Cloud Computing

Short notes for B.Tech CSE / IT · BCA · MCA · M.Tech

Uncodemy Handwritten Notes · free for students

1. Introduction

- **Cloud computing** = delivering computing services (servers, storage, databases, software) over the internet, on demand, pay-as-you-go.
- You rent resources instead of buying and running your own hardware.

➤ NIST: 5 essential characteristics

1. **On-demand self-service**: get resources without talking to the provider.
2. **Broad network access**: use from any device over the network.
3. **Resource pooling**: many customers share the same physical resources (multi-tenancy).
4. **Rapid elasticity**: scale up or down quickly.
5. **Measured service**: usage is metered and billed.

> **REMEMBER** NIST model = 5 characteristics + 3 service models + 4 deployment models.

➤ Evolution

- Mainframes → cluster computing → grid computing → utility computing → cloud computing.

Point	Grid computing	Cloud computing
Ownership	Many organisations share	Usually one provider
Goal	Solve big jobs together	Sell services on demand
Billing	Mostly free / shared	Pay-per-use
Virtualisation	Less used	Core technology

➤ Benefits and challenges

- Benefits: low upfront cost, scalability, high availability, global reach, no hardware maintenance.

- Challenges: security and privacy, vendor lock-in, network dependence, compliance, downtime.

2. Service Models

Model	Provider manages	You manage	Examples
IaaS	Hardware, network, virtualisation	OS, runtime, apps, data	AWS EC2, Azure VMs, Google Compute Engine
PaaS	Above + OS + runtime	Apps and data	Google App Engine, Heroku, Azure App Service
SaaS	Everything	Only your use and data	Gmail, Google Docs, Salesforce

```
SaaS   | app ready to use       | least control
PaaS   | platform to build apps |
IaaS   | raw VMs, storage, net  | most control
------ hardware in data centre ------
```

- Other "as a service": **FaaS / serverless** (AWS Lambda), **DBaaS**, **STaaS** (storage).

EXAM POINT Shared responsibility: in IaaS you secure the OS and apps; in SaaS the provider secures almost everything except your data and user access.

3. Deployment Models

Model	Who uses it	Pros	Cons
Public	Anyone, over the internet	Cheap, scalable	Less control
Private	One organisation	Control, security	Costly
Hybrid	Public + private together	Flexible, burst to public	Complex
Community	Organisations with shared needs	Shared cost	Limited users

- **Cloud bursting**: an app runs in private cloud and moves extra load to public cloud at peak time.
- **Multi-cloud**: using services from more than one public provider.

4. Virtualisation

- **Virtualisation** = creating virtual versions of hardware (CPU, memory, storage, network) so many OSes run on one machine.
- **Hypervisor / VMM** = software that creates and manages virtual machines (VMs).

➤ Types of hypervisor

Type	Runs on	Examples	Note
Type 1 (bare metal)	Directly on hardware	VMware ESXi, Xen, Hyper-V, KVM	Faster, used in data centres
Type 2 (hosted)	On a host OS	VirtualBox, VMware Workstation	Easy for desktops

➤ Types of virtualisation

- **Full virtualisation**: guest OS runs unchanged; hardware fully emulated / trapped.
- **Para-virtualisation**: guest OS is modified to talk to the hypervisor (hypercalls); faster.
- **Hardware-assisted**: CPU support (Intel VT-x, AMD-V).
- Also: storage virtualisation, network virtualisation, desktop virtualisation, application virtualisation.

➤ Containers vs VMs

Point	Virtual Machine	Container
Includes	Full guest OS	App + libraries, shares host kernel
Size	GBs	MBs
Start time	Minutes	Seconds
Isolation	Strong	Lighter
Tools	VMware, KVM	Docker, Kubernetes (orchestration)

- **Live migration**: moving a running VM from one host to another with little or no downtime.

> **TIP** Docker builds and runs containers. Kubernetes schedules, scales and heals many containers across machines.

5. Cloud Architecture and Scaling

- **Front end**: client, browser, apps. **Back end**: servers, storage, hypervisors, management, security.
- **Scaling**:
  - **Vertical (scale up)**: bigger machine (more CPU / RAM).
  - **Horizontal (scale out)**: more machines.
- **Elasticity** = automatic scaling with demand (auto scaling groups).
- **Load balancer** spreads requests across servers. Algorithms: round robin, least connections, IP hash.

➤ Availability

FORMULA Availability = MTBF / (MTBF + MTTR). MTBF = mean time between failures, MTTR = mean time to repair.

EXAMPLE 99.9% uptime per year allows 0.1% downtime = 0.001 × 365 × 24 = 8.76 hours of downtime.

- **SLA (Service Level Agreement)**: contract on uptime, support and penalties.

6. Cloud Storage and Data

- **Object storage**: files stored as objects with metadata in buckets (Amazon S3, Azure Blob). Good for backups, media.
- **Block storage**: raw disks attached to VMs (Amazon EBS).
- **File storage**: shared file system (Amazon EFS, NFS).
- Databases: relational (Amazon RDS) and NoSQL (DynamoDB, MongoDB Atlas).
- **CAP theorem**: a distributed store can guarantee only two of Consistency, Availability, Partition tolerance during a network partition.
- **MapReduce / Hadoop**: process big data in parallel. Map emits key-value pairs, Reduce combines them. HDFS stores data in replicated blocks.

➤ MapReduce word count

```
Input:   "cat dog"  |  "dog cat dog"
Map:     (cat,1) (dog,1) | (dog,1) (cat,1) (dog,1)
Shuffle: cat -> [1,1]     dog -> [1,1,1]
Reduce:  (cat,2)          (dog,3)
```

- **Shuffle and sort** groups all values of the same key before Reduce.
- A **combiner** can pre-add counts on the map side to cut network traffic.

7. Cloud Security

- Main threats: data breach, weak identity management, insecure APIs, account hijacking, insider attacks, DDoS, misconfiguration.
- **IAM (Identity and Access Management)**: users, groups, roles, policies. Follow least privilege.
- **Encryption**: at rest (stored data) and in transit (TLS).
- **MFA**: multi-factor authentication for accounts.
- Network security: firewalls, security groups, VPC (virtual private cloud), VPN.
- Hypervisor attacks: **VM escape** (code in a VM breaks out to the host), VM sprawl.
- Compliance and auditing with logs and monitoring.

8. Cloud Services, Economics and Trends

➤ Common service categories

Category	AWS	Azure	Google Cloud
Compute	EC2	Virtual Machines	Compute Engine
Object storage	S3	Blob Storage	Cloud Storage
Serverless	Lambda	Functions	Cloud Functions
Containers	EKS	AKS	GKE

➤ Pricing models

- **On-demand**: pay per hour / second, no commitment.
- **Reserved / committed use**: lower price for a 1 or 3 year commitment.
- **Spot / preemptible**: spare capacity, very cheap, can be taken back.
- **CapEx vs OpEx**: cloud turns capital expense (buying servers) into operating expense (monthly bill).

➤ Recent trends

- **Serverless**: run code without managing servers; pay only when it runs.
- **Edge / fog computing**: process data near where it is produced (lower latency).
- **DevOps and CI/CD** on cloud, **Infrastructure as Code** (Terraform, CloudFormation).
- **Microservices**: app split into small independent services.

9. Most Asked Questions (practice)

1. Define cloud computing. Explain the NIST essential characteristics.
2. Compare IaaS, PaaS and SaaS with examples.
3. Explain public, private, hybrid and community cloud.
4. What is virtualisation? Type 1 vs Type 2 hypervisor.
5. Full virtualisation vs para-virtualisation.
6. Containers vs virtual machines. What is Docker?
7. Vertical vs horizontal scaling. What is elasticity?
8. Explain security issues in cloud computing.
9. Grid vs cloud computing. What is an SLA?
10. Explain MapReduce with a word count example.

> **TIP** Service models, deployment models and virtualisation come in almost every paper. Draw neat diagrams and tables for them.