CSDO1010 Assignment 6

Paul Krznaric

# Repo: <https://github.com/PaulKrznaric/06.Lab6-Jenkins-Pipeline-v2>

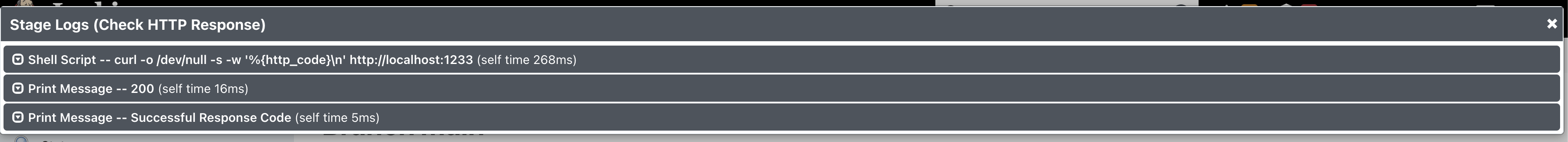
# Proof:

## Pipeline:

Graphical user interface, application

Description automatically generated

## Check HTTP Response



## Docker PS

Graphical user interface, text, application

Description automatically generated

## Browser at Port 1233

Text

Description automatically generated

## Docker Hub Account

Graphical user interface, application

Description automatically generated

## DeployToProduction Stage Logs

A screenshot of a computer

Description automatically generated

## Docker PS

A screenshot of a computer

Description automatically generated with medium confidence

## Browser at Port 1234

Text

Description automatically generated

## Console Output Graphical user interface, text Description automatically generated

Push event to branch main

04:29:06 Connecting to <https://api.github.com> with no credentials, anonymous access

Obtained Jenkinsfile from 9e74aae8eabe35a9c63872d0eb92eb88885799c0

[Pipeline] Start of Pipeline

[Pipeline] node

Running on [Jenkins](http://18.215.161.6:8080/computer/(built-in)/) in /var/lib/jenkins/workspace/sample-react-app\_main

[Pipeline] {

[Pipeline] stage

[Pipeline] { (Declarative: Checkout SCM)

[Pipeline] checkout

The recommended git tool is: NONE

No credentials specified

> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/sample-react-app\_main/.git # timeout=10

Fetching changes from the remote Git repository

> git config remote.origin.url <https://github.com/PaulKrznaric/react-app.git> # timeout=10

Fetching without tags

Fetching upstream changes from <https://github.com/PaulKrznaric/react-app.git>

> git --version # timeout=10

> git --version # 'git version 2.32.0'

> git fetch --no-tags --force --progress -- <https://github.com/PaulKrznaric/react-app.git> +refs/heads/main:refs/remotes/origin/main # timeout=10

Checking out Revision 9e74aae8eabe35a9c63872d0eb92eb88885799c0 (main)

> git config core.sparsecheckout # timeout=10

> git checkout -f 9e74aae8eabe35a9c63872d0eb92eb88885799c0 # timeout=10

Commit message: "Update Docker File name"

> git rev-list --no-walk efe0c870124f007be3303c311a88ad8cf3a35e3a # timeout=10

[Pipeline] }

[Pipeline] // stage

[Pipeline] withEnv

[Pipeline] {

[Pipeline] stage

[Pipeline] { (Build)

[Pipeline] echo

Running build automation

[Pipeline] sh

+ chmod +x gradlew

[Pipeline] sh

+ ./gradlew build --no-daemon

To honour the JVM settings for this build a single-use Daemon process will be forked. See <https://docs.gradle.org/7.0.2/userguide/gradle_daemon.html#sec:disabling_the_daemon>.

Daemon will be stopped at the end of the build

> Task :compileJava NO-SOURCE

> Task :processResources NO-SOURCE

> Task :classes UP-TO-DATE

> Task :war

> Task :assemble

> Task :compileTestJava NO-SOURCE

> Task :processTestResources NO-SOURCE

> Task :testClasses UP-TO-DATE

> Task :test NO-SOURCE

> Task :check UP-TO-DATE

> Task :build

Deprecated Gradle features were used in this build, making it incompatible with Gradle 8.0.

Use '--warning-mode all' to show the individual deprecation warnings.

See <https://docs.gradle.org/7.0.2/userguide/command_line_interface.html#sec:command_line_warnings>

BUILD SUCCESSFUL in 6s

1 actionable task: 1 executed

[Pipeline] archiveArtifacts

Archiving artifacts

[Pipeline] }

[Pipeline] // stage

[Pipeline] stage

[Pipeline] { (Build Docker Image)

[Pipeline] script

[Pipeline] {

[Pipeline] isUnix

[Pipeline] sh

+ docker build -t pkrznaric/react-app .

Sending build context to Docker daemon 1.504MB

Step 1/13 : FROM node:13.12.0-alpine as build

---> 483343d6c5f5

Step 2/13 : WORKDIR /app

---> Using cache

---> 29f4ce8da556

Step 3/13 : ENV PATH /app/node\_modules/.bin:$PATH

---> Using cache

---> c1f872925b33

Step 4/13 : COPY package.json ./

---> Using cache

---> 193858628a45

Step 5/13 : COPY package-lock.json ./

---> Using cache

---> 1b7a6b7eb64a

Step 6/13 : RUN npm ci --silent

---> Using cache

---> ee307ccc7acc

Step 7/13 : RUN npm install react-scripts@3.4.1 --silent

---> Using cache

---> cd00672fe466

Step 8/13 : COPY . ./

---> 2bac3820ead2

Step 9/13 : RUN npm run build

---> Running in 12d230d03e9a

> react-app@0.1.0 build /app

> react-scripts build

Creating an optimized production build...

Compiled successfully.

File sizes after gzip:

41.1 KB build/static/js/2.aca5397c.chunk.js

1.63 KB build/static/js/3.f3014afd.chunk.js

1.17 KB build/static/js/runtime-main.98ff7ae5.js

712 B build/static/js/main.ce5c90ef.chunk.js

556 B build/static/css/main.d1b05096.chunk.css

The project was built assuming it is hosted at /.

You can control this with the homepage field in your package.json.

The build folder is ready to be deployed.

You may serve it with a static server:

npm install -g serve

serve -s build

Find out more about deployment here:

bit.ly/CRA-deploy

Removing intermediate container 12d230d03e9a

---> c37a447646ff

Step 10/13 : FROM nginx:stable-alpine

---> 373f8d4d4c60

Step 11/13 : COPY --from=build /app/build /usr/share/nginx/html

---> Using cache

---> 18a5bac1068a

Step 12/13 : EXPOSE 80

---> Using cache

---> 3ada52745266

Step 13/13 : CMD ["nginx", "-g", "daemon off;"]

---> Using cache

---> 452dd12fdd7e

Successfully built 452dd12fdd7e

Successfully tagged pkrznaric/react-app:latest

[Pipeline] isUnix

[Pipeline] sh

+ docker inspect -f . pkrznaric/react-app

.

[Pipeline] withDockerContainer

Jenkins does not seem to be running inside a container

$ docker run -t -d -u 995:993 -w /var/lib/jenkins/workspace/sample-react-app\_main -v /var/lib/jenkins/workspace/sample-react-app\_main:/var/lib/jenkins/workspace/sample-react-app\_main:rw,z -v /var/lib/jenkins/workspace/sample-react-app\_main@tmp:/var/lib/jenkins/workspace/sample-react-app\_main@tmp:rw,z -e \*\*\*\*\*\*\*\* -e \*\*\*\*\*\*\*\* -e \*\*\*\*\*\*\*\* -e \*\*\*\*\*\*\*\* -e \*\*\*\*\*\*\*\* -e \*\*\*\*\*\*\*\* -e \*\*\*\*\*\*\*\* -e \*\*\*\*\*\*\*\* -e \*\*\*\*\*\*\*\* -e \*\*\*\*\*\*\*\* -e \*\*\*\*\*\*\*\* -e \*\*\*\*\*\*\*\* -e \*\*\*\*\*\*\*\* -e \*\*\*\*\*\*\*\* -e \*\*\*\*\*\*\*\* -e \*\*\*\*\*\*\*\* -e \*\*\*\*\*\*\*\* -e \*\*\*\*\*\*\*\* -e \*\*\*\*\*\*\*\* -e \*\*\*\*\*\*\*\* -e \*\*\*\*\*\*\*\* -e \*\*\*\*\*\*\*\* -e \*\*\*\*\*\*\*\* -e \*\*\*\*\*\*\*\* -e \*\*\*\*\*\*\*\* -e \*\*\*\*\*\*\*\* -e \*\*\*\*\*\*\*\* -e \*\*\*\*\*\*\*\* -e \*\*\*\*\*\*\*\* -e \*\*\*\*\*\*\*\* -e \*\*\*\*\*\*\*\* -e \*\*\*\*\*\*\*\* -e \*\*\*\*\*\*\*\* -e \*\*\*\*\*\*\*\* pkrznaric/react-app cat

$ docker top 646e411b48356fc55acf638493bc841ee43c0c072869653184734faa8462677b -eo pid,comm

[Pipeline] {

[Pipeline] sh

+ curl localhost:1233

% Total % Received % Xferd Average Speed Time Time Time Current

Dload Upload Total Spent Left Speed

0 0 0 0 0 0 0 0 --:--:-- --:--:-- --:--:-- 0

0 0 0 0 0 0 0 0 --:--:-- --:--:-- --:--:-- 0

curl: (7) Failed to connect to localhost port 1233 after 0 ms: Connection refused

+ echo

[Pipeline] }

$ docker stop --time=1 646e411b48356fc55acf638493bc841ee43c0c072869653184734faa8462677b

$ docker rm -f 646e411b48356fc55acf638493bc841ee43c0c072869653184734faa8462677b

[Pipeline] // withDockerContainer

[Pipeline] }

[Pipeline] // script

[Pipeline] }

[Pipeline] // stage

[Pipeline] stage

[Pipeline] { (Push Docker Image)

[Pipeline] script

[Pipeline] {

[Pipeline] withEnv

[Pipeline] {

[Pipeline] withDockerRegistry

$ docker login -u pkrznaric -p \*\*\*\*\*\*\*\* <https://registry.hub.docker.com>

WARNING! Using --password via the CLI is insecure. Use --password-stdin.

WARNING! Your password will be stored unencrypted in /var/lib/jenkins/workspace/sample-react-app\_main@tmp/12113dc3-4656-4b22-8058-9d8f2d08a883/config.json.

Configure a credential helper to remove this warning. See

<https://docs.docker.com/engine/reference/commandline/login/#credentials-store>

Login Succeeded

[Pipeline] {

[Pipeline] isUnix

[Pipeline] sh

+ docker tag pkrznaric/react-app registry.hub.docker.com/pkrznaric/react-app:2

[Pipeline] isUnix

[Pipeline] sh

+ docker push registry.hub.docker.com/pkrznaric/react-app:2

The push refers to repository [registry.hub.docker.com/pkrznaric/react-app]

110d5ac8576a: Preparing

6f44c5b5d074: Preparing

002fcf848e67: Preparing

e419fa208fe1: Preparing

112ee9c2903a: Preparing

68e5252d0d33: Preparing

1a058d5342cc: Preparing

68e5252d0d33: Waiting

1a058d5342cc: Waiting

002fcf848e67: Pushed

112ee9c2903a: Pushed

110d5ac8576a: Pushed

6f44c5b5d074: Pushed

e419fa208fe1: Pushed

1a058d5342cc: Pushed

68e5252d0d33: Pushed

2: digest: sha256:6bc08341a2bcbdf2e723b940761ec19ee09717ea71c687e0d3595fd2f4387944 size: 1778

[Pipeline] isUnix

[Pipeline] sh

+ docker tag pkrznaric/react-app registry.hub.docker.com/pkrznaric/react-app:latest

[Pipeline] isUnix

[Pipeline] sh

+ docker push registry.hub.docker.com/pkrznaric/react-app:latest

The push refers to repository [registry.hub.docker.com/pkrznaric/react-app]

110d5ac8576a: Preparing

6f44c5b5d074: Preparing

002fcf848e67: Preparing

e419fa208fe1: Preparing

112ee9c2903a: Preparing

68e5252d0d33: Preparing

1a058d5342cc: Preparing

68e5252d0d33: Waiting

1a058d5342cc: Waiting

110d5ac8576a: Layer already exists

002fcf848e67: Layer already exists

e419fa208fe1: Layer already exists

6f44c5b5d074: Layer already exists

112ee9c2903a: Layer already exists

68e5252d0d33: Layer already exists

1a058d5342cc: Layer already exists

latest: digest: sha256:6bc08341a2bcbdf2e723b940761ec19ee09717ea71c687e0d3595fd2f4387944 size: 1778

[Pipeline] }

[Pipeline] // withDockerRegistry

[Pipeline] }

[Pipeline] // withEnv

[Pipeline] }

[Pipeline] // script

[Pipeline] }

[Pipeline] // stage

[Pipeline] stage

[Pipeline] { (DeployToStaging)

[Pipeline] script

[Pipeline] {

[Pipeline] sh

+ docker pull pkrznaric/react-app:2

2: Pulling from pkrznaric/react-app

Digest: sha256:6bc08341a2bcbdf2e723b940761ec19ee09717ea71c687e0d3595fd2f4387944

Status: Downloaded newer image for pkrznaric/react-app:2

docker.io/pkrznaric/react-app:2

[Pipeline] sh

+ docker stop react-app

Error response from daemon: No such container: react-app

[Pipeline] sh

+ docker run --restart always --name react-app -p 1233:80 -d pkrznaric/react-app:2

9233a3e4aaf7d8d452a0cba362aefebc4f63e85ccb7c772c9a27f7dbc584f729

[Pipeline] }

[Pipeline] // script

[Pipeline] }

[Pipeline] // stage

[Pipeline] stage

[Pipeline] { (Check HTTP Response)

[Pipeline] script

[Pipeline] {

[Pipeline] sh

+ curl -o /dev/null -s -w '%{http\_code}\n' <http://localhost:1233>

[Pipeline] echo

200

[Pipeline] echo

Successful Response Code

[Pipeline] }

[Pipeline] // script

[Pipeline] }

[Pipeline] // stage

[Pipeline] stage

[Pipeline] { (DeployToProduction)

[Pipeline] input

Does the staging environment look OK? Did You get 200 response?

[Proceed](http://18.215.161.6:8080/job/sample-react-app/job/main/2/console) or [Abort](http://18.215.161.6:8080/job/sample-react-app/job/main/2/console)

Approved by [Paul Krznaric](http://18.215.161.6:8080/user/pkrznaric)

[Pipeline] milestone

Trying to pass milestone 1

[Pipeline] script

[Pipeline] {

[Pipeline] sh

+ docker pull pkrznaric/react-app:2

2: Pulling from pkrznaric/react-app

Digest: sha256:6bc08341a2bcbdf2e723b940761ec19ee09717ea71c687e0d3595fd2f4387944

Status: Image is up to date for pkrznaric/react-app:2

docker.io/pkrznaric/react-app:2

[Pipeline] sh

+ docker stop react-app

react-app

[Pipeline] sh

+ docker rm react-app

react-app

[Pipeline] sh

+ docker run --restart always --name react-app -p 1233:80 -d pkrznaric/react-app:2

f76205dc43db4f4e0b716fbd2a470194e9d52b57658646f1c3fe963e07e78edb

[Pipeline] }

[Pipeline] // script

[Pipeline] }

[Pipeline] // stage

[Pipeline] }

[Pipeline] // withEnv

[Pipeline] }

[Pipeline] // node

[Pipeline] End of Pipeline

Finished: SUCCESS