REACT APP DEPLOYMENT WITH DOCKER

1.Project Overview

- I implemented a complete Continuous Integration and Continuous Deployment (CI/CD) pipeline for a React application. The pipeline automates the process of building, containerizing, and deploying the application to a cloud server whenever code changes are pushed to the repository.

2. Tools & Technologies Used

- React JS – Frontend application development
- Docker – Containerization of the application
- DockerHub – Docker image storage and management
- Jenkins – CI/CD pipeline automation
- GitHub – Source code version control
- AWS EC2 – Cloud deployment server
- Nginx – Web server to serve the application

3. Architecture Workflow

1. I push the application code to the GitHub repository.
2. A webhook automatically triggers the Jenkins pipeline.
3. Jenkins pulls the latest source code.
4. Jenkins builds a Docker image for the application.
5. The Docker image is pushed to DockerHub.
6. The EC2 server pulls the latest Docker image.
7. A Docker container is deployed and exposed on port 80.
8. The React application becomes accessible through the browser.

4. Jenkins Pipeline Stages

9. Checkout source code from GitHub
10. Build Docker image
11. Authenticate and login to DockerHub
12. Push Docker image to DockerHub
13. Stop existing container (if running)
14. Pull latest Docker image

15. Run new container

5. Docker Commands Used

```
docker build -t react-cicd-app .
docker tag react-cicd-app <dockerhub-username>/react-cicd-app:latest
docker push <dockerhub-username>/react-cicd-app:latest
docker run -d -p 80:80 --name react-app <dockerhub-username>/react-cicd-app:latest
```

6. Deployment Details

I deployed the application on an AWS EC2 Ubuntu instance.

Port 80 enabled in the security group for HTTP access

Jenkins configured on port 8080 for automation

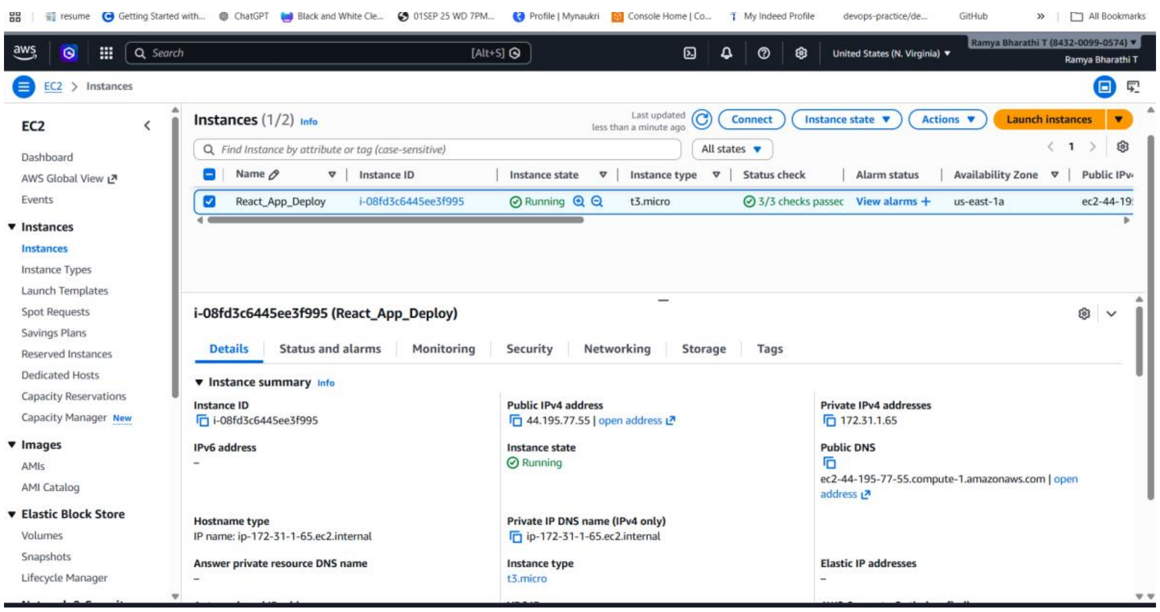
7. Outcome

Successfully implemented an automated CI/CD pipeline. Every code push triggers automatic build, Docker image creation, DockerHub push, and deployment to EC2 without manual intervention.

8. Conclusion

This project demonstrates my practical knowledge of DevOps practices including automation, containerization, and cloud deployment. It reflects real-world CI/CD implementation used in modern software development environments.

.



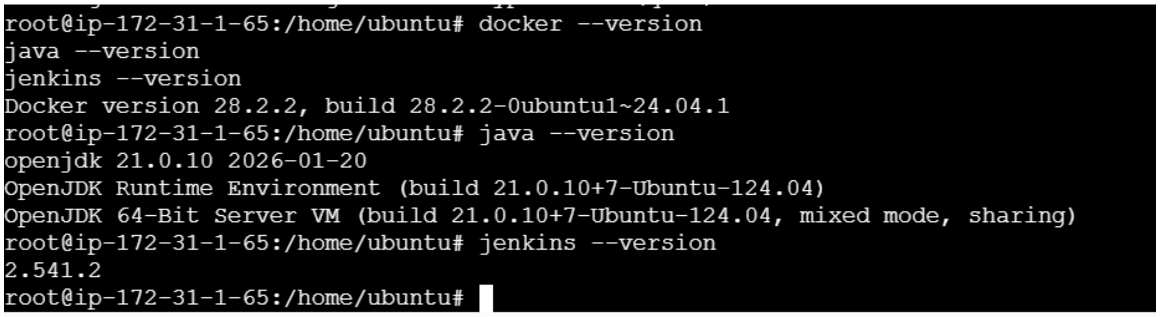
Instances (1/2)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv
React_App_Deploy	i-08fd3c6445ee3f995	Running	t3.micro	3/3 checks passed	View alarms +	us-east-1a	ec2-44-19

i-08fd3c6445ee3f995 (React_App_Deploy)

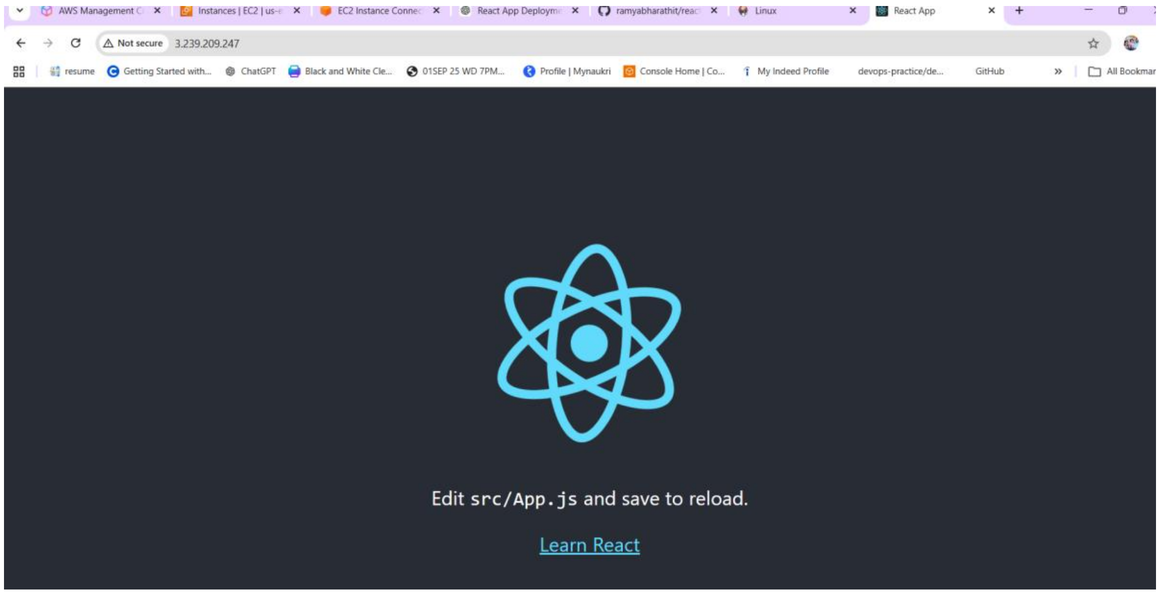
Instance summary

Instance ID: i-08fd3c6445ee3f995	Public IPv4 address: 44.195.77.55 | open address	Private IPv4 addresses: 172.31.1.65
IPv6 address: –	Instance state: Running	Public DNS: ec2-44-195-77-55.compute-1.amazonaws.com | open address
Hostname type: IP name: ip-172-31-1-65.ec2.internal	Private IP DNS name (IPv4 only): ip-172-31-1-65.ec2.internal	
Answer private resource DNS name: –	Instance type: t3.micro	Elastic IP addresses: –



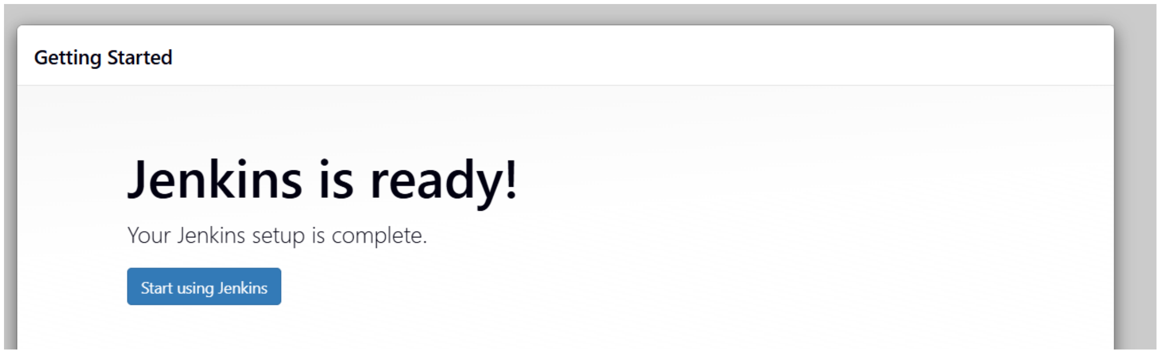
```
root@ip-172-31-1-65:/home/ubuntu# docker --version
java --version
jenkins --version
Docker version 28.2.2, build 28.2.2-0ubuntu1~24.04.1
root@ip-172-31-1-65:/home/ubuntu# java --version
openjdk 21.0.10 2026-01-20
OpenJDK Runtime Environment (build 21.0.10+7-Ubuntu-124.04)
OpenJDK 64-Bit Server VM (build 21.0.10+7-Ubuntu-124.04, mixed mode, sharing)
root@ip-172-31-1-65:/home/ubuntu# jenkins --version
2.541.2
root@ip-172-31-1-65:/home/ubuntu#
```

```
root@ip-172-31-1-65:/home/ubuntu# docker --version
java --version
jenkins --version
Docker version 28.2.2, build 28.2.2-0ubuntu1~24.04.1
root@ip-172-31-1-65:/home/ubuntu# java --version
openjdk 21.0.10 2026-01-20
OpenJDK Runtime Environment (build 21.0.10+7-Ubuntu-124.04)
OpenJDK 64-Bit Server VM (build 21.0.10+7-Ubuntu-124.04, mixed mode, sharing)
root@ip-172-31-1-65:/home/ubuntu# jenkins --version
2.541.2
root@ip-172-31-1-65:/home/ubuntu#
```



Edit `src/App.js` and save to reload.

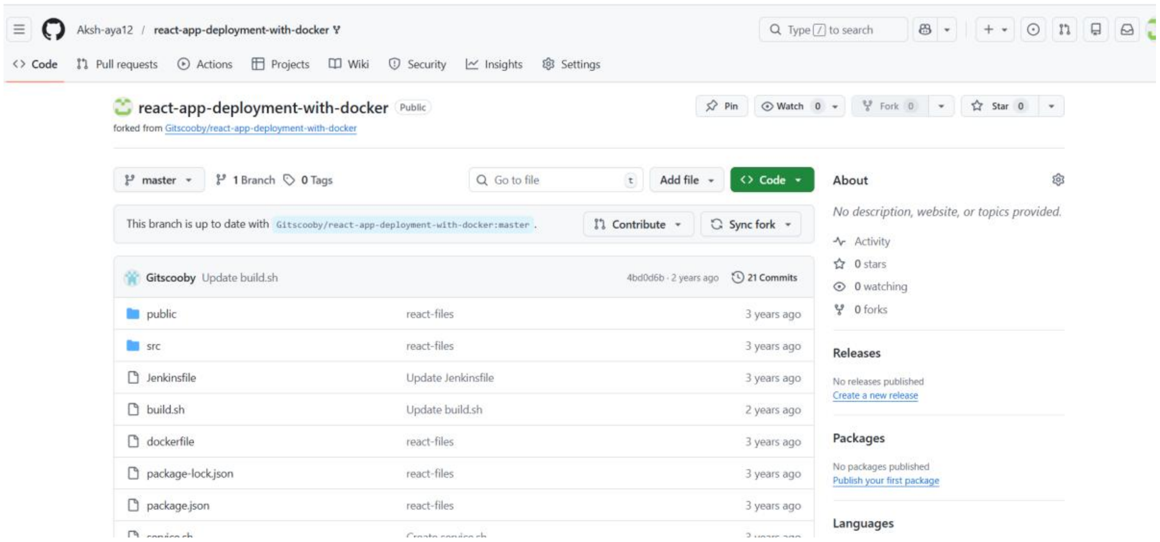
Learn React



Getting Started

Jenkins is ready!

Your Jenkins setup is complete.



react-app-deployment-with-docker Public

forked from Gitscooby/react-app-deployment-with-docker

This branch is up to date with `Gitscooby/react-app-deployment-with-docker:master` .

Gitscooby Update build.sh 4bd0d6b · 2 years ago 21 Commits

public	react-files	3 years ago
src	react-files	3 years ago
Jenkinsfile	Update Jenkinsfile	3 years ago
build.sh	Update build.sh	2 years ago
dockerfile	react-files	3 years ago
package-lock.json	react-files	3 years ago
package.json	react-files	3 years ago

About

No description, website, or topics provided.

Activity
0 stars
0 watching
0 forks

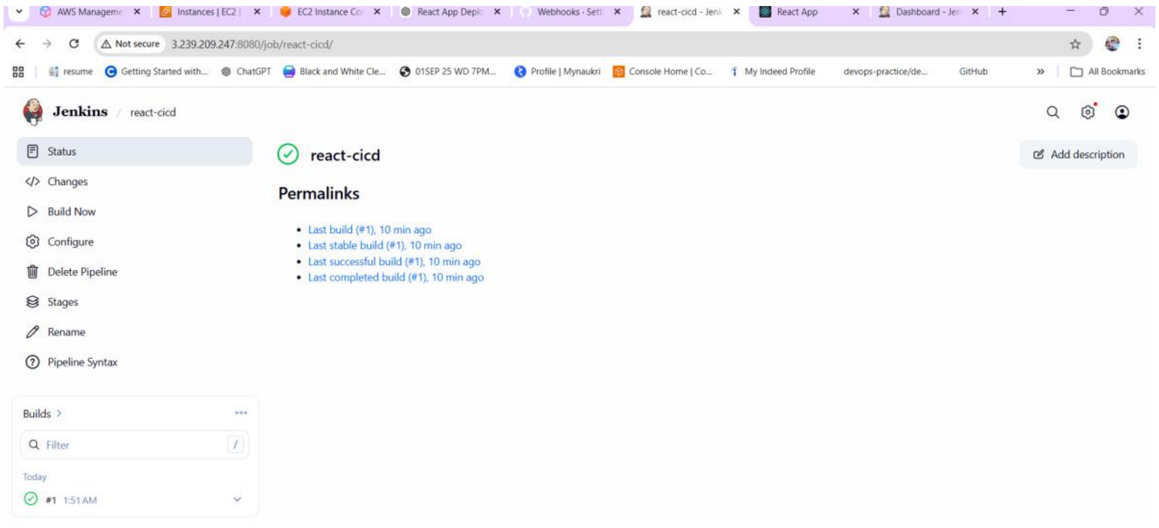
Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

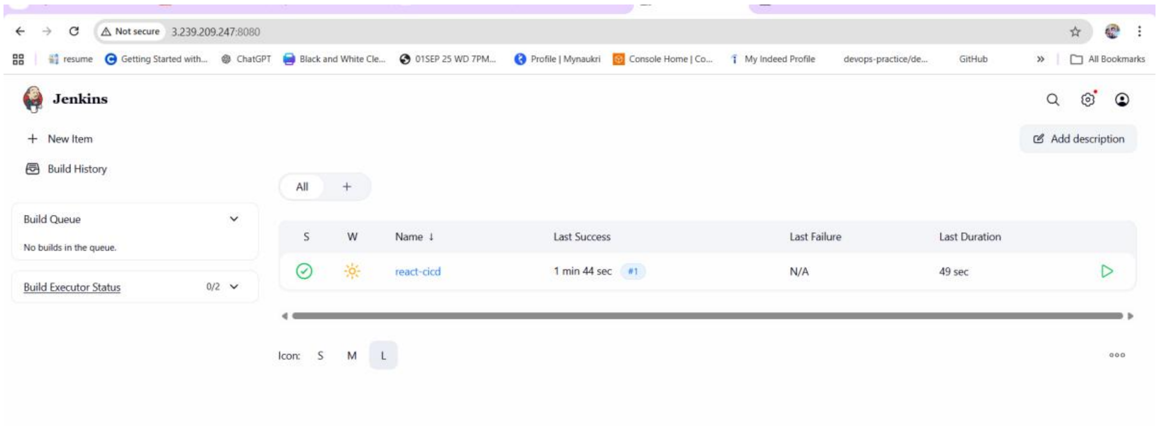
Languages



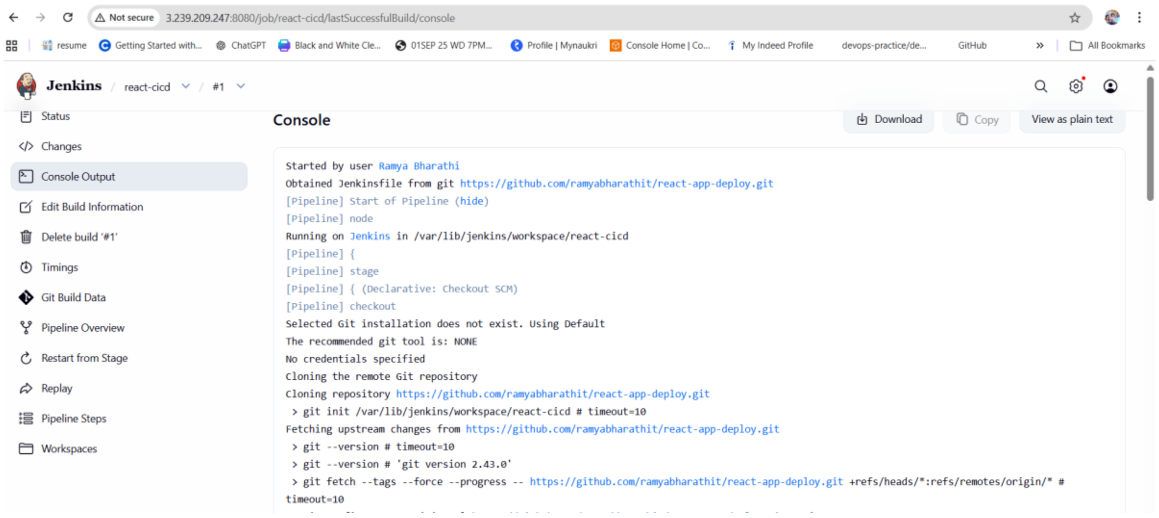
react-cicd

Permalinks

- Last build (#1), 10 min ago
- Last stable build (#1), 10 min ago
- Last successful build (#1), 10 min ago
- Last completed build (#1), 10 min ago

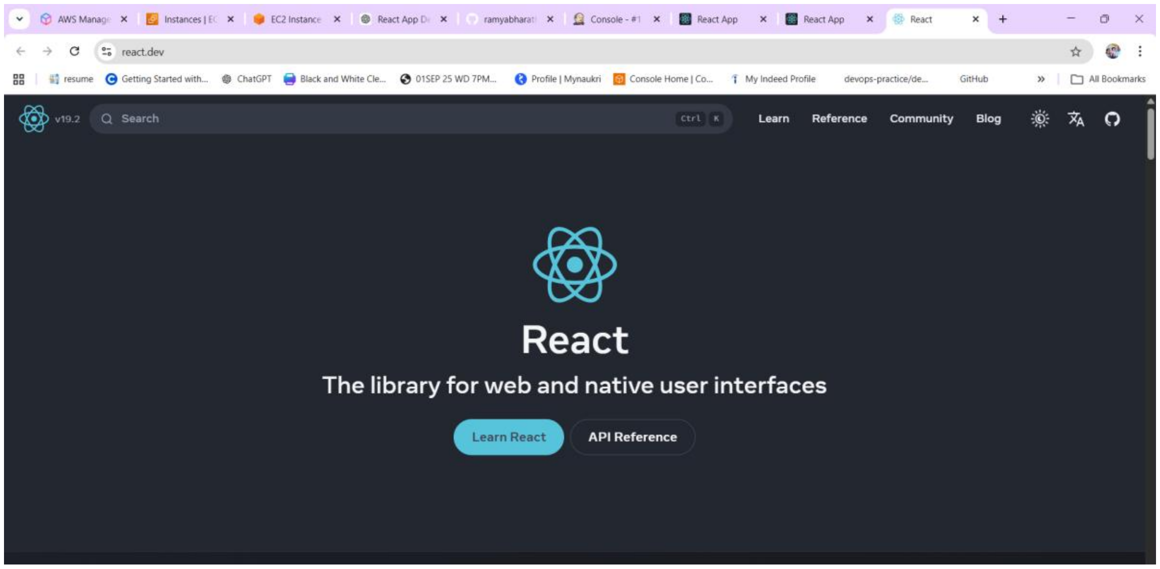


S	W	Name ↓	Last Success	Last Failure	Last Duration
✓	☀	react-cicd	1 min 44 sec #1	N/A	49 sec



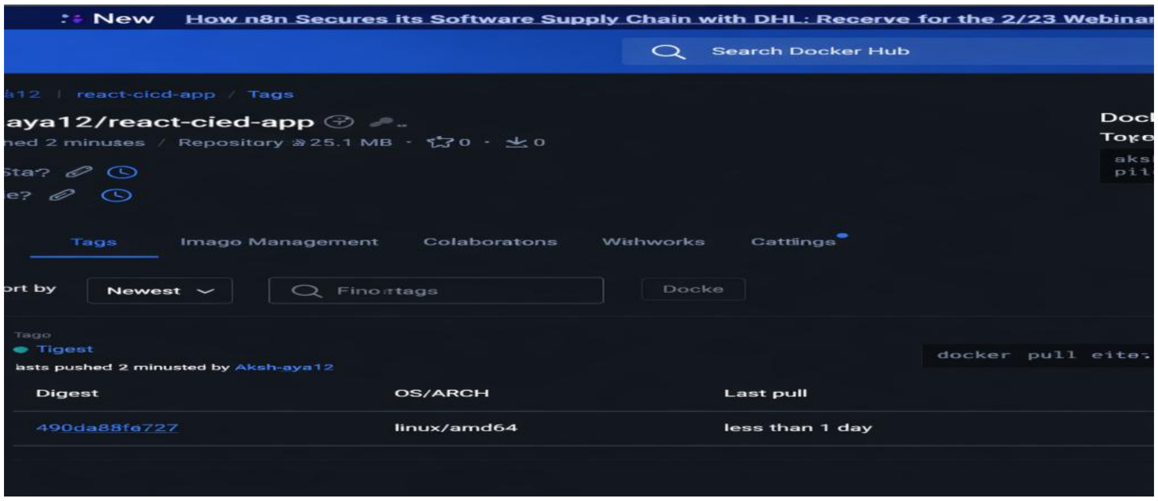
Console

```
Started by user Ramya Bharathi
Obtained Jenkinsfile from git https://github.com/ramyabharathit/react-app-deploy.git
[Pipeline] Start of Pipeline (hide)
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/react-cicd
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
[Pipeline] checkout
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/ramyabharathit/react-app-deploy.git
 > git init /var/lib/jenkins/workspace/react-cicd # timeout=10
Fetching upstream changes from https://github.com/ramyabharathit/react-app-deploy.git
 > git --version # timeout=10
 > git --version # 'git version 2.43.0'
 > git fetch --tags --force --progress -- https://github.com/ramyabharathit/react-app-deploy.git +refs/heads/*:refs/remotes/origin/* #
timeout=10
[Pipeline] sh
+ docker rm react-container
react-container
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Run New Container)
[Pipeline] sh
+ docker run -d -p 80:80 --name react-container react-app
2f495b10e6a64fda435a7d6f20f88da08d95c2fc5a58804753f7ac63cbc9f5ae
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

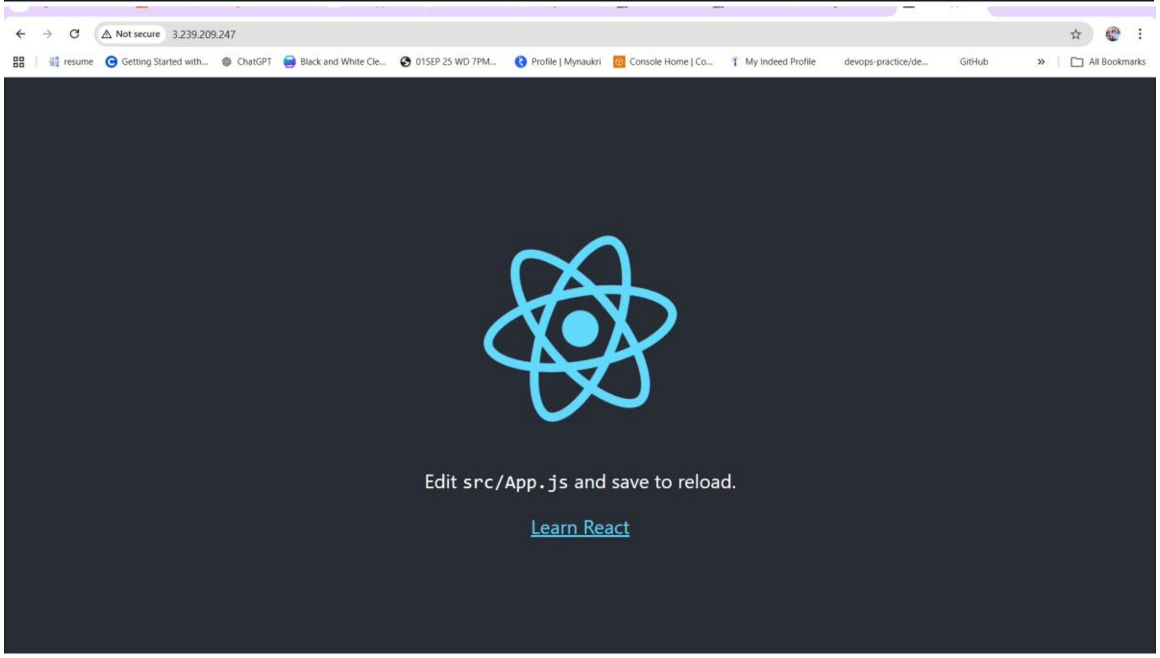


React

The library for web and native user interfaces


New
How n8n Secures its Software Supply Chain with DHL: Recerve for the 2/23 Webinar
Search Docker Hub
a12 / react-cicd-app / Tags
aya12/react-cied-app
ned 2 minuses / Repository 25.1 MB · 0 · 0
Tags
Imago Management
Colaboratons
Wishworks
Cattlings
ort by
Newest
Fino rtags
Docke
Tago
Tigest
asts pushed 2 minusted by Aksh-aya12
docker pull eite:

Digest	OS/ARCH	Last pull
490da88fe727	linux/amd64	less than 1 day



Edit `src/App.js` and save to reload.

Learn React