1. Dockerfile

```
#STAGE 1 build Angular

FROM node:18.19.0-alpine AS builder

WORKDIR /app

COPY package.json package-lock.json ./

RUN npm install

COPY . .

RUN npm run build


#STAGE 2 deploy nginx

FROM nginx:1.23.3-alpine

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

2. Docker-compose.yml

```
services:
 web:
  container_name: web-container
  image: web:latest
  build:
   context: .
   dockerfile: Dockerfile
  environment:
   - TZ=Asia/Bangkok
  ports:
   - 80:80
```

3. Jenkinsfile

```groovy
def notifyLINE(status) {

    def token = "uWN2uQ9eoLCaz4Eb9nwLsGPdvwqXjS41e7fgbY9J15k"

    def jobName = env.JOB_NAME +' '+env.BRANCH_NAME

    def buildNo = env.BUILD_NUMBER


    def url = 'https://notify-api.line.me/api/notify'

    def message = "${jobName} Build #${buildNo} ${status} \r\n"

    sh "curl ${url} -H 'Authorization: Bearer ${token}' -F 'message=${message}'"
}


pipeline {
    agent any


    tools {nodejs "node_18_19"}


    stages {
        stage('SonarQube Analysis') {
            environment {
                scannerHome = tool 'sonarqube-tool';
            }
            steps {
                withSonarQubeEnv(credentialsId: 'jenkins-sonarqube-secret',installationName:
'sonarqube-server' ) {
                    sh "${scannerHome}/bin/sonar-scanner"
```

```groovy
        }
      }
    }
    stage("Quality gate") {
      steps {
        waitForQualityGate abortPipeline: true
      }
    }
    stage('Deploy Web') {
      steps {
        sh "docker compose up -d --build"
      }
    }
  }
  post{
  success{
    notifyLINE("succeed")
  }
  failure{
    notifyLINE("failed")
  }
  }
}
```

4. sonar-project.properties

```
sonar.projectKey=jenkins-project

sonar.sources=.

sonar.host.url=http://sonarqube:9000

#exclusions

sonar.exclusions=node_modules/*;**/*.spec.ts
```