AEROASPIRE INTERN ASSESSMENT-04
KARTHIK TL

1. What’s OpenAPI / Swagger? Why document APIs?

OpenAPI (previously known as Swagger) is a specification for describing RESTful APIs in a structured, machine-readable format like JSON or YAML. Documenting APIs with OpenAPI is essential because it automatically generates interactive, up-to-date documentation, making it easier for developers to understand, test, and integrate with APIs. It also enables automation for generating client SDKs, server stubs, tests, and validation rules, improving development speed and reducing errors. Essentially, it acts as a contract between frontend and backend teams, enhancing collaboration and consistency.

2. How versioning helps backward compatibility; what versioning strategies exist (URI versioning, header versioning etc.).

API versioning helps maintain backward compatibility by allowing developers to introduce new features, fix bugs, or make changes without disrupting existing clients. Different versioning strategies include URI versioning, where the version is part of the URL path (e.g., /v1/users), and header versioning, where the version is specified in HTTP headers. URI versioning is simple but can clutter URLs, while header versioning keeps URLs clean but is harder to implement. Other methods include versioning via request/response bodies or query parameters. Using semantic versioning (major.minor.patch) and clear communication about version changes improves API stability and client experience.

3. If deploying to cloud (or staging), what environment configs will you need? give answers as if i writen this this on my own.

When deploying to cloud or staging environments, environment configurations are necessary to manage differences across environments. Common environment configurations include environment variables for runtime settings (e.g., API keys, database URLs), containerization for consistent deployment, version control integration with CI/CD pipelines, access control policies, monitoring and logging with tools like ELK or CloudWatch, and infrastructure management using Infrastructure as Code tools like Terraform or CloudFormation. Documenting these configurations and deployment processes helps maintain consistency and onboarding efficiency.