**Q-1) Explain Factory and Factory Builder design pattern**

The Factory Design Pattern is a creational pattern that provides an interface for creating objects, allowing subclasses to alter the type of objects created. It involves a product interface, concrete products, a factory interface, and concrete factories. This pattern centralizes object creation, abstracting client code from knowing specific classes. On the other hand, the Factory Builder Design Pattern extends the Factory pattern by separating the construction of a complex object from its representation.

**Q-2) What is Singleton design pattern? What is eagle and lazy initialization?**

The Singleton Design Pattern is a creational design pattern that ensures a class has only one instance and provides a global point of access to that instance. It is commonly used to control access to a shared resource, manage a unique instance of a resource-intensive class, or maintain a single point of control, such as a configuration manager or logging service. Eager initialization is a method of implementing the Singleton pattern where the instance of the class is created at the time the class is loaded or the application starts. Lazy initialization involves creating the Singleton instance only when it is first requested.

**Q-3) What is Docker? Explain the need of Docker in real time application?**

Docker is a platform and tool designed to make it easier to create, deploy, and run applications in lightweight, portable containers. Containers allow a developer to package an application with all of the parts it needs, such as libraries and other dependencies, and ship it all out as one package. Docker is essential in real-time applications for ensuring consistent and portable development environments, enabling efficient resource utilization, facilitating seamless deployment across various environments, and supporting microservices architecture. Its role in DevOps practices and security measures further enhances the reliability and scalability of real-time applications, making Docker a critical tool in modern software development and deployment.

**Q-4) Write the command to build docker image?**

docker build -t image\_name:tag path

**Q-5) What is Docker image and container?**

A Docker image is a lightweight, standalone, and executable package that includes everything needed to run a piece of software, including the code, runtime, libraries, and system tools. It is a snapshot of a file system along with parameters for running processes. A Docker container is a runnable instance of a Docker image. It encapsulates the application and its dependencies, ensuring consistency across different environments, from development to production.

**Q-6) What is Kubernetes? Why Kubernetes is needed?**

Kubernetes is an open-source container orchestration platform designed to automate the deployment, scaling, and management of containerized applications. Kubernetes is essential for automating the deployment and orchestration of containerized applications, ensuring scalability, resilience, and efficient resource utilization.

**Q-7) Explain Jenkins? Write at least 3 lines?**

Jenkins is an open-source automation server used for continuous integration and continuous delivery (CI/CD) of software projects. It facilitates the automation of building, testing, and deploying applications by orchestrating these processes in a repeatable and efficient manner. Jenkins supports integration with a variety of plugins, enabling seamless integration with various tools and version control systems.