# Task 2 – CI/CD Pipeline with Jenkins

## 📌 Project Overview

This project demonstrates a CI/CD pipeline for a Node.js application using Jenkins.  
The pipeline automates:  
- Pulling code from GitHub  
- Installing dependencies  
- Running tests  
- Building a Docker image

## 🛠️ Tools & Technologies Used

- Jenkins (LTS)  
- Node.js 18  
- Git & GitHub  
- Docker  
- Jenkins Plugins:  
 - Git Plugin  
 - NodeJS Plugin

## ⚙️ Pipeline Workflow

1. Checkout: Pulls the latest code from the GitHub repository.  
2. Install Dependencies: Runs `npm install` to install required Node.js packages.  
3. Run Tests: Executes `npm test` (currently placeholder).  
4. (Optional) Build Docker Image: Builds a Docker image for the application.

## 🚀 How to Run Locally

1. Clone the repository:  
 git clone <repo-link>  
 cd ci-cd-nodejs-app  
2. Install dependencies:  
 npm install  
3. Run the app:  
 npm start  
4. Open in browser:  
 http://localhost:3000

## 🐳 How to Run with Docker

1. Build the image:  
 docker build -t nodejs-demo-app .  
2. Run the container:  
 docker run -p 3000:3000 nodejs-demo-app

## 🖥️ Jenkins Setup

1. Install Jenkins (LTS) and required plugins.  
2. Configure NodeJS in Jenkins Global Tool Configuration (Node.js 18.20.8).  
3. Create a Pipeline job in Jenkins:  
 - Select Pipeline script from SCM  
 - SCM: Git  
 - Repository URL: <https://github.com/Divyam0017/ci-cd-nodejs-app>  
4. Save and build the pipeline.

## 📜 Jenkinsfile

pipeline {  
 agent any  
  
 tools {  
 nodejs "node18"  
 }  
  
 stages {  
 stage('Checkout') {  
 steps {  
 git branch: 'main', url: 'https://github.com/Divyam0017/ci-cd-nodejs-app.git'  
 }  
 }  
 stage('Install Dependencies') {  
 steps {  
 sh 'npm install'  
 }  
 }  
 stage('Run Tests') {  
 steps {  
 sh 'npm test'  
 }  
 }  
 }  
}

## ✅ Result

• Jenkins pipeline runs successfully on every code push.  
• Dependencies are installed, tests run, and the build completes without errors.