Cloud Fundamentals for Data Engineering

Target Audience:

- Data Engineers
- Cloud Enthusiasts
- Data Analyst

Objective:

This session provides an understanding of cloud fundamentals from a data engineering perspective, focusing on **ETL (Extract, Transform, Load) operations** for **batch and real-time data processing**. It covers cloud-based services that enable scalable, efficient, and cost-effective data workflows.

---

1. Cloud Computing Basics

Capital vs. Operational Expenditure (CapEx vs. OpEx)

Feature	IaaS	PaaS	SaaS
Infrastructure	Managed by provider	Managed by provider	Managed by provider
OS & Runtime	User-managed	Managed by provider	Managed by provider
Application Deployment	User deploys applications	Managed by provider	Managed by provider
Data Management	Handled by user	Managed by provider	Managed by provider
Security & Compliance	Shared Responsibility	Provider-managed	Provider-managed

Cloud Deployment Models

Model	Definition

On-Premises	Traditional IT setup managed in-house
Infrastructure as a Service (IaaS)	Provides virtualized computing resources, managed OS and apps
Platform as a Service (PaaS)	Provides a platform for development & deployment, managed infrastructure
Software as a Service (SaaS)	Fully managed software solutions delivered via the cloud

Comparison: Cloud vs. On-Premises vs. Hybrid Cloud

Feature	Cloud Computing	On-Premises	Hybrid Cloud
Deployment	Hosted by provider	Private data centers	Combination of both
Infrastructure Ownership	Third-party (AWS, Azure, GCP)	Fully owned by organization	Partially owned
Scalability	High (on-demand)	Limited (manual expansion)	Moderate
Upfront Cost	Low (pay-as-you-go)	High (hardware & setup)	Moderate
Operational Cost	Lower (elastic pricing)	Higher (fixed costs)	Varies (usage-based)
Performance	Dependent on provider	High (customized needs)	Balanced
Disaster Recovery	Multi-region redundancy	Manual backup required	Hybrid solutions available

2. Understanding Cloud Computing with a Pizza Analogy

Model	Explanation
On-Premises	Buy ingredients & cook yourself
IaaS	Rent a kitchen but cook yourself
PaaS	Use a pizza-making service

SaaS	Order a ready-made pizza

3. Cloud Characteristics & Features

1. **On-Demand Self-Service** - Provision resources without manual intervention.
2. **Scalability** - Expand or reduce resources based on demand.
3. **Pay-as-You-Go Pricing** - Pay only for what you use.
4. **High Availability** - Redundant systems ensure uptime.
5. **Resource Pooling** - Shared resources among multiple users.
6. **Security & Compliance** - Built-in encryption and regulatory standards.
7. **Disaster Recovery** - Automated backups and recovery solutions.
8. **Global Access** - Available from anywhere with internet access.
9. **Integration with AI/ML & Big Data** - AI-powered analytics and real-time processing.

4. Cloud Deployment Models

Model	Description
Public Cloud	Shared infrastructure (AWS, Azure, Google Cloud)
Private Cloud	Dedicated cloud infrastructure (VMware Cloud, OpenStack)
Hybrid Cloud	Combination of public & private cloud
Multi-Cloud	Using multiple cloud providers for flexibility

5. Cloud Services for Data Engineering & Application Development

Data Engineering & ETL

- **Batch & Real-time Processing:** AWS Glue, Azure Data Factory, Google Dataflow

Web App Hosting

- **AWS Elastic Beanstalk, Azure App Services, Google App Engine**

Big Data & Analytics

- **AWS Redshift, Google BigQuery, Azure Synapse Analytics**

AI & Machine Learning

- **AWS SageMaker, Google Vertex AI, Azure Machine Learning**

IoT Services

- **AWS IoT Core, Azure IoT Hub, Google IoT Core**

Disaster Recovery & Backup

- **AWS S3, Google Cloud Storage, Azure Blob Storage**

Content Delivery Networks (CDN)

- **AWS CloudFront, Azure CDN, Google Cloud CDN**

---

6. Real-World Case Studies

Case Study 1: Healthcare - Patient Data Management

Factor	Traditional (On-Premises)	Cloud-Based Solution
Scalability	Limited by hardware	Highly scalable (AWS, Azure)
Compliance	Hard to maintain	Cloud providers offer HIPAA compliance
Data Access	Centralized, hard to access remotely	Secure remote access
Costs	High CapEx	Pay-as-you-go OpEx

Case Study 2: Retail - E-commerce Scalability

Factor	Traditional (On-Premises)	Cloud-Based Solution
Traffic Handling	Limited capacity	Auto-scaling (AWS Auto Scaling, Azure VM Scale Sets)

Cost Efficiency	High for peak-load servers	Pay-per-use model
Performance	Slower page loads	CDN + Load Balancing
Disaster Recovery	Manual backups	Automated backups

Case Study 3: Banking - Secure Transactions & Fraud Detection

Factor	Traditional (On-Premises)	Cloud-Based Solution
Fraud Detection	Batch processing	AI-powered real-time analysis (AWS Fraud Detector)
Transaction Speed	Hardware-limited	Cloud-native, real-time transactions
Security	Custom security	Built-in encryption (AWS KMS, Azure Key Vault)
Disaster Recovery	Physical backups	Multi-region redundancy

7. Cloud Evolution Timeline

- **1950s-1970s:** Mainframes & Virtual Machines
- **1980s-1990s:** Distributed Computing & Early Cloud Concepts
- **2000s:** AWS, Azure, Google Cloud launch
- **2010s:** Kubernetes, AI & ML, Hybrid Cloud adoption
- **2020s:** Serverless computing, AI-native cloud services

8. Key Takeaways: Why Cloud Over Traditional Services?

Feature	Traditional (On-Premises)	Cloud-Based Solution
Initial Cost	High CapEx	Low OpEx (pay-as-you-go)
Scalability	Limited by hardware	Auto-scaling, flexible

Security	Requires manual setup	Built-in compliance features
Availability	Requires separate DR setup	Multi-region redundancy