Architecture of Distributed and Highly Scalable Systems (Duration: 32 Hours) 4 Days

Requirements

- A solid understanding of fundamental networking and system design concepts
- Practical experience with general software development practices
- Familiarity with cloud computing and API design is advantageous

Target Audience

- Software architects and technical leads
- Backend engineers and DevOps professionals
- System designers focused on building scalable cloud applications

Course Content

Introduction to Distributed Systems

- Defining a distributed system
- Common challenges: latency, consistency, and availability
- Overview of system components and communication models

Principles of Scalability

- Vertical versus horizontal scaling
- Load balancing and elasticity
- Scaling storage, compute resources, and I/O

Architectural Patterns

- Client-server and multi-tier architectures
- Service-oriented and microservice architectures
- Event-driven architecture and message queues

CAP Theorem and Consistency Models

- Explanation of the CAP theorem
- Strong versus eventual consistency
- Strategies for choosing between consistency and availability

Data Distribution and Storage Strategies

- Partitioning and sharding techniques
- Replication strategies and quorum reads/writes
- Distributed databases and key-value stores

Communication and Coordination

- REST, gRPC, and message brokers (Kafka, RabbitMQ)
- Leader election and distributed consensus mechanisms
- Using Zookeeper or etcd for coordination

Fault Tolerance and Reliability

- Designing for failure and graceful degradation
- Retry mechanisms, timeouts, and circuit breakers
- Monitoring, observability, and chaos engineering

Cloud-Native Practices

- Containers, orchestration, and Kubernetes
- Statelessness and immutability
- Security best practices

Summary

- Key takeaways and next steps