TASK-1: Automate Code Deployment Using Jenkins Pipeline

Objective: Automate the build, test, and deployment of a Node.js web app using Jenkins Pipeline, Docker, and GitHub.

Tools Used:

1. Ubuntu EC2 Instance (t2.micro)
2. Jenkins
3. Docker
4. Node.js
5. GitHub

Step 1: Setup Environment

```
sudo apt update -y sudo apt install -y git docker.io nodejs npm sudo systemctl
enable docker sudo systemctl start docker sudo usermod -aG docker jenkins sudo
systemctl restart jenkins
```

Step 2: Create Node.js Application

```
mkdir nodejs-demo-app && cd nodejs-demo-app // server.js const express =
require("express"); const app = express(); app.get("/", (req, res) => {
res.send("Hello from Node.js CI/CD pipeline!"); }); app.listen(3000, () => {
console.log("Server running on port 3000"); }); // package.json { "name":
"nodejs-demo-app", "version": "1.0.0", "main": "server.js", "scripts": { "start":
"node server.js", "test": "echo 'No tests yet' && exit 0" }, "dependencies": {
"express": "^4.18.2" } }
```

Step 3: Create Dockerfile

```
FROM node:18 WORKDIR /usr/src/app COPY package*.json ./ RUN npm install COPY . .
EXPOSE 3000 CMD ["npm", "start"]
```

```
docker build -t israyal4421/nodejs-demo-app . docker run -d -p 3000:3000
israyal4421/nodejs-demo-app docker ps
```

Step 4: Push Code to GitHub

```
git init git remote add origin https://github.com/Israyallakkineti/Intership.git
git add . git commit -m "Initial commit" git branch -M main git push -u origin
main
```

Step 5: Jenkins Freestyle Job (Test Git Integration)

1. Go to Jenkins → New Item → Freestyle Project → 'sample' 2. Add Git repository: https://github.com/Israyallakkineti/Intership.git 3. Branch: */main 4. Apply → Save → Build Now

Step 6: Jenkins Pipeline Job

```
pipeline { agent any stages { stage('Download') { steps { git branch: 'main',
url: 'https://github.com/Israyallakkineti/Intership.git' } } stage('Build Docker
Image') { steps { sh 'docker build -t israyal4421/nodejs-demo-app .' } }
stage('Push Docker Image') { steps { sh 'docker push israyal4421/nodejs-demo-app'
} } stage('Deploy') { steps { sh ''' docker stop nodejs-demo-app || true docker rm
nodejs-demo-app || true docker run -d -p 3000:3000 --name nodejs-demo-app
israyal4421/nodejs-demo-app:latest ''' } } } }
```

Step 7: Validate Deployment

```
docker ps Access: http://:3000 Expected Output: Hello from Node.js CI/CD
pipeline!
```

■ Outcome:

1 Automated build and deployment pipeline configured
2 Docker image pushed to Docker Hub
3 Node.js app deployed automatically via Jenkins