



This is another Jenkins pipeline script, like the previous one, with some minor differences in the syntax and commands used. Let's break it down step by step:

1. The **node** block specifies that the pipeline should run on a Jenkins agent (slave node).
2. The pipeline consists of several stages that define the steps to be executed.
  - a. The "Git Clone" stage checks out the source code from a GitHub repository using the **git** command. The **credentialsId** parameter specifies the ID of the Jenkins credential containing the necessary credentials for accessing the repository, and the **url** parameter specifies the URL of the repository.
  - b. The "Maven Clean Build" stage sets up Maven by defining the Maven home directory using the **tool** step. The **tool** step ensures that the specified Maven version is available on the agent. The **sh** step is then used to execute Maven commands (**clean package** and **mvn verify**) to clean and build the project.
  - c. The "SonarQube Analysis" stage sets up the SonarQube environment using the **withSonarQubeEnv** block. The **tool** step is used to define the Maven home directory. Inside the SonarQube environment, the **sh** step executes the **mvn sonar:sonar** command to perform the analysis.
  - d. The next stages involve building and working with Docker images, which are like the previous example.
  - e. The "Docker Login to Hub Docker" stage performs a Docker login operation to the Docker Hub repository. The Docker Hub password is retrieved from the Jenkins credential with ID 'DOCKER\_HUB\_PASSWORD' and passed securely to the **docker login** command using the **--password-stdin** option.
  - f. The "Docker Image Push" stage pushes the Docker image 'josh1991/king-httpd:king-httpd' to the Docker Hub repository using the **docker push** command.