# Cloud-Native Development: Questions & Answers

## Question 1. What are enterprise applications, and why are they significant for businesses?

Enterprise applications are software systems designed to meet the complex needs of large organizations. They encompass various functions, from customer relationship management to financial systems. Their significance lies in optimizing processes, integrating data, providing a competitive edge, enabling scalability, and reducing costs for businesses.

## Question 2. What challenges were associated with traditional deployment methods of enterprise applications?

Traditional deployment methods involved high initial costs, maintenance complexities, limited accessibility, and scalability issues. These challenges hindered an organization's ability to adapt to changing business needs and reduced efficiency.

## Question 3. What is cloud-native development, and how does it promise to address the limitations of traditional deployment methods?

Cloud-native development is an approach that leverages cloud computing to build and run applications in a modular and agile manner. It addresses the limitations of traditional methods through cost-efficiency, scalability, accessibility, and embracing DevOps practices. It allows organizations to iterate rapidly, reducing costs and enhancing resource utilization.

## Question 4. What are the core principles of cloud-native development?

Cloud-native development principles include microservices architecture, containerization, orchestration, DevOps practices, and resilience. Microservices break down applications into smaller, independent services, while containerization (e.g., Docker) offers consistency and portability. Orchestration (e.g., Kubernetes) automates management, and DevOps fosters collaboration between development and operations teams. Resilience ensures high availability even in the face of failures.

## Question 5. What are the advantages of cloud-native applications, and what key technologies and tools support them?

Cloud-native applications offer benefits such as flexibility, scalability, cost-efficiency, faster time-to-market, and enhanced reliability. Key technologies and tools include Docker for containerization, Kubernetes for orchestration, CI/CD pipelines for automation, service mesh for microservices communication, serverless computing for managing server infrastructure, and monitoring tools for observability.