# Week 9 – Class Discussion Evidence

Topic Focus: Event-Based Processing vs Other Approaches  
This week explored event-based processing and complex event processing (CEP) in cloud systems. We also connected these ideas to practical cloud features such as load-balancing and auto-scaling, which support scalable and resilient applications.

## 1. Cloud Computing

What Cloud Computing services does AWS provide?  
- IaaS – EC2 (compute), S3 (storage), VPC (networking)  
- PaaS – Elastic Beanstalk, RDS, AWS Lambda (serverless)  
- SaaS – Collaboration and productivity tools like WorkSpaces and Chime

How does AWS provide Cloud Computing services?  
AWS uses virtualisation, containerisation, and global data centres. Users interact with services through web consoles, SDKs, and APIs on a pay-as-you-go model.

What are the risks of using AWS?  
- Vendor lock-in (hard to move to other providers)  
- Data security and compliance challenges  
- Service downtime or outages  
- Cost overruns if usage is not monitored properly

## 2. Load-Balancing

What facilities does load-balancing give in Cloud Computing?  
Load-balancing distributes incoming requests across servers to improve availability, fault tolerance, and system responsiveness.

What is a load-balancing policy?  
A load-balancing policy defines how requests are routed. Examples:  
- Round Robin – equal distribution  
- Least Connections – new requests go to servers with the fewest active connections  
- Geo-based routing – users connect to the nearest regional server

## 3. Auto-Scaling

What facilities does auto-scaling give in Cloud Computing?  
Auto-scaling automatically increases or decreases resources based on demand. It ensures elasticity, reliability, and cost efficiency.

What is an auto-scaling policy?  
An auto-scaling policy defines when and how scaling occurs, e.g., “Add 1 EC2 instance if CPU > 80% for 5 minutes; remove an instance if CPU < 30% for 10 minutes.”

## Reflection

This week helped me see how event-driven architectures, load-balancing, and auto-scaling are core to building cloud systems that adapt to real-world workloads. Going forward, I could apply these concepts in:  
- Deploying IoT applications where devices generate continuous events.  
- Designing web services that scale up during peak hours and down during off-hours.  
- Ensuring systems remain resilient and cost-effective.