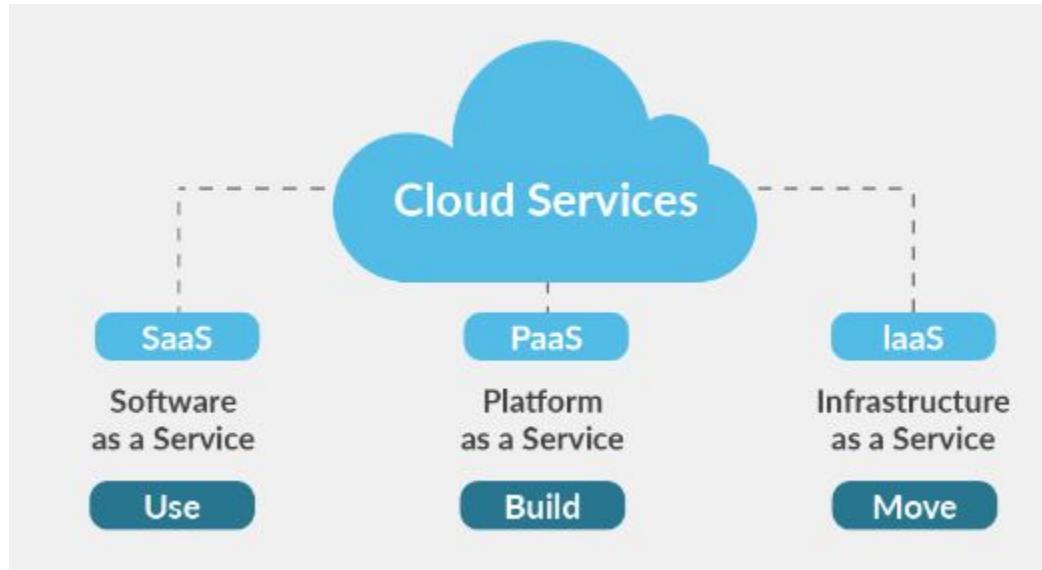
- **Analytics or business intelligence :**
  - Tools provided as a service with PaaS allow organizations to analyze and mine their data, finding insights and patterns and predicting outcomes to improve forecasting, product design decisions, investment returns, and other business decisions.
- **Additional services :**
  - PaaS providers may offer other services that enhance applications, such as workflow, directory, security, and scheduling.

# Common Platform as a Service Scenarios

- **Analytics or business intelligence :**
  - Tools provided as a service with PaaS allow organizations to analyze and mine their data, finding insights and patterns and predicting outcomes to improve forecasting, product design decisions, investment returns, and other business decisions.
- **Additional services :**
  - PaaS providers may offer other services that enhance applications, such as workflow, directory, security, and scheduling.

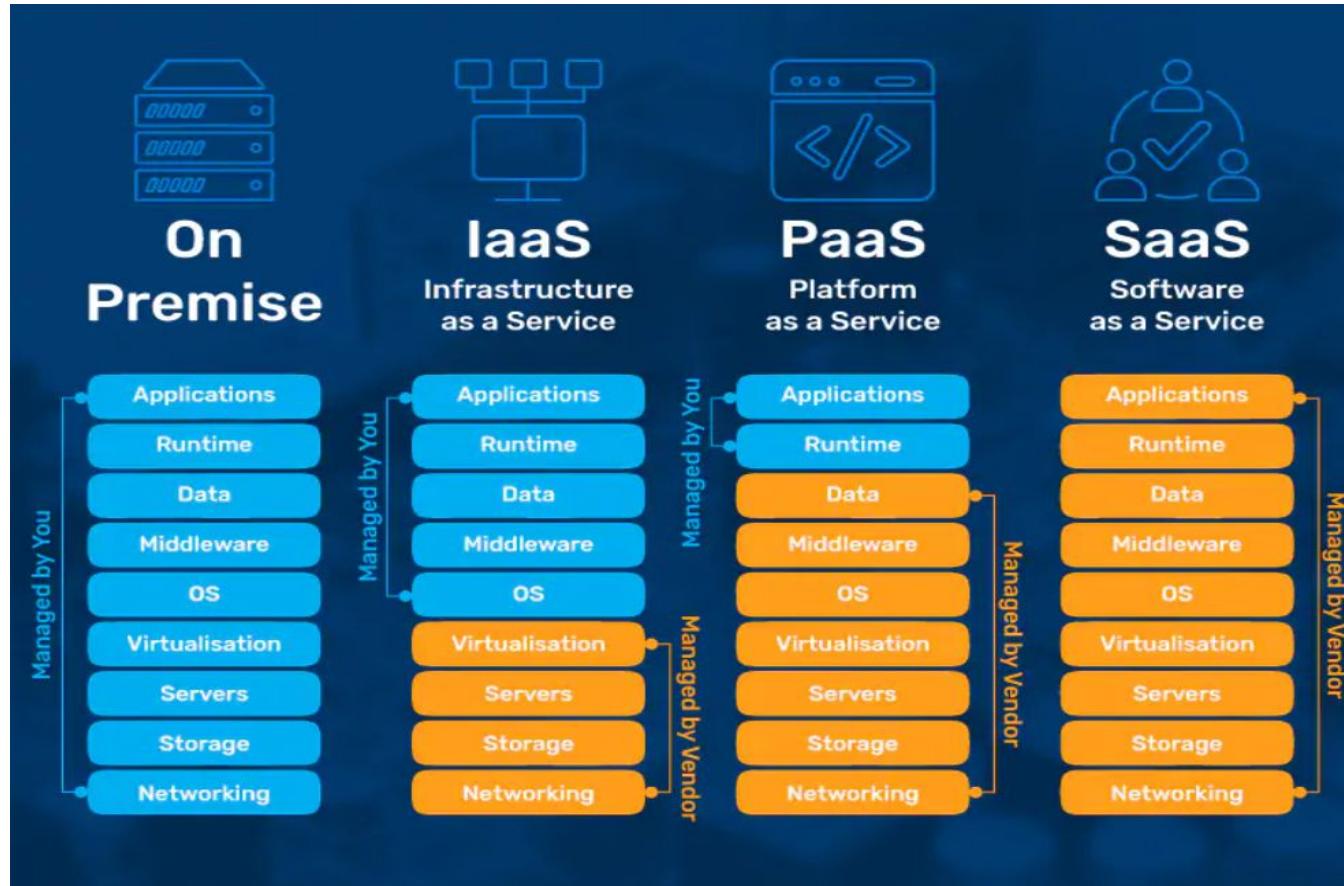
# Cloud Service Models

# Cloud Computing Types - Based on Service Models



<https://www.plesk.com/blog/various/iaas-vs-paas-vs-saas-various-cloud-service-models-compared/>

# Cloud Computing Types - Based on Service Models



# **Software as a Service**

The capability provided to the consumer is to use the provider's applications running on a cloud infrastructure. The applications are accessible from various client devices through either a thin client interface, such as a web browser (e.g., web-based email), or a program interface. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user-specific application configuration settings.

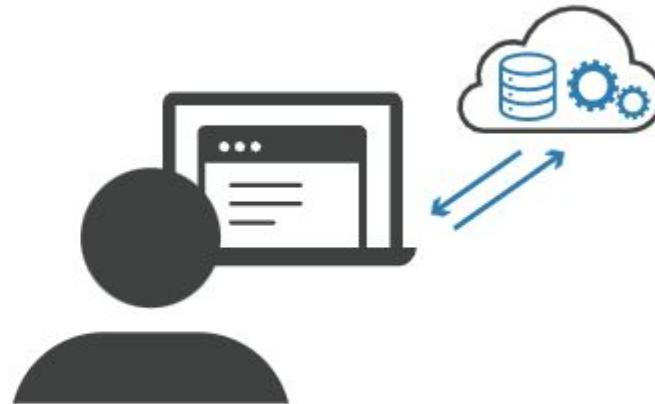
# Software as a Service

Non-SaaS Application



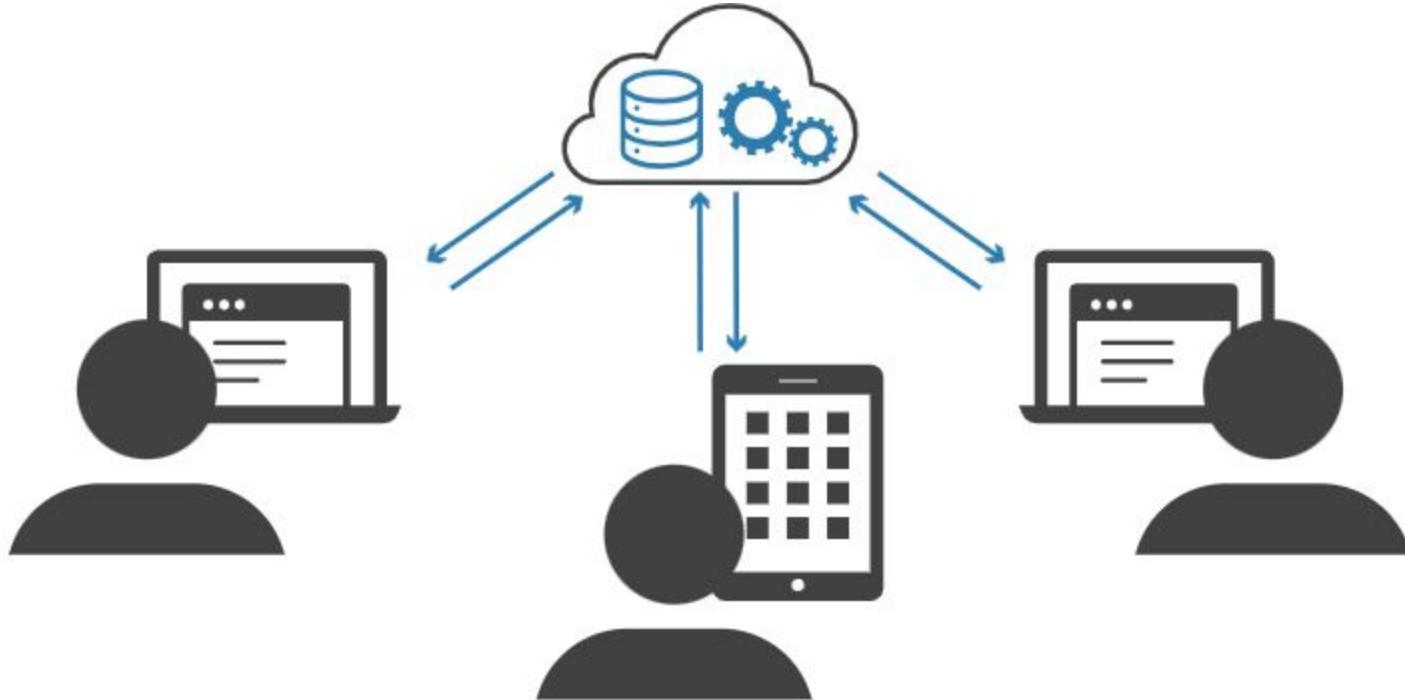
Application logic runs  
on user's computer

SaaS Application



Application logic runs  
in the cloud

# Software as a Service



Users can access SaaS applications on any device

<https://www.cloudflare.com/learning/cloud/what-is-saas/>

# **Benefits of Software as a Service**

- **Accessibility** : Ability to run via an internet browser 24/7 from any device
- **Operational Management** : No installation, equipment updates or traditional licensing management
- **Cost Effective** : No upfront hardware costs and flexible payment methods such as pay-as-you-go models
- **Scalability** : Easily scale a solution to accommodate changing needs
- **Data Storage** : Data is routinely saved in the cloud
- **Analytics** : Access to data reporting and intelligence tools
- **Increase Security** : SaaS providers invest heavily in security technology and expertise

<https://www.comptia.org/content/articles/what-is-saas>

# Challenges of Software as a Service

- **Loss of Control:** The vendor manages everything, making you dependent upon the vendor's capabilities
- **Limited Customization:** Most SaaS applications offer little in the way of customization from the vendor
- **Slower Speed:** SaaS solutions can have more latency than client/server apps
- **Security Risks:** While the SaaS provider secures the application itself, strict measures should be taken with sensitive data
- **Difficulty switching vendors**

<https://www.comptia.org/content/articles/what-is-saas>

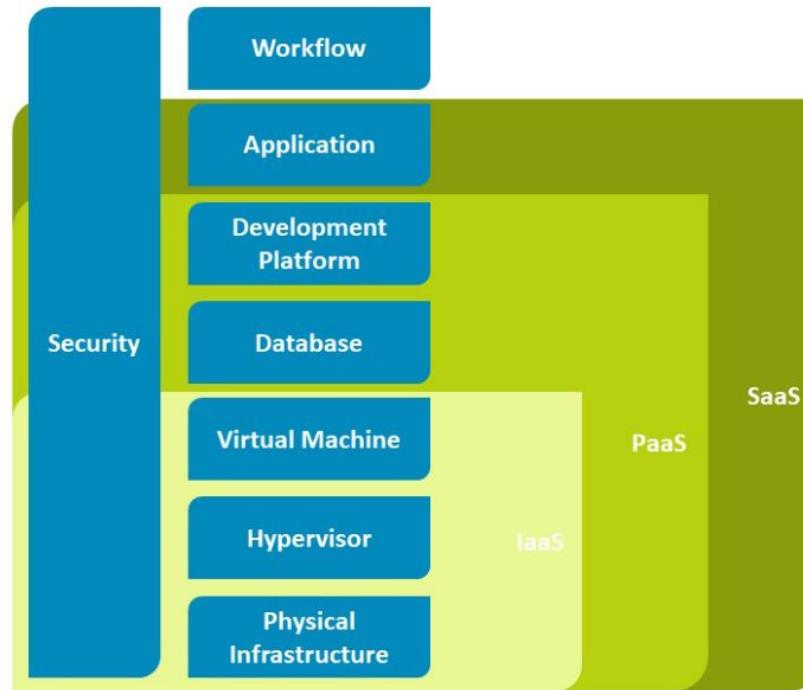
# Characteristics of Software as a Service



<https://www.qbrainx.com/blog/what-are-the-benefits-and-key-characteristics-of-saas/>

# Difference Between SaaS vs. IaaS vs. PaaS

Cloud Services Across the IT Solution Stack



<https://www.comptia.org/content/articles/what-is-saas>

# Resource as a Service

# Resource as a Service

- Instead of providers exclusively selling server equivalent virtual machines for relatively long periods of time (as done in today's IaaS clouds), providers increasingly sell individual resources (such as CPU, memory, and I/O resources) for a few seconds at a time.
- This economic model of cloud computing is called the Resource-as-a-Service (RaaS) cloud

<https://www.usenix.org/system/files/conference/hotcloud12/hotcloud12-final45.pdf>

# Why Resource as a Service?

- Now migration between cloud providers becoming easier, and competition between providers is increasing that leads the providers to economize.
- They can seek to maintain the least amount of supply (computing resources) that can profitably be sold for handling the expected demand (client requests for resources).
- When the demand exceeds the provider's expectations, it can take advantage of the different client priorities to deprive low-paying clients and sell the same resources to clients willing to pay more.
- Increased competition between providers forces them to become more efficient.
- Constant load changes render static resource allocation inefficient, because as load changes, resources go unused.
- Providers therefore seek to sell their clients the available cycles, memory, and I/O-bandwidth that the clients need, when they need it, at market-driven prices, so that no profitable resource goes unused.

<https://www.usenix.org/system/files/conference/hotcloud12/hotcloud12-final45.pdf>

# Why Resource as a Service?

- As users move more of their computing capacity to the cloud, they seek to reduce their costs.
- Since load and desired performance constantly change, clients seek to only pay for the resources they need, and only when they need them. The more flexible the provider offerings, the better control clients have over their costs and the resulting performance.
- Being able to control costs makes it possible for clients to act according to their economic incentives; these economic incentives drive clients to develop methods for efficient resource usage.

# How Resource as a Service works?

- In RaaS clouds, the client purchases upon admittance a seed virtual machine. The seed virtual machine only has a minimal initial amount of dedicated resources.
- All other resources needed for the efficient intended operation of the virtual machine are continuously rented and potentially sublet by the client at market-driven prices.
- The resources available for rent include CPU, RAM, and I/O resources, as well as emerging resources such as accelerators and memristors.
  - CPU capacity is sold on a hardware-thread basis, or even as number of cycles per unit of time;
  - RAM is sold on the basis of memory frames;
  - I/O is sold on the basis of I/O devices with associated I/O bandwidth and latency guarantees. Such devices include network interfaces, block interfaces, and possibly also accelerators such as FPGAs or GPGPUs.
- Every resource has a dynamically changing price tag attached to it.

<https://www.usenix.org/system/files/conference/hotcloud12/hotcloud12-final45.pdf>

# Implications, Challenges, Opportunities of Resource as a Service

- The RaaS cloud gives rise to a number of implications, challenges and opportunities for both providers and clients.
- The RaaS cloud requires new mechanisms for allocating, metering, charging for, reclaiming, and redistributing CPU, memory and I/O resources between untrusted, not-necessarily-cooperative clients every few seconds. These mechanisms must be efficient and reliable. In particular, they must be resistant to side-channel attacks from malicious clients.
- The RaaS cloud requires new system software and new applications.
- The RaaS cloud requires efficient methods of balancing resources within a single physical machine, while taking into consideration the different guaranteed service levels.
- RaaS cloud requires new economic models for deciding what to allocate, when to allocate it, and at what prices.

<https://www.usenix.org/system/files/conference/hotcloud12/hotcloud12-final45.pdf>

# Resource as a Service

# Resource as a Service

- The Resource-as-a-Service (RaaS) cloud is an economic model of cloud computing that allows providers to sell individual resources (such as CPU, memory, and I/O resources) for a few seconds at a time. In the RaaS cloud, clients are able to purchase exactly the resources they need when they need them.
- Instead of providers exclusively selling server equivalent virtual machines for relatively long periods of time (as done in today's IaaS clouds), providers increasingly sell individual resources (such as CPU, memory, and I/O resources) for a few seconds at a time.
- This economic model of cloud computing is called the Resource-as-a-Service (RaaS) cloud

<https://www.usenix.org/system/files/conference/hotcloud12/hotcloud12-final45.pdf>

# Transition from Infrastructure-as-a-Service

- Three trends in the construction, operation, and use of IaaS cloud computing platforms that underlie this transition
  - **Decreasing Duration of Rent :**
    - Average useful lifetime of a purchased server was approximately three years.
    - With the advent of web-hosting, clients could rent a server on a monthly basis.
    - With the introduction of on-demand EC2 instances, Amazon radically changed the time granularity of server rental, making it possible to rent a server equivalent for as little as one hour.
    - This trend is driven by economic forces that keep pushing clients to improve efficiency and minimize waste

# Transition from Infrastructure-as-a-Service

- **Decreasing Resource Granularity :**
  - In most IaaS clouds, clients rent resources as a fixed bundle of compute, memory, and I/O resources.
  - Selling resources this way provides clients with a familiar abstraction of a server equivalent.
  - Facility to add/remove different “network instances” and “block instances” from running virtual machines.
  - Renting a fixed combination of cloud resources cannot and does not reflect the interests of clients.

# Transition from Infrastructure-as-a-Service

- **Provisioning of useful service level agreements (SLAs) :**
  - Current Service Level Agreements (SLAs) sell resources, not performance
  - In usual SLAs the provider provides the client virtual machines with resources equivalent to servers of certain sizes. However, the performance of the same virtual machine, however, can vary wildly at different times, due to over-commitment, interference between virtual machines, or other reasons.
  - In practice, what clients care about is their virtual machines' subjective performance.
  - This approach is only applicable where the provider has full visibility into and cooperation from client virtual machines which is not possible all the time.
  - This situation leads to market driven approach where clients bid for resources according to their subjective valuations for those resource, thus affecting their prices.

<https://www.usenix.org/system/files/conference/hotcloud12/hotcloud12-final45.pdf>

# Transition from Infrastructure-as-a-Service

- **Provisioning of useful service level agreements (SLAs) :**
  - Having clients with different priorities is useful to the provider, since it can provide high-priority clients with elasticity and availability at the expense of lower-priority clients, while simultaneously renting out currently-spare resources to low-priority clients when high-priority clients do not need them.
  - Likewise, different priorities allow budget-constrained cloud clients cheap access to computing resources with poorer availability.
  - Allow the clients to define their own priority level—their own SLA—individually, choosing from several levels of capacity and availability which are priced accordingly.
  - This will allow the providers to simultaneously achieve high resource utilization and maintain adequate spare capacity for handling sudden loads.

# Why Resource as a Service?

- **Providers Perspective :**

- Now migration between cloud providers becoming easier, and competition between providers is increasing that leads the providers to economize.
- They can seek to maintain the least amount of supply (computing resources) that can profitably be sold for handling the expected demand (client requests for resources).
- When the demand exceeds the provider's expectations, it can take advantage of the different client priorities to deprive low-paying clients and sell the same resources to clients willing to pay more.
- Increased competition between providers forces them to become more efficient.
- Constant load changes render static resource allocation inefficient, because as load changes, resources go unused.
- Providers therefore seek to sell their clients the available cycles, memory, and I/O-bandwidth that the clients need, when they need it, at market-driven prices, so that no profitable resource goes unused.

# Why Resource as a Service?

- **Client Perspective :**
  - As users move more of their computing capacity to the cloud, they seek to reduce their costs.
  - Since load and desired performance constantly change, clients seek to only pay for the resources they need, and only when they need them. The more flexible the provider offerings, the better control clients have over their costs and the resulting performance.
  - Being able to control costs makes it possible for clients to act according to their economic incentives; these economic incentives drive clients to develop methods for efficient resource usage.

<https://www.usenix.org/system/files/conference/hotcloud12/hotcloud12-final45.pdf>

# How Resource as a Service works?

- In RaaS clouds, the client purchases upon admittance a seed virtual machine. The seed virtual machine only has a minimal initial amount of dedicated resources.
- All other resources needed for the efficient intended operation of the virtual machine are continuously rented and potentially sublet by the client at market-driven prices.
- The resources available for rent include CPU, RAM, and I/O resources, as well as emerging resources such as accelerators and memristors.
  - CPU capacity is sold on a hardware-thread basis, or even as number of cycles per unit of time;
  - RAM is sold on the basis of memory frames;
  - I/O is sold on the basis of I/O devices with associated I/O bandwidth and latency guarantees. Such devices include network interfaces, block interfaces, and possibly also accelerators such as FPGAs or GPGPUs.
- Every resource has a dynamically changing price tag attached to it.

<https://www.usenix.org/system/files/conference/hotcloud12/hotcloud12-final45.pdf>

# Implications, Challenges, Opportunities of Resource as a Service

- The RaaS cloud gives rise to a number of implications, challenges and opportunities for both providers and clients.
- The RaaS cloud requires new mechanisms for allocating, metering, charging for, reclaiming, and redistributing CPU, memory and I/O resources between untrusted, not-necessarily-cooperative clients every few seconds. These mechanisms must be efficient and reliable. In particular, they must be resistant to side-channel attacks from malicious clients.
- The RaaS cloud requires new system software and new applications.
- The RaaS cloud requires efficient methods of balancing resources within a single physical machine, while taking into consideration the different guaranteed service levels.
- RaaS cloud requires new economic models for deciding what to allocate, when to allocate it, and at what prices.

<https://www.usenix.org/system/files/conference/hotcloud12/hotcloud12-final45.pdf>

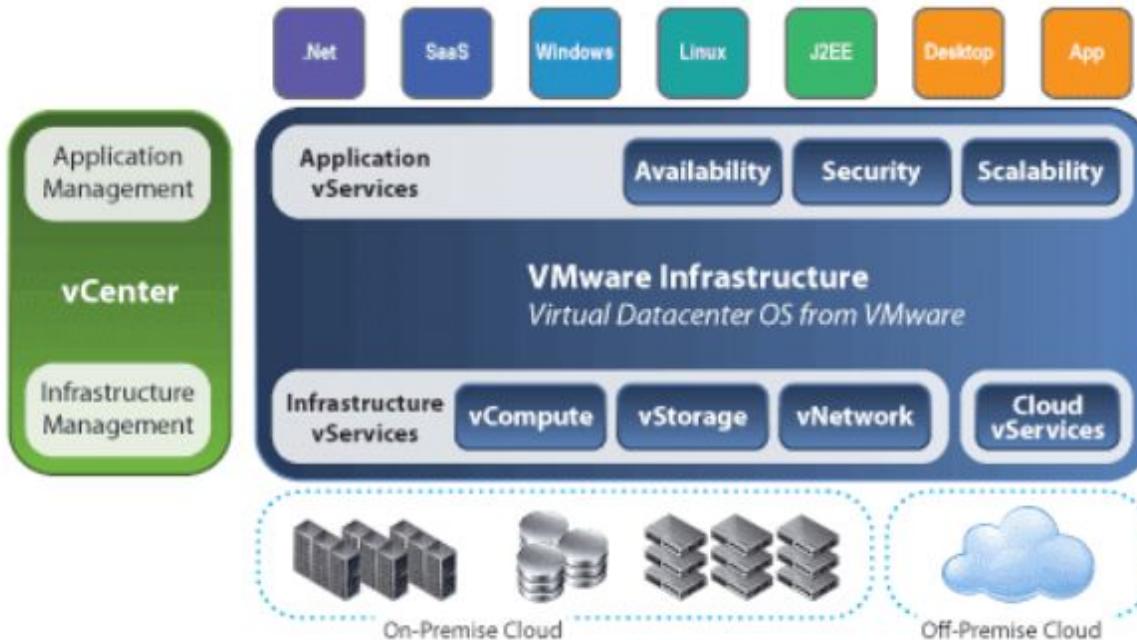
# Virtualization

# Virtualization

- Virtualization is technology that you can use to create virtual representations of servers, storage, networks, and other physical machines.
- Virtual software mimics the functions of physical hardware to run multiple virtual machines simultaneously on a single physical machine.
- Businesses use virtualization to use their hardware resources efficiently and get greater returns from their investment.
- It also powers cloud computing services that help organizations manage infrastructure more efficiently.

<https://aws.amazon.com/what-is/virtualization/>

# Virtualization



<https://aws.amazon.com/what-is/virtualization/>

# Why is virtualization important?

- By using virtualization, you can interact with any hardware resource with greater flexibility.
- Physical servers consume electricity, take up storage space, and need maintenance.
- You are often limited by physical proximity and network design if you want to access them.
- Virtualization removes all these limitations by abstracting physical hardware functionality into software.
- You can manage, maintain, and use your hardware infrastructure like an application on the web.

<https://aws.amazon.com/what-is/virtualization/>

# Basic Concepts in Virtualization

- Virtualization is a process that allows a computer to share its hardware resources with multiple digitally separated environments.
- Each virtualized environment runs within its allocated resources, such as memory, processing power, and storage.
- With virtualization, organizations can switch between different operating systems on the same server without rebooting.

<https://aws.amazon.com/what-is/virtualization/>

# Basic Concepts in Virtualization

- **Virtual Machine :**
  - A virtual machine is a software-defined computer that runs on a physical computer with a separate operating system and computing resources.
  - The physical computer is called the host machine and virtual machines are guest machines.
  - Multiple virtual machines can run on a single physical machine.
  - Virtual machines are abstracted from the computer hardware by a hypervisor.

# Basic Concepts in Virtualization

- **Hypervisor :**
  - The hypervisor is a software component that manages multiple virtual machines in a computer.
  - It ensures that each virtual machine gets the allocated resources and does not interfere with the operation of other virtual machines. There are two types of hypervisors.

# Basic Concepts in Virtualization

- **Type 1 Hypervisor :**
  - A type 1 hypervisor, or bare-metal hypervisor, is a hypervisor program installed directly on the computer's hardware instead of the operating system. Therefore, type 1 hypervisors have better performance and are commonly used by enterprise applications. KVM uses the type 1 hypervisor to host multiple virtual machines on the Linux operating system.
- **Type 2 Hypervisor :**
  - Also known as a hosted hypervisor, the type 2 hypervisor is installed on an operating system. Type 2 hypervisors are suitable for end-user computing.

# What are the benefits of virtualization?

- **Efficient resource use :**
  - Virtualization improves hardware resources used in your data center.
  - For example, instead of running one server on one computer system, you can create a virtual server pool on the same computer system by using and returning servers to the pool as required.
  - Having fewer underlying physical servers frees up space in your data center and saves money on electricity, generators, and cooling appliances.

# What are the benefits of virtualization?

- **Automated IT management :**
  - Now that physical computers are virtual, you can manage them by using software tools. Administrators create deployment and configuration programs to define virtual machine templates.
  - You can duplicate your infrastructure repeatedly and consistently and avoid error-prone manual configurations.

# What are the benefits of virtualization?

- **Faster disaster recovery :**

- When events such as natural disasters or cyberattacks negatively affect business operations, regaining access to IT infrastructure and replacing or fixing a physical server can take hours or even days.
- By contrast, the process takes minutes with virtualized environments. This prompt response significantly improves resiliency and facilitates business continuity so that operations can continue as scheduled.

# How does virtualization work?

- Virtualization uses specialized software, called a hypervisor, to create several cloud instances or virtual machines on one physical computer.
- Hypervisors is a software layer that acts as an intermediary between the virtual machines and the underlying hardware or host operating system.
- The hypervisor coordinates access to the physical environment so that several virtual machines have access to their own share of physical resources.

# Types of Virtualization

- **Server Virtualization**
  - Server virtualization is a process that partitions a physical server into multiple virtual servers.
  - It is an efficient and cost-effective way to use server resources and deploy IT services in an organization.
  - Without server virtualization, physical servers use only a small amount of their processing capacities, which leave devices idle.

# Types of Virtualization

- **Storage Virtualization**
  - Storage virtualization combines the functions of physical storage devices such as network attached storage (NAS) and storage area network (SAN).
  - You can pool the storage hardware in your data center, even if it is from different vendors or of different types.
  - Storage virtualization uses all your physical data storage and creates a large unit of virtual storage that you can assign and control by using management software.

# Types of Virtualization

- **Network Virtualization**
  - Any computer network has hardware elements such as switches, routers, and firewalls.
  - An organization with offices in multiple geographic locations can have several different network technologies working together to create its enterprise network.
  - Network virtualization is a process that combines all of these network resources to centralize administrative tasks.

# Types of Virtualization

- **Network Virtualization**
  - **Software-defined networking**
    - Software-defined networking (SDN) controls traffic routing by taking over routing management from data routing in the physical environment.
  - **Network function virtualization**
    - Network function virtualization technology combines the functions of network appliances, such as firewalls, load balancers, and traffic analyzers that work together, to improve network performance.

# Types of Virtualization

- **Data Virtualization**
  - Modern organizations collect data from several sources and store it in different formats.
  - They might also store data in different places, such as in a cloud infrastructure and an on-premises data center.
  - Data virtualization creates a software layer between this data and the applications that need it.

# Types of Virtualization

- **Application Virtualization**
  - Application virtualization software allows users to access and use an application from a separate computer than the one on which the application is installed.
  - Using application virtualization software, IT admins can set up remote applications on a server and deliver the apps to an end user's computer.

# Types of Virtualization

- **Application Virtualization**
  - **Application streaming** : Users stream the application from a remote server, so it runs only on the end user's device when needed.
  - **Server-based application virtualization** : Users can access the remote application from their browser or client interface without installing it.
  - **Local application virtualization** : The application code is shipped with its own environment to run on all operating systems without changes.

# Types of Virtualization

- **Desktop Virtualization**
  - You can use desktop virtualization to run different desktop operating systems on virtual machines, which your teams can access remotely.
  - This type of virtualization makes desktop management efficient and secure, saving money on desktop hardware.

# Types of Virtualization

- **Desktop Virtualization**
  - **Virtual desktop infrastructure** : Virtual desktop infrastructure runs virtual desktops on a remote server. Your users can access them by using client devices.
  - **Local desktop virtualization** : In local desktop virtualization, you run the hypervisor on a local computer and create a virtual computer with a different operating system. You can switch between your local and virtual environment in the same way you can switch between applications.

# Resource Overcommitment

- Resource overcommitment is a technique by which multiple virtual machines share the same physical CPU, memory and disk of the underlying hypervisor.
- Even though a virtual machine is created with certain resource capacity, for most workloads it is not always needed.
- So to derive higher efficiency, KVM transparently shares physical resource capacity between virtual machines. This type of resource sharing can lead to better utilization. However, in times of contention, a VM's performance can be impacted.

# Virtualization Provisioning and Migration Service

# Virtual Machine Life Cycle



# **Virtual Machine Life Cycle**

- The cycle starts by a request delivered to the IT department, stating the requirement for creating a new server for a particular service.
- This request is being processed by the IT administration to start seeing the servers' resource pool, matching these resources with requirements.
- Starting the provision of the needed virtual machine.
- Once it provisioned and started, it is ready to provide the required service according to an SLA.
- Virtual is being released; and free resources.

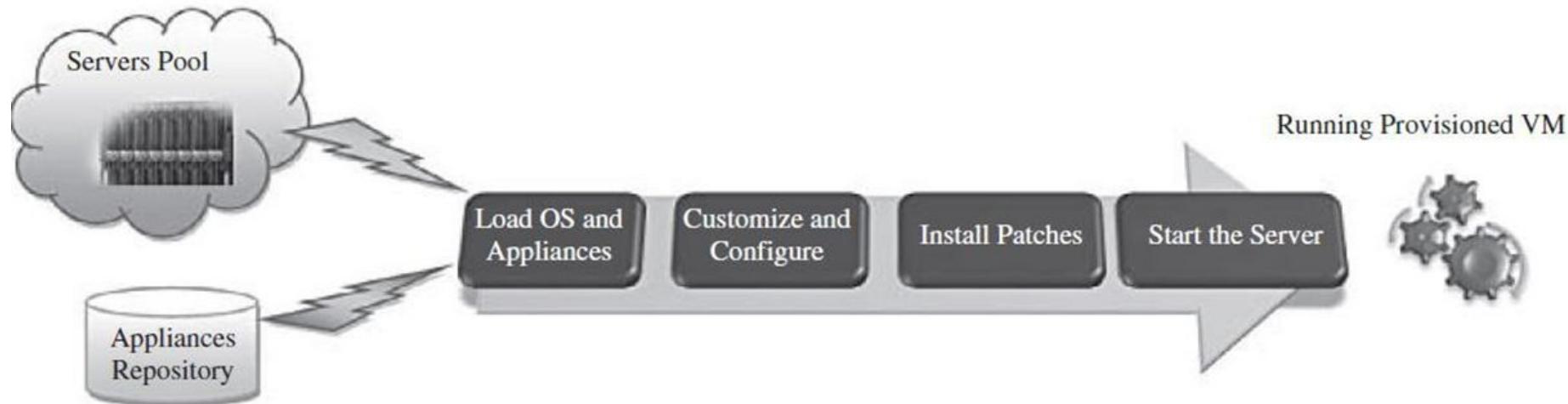
# Virtual Machine Provisioning Process

- Firstly, you need to select a server from a pool of available servers (physical servers with enough capacity) along with the appropriate OS template you need to provision the virtual machine.
- Secondly, you need to load the appropriate software (operating System you selected in the previous step, device drivers,middleware, and the needed applications for the service required).
- Thirdly, you need to customize and configure the machine (e.g.,IP address, Gateway) to configure an associated network and storage resources.

# Virtual Machine Provisioning Process

- Finally, the virtual server is ready to start with its newly loaded software.

# Migration Services



**FIGURE** Virtual machine provision process.

# Live Migration

- Live migration (which is also called hot or real-time migration) can be defined as the movement of a virtual machine from one physical host to another while being powered on.
- When it is properly carried out, this process takes place without any noticeable effect from the end user's point of view (a matter of milliseconds).

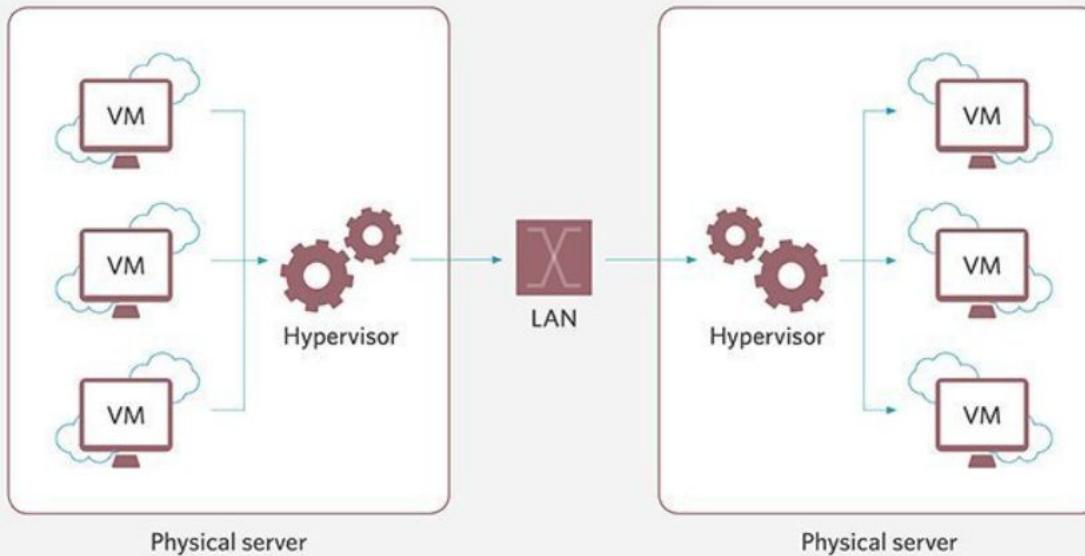
# Live Migration

- One of the most significant advantages of live migration is the fact that it facilitates proactive maintenance in case of failure, because the potential problem can be resolved before the disruption of service occurs.
- Live migration can also be used for load balancing in which work is shared among computers in order to optimize the utilization of available CPU resources.
- Some popular hypervisor products that support guest OS live migration include **VMware vMotion**, **Microsoft Hyper-V** and **Oracle Logical Domains (LDoms)** software.

<https://www.studocu.com/in/document/kannur-university/cloud-computing/cloud-computing-virtual-machines-provisioning-and-migration-services/26677585>

# Live Migration

## VM live migration



<https://www.techtarget.com/searchitoperations/definition/live-migration>

# Live Migration

- **When ?**
  - Live migration is usually performed when a physical host machine (computer or server) needs maintenance or an update, or when a VM must be switched to a different host.
- **What ?**
  - The process transfers the **VM memory, network connectivity and storage.**
- **How ?**
  - Most of the migration occurs while the OS continues to run.
  - The process allows a clean separation between hardware and software with a separation of concerns between the users and operator of a data center or cluster.
  - For these reasons, live OS migration is particularly useful for cluster administrators.

<https://www.techtarget.com/searchitoperations/definition/live-migration>

# Live Migration

- With live migration, admins can consolidate clustered hardware into a single coherent management domain.
- If they need to remove a particular physical machine from service for maintenance, they may migrate OS instances (including applications) to one or more alternative machines, freeing the original machine.
- Similarly, when hosts are congested, they may rearrange OS instances across machines in a cluster to relieve the load. In either situation, the combination of virtualization and migration eases systems management for the cluster admin.

# Live Migration

- The live migration process starts by transferring the data in the VM's memory to the target physical machine.
- Once all the data is transferred, an "**operational resource state**" consisting of CPU, memory and storage is created on the target machine.
- After this, the original VM -- along with its installed applications -- is suspended, copied and initiated on the destination.
- This entire process causes minimal downtime. Although it's not possible to completely avoid downtime, it can be further reduced with pre-paging and by using the memory's probability density function.

# Live Migration Benefits

- Live migration offers several benefits for administrators of data centers and clusters.
- Migrating an entire OS and all its applications as one unit can eliminate many of the difficulties involved in process-level migration approaches.
- This method eliminates the issue of residual dependencies that require the original host machine to remain available and network-accessible to service memory accesses or system calls on behalf of migrated processes.
- Migrating at the entire VM level also means that its in-memory state can be transferred consistently and efficiently.
- This applies to both the kernel internal state and application level state.

# Live Migration Benefits

- Live migration supports more efficient load balancing, so systems and CPU resources can be shared for optimum use.
- It also allows applications to continue running while the administrator manages maintenance activities, such as security updates, in the background.
- Users can control the software and services they want to run within their VM without providing the operator with any OS-level access.
- Moreover, the system remains active even if any hardware such as the CPU, network interface card or memory stops working.
- If the system crashes completely or the live migration fails, it will crash the VM, log a host error and automatically restart the machine.

# Live Migration Benefits

- Finally, live migration minimizes system downtime by using the pre-paging approach in which the OS guesses in advance which pages of memory will be required, and proactively pre-loads them into the main memory without halting the VM being migrated.

# Live Migration Steps

- **Pre-migration or preparation**
  - The target host (host A) is preselected for migration, and the VM is made active on the client side. The hypervisor also duplicates the memory pages from the source file to the destination file.
- **Reservation**
  - A request for migration is passed from host A to host B. With this request initialization, host B reserves a VM container of the required size. If these resources cannot be secured, the VM continues to run in host A unaffected.

# Live Migration Steps

- **Iterative (Repetitive) Pre-copy**
  - Pre-copy migration combines an iterative push phase and a stop-and-copy phase. This way, all pages from host A are transferred to host B. Further, in subsequent iterations, only pages that were altered or dirtied during the transfer process will be considered.
- **Stop-and-copy**
  - Running OS instances are suspended at host A and the network traffic is redirected to host B. The CPU state and other inconsistent memory pages are then transferred to host B. Finally, there is a consistent suspended copy of the VM in both hosts, with the copy at A considered primary. This way, migration can be resumed from A in case of failure.

# Live Migration Steps

- **Commitment**
  - Host B informs host A that it has received the consistent OS image. Host A acknowledges this message, and this becomes the commitment of migration transaction. Now host B becomes the primary host, and host A can discard the original VM.
- **Activation**
  - The migrated VM is activated on the now primary host B. Device drivers are reattached to the new machine and moved IP addresses are advertised with post-migration codes. Normal operations resume in host B.

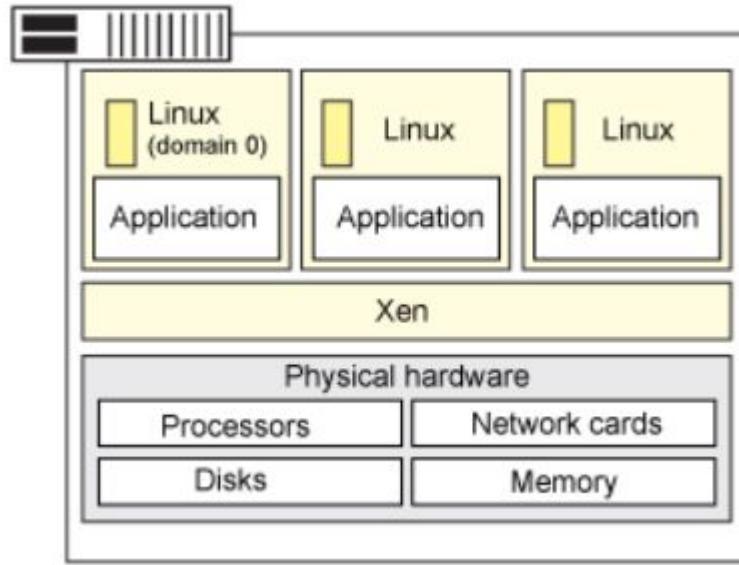
# Live Migration Requirements

- Live migration ensures that the consistent VM image remains in at least one host. However, process success hinges on two key requirements:
- **Original host remains stable:**
  - Throughout the migration process, the original host must be stable without any interruption until the commitment stage.
- **Suspending and resuming VM:**
  - The VM can be suspended and resumed in the physical host without a risk of failure.

# Xen Hypervisor

- Xen is a type 1 hypervisor that creates logical pools of system resources so that many virtual machines can share the same physical resources.
- Xen is a hypervisor that runs directly on the system hardware. Xen inserts a virtualization layer between the system hardware and the virtual machines, turning the system hardware into a pool of logical computing resources that Xen can dynamically allocate to any guest operating system.
- The operating systems running in virtual machines interact with the virtual resources as if they were physical resources.

# Xen Hypervisor



# Xen Hypervisor : Features

- Full virtualization.
  - Most hypervisors are based on full virtualization which means that they completely emulate all hardware devices to the virtual machines.
  - Guest operating systems do not require any modification and behave as if they each have exclusive access to the entire system.
  - Full virtualization often includes performance drawbacks because complete emulation usually demands more processing resources (and more overhead) from the hypervisor.
  - Xen is based on *paravirtualization*; it requires that the guest operating systems be modified to support the Xen operating environment. However, the user space applications and libraries do not require modification.

# Xen Hypervisor : Features

- Xen can run multiple guest OS, each in its own VM.
  - Xen can run several guest operating systems each running in its own virtual machine or domain. When Xen is first installed, it automatically creates the first domain, Domain 0 (or dom0).
  - Domain 0 is the management domain and is responsible for managing the system.
  - It performs tasks like building additional domains (or virtual machines), managing the virtual devices for each virtual machine, suspending virtual machines, resuming virtual machines, and migrating virtual machines.
  - Domain 0 runs a guest operating system and is responsible for the hardware devices.

# Xen Hypervisor : Features

- Instead of a driver, lots of great stuff happens in the Xen daemon, xend.
  - The Xen daemon, xend, is a Python program that runs in dom0.
  - It is the central point of control for managing virtual resources across all the virtual machines running on the Xen hypervisor.
  - Most of the command parsing, validation, and sequencing happens in user space in xend and not in a driver.

# Cold Migration

- A powered down Virtual Machine is carried to separate host or data store.
- Virtual Machine's power state is OFF and there is no need of common shared storage.
- There is a lack of CPU check and there is long shortage time. Log files and configuration files are migrated from the source host to the destination host.
- The first host's Virtual Machine is shut down and again started on next host.
- Applications and OS are terminated on Virtual Machines before moving them to physical devices.
- User is given choice of movement of disks associated from one data store to another one.

<https://www.geeksforgeeks.org/hot-and-cold-migrations/>

# Cold Migration

- You have options of moving the associated disks from one data store to another.
- The virtual machines are not required to be on a shared storage.
- Live migrations needs to a shared storage for virtual machines in the server's pool, but cold migration does not. 2) In live migration for a virtual machine between two hosts, there should be certain CPU compatibility checks, but in cold migration this checks do not apply.

# Cold Migration

- Cold migration (VMware product ) is easy to implement and is summarized as follows:
- The configuration files, including NVRAM file (BIOS Setting), log files, and the disks of the virtual machines, are moved from the source host to the destination host's associated storage area.
- The virtual machine is registered with the new host.
- After the migration is completed, the old version of the virtual machine is deleted from the source host.

# Scheduling Techniques Of Virtual Machines For Resource Reservation

# What is Scheduling ?

- The major objective of scheduling in distributed systems is to maximize the processor utilization and minimize the task execution time by distributing the processor load when a need arises in the dynamic environment.
- Job scheduling, an optimized problem, plays a vital role for improving the reliability and flexibility of the systems.
- The role of scheduling algorithms is to find out a proper sequence in which the jobs can be executed under the constraints within a reasonable time.

# Type of Scheduling

<b>Static Scheduling</b>	<b>Dynamic Scheduling</b>
Suitable for homogeneous and stable environment.	Suitable for heterogeneous and dynamic environment.
Gives low performance and might have lots of overhead compared with dynamic scheduling algorithm.	Gives higher performance than static scheduling algorithm.
Attributes are mostly not taken into consideration during execution.	Considers different types of attributes in the system both prior and during the execution time.
As it is more suitable for static environment, this scheduling might provide good results in such environment.	Some of the considered attributes might make the system complicated and inefficient as well as overhead during execution and might degrade the services provided.

# Why Scheduling?

- When virtualization is achieved, for effective management of the resources, a scheduler is put into place for utilizing the resources.
- The important issue that needs to be addressed in computing environment is scheduling.
- In cloud environment, scheduler's responsibility is to order the arriving jobs in such a manner that it must maintain the fairness among the requesting job as well as must schedule those in an efficient way without compromising the Quality of Service (QoS).

# Why Scheduling?

- In the cloud computing environment, virtual machine scheduling algorithms are used for scheduling the Virtual Machine request to the Physical Machines (PM) that belongs to a particular Data Center (DC).
- The scheduling is done as per the requirements received and based on the resources/compute availability i.e, RAM, Bandwidth, Storage, Processors etc.

# Scheduling Criterions

- **Resource Utilization** : It is used for testing the resource utilization. It is expected maximum for an efficient scheduling algorithm.
- **Throughput** : It is defined as the number of tasks executed at a fixed interval of time.
- **Response Time** : It can be defined as the amount of time taken by the system to produce the first output from the time of submission. It is expected to be reduced for a better performance.
- **Performance** : It is defined as the system's efficiency and the scheduling algorithm must be able to improve it.

# Scheduling Criterions

- **Scalability** : It is used for measuring the quality of service as the number of nodes increases in the computing network. It is expected to be the same even if maximum numbers of nodes are getting added. The nodes must be added without affecting the services.
- **Association of overhead** : Overhead occurs due to Interprocess communication and task movement. The overhead must be reduced and the loads must be balanced properly. The algorithm must work well to satisfy the criteria.
- **Fault Tolerance** : It is defined as the ability of the system to perform uniform balancing of the load across the available resources in spite of failure of the node(s).

# Scheduling Algorithms

- **Round Robin Algorithm**

- The round robin algorithm's major objective is to distribute the jobs equally to all of the existing nodes. The scheduler allocates one virtual machine to one node in a cyclic manner. This algorithm ensures fairness in allocation of jobs to the available virtual machine and utilizes the resources in a balanced manner.
-

# Scheduling Algorithms

- **Weighted Round Robin Algorithm**
  - This algorithm works on the basis of weight's allocated. The incoming requests are allocated to the available virtual machine in a round robin manner that takes the weight into consideration. This algorithm does not consider the current load that is allocated to the virtual machine at the given instance. This might make a machine overloaded where as another machine under loaded.

# Scheduling Algorithms

- **Priority Scheduling Algorithm**

- The general idea behind priority based scheduling algorithm is, each VM is assigned with an internally defined priority based on the characteristics such as the amount of work it can do, time taken for executing the job, and it is allowed to run on each machine.
- Instances with equal priority are scheduled in FCFS order. Such assigned priorities can be changed dynamically using the aging technique, where the priority of the VM is increased based on the total amount of time VM remains in the ready queue for execution.
- By implementing such technique, the priority of the VM keeps increasing and at one point of time it reaches a higher priority.

# Memory Virtualization And Storage Virtualization

# Memory Virtualization

- Virtualization is the abstraction of IT resources, separating their physical instance and boundaries from their function.
- Memory is required in every digital machine; switches, routers, appliances and servers. Each contains physical memory alongside the logic that manipulates the 1's and 0's. Memory is closely coupled with compute logic, and when performance gains are needed enterprises typically add more memory, which can be very expensive.
- Memory Virtualization introduces a way to decouple memory from the processor, AND, from the server to provide a shared, distributed or networked function. This is not more addressable memory but virtualized memory shared between multiple machines.
- Memory and storage are not synonymous.

# Memory Virtualization

- Memory virtualization decouples volatile random access memory (RAM) resources from individual systems in the data centre, and then aggregates those resources into a virtualized memory pool available to any computer in the cluster.
- The memory pool is accessed by the operating system or applications running on top of the operating system.
- The distributed memory pool can then be utilized as a high-speed cache, a messaging layer, or a large, shared memory resource for a CPU or a GPU application.

# Memory Virtualization

- Memory virtualization enables networked, and hence distributed, servers to share a pool of memory to bypass physical memory constraints, a common bottleneck in software performance.
- Applications can take advantage of a huge quantity of memory with this feature integrated into the network, improving overall performance, system utilization, memory usage efficiency, and enabling new use cases.
- Nodes can connect to the memory pool using software on the memory pool nodes (servers) to donate memory and store and retrieve data.

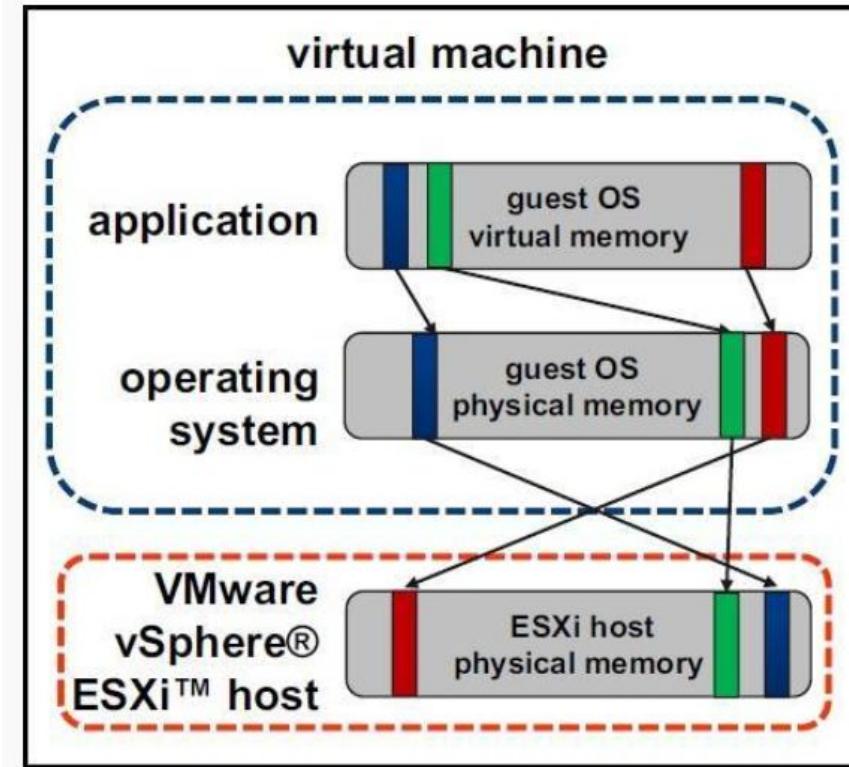
# Memory Virtualization

- Shared memory, data insertion, eviction, provisioning policies, data assignment to contributing nodes, and requests from client nodes are managed by management software and memory overcommitment technologies. The memory pool can be accessed from either the program or the operating system.
- To establish a high-speed shared memory cache, the pool is accessed through an API or as a networked file system at the application level. A page cache can use the pool as a very large memory resource that is substantially faster than local or networked storage at the operating system level.

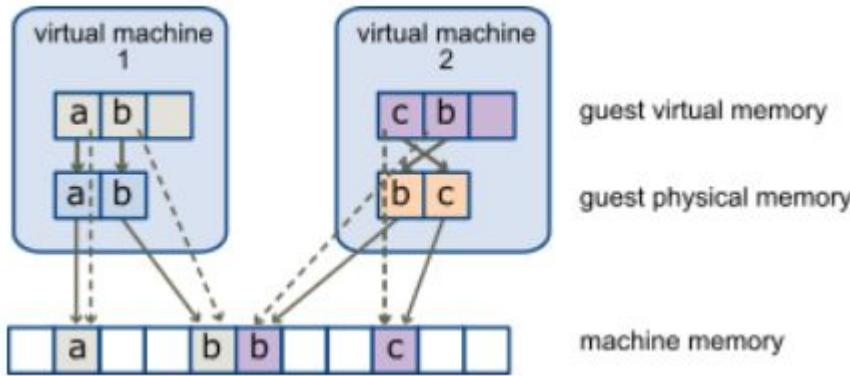
# Memory Virtualization

- Virtual memory virtualization is similar to modern operating systems' virtual memory capabilities.
- The operating system maintains virtual memory to machine memory mappings using page tables in a typical execution environment, a one-stage mapping from virtual memory to machine memory.
- Modern x86 CPUs feature a memory management unit (MMU) and a translation lookaside buffer (TLB) to maximize virtual memory performance. On the other hand, virtual memory virtualization in a virtual execution environment entails sharing the physical system memory in RAM and dynamically assigning it to the physical memory of the VMs.

# Two Stage Memory Mapping



# Two Stage Memory Mapping



- The boxes represent pages, and the arrows show the different memory mappings.
- The arrows from guest virtual memory to guest physical memory show the mapping maintained by the page tables in the guest operating system. (The mapping from virtual memory to linear memory for x86-architecture processors is not shown.)
- The arrows from guest physical memory to machine memory show the mapping maintained by the VMM.
- The dashed arrows show the mapping from guest virtual memory to machine memory in the shadow page tables also maintained by the VMM. The underlying processor running the virtual machine uses the shadow page table mappings.

# Two Stage Memory Mapping

- That means a two-stage mapping process should be maintained by the guest OS and the VMM (Virtual Machine Monitors), respectively:
  - virtual memory to physical memory
  - physical memory to machine memory
- Furthermore, MMU (memory management unit) virtualization should be supported, which is transparent to the guest OS.
- The guest OS continues to control the mapping of virtual addresses to the physical memory addresses of VMs.
- But the guest OS cannot directly access the actual machine memory.
- The VMM is responsible for mapping the guest physical memory to the actual machine memory.

# Two Stage Memory Mapping

- Since each page table of the guest OSes has a separate page table in the VMM corresponding to it, the VMM page table is called the **shadow page table**.
- Nested page tables add another layer of indirection to virtual memory.
- The MMU already handles virtual-to-physical translations as defined by the OS. Then the physical memory addresses are translated to machine addresses using another set of page tables defined by the hypervisor.
- Since modern operating systems maintain a set of page tables for every process, the shadow page tables will get flooded. Consequently, the performance overhead and cost of memory will be very high.

# Two Stage Memory Mapping

- VMware uses shadow page tables to perform virtual-memory-to-machine-memory address translation.
- Processors use TLB hardware to map the virtual memory directly to the machine memory to avoid the two levels of translation on every access. When the guest OS changes the virtual memory to a physical memory mapping, the VMM updates the shadow page tables to enable a direct lookup.
- The AMD Barcelona processor has featured hardware-assisted memory virtualization since 2007. It provides hardware assistance to the two-stage address translation in a virtual execution environment by using a technology called nested paging.

# Storage Virtualization

- Storage virtualization is the pooling of physical storage from multiple storage devices into what appears to be a single storage device -- or pool of available storage capacity. A central console manages the storage.
- The technology relies on software to identify available storage capacity from physical devices and to then aggregate that capacity as a pool of storage that can be used by traditional architecture servers or in a virtual environment by virtual machines.

# Storage Virtualization

- The virtual storage software intercepts input/output (I/O) requests from physical or virtual machines and sends those requests to the appropriate physical location of the storage devices that are part of the overall pool of storage in the virtualized environment.
- To the user, the various storage resources that make up the pool are unseen, so the virtual storage appears like a single physical drive, share or logical unit number (LUN) that can accept standard reads and writes.
- Even a redundant array of independent disks, or RAID, array can sometimes be considered a type of storage virtualization.
- Multiple physical drives in the array are presented to the user as a single storage device that, in the background, stripes and replicates data to multiple disks to improve I/O performance and to protect data in case a single drive fails.

<https://www.techtarget.com/searchstorage/definition/storage-virtualization>

# Type of Storage Virtualization

- **File-based :**
  - File-based storage virtualization is a specific use, applied to network-attached storage (NAS) systems.
  - Using Server Message Block in Windows server environments or Network File System protocols for Linux systems, file-based storage virtualization breaks the dependency in a normal NAS array between the data being accessed and the location of physical memory.
  - The pooling of NAS resources makes it easier to handle file migrations in the background, which will help improve performance.
  - Typically, NAS systems are not that complex to manage, but storage virtualization greatly simplifies the task of managing multiple NAS devices through a single management console.

# Type of Storage Virtualization

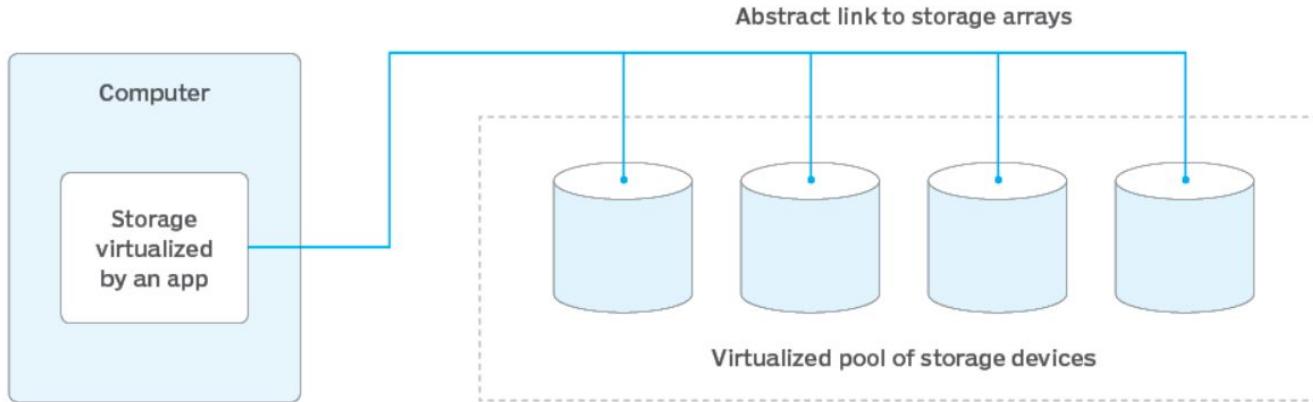
- **Block-based or block access storage:**
  - Storage resources typically accessed via a Fibre Channel (FC) or Internet Small Computer System Interface (iSCSI) storage area network (SAN).
  - It is more frequently virtualized than file-based storage systems.
  - Block-based systems abstract the logical storage, such as a drive partition, from the actual physical memory blocks in a storage device, such as a hard disk drive (HDD) or solid-state memory device.
  - Because it operates in a similar fashion to the native drive software, there's less overhead for read and write processes, so block storage systems will perform better than file-based systems.

# Type of Storage Virtualization

- The block-based operation enables the virtualization management software to collect the capacity of the available blocks of storage space across all virtualized arrays.
- It pools them into a shared resource to be assigned to any number of VMs, bare-metal servers or containers. Storage virtualization is particularly beneficial for block storage.
- Unlike NAS systems, managing SANs can be a time-consuming process.
- Consolidating a number of block storage systems under a single management interface that often shields users from the tedious steps of LUN configuration, for example, can be a significant timesaver.

# How Storage Virtualization works

## Storage virtualization architecture



# How Storage Virtualization works

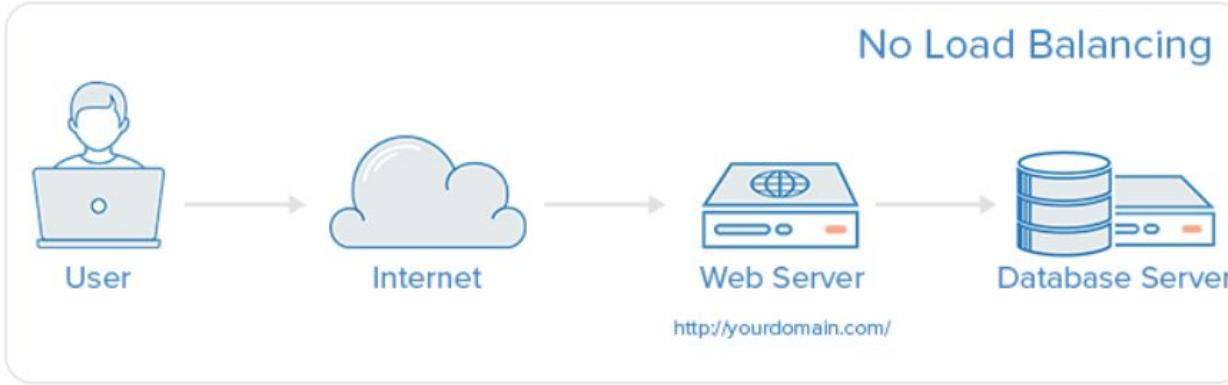
- To provide access to the data stored on the physical storage devices, the virtualization software needs to either create a map using metadata or, in some cases, use an algorithm to dynamically locate the data on the fly.
- The virtualization software then intercepts read and write requests from applications. Using the map it has created, it can find or save the data to the appropriate physical device. This process is similar to the method used by PC OSes when retrieving or saving application data.
- Storage virtualization disguises the actual complexity of a storage system, such as a SAN, which helps a storage administrator perform the tasks of backup, archiving and recovery more easily and in less time.

# In-band vs. out-of-band virtualization

- **In-band virtualization** : It is also called symmetric virtualization -- handles the data that's being read or saved and the control information, such as I/O instructions and metadata, in the same channel or layer. This setup enables the storage virtualization to provide more advanced operational and management functions such as data caching and replication services.
- **Out-of-band virtualization** : It is also called asymmetric virtualization -- splits the data and control paths. Since the virtualization facility only sees the control instructions, advanced storage features are usually unavailable.

# Load Balancing And Horizontal and Vertical Scaling

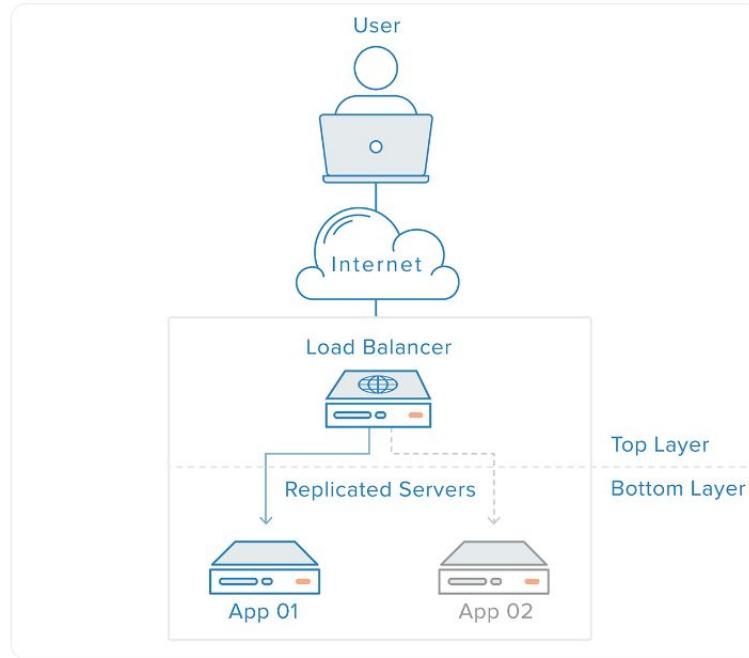
# Load Balancing



A web infrastructure with no load balancer

- No service during down time.
- Unmanageable load during multiple simultaneous server access.

# Load Balancing



A web infrastructure with load balancer

# What kind of traffic can load balancers handle?

- HTTP
- HTTPS
- TCP
- UDP

# How does the load balancer choose the backend server?

- **Health Checks :**
  - Load balancers should only forward traffic to “healthy” backend servers.
  - To monitor the health of a backend server, health checks regularly attempt to connect to backend servers using the protocol and port defined by the forwarding rules to ensure that servers are listening.
  - If a server fails a health check, and therefore is unable to serve requests, it is automatically removed from the pool, and traffic will not be forwarded to it until it responds to the health checks again.

# How does the load balancer choose the backend server?

- **Load Balancing Algorithms :**
  - The load balancing algorithm that is used determines which of the healthy servers on the backend will be selected. A few of the commonly used algorithms are:
  - **Round Robin** : Servers will be selected sequentially. The load balancer will select the first server on its list for the first request, then move down the list in order, starting over at the top when it reaches the end.
  - **Least Connections** : Least Connections means the load balancer will select the server with the least connections and is recommended when traffic results in longer sessions.

# How does the load balancer choose the backend server?

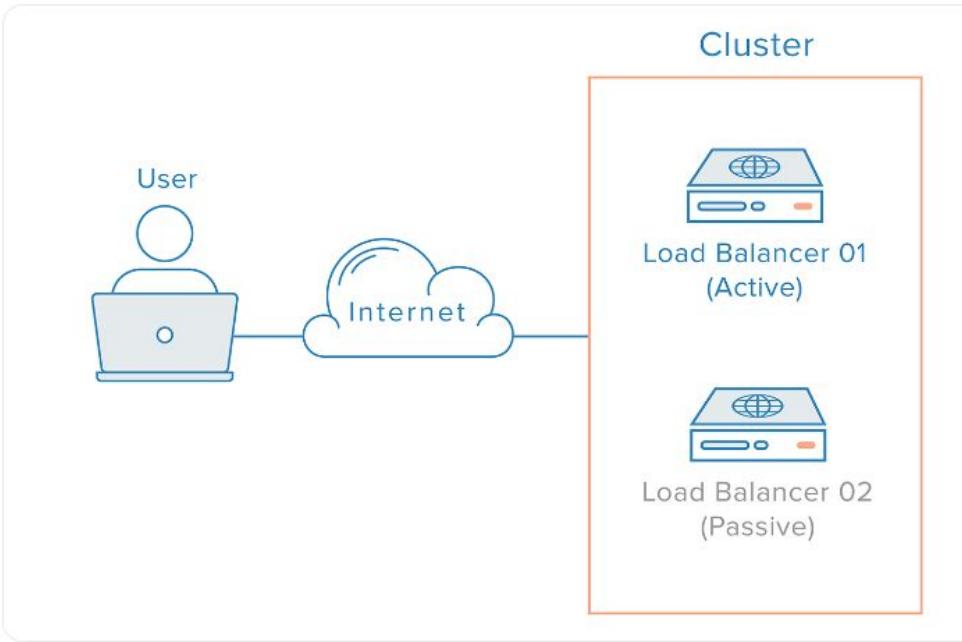
- **Load Balancing Algorithms :**
  - **Source** : With the Source algorithm, the load balancer will select which server to use based on a hash of the source IP of the request, such as the visitor's IP address. This method ensures that a particular user will consistently connect to the same server.

Note : The algorithms available to administrators vary depending on the specific load balancing technology in use.

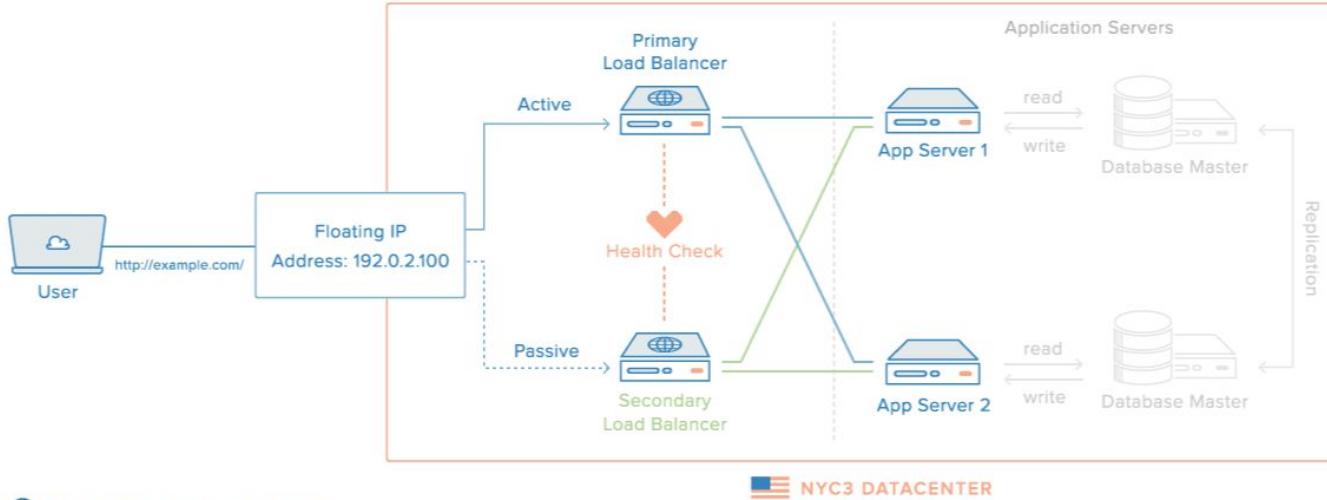
# How do load balancers handle state?

- Some applications require that a user continues to connect to the same backend server. A Source algorithm creates an affinity based on client IP information. Another way to achieve this at the web application level is through **sticky sessions**, where the load balancer sets a cookie and all of the requests from that session are directed to the same physical server.

# Redundant Load Balancers



# Redundant Load Balancers



# Type of Load Balancing Algorithm

- **Load Balancing Algorithm**
  - **Static Load Balancing :**
    - **Round-robin method**
    - **Weighted round-robin method**
    - **IP hash method**

# Type of Load Balancing Algorithm

- **Load Balancing Algorithm**
  - **Dynamic Load Balancing :**
    - **Least connection method**
    - **Weighted least connection method**
      - Weighted least connection algorithms assume that some servers can handle more active connections than others.
      - Therefore, you can assign different weights or capacities to each server, and the load balancer sends the new client requests to the server with the least connections by capacity.

<https://aws.amazon.com/what-is/load-balancing/>

# Type of Load Balancing Algorithm

## ■ Least response time method

- The response time is the total time that the server takes to process the incoming requests and send a response.
- The least response time method combines the server response time and the active connections to determine the best server.
- Load balancers use this algorithm to ensure faster service for all users.

# Type of Load Balancing Algorithm

## ■ Resource-based method

- In the resource-based method, load balancers distribute traffic by analyzing the current server load.
- Specialized software called an agent runs on each server and calculates usage of server resources, such as its computing capacity and memory.
- Then, the load balancer checks the agent for sufficient free resources before distributing traffic to that server.

# Type of Load Balancing

- **Application load balancing**
  - Complex modern applications have several server farms with multiple servers dedicated to a single application function.
  - It looks at the request content, such as HTTP headers or SSL session IDs, to redirect traffic.

# Type of Load Balancing

- **Network load balancing**
  - Examine IP addresses and other network information to redirect traffic optimally.
  - They track the source of the application traffic and can assign a static IP address to several servers.
  - Network load balancers use the static and dynamic load balancing algorithms described earlier to balance server load.

# Type of Load Balancing

- **Global server load balancing**

- Global server load balancing occurs across several geographically distributed servers.
- In this case, local load balancers manage the application load within a region or zone.
- They attempt to redirect traffic to a server destination that is geographically closer to the client.
- They might redirect traffic to servers outside the client's geographic zone only in case of server failure.

<https://aws.amazon.com/what-is/load-balancing/>

# Type of Load Balancing

- **DNS load balancing**
  - In DNS load balancing, you configure your domain to route network requests across a pool of resources on your domain.
  - A domain can correspond to a website, a mail system, a print server, or another service that is made accessible through the internet.
  - DNS load balancing is helpful for maintaining application availability and balancing network traffic across a globally distributed pool of resources.

# Types of Load Balancing Technology

- **Hardware load balancers**
  - Not flexible
  - Costly
  - A hardware-based load balancer is a hardware appliance that can securely process and redirect gigabytes of traffic to hundreds of different servers.
  - You can store it in your data centers and use virtualization to create multiple digital or virtual load balancers that you can centrally manage.

# Types of Load Balancing Technology

- **Software load balancers**
  - Less expensive
  - Flexible
  - Software-based load balancers are applications that perform all load balancing functions.
  - You can install them on any server or access them as a fully managed third-party service.

# Scaling

- **Scalability of a Cloud Infrastructure**
  - The system's capacity to expand from its existing configuration to handle the rising workload is termed Scalability.
- **Benefits of Cloud Scaling**
  - **Fast and Easy** : Cloud Scaling architecture enables instructing additional VMs to manage the increasing workloads with just a few clicks. It eliminates the delay caused due to rising workloads.
  - **Cost efficiency** : Scaling horizontally or vertically is highly cost-efficient compared to an infrastructure where resources are idle most of the time.

# Scaling

- **Benefits of Cloud Scaling**
  - **Optimized Performance** : A scalable Cloud architecture efficiently manages a drastic rise and fall in traffic, thus optimizing the performance. It ensures the effective utilization of resources, thus eliminating idle resources or insufficient resource circumstances.
  - **Capacity** : The scalable architecture of the Cloud expands its capacity to manage the growing business requirements.

# Vertical Scaling



- Vertical Scaling is termed the Scale-up approach.
- Vertical Scaling is defined as increasing a single machine's capacity with the rising resources in the same logical server or unit.
- Involves adding resources like processing power, storage, and memory to the existing hardware or software, enhancing the system's capacity.

# Benefits of Vertical Scaling

- Flexible Scaling of resources
- Consumes less power
- Lower cooling cost
- Software cost-effective
- Reduced administrative efforts to manage a single system

# Horizontal Scaling

- Horizontal Scaling is an approach to enhance the performance of the server node by adding new instances of the server to the existing servers to distribute the workload equally.
- The strategy involves decreasing the server's load rather than expanding the capacity of the individual server.
- The Horizontal Scaling strategy is also termed Scaling out.
- Load balancing, clustering, and distributed file system are crucial strategies for Horizontal Scaling.
- Horizontal Scaling caters to portioning of the data where each node contains a single part of the data.

# Horizontal Scaling



- Vertical Scaling is termed the Scale-up approach.

# Benefits of Horizontal Scaling

- Flexible Scaling tools
- Easy fault-tolerance
- Enhanced resilience due to discrete, multiple systems
- Easy to upgrade
- Limitless Scaling with endless addition of server instances
- Cost-effective implementation

# Vertical vs Horizontal Scaling

Horizontal Scaling is defined as the ability to extend capacity by interfacing different hardware or software entities.	Vertical Scaling is defined as the ability to increase an existing system's capacity by adding resources.
It is based on partitioning where each node contains a single part of data.	The data is present on a single node and is scaled through multicore.
It is referred to as Scale-out.	It is referred to as Scale-up.

# Vertical vs Horizontal Scaling

The licensing fee is costly	The licensing is cost-effective
It requires a load balancer to distribute load among the servers within a system.	Scaling the server capacity enhances the load capacity of the server.
It implies boosting the power of individual server with the existing server.	It implies boosting the power of the individual server.

# Scheduling and its types

Dr Hitesh Mohapatra

Associate Professor

KIIT University, Bhubaneswar, Odisha, India

# Definition

The concept of scheduling in cloud computing refers to **the technique of mapping a set of jobs to a set of virtual machines (VMs) or allocating VMs to run on the available resources in order to fulfil users' demands**

# Cont.

- Scheduling is a method that is used to distribute valuable computing resources, usually processor time, bandwidth and memory, to the various processes, threads, data flows and applications that need them.
- Scheduling is done to balance the load on the system and ensure equal distribution of resources and give some prioritization according to set rules.
- This ensures that a computer system is able to serve all requests and achieve a certain quality of service.
- Scheduling is also known as process scheduling.

# Types

Scheduling in a system is done by the aptly named scheduler, which is mainly concerned with three things:

- **Throughput**, or how fast it can finish a certain number of tasks from beginning to end per unit of time
- **Latency**, which is the turnaround time or the time it takes to finish the task from the time of request or submission until the finish, which includes the waiting time before it could be served
- **Response time**, which is the time it takes for the process or request to be served, in short, the waiting time

# Cont.

- Scheduling is largely based on the factors mentioned in the previous slide and varies depending on the system and the programming of the system's or user's preferences and objectives.
- In modern computers such as PCs with large amounts of processing power and other resources and with the ability to multitask by running multiple threads or pipelines at once.
- Scheduling is no longer a big issue and most times processes and applications are given free rein with extra resources, but the scheduler is still hard at work managing requests.

# Types of scheduling include:

- **First come, first served** — The most straightforward approach and may be referred to as first in, first out; it simply does what the name suggests.
- **Round robin** — Also known as time slicing, since each task is given a certain amount of time to use resources. This is still on a first-come-first-served basis.
- **Shortest remaining time first** — The task which needs the least amount of time to finish is given priority.
- **Priority** — Tasks are assigned priorities and are served depending on that priority. This can lead to the starvation of the least important tasks as they are always preempted by more important ones.

# Scheduling levels

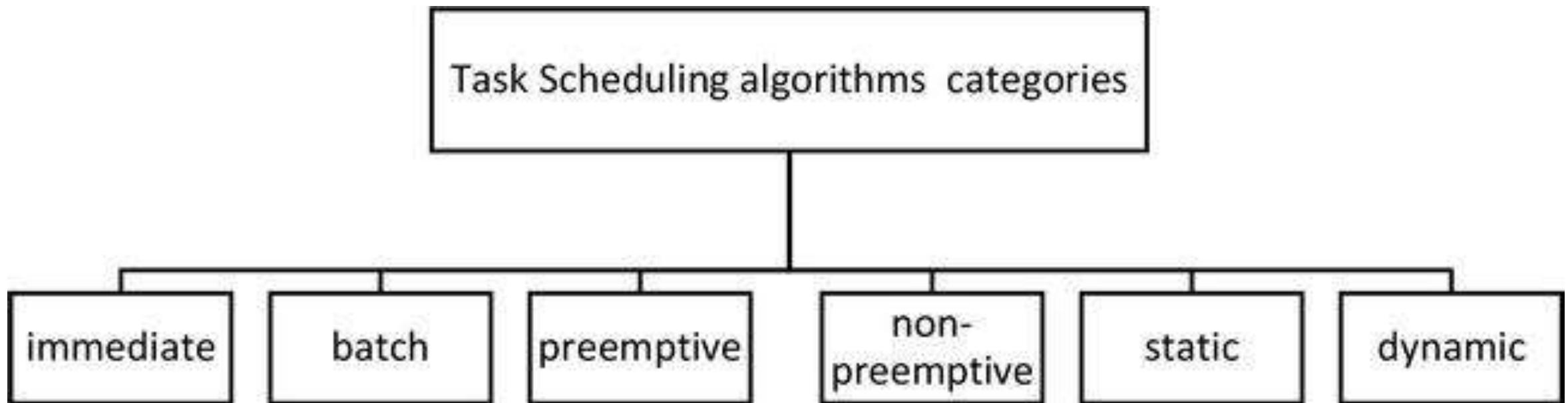
In the cloud computing environment there are two levels of scheduling algorithms:

- First level: in the host level where a set of policies to distribute VMs in the host.
- Second level: in VM level where a set of policies to distribute tasks to the VM.

# Task scheduling algorithms advantages

- Manage cloud computing performance and QoS.
- Manage the memory and CPU.
- Good scheduling algorithms maximize resource utilization while minimizing the total task execution time.
- Improving fairness for all tasks.
- Increasing the number of successfully completed tasks.
- Scheduling tasks on a real-time system.
- Achieving a high system throughput.
- Improving load balance.

# Tasks scheduling algorithms classifications



# Tasks scheduling algorithms can be classified as follows

- **Immediate scheduling:** when new tasks arrive, they are scheduled to VMs directly.
- **Batch scheduling:** tasks are grouped into a batch before being sent; this type is also called mapping events.
- **Static scheduling:** is considered very simple compared to dynamic scheduling; it is based on prior information of the global state of the system. It does not take into account the current state of VMs and then divides all traffic equivalently among all VMs in a similar manner such as round robin (RR) and random scheduling algorithms.

Cont.

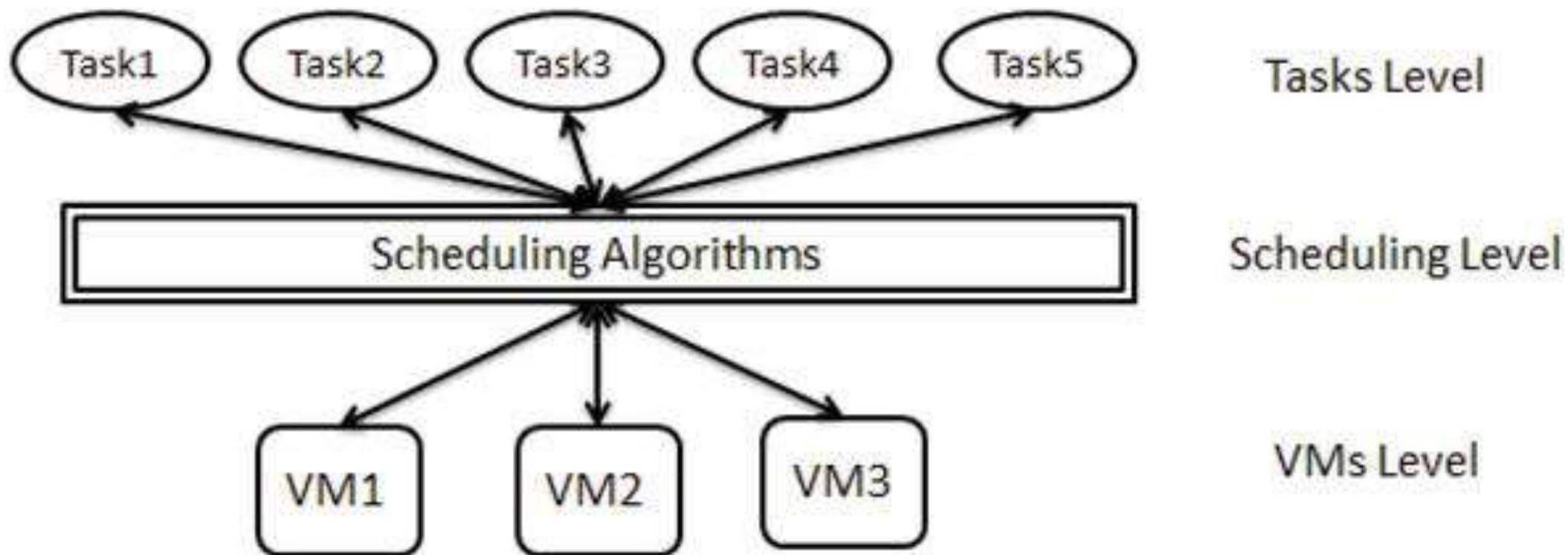
- **Dynamic scheduling:** takes into account the current state of VMs and does not require prior information on the global state of the system and distributes the tasks according to the capacity of all available VMs.
- **Preemptive scheduling:** each task is interrupted during execution and can be moved to another resource to complete execution.
- **Non-preemptive scheduling:** VMs are not reallocated to new tasks until finishing the execution of the scheduled task.

# Task scheduling system in cloud computing

The task scheduling system in cloud computing passes through three levels.

- **The first task level:** is a set of tasks (Cloudlets) that are sent by cloud users, which are required for execution.
- **The second scheduling level:** is responsible for mapping tasks to suitable resources to get the highest resource utilization with minimum make span. The make span is the overall completion time for all tasks from the beginning to the end.
- **The third VMs level:** is a set of (VMs) which are used to execute the tasks as in Figure 2.

# Scheduling



# This level passes through two steps

- The first step is discovering and filtering all the VMs that are presented in the system and collecting status information related to them by using a datacenter broker.
- In the second step a suitable VM is selected based on task properties.

# Static tasks scheduling algorithms in cloud computing environment

**FCFS:** the order of tasks in the task list is based on their arrival time and then assigned to VMs.

**Advantages:**

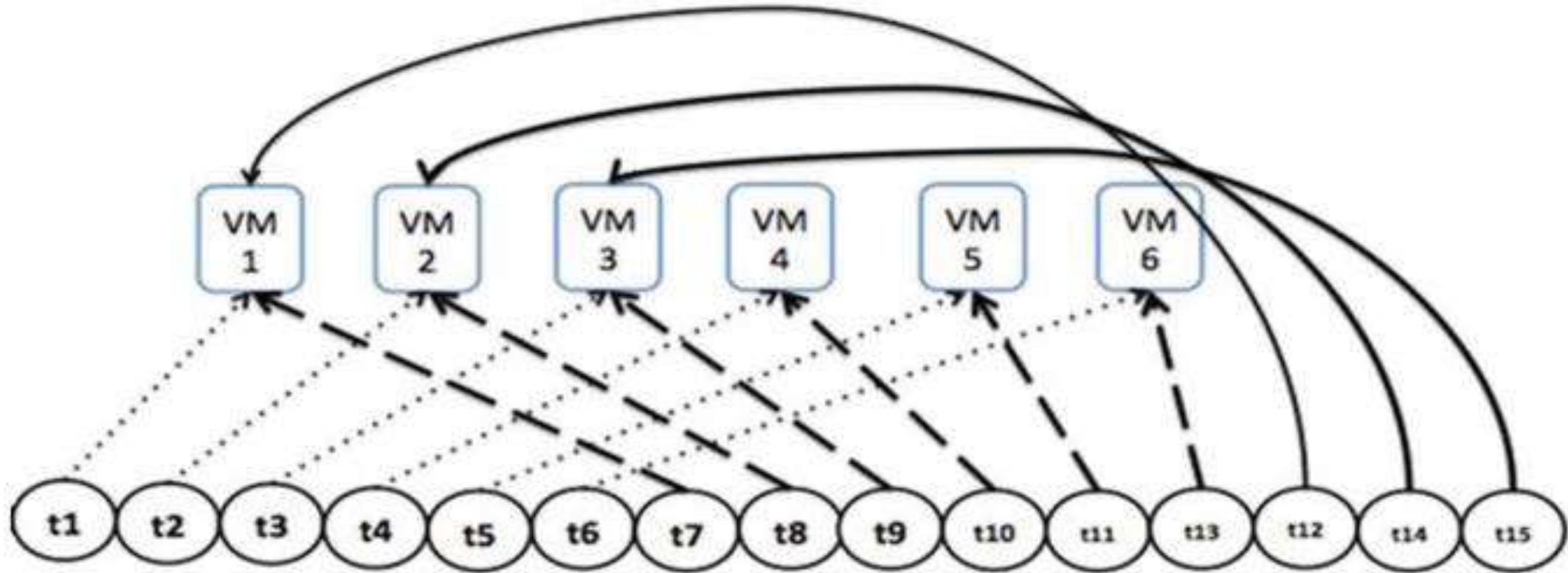
- Most popular and simplest scheduling algorithm.
- Fairer than other simple scheduling algorithms.
- Depend on the FIFO rule in scheduling tasks.
- Less complexity than other scheduling algorithms.

Cont.

### **Disadvantages of FCFS**

- Tasks have high waiting time.
- Not give any priority to tasks. That means when we have large tasks in the begin tasks list, all tasks must wait a long time until the large tasks to finish.
- Resources are not consumed in an optimal manner.
- In order to measure the performance achieved by this method, we will be testing them and then measuring its impact on (fairness, ET, TWT, and TFT).

When applying FCFS, work mechanism will be as following



- Dot arrows refer to the first set of tasks scheduling based on their arrival time.
- Dash arrows refer to second set of tasks scheduling based on their arrival time.
- Solid arrows refer to third set of tasks scheduling based on their arrival time.

# Example: FCFS

Task	Length
t1	100000
t2	70000
t3	5000
t4	1000
t5	3000
t6	10000
t7	90000
t8	100000
t9	15000
t10	1000
t11	2000
t12	4000
t13	20000
t14	25000
t15	80000

Assume we have six VMs with different properties based on tasks size:

VM list={VM1,VM2,VM3,VM4,VM5,VM6}.

MIPS of VM list={500,500,1500,1500,2500,2500}.

# Execution

$VM1 = \{t1 \rightarrow t7 \rightarrow t12\}$ .

$VM2 = \{t2 \rightarrow t8 \rightarrow t14\}$ .

$VM3 = \{t3 \rightarrow t9 \rightarrow t15\}$ .

$VM4 = \{t4 \rightarrow t10\}$ .

$VM5 = \{t5 \rightarrow t11\}$ .

$VM6 = \{t6 \rightarrow t13\}$ .

<i>Task</i>	<i>ET</i>	<i>Waiting time</i>	
t1	200	VM1	
t2	140	VM2	
t3	3.33	VM3	
t4	0.66	VM4	
t5	1.2	VM5	
t6	4	VM6	
t7	180	Wait(200)	VM1
t8	200	Wait(140)	VM2
t9	10	Wait(3.33)	VM3
t10	0.66	Wait(0.66)	VM4
t11	0.8	Wait(1.2)	VM5
t12	1.6	Wait(4)	VM6
t13	40	Wait(380)	VM1
t14	50	Wait(340)	VM2
t15	53.33	Wait(13.33)	VM3

# SJF

Tasks are sorted based on their priority. Priority is given to tasks based on task lengths and begins from (the smallest task  $\equiv$  highest priority).

## **Advantages**

Wait time is lower than FCFS.

SJF has a minimum average waiting time among all task scheduling algorithms.

## **Disadvantages**

Unfairness to some tasks when tasks are assigned to VM, due to the long tasks tending to be left waiting in the task list while small tasks are assigned to VM.

Taking long execution time and TFT.

# SJF work mechanism

When applying SJF, work mechanism will be as follows:

- Assume we have 15 tasks as in Table 1 above. We will be sorting tasks in the task list, as in Table 3. Tasks are sorted from smallest task to largest task based on their lengths as in Table 3, and then assigned to the VMs list sequentially.

Tasks	t4	t10	t11	t5	t12	t3	t6	t9	t13	t14	t2	t15	t7	t1	t8
lengths	1000	1000	2000	3000	4000	5000	10000	15000	20000	25000	70000	80000	90000	100000	100000

# Execution: SJF

VM1={t4→t6→t7}.  
VM2={t10→t9→t1}.  
VM3={t11→t13→t8}.  
VM4={t5→t14}.  
VM5={t12→t2}.  
VM6={t3→t15}.

<i>Task</i>	<i>ET</i>	<i>Waiting time</i>	
<i>t4</i>	2	<i>VM1</i>	
<i>t10</i>	2	<i>VM2</i>	
<i>t11</i>	1.33	<i>VM3</i>	
<i>t5</i>	2	<i>VM4</i>	
<i>t12</i>	1.6	<i>VM5</i>	
<i>t3</i>	2	<i>VM6</i>	
<i>t6</i>	20	<i>Wait(2)</i>	<i>VM1</i>
<i>t9</i>	30	<i>Wait(2)</i>	<i>VM2</i>
<i>t13</i>	13.33	<i>Wait(1.33)</i>	<i>VM3</i>
<i>t14</i>	16.66	<i>Wait(2)</i>	<i>VM4</i>
<i>t2</i>	28	<i>Wait(1.6)</i>	<i>VM5</i>
<i>t15</i>	32	<i>Wait(2)</i>	<i>VM6</i>
<i>t7</i>	180	<i>Wait(22)</i>	<i>VM1</i>
<i>t1</i>	200	<i>Wait(32)</i>	<i>VM2</i>
<i>t8</i>	66.66	<i>Wait(14.66)</i>	<i>VM3</i>

# MAX-MIN

In MAX-MIN tasks are sorted based on the completion time of tasks; long tasks that take more completion time have the highest priority. Then assigned to the VM with minimum overall execution time in the VMs list.

## Advantages

- Working to exploit the available resources in an efficient manner.
- This algorithm has better performance than the FCFS, SJF, and MIN-MIN algorithms

## Disadvantages

- Increase waiting time for small and medium tasks; if we have six long tasks, in the MAX-MIN scheduling algorithm they will take priority in six VMs in the VM list, and short tasks must be waiting until the large tasks finish.

# Max-Min

Assume we have 15 tasks as in [Table 1](#) above. We will be sorting tasks in the task list as in [Table 5](#). Tasks are sorted from largest task to smallest task based on the highest completion time. They are then assigned to the VMs with minimum overall execution time in the VMs list.

Tasks	t1	t8	t7	t15	t2	t14	t13	t9	t6	t3	t12	t5	t11	t10	t4
lengths	100000	100000	90000	80000	70000	25000	20000	15000	10000	5000	4000	3000	2000	1000	1000

# Execution: Max-Min

VM6={t1→t13→t11}.

VM5={t8→t9→t10}.

VM4={t7→t6→t4}.

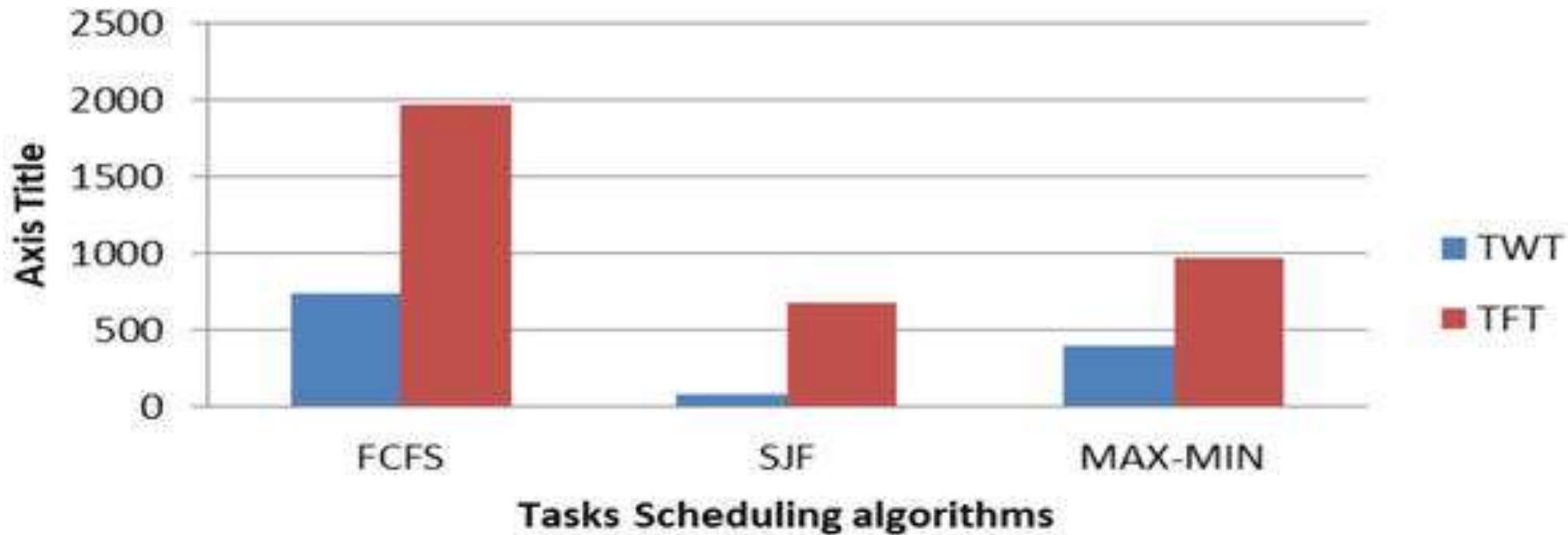
VM3={t15→t3}.

VM2={t2→t12}.

VM1={t14→t5}.

Task	ET	Waiting time	
t1	40	VM6	
t8	40	VM5	
t7	60	VM4	
t15	53.33	VM3	
t2	140	VM2	
t14	50	VM1	
t13	8	Wait(40)	VM6
t9	6	Wait(40)	VM5
t6	6.66	Wait(60)	VM4
t3	3.33	Wait(53.33)	VM3
t12	8	Wait(140)	VM2
t5	6	Wait(50)	VM1
t11	0.8	Wait(48)	VM6
t4	0.4	Wait(46)	VM5
t10	0.67	Wait(66.67)	VM4

# Result Comparison



# References

1. Ramotra A, Bala A. Task-Aware Priority Based Scheduling in Cloud Computing [master thesis]. Thapar University; 2013
2. Microsoft Azure website. [Accessed: 01 October 2017]
3. Kumar Garg S, Buyya R. Green Cloud Computing and Environmental Sustainability, Australia: Cloud Computing and Distributed Systems (CLOUDS) Laboratory Department of Computer Science and Software Engineering, The University of Melbourne; 2012
4. Al-maamari A, Omara F. Task scheduling using PSO algorithm in cloud computing environments. International Journal of Grid Distribution Computing. 2015;8(5):245-256
5. <http://www.pbenson.net/2013/04/the-cloud-defined-part-1-of-8-on-demand-self-service/> [Accessed: 01 October 2017]
6. Endo P, Rodrigues M, Gonçalves G, Kelner J, Sadok D, Curescu C. High availability in clouds: Systematic review and research challenges. Journal of Cloud Computing Advances, Systems and Applications. 2016
7. <http://www.techinmind.com/what-is-cloud-computing-what-are-its-advantages-and-disadvantages/> [Accessed: 01 October 2017]
8. <https://siliconangle.com/blog/2016/04/29/survey-sees-rapid-growth-in-enterprise-cloud-adoption/> [Accessed: 01 October 2017]
9. Aladwani, T. (2020). Types of Task Scheduling Algorithms in Cloud Computing Environment. Scheduling Problems - New Applications and Trends. doi: 10.5772/intechopen.86873

# **RAID (REDUNDANT ARRAY OF INDEPENDENT DISKS)**

Dr Hitesh Mohapatra

# AGENDA

Introduction

What is RAID?

RAID controller

RAID levels

Benefits of RAID

Summary

# WHAT IS RAID?

- RAID (redundant array of independent disks) is a way of storing the same data in different places on multiple hard disks or solid-state drives (SSDs) to protect data in the case of a drive failure.
- There are different RAID levels, however, and not all aim to provide redundancy.

# HOW RAID WORKS?

- RAID works by placing data on multiple disks and allowing input/output (I/O) operations to overlap in a balanced way, improving performance.
- Because using multiple disks increases the mean time between failures, storing data redundantly also increases fault tolerance.

# AN IMAGE OF A HARD DRIVE



# HOW IT WORKS?

- RAID arrays appear to the operating system (OS) as a single logical drive.
- RAID employs the techniques of disk mirroring or disk striping.
- Mirroring will copy identical data onto more than one drive.
- Striping partitions help spread data over multiple disk drives.
- Each drive's storage space is divided into units ranging from a sector of 512 bytes up to several megabytes.
- The stripes of all the disks are interleaved and addressed in order.
- Disk mirroring and disk striping can also be combined in a RAID array.\
- In a single-user system where large records are stored, the stripes are typically set up to be small (512 bytes, for example) so that a **single record spans all the disks and can be accessed quickly by reading all the disks at the same time.**
- In a multiuser system, better performance requires a stripe wide enough to hold the typical or maximum size record, enabling overlapped disk I/O across drives.

# RAID CONTROLLER

- A RAID controller is a device used to manage hard disk drives in a storage array.
- It can be used as a level of abstraction between the OS and the physical disks, presenting groups of disks as logical units.
- Using a RAID controller can improve performance and help protect data in case of a crash.

“ COMPUTER TECHNOLOGIES ARE  
LIKE BUSES. THERE'S ALWAYS  
ANOTHER ONE COMING. ”

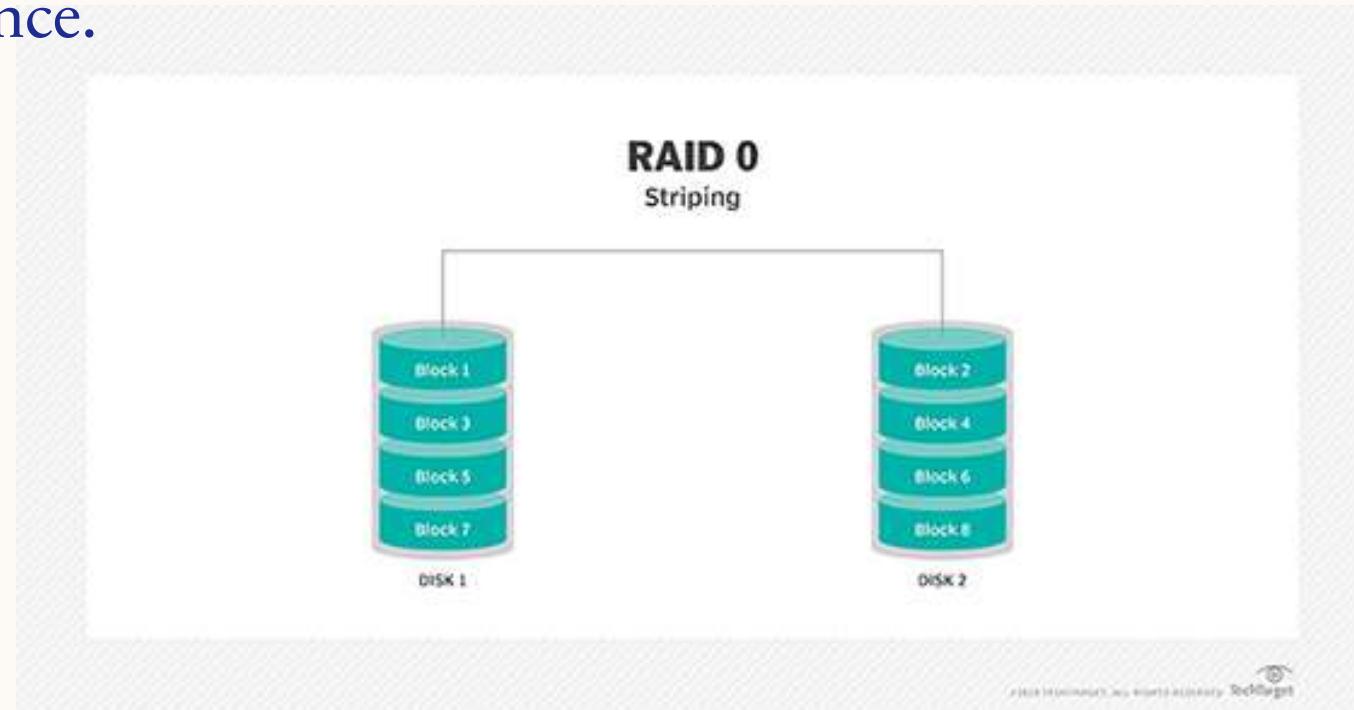
??

# RAID LEVELS

- RAID devices use different versions, called levels.
- The original paper that coined the term and developed the RAID setup concept defined six levels of RAID -- 0 through 5.
- This numbered system enabled those in IT to differentiate RAID versions.
- The number of levels has since expanded and has been broken into three categories: standard, nested and nonstandard RAID levels.

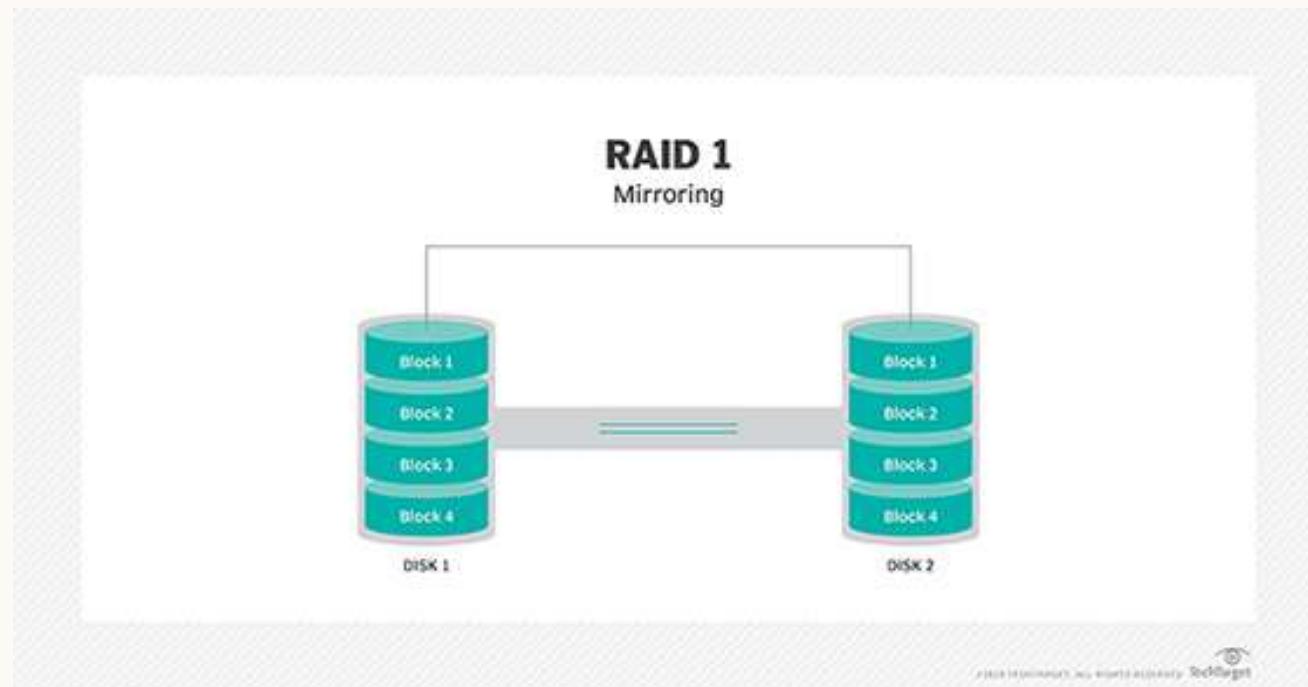
# RAID 0

- RAID 0. This configuration has striping but no redundancy of data. It offers the best performance, but it does not provide fault tolerance.



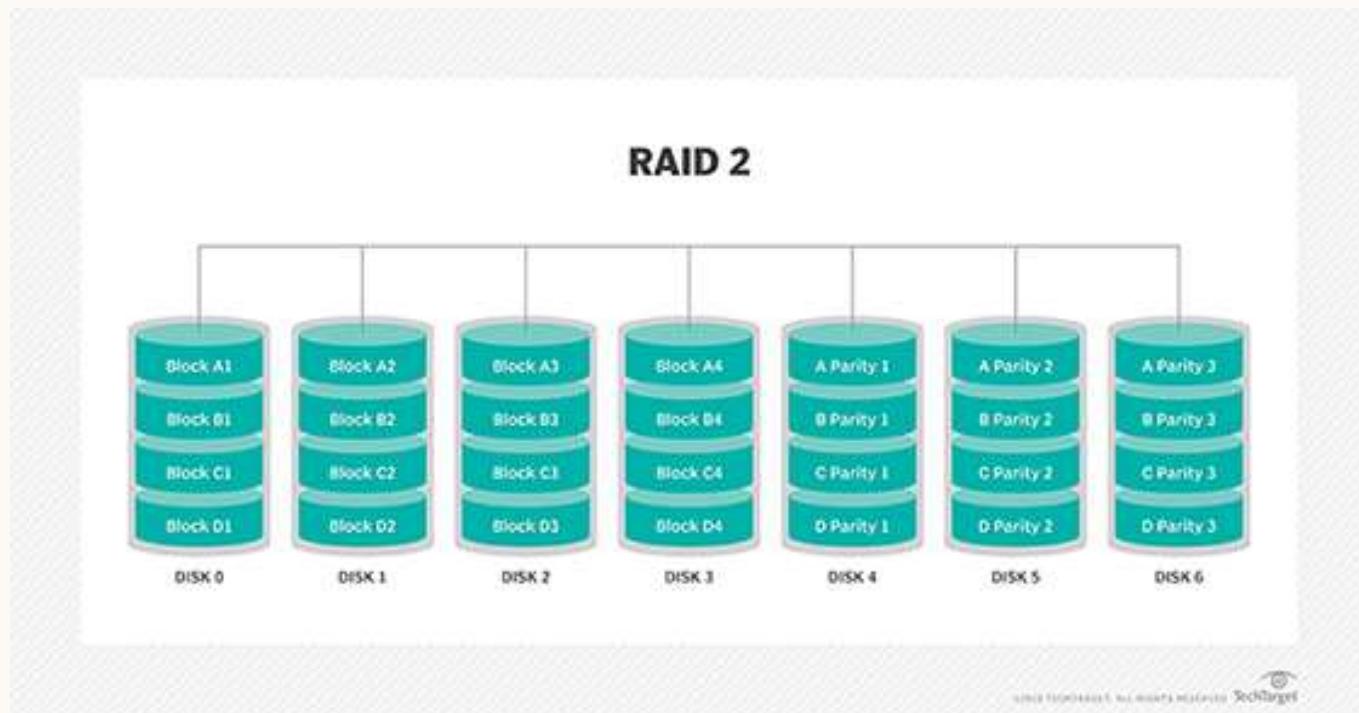
# RAID 1

- Also known as **disk mirroring**, this configuration consists of at least two drives that **duplicate the storage of data**. There is no striping. **Read performance is improved** since either disk can be read at the same time. Write performance is the same as for single-disk storage.



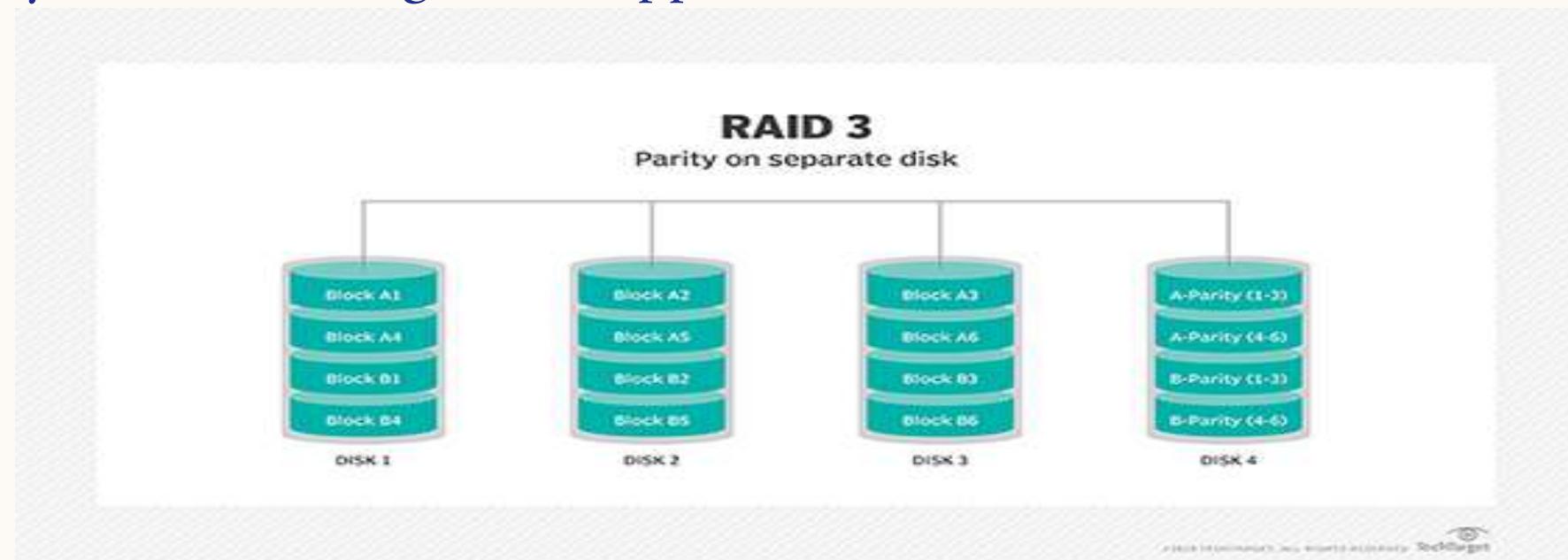
# RAID 2

- This configuration uses striping across disks, with some disks storing error checking and correcting (ECC) information. RAID 2 also uses a dedicated Hamming code parity, a linear form of ECC. RAID 2 has no advantage over RAID 3 and is no longer used.



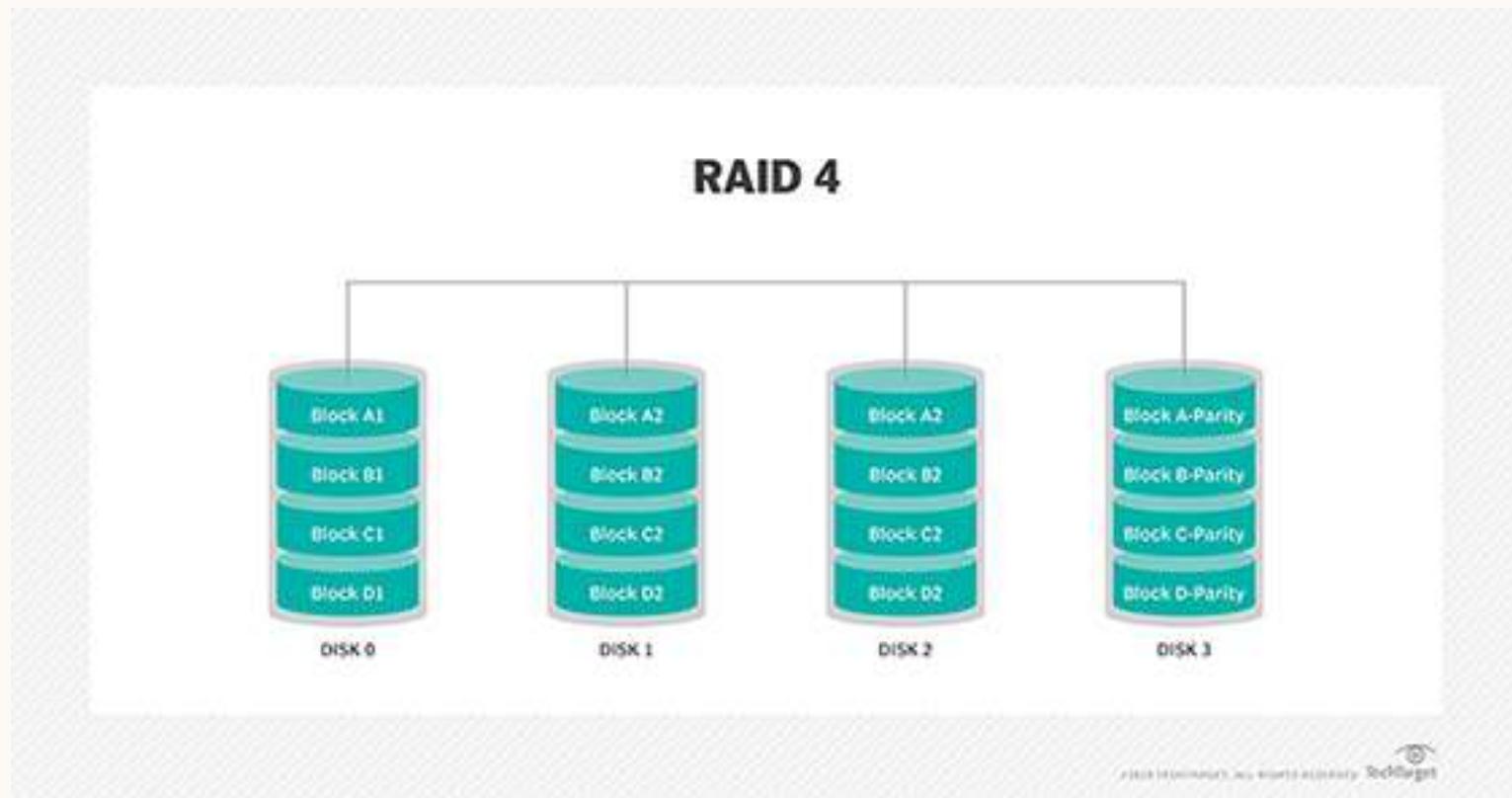
# RAID 3

- This technique uses striping and dedicates one drive to storing parity information. The embedded ECC information is used to detect errors. Data recovery is accomplished by calculating the exclusive information recorded on the other drives. Because an I/O operation addresses all the drives at the same time, RAID 3 cannot overlap I/O. For this reason, RAID 3 is best for single-user systems with long record applications.



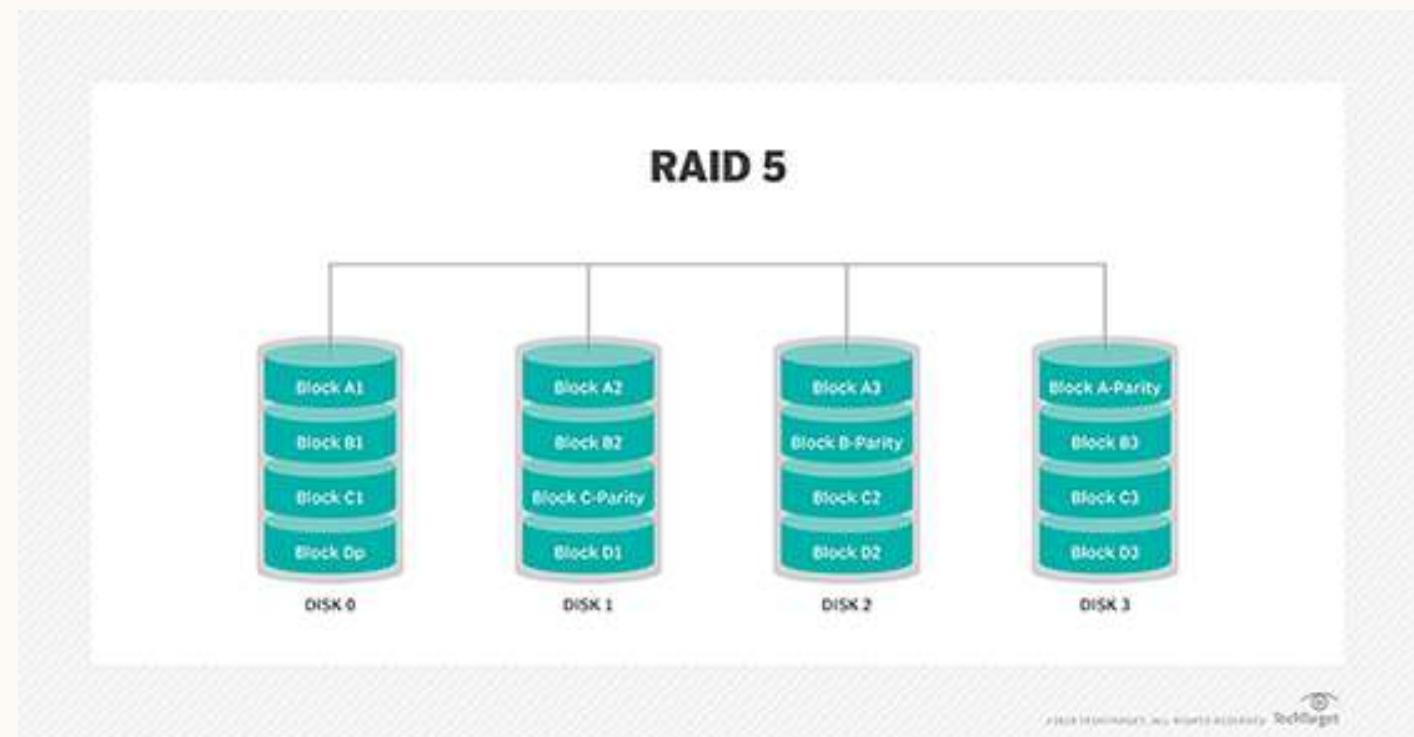
# RAID 4

- This level uses large stripes, which means a user can read records from any single drive. Overlapped I/O can then be used for read operations. Because all write operations are required to update the parity drive, no I/O overlapping is possible.



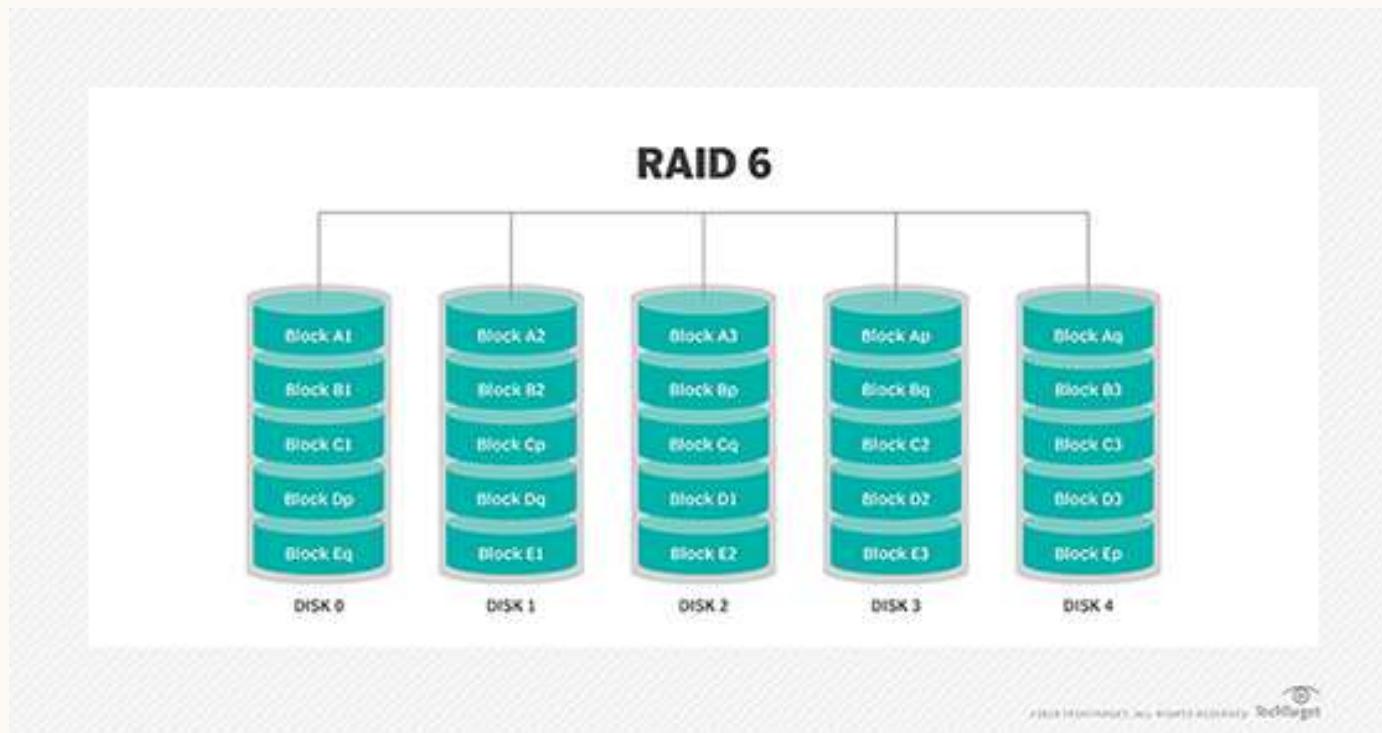
# RAID 5

- This level is based on parity block-level striping. The parity information is striped across each drive, enabling the array to function, even if one drive were to fail. The array's architecture enables read and write operations to span multiple drives. This results in performance better than that of a single drive, but not as high as a RAID 0 array. RAID 5 requires at least three disks, but it is often recommended to use at least five disks for performance reasons.

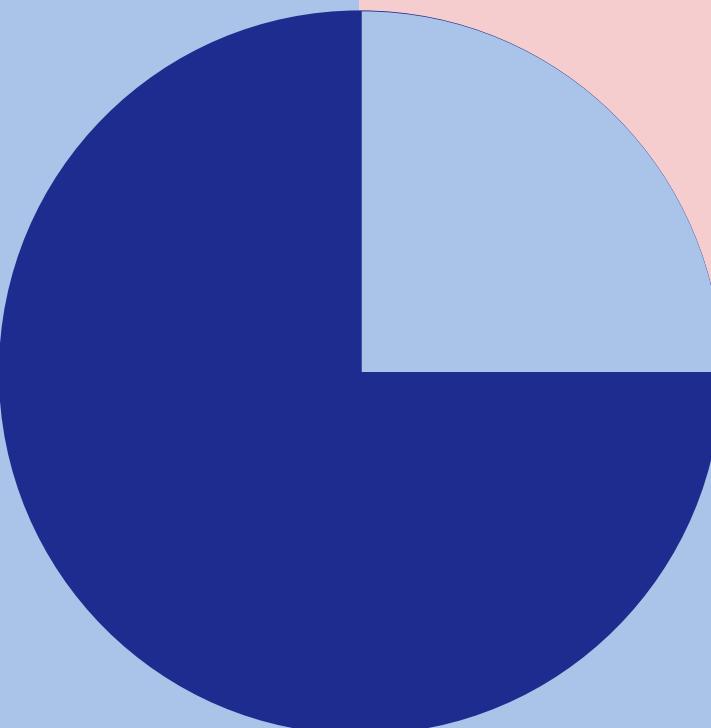


# RAID 6

- This technique is similar to RAID 5, but it includes a second parity scheme distributed across the drives in the array. The use of additional parity enables the array to continue functioning, even if two disks fail simultaneously. However, this extra protection comes at a cost. RAID 6 arrays often have slower write performance than RAID 5 arrays.

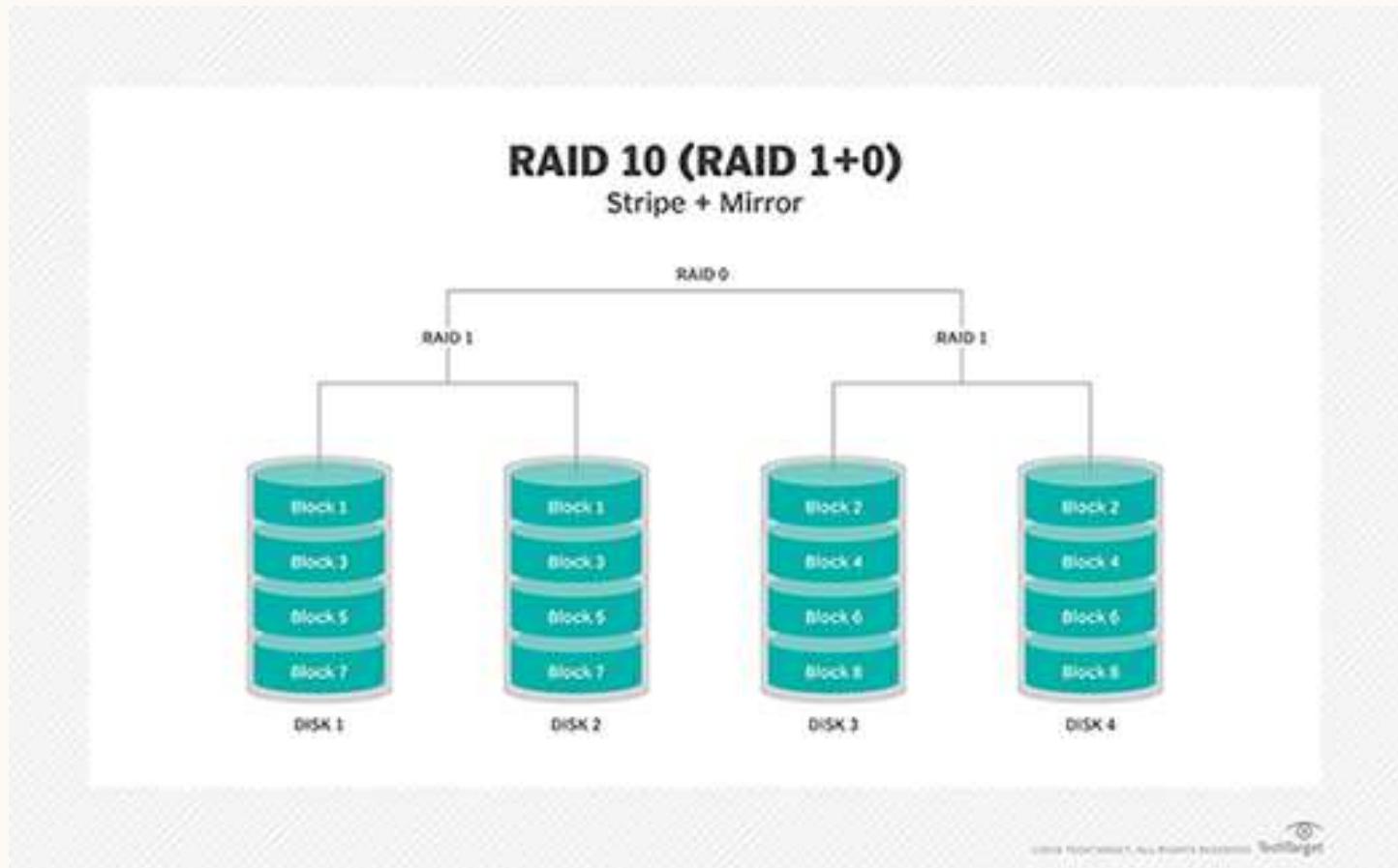


**RAID 10 (RAID 1+0).** Combining RAID 1 and RAID 0, this level is often referred to as RAID 10, which offers higher performance than RAID 1, but at a much higher cost. In RAID 1+0, the data is mirrored and the mirrors are striped.



## NESTED RAID LEVELS

- Some RAID levels that are based on a combination of RAID levels are referred to as nested RAID. Here are some examples of nested RAID levels.



# BENEFITS OF RAID

- Improved cost-effectiveness because lower-priced disks are used in large numbers.
- Using multiple hard drives enables RAID to improve the performance of a single hard drive.
- Increased computer speed and reliability after a crash, depending on the configuration.
- Reads and writes can be performed faster than with a single drive with RAID 0. This is because a file system is split up and distributed across drives that work together on the same file.
- There is increased availability and resiliency with RAID 5. With mirroring, two drives can contain the same data, ensuring one will continue to work if the other fails.

# SUMMARY

The rise of SSDs is also seen as alleviating the need for RAID. SSDs have no moving parts and do not fail as often as hard disk drives. SSD arrays often use techniques such as wear leveling instead of relying on RAID for data protection. Modern SSDs are fast enough that modern servers may not need the slight performance boost that RAID offers. However, they still may be currently used to prevent data loss.

# **THANK YOU**

hiteshmahapatra.fcs@kiit.ac.in

# *Load Balancing*

What is load balancing?

What are the benefits of load balancing?

What are load balancing algorithms?

How does load balancing work?

What are the types of load balancing?

What are the types of load balancing technology?

How can AWS help with load balancing?

# What is load balancing?

- Load balancing is the method of distributing network traffic equally across a pool of resources that support an application. Modern applications must process millions of users simultaneously and return the correct text, videos, images, and other data to each user in a fast and reliable manner.
- To handle such high volumes of traffic, most applications have many resource servers with duplicate data between them.
- A load balancer is a device that sits between the user and the server group and acts as an invisible facilitator, ensuring that all resource servers are used equally.

# What are the benefits of load balancing?

- Load balancing directs and controls internet traffic between the application servers and their visitors or clients. As a result, it improves an application's availability, scalability, security, and performance.

## Application availability

- Server failure or maintenance can increase application downtime, making your application unavailable to visitors. Load balancers increase the fault tolerance of your systems by automatically detecting server problems and redirecting client traffic to available servers. You can use load balancing to make these tasks easier:
  - Run application server maintenance or upgrades without application downtime
  - Provide automatic disaster recovery to backup sites
  - Perform health checks and prevent issues that can cause downtime

## Application scalability

- You can use load balancers to direct network traffic intelligently among multiple servers. Your applications can handle thousands of client requests because load balancing does the following:
  - Prevents traffic bottlenecks at any one server
  - Predicts application traffic so that you can add or remove different servers, if needed
  - Adds redundancy to your system so that you can scale with confidence

# Contd.

## **Application security**

- Load balancers come with built-in security features to add another layer of security to your internet applications. They are a useful tool to deal with distributed denial of service attacks, in which attackers flood an application server with millions of concurrent requests that cause server failure. Load balancers can also do the following:
- Monitor traffic and block malicious content
- Automatically redirect attack traffic to multiple backend servers to minimize impact
- Route traffic through a group of network firewalls for additional security

## **Application performance**

- Load balancers improve application performance by increasing response time and reducing network latency. They perform several critical tasks such as the following:
- Distribute the load evenly between servers to improve application performance
- Redirect client requests to a geographically closer server to reduce latency
- Ensure the reliability and performance of physical and virtual computing resources

# WHAT ARE LOAD BALANCING ALGORITHMS?

- A load balancing algorithm is the set of rules that a load balancer follows to determine the best server for each of the different client requests. Load balancing algorithms fall into two main categories.

## Static load balancing

- Static load balancing algorithms follow fixed rules and are independent of the current server state. The following are examples of static load balancing.
- ***Round-robin method***
- Servers have IP addresses that tell the client where to send requests. The IP address is a long number that is difficult to remember. To make it easy, a Domain Name System maps website names to servers. When you enter [aws.amazon.com](http://aws.amazon.com) into your browser, the request first goes to our name server, which returns our IP address to your browser.
- In the round-robin method, an authoritative name server does the load balancing instead of specialized hardware or software. The name server returns the IP addresses of different servers in the server farm turn by turn or in a round-robin fashion.
- ***Weighted round-robin method***
- In weighted round-robin load balancing, you can assign different weights to each server based on their priority or capacity. Servers with higher weights will receive more incoming application traffic from the name server.
- ***IP hash method***
- In the IP hash method, the load balancer performs a mathematical computation, called hashing, on the client IP address. It converts the client IP address to a number, which is then mapped to individual servers.

# Dynamic load balancing

- Dynamic load balancing algorithms examine the current state of the servers before distributing traffic. The following are some examples of dynamic load balancing algorithms.
- ***Least connection method***
- A connection is an open communication channel between a client and a server. When the client sends the first request to the server, they authenticate and establish an active connection between each other. In the least connection method, the load balancer checks which servers have the fewest active connections and sends traffic to those servers. This method assumes that all connections require equal processing power for all servers.
- ***Weighted least connection method***
- Weighted least connection algorithms assume that some servers can handle more active connections than others. Therefore, you can assign different weights or capacities to each server, and the load balancer sends the new client requests to the server with the least connections by capacity.
- ***Least response time method***
- The response time is the total time that the server takes to process the incoming requests and send a response. The least response time method combines the server response time and the active connections to determine the best server. Load balancers use this algorithm to ensure faster service for all users.
- ***Resource-based method***
- In the resource-based method, load balancers distribute traffic by analyzing the current server load. Specialized software called an agent runs on each server and calculates usage of server resources, such as its computing capacity and memory. Then, the load balancer checks the agent for sufficient free resources before distributing traffic to that server.

# What are the types of load balancing?

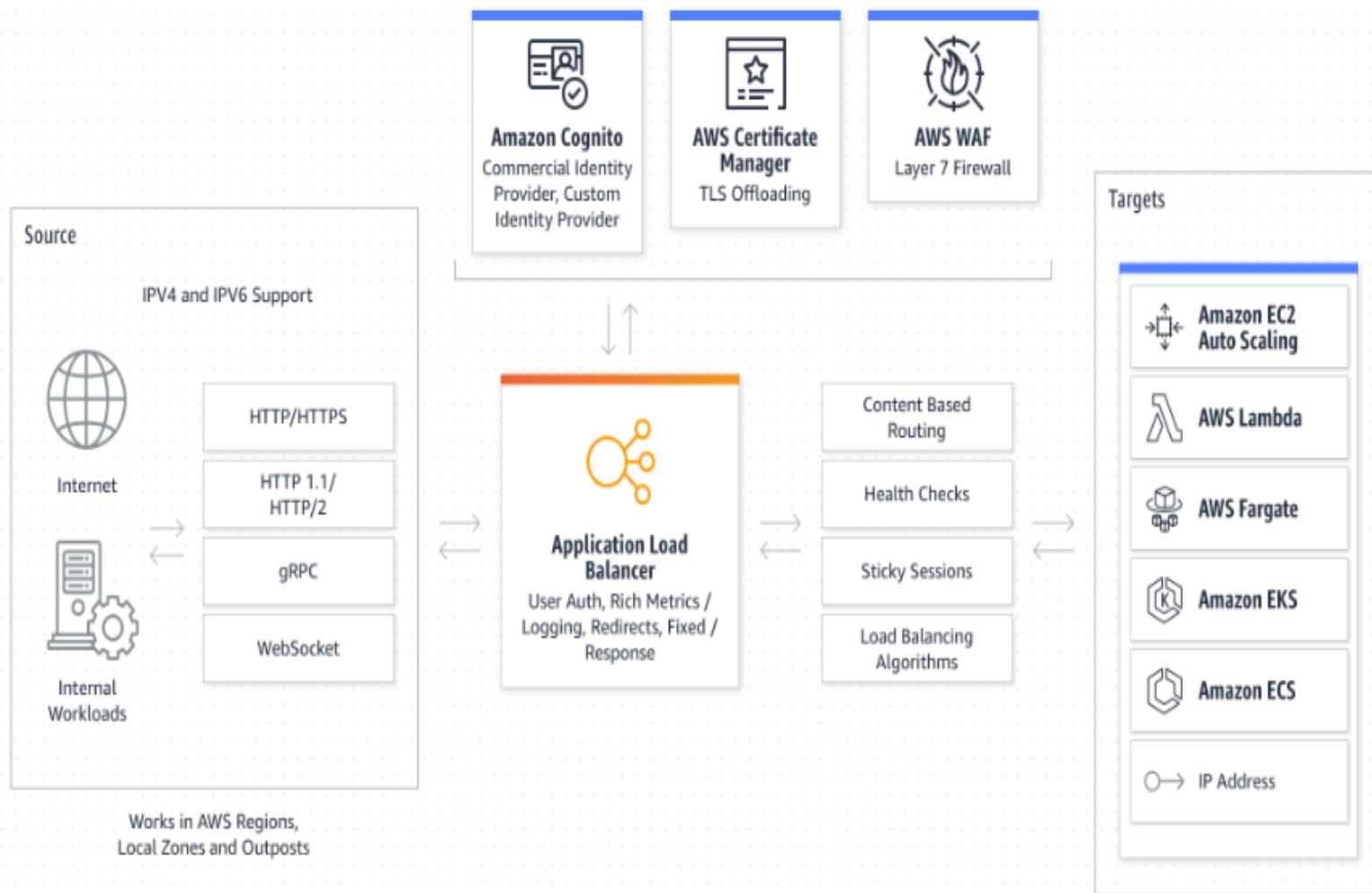
- **Application load balancing**
- Complex modern applications have several server farms with multiple servers dedicated to a single application function. Application load balancers look at the request content, such as HTTP headers or SSL session IDs, to redirect traffic.
- For example, an ecommerce application has a product directory, shopping cart, and checkout functions. The application load balancer sends requests for browsing products to servers that contain images and videos but do not need to maintain open connections. By comparison, it sends shopping cart requests to servers that can maintain many client connections and save cart data for a long time.
- **Network load balancing**
- Network load balancers examine IP addresses and other network information to redirect traffic optimally. They track the source of the application traffic and can assign a static IP address to several servers. Network load balancers use the static and dynamic load balancing algorithms described earlier to balance server load.
- **Global server load balancing**
- Global server load balancing occurs across several geographically distributed servers. For example, companies can have servers in multiple data centers, in different countries, and in third-party cloud providers around the globe. In this case, local load balancers manage the application load within a region or zone. They attempt to redirect traffic to a server destination that is geographically closer to the client. They might redirect traffic to servers outside the client's geographic zone only in case of server failure.
- **DNS load balancing**
- In DNS load balancing, you configure your domain to route network requests across a pool of resources on your domain. A domain can correspond to a website, a mail system, a print server, or another service that is made accessible through the internet. DNS load balancing is helpful for maintaining application availability and balancing network traffic across a globally distributed pool of resources.

# **What are the types of load balancing technology?**

- **Hardware load balancers**
- A hardware-based load balancer is a hardware appliance that can securely process and redirect gigabytes of traffic to hundreds of different servers. You can store it in your data centers and use virtualization to create multiple digital or virtual load balancers that you can centrally manage.
- **Software load balancers**
- Software-based load balancers are applications that perform all load balancing functions. You can install them on any server or access them as a fully managed third-party service.
- **Comparison of hardware balancers to software load balancers**
- Hardware load balancers require an initial investment, configuration, and ongoing maintenance. You might also not use them to full capacity, especially if you purchase one only to handle peak-time traffic spikes. If traffic volume increases suddenly beyond its current capacity, this will affect users until you can purchase and set up another load balancer.
- In contrast, software-based load balancers are much more flexible. They can scale up or down easily and are more compatible with modern cloud computing environments. They also cost less to set up, manage, and use over time.

# How does load balancing work?

- Companies usually have their application running on multiple servers. Such a server arrangement is called a server farm.
- User requests to the application first go to the load balancer. The load balancer then routes each request to a single server in the server farm best suited to handle the request.
- Load balancing is like the work done by a manager in a restaurant. Consider a restaurant with five waiters. If customers were allowed to choose their waiters, one or two waiters could be overloaded with work while the others are idle. To avoid this scenario, the restaurant manager assigns customers to the specific waiters who are best suited to serve them.



# AWS Lambda

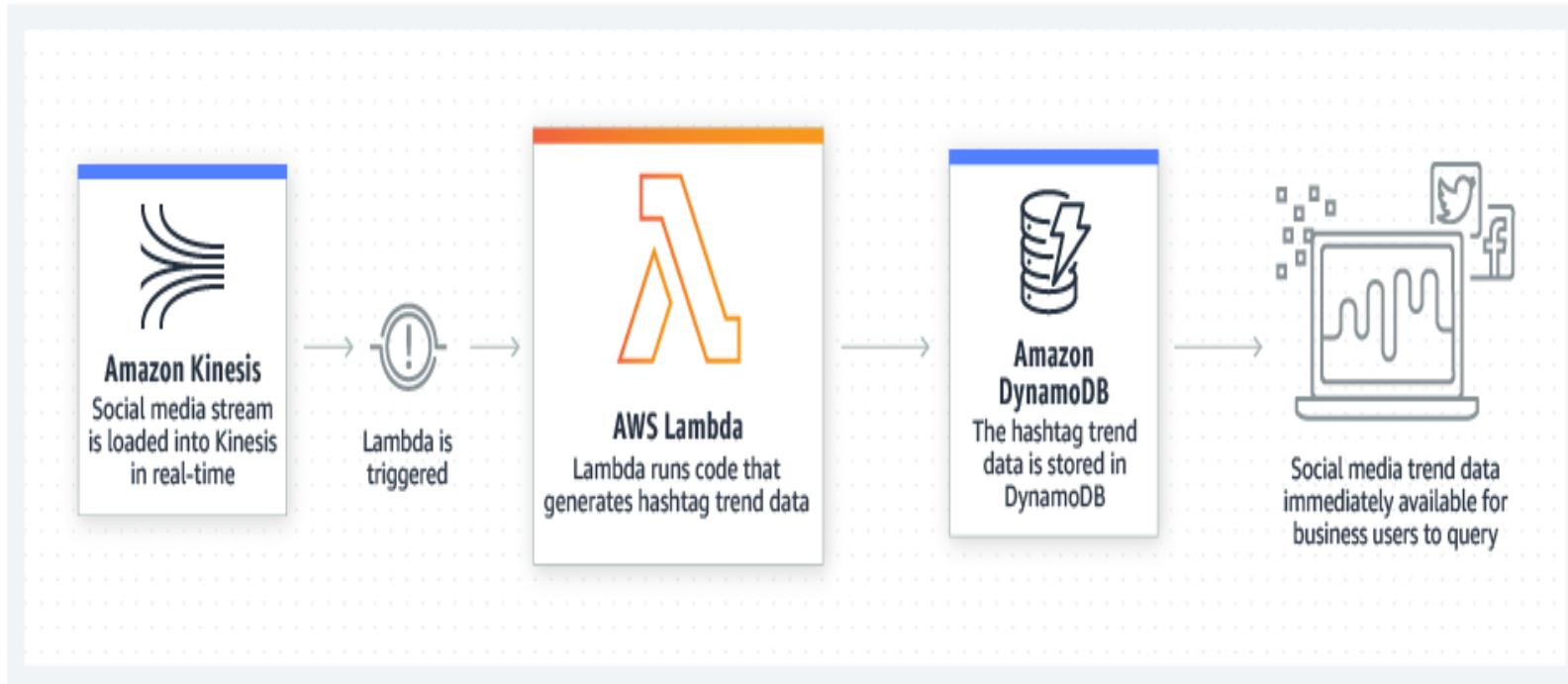
## Run code without thinking about servers or clusters

- AWS Lambda is a serverless, event-driven compute service that lets you run code for virtually any type of application or backend service without provisioning or managing servers.
- File processing
- Stream Processing
- Web application'
- IOT backends
- Mobile backends

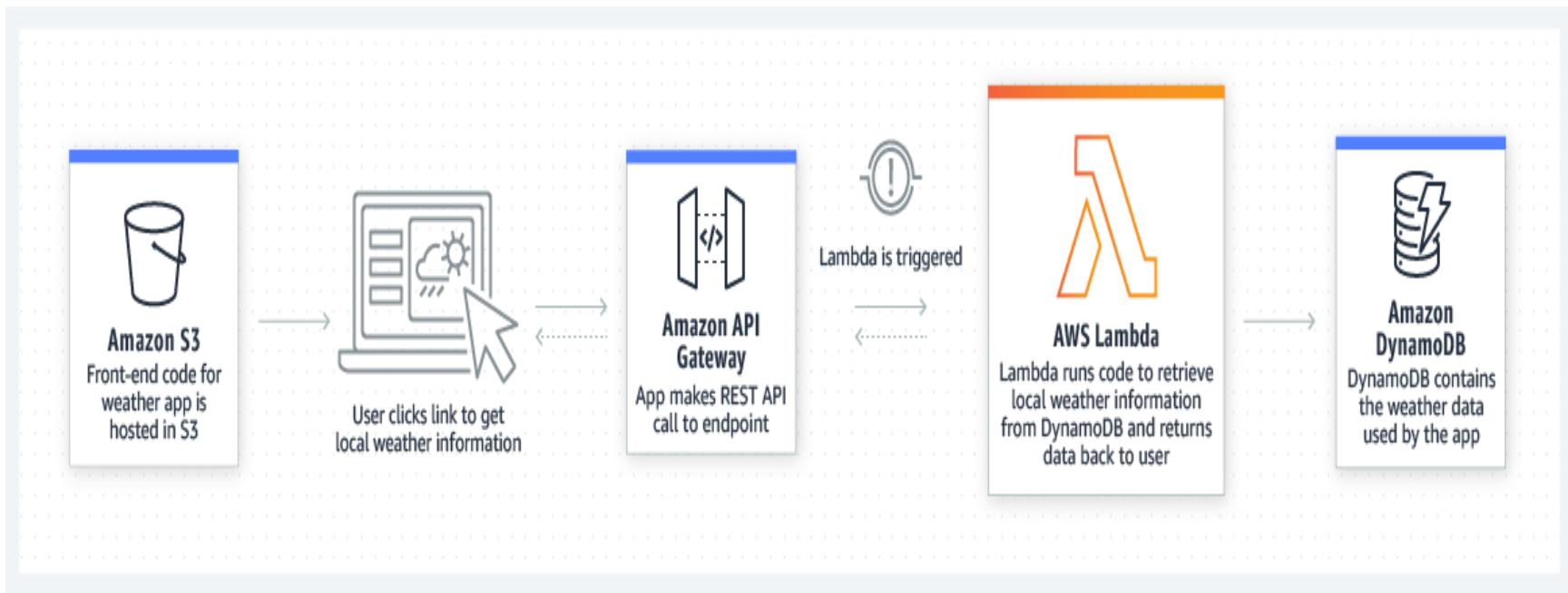
# File Processing



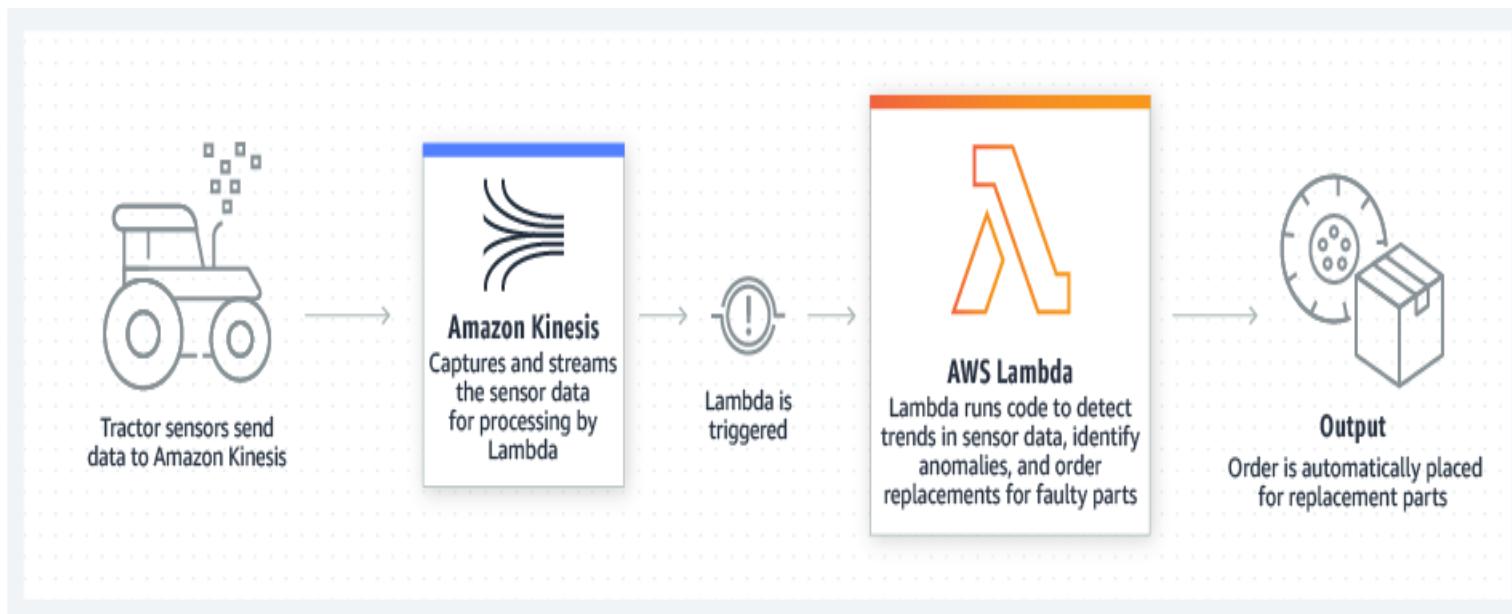
# Stream Processing



# Web application



# IOT backends



# Mobile backends



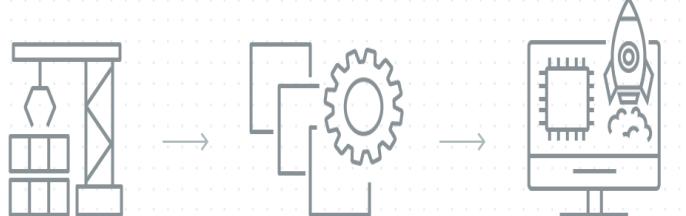
# AWS Fargate

## Serverless compute for containers

### How it works ?

- AWS Fargate is a serverless, pay-as-you-go compute engine that lets you focus on building applications without managing servers. AWS Fargate is compatible with both Amazon Elastic Container Service (ECS) and Amazon Elastic Kubernetes Service (EKS).

### Without Fargate



Build your container image

Define and deploy the EC2 Instances

Provision and manage compute and memory resources



Isolate applications in separate VMs



Run and manage both applications and infrastructure



Pay for EC2 Instances

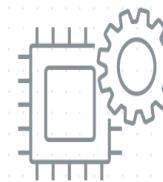
### With Fargate



AWS Fargate



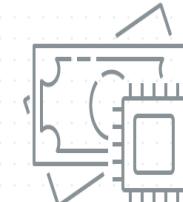
Build container image



Define memory and compute resources required



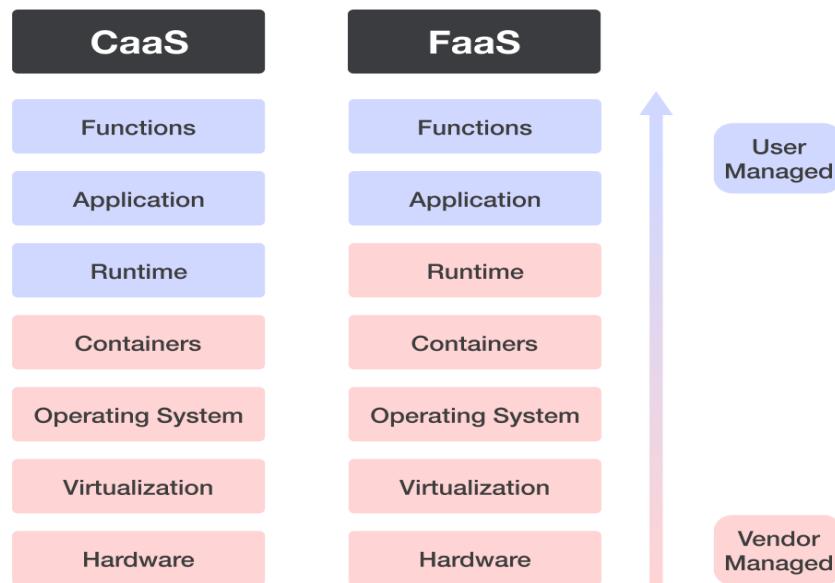
Run and manage applications



Pay for requested compute resources when used.  
Application isolation by design

# What is the difference between EC2 Lambda and Fargate?

- While Fargate is a Container as a Service (CaaS) offering, AWS Lambda is a Function as a Service (FaaS offering). Therefore, Lambda functions do not necessarily need to be packaged into containers, making it easier to get started with Lambda. But if you have containerized applications, Fargate is the way to go.



# How can AWS help with load balancing?

- Elastic Load Balancing (ELB) is a fully managed load balancing service that automatically distributes incoming application traffic to multiple targets and virtual appliances across AWS and on-premises resources.
- You can use it to scale modern applications without complex configurations or API gateways.
- You can use ELB to set up four different types of software load balancers.
- An Application Load Balancer routes traffic for HTTP-based requests.
- A Network Load Balancer routes traffic based on IP addresses. It is ideal for balancing TCP and User Datagram Protocol (UDP)-based requests.
- A Gateway Load Balancer routes traffic to third-party virtual appliances. It is ideal for incorporating a third-party appliance, such as a network firewall, into your network traffic in a scalable and easy-to-manage way.
- A Classic Load Balancer routes traffic to applications in the Amazon EC2-Classic network—a single, flat network that you share with other customers.

# Example

- For example, [Terminix](#), a global pest control brand, uses Gateway Load Balancer to handle 300% more throughput.
- [Second Spectrum](#), a company that provides artificial intelligence-driven tracking technology for sports broadcasts, uses AWS Load Balancer Controller to reduce hosting costs by 90%.
- [Code.org](#), a nonprofit dedicated to expanding access to computer science in schools, uses Application Load Balancer to handle a 400% spike in traffic efficiently during online coding events.

# References

- <https://aws.amazon.com/blogs/containers/amazon-ecs-vs-amazon-eks-making-sense-of-aws-container-services/>
- <https://aws.amazon.com/ecs/>
- <https://aws.amazon.com/eks/>
- <https://bluexp.netapp.com/blog/aws-cvo-blg-aws-ecs-vs-eks-6-key-differences#:~:text=4.-,Portability,support%20for%20portability%20of%20workloads.>
- <https://docs.aws.amazon.com/prescriptive-guidance/latest/patterns/deploy-a-grpc-based-application-on-an-amazon-eks-cluster-and-access-it-with-an-application-load-balancer.html>