

Automating Code Deployment Using Jenkins CI/CD Pipeline

Objective: Set up a CI/CD pipeline using Jenkins to build, test, containerize, and deploy a Node.js web app automatically.

Tools Used:

- 1 GitHub – code repository
- 2 Jenkins – CI/CD automation server
- 3 Node.js – application runtime
- 4 Docker – containerization
- 5 DockerHub – image registry

Step-by-Step Process:

1. Created a simple Node.js web application with Express (server.js, package.json).
2. Wrote a Dockerfile to containerize the app using Node:18 base image.
3. Pushed the project files (server.js, Dockerfile, Jenkinsfile, package.json) to GitHub.
4. Installed and configured Jenkins on the server, including Git and Docker plugins.
5. Added DockerHub credentials in Jenkins under Manage Jenkins → Credentials → Global.
6. Created a Jenkins Pipeline job and linked it to the GitHub repository (main branch).
7. Wrote a Jenkinsfile that defines stages for Checkout → Build → Docker Build → Push → Deploy.
8. Jenkins automatically triggered the pipeline on code push and built the Docker image.
9. Jenkins pushed the image to DockerHub using secure credentials.
10. Jenkins deployed the latest container version automatically on port 3000.

Sample Jenkinsfile:

```
pipeline
{
    agent any
    stages
    {
        stage('Download')
        {
            steps
            {
                git branch: 'main', url: 'https://github.com/KManikanta-DevOps/internship1.git'
            }
        }
        stage('Build Docker Image')
        {
            steps
            {
                sh 'cd /var/lib/jenkins/workspace/NODEJS; docker build -t manikanta0404/nodejs-demo-app .'
            }
        }
    }
}
```

```

stage('Push Docker Image')
{
  steps
  {
    sh 'docker push manikanta0404/nodejs-demo-app'
  }
}
stage('Deploy')
{
  steps
  {
    sh 'docker stop nodejs-demo || true'
    sh 'docker rm nodejs-demo || true'
    sh 'docker run -d -p 3000:3000 --name nodejs-demo-app manikanta0404/nodejs-
demo-app:latest'
  }
}
}
}
}

```

Explanation:

When code is pushed to GitHub, Jenkins automatically pulls the latest changes and runs the pipeline. It installs dependencies, runs tests, builds a Docker image, pushes it to DockerHub, and finally deploys the app container. This achieves full CI/CD automation using Jenkins.