Dockerized Java Web Application Deployment Using Jenkins

Step1 . CI/CD pipeline implementation for deploying a **Java web application** using:

1. Git & GitHub
2. Maven
3. Jenkins
4. Docker
5. MySQL
6. Apache Tomcat
7. AWS EC2 (Linux)

Step2 . System Requirements

Software

- Linux (Ubuntu / Amazon Linux)
- Java 17
- Maven
- Docker
- Jenkins

AWS

- EC2 instance (t2.micro or above)
- Ports opened:
  - 8080 / 8081 (Application)
  - 3306 (MySQL)
  - 22 (SSH)

Step3 . Tools Used

Tool	**Purpose**
Git & GitHub	Source code management
Maven	Build & package WAR file
Jenkins	CI/CD automation
Docker	Containerization
MySQL	Database

Tool	**Purpose**
Tomcat	Application server
AWS EC2	Hosting server

Step 4 . **Docker Configuration**

Application Dockerfile (Docker-app/Dockerfile)

- Uses **Tomcat 9 + Java 17**
- Removes default webapps
- Copies WAR file as ROOT.war
- Exposes port 8080

Database Dockerfile (Docker-db/Dockerfile)

```
FROM mysql:5.7.25

ENV MYSQL_ROOT_PASSWORD=devopspassword
ENV MYSQL_DATABASE=accounts

ADD db_backup.sql docker-entrypoint-initdb.d/db_backup.sql
```

- MySQL 5.7
- Database auto-created
- SQL auto-restored on first run

Step 5. **Jenkins Pipeline Workflow**

Pipeline Stages Explanation

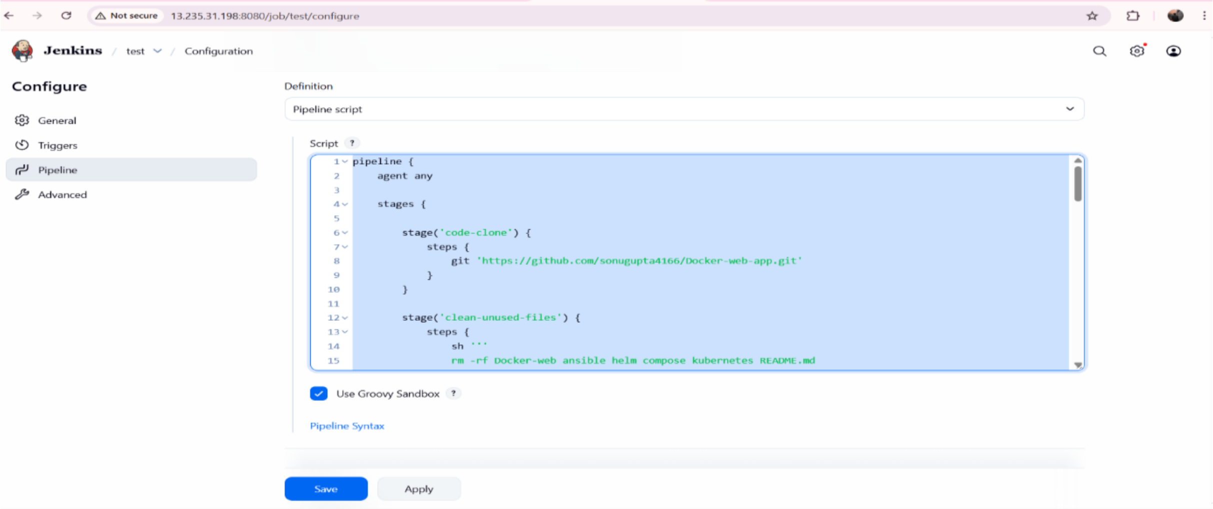
1. **Code Clone**
   Clones the GitHub repository.
2. **Remove Unused Files**
   Deletes unnecessary folders to reduce build context size.
3. **Build Application**
   Maven builds the WAR file inside target/.
4. **Copy WAR to Docker Context**
   Copies target/*.war into Docker-app/target.
5. **Build Docker Images**
   - App image from Docker-app
   - DB image from Docker-db

6. **Run Database Container**
   Starts MySQL container.
7. **Run Application Container**
   Starts Tomcat container and links DB.
8. **Verify Deployment**
   Checks running containers.

7. Final Jenkins Pipeline

```
pipeline {
   agent any

   stages {

      stage('code-clone') {
         steps {
            git 'https://github.com/sonugupta4166/Docker-web-app.git'
         }
      }

      stage('clean-unused-files') {
         steps {
            sh '''
            rm -rf Docker-web ansible helm compose kubernetes README.md
            '''
         }
      }

      stage('code-build') {
         steps {
            sh 'mvn clean package'
         }
      }

      stage('copy-war-to-docker-context') {
         steps {
            sh '''
            rm -rf Docker-app/target
            mkdir -p Docker-app/target
            cp target/*.war Docker-app/target/
            '''
         }
      }

      stage('image-build') {
         steps {
            sh 'docker build -t appimage ./Docker-app'
         }
      }

      stage('db-build') {
         steps {
            sh 'docker build -t dbimage ./Docker-db'
         }
      }

      stage('check-images') {
         steps {
            sh 'docker images'
         }
      }

      stage('dbimage-run') {
         steps {
            sh '''
            docker rm -f devopsdb || true
            docker run -d --name devopsdb -p 3306:3306 dbimage
            '''
         }
      }

      stage('appimage-run-with-link-db') {
         steps {
            sh '
            docker rm -f devopsapp || true
```

```
            docker run -d --name devopsapp  - p 8081:8080 --link devopsdb:mysqlconnect appimage '
          }
      }

      stage('verify') {
          steps {
              sh 'docker ps'
          }
      }
  }
```



Definition

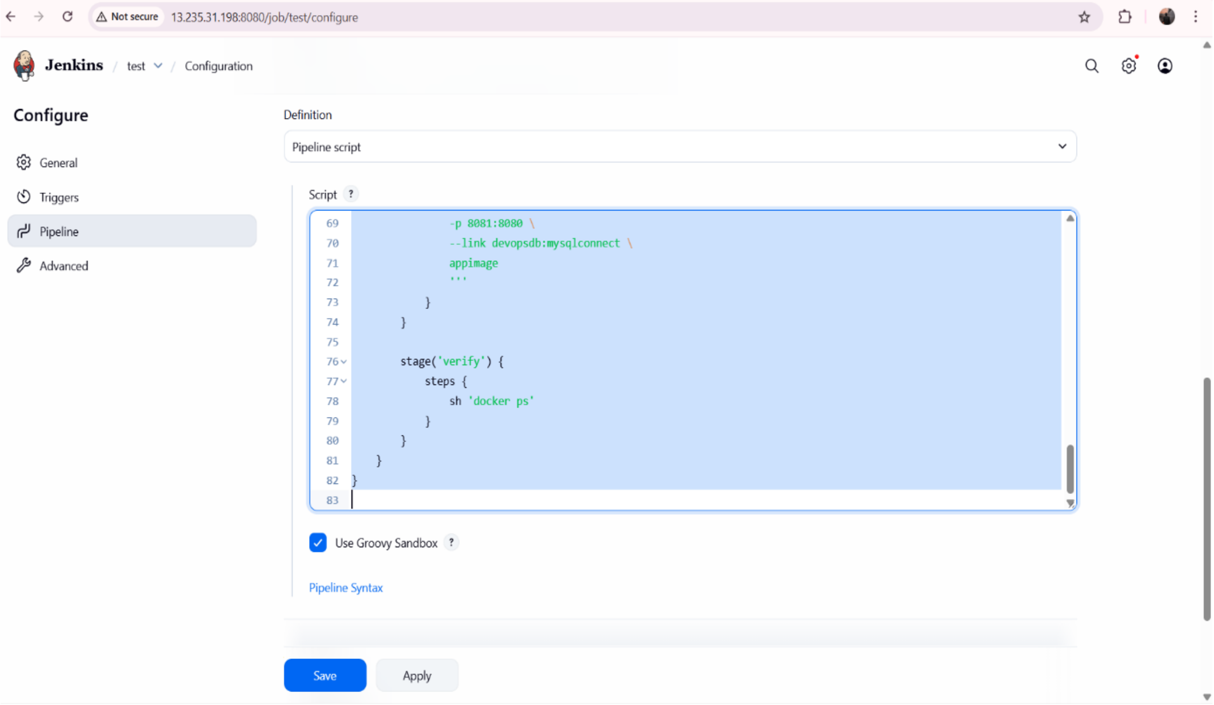
Pipeline script

Script

```
pipeline {
    agent any

    stages {

        stage('code-clone') {
            steps {
                git 'https://github.com/sonugupta4166/Docker-web-app.git'
            }
        }

        stage('clean-unused-files') {
            steps {
                sh '''
                rm -rf Docker-web ansible helm compose kubernetes README.md
```

☑ Use Groovy Sandbox

Pipeline Syntax



Definition

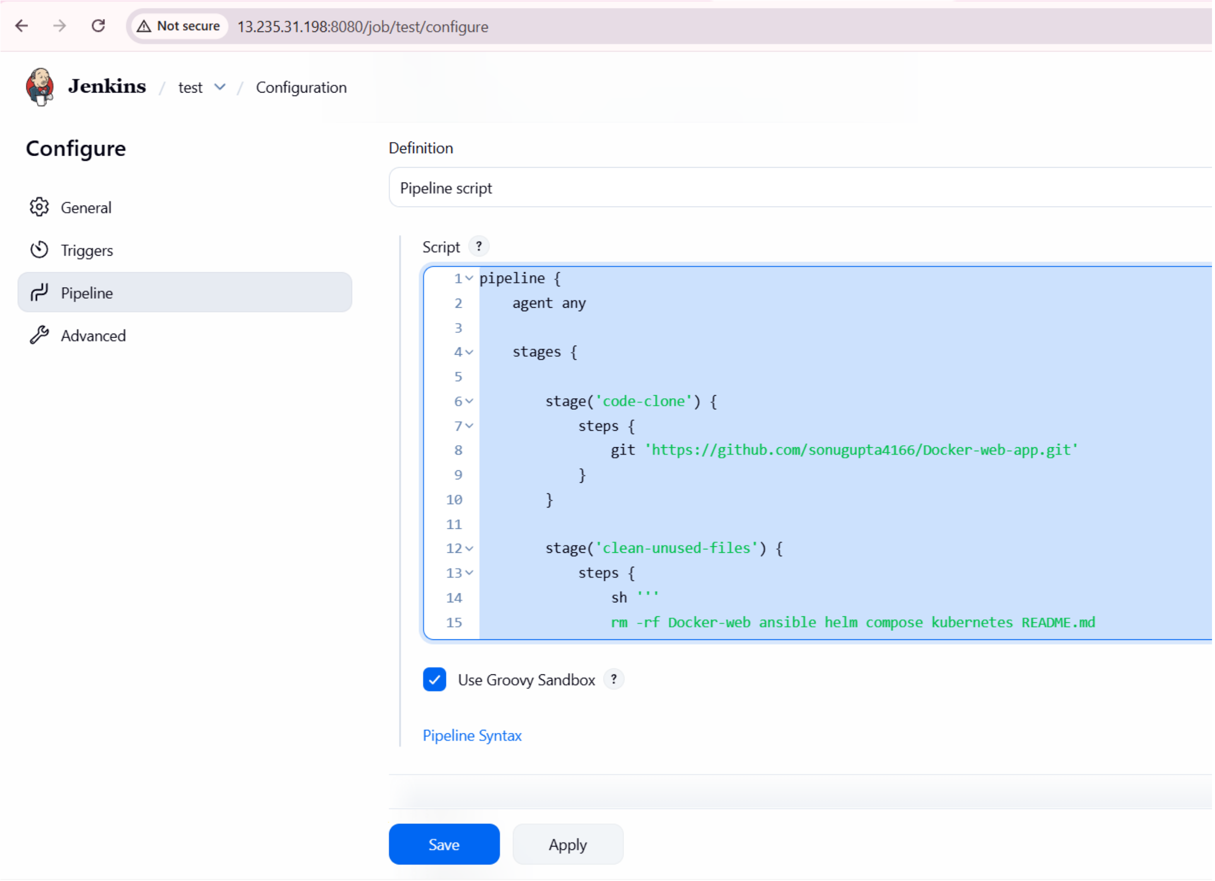
Pipeline script

Script

```
                -p 8081:8080 \
                --link devopsdb:mysqlconnect \
                appimage
                '''
            }
        }

        stage('verify') {
            steps {
                sh 'docker ps'
            }
        }
    }
}

```

☑ Use Groovy Sandbox

Pipeline Syntax

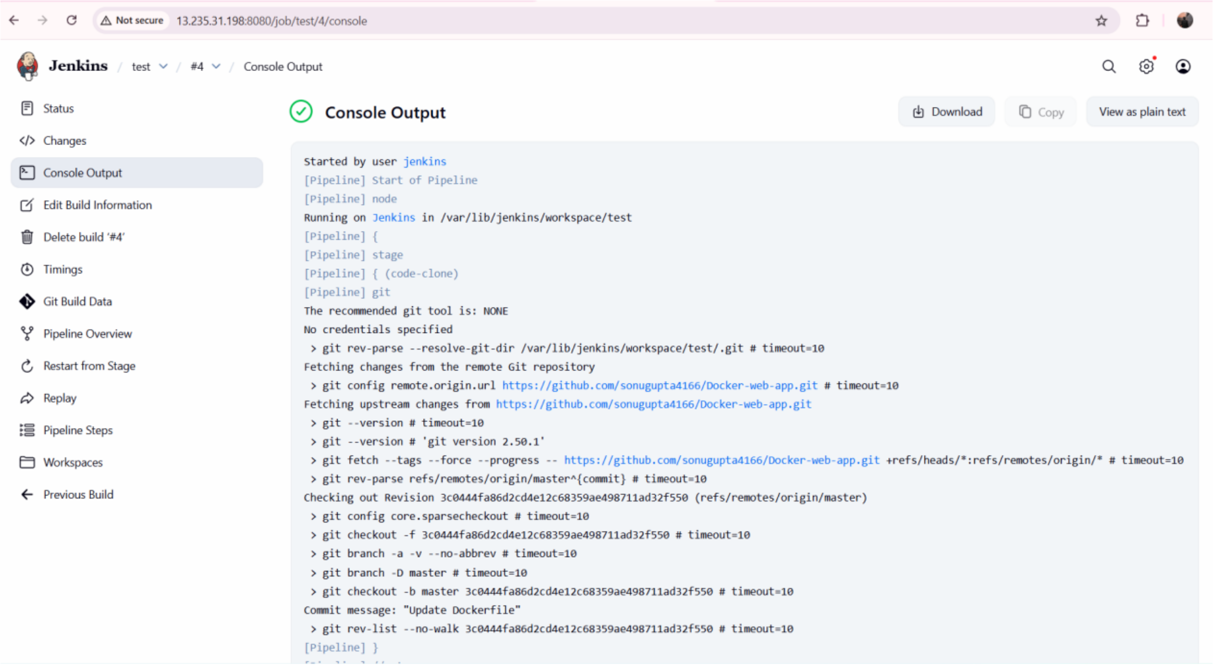


Definition

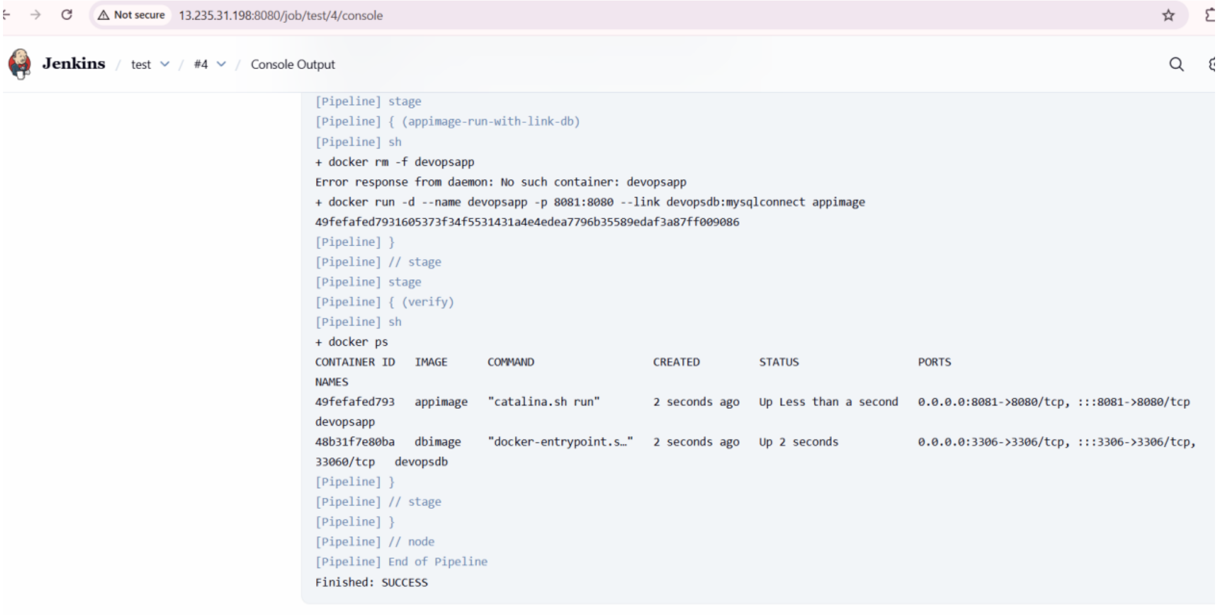
Pipeline script

```
pipeline {
    agent any

    stages {

        stage('code-clone') {
            steps {
                git 'https://github.com/sonugupta4166/Docker-web-app.git'
            }
        }

        stage('clean-unused-files') {
            steps {
                sh '''
                rm -rf Docker-web ansible helm compose kubernetes README.md
```

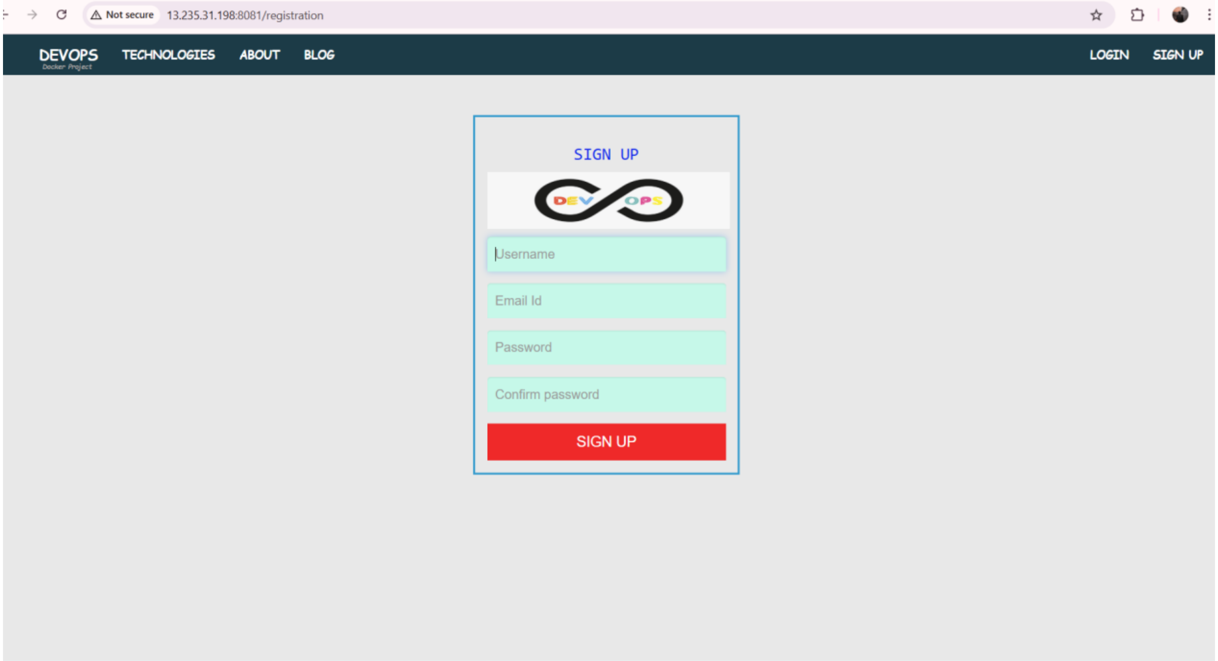
☑ Use Groovy Sandbox ?

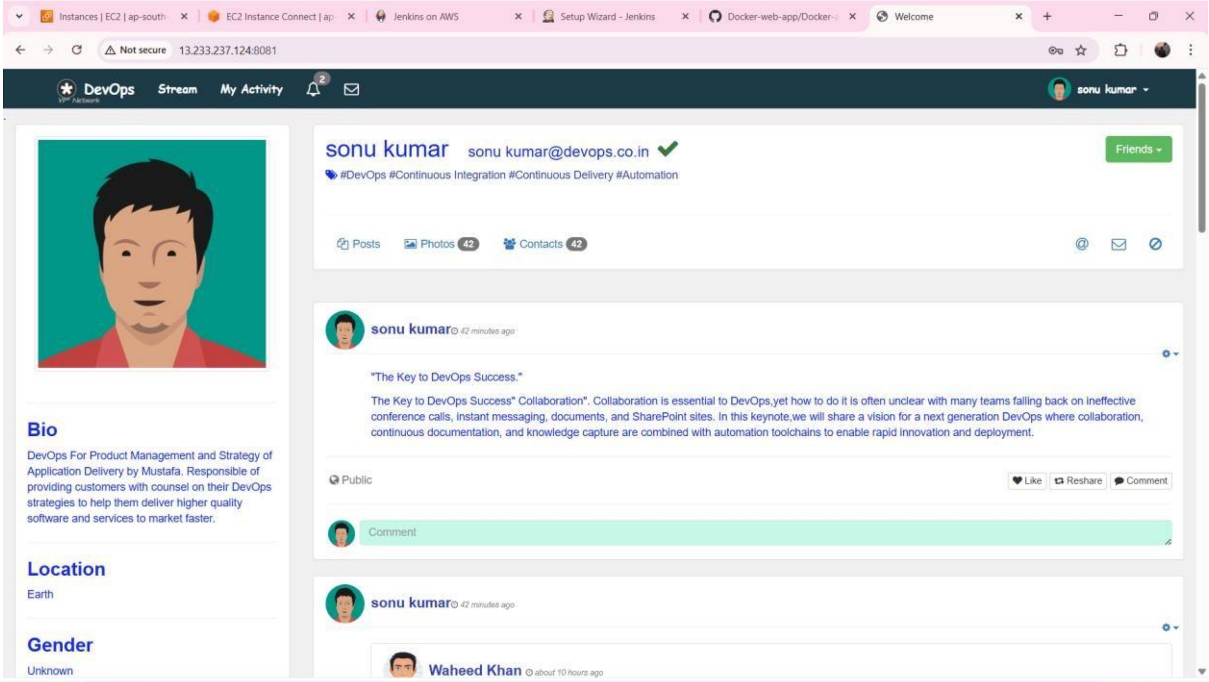


```
Started by user jenkins
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/test
[Pipeline] {
[Pipeline] stage
[Pipeline] { (code-clone)
[Pipeline] git
The recommended git tool is: NONE
No credentials specified
 > git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/test/.git # timeout=10
Fetching changes from the remote Git repository
 > git config remote.origin.url https://github.com/sonugupta4166/Docker-web-app.git # timeout=10
Fetching upstream changes from https://github.com/sonugupta4166/Docker-web-app.git
 > git --version # timeout=10
 > git --version # 'git version 2.50.1'
 > git fetch --tags --force --progress -- https://github.com/sonugupta4166/Docker-web-app.git +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision 3c0444fa86d2cd4e12c68359ae498711ad32f550 (refs/remotes/origin/master)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f 3c0444fa86d2cd4e12c68359ae498711ad32f550 # timeout=10
 > git branch -a -v --no-abbrev # timeout=10
 > git branch -D master # timeout=10
 > git checkout -b master 3c0444fa86d2cd4e12c68359ae498711ad32f550 # timeout=10
Commit message: "Update Dockerfile"
 > git rev-list --no-walk 3c0444fa86d2cd4e12c68359ae498711ad32f550 # timeout=10
[Pipeline] }
```



```
[Pipeline] stage
[Pipeline] { (appimage-run-with-link-db)
[Pipeline] sh
+ docker rm -f devopsapp
Error response from daemon: No such container: devopsapp
+ docker run -d --name devopsapp -p 8081:8080 --link devopsdb:mysqlconnect appimage
49fefafed7931605373f34f5531431a4e4edea7796b35589edaf3a87ff009086
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (verify)
[Pipeline] sh
+ docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED         STATUS                  PORTS
NAMES
49fefafed793   appimage   "catalina.sh run"        2 seconds ago   Up Less than a second   0.0.0.0:8081->8080/tcp, :::8081->8080/tcp
devopsapp
48b31f7e80ba   dbimage    "docker-entrypoint.s…"   2 seconds ago   Up 2 seconds            0.0.0.0:3306->3306/tcp, :::3306->3306/tcp,
33060/tcp   devopsdb
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```





13.233.237.124:8081

sonu kumar sonu kumar@devops.co.in ✓

#DevOps #Continuous Integration #Continuous Delivery #Automation

Bio

DevOps For Product Management and Strategy of Application Delivery by Mustafa. Responsible of providing customers with counsel on their DevOps strategies to help them deliver higher quality software and services to market faster.

Location

Earth

Gender

Unknown

sonu kumar 42 minutes ago

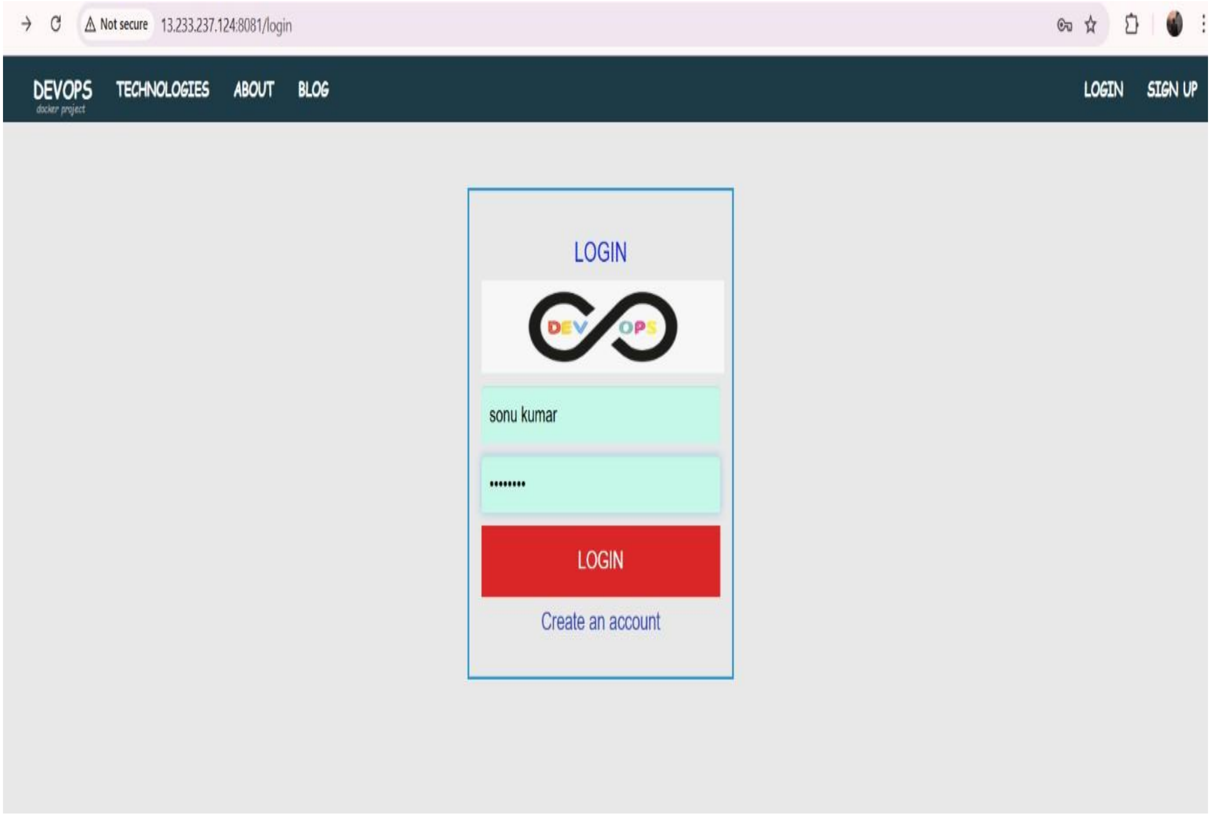
"The Key to DevOps Success."

The Key to DevOps Success" Collaboration". Collaboration is essential to DevOps,yet how to do it is often unclear with many teams falling back on ineffective conference calls, instant messaging, documents, and SharePoint sites. In this keynote,we will share a vision for a next generation DevOps where collaboration, continuous documentation, and knowledge capture are combined with automation toolchains to enable rapid innovation and deployment.

Public

sonu kumar 42 minutes ago

Waheed Khan about 10 hours ago



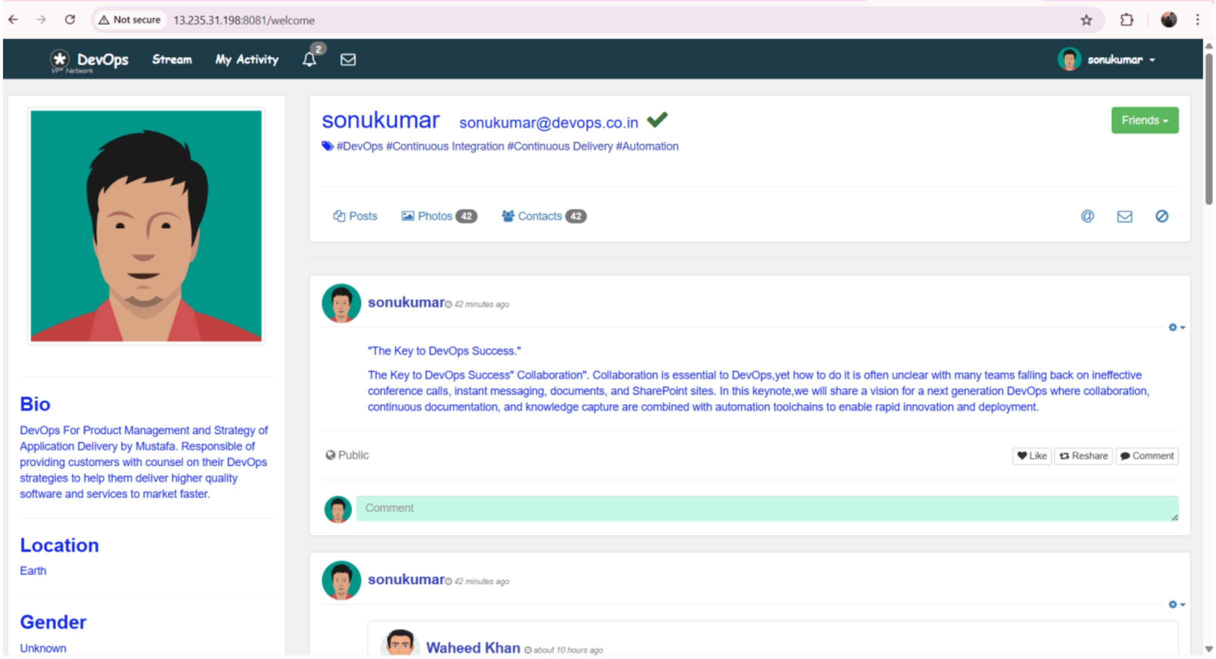
13.233.237.124:8081/login

LOGIN

sonu kumar

••••••••

http://EC2-Public-IP:8080



sonukumar sonukumar@devops.co.in ✓

#DevOps #Continuous Integration #Continuous Delivery #Automation

Bio

DevOps For Product Management and Strategy of Application Delivery by Mustafa. Responsible of providing customers with counsel on their DevOps strategies to help them deliver higher quality software and services to market faster.

Location

Earth

Gender

Unknown

sonukumar 42 minutes ago

"The Key to DevOps Success."

The Key to DevOps Success" Collaboration". Collaboration is essential to DevOps,yet how to do it is often unclear with many teams falling back on ineffective conference calls, instant messaging, documents, and SharePoint sites. In this keynote,we will share a vision for a next generation DevOps where collaboration, continuous documentation, and knowledge capture are combined with automation toolchains to enable rapid innovation and deployment.

sonukumar 42 minutes ago

Waheed Khan about 10 hours ago