

ICCD Automation in DevOps

- Run Book Documentation

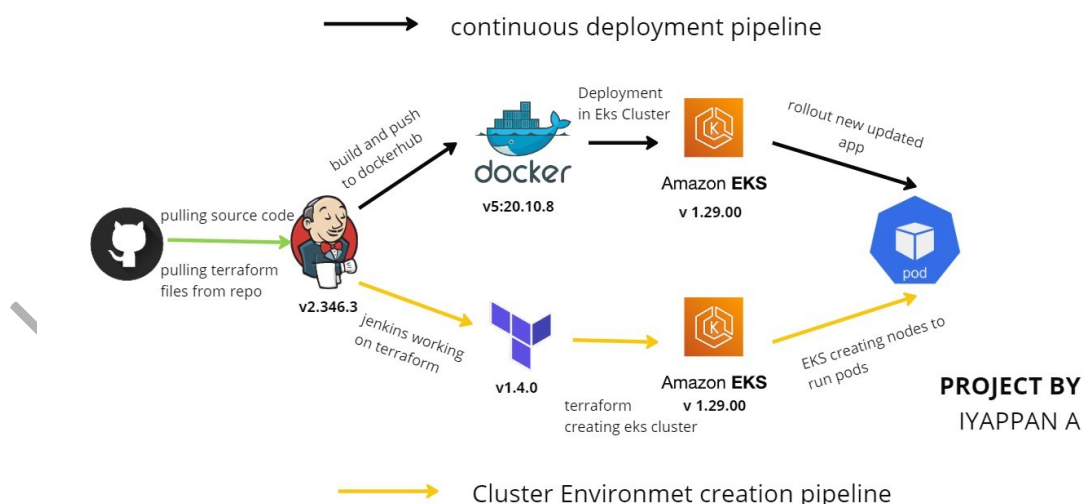
Enumeration of given task :

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:

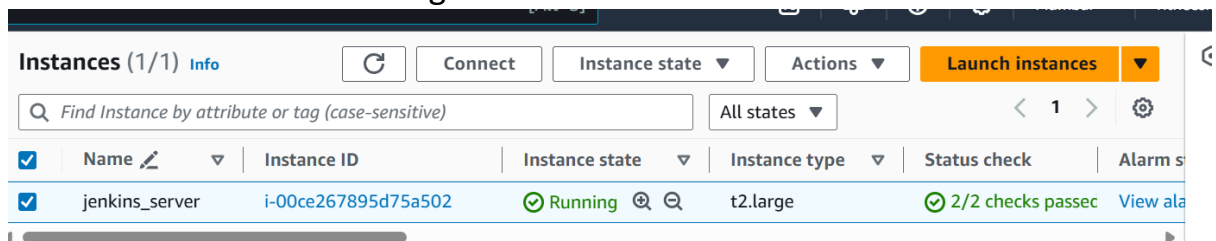
1. Selecting of appropriate tool
 - a. Github - repository
 - b. Docker – containerisation tool
 - c. Terraform – Infrastructure As A Code tool
 - d. AWS EKS – kubernetes cluster formation services from AWS
 - e. Jenkins – Automation tool

2. Creating mind map for workflow

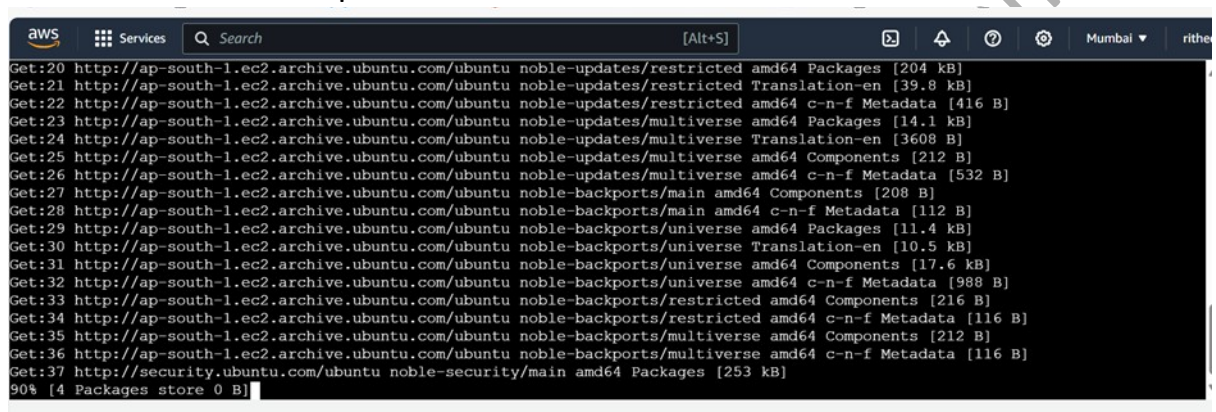


Run book:

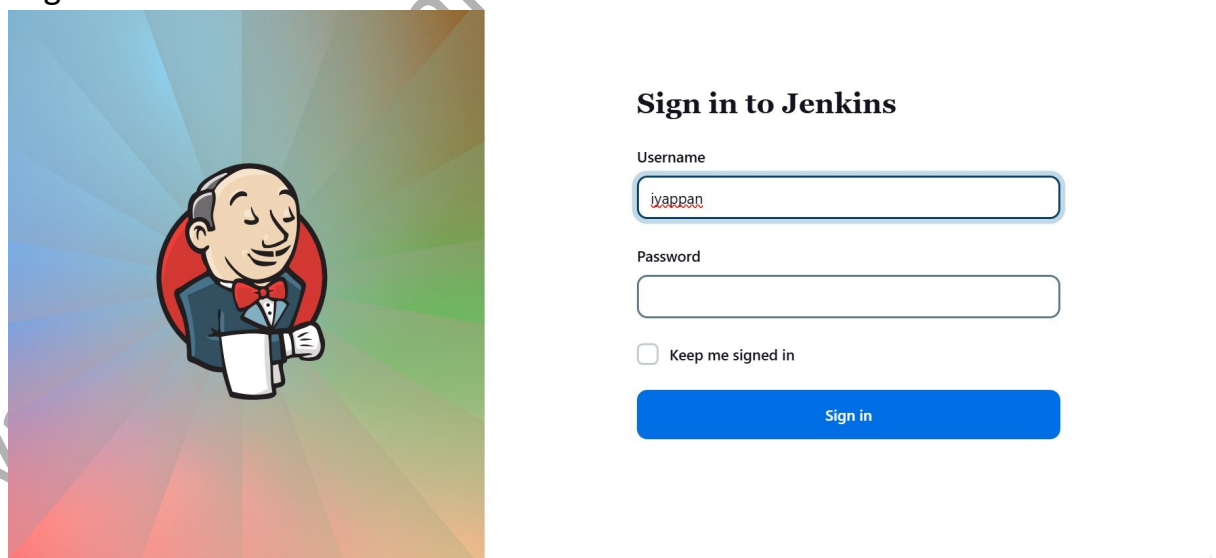
1. Create a instance for running Jenkins



2. Install tools and its dependencies



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



- 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

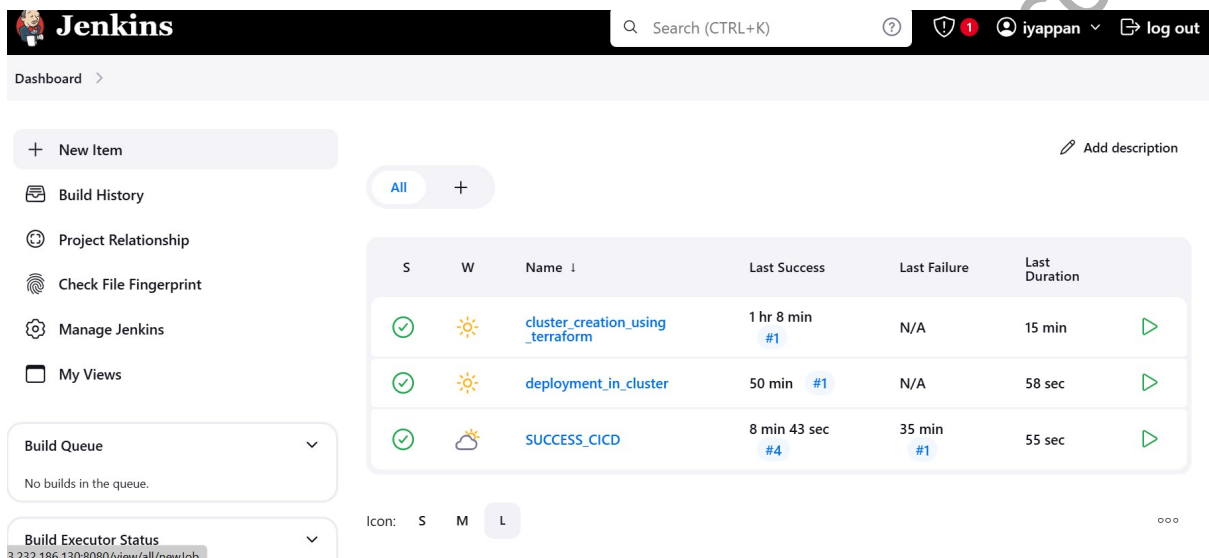
a. Pipeline – cluster_creation_using_terraform

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

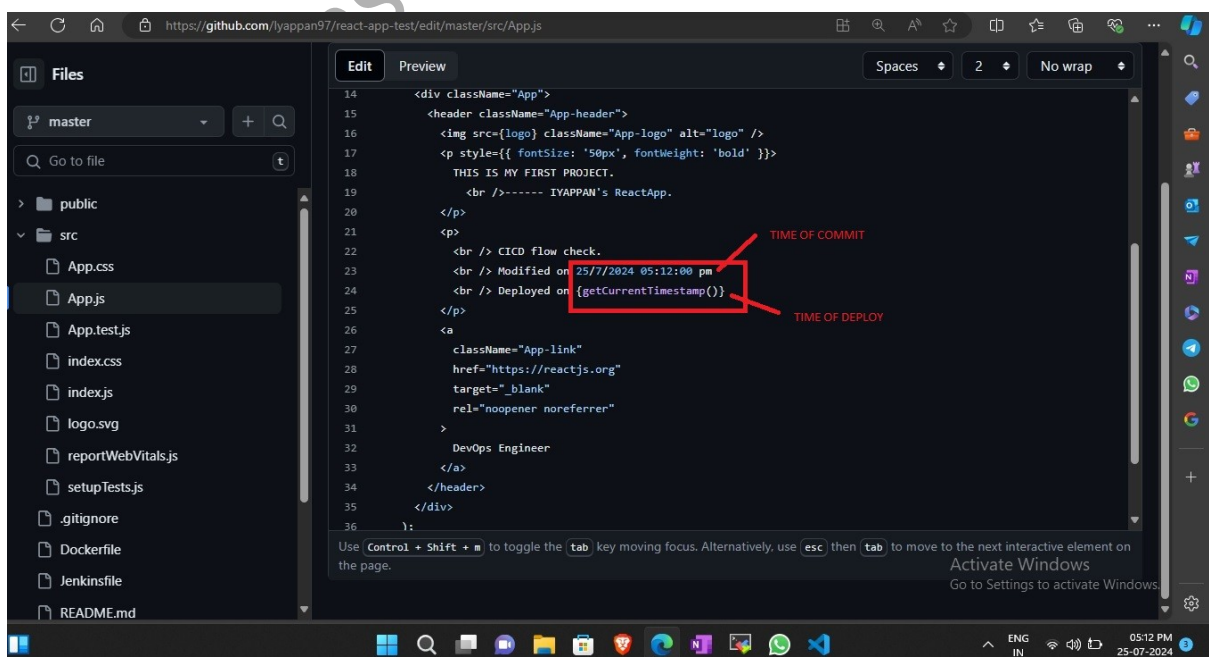


The screenshot shows the Jenkins Dashboard. On the left, there's a sidebar with navigation links: New Item, Build History, Project Relationship, Check File Fingerprint, Manage Jenkins, and My Views. Below these are sections for Build Queue (No builds in the queue) and Build Executor Status. The main area displays a table of pipelines. The table has columns for status (S), warnings (W), name, last success, last failure, last duration, and a play button icon. Three pipelines are listed: 'cluster_creation_using_terraform', 'deployment_in_cluster', and 'SUCCESS_CICD'. The 'SUCCESS_CICD' pipeline is highlighted with a green checkmark and a play button icon.

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



The screenshot shows a Github repository page for 'react-app-test'. The 'App.js' file is selected, and its content is displayed in a code editor. The code is a simple React application. A red box highlights the commit message 'Modified on 25/7/2024 05:12:00 pm' and the deployment command 'getCurrentTimestamp()'. Red arrows point to these elements with labels 'TIME OF COMMIT' and 'TIME OF DEPLOY' respectively. The file explorer on the left shows the project structure, including 'public' and 'src' directories.

```
14 <div className="App">
15 <header className="App-header">
16 <img src={logo} className="App-logo" alt="logo" />
17 <p style={{ fontSize: '50px', fontWeight: 'bold' }}>
18   THIS IS MY FIRST PROJECT.
19 <br />----- IYAPPAN's ReactApp.
20 </p>
21 <p>
22 <br /> CICD flow check.
23 <br /> Modified on 25/7/2024 05:12:00 pm
24 <br /> Deployed on {getCurrentTimestamp()}
25 </p>
26 <a
27   className="App-link"
28   href="https://reactjs.org"
29   target="_blank"
30   rel="noopener noreferrer"
31 >
32   DevOps Engineer
33 </a>
34 </header>
35 </div>
36 </pre>
```

9. Testing

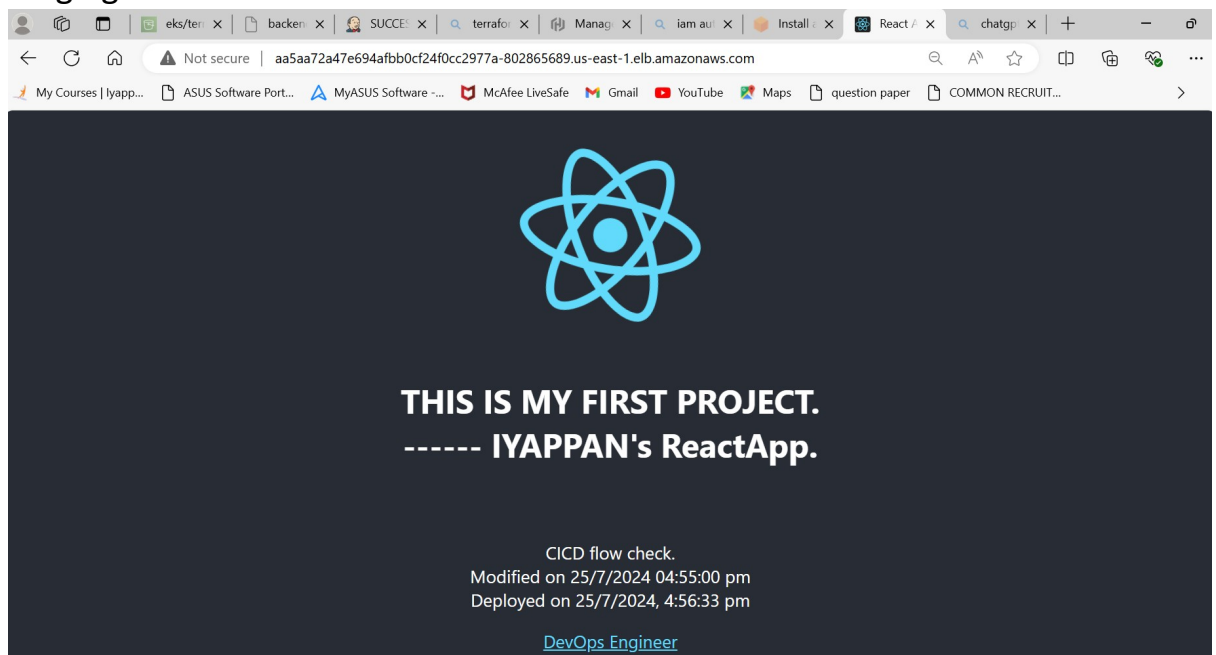
- a. This new commit will automatically trigger pipeline to build (SUCCESS_CICD)

```
[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



- 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.