⛈️ Cloud Computing

1. Cloud Computing Basics

📘 What is Cloud Computing?

- Delivery of computing services (servers, storage, databases, networking, software, etc.) over the internet (“the cloud”).
- Pay-as-you-go model.
- Eliminates the need for owning and maintaining physical servers.

🔑 Key Characteristics:

- **On-demand self-service**
- **Broad network access**
- **Resource pooling**
- **Rapid elasticity**
- **Measured service**

---

2. Cloud Deployment Models

Model	Description	Example
Public	Owned by providers (AWS, Azure, GCP). Shared infrastructure.	Hosting a web app on AWS EC2
Private	Dedicated to one organization. More control & security.	VMware on private datacenter
Hybrid	Combines public + private.	Sensitive data on private; web on public

---

3. Cloud Service Models

Model	Description	Control	Examples	Use Case
IaaS (Infrastructure as a Service)	Virtualized hardware over the internet	Full (OS, runtime, apps)	AWS EC2, Azure VM, GCP Compute Engine	Self-managed applications

PaaS (Platform as a Service)	Platform for app development	Medium (code only)	AWS Elastic Beanstalk, Azure App Service, GCP App Engine	Deploy without managing infrastructure
SaaS (Software as a Service)	Software delivered via web	Minimal	Gmail, Dropbox, Salesforce	Email, storage, CRM

---

4. Key Concepts to Master

🧱 Virtualization & Containers

- **VMs**: Full OS instance; heavy.
- **Containers (e.g., Docker)**: Lightweight, isolated; faster boot.

🛠️ DevOps and CI/CD

- Automate deployments and scaling.
- Tools: GitHub Actions, AWS CodePipeline, Jenkins, Azure DevOps.

📦 Storage Types

- **Object storage**: AWS S3, Azure Blob.
- **Block storage**: AWS EBS, Azure Disks.
- **File storage**: AWS EFS, Azure Files.

🔒 Security

- Identity and Access Management (IAM)
- Encryption (at-rest & in-transit)
- Firewalls (Security Groups, NSGs)

---

5. AWS (Amazon Web Services) Fundamentals

✅ Core Services to Know:

Category	Service	Description	Real-World Use
Compute	EC2	Virtual servers	Run apps, host websites
Containers	ECS, EKS	Managed Docker/Kubernetes	Microservices, orchestration

Serverless	Lambda	Code without servers	Event-driven apps, CRON jobs
Storage	S3	Object storage	Backup, static website hosting
Databases	RDS (SQL), DynamoDB (NoSQL)	Managed databases	App backend, user data
Networking	VPC, Route 53	Virtual network, DNS	Isolated environments
IAM	IAM Roles & Policies	Permissions	Secure access control
Monitoring	CloudWatch, X-Ray	Logs, metrics	Debugging, alerts
DevOps	CodeBuild, CodePipeline	CI/CD	Automate deployment
Analytics	Athena, Redshift	Query & warehousing	BI dashboards

🔁 **Deployment Models in AWS:**

- EC2: Fully managed VM
- ECS (Fargate): Containers without VM management
- Lambda: Just your code, auto-scaled

🛡️ **Security & Identity:**

- Least privilege principle (IAM)
- MFA (Multi-Factor Authentication)
- KMS (Key Management Service) for encryption

---

6. Azure and Google Cloud: Key Highlights

☁️ **Microsoft Azure**

- Strong in hybrid cloud and enterprise integration
- Tightly integrated with Microsoft products (AD, Office 365)
- **Key services**:
  - **App Services**: PaaS for .NET, Java, Node
  - **Azure DevOps**: CI/CD
  - **Azure Functions**: Serverless
  - **Azure Cosmos DB**: Globally distributed NoSQL

☁️ **Google Cloud Platform (GCP)**

- Known for data analytics, ML, container support
- Strong Kubernetes support (original creators)
- **Key services**:
    - **Compute Engine**: VMs
    - **Cloud Run**: Container serverless
    - **BigQuery**: Serverless SQL analytics
    - **Firebase**: Mobile/backend as a service

---

7. Real-World Use Cases

Industry	Cloud Use	Example
E-commerce	Scalable apps, disaster recovery	Shopify uses GCP
Healthcare	Secure data storage, compliance	Philips on AWS
Media	Content delivery, transcoding	Netflix on AWS
Finance	High availability, secure compute	Capital One on AWS
Gaming	Real-time gaming, autoscaling	Fortnite on GCP

🚀 What is CI/CD?

CI/CD stands for **Continuous Integration** and **Continuous Deployment/Delivery**. It automates your software delivery process from code commit to production deployment.

🧩 CI (Continuous Integration)

- Developers push code regularly to a shared repo
- Code is automatically tested and built
- Ensures codebase remains stable and functional

📦 CD (Continuous Deployment / Delivery)

- Automatically deploys changes after successful build/test
- **Delivery** = deploy to staging
- **Deployment** = deploy to production