# **Documentation: Deploying a React Vite App with Docker and Jenkins**

## **1. Project Folder Structure**

Full-stack-demo/

│── task-manager/

│ ├── Dockerfile

│ ├── nginx.conf

│ ├── package.json

│ ├── package-lock.json

│ ├── src/

│ ├── public/

│ ├── dist/ (generated after build)

│ ├── ...

│── Jenkinsfile

│── other-backend-files/

## **2. Prerequisites**

Before starting, ensure you have the following installed:

* **Jenkins** with Docker plugin
* **Docker** installed and running
* **GitHub repository** containing your React Vite project
* **Jenkins credentials** set up for GitHub and DockerHub

## **3. Jenkinsfile (CI/CD Pipeline)**

This Jenkins pipeline automates the process of pulling the code, building the React app, creating a Docker image, pushing it to Docker Hub, and deploying it.

pipeline {

agent any

tools {

jdk 'jdk17'

nodejs 'node20'

}

stages {

stage('Clean Workspace') {

steps {

script {

echo "Cleaning workspace..."

deleteDir()

}

}

}

stage('Git Checkout') {

steps {

script {

git branch: 'main',

credentialsId: 'github\_secret',

url: 'https://github.com/Immanuvel1207/Full-stack-demo.git'

}

}

}

stage('Install Dependencies & Build') {

steps {

script {

sh '''

cd task-manager

if [ -f package.json ]; then

echo "package.json found. Running npm install..."

npm install

npm run build

else

echo "ERROR: package.json is missing in task-manager. Skipping npm install."

exit 1

fi

'''

}

}

}

stage('Docker Build & Push') {

steps {

script {

withDockerRegistry(credentialsId: 'docker', toolName: 'docker') {

def imageName = "immanuvel12/task-manager"

def tag = "latest"

sh """

cd task-manager

docker build -t ${imageName} .

docker tag ${imageName} ${imageName}:${tag}

docker push ${imageName}:${tag}

"""

}

}

}

}

stage('Deploy Docker Container') {

steps {

script {

def containerName = "task\_manager\_container"

def imageName = "immanuvel12/task-manager:latest"

sh """

docker stop ${containerName} || true

docker rm ${containerName} || true

docker run -d --name ${containerName} -p 3002:3000 ${imageName}

"""

}

}

}

}

}

## **4. Dockerfile (Inside task-manager/)**

This Dockerfile creates a production-ready container for the React Vite application using **NGINX**.

# Use Node.js to build the Vite app

FROM node:20 AS builder

WORKDIR /app

COPY package.json package-lock.json ./

RUN npm install

COPY . .

RUN npm run build

# Use NGINX as the production web server

FROM nginx:latest

RUN rm -rf /etc/nginx/conf.d

COPY nginx.conf /etc/nginx/conf.d/default.conf

COPY --from=builder /app/dist /usr/share/nginx/html

EXPOSE 3000

CMD ["nginx", "-g", "daemon off;"]

## **5. NGINX Configuration File (task-manager/nginx.conf)**

server {

listen 3000;

server\_name localhost;

location / {

root /usr/share/nginx/html;

index index.html;

try\_files $uri /index.html;

}

error\_page 404 /index.html;

}

## **6. Deployment Steps**

### **Step 1: Set Up Jenkins Credentials**

* Go to **Jenkins Dashboard > Manage Jenkins > Credentials**
* Add **GitHub Credentials** (ID: github\_secret)
* Add **DockerHub Credentials** (ID: docker)

### **Step 2: Configure Jenkins Job**

* Create a new **Pipeline Job** in Jenkins
* Select "Pipeline script from SCM"

pipeline {

agent any

tools {

jdk 'jdk17'

nodejs 'node20'

}

stages {

stage('Clean Workspace') {

steps {

script {

echo "Cleaning workspace..."

deleteDir() // Deletes everything in the Jenkins workspace before starting

}

}

}

stage('Git Checkout') {

steps {

script {

git branch: 'main',

credentialsId: 'github\_secret',

url: 'https://github.com/Immanuvel1207/Full-stack-demo.git'

}

}

}

stage('Install Dependencies & Build') {

steps {

script {

sh '''

cd task-manager

if [ -f package.json ]; then

echo "package.json found. Running npm install..."

npm install

npm run build

else

echo "ERROR: package.json is missing in task-manager. Skipping npm install."

exit 1

fi

'''

}

}

}

stage('Docker Build & Push') {

steps {

script {

withDockerRegistry(credentialsId: 'docker', toolName: 'docker') {

def imageName = "immanuvel12/devops"

def tag = "todo"

sh """

cd task-manager

docker build -t ${imageName} .

docker tag ${imageName} ${imageName}:${tag}

docker push ${imageName}:${tag}

"""

}

}

}

}

stage('Deploy Docker Container') {

steps {

script {

def containerName = "jusjes\_container"

def imageName = "immanuvel12/devops:todo"

// Stop and remove the existing container if running

sh """

docker stop ${containerName} || true

docker rm ${containerName} || true

"""

// Run the new container on port 3002

sh """

docker run -d --name ${containerName} -p 3002:3000 ${imageName}

"""

}

}

}

}

}

* Set the **Git Repository URL**: https://github.com/Immanuvel1207/Full-stack-demo.git
* Define the **Branch**: main
* Set the **Script Path**: Jenkinsfile

### **Step 3: Run the Pipeline**

* Click **Build Now** in Jenkins
* Monitor the logs to ensure each stage runs successfully

### **Step 4: Verify the Deployment**

* Check running containers: docker ps
* The application should be running at **http://localhost:3002**



