CI/CD Deployment for Springboot Application

Project Objective:

As a Full Stack Developer, you have to build a CI/CD pipeline to demonstrate continuous deployment and host the application on AWS EC2 instance.

Background of the problem statement:

As the project is in the final stage, management has asked you to automate the integration and deployment of the web application. You are required to set up an environment where the application will be hosted and accessed by users. The source code is supposed to be fetched from a GitHub repository.

Software Requried:

- Eclipse
- GitHub
- Jenkins
- AWS EC2/ Virtual machine

Steps:

- Created an EC2 instance and started Jenkins on it:

An ec2 instance which has configured security group that allows users to access it through http protocol on port 80 and then install Java, Jenkins, maven, git and docker on it and make it accessible on port 8080 (default port of Jenkins server).

- Made a springboot application:

  A springboot application that has following request urls.
  1) /greeting -Which will show a sample greeting page.
  2) / -sample index page
  3) /hello another sample page.
  It has Dockerfile with required dependency JDK-11.

- Upload it on Github:

  Tracked the springboot application with version controlling system i.e. Git and then connected it with remote repository on Source code management system i.e. Github in the main branch.

- Make an host ec2 instance:

  An ec2 instance which has configured security group that allows users to access it through http protocol on port 80 and install docker on it to run docker container.

- Created a Jenkins Pipeline job:
  Created a pipeline job with steps:
  1) SCM pull: pulls the source code from Github of the created springboot app.
  2) Maven package: packages the source code pulled by step 1.
  3) Docker Build : makes an image of the package created by Maven package with dockerfile in that springboot application.
  4) Docker push: push the created image to dockerhub.
  5) SSH login to Host ec2 instance through Jenkins credentials that has private key for the SSH.
  6) Pull the docker image uploaded by Docker push step and run it as docker container and while running map the internal 8081 port (default springboot app server port) to that ec2 instance's port 8081.
  7) Running application can accessed by ec2 public ip : 8081 port.