1. What is Jenkins and why is it used in DevOps?

Jenkins is an open-source automation server that helps developers and DevOps teams automate the process of building, testing, and deploying applications.

◇ In DevOps, Jenkins plays a critical role by enabling Continuous Integration (CI) and Continuous Deployment (CD).
◇ It monitors your version control system (like GitHub), and every time a developer pushes code, Jenkins automatically runs jobs like testing, building Docker images, and deploying to servers or Kubernetes.

2. How does a Jenkinsfile work?

A Jenkinsfile is a text file that contains the scripted instructions Jenkins follows to run your CI/CD pipeline.

◇ It defines pipeline stages such as Build, Test, Deploy, etc.
◇ It's stored inside your project repository (like on GitHub), so your pipeline is version-controlled with your code.
◇ Jenkins reads the Jenkinsfile and runs each stage sequentially or in parallel, depending on your configuration.

3. What is the purpose of multistage Docker builds?

Multistage Docker builds are used to create smaller, cleaner, and more secure Docker images.

◇ You use one stage (like builder) to compile or build your app (e.g., using node:18).
◇ Then, you use a second stage (like nginx:alpine) to copy only the final output (e.g., React build/ folder).
◇ This removes all unnecessary files and tools (like source code, node_modules, build tools) from the final image.

4. How do GitHub webhooks integrate with Jenkins?

GitHub webhooks allow GitHub to automatically notify Jenkins whenever an event happens (like pushing code to a branch).

◇ When a push or pull request happens, GitHub sends a webhook payload (a small JSON message) to Jenkins.
◇ Jenkins then triggers a job automatically — no need to click "Build Now".

5. What happens in each stage of a typical CI/CD pipeline?

A typical Jenkins CI/CD pipeline includes several key stages:

🛠 1. Checkout / Clone Code
Pulls the latest code from GitHub using git.

📦 2. Install Dependencies
Installs required libraries or modules (npm install, pip install, etc.).

🧪 3. Run Tests
Executes unit, integration, or UI tests to ensure the code works.

🏗 4. Build
Compiles or bundles the application (like npm run build for React).

🐳 5. Build Docker Image
Packages the app into a Docker image, ready for deployment.

🚀 6. Deploy
Deploys the image to production, staging, or Kubernetes, or runs a Docker container.

✅ 7. Post-build Actions
Notifications (Slack, email), clean-up, or archiving artifacts.