Microservice Deployment DevOps Test

User story

We want to have an Importer Microservice to be built and deployed using Jenkins. All services should be Dockerized.

System requirement

- Jenkins (newest version)
- Docker (newest version)
- Maven according to pom file
- GitHub (if you have an existing account use that or create a new free account to test)
- Postman for testing endpoints

Service Requirement and Process

1. **Microservice Application build and push**

   Dockerize the Spring Boot application run Maven built, test and install locally with Java 11. After building is successful push the code it into your GitHub repository.

2. **Jenkins**

   a. Dockerize Jenkins. Write a docker script to pull a new version of Jenkins and install it locally (container) and spin-up the Jenkins server locally on a given port.

   b. Setup Jenkins writing a Jenkins reusable pipline script to pull the microservice code from your GitHub repository and build and compile.
   It should NOT build automatically when code is pushed in rather it should build when user wishes to run build process in Jenkins.
   The reusable pipline script should be imported into Jenkins. We need a reusable script solution to spin-up Jenkin anywhere dynamically.
   Script / pipline steps: Build (mvn), Run Test (mvn), Push to local file space (anywhere locally on your file system) and Notify status into Slack.

      Note: Notifying to Slack is not important and we don't expect you to finish the Slack notification task! We will be happy if you discover and give it some thoughts in order to discuss during the interview.

3. **Spin-up and Run Microservice Application**

   Dockerize the Spring Boot application which was built (by Jenkins) and push into your local file system. Start the application with Java 11 on a given port (9091).

4. **Test Microservice Application**

   Test the microservice Endpoint using Postman
   Request: GET http://localhost:9091/import/load
   Response: success

Please implement this solution and run us through your solution during the interview.
Thank you and good luck!