€TRAINING

Microservice Architecture

Microservice Architecture

Introduction:

This training program provides participants with essential knowledge and skills in Microservice Architecture. It empowers them to understand the principles, design patterns, and best practices for developing and managing microservices.

Program Objectives:

At the end of this program, participants will be able to:

- Understand the fundamentals and benefits of microservice architecture.
- Design and develop microservices using industry-standard practices.
- Implement inter-service communication and data management strategies.
- Monitor, troubleshoot, and optimize microservice-based applications.
- Apply best practices for deploying and managing microservices in production environments.

Targeted Audience:

- Software Architects.
- Developers.
- DevOps Engineers.
- IT Professionals involved in application development and deployment.

Program Outline:

Unit 1:

Introduction to Microservice Architecture:

- Definition and principles of microservice architecture.
- Benefits and challenges of microservices.
- Comparing monolithic and microservice architectures.
- Key components of microservice-based applications.

- Use cases and real-world examples of microservices.

Unit 2:

Designing Microservices:

- Principles of domain-driven design.
- Identifying and defining service boundaries.
- Designing APIs for microservices.
- Patterns for service communication REST, gRPC, messaging.
- Data management strategies database per service, CQRS, event sourcing.

Unit 3:

Developing Microservices:

- Choosing the right technology stack.
- Implementing microservices using popular frameworks Spring Boot, Node.js, etc..
- Handling service discovery and load balancing.
- Implementing security in microservices authentication, authorization.
- Writing tests for microservices unit tests, integration tests.

Unit 4:

Deploying and Managing Microservices:

- Containerization with Docker.
- Orchestration with Kubernetes.
- Continuous Integration and Continuous Deployment CI/CD pipelines.
- Service mesh for microservices management Istio, Linkerd.
- Monitoring and logging Prometheus, ELK stack.

Unit 5:

Optimization and Best Practices:

- Performance tuning for microservices.
- Handling failures and implementing resiliency circuit breakers, retries.
- Scaling microservices horizontally and vertically.
- Ensuring data consistency and managing transactions.
- Future trends and innovations in microservice architecture.