

## 1.PROJECT OVERVIEW:

The project implements a complete **CI/CD pipeline** using **Github** -> **Jenkins** -> **Docker** -> **AWS EC2** for automated deployment of a Node.js Portfolio application.

## 2. Tools & Technologies Used

- AWS EC2 (Amazon Linux 2023)
- Jenkins for CI/CD automation
- GitHub for source code management
- Docker for containerization
- Node.js & npm for runtime
- GitHub Webhook for auto-triggering builds

## 3. EC2 Instance Setup

1. Launched Amazon Linux 2023 EC2 instance.
2. Opened security group ports:  
22 (SSH), 8080 (Jenkins), 3000 (App), 80 (optional).

The screenshot shows the AWS EC2 Instance details page for an instance named 'jen1' with ID 'i-09d21747ceccc4833'. The instance is running on an Amazon Linux 2023 AMI and is assigned a Public IPv4 address of 18.234.153.217 and a Private IP address of 172.31.35.25. It is connected to a VPC with ID vpc-0610fd9fedfd36a33 and a subnet with ID subnet-064c2e2616cd0f429. The instance type is t2.medium. The public DNS name is ec2-18-234-153-217.compute-1.amazonaws.com. The instance was updated about 7 hours ago and is currently running.

EC2 Instances i-09d21747cecc4833

Security details

IAM Role: -

Owner ID: 253490795695

Launch time: Mon Nov 24 2025 09:49:31 GMT+0530 (India Standard Time)

Security groups: sg-0baacdc6f55ececde (jenkins\_security)

Inbound rules:

Name	Security group rule ID	Port range	Protocol	Source
-	sgr-079bf3602103b1b2c	80	TCP	0.0.0.0/0
-	sgr-0445a1a78a650d207	22	TCP	0.0.0.0/0
-	sgr-08f36d0fc4055bc77	3000	TCP	0.0.0.0/0
-	sgr-0a50698847c5635b6	8080	TCP	0.0.0.0/0

### 3. Updated system packages:

```
sudo dnf update -y  
sudo yum update -y
```

## 4. Installed Jenkins

1. Added Jenkins repository & GPG key.
2. Installed Jenkins(also java for Jenkins):  
sudo dnf install jenkins -y

### 3. Started service:

```
sudo systemctl start jenkins  
sudo systemctl enable Jenkins
```

```
sudo dnf install java-17-amazon-corretto -y  
sudo wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-stable/jenkins.repo  
sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io.key  
sudo dnf install jenkins -y  
sudo systemctl enable jenkins
```

### 4. Access Jenkins at:

<http://<EC2-IP>:8080>

Jenkins

+ New Item

+ Build History

All +

Add description

## 5. Installed Docker

### 1. Installed Docker on EC2:

```
sudo dnf install docker -y
```

### 2. Enabled and started Docker:

```
sudo systemctl start docker
```

```
sudo systemctl enable docker
```

### 3. Added Jenkins to Docker group:

```
sudo usermod -aG docker Jenkins
```

```
sudo dnf install docker -y
sudo systemctl enable docker
sudo systemctl start docker
sudo usermod -aG docker ec2-user
sudo usermod -aG docker jenkins
```

## 6. Installed Node.js

### sudo dnf install nodejs -y

```
[ec2-user@ip-172-31-35-25 ~]$ sudo dnf install nodejs -y
Last metadata expiration check: 2:59:27 ago on Mon Nov 24 04:29:35 2025.
Dependencies resolved.
```

Package	Architecture	Version	Repository
nodejs	x86_64	1:18.20.8-1.amzn2023.0.2	amazonlinux

```
[ec2-user@ip-172-31-35-25 ~]$ node -v
v18.20.8
[ec2-user@ip-172-31-35-25 ~]$ npm -v
10.8.2
```

## 7. Jenkins Plugin Installation

### Installed required plugins:

Git

GitHub

Pipeline

Docker Pipeline

Credentials Binding

The screenshot shows the Jenkins plugin manager interface. On the left, there's a sidebar with options: Updates, Available plugins, **Installed plugins** (which is selected), and Advanced settings. The main area has a search bar at the top labeled "Search installed plugins". Below the search bar is a table with columns for Name, Health, and Enabled. The table lists several plugins:

Name	Health	Enabled
Ant Plugin	96	<input checked="" type="checkbox"/>
Apache HttpComponents Client 4x API Plugin	96	<input checked="" type="checkbox"/>
Apache HttpComponents Client 5x API Plugin	100	<input checked="" type="checkbox"/>
ASM API Plugin	100	<input checked="" type="checkbox"/>
Authentication Tokens API Plugin	100	<input checked="" type="checkbox"/>
Bootstrap 5 API Plugin	96	<input checked="" type="checkbox"/>

## 8.Uploading my-portfolio project on GitHub

### 1.Uploaded files

The screenshot shows the GitHub repository 'my-portfolio' with a red box highlighting the repository name. The commit history lists several changes made by 'uddhavgun'.

File / Commit Message	Author	Date	Commit ID
public	uddhavgun	4 days ago	eeb84b9
views	uddhavgun	4 days ago	
Dockerfile	uddhavgun	4 hours ago	
Jenkinsfile	uddhavgun	4 days ago	
README.md	uddhavgun	3 hours ago	
index.js	uddhavgun	4 days ago	
package-lock.json	uddhavgun	4 days ago	
package.json	uddhavgun	4 days ago	

### 2.Dockerfile fix

The screenshot shows the GitHub repository 'my-portfolio' with a red box highlighting the 'Dockerfile' tab. The code editor displays the Dockerfile content.

```
FROM node:18
WORKDIR /app
COPY package*.json .
RUN npm install
COPY .
EXPOSE 3000
CMD ["node", "index.js"]
```

### 3.Jenkinsfile fix

The screenshot shows a GitHub repository page for 'my-portfolio'. The 'Jenkinsfile' file is open, displaying the following Jenkins pipeline script:

```
pipeline {
    agent any

    stages {
        stage('Clone') {
            steps {
                checkout scm
            }
        }
        stage('Install Dependencies') {
            steps {
                sh ...
                npm install
                ...
            }
        }
        stage('Deploy to EC2') {
            steps {
                sh ...
                sudo rm -rf /var/www/html/*
                cp -r * /var/www/portfolio/
                ...
            }
        }
    }
}
```

## 9. GitHub Credentials Setup

### 1. Created a Personal Access Token on GitHub.

The screenshot shows the GitHub 'Developer Settings' page under 'Personal access tokens (classic)'. A token named 'jenkinsmy' has been generated, with the details shown in a red box:

jenkinsmy — admin:repo_hook:repo:workflow	Never used	<a href="#">Delete</a>
Expires on Sat, Dec 20 2025.		

A note at the bottom explains: "Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#)."

### 2. Added it into Jenkins:

Manage Jenkins → Credentials → Global → Add Credentials

The screenshot shows the Jenkins 'Global credentials (unrestricted)' page. A single credential entry is listed:

ID	Name	Kind	Description
221615c7-ce40-47fd-9564-1e5f3d6071f9	uddhavgund/*****	Username with password	

At the bottom, there are icons for S, M, and L.

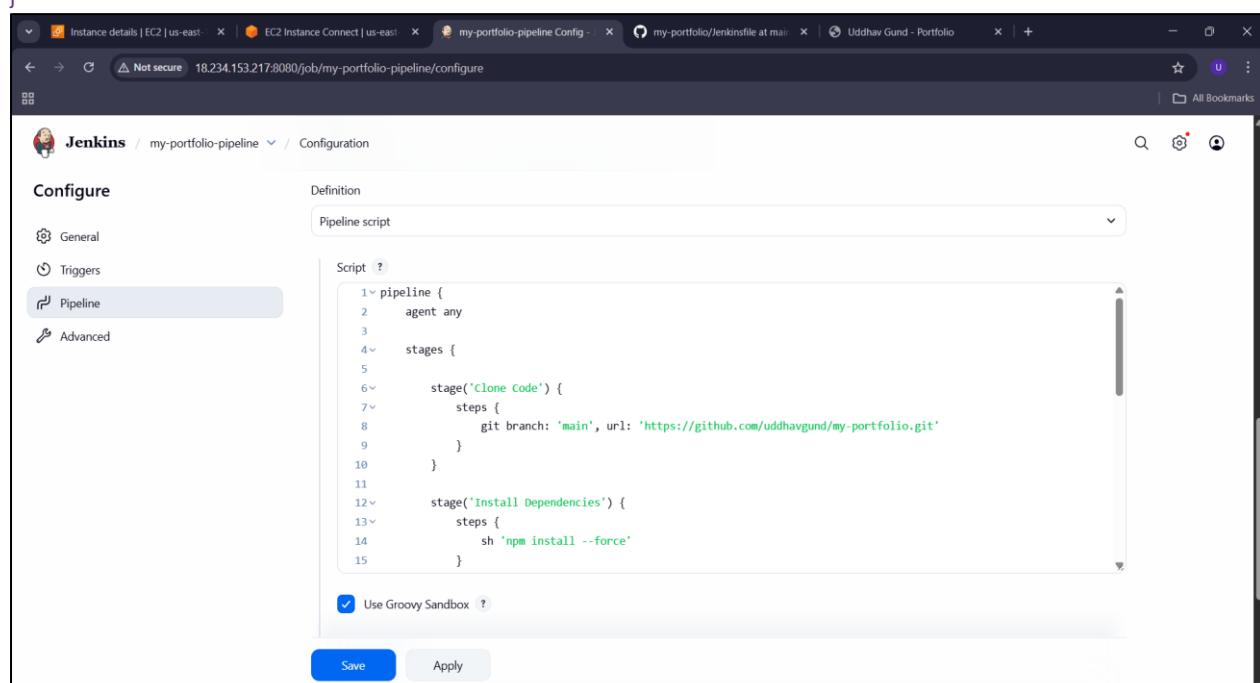
## 10. Jenkins Pipeline Job Setup

1. Created new Jenkins job: **my-portfolio-pipeline**

2. Selected Pipeline project type

3. Added pipeline script:

```
pipeline {  
    agent any  
    stages {  
        stage('Clone Code') {  
            steps {  
                git branch: 'main', url: 'https://github.com/uddhavgund/my-portfolio.git'  
            }  
        }  
        stage('Install Dependencies') {  
            steps {  
                sh 'npm install --force'  
            }  
        }  
        stage('Docker Build') {  
            steps {  
                sh 'docker build -t my-portfolio .'  
            }  
        }  
        stage('Stop Old Container') {  
            steps {  
                sh ""  
                docker ps -q --filter "name=my-portfolio" | grep -q . && docker stop my-portfolio || true  
                docker rm my-portfolio || true  
                ...  
            }  
        }  
        stage('Run New Container') {  
            steps {  
                sh 'docker run -d -p 3000:3000 --name my-portfolio my-portfolio'  
            }  
        }  
    }  
}
```



The screenshot shows the Jenkins dashboard with a single build item listed:

S	W	Name ↓	Last Success	Last Failure	Last Duration
		my-portfolio-pipeline	57 min #6	2 hr 0 min #4	18 sec

Build Queue: No builds in the queue.

Build Executor Status: 0/2

Icon: S M L

## 11. GitHub Webhook Setup (Auto Deployment)

1. GitHub → Repo → Settings → Webhooks → Add Webhook

2. Payload URL:

<http://<EC2-IP>:8080/github-webhook/>

3. Event: Just the push event

The screenshot shows the GitHub repository settings page for 'uddhavgund - my-portfolio'. The 'Webhooks' section is open, showing the configuration for a new webhook:

- Payload URL:** http://18.234.153.217:8080/github-webhook/
- Content type:** application/json
- Secret:** (empty field)
- SSL verification:** Enable SSL verification (radio button selected)
- Which events would you like to trigger this webhook?**
  - Just the push event. (radio button selected)
  - Send me everything.
  - Let me select individual events.
- Active:** (checkbox checked)

At the bottom is a green 'Add webhook' button.

The screenshot shows the GitHub 'Webhooks' settings page for a repository named 'my-portfolio'. The left sidebar has 'Webhooks' selected under the 'Code and automation' section. A single webhook is listed with a green checkmark and the URL 'http://18.234.153.217:8080/github-webhook'. Below it, a message says 'Last delivery was successful.' There are 'Edit' and 'Delete' buttons.

#### 4. Jenkins auto-triggers on every Git push.

The screenshot shows the Jenkins 'Configuration' page for a pipeline named 'my-portfolio-pipeline'. Under the 'Triggers' tab, the 'GitHub hook trigger for GITScm polling' option is selected. The 'Pipeline' section shows a 'Definition' dropdown set to 'Pipeline script'. At the bottom are 'Save' and 'Apply' buttons.

## 12. Final Deployment

After successful build, application runs on:

<http://<EC2-IP>:3000>

The screenshot shows a web browser window with the URL [18.234.153.217:3000](http://18.234.153.217:3000). The page title is "Uddhav Gund - Portfolio". The main content features a profile picture of a man with dark hair and a beard, wearing a blue shirt and tie. To the right of the picture, the name "Uddhav Gund" is displayed in bold black font, followed by the heading "A Bit About Me". Below this, a short bio states: "I am a Jr. AWS & DevOps Developer with experience in Node.js, Docker, and AWS cloud deployments. I enjoy building portfolio projects and learning new cloud technologies." At the bottom of the page are four buttons: "Resume" (green), "Research" (orange), "Outreach" (teal), and "Personal" (grey).

## Successful Console Output

The screenshot shows a Jenkins pipeline console output for build #5. The left sidebar lists build-related actions: Status, Changes, Console Output (which is selected), Edit Build Information, Delete build '#5', Timings, Git Build Data, Pipeline Overview, Restart from Stage, Replay, Pipeline Steps, Workspaces, and Previous Build. The main area is titled "Console Output" and shows the following log entries:

```
Started by user uddhav
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/my-portfolio-pipeline
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Clone Code)
[Pipeline] git
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/my-portfolio-pipeline/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/uddhavgund/my-portfolio.git # timeout=10
Fetching upstream changes from https://github.com/uddhavgund/my-portfolio.git
> git --version # timeout=10
> git --version # 'git version 2.50.1'
> git fetch --tags --progress -- https://github.com/uddhavgund/my-portfolio.git +refs/heads/*:refs/remotes/origin/*
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
> git config core.sparsecheckout # timeout=10
> git checkout -f 9e688ee44aeb4a0a630b3f75b3a923219f46408f (refs/remotes/origin/main)
Checking out Revision 9e688ee44aeb4a0a630b3f75b3a923219f46408f (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f 9e688ee44aeb4a0a630b3f75b3a923219f46408f # timeout=10
> git branch -a -v --no-abbrev # timeout=10
> git branch -D main # timeout=10
> git checkout -b main 9e688ee44aeb4a0a630b3f75b3a923219f46408f # timeout=10
```

The screenshot shows the Jenkins dashboard with the 'my-portfolio-pipeline' job highlighted in the queue. The job status is green (S) and it has been last successful 57 minutes ago. It has a duration of 18 seconds.

## Auto-deployment of GitHub Push

The screenshot shows the Jenkins GitHub Hook Log for the 'my-portfolio-pipeline' job. It details the last GitHub push, which was triggered by a webhook from https://140.82.115.242 at 08:35:53 UTC on Nov 24, 2025. The log shows the command used to pull the latest changes and the resulting Docker image generation.

Docker images generated !

```
[ec2-user@ip-172-31-35-25 ~]$ docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
my-portfolio    latest   5ad711b3d40c  4 minutes ago  1.1GB
<none>          <none>   8d195c587983  About an hour ago  1.1GB
```

Docker Container Generated!

```
[ec2-user@ip-172-31-35-25 ~]$ docker ps -a
CONTAINER ID      IMAGE      COMMAND      CREATED      STATUS      PORTS      NAMES
830d9390bdc9    my-portfolio "docker-entrypoint.s..."  5 minutes ago  Up 5 minutes  0.0.0.0:3000->3000/tcp, :::3000->3000/tcp  my-portfolio
[ec2-user@ip-172-31-35-25 ~]$
```

Docker container automatically updates on every GitHub push.

!!!!!! Pipeline Completed Successfully !!!!!!