



**Sri Eshwar**  
**College of Engineering**  
**Coimbatore | Tamilnadu**  
**An Autonomous Institution**  
Affiliated to Anna University, Chennai



## **PROJECT REPORT**

### **DEVOPS AND SITE RELIABILITY ENGINEERING**

**TITLE : YUGIXX, ReactApp, CI/CD Pipeline**

**NAME : NAVEEN PRASANTH P**

**DEPARTMENT: B.E CSE**

**ROLL NO: 23CS103**

**REGISTER NUMBER: 722823104103**

<b>S. No</b>	<b>Contents</b>	<b>Page No</b>
1	Abstract	1
2	Introduction	2
3	Objective	3
4	System Architecture	4
5	Methodology / Workflow	5
6	Implementation Steps	6
7	Screenshots of Execution	9
8	Conclusion and Future Scope	11

## **1. Abstract**

This project demonstrates the practical implementation of a Continuous Integration and Continuous Deployment (CI/CD) pipeline using Jenkins for automating the deployment of a React application. The source code is hosted on GitHub, Jenkins is installed and configured on the local system, and the final deployment is carried out on a remote AWS EC2 instance named nginx. The EC2 instance runs Nginx as the web server to serve the React application.

The main objective of the project is to reduce manual steps in deployment and improve the speed, reliability.

## **2. Introduction**

DevOps is a combination of practices and tools that increase an organization's ability to deliver applications and services efficiently. Continuous Integration and Continuous Deployment (CI/CD) are critical components of DevOps. In this project, Jenkins is used to automate the build and deployment of a React web application, improving development workflows and reducing deployment times.

## **3. Objective**

The objective of this project is to:

- Build and automate a CI/CD pipeline using Jenkins.
- Deploy a React-based web application automatically upon code changes.
- Minimize manual intervention and improve delivery consistency.
- Gain hands-on experience with DevOps tools and practices.

## 4. System Architecture

Developer Pushes Code → GitHub Repository



GitHub Webhook Triggers Jenkins Build



Jenkins Pulls Code → Builds React App



(Optional) Dockerizes and Deploys to Server



Application Available for Use

## 5. Methodology / Workflow

The development and deployment process followed these key steps:

- The React application was created and hosted on GitHub.
- Jenkins was installed and configured for CI/CD.
- A Jenkins pipeline job was set up to pull the latest code from GitHub.
- Webhooks were added for automatic triggering on every commit.

- The pipeline was defined to install dependencies, build the app, and deploy it.
- Post-build actions ensured the updated app was live instantly.

## **6. Implementation Steps**

### **1. Create React App**

- Used `npx create-react-app my-app`.

### **2. Push to GitHub**

- Initialize Git and push the app to a remote GitHub repository.

### **3. Install Jenkins**

- Jenkins was installed on a server (local or cloud-based).

### **4. Configure Jenkins**

- Installed necessary plugins (Git, NodeJS, Pipeline).
- Created a freestyle or pipeline job.

### **5. Connect GitHub**

- Configured GitHub Webhook to trigger Jenkins on code changes.

## 6. Pipeline Stages

- Install dependencies: `npm install`
- Build app: `npm run build`
- Deploy: Copy build files to server (or use Docker/Nginx)

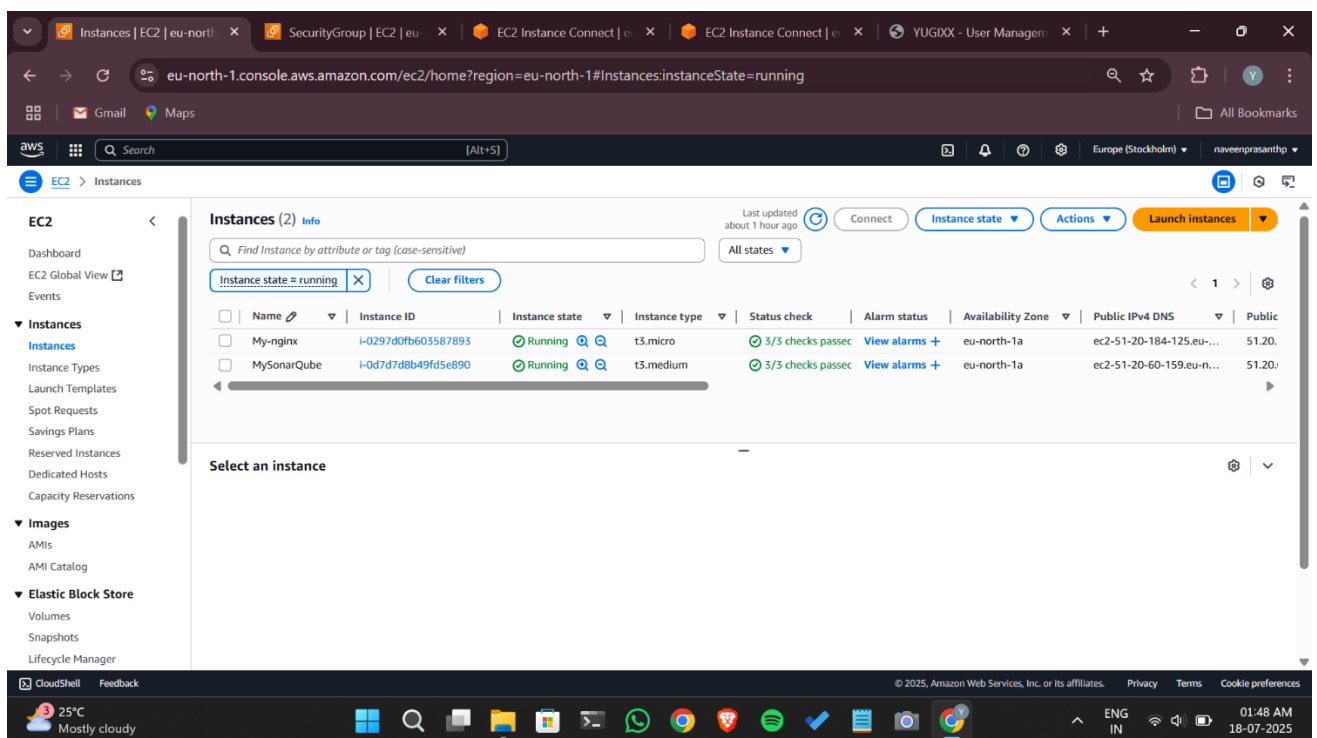
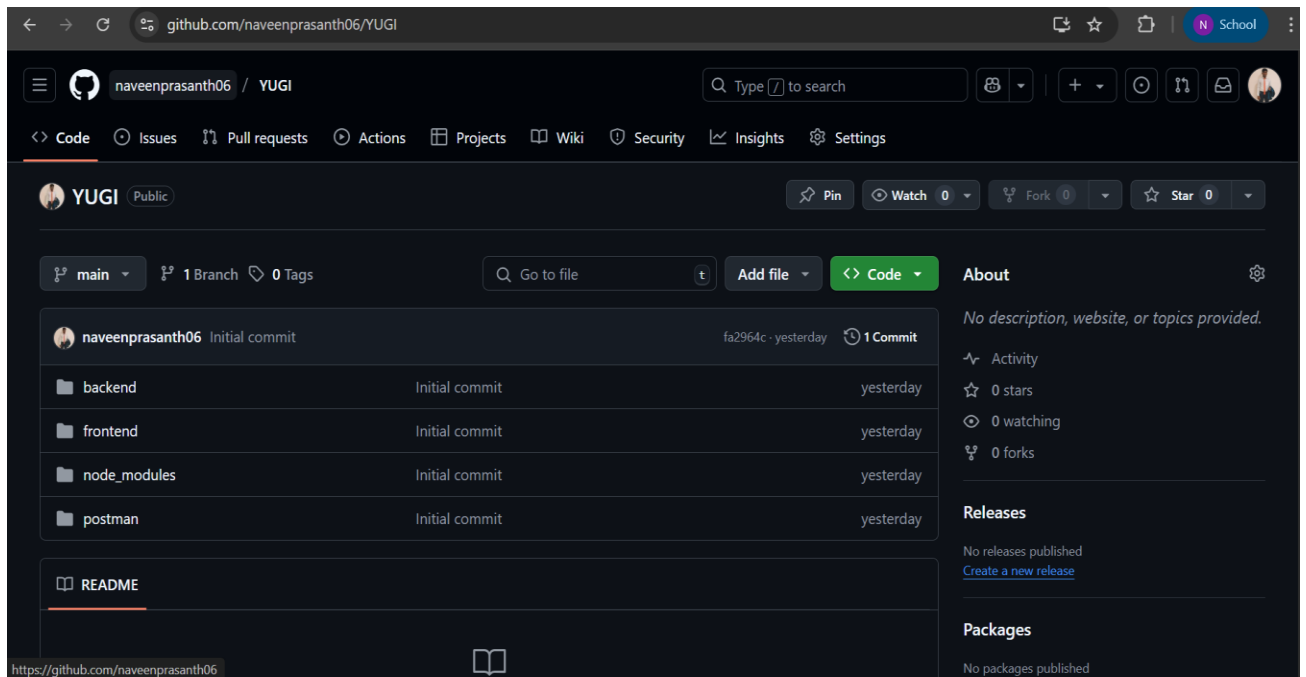
## 7. Automation Verified

- Made a code change → commit → auto trigger → deployed output verified.

## 7. Screenshots of Execution

Include the following screenshots with proper captions:

- Jenkins Job Configuration
- Webhook Setup in GitHub
- Console Output of Jenkins Build
- React App Output in Browser
- Jenkins Pipeline View



eu-north-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?region=eu-north-1&connType=standard&instanceId=i-0d7d7d8b49fd5e8...

```
Memory usage: 8%      Users logged in: 0
Swap usage:  0%      IPv4 address for ens5: 172.31.4.194

* Ubuntu Pro delivers the most comprehensive open source security and
  compliance features.

https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

80 updates can be applied immediately.
56 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Thu Jul 17 16:25:20 2025 from 13.48.4.202
ubuntu@sonarqube:~$ sudo -i
root@sonarqube:~$ sudo vi /etc/sysctl.conf
root@sonarqube:~$ reboot

Broadcast message from root@ip-172-31-4-194 on pts/3 (Thu 2025-07-17 16:46:22 UTC):

The system will reboot now!

root@sonarqube:~$
```

i-0d7d7d8b49fd5e890 (MySonarQube)

PublicIPs: 51.20.60.159 PrivateIPs: 172.31.4.194

CloudShell Feedback

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

25°C Mostly cloudy

ENG IN 01:48 AM 18-07-2025

eu-north-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?region=eu-north-1&connType=standard&instanceId=i-0297d0fb6035878...

```
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<title>YUGIXX - User Management Platform</title>
<script type="module" crossorigin src="/assets/index-C7JaaQ_j.js"></script>
<link rel="stylesheet" crossorigin href="/assets/index-D_uCs2F1.css">
</head>
<body>
  <div id="root"></div>
</body>
</html> ubuntu@ip-172-31-7-25:/var/www/html$ sudo systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: enabled)
   Active: active (running) since Thu 2025-07-17 10:35:51 UTC; 9h ago
     Docs: man:nginx(8)
   Main PID: 2127 (nginx)
    Tasks: 3 (limit: 1072)
   Memory: 2.5M (peak: 5.0M)
      CPU: 42ms
   CGroup: /system.slice/nginx.service
           └─2127 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
             └─2129 "nginx: worker process"
               └─2130 "nginx: worker process"

Jul 17 10:35:51 ip-172-31-7-25 systemd[1]: Starting nginx.service - A high performance web server and a reverse proxy server...
Jul 17 10:35:51 ip-172-31-7-25 systemd[1]: Started nginx.service - A high performance web server and a reverse proxy server.
ubuntu@ip-172-31-7-25:/var/www/html$ ^C
ubuntu@ip-172-31-7-25:/var/www/html$ ^C
ubuntu@ip-172-31-7-25:/var/www/html$ sudo systemctl restart nginx
ubuntu@ip-172-31-7-25:/var/www/html$
```

i-0297d0fb603587893 (My-nginx)

PublicIPs: 51.20.184.125 PrivateIPs: 172.31.7.25

CloudShell Feedback

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

25°C Mostly cloudy

ENG IN 01:48 AM 18-07-2025



Dashboard - Jenkins x +

http://localhost:8080

# Jenkins

Dashboard >

+ New Item

Build History

Project Relationship

Check File Fingerprint

Manage Jenkins

Add description

S	W	Name ↓	Last Success	Last Failure	Last Duration
🟢	🔔	YUGIXX	1 hr 14 min #15	N/A	3 min 55 sec

Build Queue

No builds in the queue.

Build Executor Status 0/2

Icon: S M L

REST API Jenkins 2.504.3

All - Jenkins x +

http://localhost:8080/view/all/builds

# Jenkins

Dashboard > All > Build History

+ New Item

Build History

Project Relationship

Check File Fingerprint

Manage Jenkins

## Build History of Jenkins

S	Build	Time Since ↑	Status
🟢	YUGIXX #15	1 hr 15 min	stable
🟢	YUGIXX #12	1 hr 48 min	stable

Icon: S M L

Build Queue

No builds in the queue.

Build Executor Status 0/2

http://localhost:8080/view/all/builds

REST API Jenkins 2.504.3

YUGIXX - Jenkins

http://localhost:8080/job/YUGIXX/

# Jenkins

Dashboard > YUGIXX >

Status ✓ YUGIXX [Add description](#)

Changes

Build Now

Configure

Delete Pipeline

SonarQube

Stages

Rename

Pipeline Syntax

## SonarQube Quality Gate

naveen-project **Passed**

server-side processing: **Success**

### Permalinks

- Last build (#15), 1 hr 15 min ago
- Last stable build (#15), 1 hr 15 min ago
- Last successful build (#15), 1 hr 15 min ago
- Last completed build (#15), 1 hr 15 min ago

Builds

Filter

# Jenkins

Dashboard > YUGIXX > Stages

## Stages

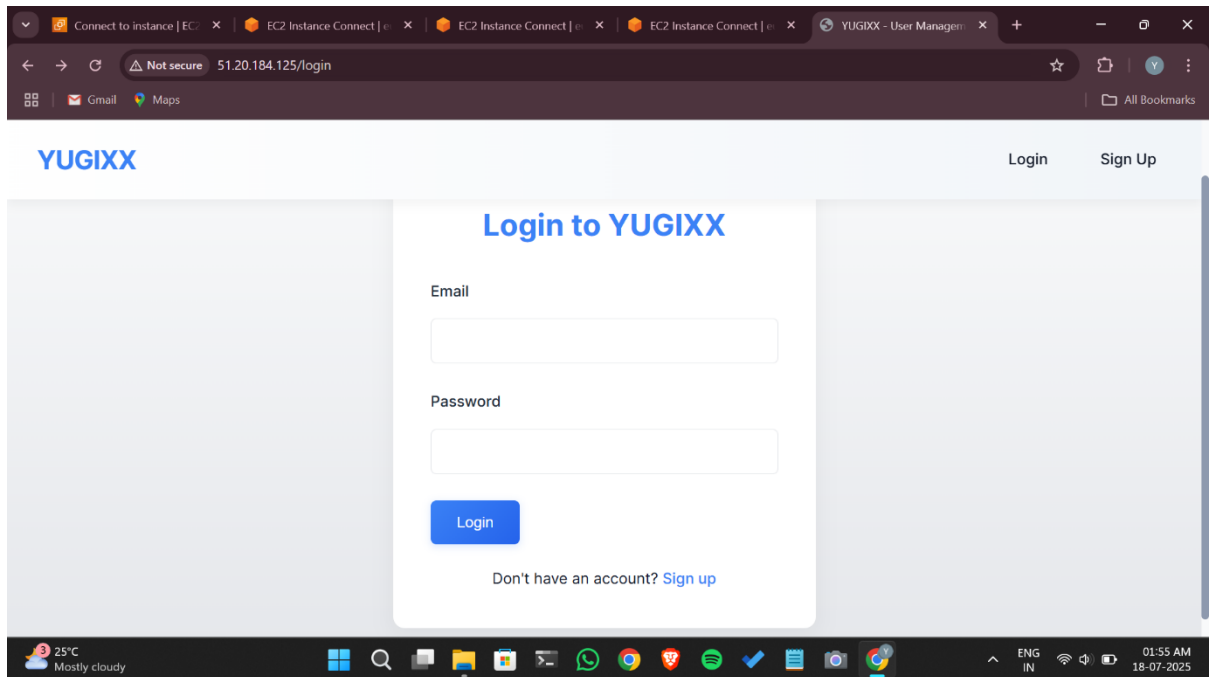
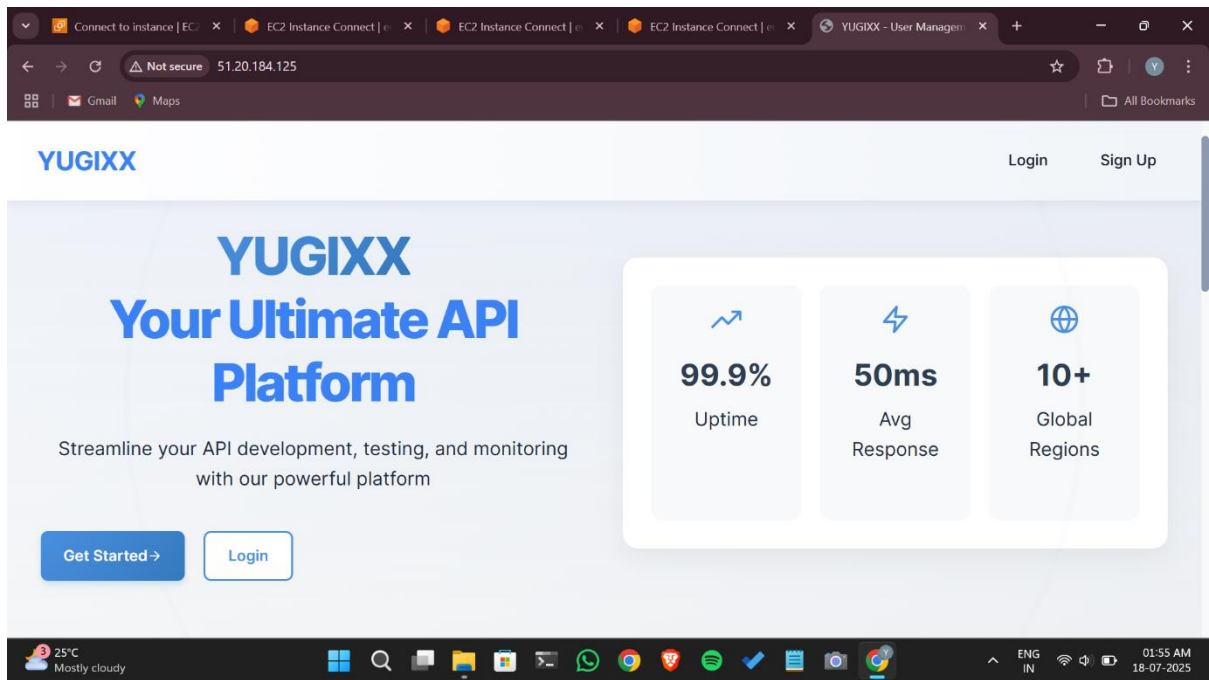
July 18, 2025

[Build](#) [...](#)

#15  
01:08 - 3m 55s

Start Tool Install Package... SonarQub... Build... Deploy to EC2 End

Jenkins 2.504.3



## **8. Conclusion and Future Scope**

This project successfully demonstrated the use of CI/CD with Jenkins for automating the deployment of a React application. It reduced manual deployment time and errors. In the future, this can be enhanced by integrating Docker for containerization and Kubernetes for orchestrated deployments, making it scalable and cloud-ready.