

Final Project

Dashboard > portfolio_krithick >

Status

Changes

Build Now

Configure

Delete Pipeline

Full Stage View

Stages

Rename

Pipeline Syntax

Builds

Filter

Today

#2 5:56 AM

portfolio_krithick

Add description

Stage View

Average stage times:
(full run time: ~4min 50s)

	Declarative: Tool Install	Clean Workspace	Git Checkout	Install Dependencies	Docker Build & Push	Deploy Docker Container	Declarative: Post Actions
05:56 1	112ms	533ms	28s	4s	4min 12s	1s	116ms

Permalinks

- Last build (#2), 15 min ago
- Last stable build (#2), 15 min ago
- Last successful build (#2), 15 min ago
- Last completed build (#2), 15 min ago

REST API Jenkins 2.492.2

Dashboard > portfolio_krithick > #2 > Pipeline Console

Build #2

Rebuild Overview Configure

Success 28 min ago in 4 min 50 sec

Tool Install

Clean Workspace

Git Checkout

Install Dependencies

Docker Build & Push

Deploy Docker Container

Post Actions

Stage 'Post Actions'

Started 23 min ago

Queued 0 ms

Took 98 ms

Success

Running on Jenkins

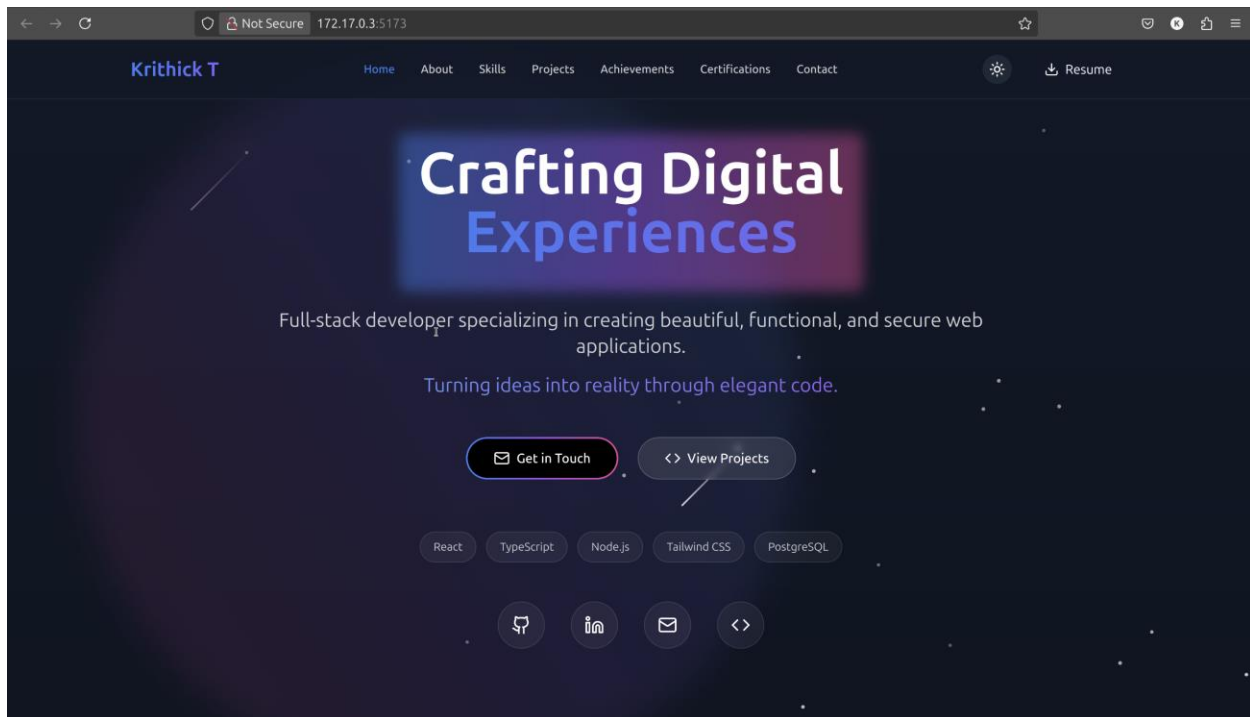
View as plain text

Pipeline executed successfully!

Print Message

0 Pipeline executed successfully!

Jenkins 2.492.2



```
krithick@krithick:~$ docker stop krithick_container
krithick_container
krithick@krithick:~$ docker run -it --rm -p 5173:5173 krithickthangaraj/krithick_repo:latest /bin/sh
# cd /app
npm run dev -- --host 0.0.0.0
#
> professional-portfolio@1.0.0 dev
> vite --host 0.0.0.0

Re-optimizing dependencies because vite config has changed

VITE v5.4.14 ready in 238 ms
  → Local:   http://localhost:5173/
  → Network: http://172.17.0.4:5173/
  → press h + enter to show help
```

```

krithick_container
[Pipeline] sh
+ docker rm krithick_container
krithick_container
[Pipeline] echo
Running new container...
[Pipeline] sh
+ docker run -d --name krithick_container -p 3001:3000 krithickthangaraj/krithick_repo:latest
6982fb76757408856e3520979264a17333e0e8a9d25c06b734d6bcd825108bd6
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Declarative: Post Actions)
[Pipeline] echo
Pipeline executed successfully! 🎉
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS

```

PIPELINE CODE:

```
pipeline { agent any tools { jdk 'jdk17' nodejs 'node20' }
```

```

environment {
    DOCKER_CREDENTIALS = 'docker' // Jenkins Docker credentials ID
    DOCKER_IMAGE = 'krithickthangaraj/krithick_repo'
    GIT_REPO = 'https://github.com/krithickthangaraj/portfolio_krithick.git'
    CONTAINER_NAME = 'krithick_container'
}

```

```

stages {
    stage('Clean Workspace') {
        steps {
            script {
                echo "Cleaning workspace..."
                deleteDir() // Ensures a fresh start before cloning
            }
        }
    }
}

```

```
stage('Git Checkout') {
  steps {
    script {
      git branch: 'main',
        credentialsId: 'github_seccred',
        url: "${GIT_REPO}"
    }
  }
}
```

```
stage('Install Dependencies') {
  steps {
    script {
      sh """
      if [ -f package.json ]; then
        echo "package.json found. Running npm install..."
        npm install
      else
        echo "ERROR: package.json is missing. Stopping pipeline."
        exit 1
      fi
      """
    }
  }
}
```

```
stage('Docker Build & Push') {
  steps {
    script {
      withDockerRegistry(credentialsId: "${DOCKER_CREDENTIALS}") {
        def tag = "latest"
      }
    }
  }
}
```

```

        sh "docker build -t ${DOCKER_IMAGE} ."
        sh "docker tag ${DOCKER_IMAGE} ${DOCKER_IMAGE}:${tag}"
        sh "docker push ${DOCKER_IMAGE}:${tag}"
    }
}
}

stage('Deploy Docker Container') {
    steps {
        script {
            echo "Stopping and removing any existing container..."
            sh "docker stop ${CONTAINER_NAME} || true"
            sh "docker rm ${CONTAINER_NAME} || true"

            echo "Running new container..."
            sh "docker run -d --name ${CONTAINER_NAME} -p 3001:3000
${DOCKER_IMAGE}:latest"
        }
    }
}

post {
    success {
        echo "Pipeline executed successfully! \U0001f680"
    }
    failure {
        echo "Pipeline failed! \u274c Check logs for issues."
    }
}

```

}