Jenkins File

pipeline {

agent any

environment {

DOCKER\_CONTAINER\_ID = ''

}

stages {

stage('Checkout') {

steps {

checkout scm

}

}

stage('Build and Run images') {

steps {

script {

// Build Docker image

sh 'docker build -t kavitha/react .'

// Run Docker container and capture the container ID

DOCKER\_CONTAINER\_ID = sh(script: 'docker run -p 8000:8000 -d kavitha/react', returnStdout: true).trim()

}

}

}

stage('Run Tests') {

steps {

script {

// Install dependencies and run tests inside the Docker container

sh "docker exec ${DOCKER\_CONTAINER\_ID} pip install --no-cache-dir -r requirements.txt"

sh "docker exec ${DOCKER\_CONTAINER\_ID} pytest tests"

}

}

}

}

post {

success {

script {

// Print the contents of the workspace for debugging

sh 'ls -R'

// Find and print the test result files

def testResultFiles = findFiles(glob: 'test-reports/\*\*/\*.xml')

echo "Test result files found: ${testResultFiles.join(', ')}"

echo 'All tests passed!'

echo "Test report path: ${WORKSPACE}/test-reports"

}

junit 'test-reports/\*\*/\*.xml'

}

failure {

script {

// Print the contents of the workspace for debugging

sh 'ls -R'

// Find and print the test result files

def testResultFiles = findFiles(glob: 'test-reports/\*\*/\*.xml')

echo "Test result files found: ${testResultFiles.join(', ')}"

echo 'Tests failed! Build marked as FAILURE.'

error('Tests failed! Build marked as FAILURE.')

echo "Test report path: ${WORKSPACE}/test-reports"

}

junit 'test-reports/\*\*/\*.xml'

}

cleanup {

// Stop and remove the Docker container after the tests

script {

sh "docker stop ${DOCKER\_CONTAINER\_ID}"

sh "docker rm ${DOCKER\_CONTAINER\_ID}"

}

}

}

}

Docker file

# Using official Python runtime as a parent image

FROM python:3.9

# Set environment variables

ENV PYTHONDONTWRITEBYTECODE 1

ENV PYTHONUNBUFFERED 1

# Set the working directory to /app

WORKDIR /app

# Copy the current directory contents into the container at /app

COPY . /app

# Install any needed packages specified in requirements.txt

RUN pip install --upgrade pip && \

pip install -r requirements.txt && \

pip install pytest

# Make port 8000 available to the world outside this container

EXPOSE 8000

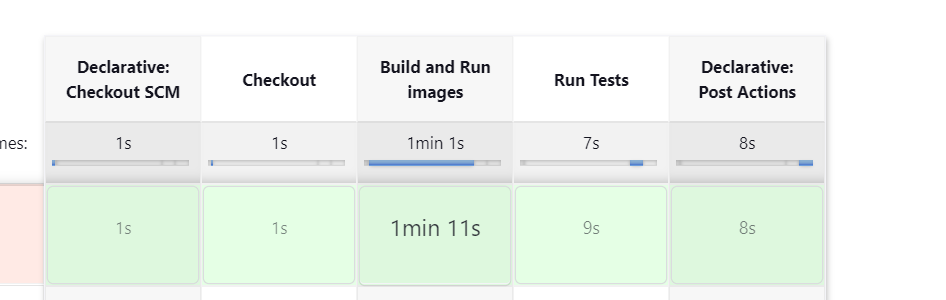
# Run manage.py when the container launches

CMD ["python", "manage.py", "runserver", "0.0.0.0:8000"]

Git hub

<https://github.com/KavithaVenugopal/react-django-app.git>

Jenkins pipeline



Output

