

This document provides a comprehensive, step-by-step guide to deploying a production-ready React application using modern DevOps practices. It walks through containerizing the application with Docker, setting up a continuous integration and deployment pipeline using Jenkins, hosting the application on AWS EC2, and implementing open-source monitoring using Prometheus and Grafana. The guide is designed for developers and DevOps professionals seeking to automate and scale application deployments while ensuring reliability through effective monitoring and alerting systems.

React App Deployment - Complete Step- by-Step Guide & Project Report

JUNE 2

**Author : SREEJESH T
GUVI Final Project**

Table of Contents

Objective	2
Project Setup	2
1. AWS EC2 Setup.....	2
1.1 Launch EC2 Instance	2
1.2 Create Your GitHub Repo.....	3
1.2 Clone Starter Code.....	4
1.3 Initialize and Push Project to GitHub	4
1.5 React App Dockerfile Setup	5
2. Docker Hub Configuration	6
2.1 Create two repositories:	6
2.2 Docker Hub Repository Creation from GUI	6
3. Jenkins CI/CD Setup	9
3.1 Install Jenkins	9
3.2 Add Credentials	9
3.3 Jenkinsfile	9
5. Monitoring with Prometheus + Grafana	17
5.1 Create Docker Network	17
5.2 Prometheus Config	17
5.3 Run Prometheus and Grafana.....	18
5.4 Configure Grafana	18
6.0 Brief, Auto trigger Jenkins & Monitoring	20

Objective

Deploy a React application using Docker, set up a CI/CD pipeline with Jenkins, host it on AWS EC2, and configure monitoring using Prometheus and Grafana.

Project Setup

1. AWS EC2 Setup

1.1 Launch EC2 Instance

Launch EC2 Instance (Ubuntu 22.04, t2.micro)

Go to AWS EC2 Dashboard

Click Launch Instance

Choose AMI: Ubuntu Server 22.04 LTS

Choose instance type: t2.micro (Free Tier eligible)

Configure storage and tags as needed (defaults are fine)

Create or select a key pair (you will need the .pem file later)

Configure Security Group:

Allow SSH (port 22)

Allow HTTP (port 80)

Allow Prometheus (port 9090)

Allow Grafana (port 3000)

Launch the instance

The screenshot shows a green success message: "Success: Successfully initiated launch of instance (i-0b74f33cba2916555)". Below the message is a "Launch log" button. At the bottom, there's a "Next Steps" section with a search bar and a "Create alarm" button.

The screenshot shows the security group rules for the launched instance. It lists four rules: one for HTTP (port 80), one for SSH (port 22), one for HTTPS (port 443), and two for custom TCP ports (9090 and 3000). Each rule has a "Delete" button.

The screenshot shows the AWS EC2 Instances details page for instance i-0e2b6839a1db782b3. Key details include:

- Instance ID:** i-0e2b6839a1db782b3
- Public IP:** 35.90.101.84
- Instance State:** Running
- VPC ID:** vpc-080727b66df770224

After launch, connect via SSH: ssh -i "your-key.pem" ubuntu@<EC2-PUBLIC-IP>

ssh -l "your-key.pem" ubuntu@ ec2-35-90-101-84.us-west-2.compute.amazonaws.com

1.2 Create Your GitHub Repo

Go to <https://github.com/sreejesht>

Repository Name: react-app-deployment

Set to Public

Do not add README or. Gitignore

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

Required fields are marked with an asterisk (*).

Repository template

No template

Start your repository with a template repository's contents.

Owner *

sreejesht

Repository name *

react-app-deploy-ec2-docker-jenkins

Your new repository will be created as react-app-deploy-ec2-docker-jenkins-. The repository name can only contain ASCII letters, digits, and the characters ., -, and _.

Great repository names are short and memorable. Need inspiration? How about super-duper-tribble ?

Description (optional)

react-app-deploy-ec2-docker-jenkins. repo created for react App project

Public

Anyone on the internet can use this repository. You choose who can commit.

Private

You choose who can see and commit to this repository.

Initialize this repository with:

Add a README file

This is where you can write a long description for your project. [Learn more about READMEs](#)

Add .gitignore

gitignore template None

Choose which files not to track from a list of templates. [Learn more about ignoring files](#)

Choose a license

License Now

A license tells others what they can and can't do with your code. [Learn more about licenses](#)

① You are creating a public repository in your personal account.

[Create repository](#)

<https://github.com/sreejesht/react-app-deployment.git>

1.2 Clone Starter Code

cd react-app-deploy-ec2-docker-jenkins

git clone <https://github.com/sriram-R-krishnan/devops-build.git>

cd devops-build

1.3 Initialize and Push Project to GitHub

cd react-app-deploy-ec2-docker-jenkins

Copy project files here (write all required files here, including dockerfile, bash scripts)

git checkout -b dev

git add .

git commit -m "Initial project files for Docker + Jenkins + Monitoring"

git push origin dev

```
[root@localhost react-app-deployment]# git init
Reinitialized existing Git repository in /root/react-app-deployment/.git/
[root@localhost react-app-deployment]# git add .
[root@localhost react-app-deployment]# git commit -m "Initial commit"
[dev 3ff52d3] Initial commit
 3 files changed, 12 insertions(+), 6 deletions(-)
 create mode 160000 devops-build
[root@localhost react-app-deployment]# git push origin dev
Username for 'https://github.com': sreejesht
Password for 'https://sreejesht@github.com':
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 2 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 752 bytes | 752.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/sreejesht/react-app-deployment.git
   2252e7e...3ff52d3  dev -> dev
[root@localhost react-app-deployment]# ll
total 24
-rw-rw-rwx. 1 root root 400 May 26 00:46 build.sh
-rw-rw-rwx. 1 root root 355 May 26 00:47 deploy.sh
drwxr-xr-x. 4 root root 31 May 26 00:45 devops-build
-rw-r--r--. 1 root root 75 May 25 09:30 docker-compose.yml
-rw-r--r--. 1 root root 58 May 25 09:43 Dockerfile
-rw-r--r--. 1 root root 765 May 25 09:30 Jenkinsfile
-rw-r--r--. 1 root root 269 May 25 09:30 README.md
[root@localhost react-app-deployment]# 
```

1.4 Install required packages

cd react-app-deploy-ec2-docker-jenkins/

./install_pkgs.sh

```
ubuntu@ip-172-31-47-113:~/react-app-deploy-ec2-docker-jenkins$ cat install_pkgs.sh
#!/bin/bash

# -----
# Script Name : install_pkgs.sh
# Description : Installs Docker, Java (OpenJDK 11), and Jenkins on a Debian-based system
# Author      : Sreejesh T
# -----


set -e # Exit immediately if a command exits with a non-zero status
echo "Starting package installation..."

# Update package index
echo "Updating package index..."
sudo apt-get update -y

# Install Java (OpenJDK 11)
echo "Installing OpenJDK 11..."
sudo apt-get install -y openjdk-11-jre

# Install Docker
echo "Installing Docker..."
sudo apt-get install -y docker.io

# Install Jenkins
echo "Setting up Jenkins repository and key..."
sudo wget -O /usr/share/keyrings/jenkins-keyring.asc https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
echo "Adding Jenkins repository to sources list..."
echo "deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] https://pkg.jenkins.io/debian-stable binary/" | sudo tee /etc/apt/sources.list.d/jenkins.list > /dev/null
```

```

echo "Updating package index with Jenkins repository..."
sudo apt-get update -y
echo "Installing Jenkins..."
sudo apt-get install -y jenkins

# Verify installations
echo "Verifying installations..."
echo "Java version:"
java --version

echo "Jenkins version:"
jenkins --version || echo "Jenkins is installed but not yet started."
echo "Docker version:"
docker --version

echo "Package installation completed successfully."
ubuntu@ip-172-31-47-113:~/react-app-deploy-ec2-docker-jenkins$ 

```

./ install_and_configure_git.sh

```

ubuntu@ip-172-31-47-113:~/react-app-deploy-ec2-docker-jenkins$ cat install_and_configure_git.sh
#!/bin/bash

# -----
# Script Name : install_and_configure_git.sh
# Description : Installs Git and configures global username/email
# Author      : Sreejesh T
# -----


set -e

echo " Updating package index..."
sudo apt update

echo " Installing Git..."
sudo apt install -y git

echo " Git installed successfully:"
git --version

echo " Configuring Git global settings..."
git config --global user.name "Sreejesh T"
git config --global user.email "sreejesh.t@gmail.com"

echo " Verifying Git configuration..."
git config --global --list

echo " Git setup completed successfully for user: Sreejesh T"
ubuntu@ip-172-31-47-113:~/react-app-deploy-ec2-docker-jenkins$ 

```

1.5 React App Dockerfile Setup

1.5.1 Create Dockerfile

```
cd /home/ubuntu/react-app-deploy-ec2-docker-jenkins/devops-build
```

```

ubuntu@ip-172-31-47-113:~/react-app-deploy-ec2-docker-jenkins/devops-build$ tree
.
└── Dockerfile
└── build
    ├── _redirects
    ├── asset-manifest.json
    ├── favicon.ico
    ├── index.html
    ├── logo192.png
    ├── logo512.png
    ├── manifest.json
    ├── robots.txt
    └── static
        ├── css
        │   ├── main.cf5c13c5.css
        │   └── main.cf5c13c5.css.map
        └── js
            ├── 787.2f5360e2.chunk.js
            ├── 787.2f5360e2.chunk.js.map
            ├── main.flc48542.js
            └── main.flc48542.js.LICENSE.txt
            └── main.flc48542.js.map

5 directories, 16 files

```

vi Dockerfile

```
# Use official Nginx base image for serving static content
FROM nginx:alpine

# Clean the default html directory
RUN rm -rf /usr/share/nginx/html/*

# Copy the React production build into Nginx html directory
COPY build/ /usr/share/nginx/html/

# Expose port 80 to allow web traffic
EXPOSE 80

# Start Nginx in the foreground
CMD ["nginx", "-g", "daemon off;"]
```

2. Docker Hub Configuration

2.1 Create two repositories:

react-app-dev (Public)
react-app-prod (Private)
Docker ID: screedocker911
Create Docker Repos:
react-app-dev (Public)
react-app-prod (Private)

2.2 Docker Hub Repository Creation from GUI

2.2.1. Login to Docker Hub

Go to:

<https://hub.docker.com/>

Sign in using your Docker ID: screedocker911

2.2.2. Create react-app-dev Repository (Public)

Click your profile icon (top-right corner) → "Repositories" or go directly to:

<https://hub.docker.com/repositories>

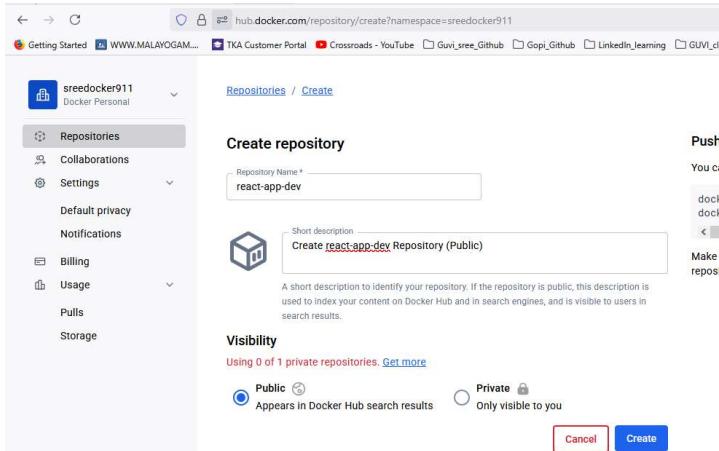
Click the "Create Repository" button.

Fill out the form:

Repository Name: react-app-dev

Visibility: Select Public

Click "Create".



2.2.3. Create react-app-prod Repository (Private)

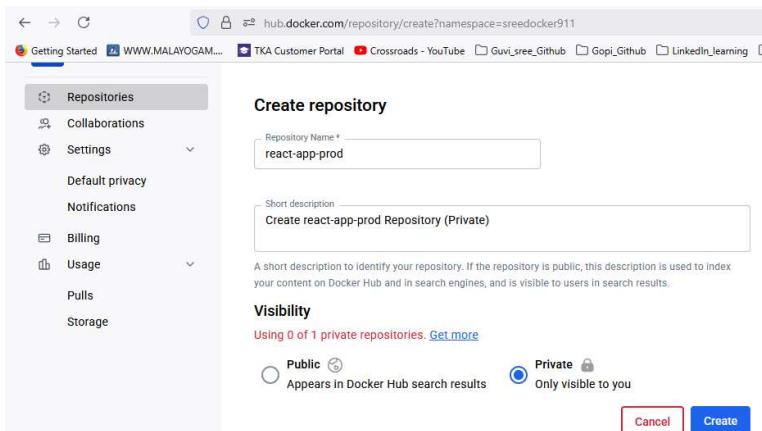
Again, go to "Create Repository".

Fill out:

Repository Name: react-app-prod

Visibility: Select Private

Click "Create".



From your terminal (after building the images & created the repo in Docker hub):

2.2.4 Run tagging & push images to respective repo,

```
[root@localhost ~]# docker login
Authenticating with existing credentials... [Username: sreedocker911]
[!] Info - To login with a different account, run 'docker logout' followed by 'docker login'

Login Succeeded
[root@localhost ~]# docker tag sreedocker911/react-app:dev sreedocker911/react-app-dev:latest
[root@localhost ~]# docker tag sreedocker911/react-app:prod sreedocker911/react-app-prod:latest
[root@localhost ~]# docker push sreedocker911/react-app-dev:latest
The push refers to repository [docker.io/sreedocker911/react-app-dev]
9bb98e7bbdal: Pushed
7f0c8e165d83: Pushed
1f2d1669327d: Pushed
82140d9a70a7: Mounted from library/node
f3b40b0ccdb1c: Mounted from library/node
0b1f26057bd0: Mounted from library/node
080000c18d16d: Mounted from library/node
latest: digest: sha256:d6c8d18bbcd2a067421237d7af3f7ab38a1387e415e051276cd3a414036bbel9 size: 1785
[root@localhost ~]#
```

[root@localhost ~]# docker push sreedocker911/react-app-prod:latest
The push refers to repository [docker.io/sreedocker911/react-app-prod]
9bb98e7bbdal: Mounted from sreedocker911/react-app-dev
7f0c8e165d83: Mounted from sreedocker911/react-app-dev
1f2d1669327d: Mounted from sreedocker911/react-app-dev
82140d9a70a7: Mounted from sreedocker911/react-app-dev
f3b40b0cc0bc: Mounted from sreedocker911/react-app-dev
0b1f26057bd0: Mounted from sreedocker911/react-app-dev
08000c18d16d: Mounted from sreedocker911/react-app-dev
latest: digest: sha256:a6c8d18bbcd2a067421237d7a3f7ab38a1387e415e051276cd3a414036bbe19 size: 1785
[root@localhost ~]#

View local images
docker images

```
[root@localhost react-app-deployment]# docker images
REPOSITORY          TAG      IMAGE ID      CREATED       SIZE
sreedocker911/react-app-dev    latest   398b8f7321dc  About an hour ago  142MB
sreedocker911/react-app-prod    latest   398b8f7321dc  About an hour ago  142MB
sreedocker911/react-app        dev     398b8f7321dc  About an hour ago  142MB
sreedocker911/react-app        prod    398b8f7321dc  About an hour ago  142MB
```

3. Jenkins CI/CD Setup

3.1 Install Jenkins

Install on the same EC2 or another server or on container

Install plugins: Docker, GitHub, Pipeline, SSH Agent

3.2 Add Credentials

For DockerHub: <https://hub.docker.com/repositories/sreedocker911>

For EC2 SSH Key:

For Github : <https://github.com/sreejesht/react-app-deploy-ec2-docker-jenkins>

3.3 Jenkinsfile

```
1 pipeline {
2     agent any
3
4     environment {
5         IMAGE_NAME = "react-static-app"
6         DOCKERHUB_USER = "sreedocker911"
7     }
8
9     triggers {
10        pollSCM('* * * * *')
11    }
12
13     stages {
14         stage('Clone Repo') {
15             steps {
16                 script {
17                     def branchToUse = env.BRANCH_NAME ?: 'dev'
18                     echo "Cloning branch: ${branchToUse}"
19                     checkout([>class: 'GitSCM',
20                         branches: [name: "${branchToUse}"],
21                         userRemoteConfigs: [[url: 'https://github.com/sriram-R-Krishnan/devops-build']]]
22                 })
23             }
24         }
25     }
26
27     stage('Build Docker Image') {
28         steps {
29             echo "Building Docker image..."
30             sh './build.sh'
31         }
32     }
33
34     stage('Tag Docker Images') {
35         steps {
36             echo "Tagging Docker image for dev/prod..."
37             sh '''
38                 docker tag react-static-app $DOCKERHUB_USER/react-app-dev:latest
39                 docker tag react-static-app $DOCKERHUB_USER/react-app-prod:latest
40             '''
41         }
42     }
43
44     stage('Push to DockerHub') {
45         steps {
46             withCredentials([usernamePassword(credentialsId: 'dockershub-creds', usernameVariable: 'DOCKER_USER', passwordVariable: 'DOCKER_PASS')])
47             sh '''
48                 echo $DOCKER_PASS | docker login -u $DOCKER_USER --password-stdin
49                 if [ "$BRANCH_NAME" == "master" ]; then
50                     docker push $DOCKERHUB_USER/react-app-prod:latest
51                 else
52                     docker push $DOCKERHUB_USER/react-app-dev:latest
53                 fi
54             ...
55         }
56     }
57
58     stage('Deploy Container') {
59         steps {
60             echo "Deploying container..."
61             sh './deploy.sh'
62         }
63     }
64 }
65
66 post {
67     success {
68         echo "CI/CD Pipeline completed successfully."
69     }
70     failure {
71         echo "CI/CD Pipeline failed."
72     }
73 }
74 }
```

build.sh

```
ubuntu@ip-172-31-47-113: ~/react-app-deploy-ec2-docker-jenkins
```

```
#!/bin/bash
echo "Building Docker image for React app..."
docker build -t react-static-app ./devops-build
```

deploy.sh

```

#!/bin/bash
echo "Deploying React app container..."

# Stop and remove existing container if it exists
docker stop react-app
docker rm react-app
# Run new container on devops-net with port 80 exposed
docker run -d --name react-app --network devops-net -p 80:80 react-static-app
echo "react-app container deployed and running on http://localhost:80"

```

docker-compose.yml

```

ubuntu@ip-172-31-47-113: ~/react-app-deploy-ec2-docker-jenkins

version: '3.8'
services:
  jenkins:
    build:
      context: .
      dockerfile: Dockerfile.jenkins
    container_name: jenkins
    restart: unless-stopped
    ports:
      - "8080:8080"
      - "50000:50000"
    volumes:
      - /var/run/docker.sock:/var/run/docker.sock
    networks:
      - devops-net

  react-app:
    build:
      context: ./devops-build
      dockerfile: Dockerfile
    container_name: react-app
    restart: unless-stopped
    ports:
      - "80:80"
    networks:
      - devops-net

  node-exporter:
    image: prom/node-exporter
    container_name: node-exporter
    restart: unless-stopped
    ports:
      - "9100:9100"
    networks:
      - devops-net

volumes:
  jenkins_home:

networks:
  devops-net:
    driver: bridge

```

"docker-compose.yml" 42L, 753B

chmod 777 build.sh deploy.sh

```
[root@localhost react-app-deployment]# ll
total 24
-rwxrwxrwx. 1 root root 178 May 25 18:55 build.sh
-rwxrwxrwx. 1 root root 427 May 25 19:10 deploy.sh
-rw-r--r--. 1 root root 75 May 25 09:30 docker-compose.yml
-rw-r--r--. 1 root root 588 May 25 09:43 Dockerfile
-rw-r--r--. 1 root root 765 May 25 09:30 Jenkinsfile
-rw-r--r--. 1 root root 269 May 25 09:30 README.md
[root@localhost react-app-deployment]#
```

docker-compose.monitoring.yml

 ubuntu@ip-172-31-47-113: ~/react-app-deploy-ec2-docker-jenkins

Git repo update with latest changes,

git status

git add .

git commit -m "repo updates with latest changes"

git push origin dev

username : sreejesht

passwd : <give the token to login ghp_g9wLlj2VJ98gD9ru*>

```
[root@localhost react-app-deployment]# git add .
[root@localhost react-app-deployment]# git commit -m "Initial project files"
[dev 2252e7e] Initial project files
 2 files changed, 23 insertions(+), 2 deletions(-)
[root@localhost react-app-deployment]# git push origin dev
Username for 'https://github.com': sreejesht
Password for 'https://sreejesht@github.com':
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 2 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 656 bytes | 218.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/sreejesht/react-app-deployment.git
 8acfcb0a..2252e7e  dev -> dev
[root@localhost react-app-deployment]#
```

sudo curl -L "https://github.com/docker/compose/releases/download/1.29.2/docker-compose-\$(uname -s)-\$(uname -m)" -o /usr/local/bin/docker-compose

```
ubuntu@ip-172-31-47-113:~/react-app-deploy-ec2-docker-jenkins$ docker-compose up -d --build
Command 'docker-compose' not found, but can be installed with:
sudo snap install docker      # version 28.1.1, or
sudo snap install docker      # version 27.5.1
sudo apt install docker-compose # version 1.29.2-6
See 'snap info <snapname>' for additional versions.
ubuntu@ip-172-31-47-113:~/react-app-deploy-ec2-docker-jenkins$ sudo curl -L "https://github.com/docker/compose/releases/download/1.29.2/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
% Total    % Received % Xferd  Average Speed   Time   Time  Current
          Dload Upload   Total Spent  Left Speed
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
100 12.1M 100 12.1M 0  0  3377k  0  0:00:03 0:00:03 3917k
ubuntu@ip-172-31-47-113:~/react-app-deploy-ec2-docker-jenkins$ sudo chmod +x /usr/local/bin/docker-compose
ubuntu@ip-172-31-47-113:~/react-app-deploy-ec2-docker-jenkins$ docker-compose --version
docker-compose version 1.29.2, build 5becea4c
ubuntu@ip-172-31-47-113:~/react-app-deploy-ec2-docker-jenkins$
```

- ✓ docker stop -f jenkins react-app node-exporter
- ✓ docker rm -f jenkins react-app node-exporter
- ✓ docker-compose down
- ✓ docker-compose build Jenkins
- ✓ docker-compose up -d
- ✓ docker network inspect devops-net

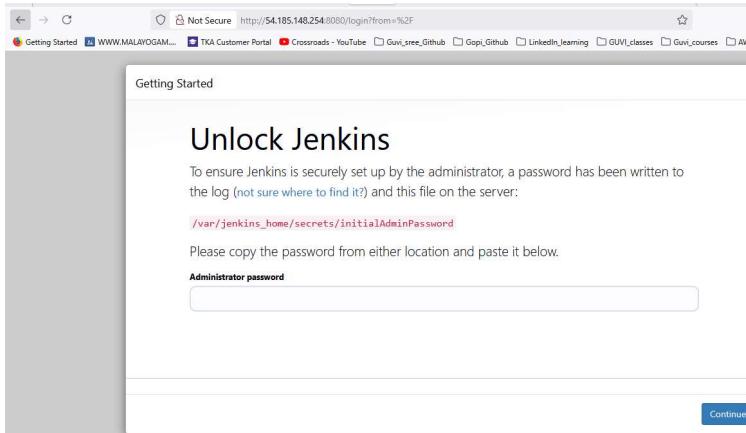
✓ docker-compose -f docker-compose.yml up -d –build

```
ubuntu@ip-172-31-47-113:~/react-app-deploy-ec2-docker-jenkins$ docker-compose up -d --build
Creating network "react-app-deploy-ec2-docker-jenkins_devops-net" with driver "bridge"
Creating volume "react-app-deploy-ec2-docker-jenkins_jenkins_home" with default driver
Building react-app
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
  Install the buildx component to build images with BuildKit:
    https://docs.docker.com/go/buildx/
Sending build context to Docker daemon 2.62MB
Step 1/5 : FROM nginx:alpine
alpine: Pulling from library/nginx
f19232174bc9: Pull complete
61caef733c80: Pull complete
b446cf0fae3: Pull complete
d7e507024086: Pull complete
91bdded7ec67: Pull complete
197eb5867ef: Pull complete
241615a2a000: Pull complete
39c2addfd6010: Pull complete
Digest: sha256:65645c7b6a061892a8b03b89d0743208a19dd2f3f17a54ef4b76fb8e2f2a10
Status: Downloaded newer image for nginx:alpine
---- 6769dc3a703c
Step 2/5 : RUN rm -rf /usr/share/nginx/html/*
---- Running in 525923ad5c5f
---- Removed intermediate container 525923ad5c5f
---- efd1e10dafa2
Step 3/5 : COPY build/ /usr/share/nginx/html/
---- 601545dfa2b3
Step 4/5 : EXPOSE 80
---- Running in b7d09e95fe4a
---- Removed intermediate container b7d09e95fe4a
---- 7422360f
Step 5/5 : CMD ["nginx", "-g", "daemon off;"]
---- Running in 271a3b49cc02
---- Removed intermediate container 271a3b49cc02
---- bae3c778df8f
Successfully built bae3c778df8f
Successfully tagged react-app-deploy-ec2-docker-jenkins_react-app:latest
Creating jenkins ... done
Creating react-app ... done
ubuntu@ip-172-31-47-113:~/react-app-deploy-ec2-docker-jenkins$
```

```
ubuntu@ip-172-31-47-113:~/react-app-deploy-ec2-docker-jenkins$ docker-compose ps
     Name          Command         State           Ports
jenkins   /usr/bin/tini -- /usr/local/bin/docker-entrypoint.sh nginx ... Up      0.0.0.0:50000->50000/tcp,:::50000->50000/tcp, 0.0.0.0:8080->8080/tcp,:::8080->8080/tcp
react-app  /docker-entrypoint.sh nginx ... Up      0.0.0.0:80->80/tcp,:::80->80/tcp
ubuntu@ip-172-31-47-113:~/react-app-deploy-ec2-docker-jenkins$
```

```
ubuntu@ip-172-31-47-113:~/react-app-deploy-ec2-docker-jenkins$ export $DOCKER_USER=sreedocker911
-bash: export: `=sreedocker911': not a valid identifier
ubuntu@ip-172-31-47-113:~/react-app-deploy-ec2-docker-jenkins$ export DOCKER_USER=sreedocker911
ubuntu@ip-172-31-47-113:~/react-app-deploy-ec2-docker-jenkins$ echo $DOCKER_USER
sreedocker911
ubuntu@ip-172-31-47-113:~/react-app-deploy-ec2-docker-jenkins$ export DOCKER_PASS=dckr_pat_Lbp-vsxxGusI297WooISIuoao7I
ubuntu@ip-172-31-47-113:~/react-app-deploy-ec2-docker-jenkins$ echo $DOCKER_PASS
dckr_pat_Lbp-vsxxGusI297WooISIuoao7I
ubuntu@ip-172-31-47-113:~/react-app-deploy-ec2-docker-jenkins$
```

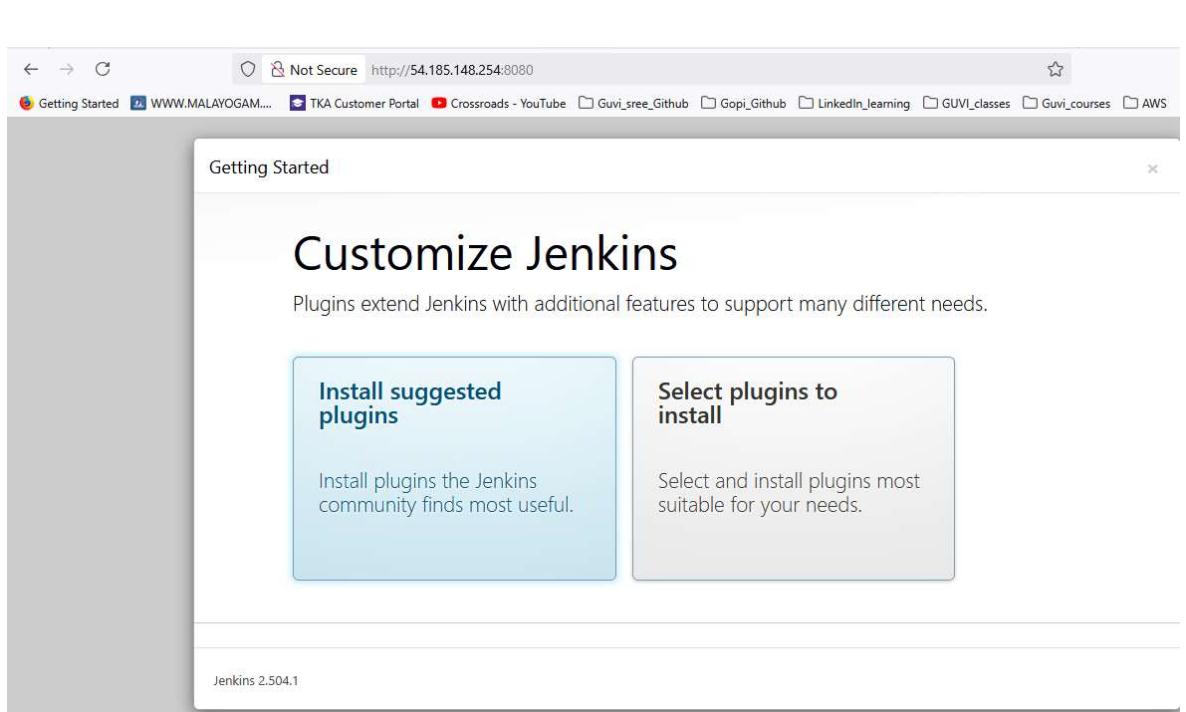
Jenkins: <http://ec2-35-90-101-84.us-west-2.compute.amazonaws.com:8080>



sudo docker exec jenkins cat /var/jenkins_home/secrets/initialAdminPassword

```
ubuntu@ip-172-31-47-113:~/react-app-deploy-ec2-docker-jenkins$ sudo docker exec jenkins cat /var/jenkins_home/secrets/initialAdminPassword
d2e5b33f32e24e61833f6cd2042df12c
ubuntu@ip-172-31-47-113:~/react-app-deploy-ec2-docker-jenkins$
```

d2e5b33f32e24e61833f6cd2042df12c



→ Then click the install suggested plugins options:

The screenshot shows the Jenkins 'Getting Started' page with a context menu open over the 'Build Timeout' plugin entry in the list. The menu is titled 'Build Timeout' and includes options like 'Install Now', 'Uninstall', 'Disable', 'Enable', 'Configure', 'API Documentation', and 'GitHub'. The menu also indicates that 'Build Timeout' is a required dependency. The table below lists various Jenkins plugins. At the bottom left, it says 'Jenkins 2.504.1'.

✓ Folders	✓ OWASP Markup Formatter	✓ Build Timeout	✗ Credentials Binding
Timestamper	Workspace Cleanup	Ant	Gradle
Pipeline	GitHub Branch Source	Pipeline: GitHub Groovy Libraries	Pipeline Graph View
Git	SSH Build Agents	Matrix Authorization Strategy	PAM Authentication Strategy
LDAP	Email Extension	Mailer	Dark Theme

Getting Started

Create First Admin User

Username

Confirm password

Full name

Jenkins 2.504.1

Skip and continue as admin

Save and Continue

admin / <password> then email ID

ID	Name	Kind	Description
sreejesht	sreejesht/******** (gitbut login for Jenkins)	Username with password	gitbut login for Jenkins
sreedocker911	sreedocker911/******** (docker hub from Jenkins)	Username with password	docker hub from Jenkins

Icon: S M L

New Item

Enter an item name

» This field cannot be empty, please enter a valid name

Select an item type



Freestyle project

Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.



Pipeline

Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.



Multi-configuration project

Suitable for projects that need a large number of different configurations, such as testing on multiple environments.

OK

Configure

General

Description: This OCD pipeline builds, pushes and deploys a Dockerized ReactApp using the code from the Dev Branch

Pipeline

Definition: Pipeline script from SCM

SCM: Git

Repositories: Repository URL: https://github.com/reejesh/react-app-deploy-ec2-docker-jenkins.git
Credentials: sreejesh.t@gmail.com/***** (GIT login for Jenkins)

Advanced

Branch Specifier (blank for 'any'): */dev

Script Path: Jenkinsfile

Lightweight checkout:

Save **Apply**

apply & save

Jenkins

Dashboard >

S	W	Name	Last Success	Last Failure	Last Duration
4	22 min	REACT_APP_CICD	4 hr 22 min #5	4 hr 44 min #3	7.8 sec

Build Queue: No builds in the queue.

Build Executor Status: 0/2

The screenshot shows the Jenkins dashboard for a job named 'REACT_APP_CICD'. Build #6 was triggered by a GitHub push from 'sreejesh' on May 30, 2025, at 10:45:21 PM. The build took 8.2 seconds and completed 15 seconds ago. The build log indicates it started with a GitHub push and used a Docker container named 'react-app'.

GitHub Webhook Configuration:

- Webhook URL:** http://ec2-35-90-101-84.us-west-2.compute.amazonaws.com:8080/github-webhook/
- Content type:** application/json
- Secret:** [redacted]
- SSL verification:** Only verify SSL certificates when delivering payloads (checked)
- Which events would you like to trigger this webhook?**
 - Just the pull event (checked)
 - Send me everything
 - Let me select individual events
- At most:** 100 (The total number of events which will be triggered)

Build now: http://ec2-35-90-101-84.us-west-2.compute.amazonaws.com:8080/view/all/job/REACT_APP_CICD/lastBuild/
React App : <http://ec2-35-90-101-84.us-west-2.compute.amazonaws.com/>

5. Monitoring with Prometheus + Grafana

5.1 Create Docker Network

docker network create devops-net

5.2 Prometheus Config

docker-compose -f docker-compose.monitoring.yml up -d –build /etc/prometheus/prometheus.yml

global:

scrape_interval: 15s

scrape_configs:

- job_name: 'react-app'

static_configs:

- targets: ['localhost:80']

```
ubuntu@ip-172-31-47-113:~/react-app-deploy-ec2-docker-jenkins$ vim prometheus.yml
ubuntu@ip-172-31-47-113:~/react-app-deploy-ec2-docker-jenkins$ ls
Jenkinsfile  deploy.sh  docker-compose.monitoring.yml  install_and_configure_git.sh  prometheus.yml
build.sh    devops-build  docker-compose.yml          install_pkgs.sh           push-docker-images.sh
ubuntu@ip-172-31-47-113:~/react-app-deploy-ec2-docker-jenkins$ vim docker-compose.yml
ubuntu@ip-172-31-47-113:~/react-app-deploy-ec2-docker-jenkins$ vim docker-compose.monitoring.yml
ubuntu@ip-172-31-47-113:~/react-app-deploy-ec2-docker-jenkins$ docker-compose -f docker-compose.monitoring.yml up -d
Creating network "react-app-deploy-ec2-docker-jenkins_default" with the default driver
WARNING: Found experimental feature(s) in your compose file. If you are using Docker 1.12 or higher, you can enable experimental features by setting the experimental.fielder option to true in your daemon configuration. See https://docs.docker.com/docker-for-mac/experimental/
Removing orphaned services (jenkins) for this project. If you removed or renamed this service in your compose file, you can run this command with the --remove-orphan flag to clean it up.
Pulling prometheus (prom/prometheus)...
latest: Pulling from prom/prometheus
9t59226be034: Pull complete
1617e25568b2: Pull complete
01e28ff2233c: Pull complete
0e15545833: Extracting: =====> [ 18.38MB/56.5MB
303f67c648a5: Download complete
d2597eacb5f4: Download complete
744c9ca72725: Download complete
e52136ad7a1b: Download complete
44750a2Seb21: Download complete
b6b811691043: Download complete
```

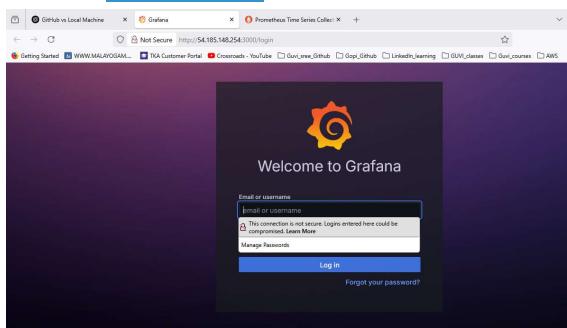
5.3 Run Prometheus and Grafana

- ✓ docker run -d --name prometheus --net monitor-net -p 9090:9090 \ -v /etc/prometheus/prometheus.yml:/etc/prometheus/prometheus.yml prom/Prometheus
- ✓ docker run -d --name=grafana --net monitor-net -p 3000:3000 grafana/Grafana
or
- ✓ docker-compose -f docker-compose.monitoring.yml up -d --build

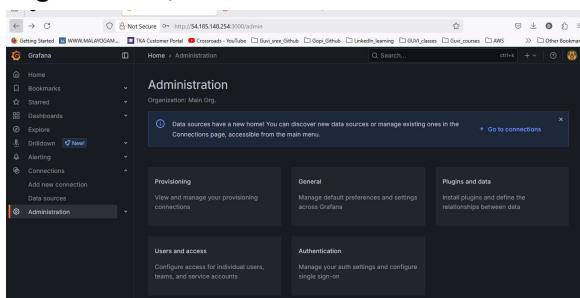
```
ubuntu@ip-172-31-47-113:~/react-app-deploy-ec2-docker-jenkins$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS
9ce1354649ff        react-static-app   "/docker-entrypoint...."   41 minutes ago     Up 41 minutes      0.0.0.0:80->80/tcp, :::80->80/tcp
6136a5358d13        prom/node-exporter   "/bin/node_exporter"    2 days ago        Up 2 days         0.0.0.0:9100->9100/tcp, :::9100->9100/tcp
d3fd3960d69a        react-app-deploy-ec2-docker-jenkins-jenkins   "/usr/bin/tini -- /u..."   2 days ago        Up 2 days         0.0.0.0:8080->8080/tcp, :::8080->8080/tcp, 0.0.0.0:50000->50000/tcp, :::50000->50000/tcp
fb4fbcb8501a8       prom/prometheus      "/bin/prometheus --c..."   2 days ago        Up 2 days         0.0.0.0:9090->9090/tcp, :::9090->9090/tcp
0f8a1cfec4f        prom/blackbox-exporter   "/bin/blackbox_expor..."   2 days ago        Up 2 days         0.0.0.0:9115->9115/tcp, :::9115->9115/tcp
9aac2a2868d9        grafana/grafana      "/run.sh"           2 days ago        Up 2 days         0.0.0.0:3000->3000/tcp, :::3000->3000/tcp
ubuntu@ip-172-31-47-113:~/react-app-deploy-ec2-docker-jenkins$
```

5.4 Configure Grafana

- ✓ Access: <http://:3000>



- ✓ Login: admin/admin



- ✓ Add Prometheus as data source: <http://prometheus:9090>

- ✓ Create dashboard for app status

Note:-

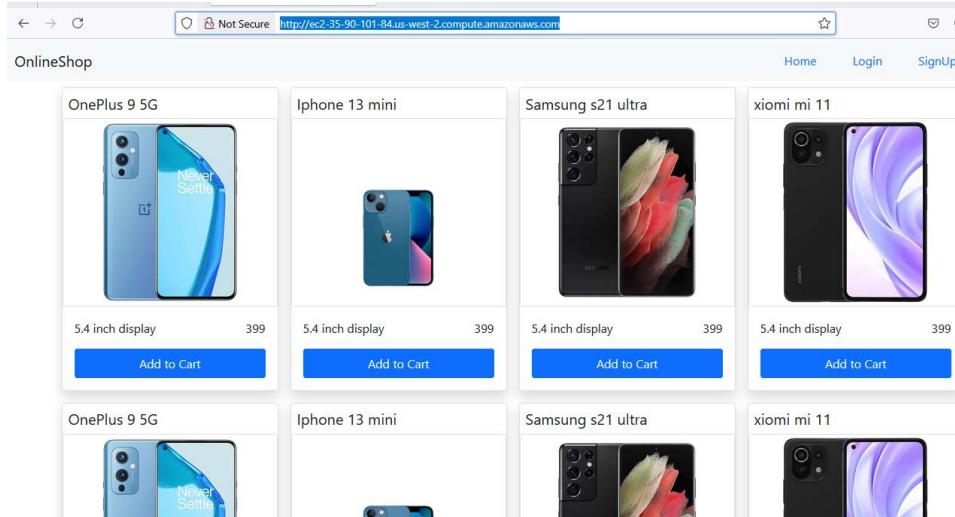
`docker-compose -f docker-compose.base.yml -f docker-compose.monitoring.yml up -d --build`

Use Multiple Compose Files (Advanced)

- ✓ You can layer multiple Compose files:
- ✓ `docker-compose.base.yml` → for app services
- ✓ `docker-compose.monitoring.yml` → for monitoring stack
- ✓ Then bring everything up like this:

6.0 Brief, Auto trigger Jenkins & Monitoring

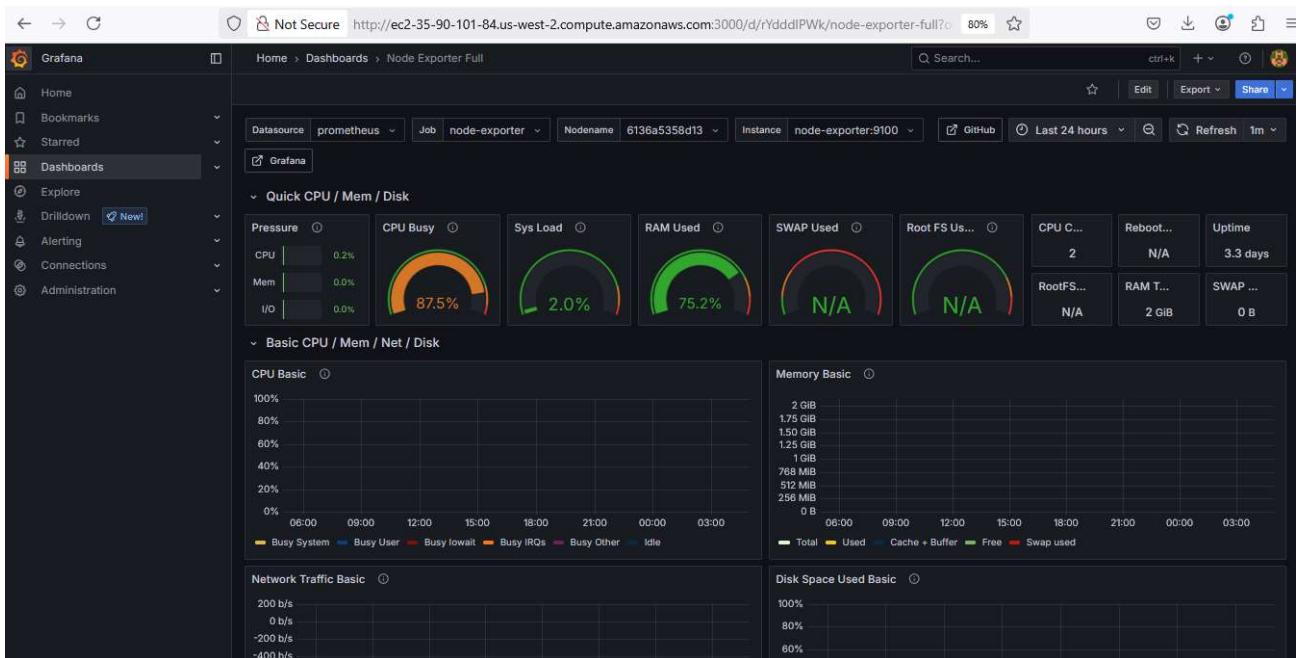
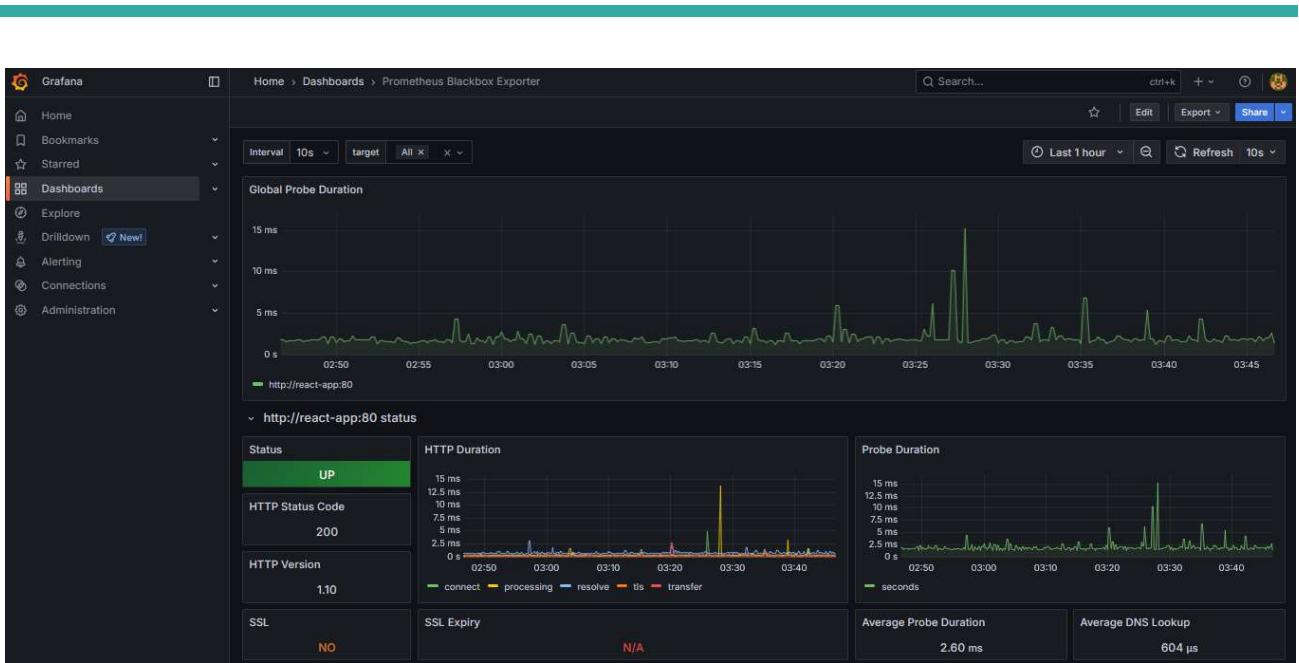
REACT APP : <http://ec2-35-90-101-84.us-west-2.compute.amazonaws.com/>



JENKINS :http://ec2-35-90-101-84.us-west-2.compute.amazonaws.com:8080/job/REACT_APP_CICD/lastBuild/console

Grafana dashboard:

[http://ec2-35-90-101-84.us-west-2.compute.amazonaws.com:3000/d/xtkCtBkiz/prometheus-blackbox-exporter?var-interval=10s&orgId=1&from=now-1h&to=now&timezone=browser&var-target=\\$_all&refresh=10s](http://ec2-35-90-101-84.us-west-2.compute.amazonaws.com:3000/d/xtkCtBkiz/prometheus-blackbox-exporter?var-interval=10s&orgId=1&from=now-1h&to=now&timezone=browser&var-target=$_all&refresh=10s)



Prometheus

<http://ec2-35-90-101-84.us-west-2.compute.amazonaws.com:9090/targets>

The screenshot shows the Prometheus interface with the URL <http://ec2-35-90-101-84.us-west-2.compute.amazonaws.com:9090/targets>. The page displays three target健康 (Health) sections:

- node-exporter**: Last scrape 8.536s ago, 1 / 1 up.
- prometheus**: Last scrape 12.097s ago, 1 / 1 up.
- react-app-health**: Last scrape 9.424s ago, 1 / 1 up.

Each section lists endpoints and their labels, such as `http://node-exporter:9100/metrics` with `instance="node-exporter:9100"` and `job="node-exporter"`.

6.1 Auto Trigger Jenkins on GitHub Push to dev Branch

Go to your Jenkins server → Open the job you created (e.g., `react-app-dev-build`). Click Configure. --- > Check this option: GitHub hook trigger for GITScm polling

Triggers

Set up automated actions that start your build based on specific events, like code changes or scheduled times.

- Build after other projects are built ?
 - Build periodically ?
 - GitHub hook trigger for GITScm polling ?
 - Poll SCM ?
- Schedule ?

6.2 Add GitHub Webhook

Go to your GitHub repo:

<https://github.com/sreejesht/react-app-deploy-ec2-docker-jenkins.git>

Click on Settings → Webhooks → Click Add webhook

Fill in the Webhook form:

- Payload URL:
`http://<your-jenkins-ip>:8080/github-webhook/`
(Replace `<your-jenkins-ip>` with your actual IP or domain; use https if applicable)
- Content type:
`application/json`
- Secret: (Optional)
Can be left blank unless you want to secure the webhook
- Which events would you like to trigger this webhook?
Select: Just the push event
Click Add Webhook

7.0 Enable and Test Alerting System

Ensure All Config Files Are in Place

- ✓ prometheus.yml (includes alerting config)
- ✓ alertmanager.yml (Gmail credentials and receiver)
- ✓ alerting-rules.yml (rules for alerts like high CPU load)
- ✓ Start Prometheus, Node Exporter, and Alertmanager

```
docker-compose -f docker-compose.monitoring.yml up -d  
docker ps
```

```
ubuntu@ip-172-31-47-113:~/react-app-deploy-ec2-docker-jenkins$ docker-compose -f docker-compose.monitoring.yml up -d  
WARN[0000] /home/ubuntu/react-app-deploy-ec2-docker-jenkins/docker-compose.monitoring.yml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion  
WARN[0000] Found orphan containers ([node-exporter jenkins]) for this project. If you removed or renamed this service in your compose file, you can run this command with the --remove-orphan flag to clean it up.  
[*] Running 3/3  
  ✓ Container grafana    Running  
  ✓ Container blackbox   Running  
  ✓ Container prometheus Running  
ubuntu@ip-172-31-47-113:~/react-app-deploy-ec2-docker-jenkins$ docker ps  
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS  
 NAMES  
9cc135649ff        react-static-app      "/docker-entrypoint..."  2 hours ago       Up 2 hours          0.0.0.0:80->80/tcp, :::80->80/tcp  
6136a5358d13       prom/node-exporter     "/bin/node_exporter"  2 days ago        Up 2 days          0.0.0.0:9100->9100/tcp, :::9100->9100/tcp  
d3fd3960a69a       react-app-deploy-ec2-docker-jenkins-jenkins  "/usr/bin/tini -- /u..."  2 days ago        Up 2 days          0.0.0.0:8080->8080/tcp, :::8080->8080/tcp, 0.0.0.0:5000  
0->50000/tcp, :::50000->50000/tcp  jenkins  
fb4fbc8501a8       prom/prometheus      "/bin/prometheus --c..."  3 days ago        Up 2 days          0.0.0.0:9090->9090/tcp, :::9090->9090/tcp  
0ff8alcfecc4f     prom/blackbox-exporter  "/bin/blackbox_expor..."  3 days ago        Up 3 days          0.0.0.0:9115->9115/tcp, :::9115->9115/tcp  
9aac2a2868d9       grafana/grafana     "/run.sh"           3 days ago        Up 3 days          0.0.0.0:3000->3000/tcp, :::3000->3000/tcp  
ubuntu@ip-172-31-47-113:~/react-app-deploy-ec2-docker-jenkins$
```

```
ubuntu@ip-172-31-47-113:~/react-app-deploy-ec2-docker-jenkins$
```

```
groups:  
  - name: example  
    rules:  
      - alert: HighLoad  
        expr: node_load1 > 0.85  
        for: 5m  
        labels:  
          severity: critical  
        annotations:  
          summary: High load on '{{ $labels.instance }}'  
          description: '{{ $labels.instance }} has had a load higher than 0.85 for more than 5 minutes.  
~  
~  
~  
~  
~
```

```
global:  
  smtp_smarthost: 'smtp.gmail.com:465'  
  smtp_from: 'sreejesh.t@gmail.com'  
  smtp_auth_username: 'sreejesh.t@gmail.com'  
  smtp_auth_password: '████████████████████████████████'  
  
route:  
  receiver: 'email_notifications'  
  
receivers:  
  - name: 'email_notifications'  
    email_configs:  
      - to: 'sreejesh.t@gmail.com'  
        send_resolved: true  
~
```

```

ubuntu@ip-172-31-47-113:~/react-app-deploy-ec2-docker-jenkins$ tree
.
├── Dockerfile.jenkins
├── Jenkinsfile
├── Jenkinsfile.old
├── Jenkinsfile_dev_prod
├── alerting-rules.yml
├── alertmanager.yml
├── build.sh
├── deploy.sh
├── deploy.sh.old
└── devops-build
    ├── Dockerfile
    └── build
        ├── _redirects
        ├── asset-manifest.json
        ├── favicon.ico
        ├── index.html
        ├── logo192.png
        ├── logo512.png
        ├── manifest.json
        └── robots.txt
        └── static
            ├── css
            │   ├── main.cf5c13c5.css
            │   └── main.cf5c13c5.css.map
            └── js
                ├── 787.2f5360e2.chunk.js
                ├── 787.2f5360e2.chunk.js.map
                ├── main.flc48542.js
                ├── main.flc48542.js.LICENSE.txt
                └── main.flc48542.js.map
├── docker-compose.monitoring.yml
├── docker-compose.yml
├── ec2_rescue.sh
├── install_and_configure_git.sh
├── install_pkgs.sh
├── prometheus.yml
├── push-docker-images.sh
└── setup_jenkins_docker.sh

• 6 directories, 33 files

```

Final Submission Check

Checkpoint	Status
GitHub repo with main/dev	✓
Jenkinsfile with multibranch	✓
Docker Hub: Dev/Prod repos	✓
Scripts: build & deploy	✓
Monitoring setup	✓
Directory structure	✓

THANKS!!!!