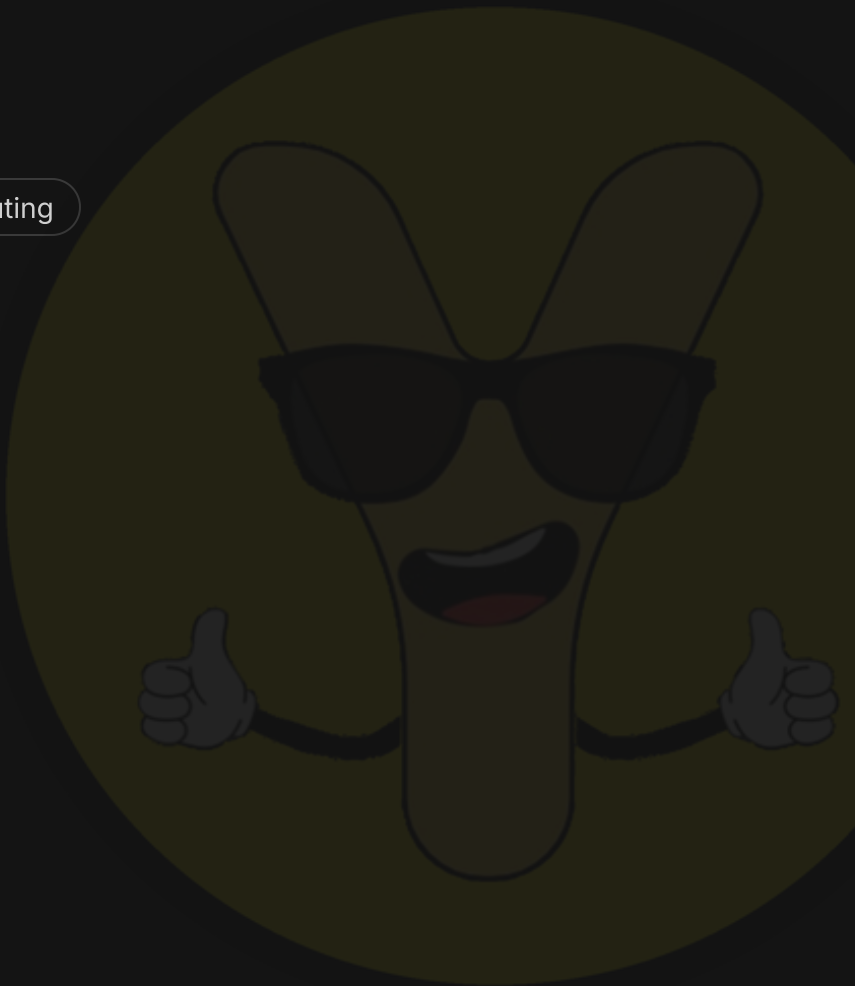
AKTU · 4TH YEAR · SEMESTER 7 · BCS071

# Cloud Computing

## Unit 1 — Introduction To Cloud Computing

- Definition of Cloud
- Evolution of Cloud Computing
- Underlying Principles of Parallel and Distributed Computing
- Cloud Characteristics
- Elasticity in Cloud
- On-demand Provisioning




DECODED BY

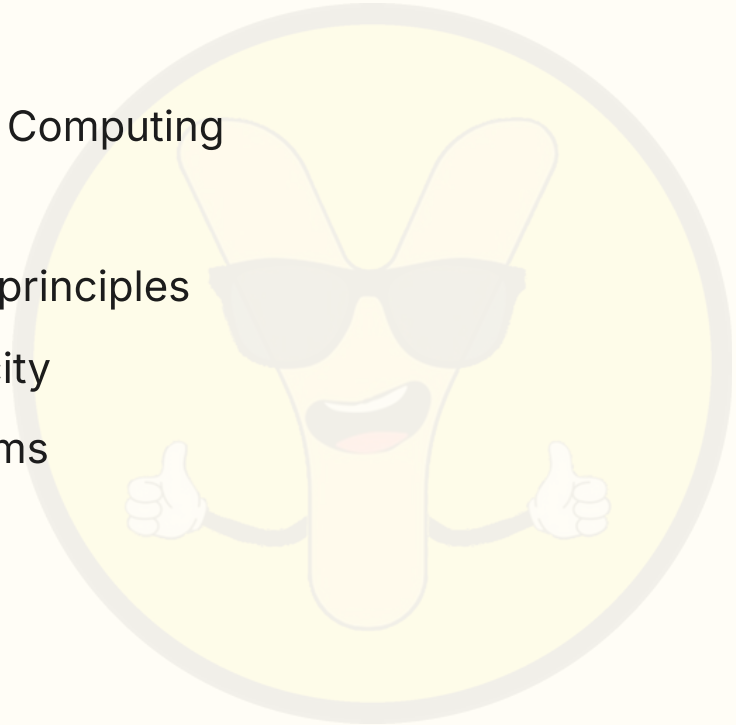
**Chirag Jain**

Founder, Ycotes · Gold Medalist in Artificial Intelligence, IIT Madras · taught 10,000+ students

# Unit 1: Introduction to Cloud Computing

Welcome to Cloud Computing. This lecture covers foundational definitions, evolution, paradigms, and core operational models.

- Definition and NIST model of Cloud Computing
- Evolution from Mainframe to Cloud
- Parallel and Distributed Computing principles
- Essential characteristics and elasticity
- On-demand provisioning mechanisms



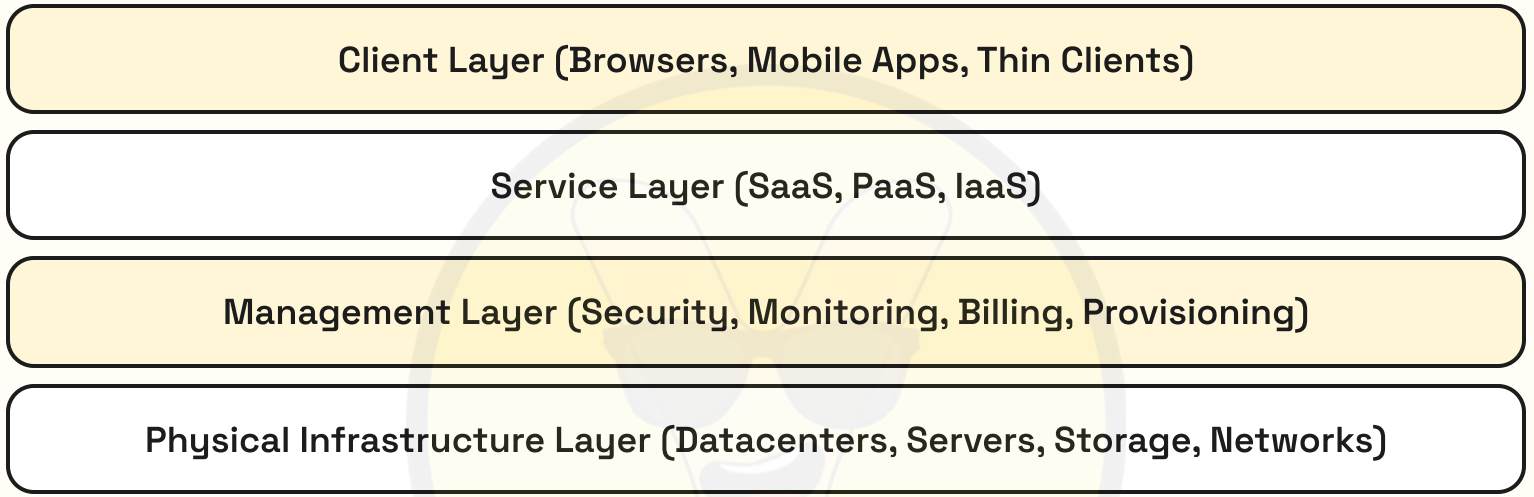
# Definition of Cloud Computing

**CLOUD COMPUTING**

A model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (networks, servers, storage, applications) that can be rapidly provisioned and released with minimal management effort.

- NIST (National Institute of Standards and Technology) standard definition.
- Shifts IT from capital expenditure (CapEx) to operational expenditure (OpEx).
- Abstracts underlying physical hardware complexity from the end user.

# High-Level Cloud Architecture




Layered view of Cloud Computing architecture showing abstraction from physical hardware to end-user clients.

# Evolution of Cloud Computing

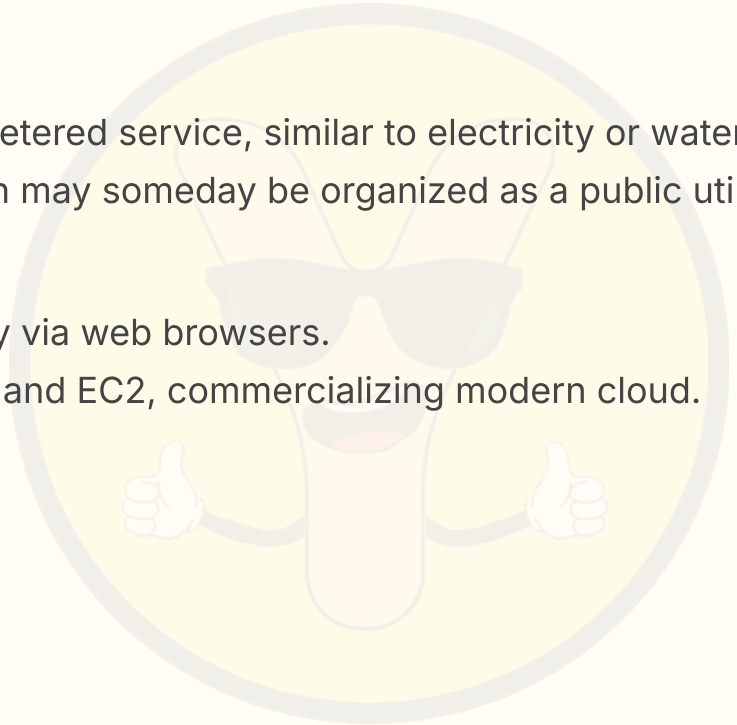
Cloud computing evolved over decades through advancements in virtualization, internet bandwidth, and utility pricing models.

- Mainframe Computing (1950s-1960s):
  - Centralized processing with dumb terminals.
  - Time-sharing allowed multiple users to share a single mainframe.
- Client-Server Architecture (1980s-1990s):
  - Shift to decentralized desktop computers.
  - Workloads split between client requests and server responses.
- Cluster and Grid Computing (Late 1990s):
  - Clusters linked homogeneous computers for parallel tasks.
  - Grids linked heterogeneous resources across organizations for large-scale scientific problems.

# Evolution: From Utility Computing to Cloud

The final stepping stones before modern cloud computing.

- Utility Computing (Early 2000s):
  - Computing resources packaged as a metered service, similar to electricity or water.
  - Coined by John McCarthy: computation may someday be organized as a public utility.
- SaaS and Web 2.0 (Mid 2000s):
  - Salesforce introduced software delivery via web browsers.
  - Amazon launched AWS (2006) with S3 and EC2, commercializing modern cloud.



# Comparison of Computing Paradigms

| Paradigm | Resource Scope | Coupling | Primary Focus |
|---|---|---|---|
| **Mainframe** | Single central system | Tight | Centralized batch processing |
| **Cluster** | Local area network (LAN) | Tightly coupled | High-performance parallel tasks |
| **Grid** | Wide area network (WAN) | Loosely coupled | Distributed scientific computing |
| **Cloud** | Global Internet | On-demand/Elastic | Utility-based scalable service delivery |

# Underlying Principles: Parallel Computing

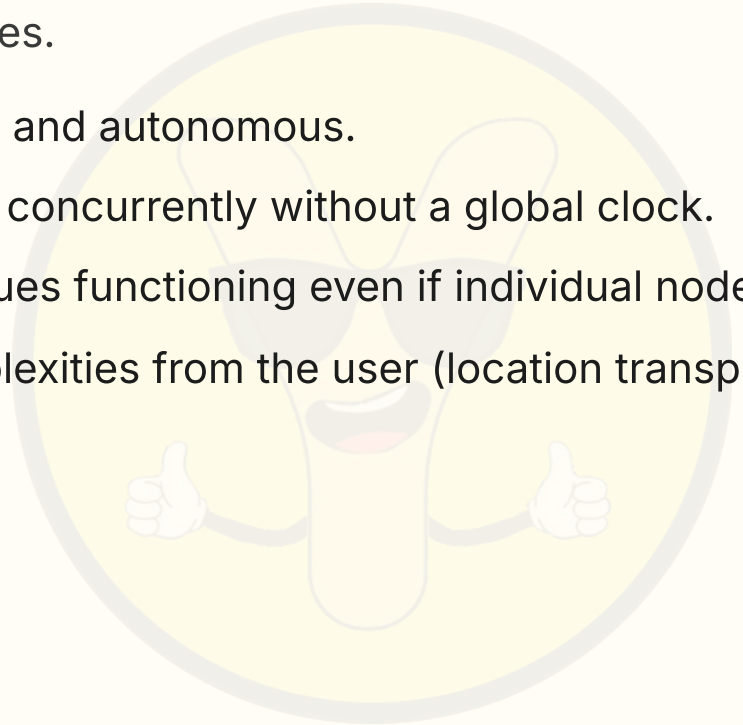
Parallel computing executes multiple instructions simultaneously to reduce execution time for large problems.

- Involves a single computer with multiple cores, processors, or symmetric multiprocessing (SMP).
- Shared Memory Architecture: All processors access a global physical address space; communication is fast but scalability is limited by memory contention.
- Distributed Memory Architecture: Each processor has its own private memory; processors communicate via message passing (e.g., MPI).

# Underlying Principles: Distributed Computing

Distributed computing connects multiple autonomous computers across a network that communicate and coordinate actions by passing messages.

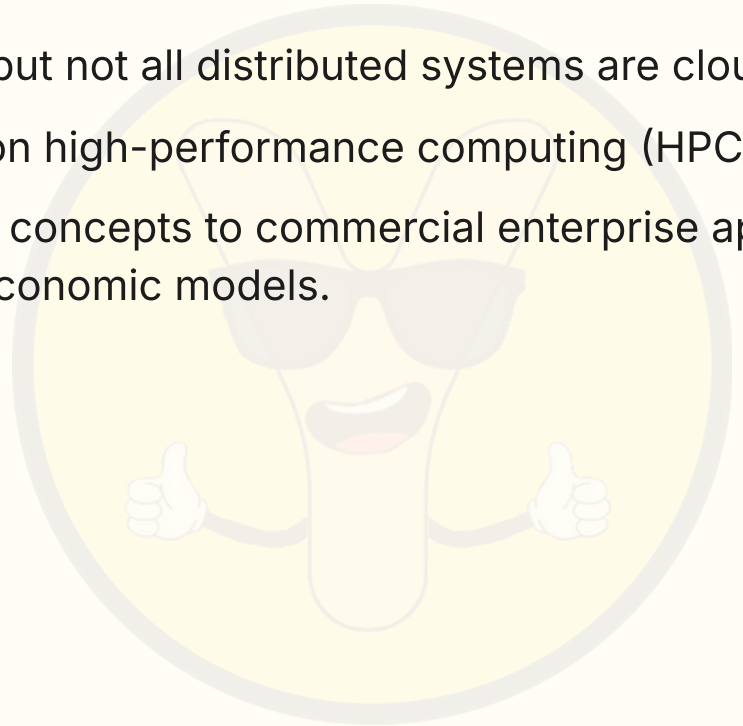
- Resources are physically separated and autonomous.
- Concurrency: Components operate concurrently without a global clock.
- Fault Tolerance: The system continues functioning even if individual nodes fail.
- Transparency: Hides network complexities from the user (location transparency, failure transparency).



# Relationship: Distributed, Grid, Cluster, and Cloud

Cloud computing builds directly upon the foundations laid by distributed and parallel systems.

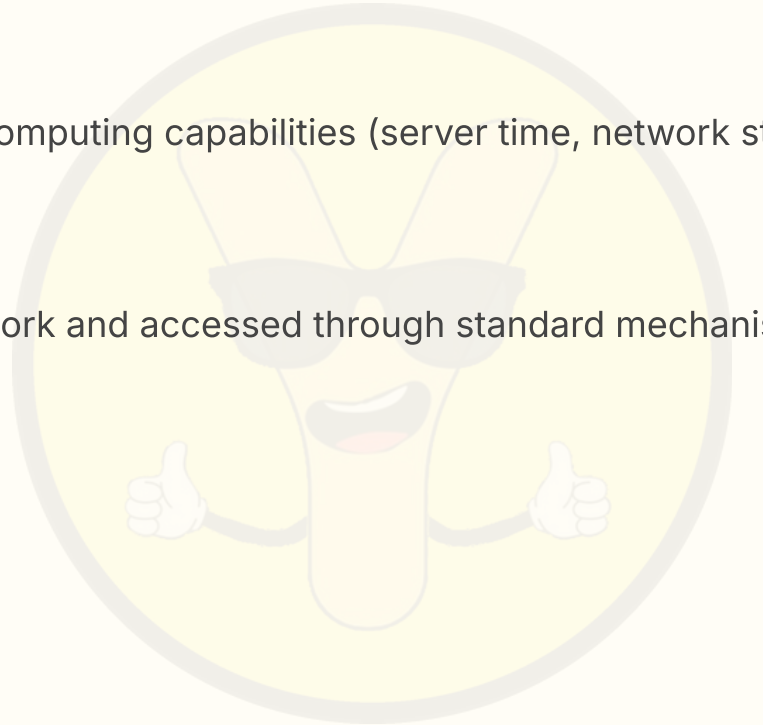
- All clouds are distributed systems, but not all distributed systems are clouds.
- Clusters and Grids focus primarily on high-performance computing (HPC) for science.
- Cloud computing generalizes these concepts to commercial enterprise applications with multi-tenancy, virtualization, and pay-as-you-go economic models.



# Essential Characteristics of Cloud Computing

NIST defines five essential characteristics that distinguish cloud computing from traditional IT hosting.

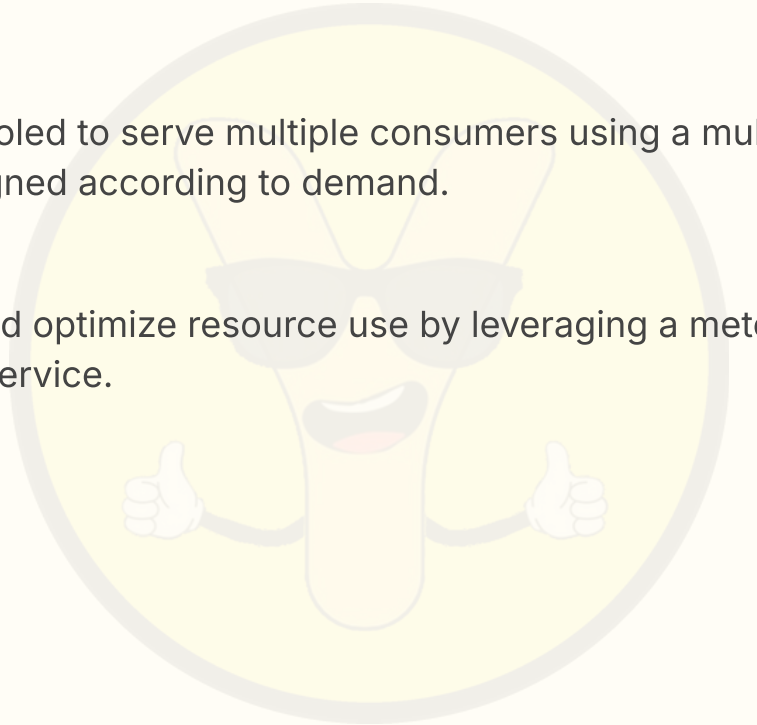
- 1. On-Demand Self-Service:
  - Consumers can unilaterally provision computing capabilities (server time, network storage) automatically without human interaction with the service provider.
- 2. Broad Network Access:
  - Capabilities are available over the network and accessed through standard mechanisms promoting use by heterogeneous thin or thick client platforms.



# Essential Characteristics (Contd.)

Resource pooling, rapid elasticity, and measured service.

- 3. Resource Pooling:
  - Provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned according to demand.
- 4. Measured Service:
  - Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service.

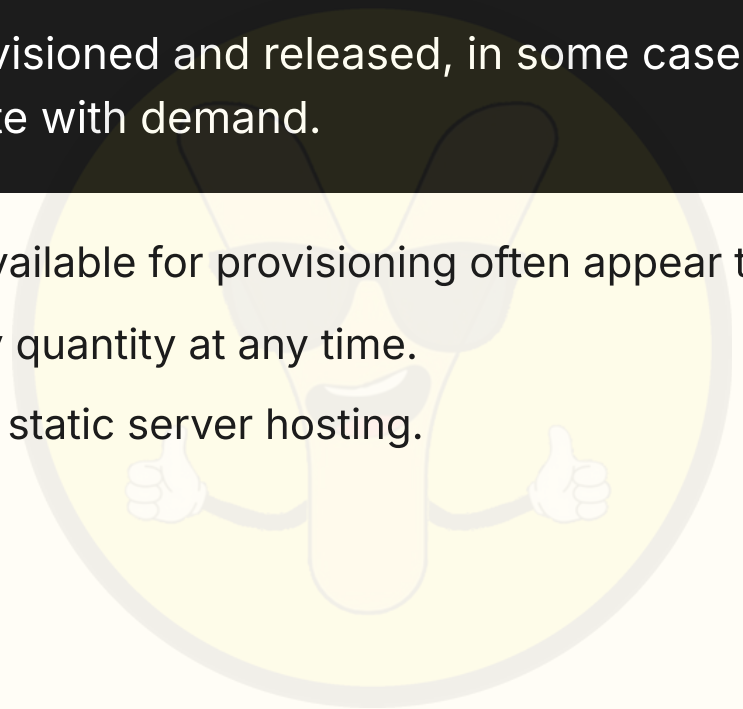


# Essential Characteristic: Rapid Elasticity

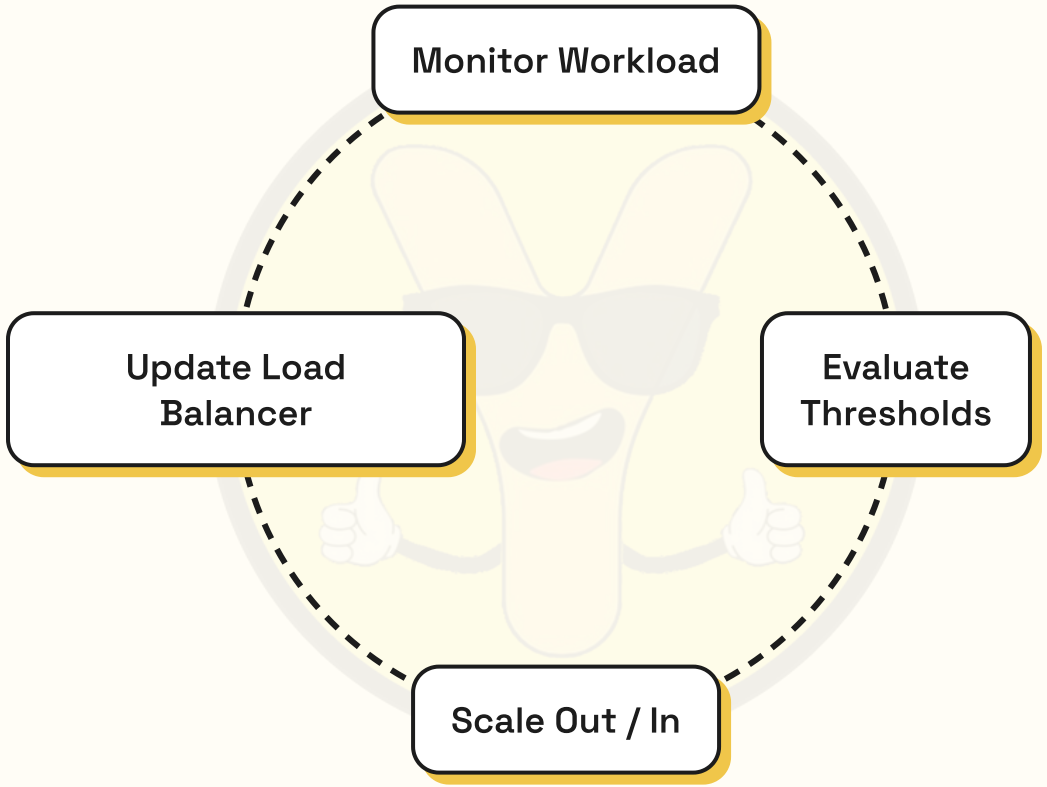
**RAPID ELASTICITY**

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.
- Resources can be purchased in any quantity at any time.
- Differentiates cloud from traditional static server hosting.



# Elastic Scaling Cycle

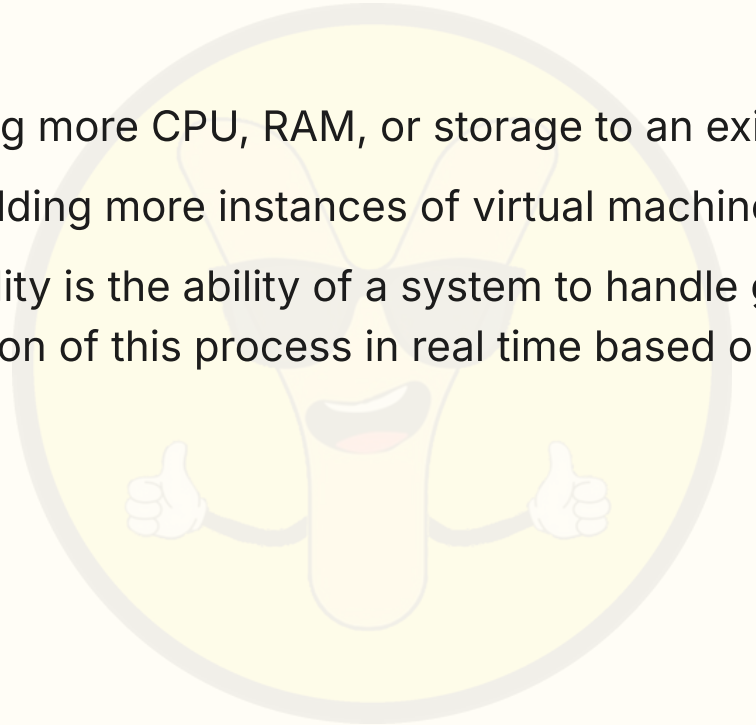



Continuous feedback loop governing dynamic elasticity in cloud environments.

# Deep Dive into Elasticity in Cloud

Elasticity is the degree to which a system adapts to workload changes by provisioning and de-provisioning resources automatically.

- Scaling Up (Vertical Scaling): Adding more CPU, RAM, or storage to an existing virtual machine instance.
- Scaling Out (Horizontal Scaling): Adding more instances of virtual machines to handle increased traffic load.
- Distinction from Scalability: Scalability is the ability of a system to handle growing workloads by adding resources; elasticity is the automation of this process in real time based on demand.



# Worked Example: Elasticity Cost-Benefit Analysis

An e-commerce portal experiences peak traffic during a 6-hour flash sale requiring 50 servers instead of the baseline 10 servers. Static provisioning requires 50 servers 24/7 at \(\\)0.10$/hr per server. Cloud elastic provisioning costs $\$0.10$/hr for 50 servers during the 6-hour sale, and $\$0.10$/hr for 10 servers during the remaining 18 hours. Calculate daily savings.

1. Static Daily Cost = \(50 \text{ servers} \times 24 \text{ hours} \times \\)0.10 = \$120.00$
2. Cloud Peak Cost = \(50 \text{ servers} \times 6 \text{ hours} \times \\)0.10 = \$30.00$
3. Cloud Off-Peak Cost = \(10 \text{ servers} \times 18 \text{ hours} \times \\)0.10 = \$18.00$
4. Total Cloud Daily Cost = \(\\)30.00 + \$18.00 = \$48.00$
5. Daily Savings = \$120.00 - \$48.00 = \$72.00

**Daily savings using elastic cloud provisioning amount to $72.00 per day.**

# On-Demand Provisioning

**ON-DEMAND PROVISIONING**

The automated setup, configuration, and deployment of cloud computing resources via APIs or web dashboards without requiring manual intervention from the service provider's personnel.

- Reduces deployment time from weeks/months (physical hardware procurement) to minutes/seconds.
- Eliminates over-provisioning and under-provisioning risks.
- Enables DevOps pipelines and Infrastructure as Code (IaC).

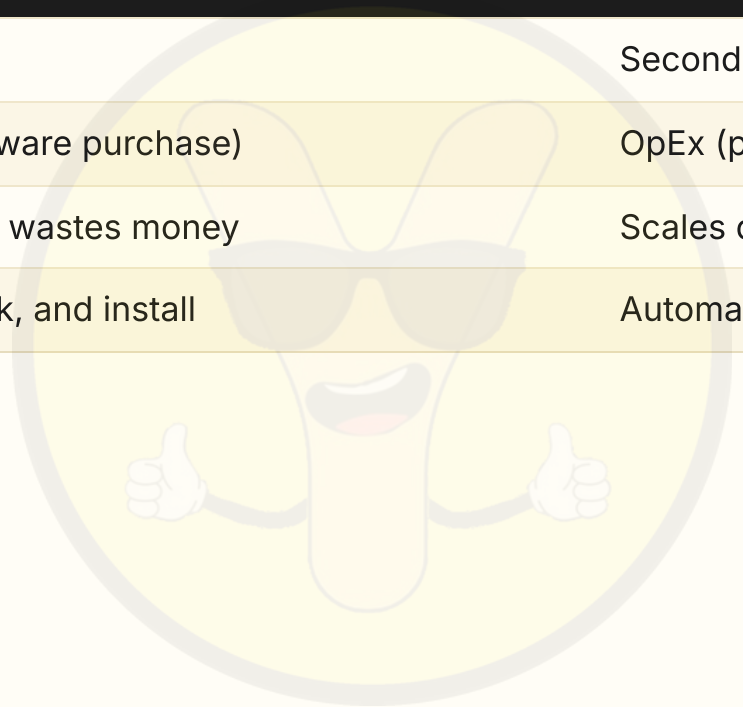
# Mechanism of On-Demand Provisioning

How cloud providers deliver instantaneous resource allocation behind the scenes.

- Virtualization Layer: Hypervisors (KVM, Xen, VMware) slice physical servers into isolated Virtual Machines (VMs).
- API Gateway: RESTful APIs receive user requests for compute, storage, or networking.
- Orchestration Engine: Automates resource placement, IP allocation, security group configuration, and storage mounting.
- Billing & Metering Integration: Instantly updates usage tracking databases for pay-as-you-go billing.

# Static Provisioning vs On-Demand Provisioning

| Feature | Static Provisioning | On-Demand Provisioning |
|---|---|---|
| **Lead Time** | Weeks or months | Seconds or minutes |
| **Cost Model** | High CapEx (hardware purchase) | OpEx (pay-as-you-go utility) |
| **Resource Risk** | Over-provisioning wastes money | Scales dynamically with demand |
| **Management** | Manual rack, stack, and install | Automated via APIs and consoles |



# Exam corner

- **Q1** Define Cloud Computing according to NIST and explain its five essential characteristics.
- **Q2** Trace the evolution of cloud computing from mainframe architecture to modern utility and cloud computing.
- **Q3** Differentiate between parallel computing and distributed computing with suitable examples.
- **Q4** What is rapid elasticity? Explain the difference between horizontal scaling (scaling out) and vertical scaling (scaling up).
- **Q5** Explain the concept of on-demand provisioning and how it differs from traditional static IT resource allocation.
- **Q6** Solve a cost comparison problem between static server allocation and elastic cloud provisioning for fluctuating workloads.

# Unit 1 done. You are exam-ready.

Revise the Exam corner once more, then open Unit 2 on theycotes.com. Every AKTU subject, unit by unit, free. Notes by Chirag Jain, Ycotes.

