**Configure Jenkins CI/CD Pipeline for Smart-Commute-Planner**

1. Installation
   1. Jenkins can be installed on Personal Computer, AWS Cloud (EC2 Instance). It is recommended to install Jenkins in any Cloud Provider.

For this project we will install Jenkins in AWS EC2 Instance.

1. Steps to Install:
   1. Create any EC2 instance based on the choice in AWS (Recommended to enable the Public IP Address).
   2. Once created, allow the Custom TCP Port 8080 from any IPV4 (0.0.0.0/0) which allows Traffic to Jenkins.
   3. Login to the EC2 instance with the Public key provided by the AWS with user **ubuntu.**
   4. Prerequisites:
      1. JDK (Jenkins runs on Java).
      2. Install Java.
         1. Install Java:
            1. sudo apt update
            2. sudo apt install fontconfig openjdk-17-jre
         2. Verification:
            1. java -version
   5. Now, Proceed with Jenkins Installations.
      1. Visit the URL - <https://www.jenkins.io/doc/book/installing/linux/#debianubuntu>
      2. Install the Long Term Support Release. Copy the command provided there and paste in the console.
2. Login to Jenkins:
   1. Enter the URL: <AWS EC2 Public IP>:8080 in any browser of your choice.
   2. Enter the administrative password in the first screen.
      1. You will find the administrative password on the path provided there in the screen.
   3. After successful login, Install the recommended plugins coming in the next window.
   4. Create one admin user for the future use purposes.
3. Docker Slave Configuration:
   1. Overview: We are configuring Docker to run the stages in the pipeline.
   2. Run below command to install docker in the instance.
      1. sudo apt update
      2. sudo apt install docker.io
   3. Grant permissions to the user: **ubuntu & jenkins.**
      1. sudo su –
      2. usermod -aG docker jenkins
      3. usermod -aG docker ubuntu
      4. systemctl restart docker
   4. Verify, if the docker is running.
      1. docker run hello-world
4. Create Project:
   1. Click on New Item to create the project.
   2. Give the name of the project **smart-commute-planner-server/smart-commute-planner-client** and Choose **Pipeline project** from the list available there.
   3. Fill the necessary details like **Repository URL, Branch Name & Jenkinsfile location.**
      1. **Repo URL:** <https://github.com/somjeet2000/smart-commute-planner-server.git>
      2. **Branch-Name:** development
      3. **File location:** deploy/Jenkinsfile
5. Install required plugins.
   1. Overview: We need to install necessary plugins to run the Jenkins pipeline for our project.
   2. Install Docker Pipeline.
      1. Click on Manage Jenkins -> Plugins -> Available Plugins.
      2. Search for Docker Pipeline and Select the plugin. After that click on Install.
      3. After the installation restart the Jenkins.
         1. Browser URL: <AWS EC2 INSTANCE IP>:8080/restart
         2. Click on **Yes.**
6. Add Required Credentials.
   1. Overview: Adding credentials are essential when you’re running the Jenkins Pipeline, it will work as an environment variable.
   2. Add Docker Credentials.
      1. Overview: Docker credentials are required to login into the **DockerHub** and upload the image there, so that it can be pulled to any deployment services.
      2. Click on Manage Jenkins -> Credentials -> Click on **System** -> Global credentials (unrestricted).
      3. Click on button **Add credentials.**
      4. Choose Username with password for the docker credentials.
         1. Scope will be **Global.**
         2. Put the username and password.
         3. Give the ID : **dockerhub-cred** (You can give any ID, but it should be matching with the name provided in the Jenkinsfile).
   3. Add environment variables to run the application.
      1. Overview: Environment variables are required to run the application, since it contains sensitive information like Database URL, JWT Secrets, API Keys, those should not be visible in the Repository.
      2. Click on Manage Jenkins -> Credentials -> Click on **System** -> Global credentials (unrestricted).
      3. Click on button **Add credentials.**
      4. Choose Secret text from the dropdown menu.
         1. Scope will be **Global.**
         2. Copy and paste the complete text from your .env file to the input area.
         3. Give the ID: **ENV\_FILE\_CONTENTS.**