

Capstone Project Files

1. GitHub PR Link

<https://github.com/SatyamJoon/Book-My-Show/pull/1>

2. Jenkinsfile

```
pipeline {
    agent any

    tools {
        jdk 'Java17'
    }

    environment {
        SCANNER_HOME = tool 'sonar-scanner'
    }

    stages {
        stage('Clean Workspace') {
            steps {
                cleanWs()
            }
        }

        stage('Checkout from Git') {
            steps {
                git branch: 'master', url: 'https://github.com/SatyamJoon/Book-My-Show.git'
                sh '''ls -la'''
            }
        }

        stage('SonarQube Analysis') {
```

```

steps {
    withSonarQubeEnv('sonar') {
        sh """$SCANNER_HOME/bin/sonar-scanner \
        -Dsonar.projectKey=Book-my-show \
        -Dsonar.sources=. \
        -Dsonar.java.binaries=target/classes"""

    }
}

stage('Quality Gate') {
    steps {
        timeout(time: 5, unit: 'MINUTES') {
            script {
                def qg = waitForQualityGate()

                echo "SonarQube Quality Gate status: ${qg.status}"

                if (qg.status != 'OK') {
                    echo " Quality Gate failed, but continuing pipeline..."
                    // You can choose: send email, mark unstable, etc.
                    currentBuild.result = 'UNSTABLE'
                } else {
                    echo " Quality Gate passed!"
                }
            }
        }
    }
}

stage('Install Dependencies') {
    steps {
        sh ""

        cd bookmyshow-app

        ls -la # Verify package.json exists
    }
}

```

```

    if [ -f package.json ]; then
        rm -rf node_modules package-lock.json # Remove old dependencies
        npm i chokidar
        npm install # Install fresh dependencies
    else
        echo "Error: package.json not found in bookmyshow-app!"
        exit 1
    fi
    ""
}
}

stage('Docker Build & Push') {
    steps {
        script {
            sh ""
            echo "Building Docker image..."
            docker build --no-cache -t satyam744/bms:latest -f bookmyshow-app /Dockerfile bookmyshow-app

            echo "Logging into Docker Hub..."
            echo " dckr_pat_GNMsjbLHw7kavu_EvXgOqUo2huU" | docker login -u "satyam744" --password-stdin

            echo "Pushing Docker image to registry..."
            docker push satyam744/bms:latest

            echo "Logging out from Docker Hub..."
            docker logout
            ""
        }
    }
}

stage('Deploy to Container') {
    steps {
        sh ""
    }
}

```

```
echo "Stopping and removing old container..."
```

```
docker stop bms || true
```

```
docker rm bms || true
```

```
echo "Running new container on port 3000..."
```

```
docker run -d --restart=always --name bms -p 3010:3000 satyam744/bms:latest
```

```
echo "Checking running containers..."
```

```
docker ps -a
```

```
echo "Fetching logs..."
```

```
sleep 5 # Give time for the app to start
```

```
docker logs bms
```

```
'''
```

```
}
```

```
}
```

```
}
```

```
post {
```

```
  success {
```

```
    emailtext (
```

```
      subject: "SUCCESS: SonarQube Quality Gate Passed",
```

```
      body: ""
```

```
      Hi Team,
```

The SonarQube quality gate passed successfully.

You can view the report here:

<http://54.153.88.49:9000//dashboard?id=admin>

username: admin, passwd: Satyam123456@

If you want to download a PDF report, please use the SonarQube interface or a PDF plugin.

Regards,

Jenkins

""",

to: 'satyamchaudhary744@gmail.com'

)

}

failure {

emailtext (

subject: "FAILED: SonarQube Quality Gate",

body: ""

Hi Team,

The SonarQube quality gate has failed.

Please check the details:

<http://3.138.102.215/:9000/dashboard?id=Book-my-show>

username: admin, passwd: Satyam123456@

Regards,

Jenkins

""",

to: 'satyamchaudhary744@gmail.com'

)

}

}

}

3. [Docker-Hub Repository Link](https://hub.docker.com/repositories/satyam744)

<https://hub.docker.com/repositories/satyam744>

4. [Dockerfile](#)

Use Node.js 18 (or your Jenkins-configured version)

FROM node:18

Set working directory

WORKDIR /app

Copy package.json and package-lock.json

COPY package.json package-lock.json ./

Force install a compatible PostCSS version to fix the issue

RUN npm install postcss@8.4.21 postcss-safe-parser@6.0.0 --legacy-peer-deps

#adding chkidar here

RUN npm i chokidar

Install dependencies

RUN npm install

Copy the entire project

COPY . .

Expose port 3000

EXPOSE 3000

Set environment variable to prevent OpenSSL errors

ENV NODE_OPTIONS=--openssl-legacy-provider

ENV PORT=3000

Start the application

CMD ["npm", "start"]

5. [Docker-compose.yaml](#)

version: "3.9"

services:

bms-app:

image: satyam744/bms:latest # Use your Docker Hub image

container_name: bms-app

restart: unless-stopped

ports:

- "3010:3000" # Maps container port to host port for local access

networks:

- bms-net

networks:

bms-net:

driver: bridge

6. [Kubernetes manifest files – deployment.yaml and service.yaml.](#)

Deployment.yaml

apiVersion: apps/v1

kind: Deployment

metadata:

name: bms-app

namespace: satyam-ns

labels:

app: bms

spec:

replicas: 2

selector:

matchLabels:

app: bms

template:

metadata:

labels:

app: bms

spec:

containers:

- name: bms-container

image: satyam744/bms:latest # Replace with your Docker image

ports:

- containerPort: 3000 # Replace with the port your app runs on

Service.yaml

```
apiVersion: v1
kind: Service
metadata:
  name: bms-service
  namespace: satyam-ns
  labels:
    app: bms
spec:
  type: LoadBalancer
  ports:
    - port: 80
      targetPort: 3000 # Replace with the port your app runs on
  selector:
    app: bms
```