Deliverables

CI CD Jenkins pipeline deployment of a Django application to an EC2 instance using Docker and Nginx reverse proxy.

Date: 15 Dec 2025

Project summary

This document captures the key evidence and configuration used to build, push, and deploy the application using Jenkins, and to expose the running service through Nginx on port 80.

Live application endpoints

EC2 public DNS (reverse proxy, no port): http://ec2-54-234-81-108.compute-1.amazonaws.com

EC2 public IPv4 (reverse proxy, no port): http://54.234.81.108

Reverse proxy note: Nginx listens on port 80 and forwards traffic to http://127.0.0.1:8085 on the instance.

Contributors

1. Aisha Funmi Yusuf
2. Ayorinde Anna Ayomiposi
3. Oluwaseunfunmi Sowunmi
4. Amechi Susan Ogechi
5. Haruna Hope Michael
6. Akindele Abosede
7. Akachukwu Queenesther

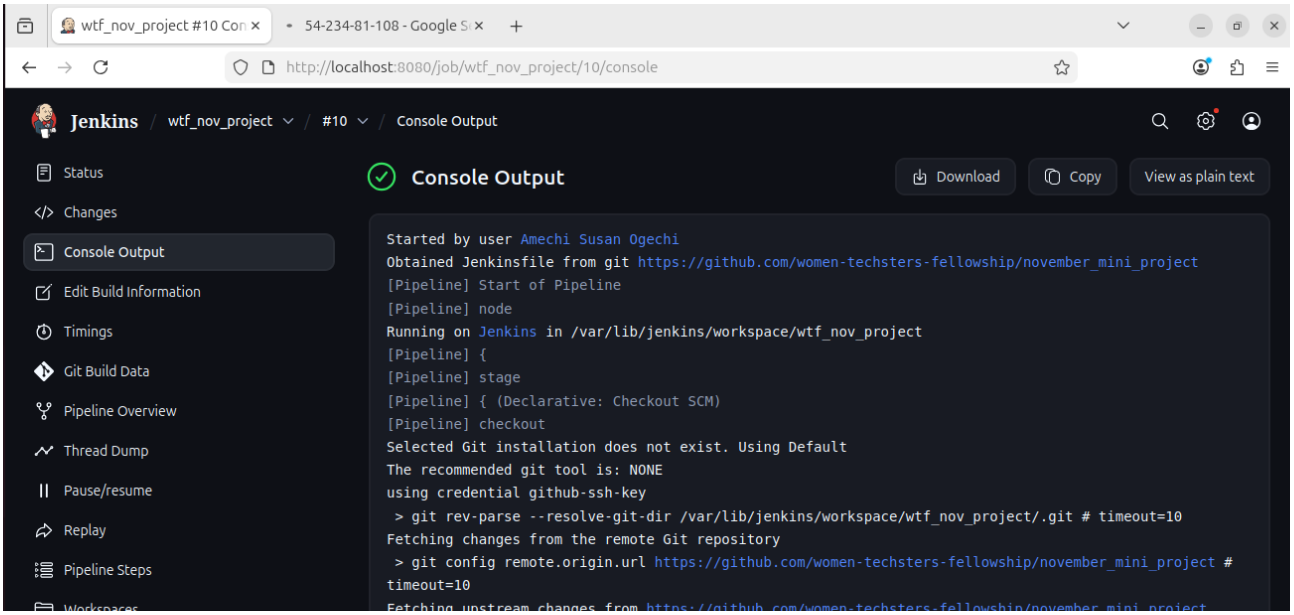
Jenkins pipeline configuration

The pipeline builds the Docker image, pushes it to Docker Hub, then deploys to EC2 over SSH. The EC2 host value is injected via Jenkins credentials (ID: ec2-host).

```
pipeline {
  agent any

  environment {
    DOCKERHUB_REPO = 'susan22283/wtf_nov_project'
    CONTAINER_NAME = 'wtf_nov_mini_project'
    IMAGE_TAG      = "build-${env.BUILD_NUMBER}"
  }

  stages {
    stage('Checkout') { steps { checkout scm } }

    stage('Build Docker image') {
      steps {
        sh '''
          docker build -t ${DOCKERHUB_REPO}:${IMAGE_TAG} .
        '''
      }
    }

    stage('Push image to Docker Hub') {
      steps {
        withCredentials([usernamePassword(
          credentialsId: 'dockerhub-creds',
          usernameVariable: 'DOCKER_USER',
          passwordVariable: 'DOCKER_PASS'
        )]) {
          sh '''
            echo "$DOCKER_PASS" | docker login -u "$DOCKER_USER" --password-stdin
            docker push ${DOCKERHUB_REPO}:${IMAGE_TAG}
            docker logout
          '''
        }
      }
    }

    stage('Deploy to EC2') {
      steps {
        withCredentials([string(credentialsId: 'ec2-host', variable: 'EC2_HOST')]) {
          sshagent(credentials: ['ec2-ssh-key']) {
            sh '''
              echo "Deploying to EC2..."
              ssh -o StrictHostKeyChecking=no ${EC2_HOST} "
                docker pull ${DOCKERHUB_REPO}:${IMAGE_TAG} &&
                docker rm -f ${CONTAINER_NAME} || true &&
                docker run -d --name ${CONTAINER_NAME} -p 8085:8000
  ${DOCKERHUB_REPO}:${IMAGE_TAG}
              "
            '''
          }
        }
      }
    }
  }
}
```

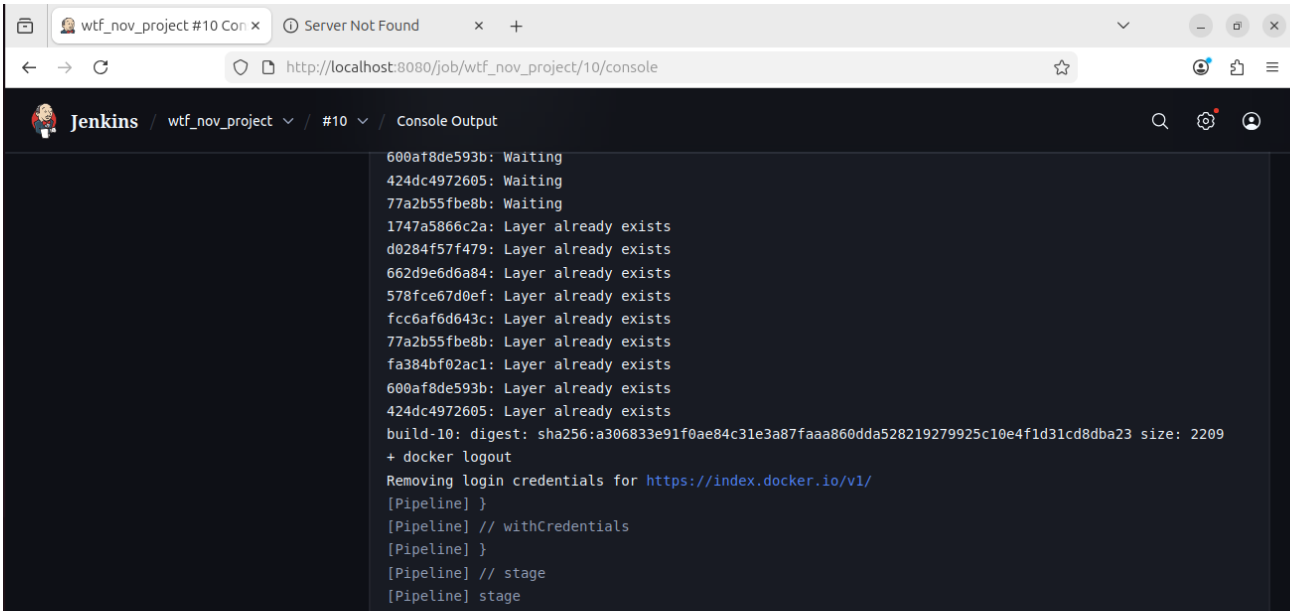
Evidence screenshots

Screenshots below provide evidence of successful pipeline execution and a live application endpoint.



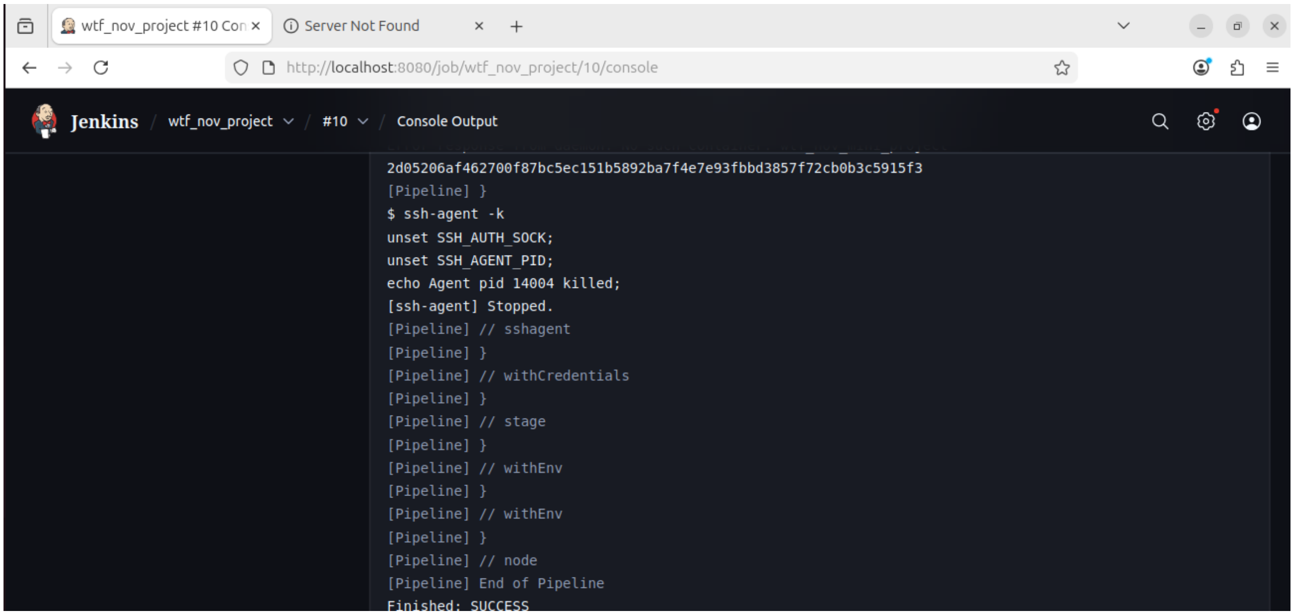
```
Started by user Amechi Susan Ogechi
Obtained Jenkinsfile from git https://github.com/women-techsters-fellowship/november_mini_project
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/wtf_nov_project
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
[Pipeline] checkout
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
using credential github-ssh-key
 > git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/wtf_nov_project/.git # timeout=10
Fetching changes from the remote Git repository
 > git config remote.origin.url https://github.com/women-techsters-fellowship/november_mini_project #
timeout=10
Fetching upstream changes from https://github.com/women-techsters-fellowship/november_mini_project
```

Figure 1. Jenkins console output showing pipeline start, checkout, and workspace context.



```
600af8de593b: Waiting
424dc4972605: Waiting
77a2b55fbe8b: Waiting
1747a5866c2a: Layer already exists
d0284f57f479: Layer already exists
662d9e6d6a84: Layer already exists
578fce67d0ef: Layer already exists
fcc6af6d643c: Layer already exists
77a2b55fbe8b: Layer already exists
fa384bf02ac1: Layer already exists
600af8de593b: Layer already exists
424dc4972605: Layer already exists
build-10: digest: sha256:a306833e91f0ae84c31e3a87faaa860dda528219279925c10e4f1d31cd8dba23 size: 2209
+ docker logout
Removing login credentials for https://index.docker.io/v1/
[Pipeline] }
[Pipeline] // withCredentials
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
```

Figure 2. Jenkins console output showing Docker image push to Docker Hub.



```
2d05206af462700f87bc5ec151b5892ba7f4e7e93fbbd3857f72cb0b3c5915f3
[Pipeline] }
$ ssh-agent -k
unset SSH_AUTH_SOCK;
unset SSH_AGENT_PID;
echo Agent pid 14004 killed;
[ssh-agent] Stopped.
[Pipeline] // sshagent
[Pipeline] }
[Pipeline] // withCredentials
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

Figure 3. Jenkins console output showing pipeline completion with SUCCESS status.

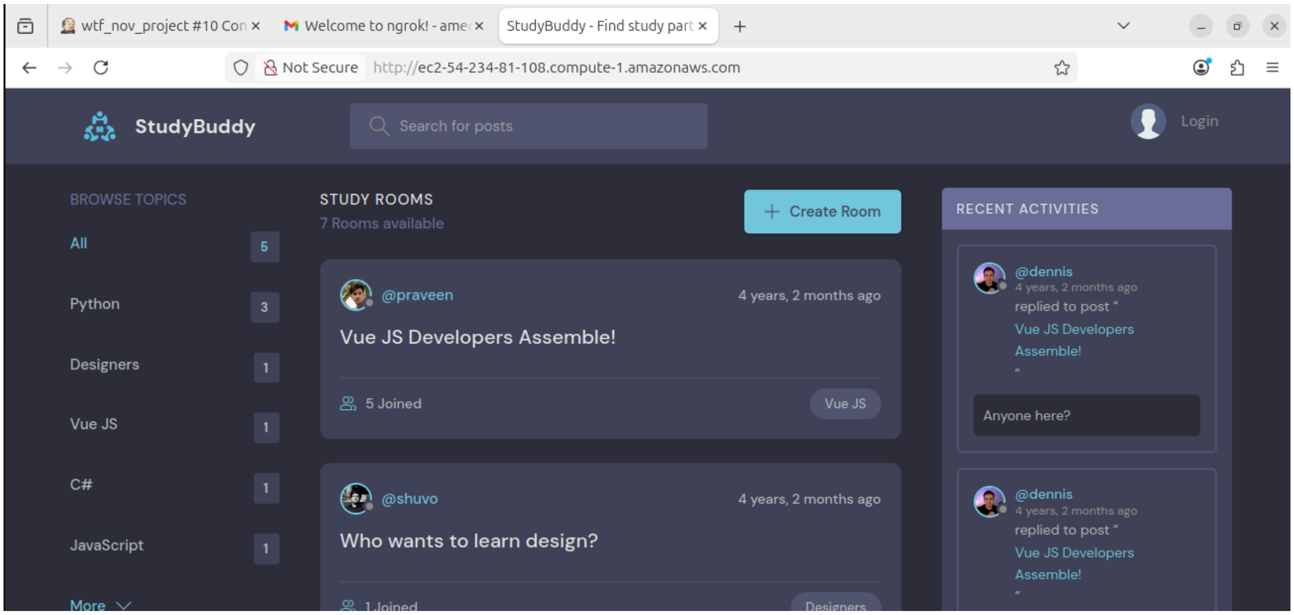


Figure 4. Application running live on EC2 through Nginx reverse proxy on port 80 using the EC2 public DNS.

Live application

URL (DNS, no port): http://ec2-54-234-81-108.compute-1.amazonaws.com

URL (IPv4, no port): http://54.234.81.108

Reverse proxy configuration was applied using an Nginx site file under /etc/nginx/sites-available and enabled via /etc/nginx/sites-enabled.