**Capstone Project: DevOps Automation for Banking Microservices -Mid term 1 day**

**Project Overview:** ABC Software Company is currently following Agile methodology for a banking client. Most of the Software Development Life Cycle (SDLC) steps are manual. The client is migrating their applications to a microservices architecture and RESTful APIs. To enhance efficiency and streamline deployment, the client wants to automate the project using a Continuous Integration/Continuous Deployment (CI/CD) pipeline with Jenkins.

**Project Requirements**

**Git Requirements**

1. **Repository Creation:**
   * Create a GitHub repository named **BankingMicroservice**.
2. **Branch Management:**
   * As a code reviewer, create individual branches for each developer involved in the project.
   * Developers will push their code to their respective branches upon completing their work.
   * Review the code changes and merge approved changes into the **Master** branch, which contains the latest stable code.
3. **Commit Tracking:**
   * Record all commits in the local repository.
   * Push changes to the remote GitHub repository **BankingMicroservice** once development is complete.
4. **Documentation:**
   * Create a README.md file in the remote repository, serving as documentation for the project.

**Maven Requirements**

1. **Build Automation:**
   * The project is a Maven-based microservice featuring a main file named **App.java**.
   * A unit test program is located in the src/test/ folder.
   * Code should be built automatically as soon as updates are made to the **Master** branch.
   * Required dependencies, such as the JUnit framework, must be added automatically.
   * A JAR file should be generated in the target folder upon successful builds.

**Docker Requirements**

1. **Docker Configuration:**
   * Create a Dockerfile in the **BankingMicroservice** repository.
   * A new Docker image should be automatically created whenever changes are made to the source code, using a new tag for each image.
   * After the Docker image is created, a new Docker container should be started with the latest image.

**Testing Requirements**

1. **Testing Job:**
   * Implement a dummy job that tests the application.
   * If the tests are unstable, an email notification should be sent to the developer.
2. **Email Notifications:**
   * Configure email notifications as part of the CI process to alert developers of test outcomes.

**Production Server Preparation**

1. **Server Environment:**
   * Prepare a Linux production server (Ubuntu or RedHat).
2. **Software Installation:**
   * Use Ansible to install Java and Apache on the production server.
   * Start the Apache service.
3. **Docker Container Deployment:**
   * Run the Docker container on the production server.
   * Repeat the Docker requirements step for the production machine to ensure deployment consistency.

**Conclusion**

This project aims to fully automate the development, testing, and deployment processes of the Banking Microservices application, providing a robust CI/CD pipeline using Jenkins, Git, Maven, and Docker. By implementing these practices, ABC Software Company can ensure faster releases, improved code quality, and streamlined collaboration among developers.