# ICCD Automation in DevOps

## - Run Book Documentation

Link to source code for this project: https://github.com/Iyappan97/react-app-test.git
https://github.com/Iyappan97/Terraform_Eks_cluster.git

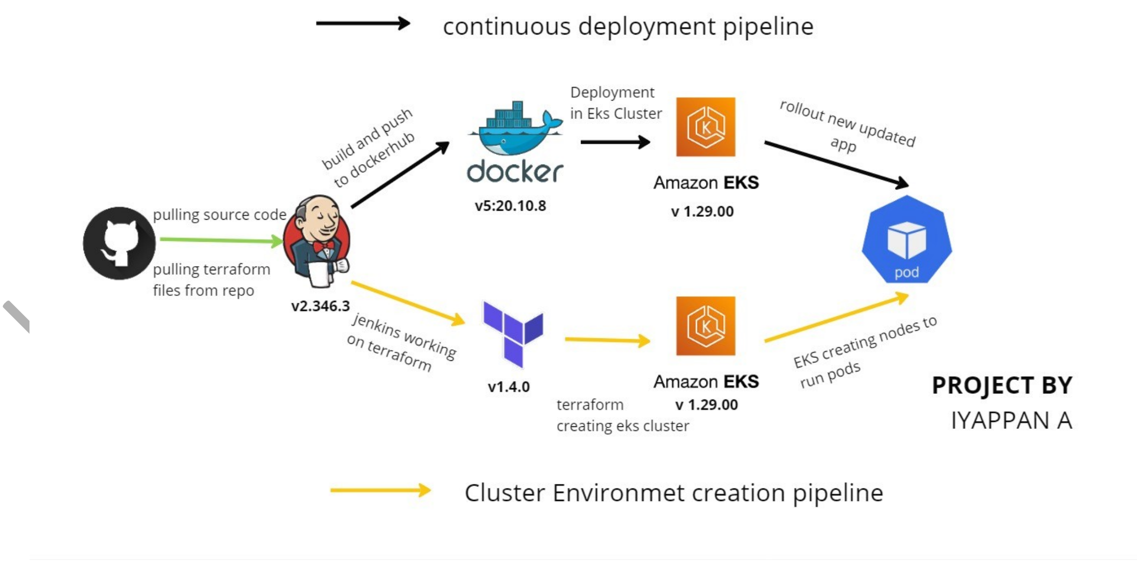
Enumeration of given task :

Create a react.js app , build it as deployable container image, deploy it in cloud instances. Create a cicd flow any commit in github should trigger the build and deploy process.

Solution:

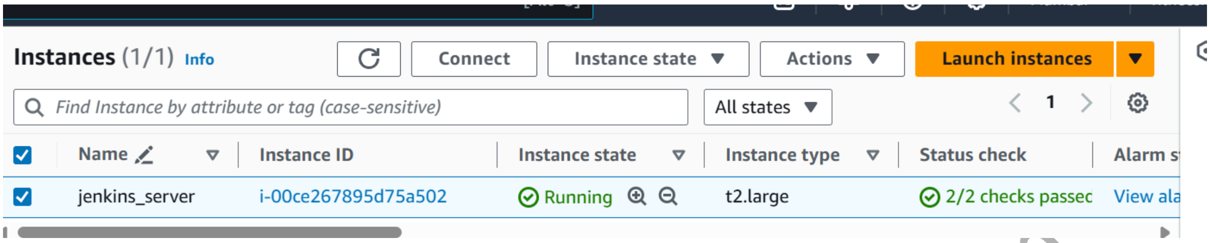
1. Selecting of appropriate tool
    a. Github - repository
    b. Docker – containerisation tool
    c. Terraform – Infrasctructure As A Code tool
    d. AWS EKS – kubernetes cluster formation services from AWS
    e. Jenkins – Automation tool

2. Creating mind map for workflow


continuous deployment pipeline
pulling source code
pulling terraform files from repo
v2.346.3
build and push to dockerhub
docker
v5:20.10.8
Deployment in Eks Cluster
Amazon EKS
v 1.29.00
rollout new updated app
pod
jenkins working on terraform
v1.4.0
terraform creating eks cluster
Amazon EKS
v 1.29.00
EKS creating nodes to run pods
PROJECT BY
IYAPPAN A
Cluster Environmet creation pipeline

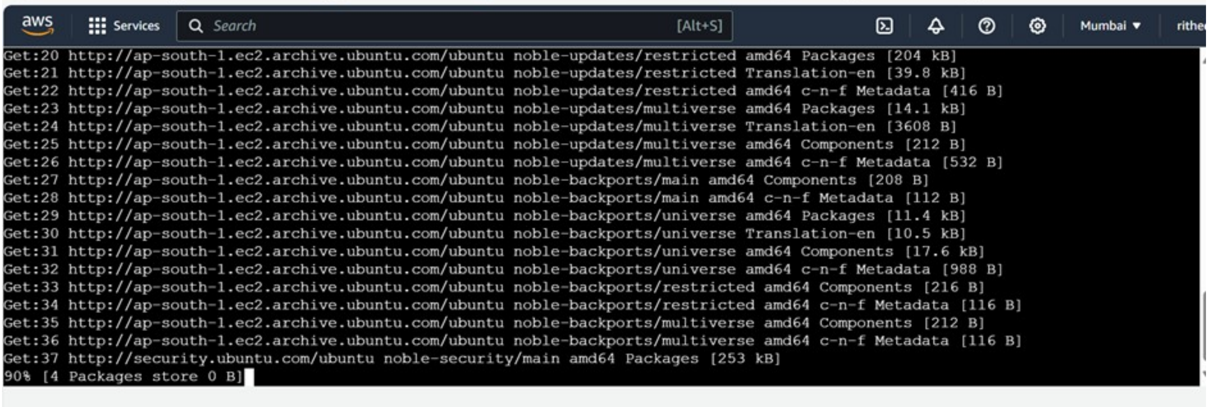
Run book:

1. Create a instance for running Jenkins



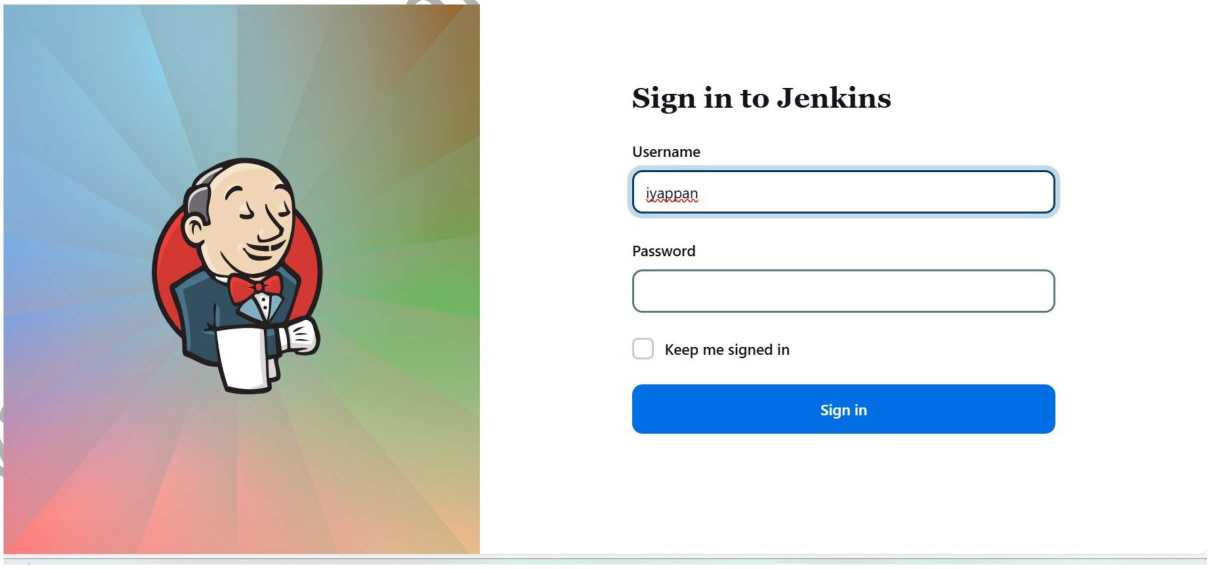
| ☑ | Name | Instance ID | Instance state | Instance type | Status check | Alarm s |
|---|---|---|---|---|---|---|
| ☑ | jenkins_server | i-00ce267895d75a502 | ✓ Running | t2.large | ✓ 2/2 checks passec | View ala |

2. Install tools and its dependencies



```
Get:20 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [204 kB]
Get:21 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted Translation-en [39.8 kB]
Get:22 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 c-n-f Metadata [416 B]
Get:23 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Packages [14.1 kB]
Get:24 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse Translation-en [3608 B]
Get:25 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [212 B]
Get:26 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 c-n-f Metadata [532 B]
Get:27 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [208 B]
Get:28 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 c-n-f Metadata [112 B]
Get:29 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [11.4 kB]
Get:30 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe Translation-en [10.5 kB]
Get:31 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [17.6 kB]
Get:32 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 c-n-f Metadata [988 B]
Get:33 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [216 B]
Get:34 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 c-n-f Metadata [116 B]
Get:35 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [212 B]
Get:36 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 c-n-f Metadata [116 B]
Get:37 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [253 kB]
90% [4 Packages store 0 B]
```

3. Configure Jenkins to use those tools , give appropriate privileges to Jenkins.
4. Login into Jenkins console



Sign in to Jenkins

Username: iyappan

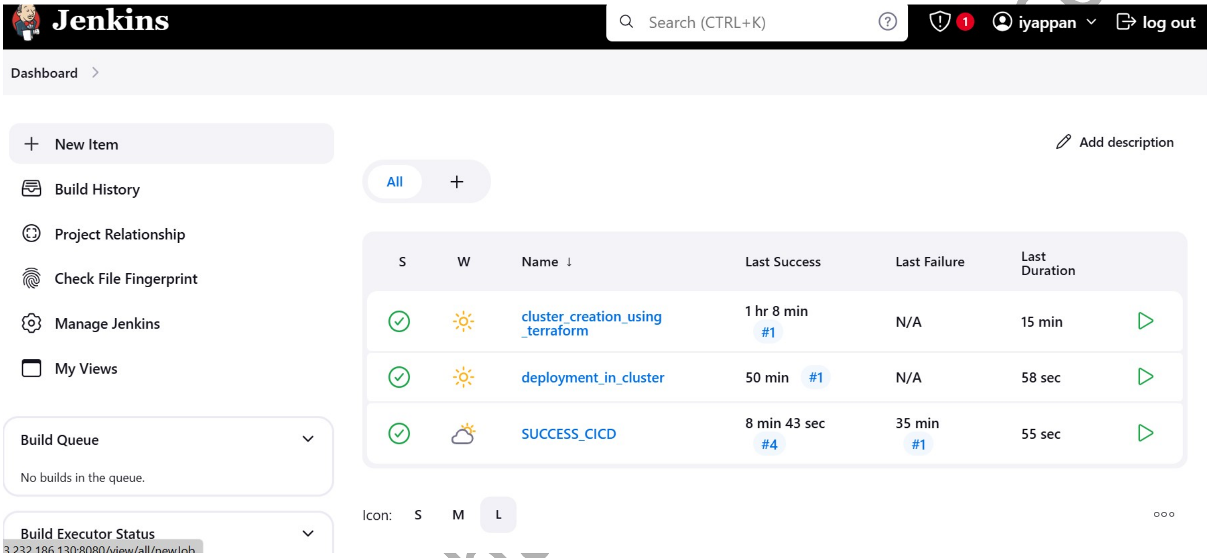
Password:

□ Keep me signed in

   a. Configure Jenkins with credentials ,tools and plugins

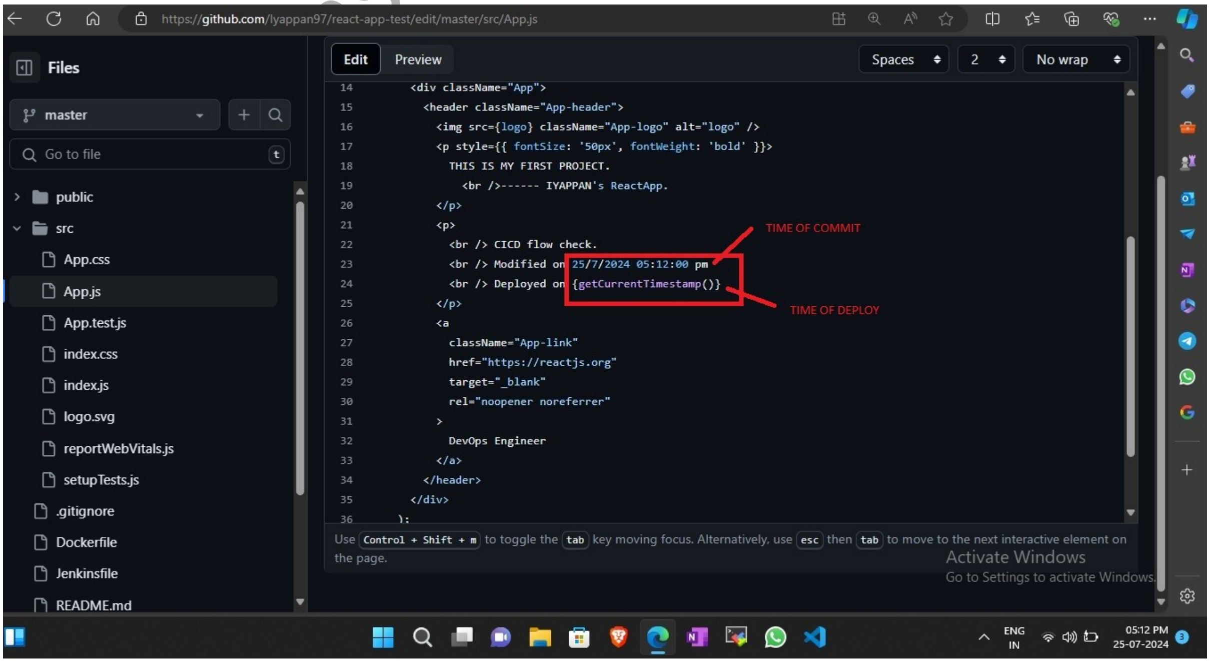
5. Create a react js App using npm packages and upload it to Github
6. Create terraform file to establish an eks cluster and push it to Github

7. Create pipeline
   a. Pipeline – cluster_creation_using_terraform
   b. Pipeline – app deployment in cluster
      # first manual deployment is done to check whether the service is running properly or not
   c. Pipeline – SUCCESS_CICD pipeline is final automated pipeline
      # this pipeline is responsible for checking Github for any new commit and triggers build process automatically



| S | W | Name ↓ | Last Success | Last Failure | Last Duration |
|---|---|---|---|---|---|
| ✓ | ☀ | cluster_creation_using_terraform | 1 hr 8 min #1 | N/A | 15 min |
| ✓ | ☀ | deployment_in_cluster | 50 min #1 | N/A | 58 sec |
| ✓ | ⛅ | SUCCESS_CICD | 8 min 43 sec #4 | 35 min #1 | 55 sec |

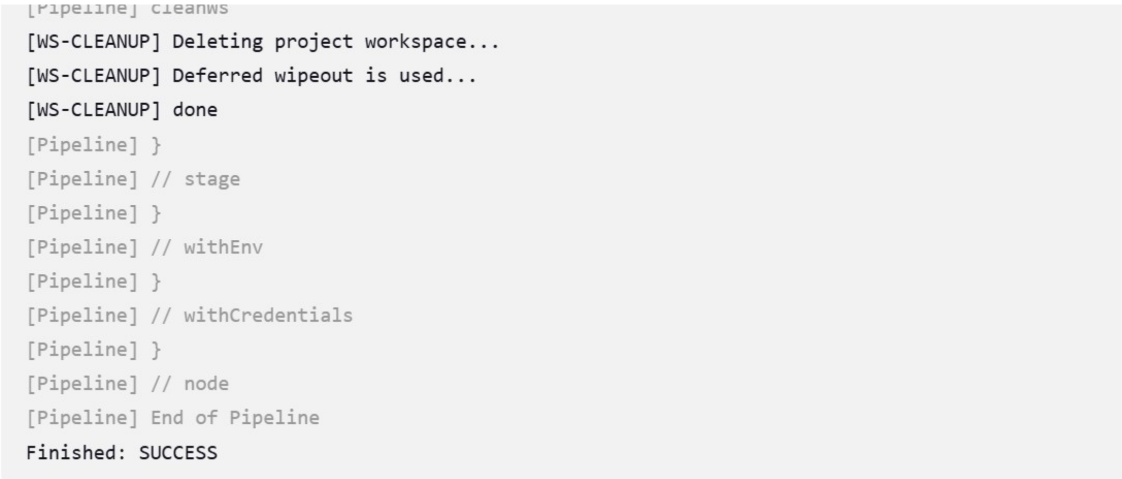
8. Make any new commit in Github repository
   # While committing I add time manually  of commit for reference



```
    <div className="App">
      <header className="App-header">
        <img src={logo} className="App-logo" alt="logo" />
        <p style={{ fontSize: '50px', fontWeight: 'bold' }}>
          THIS IS MY FIRST PROJECT.
            <br />------ IYAPPAN's ReactApp.
        </p>
        <p>
          <br /> CICD flow check.
          <br /> Modified on 25/7/2024 05:12:00 pm
          <br /> Deployed on {getCurrentTimestamp()}
        </p>
        <a
          className="App-link"
          href="https://reactjs.org"
          target="_blank"
          rel="noopener noreferrer"
        >
          DevOps Engineer
        </a>
      </header>
    </div>
  );
```

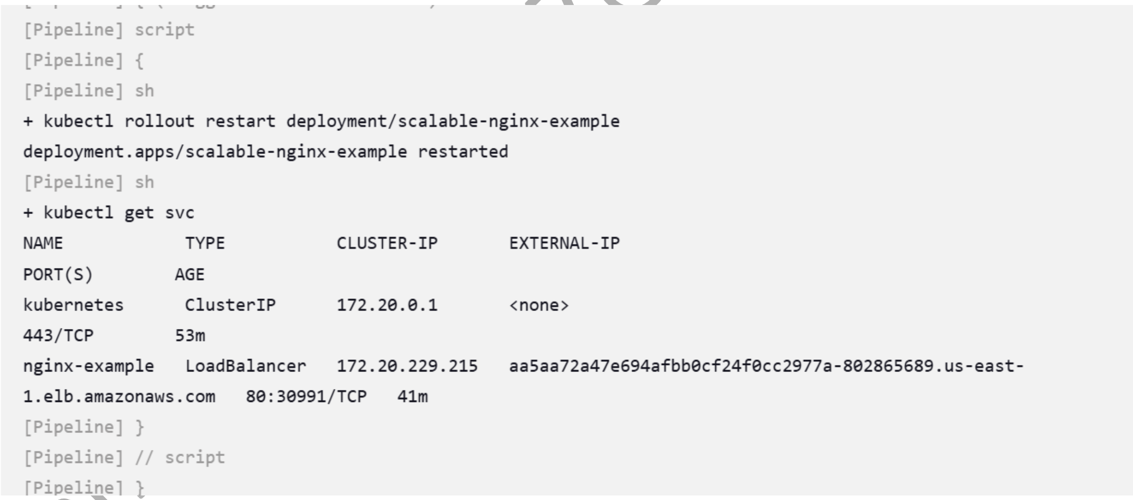
Use Control + Shift + m to toggle the tab key moving focus. Alternatively, use esc then tab to move to the next interactive element on the page.

9. Testing
    a. This new commit will automatically trigger pipeline to build (SUCCESS_CICD)



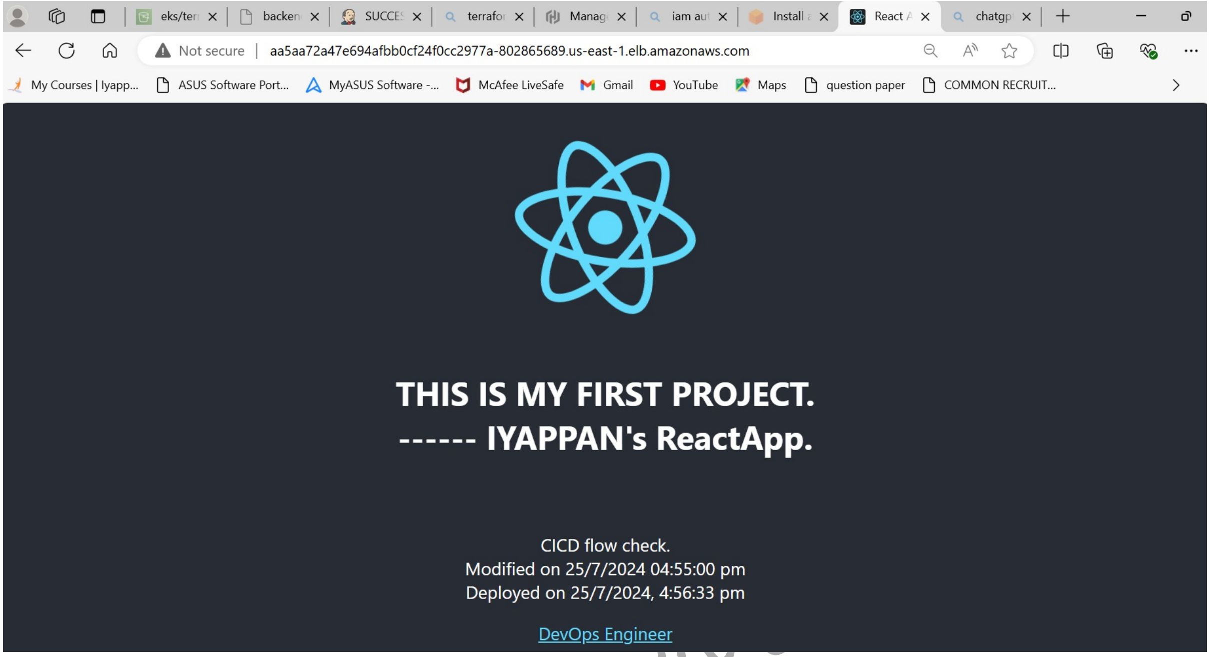
```
[Pipeline] cleanws
[WS-CLEANUP] Deleting project workspace...
[WS-CLEANUP] Deferred wipeout is used...
[WS-CLEANUP] done
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // withCredentials
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

    b. Getting Loadbalancer IP from logs



```
[Pipeline] script
[Pipeline] {
[Pipeline] sh
+ kubectl rollout restart deployment/scalable-nginx-example
deployment.apps/scalable-nginx-example restarted
[Pipeline] sh
+ kubectl get svc
NAME            TYPE           CLUSTER-IP       EXTERNAL-IP
PORT(S)        AGE
kubernetes      ClusterIP      172.20.0.1       <none>
443/TCP        53m
nginx-example   LoadBalancer   172.20.229.215   aa5aa72a47e694afbb0cf24f0cc2977a-802865689.us-east-
1.elb.amazonaws.com   80:30991/TCP   41m
[Pipeline] }
[Pipeline] // script
[Pipeline] }
```

10.Pinging the url from browser



Not secure | aa5aa72a47e694afbb0cf24f0cc2977a-802865689.us-east-1.elb.amazonaws.com

**THIS IS MY FIRST PROJECT.**
**------ IYAPPAN's ReactApp.**

CICD flow check.
Modified on 25/7/2024 04:55:00 pm
Deployed on 25/7/2024, 4:56:33 pm

DevOps Engineer

- Newly committed changes in the React app source code are visible in the browser.
- The provided timestamp shows the time of the commit.
- The deployment time is also displayed.
- The automated pipeline took only 1.5 minutes to build and deploy the app from the newly committed source code.
- Finally ,The CI/CD pipeline has been created from scratch and tested.