Author: Madhurima Rawat

Cloud Applications Concepts

Cloud computing has revolutionized the fields of AI and data science, providing scalable infrastructure, cost-effective solutions, and powerful computational capabilities. Leading cloud platforms such as AWS, Google Cloud, and Microsoft Azure offer specialized services that enable AI development, data processing, and machine learning deployment at scale.

Deployment of Cloud-Based Applications

🌐 What is Cloud Deployment?

Cloud deployment refers to hosting software applications on cloud platforms rather than traditional on-premise servers. It enables **scalability**, **accessibility**, and **cost savings** by using infrastructure and platforms managed by providers like AWS, Azure, or GCP.

📚 Core Concepts

- **IaaS (Infrastructure as a Service)**: Provides virtual machines, networking, and storage. You manage OS and software.
  🧠 *Example*: AWS EC2, Azure Virtual Machines.

- **PaaS (Platform as a Service)**: Provides a ready-to-use platform with OS, runtime, and middleware. Focus on writing code.
  🧠 *Example*: Google App Engine, Azure App Service.

- **SaaS (Software as a Service)**: Fully managed software over the internet. No infrastructure or development effort.
  🧠 *Example*: Dropbox, Salesforce, Gmail.

🛠️ Tools and Services

Platform	Use Case
AWS Elastic Beanstalk	Auto-deploy web applications
Azure App Service	API + web app hosting
Google App Engine	Scalable apps with minimal configuration

🌍 Real-World Example

Netflix leverages AWS for content delivery. When millions log in to stream simultaneously, cloud-based auto-scaling ensures smooth playback without buffering or downtime.

Serverless Computing

⚙️ What is Serverless Computing?

Serverless is a cloud-native model where the cloud provider automatically manages infrastructure. Developers only write the code — **no provisioning, scaling, or maintaining servers**.

📚 Core Concepts

- **Function-as-a-Service (FaaS):** Deploy small code snippets (functions) that run in response to events (e.g., HTTP request, database write).
- **Event-Driven:** Each function is invoked by a specific trigger.
- **Stateless Execution:** Functions don’t maintain state between invocations.
- **Pay-as-you-use:** Billed per execution time, not uptime.

🧰 Popular Platforms

Platform	Provider
AWS Lambda	Amazon
Azure Functions	Microsoft
Google Cloud Functions	Google
Cloudflare Workers	Edge serverless compute

🌍 Real-World Example

Airbnb uses AWS Lambda to auto-generate thumbnails whenever a new listing image is uploaded. It’s cost-effective, scales automatically, and needs no backend maintenance.

🎯 Common Use Cases

- Processing uploaded files (image/video compression)
- Event-based workflows (e.g., IoT sensors)
- API backends for web/mobile apps
- Scheduled jobs (daily cleanup tasks)

Containerization with Docker and Kubernetes

🐳 What is Containerization?

Containerization packages an application along with all its dependencies into a **single unit** called a container. This ensures consistent behavior across environments.

📚 Core Concepts

- **Docker**: An open-source platform that allows developers to automate the deployment of applications inside containers.
- **Containers vs VMs**: Containers share the host OS kernel, making them lightweight and faster than traditional virtual machines.
- **Images**: Templates for containers defined using `Dockerfile`.

☸️ Kubernetes (K8s)

Kubernetes is an **orchestration tool** for deploying, managing, and scaling containerized applications.

Key Features:

- **Pods**: The smallest deployable unit.
- **Services**: Abstract a set of pods and provide stable networking.
- **Horizontal Scaling**: Automatically adjusts number of containers.
- **Helm Charts**: Package manager for Kubernetes apps.

🧰 Tools and Platforms

Tool	Purpose
Docker	Create and run containers
Docker Compose	Multi-container applications
Kubernetes	Cluster orchestration
Minikube	Run Kubernetes locally
GKE / AKS / EKS	Managed Kubernetes on GCP/Azure/AWS

🌍 Real-World Examples

- **Spotify** uses Docker to isolate services for music recommendations, user profiles, etc.
- **Pinterest** uses Kubernetes to manage over 1,000 services with millions of requests/day.

Cloud Cost Optimization Strategies

💸 Why Optimize Cloud Costs?

Cloud services are billed on a usage basis. Without proper optimization, companies can overspend or underutilize resources.

📚 Core Strategies

1. Right-Sizing Resources

Analyze usage metrics and adjust compute/storage sizes.

- Replace underutilized large VMs with smaller ones.
- Use cloud advisors (e.g., AWS Compute Optimizer).

2. Auto Scaling

Automatically increase/decrease compute resources based on demand.

- Avoid idle costs.
- Handle traffic spikes.

3. Use Reserved/Spot Instances

- **Reserved Instances**: Up to 72% cheaper for predictable workloads.
- **Spot Instances**: Up to 90% cheaper for short-term, interruptible jobs.

4. Storage Tiering

Use different storage tiers for different access needs:

- AWS S3 Standard for active data
- S3 Glacier for archived backups

5. Monitor & Set Budgets

- Use tools like:
  - **AWS Cost Explorer**
  - **GCP Billing Reports**
  - **Azure Cost Management**
- Set **alerts** and **quotas** for teams.

🌍 Real-World Example

The New York Times moved archived image files to S3 Glacier, reducing annual cloud storage cost by **$1.5 million** without impacting accessibility.

Zynga, a mobile game company, used **spot instances** to run analytics jobs, saving up to **80%** on compute resources.