

Introduction to Cloud Computing

Prepared by Dr. Sumaya Alkubaisi

FNT625
Cybersecurity Concepts

Lecture 10



Lecture

10

Learning Goals

By the end of this lecture, you will be able to:

- Define cloud computing and describe its essential characteristics.
- Differentiate between cloud service models (IaaS, PaaS, SaaS) and deployment models (public, private, hybrid, community).
- Explain virtualization and containerization, and how they enable cloud scalability and resource efficiency.
- Understand cloud security principles, including identity management, encryption, access control, and shared responsibility.
- Identify common cloud risks, such as misconfigurations, vendor lock-in, insider threats, and insecure APIs.
- Analyze modern cybersecurity research and understand how new advancements influence cloud technologies.

What Is Cloud Computing?

- Cloud computing means getting computing resources (servers, storage, databases, applications) over the internet instead of owning or maintaining physical devices. You use computing the same way you use electricity — on demand, pay only when needed.

Why Cloud?

- No need to buy or maintain hardware
- Very fast to deploy systems
- Scales globally and easily
- Pay only for what you use
- More room for innovation without technical limitations

Essential Cloud Characteristics (NIST)

- **On-demand self-service:** Users create resources (servers, storage, databases) instantly

Example:

You open AWS → click “Create Server” → in 30 seconds you have a working virtual machine.
No hardware, no waiting.

- **Broad network access:** Available on any device with internet

Example:

You can open Google Drive or Office 365 from:

your phone

your laptop

your friend’s computer

Everything works the same.

Essential Cloud Characteristics (NIST)

- **Resource pooling:** Shared infrastructure; Cloud providers share large data centers among many customers, using virtualization to isolate resources.

Example: Many companies run their apps on the same AWS physical servers, but each one is isolated in its own virtual environment.

- **Rapid elasticity:** Auto-scale up/down; The system can automatically scale up or down depending on demand.

Example:

A shopping website on Black Friday gets 100× more traffic.

The cloud automatically adds more servers.

After the event, the servers auto-shutdown to save cost.

It expands and shrinks like an elastic band.

Essential Cloud Characteristics (NIST)

- **Measured service** (Pay-as-you-go): You only pay for what you actually use.

Example:

- You used a server for 2 hours → you pay for 2 hours.
- You stored 30 GB → you pay only for 30 GB.
- You used 1 million API calls → you pay for 1 million calls.

Lecture
10

Cloud Deployment Models

1. Public Cloud:

A cloud environment owned and operated by a provider (AWS, Azure, Google Cloud). Many customers share the same underlying infrastructure.

2. Private Cloud

A cloud environment dedicated to one organization only. It can be hosted on the company's premises or by a private provider.

Example:

A bank builds its own private cloud inside its data center to store sensitive financial data.

Cloud Deployment Models

3. Hybrid Cloud

A combination of public cloud + private cloud working together.
Data and applications can move between them.

Example:

A university keeps student records in a private cloud, but uses public cloud for hosting its website and e-learning platform.

4. Community Cloud

A cloud shared by several organizations that have similar missions, requirements, or regulations.

Example:

Several hospitals share a community cloud built specifically for healthcare systems

Cloud Service Models

1. IaaS — Infrastructure as a Service

- The cloud provider gives you virtualized hardware resources — servers, storage, and networks.
- You manage the OS, applications, and data.
- You control more, but also have more responsibility (security, OS updates).

Example:

- AWS EC2, Google Compute Engine
- You rent virtual machines to run your apps, backups, or test environments.

Cloud Service Models

2. PaaS — Platform as a Service

- The cloud provider gives you a platform to develop, test, and deploy applications.
- You don't manage servers, OS, or storage — only your application and data.
- Less responsibility **than IaaS, more focus on coding and application logic.**

Example:

- Heroku, AWS Elastic Beanstalk
- Developers can deploy a web app without worrying about the underlying servers.

Cloud Service Models

3. SaaS — Software as a Service

- The cloud provider delivers ready-to-use software through a browser.
- You don't manage infrastructure, platform, or updates.
- Minimal responsibility for the user; everything is handled by the provider.

Example:

- Google Workspace, Zoom, Salesforce
- Users simply log in and use the software online.

Cloud Shared Responsibility Model

The Shared Responsibility Model tells you who is responsible for what in the cloud.

Cloud Provider Responsibilities

The provider (AWS, Azure, Google Cloud) secures the cloud infrastructure itself:

- Physical data centers
- Servers and networking
- Virtualization layers
- Hardware maintenance

Example:

If the physical server breaks, AWS fixes it , not you.

Cloud Shared Responsibility Model

User Responsibilities

You (the customer) secure what you put inside the cloud:

- Your data
- User accounts and passwords
- Access control (IAM permissions)
- Configurations of virtual machines, databases, storage

Example:

If you make your database public by mistake, that's YOUR responsibility.

This is why misconfigurations are the #1 cause of cloud breaches.

Lecture
10

Cloud Architecture Basics

Cloud architecture has several layers working together.

1. Front-End (Client Side)

These are the devices and applications the user interacts with (user interface): Laptops, phones, browsers (Chrome, Safari), Apps (Outlook, Zoom, Salesforce)

Example:

You open your browser and access Google Drive → that's the front-end.

2. Back-End (Cloud Side)

This is where the heavy work happens: Cloud servers, databases, storage (e.g., S3, Azure Blob), application logic

Example:

When you upload a file to Google Drive, it is stored in Google's cloud storage servers.

Lecture
10

Cloud Architecture Basics

3. Cloud Platform Layer

This layer provides the “brains” that control the cloud: APIs, virtualization, identity & Access management, orchestration tools, container platforms

Example:

When AWS automatically creates a new virtual machine for you, that’s the platform layer working.

4. Network Layer

This is the connection between all components: Internet backbone, fiber cables, cloud networking (VPCs, subnets), load balancers

Example:

Your device sends data to the cloud through the internet → routed to the right server → response comes back.

Virtualization in Cloud

- Virtualization means turning one physical server into many virtual servers (VMs).
- Each virtual machine acts like a real computer with its own: CPU, RAM, storage, Operating system
- But all of them run on a single physical machine.

Advantages

1. Efficiency: Instead of one server doing one job, it can run multiple VMs and handle many tasks.
2. Isolation: Each VM is separate; If one crashes, the others keep running.
3. Scalability: You can create or delete virtual machines instantly in the cloud.

Example

You have a powerful server, you split it into 3 virtual machines:

VM 1: runs a website, VM 2: runs a database, VM 3: runs backups

All of them work independently on one physical machine.

Lecture

10

Containers in Cloud

- Containers are lightweight packages that include: Your application, its code, its libraries and dependencies
- They are much smaller and faster than virtual machines.

Advantages:

1. Fast Deployment: Start in milliseconds, not minutes.
2. Very Lightweight: Many containers can run on one server with little overhead.
3. Same Behavior Everywhere: A container runs the same on your laptop, a server, the cloud

Example: You package your web app in a Docker container, you can run this container on: AWS, Azure, Google Cloud, your laptop, or a small server and it will behave exactly the same everywhere.

- ❖ Use virtualization with big apps and full OS needed, use containers with microservices and cloud apps

Lecture
10

Serverless Computing

- You run code without managing servers, the cloud provider handles: Servers, scaling, maintenance, availability.
- You only upload your function (your code), and it runs when needed.

Example:

You want to run a function when a user uploads a photo.

With serverless:

- You upload your code to AWS Lambda
- When the user uploads a photo, Lambda automatically runs your code
- You pay only for the milliseconds the code is running
- No servers, no operating systems, no maintenance

Lecture

10

Serverless Computing

Advantages:

- No server management
- Auto-scaling (if 1 user or 1 million users → cloud handles it)
- Extremely cost-efficient
- Faster development

Real Examples:

- Resize images after upload
- Trigger email notifications
- Process data from IoT devices
- Run backend logic for mobile apps

Lecture

10

Edge Computing

Instead of sending all data to the cloud for processing, edge computing processes data close to where it is generated — “at the edge.”

This reduces: Latency, internet bandwidth, delay in response

Why we needed this technology?

Some applications require very fast response, and waiting for cloud servers (far away) is too slow.

Example: Smart Camera (With normal cloud):

A security camera detects motion → Camera sends video to the cloud → Cloud analyzes it → Cloud sends back: “There is a person”

This takes time!

Edge Computing

Example: Smart Camera (With edge computing):

The camera has a small local processor → It analyzes the video immediately → Only sends important results to the cloud

Much faster, no internet delay.

Other Real-Life Examples:

1. ATMs (Banks): ATMs process transactions locally to speed up approval. They only send final summaries to the bank's cloud.
2. Self-driving cars: Cars process sensor data on-board (edge), because sending it to the cloud would be too slow.
3. Mobile Apps / IoT: Smartwatches, sensors, drones — all use edge computing for speed.

Advantages:

- Faster innovation
- Real-time analytics
- Global customer access
- Lower operational cost
- Strong built-in security tools

Cloud Use Cases in Banking

Mobile banking apps

Fraud detection

Digital payments

Loan scoring

Customer analytics

Why FinTech Uses Cloud?

FinTech companies (startups, mobile apps, payment apps, online banking apps) prefer the cloud for building and growing their products for many reasons:

- Faster innovation (build new app features quickly)
- Real-time analytics (for fraud detection, instant payment checks)
- Global customer access (app works everywhere)
- Lower operational cost (cheaper than running your own servers)
- Built-in security (cloud gives encryption, monitoring, IAM, etc.)

Cloud Benefits for Financial Institutions

- Easier compliance (regulations, auditing, legal requirements)
- Handling peak loads (e.g., salary days)
- Integrating legacy systems with AI/ML
- Controlled migration to cloud
- Faster deployment of banking services

1. Misconfigurations

This means the cloud is set up incorrectly. For example:

- Leaving a storage bucket “public” by mistake
- Allowing too much access to a database
- Weak security settings

Result: Hackers can access customer financial data.

Misconfigurations are the #1 cause of cloud breaches globally.

2. Identity Theft

If attackers steal: User login credentials, API keys, or developer accounts, they can log in as a real user and perform actions like:

- Transferring money
- Accessing payment data
- Making fraudulent transactions

3. Data Breaches

This happens when sensitive financial data is exposed, such as:

- Credit card numbers
- Bank account info
- Transaction history
- Personal identity information (PII)

It can be caused by:

- Hacking
- Weak encryption
- Misconfigured databases
- insider misuse

A data breach can cost millions and damage customer trust.

4. Insider Threats

Risks from people inside the organization, such as:

- Employees
- Contractors
- Administrators

They may:

- Steal financial data
- Abuse privileged access
- Accidentally leak information

5. Vendor Dependency (Vendor Lock-In)

When a FinTech company uses only one cloud provider (AWS, Azure, GCP), it becomes dependent on that provider.

Risks include:

- Hard to move to another cloud
- Higher costs over time
- Limited flexibility
- If the provider has downtime, your app goes down too

FinTech companies need high availability, so lock-in is a real risk.

Cloud Security Toolkit for FinTech

Encryption (at rest & in transit): Protects financial data by encrypting it while stored and while moving across the network.

MFA and Zero Trust: Adds strong authentication and assumes no user or device is trusted by default.

Logging & auditing: Tracks all actions and access to detect suspicious activity and support compliance.

Secure APIs: Ensures financial apps communicate safely using authentication, rate limits, and validation.

Data tokenization & masking: Replaces sensitive data (e.g., card numbers) with safe placeholders to reduce exposure risk.

Modern Cloud Trends

AI-Powered Cloud

- Threat detection: AI analyzes cloud logs and network behavior to automatically detect suspicious or malicious activity in real time.
- Resource optimization: AI dynamically adjusts computing resources to reduce cost and improve performance based on usage patterns.
- Automation: AI automates routine cloud tasks such as scaling, backups, and incident responses, reducing human effort.
- Prediction models: AI forecasts future workloads, failures, or security risks, helping cloud systems prepare and respond proactively.

Multi-Cloud Strategy

Organizations use multiple clouds to avoid vendor lock-in.

Example: using AWS for compute + Azure for analytics.

Modern Cloud Trends

Cloud Automation

Cloud automation uses Infrastructure-as-Code (IaC) tools (like Terraform and CloudFormation) to automatically provision, configure, and deploy cloud resources with consistency and compliance.

Cloud APIs

Cloud APIs allow applications to securely access cloud services for data retrieval, automation, monitoring, and machine-learning functions, and must be protected to prevent unauthorized access.

SASE (Secure Access Service Edge)

SASE is a cloud-based security model that combines networking and security—such as Zero Trust, cloud access control, and SD-WAN—to protect remote users and distributed teams.

Modern Cloud Trends

Cloud-Native Applications

Cloud-native applications are built using microservices, containers, and APIs to enable rapid scaling, modularity, and efficient deployment in cloud environments.

Cloud Storage Types

Cloud storage offers block, file, and object storage, with object storage being the most widely used for scalable cloud-native applications.

Cloud Backup & Disaster Recovery

Cloud backup and disaster recovery provide automated backups, geographic redundancy, and real-time replication to ensure fast recovery from failures.

Generative AI & LLMOps in Cloud

Generative AI in Cloud: Cloud platforms provide scalable infrastructure to build, deploy, and run large language models (LLMs) for generating text, insights, and decisions.

LLMOps (Large Language Model Operations): A lifecycle management approach for LLMs, including:

- Data integration (e.g., financial datasets)

- Model deployment & scaling

- Monitoring, evaluation, and governance

RAG (Retrieval-Augmented Generation):

Combines LLMs with real-time data sources to improve accuracy, explainability, and compliance.

Modern Cloud Trends

Generative AI & LLMOps in Cloud

FinTech Use Cases

Fraud Detection:

LLMs analyze transaction history + contextual signals (user behavior, logs) to detect anomalies

AI Customer Support:

Secure banking chatbots providing personalized, compliant responses

Risk Analysis:

Extract insights from unstructured financial data (reports, contracts, emails)

Latest Cybersecurity Research & Trends

Real-Time Payment Security

Securing financial transactions that are processed instantly (within seconds) with no possibility of reversal.

Unlike traditional payment systems:

- No time for manual verification
- Fraudulent transactions can be executed immediately

Therefore, systems rely on:

- **Real-Time Anomaly Detection:** Detect suspicious behavior during the transaction
- **Behavioral Biometrics:** Verify user identity based on behavior (typing patterns, device usage, interaction style)

Latest Cybersecurity Research & Trends

Real-Time Payment Security (Cloud-Enabled in FinTech)

FinTech Use Cases

- Instant Payment Fraud Detection:
Identify and block fraudulent transactions before completion
- Account Takeover Prevention:
Detect unusual login or transaction behavior beyond passwords
- Secure Mobile Banking:
Protect payment apps using continuous behavioral monitoring

Latest Cybersecurity Research & Trends

Post-Quantum Cryptography Progress (PCQ)

- In 2024, NIST officially approved the new post-quantum algorithms Kyber (for encryption) and Dilithium (for digital signatures), and in 2025 released updated guidance on how to migrate to them (NIST, 2025).
- Studies show that using a hybrid key exchange (classical + PQC together) is the safest way to transition during the next few years.
- FinTech impact: systems that store data for many years must start using post-quantum-safe encryption to stay secure against future quantum attacks.

Latest Cybersecurity Research & Trends

Hashing & Integrity: New Benchmarks

- BLAKE3 is now faster than SHA-256 and SHA-3, especially on multicore processors.
- Researchers are exploring hashes that resist quantum attacks and can run in parallel for better performance.

Use cases: blockchain systems, cloud logging, and data deduplication.

Latest Cybersecurity Research & Trends

Confidential Computing Expansion

- Cloud providers are increasingly using secure enclaves like Intel TDX, AMD SEV-SNP, and Arm CCA to protect data while it's being processed.
- These technologies are applied in financial analytics, federated learning, and other sensitive cloud workloads.

FinTech impact: data can be processed securely without exposing encryption keys, reducing the risk of breaches.

Latest Cybersecurity Research & Trends

AI-Driven Threats & AI Defense

- Attackers are using AI/LLMs to automate phishing, hide malware, and find vulnerabilities (reporting from Microsoft, OpenAI, and other security vendors).
- Security systems now use machine learning to detect unusual activity, model endpoint behavior, and respond automatically.
- Researchers caution that adversarial attacks on ML models can compromise their accuracy and reliability.

Latest Cybersecurity Research & Trends

Zero Trust Security Evolution

- ZT now focuses heavily on continuous identity verification, automation, and short-lived credentials.
- NIST ZTA guidelines (2024) highlight continuous authentication, identity-first controls, and automated security policies.
- Research shows that Zero Trust reduces lateral movement in networks and limits the impact of cloud breaches.

FinTech impact: enforce conditional access and use short-lived credentials to minimize risk.

Latest Cybersecurity Research & Trends

Blockchain Security & Zero-Knowledge Proofs

Zero-Knowledge Proofs (ZKPs): let you prove something is true without revealing the underlying data.

Example: Prove you are over 18 without showing your birthdate.

- zk-SNARKs and zk-STARKs have been improved for smaller proof sizes and faster verification.
- zk-SNARKs: compact, fast proofs with setup requirements.
- zk-STARKs: larger proofs, fully transparent, and quantum-resistant.
- Post-quantum blockchain signatures are being explored to protect long-term ledger security.

Applications: privacy-preserving payments, secure identity verification, and automated compliance in FinTech.

Lecture
10

Hands-On Exercise (Optional) - Hashing a File (How Cloud Storage Verifies Integrity)

Example: Calculate file hash to verify integrity before/after upload (Educational example using Python — mirrors how AWS S3, Azure Blob, and GCP Storage verify file integrity)

```
import hashlib
def compute_sha256(path):
    sha256 = hashlib.sha256()
    with open(path, "rb") as f:
        while chunk := f.read(8192):
            sha256.update(chunk)
    return sha256.hexdigest()

# Example use
file_path = "example.txt"
print("SHA-256:", compute_sha256(file_path))
```

Lecture

10

Recap

- Cloud computing provides on-demand, scalable, virtualized resources, enabling modern financial systems, applications, and analytics.
- Key service models — IaaS, PaaS, SaaS — offer different levels of abstraction and responsibility.
- Core technologies like virtual machines, containers, orchestration, and APIs form the cloud's operational foundation.
- Cloud security depends on shared responsibility, strong identity controls, encryption, monitoring, and secure configuration.
- Major cloud risks include misconfigurations, insecure APIs, insider threats, DDoS, and vendor lock-in.
- The latest cybersecurity research highlights trends such as PQC migration, confidential computing, AI-driven cyber threats, and Zero Trust adoption.
- Cloud computing continues to evolve as a critical backbone for digital transformation, especially in FinTech and data-intensive sectors.

Lecture

10

References

1. Stallings, W. (2023). Cryptography and Network Security: Principles and Practice (8th Edition). Pearson.
2. William Stallings (2021). Network Security Essentials: Applications and Standards (6th Edition). Pearson.
3. NIST Special Publications — various documents on cloud security, Zero Trust, and post-quantum cryptography.
4. Mell, P., & Grance, T. (2020). The NIST Definition of Cloud Computing.
5. Amazon Web Services (2024). Security Best Practices for AWS.
6. Google Cloud Security Team (2024). Principles of Secure Cloud Architecture.
7. Microsoft Azure (2024). Cloud Adoption Framework – Security Guidelines.
8. Suryateja et al. (2024) — A Survey on Lightweight Cryptographic Algorithms in IoT, Communications in Information and Computer Technology (2024).
9. Zhong (2024) — Lightweight block ciphers for resource-constrained devices, Elsevier (2024).
10. NIST et al. (2024) — Post-Quantum Cryptography Standardization: FIPS 203/204/205 & PQC Program Status, NIST (Aug 13, 2024).
11. Chen et al. (2024) — Hybrid Classical–PQC Key Exchange for Large-Scale Systems, Cryptology ePrint / PQC transition analyses (2024).
12. Kerkour (2024) — Choosing a hash function for 2030 and beyond: SHA-2 vs SHA-3 vs BLAKE3 (technical comparison/blog, 2024).

References

13. arXiv benchmark (2024) — Performance Evaluation of Hashing Algorithms (BLAKE3 vs SHA family), arXiv (Jul 2024).
14. Misono et al. (2024) — An Empirical Analysis of AMD SEV-SNP and Intel TDX, Proc. ACM Meas. Anal. Comput. Syst. (Dec 2024).
15. Bertani (2025) — Confidential Computing: A Security Overview and Future Directions, CEUR Workshop Proceedings (2025).
16. More et al. (2025) — Comparative Performance Analysis of Fully Homomorphic Schemes, Scientific Reports / Nature family (2025).
17. Correia (2025) — Hybrid Homomorphic Encryption Approaches (HHE), arXiv / preprint (2025).
18. OpenAI et al. (2025) — Disrupting Malicious Uses of AI: Threat Intelligence Report (Jun 2025).
19. Microsoft / industry reports (2024) — public briefings on AI-assisted cyber activity (news coverage & vendor disclosures, 2024).
20. CERT / Digital Threat Report (2024) — Digital Threat Report 2024 (threat trends including AI/LLM impact).
21. NIST ZTA Working Group (2020 / implementations 2024) — SP 800-207 Zero Trust Architecture and 2024 implementation discussions
22. El-Hajj (2024) — Evaluating zk-SNARK, zk-STARK and related proof systems (MDPI Information journal, 2024)
23. Liu et al. / USENIX (2024) — Scalable Collaborative zk-SNARK and Applications (USENIX/crypto ePrint, 2024).
- Arade (2025) — ZK-STARK Mathematical Foundations & Applications (2025). .

References

24. AWS Generative AI Architecture:

<https://aws.amazon.com/blogs/machine-learning/architect-a-mature-generative-ai-foundation-on-aws/>

25. Cloud Platforms for Generative AI (2024):

https://www.researchgate.net/publication/386577555_Cloud_Platforms_for_Developing_Generative_AI_Solutions_A_Scoping_Review_of_Tools_and_Services

26. Empirical Study of Generative AI Cloud Services (2025):

<https://arxiv.org/abs/2504.08865>

27. European Central Bank – Instant Payments Overview & Risks:

https://www.ecb.europa.eu/paym/integration/retail/instant_payments/html/index.en.html