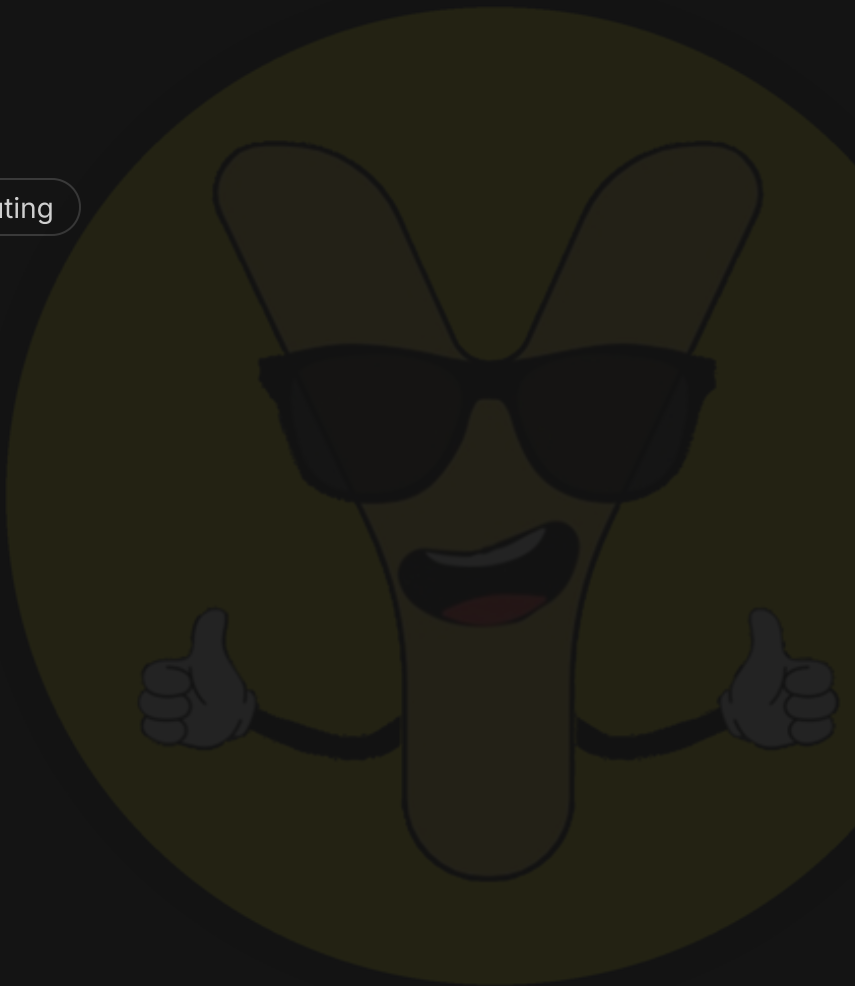
AKTU · 3RD YEAR · SEMESTER 5 · BCAM051

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

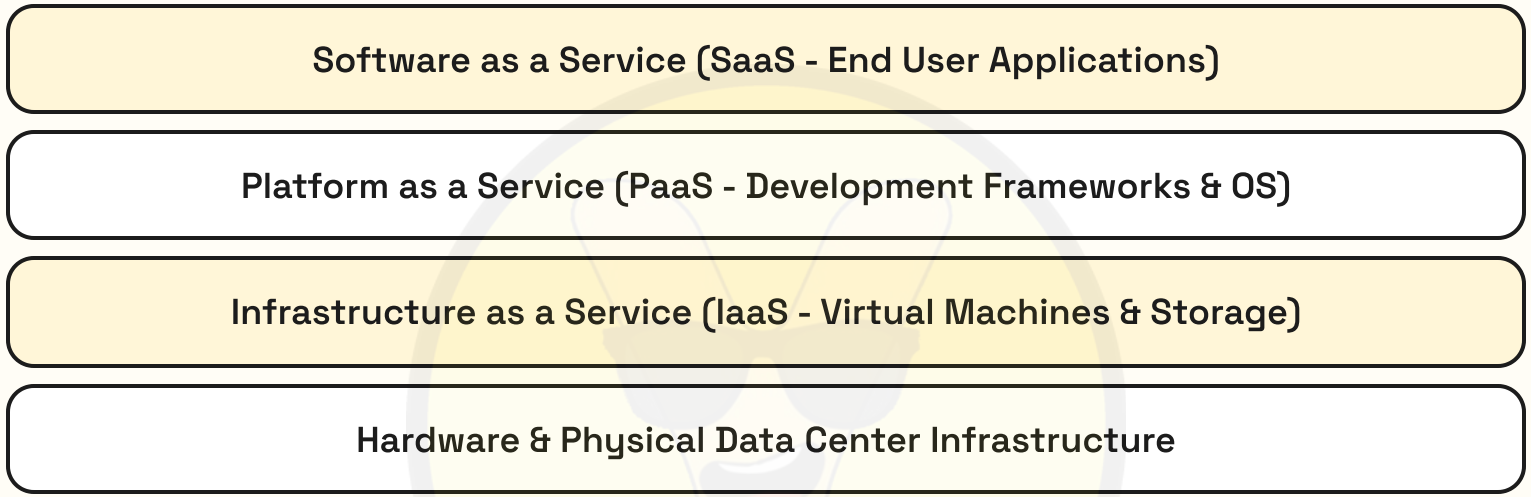
Definition of Cloud Computing

CLOUD COMPUTING

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

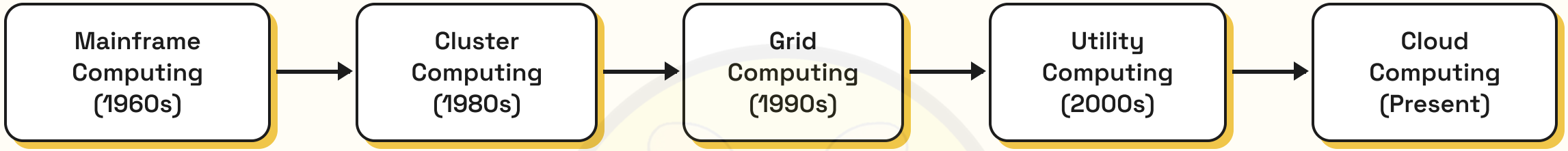
- Standard definition standardized by NIST (National Institute of Standards and Technology).
- Replaces heavy upfront capital expenditure (CapEx) with operational expenditure (OpEx).
- Delivers computing as a utility over the Internet rather than local physical infrastructure.

Architectural Layers of Cloud Services



Layered Service Architecture of Cloud Computing

Evolution of Cloud Computing



Chronological Evolution from Centralized Mainframes to Modern Cloud Infrastructure

Key Milestones in Cloud Evolution

Cloud computing evolved by integrating concepts from distributed systems, virtualization, and web service technologies.

- Mainframe Era: Shared central processing power using dumb terminals with time-sharing OS.
- Cluster Computing: Connecting low-cost standalone computers over LANs to act as a single system.
- Grid Computing: Combining geographically dispersed heterogeneous resources across multiple domains.
- Utility Computing: Packaging computing resources as a metered service, akin to electricity or water.
- Virtualization Era: Introduction of hypervisors enabling multi-tenant isolated VM execution on shared hardware.

Principles of Parallel Computing

Parallel computing executes multiple concurrent instructions simultaneously by splitting tasks across processors.

- Tightly Coupled Systems: Processors share physical memory and communicate via high-speed system buses.
- Granularity: Ratio of computation to communication; classified into fine-grained and coarse-grained.
- Hardware Models: Includes Symmetric Multiprocessing (SMP) and Non-Uniform Memory Access (NUMA).
- Flynn's Taxonomy: Classifies architectures into SISD, SIMD, MISD, and MIMD execution models.
- Speedup Ratio: Quantified as $S = \frac{T_1}{T_N}$, where T_1 is sequential execution time and T_N is parallel time on N processors.

Amdahl’s Law for Parallel Processing

Calculate the maximum theoretical speedup $S(N)$ of an application running on 8 parallel processors if 20% of the program execution is strictly sequential.

1. Identify parallel fraction $P = 0.80$ and sequential fraction $(1 - P) = 0.20$.
2. Identify total processor count $N = 8$.
3. Apply Amdahl's Law formula: $S(N) = \frac{1}{(1-P)+\frac{P}{N}}$.
4. Substitute values: $S(8) = \frac{1}{0.20+\frac{0.80}{8}} = \frac{1}{0.20+0.10} = \frac{1}{0.30}$.

Maximum theoretical speedup $S(8) \approx 3.33\times$ over single-processor execution.

Principles of Distributed Computing

Distributed systems connect independent, autonomous computing nodes over network infrastructure.

- Loosely Coupled Systems: Independent nodes with private memory communicating via network messages.
- Distribution Transparency: Hides system distribution complexity (Location, Access, Failure, Migration transparency).
- Scalability: Ability to grow horizontally by adding nodes without degrading overall system throughput.
- Concurrency: Independent execution of processes across autonomous machines simultaneously.
- Fault Tolerance: Ability of system to continue functioning despite node crashes or network partitions.

Comparison: Parallel vs Distributed Computing

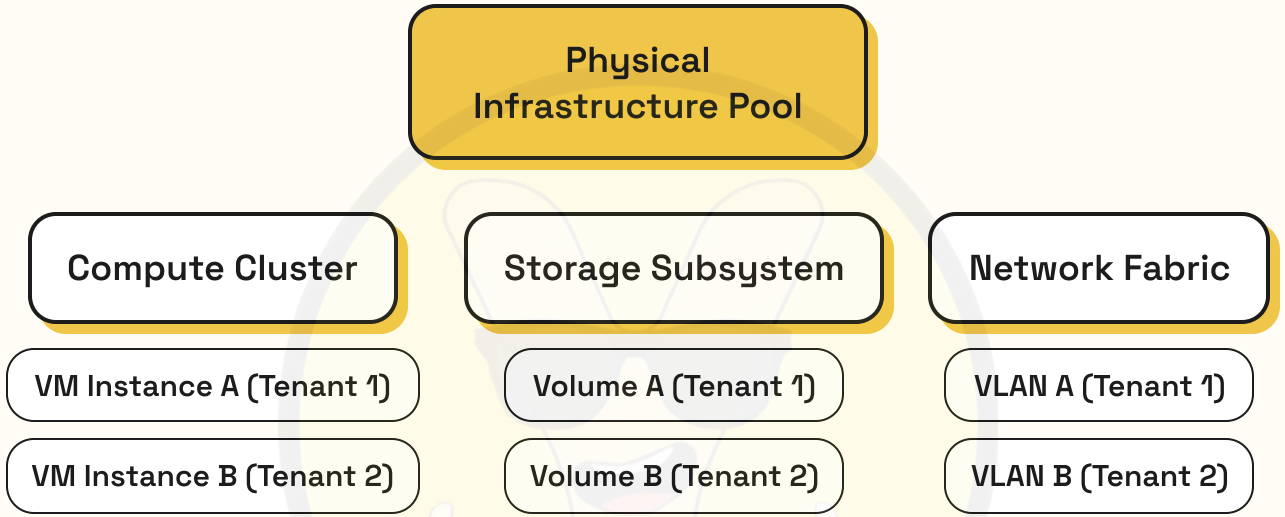
Parameter	Parallel Computing	Distributed Computing
Memory Architecture	Shared or closely coupled RAM	Distributed / Private memory per node
System Coupling	Tightly coupled processors	Loosely coupled autonomous nodes
Communication	System bus / Shared memory inter-process	Network protocols (TCP/IP, HTTP, RPC)
Synchronization	Hardware clock / Mutex / Locks	Logical clocks (Lamport) / Message passing
Fault Tolerance	Low (single node failure stops execution)	High (system survives node crashes)
Primary Objective	High-performance execution speed	Resource sharing & high availability

Essential Cloud Characteristics (NIST)

NIST defines five core technical characteristics mandatory for any valid cloud deployment.

- On-Demand Self-Service: Automatic provision of resources without requiring human provider intervention.
- Broad Network Access: Capabilities available over standard networks accessed via heterogeneous clients.
- Resource Pooling: Provider resources pooled to serve multiple consumers using a multi-tenant model.
- Rapid Elasticity: Dynamic provisioning and release of resources to scale outward and inward with demand.
- Measured Service: Resource usage automatically monitored, controlled, optimized, and billed transparently.

Multi-Tenant Resource Pooling Hierarchy



Resource Allocation and Multi-Tenant Isolation in Cloud Pools

Broad Network Access & Measured Service

Cloud platforms rely on accessible interface standards and granular telemetry for pay-per-use billing.

- Heterogeneous Client Support: Accessible from laptops, mobile devices, workstations, and IoT gateways.
- Standard Communication Protocols: Interaction via web protocols such as HTTPS, REST APIs, and SOAP.
- Metered Resource Usage: System tracks resource utilization at a fine-grained level per tenant.
- Key Metrics Monitored: CPU compute hours, storage size (GB/TB), network bandwidth, and API request count.
- Usage Transparency: Provides detailed audit logs and cost dashboards to consumers and billing systems.

Concept of Cloud Elasticity

CLOUD ELASTICITY

Elasticity is the degree to which a cloud system is capable of adapting to workload changes by provisioning and de-provisioning computing resources in an automated dynamic manner, matching resource allocation to real-time resource demand.

- Prevents Under-provisioning: Avoids system failure and performance degradation during load spikes.
- Prevents Over-provisioning: Avoids paying for idle, unutilized computing capacity.
- Driven by real-time telemetry metrics and predefined auto-scaling rules.

Comparison: Scalability vs Elasticity

Parameter	Scalability	Elasticity
Primary Goal	Handling long-term, predictable growth	Handling short-term workload fluctuations
Time Horizon	Long-term planning (weeks/months)	Real-time / Dynamic (seconds/minutes)
Automation Level	Can be manual or automated	Fully automated provisioning and teardown
Metric Focus	Maximum peak capacity limit	Exact alignment of capacity with live demand
Operational Focus	Prevents infrastructure bottlenecks	Optimizes pay-as-you-go operational cost
Scaling Types	Vertical (Scale Up) & Horizontal (Scale Out)	Automatic Horizontal Scale-Out & Scale-In

Measuring Cloud Elasticity Latency

A web app auto-scaler provisions 8 extra virtual machines in 40 seconds during a traffic surge, and releases them in 60 seconds when traffic normalizes. Calculate total adaptation latency.

1. Identify scale-out provisioning latency: $T_{provision} = 40$ s.
2. Identify scale-in de-allocation latency: $T_{deallocate} = 60$ s.
3. Apply total adaptation latency equation: $T_{adaptation} = T_{provision} + T_{deallocate}$.
4. Calculate: $T_{adaptation} = 40 + 60 = 100$ seconds.

Total system adaptation latency for the elastic scaling operation is 100 seconds.

Types of Scaling Mechanisms

Cloud elasticity operates through two fundamental dimensional scaling strategies.

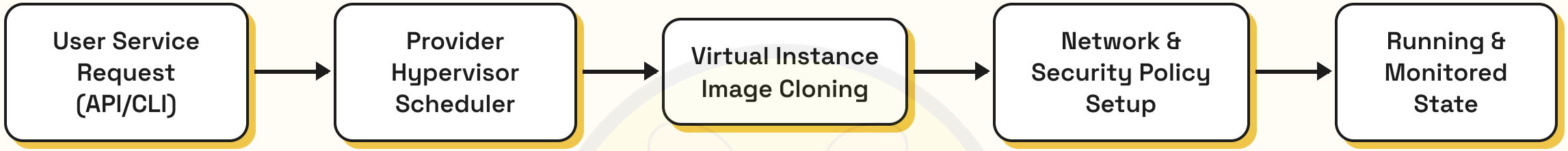
- Vertical Scaling (Scale Up / Down): Adding or removing resources (RAM, CPU) to an existing single node.
- Vertical Scaling Limits: Bound by physical hardware limits; often requires VM downtime/reboot.
- Horizontal Scaling (Scale Out / In): Adding or removing discrete computing instances to a resource pool.
- Horizontal Scaling Advantage: Offers virtual unlimited capacity with zero downtime operations.
- Load Balancer Integration: Horizontal scaling requires load balancers to distribute incoming web traffic.

On-Demand Provisioning Concept

On-demand provisioning grants users rapid, self-service access to resources as required.

- Instant Resource Delivery: Provision virtual servers, storage, and networks in minutes instead of weeks.
- Self-Service Portals: Managed via web consoles, Command Line Interfaces (CLI), or Infrastructure as Code (IaC).
- Dynamic Allocation: Resources requested on-the-fly when application execution metrics breach limits.
- CapEx to OpEx Shift: Removes physical hardware procurement costs; converts spending to subscription fees.
- Rapid De-provisioning: Unused resources are instantly shut down and removed to stop billing.

On-Demand Provisioning Workflow



Automated On-Demand Virtual Resource Provisioning Lifecycle

Auto-Scaling & Provisioning Strategies

Cloud management systems use explicit control strategies to automate resource provisioning.

- Reactive Provisioning: Triggers resource creation after a performance threshold (e.g., CPU > 80%) is breached.
- Proactive / Predictive Provisioning: Uses historical trend data and machine learning to pre-allocate capacity before spikes.
- Schedule-Based Provisioning: Scales resources up or down according to known time patterns (e.g., business hours).
- Resource Pre-Warming: Keeping warm VM pools ready to eliminate cold-start boot latency during sudden bursts.
- Over-Provisioning Buffer: Maintaining a small safety margin of idle capacity to handle instantaneous traffic spikes.

Summary: Benefits vs Challenges of Cloud Computing

Dimension	Key Benefits	Key Challenges
Financial Model	Zero upfront CapEx; pay-as-you-go OpEx	Risk of unexpected usage billing spikes
Agility & Speed	Rapid provisioning in minutes	Vendor lock-in & complex migration
Resource Scale	Near-infinite horizontal elasticity	Network latency over public channels
Infrastructure	Provider handles maintenance & hardware	Lack of low-level hardware control
System Reliability	Built-in redundancy & disaster recovery	Outage risks due to provider failure
Security	Enterprise-grade security mechanisms	Shared-tenant data privacy concerns

Exam Corner: Important AKTU Questions

- **Q1** Define Cloud Computing according to NIST. Explain the 5 essential characteristics of Cloud Computing in detail.
- **Q2** Differentiate between Parallel Computing and Distributed Computing with respect to memory architecture, coupling, and fault tolerance.
- **Q3** State Amdahl's Law. Calculate theoretical speedup for an application running on 16 processors with 10% sequential code.
- **Q4** What is Cloud Elasticity? Clearly differentiate between Scalability and Elasticity in cloud systems.
- **Q5** Explain the process of On-Demand Provisioning. Compare Reactive Provisioning with Predictive Provisioning.
- **Q6** Compare Horizontal Scaling (Scale Out/In) and Vertical Scaling (Scale Up/Down) with suitable use-cases.
- **Q7** Trace the evolutionary path of Cloud Computing from Mainframes, Cluster Computing, Grid Computing to Utility Computing.

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.

