# Unit 4 Flashcards –

## Flashcard 1

* Q: What is microservices architecture?
* A: An approach where an application is structured as a collection of loosely coupled, independently deployable services.

## Flashcard 2

* Q: What are the benefits of microservices?
* A: Scalability, flexibility, fault isolation, ease of deployment, and independent development.

## Flashcard 3

* Q: What is service decomposition?
* A: Breaking a monolithic application into smaller, domain-specific services.

## Flashcard 4

* Q: What is a bounded context?
* A: A design boundary where a service owns its data and logic.

## Flashcard 5

* Q: How do microservices communicate?
* A: Primarily via REST APIs or messaging queues like Kafka.

## Flashcard 6

* Q: What is the role of an API Gateway?
* A: To handle routing, authentication, rate-limiting, and aggregate data from multiple services.

## Flashcard 7

* Q: What is service discovery?
* A: A mechanism for services to dynamically locate each other.

## Flashcard 8

* Q: Why is decentralized data management important?
* A: Each service manages its own database, improving autonomy and scalability.

## Flashcard 9

* Q: What is the purpose of domain-driven design?
* A: To align microservices around real business domains.

## Flashcard 10

* Q: What are common tech stacks for microservices?
* A: Java with Spring Boot, Node.js with Express, Python with Flask.

## Flashcard 11

* Q: Why is unit testing critical in microservices?
* A: It ensures individual components work correctly and supports CI/CD.

## Flashcard 12

* Q: What is mocking in unit testing?
* A: Simulating dependencies so a service can be tested in isolation.

## Flashcard 13

* Q: What is contract testing?
* A: Ensures that the communication between services conforms to agreed-upon interfaces.

## Flashcard 14

* Q: What are integration tests?
* A: Tests that verify interactions between microservices and external systems.

## Flashcard 15

* Q: What tools support unit testing?
* A: JUnit (Java), PyTest (Python), Mocha (Node.js).

## Flashcard 16

* Q: What is the role of Docker in microservices?
* A: Docker packages each service with its dependencies, ensuring consistency across environments.

## Flashcard 17

* Q: What is a Dockerfile?
* A: A file containing instructions to build a Docker image.

## Flashcard 18

* Q: What is the command to build a Docker image?
* A: docker build -t service-name:version .

## Flashcard 19

* Q: How do you run a Docker container?
* A: docker run -p hostPort:containerPort image-name

## Flashcard 20

* Q: What is the advantage of using containers?
* A: Fast startup, portability, and isolated environments.

## Flashcard 21

* Q: Why use Kubernetes with microservices?
* A: For automated scaling, self-healing, and service management.

## Flashcard 22

* Q: What is a Deployment in Kubernetes?
* A: A declarative way to manage Pod creation and updates.

## Flashcard 23

* Q: What is a Service in Kubernetes?
* A: It exposes a set of Pods to network requests.

## Flashcard 24

* Q: What is a ConfigMap in Kubernetes?
* A: Used to inject configuration data into Pods.

## Flashcard 25

* Q: How do you apply a Kubernetes config?
* A: kubectl apply -f deployment.yaml

## Flashcard 26

* Q: Why is CI/CD essential for microservices?
* A: It ensures fast, safe, and consistent deployment across distributed teams.

## Flashcard 27

* Q: What CI/CD tools are used with microservices?
* A: Jenkins, GitHub Actions, GitLab CI.

## Flashcard 28

* Q: What is blue-green deployment?
* A: A strategy that maintains two environments for seamless switch during deployment.

## Flashcard 29

* Q: What is canary release?
* A: Releasing new code to a small group before full rollout.

## Flashcard 30

* Q: How does CI/CD integrate with Docker?
* A: CI pipelines build Docker images and deploy them automatically.

## Flashcard 31

* Q: Do microservices always improve performance?
* A: No, they may add network overhead and complexity.

## Flashcard 32

* Q: Are containers the same as virtual machines?
* A: No, containers share the host OS and are more lightweight.

## Flashcard 33

* Q: Is Kubernetes only for large applications?
* A: No, it benefits projects of all sizes.

## Flashcard 34

* Q: Does containerization remove the need for orchestration?
* A: No, orchestration is still needed to manage services.

## Flashcard 35

* Q: Can microservices only be written in one language?
* A: No, they can be developed in any language that supports APIs.

## Flashcard 36

* Q: Why is logging important in microservices?
* A: To trace requests and identify failures across services.

## Flashcard 37

* Q: What tools help in monitoring microservices?
* A: Prometheus, Grafana, ELK stack.

## Flashcard 38

* Q: What is circuit breaker pattern?
* A: A pattern to prevent a failing service from overloading others.

## Flashcard 39

* Q: What is API rate limiting?
* A: Restricting how many times a user can call an API in a time frame.

## Flashcard 40

* Q: How is security managed in microservices?
* A: Using HTTPS, tokens, OAuth2, and API gateways.