```
pipeline {
    agent any

    environment {
        // Define environment variables for Docker registry and image
        DOCKER_IMAGE = 'myapp_web'
        DOCKER_TAG = 'latest'
        DOCKER_REGISTRY = 'docker.io' // Replace with your registry if needed
    }

    stages {
        stage('Checkout Code') {
            steps {
                script {
                    // Checkout the source code from GitHub
                    git 'https://github.com/1986sabyasachee/CI_CD.git'
                }
            }
        }

        stage('Build Docker Image') {
            steps {
                script {
                    // Ensure Docker is available
                    sh 'docker --version'

                    // Build the Docker image
                    sh """
                        docker build -t ${DOCKER_REGISTRY}/${DOCKER_IMAGE}:${DOCKER_TAG} .
                    """
                }
            }
        }

        //stage('Push Docker Image') {
        //    steps {
        //        script {
                    // Log in to Docker registry (if needed)
                    // You can use Jenkins credentials here for authentication
        //            withCredentials([usernamePassword(credentialsId: 'dockerhub-credentials',
passwordVariable: 'DOCKER_PASSWORD', usernameVariable: 'DOCKER_USERNAME')]) {
        //                sh """
        //                    echo $DOCKER_PASSWORD | docker login -u $DOCKER_USERNAME --password-
stdin ${DOCKER_REGISTRY}
        //                    docker push ${DOCKER_REGISTRY}/${DOCKER_IMAGE}:${DOCKER_TAG}
        //                """
        //            }
        //        }
```

```
        //    }
        //}

        stage('Deploy Docker Container') {
            steps {
                script {
                    // Deploy the Docker container (stop and remove old container if necessary)
                    sh """
                        docker run -d --name ${DOCKER_IMAGE} -p 8091:80
${DOCKER_REGISTRY}/${DOCKER_IMAGE}:${DOCKER_TAG}
                    """
                }
            }
        }
    }

    post {
        always {
            // Clean up any resources or Docker containers if needed
            sh 'docker system prune -f'
        }
        success {
            echo 'Pipeline completed successfully!'
        }
        failure {
            echo 'Pipeline failed. Check logs for details.'
        }
    }
}
```