JUNIOR DEVOPS ASSINGMENT

Banking and Finance Domain

SUBMITTED BY: SALMAN SADIQ

DATE:23/03/2024

Junior Level Assignment

Automating Dockerized Deployments

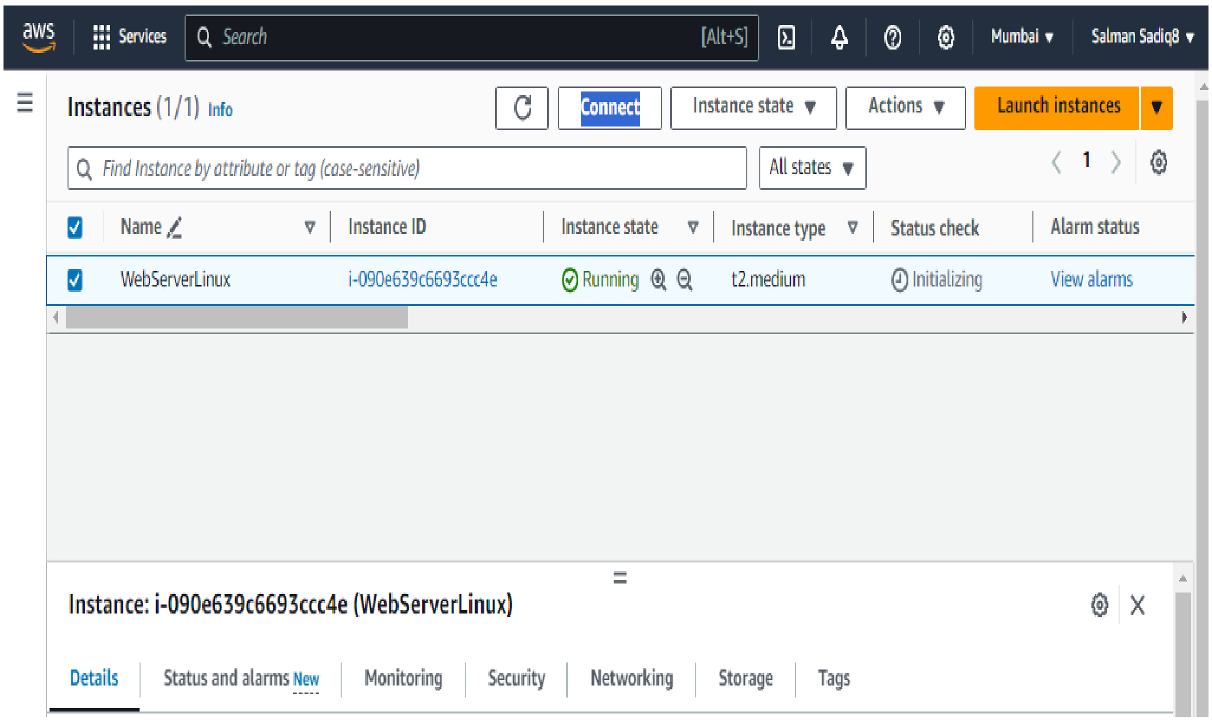
Scenario:

You are tasked with automating the deployment process for a Dockerized web application. The goal is to set up a continuous integration and continuous deployment (CI/CD) pipeline using basic scripting and Docker concepts.

Requirements:

- Docker Setup: Install Docker on a Linux server and set up a basic Docker environment.
- Git Repository: Create a Git repository for the web application code.
- CI Pipeline: Set up a basic CI pipeline using a CI tool (e.g., Jenkins, GitLab CI). The pipeline should:
- Trigger on code commits to the Git repository.
- Build the Docker image for the web application.
- Push the Docker image to a Docker registry (e.g., Docker Hub).
- CD Pipeline: Implement a basic CD pipeline to deploy the Dockerized application. The pipeline should:
- Pull the latest Docker image from the registry.
- Stop and remove existing containers.
- Run a new container with the updated image.
- Bash Scripts: Write Bash scripts to automate Docker-related tasks, such as building images, pushing to registries, and deploying containers.

Step 1:

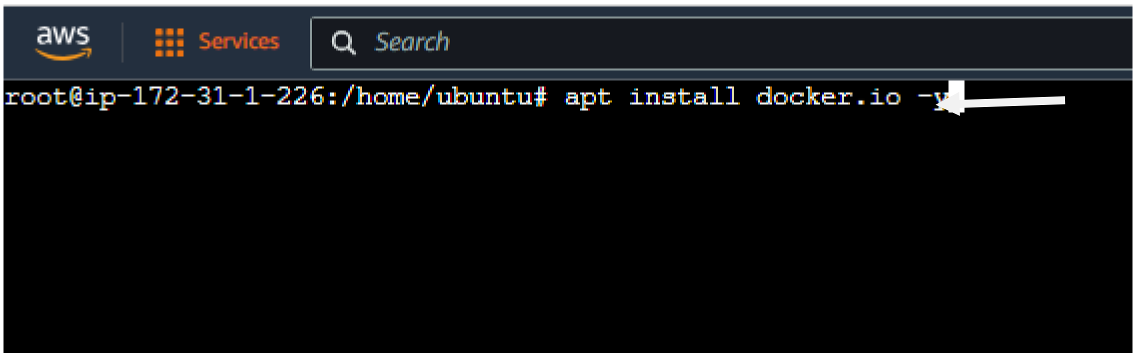


Name	Instance ID	Instance state	Instance type	Status check	Alarm status
WebServerLinux	i-090e639c6693ccc4e	Running	t2.medium	Initializing	View alarms

Instance: i-090e639c6693ccc4e (WebServerLinux)

Here we created AWS Ec2 Ubuntu Instance

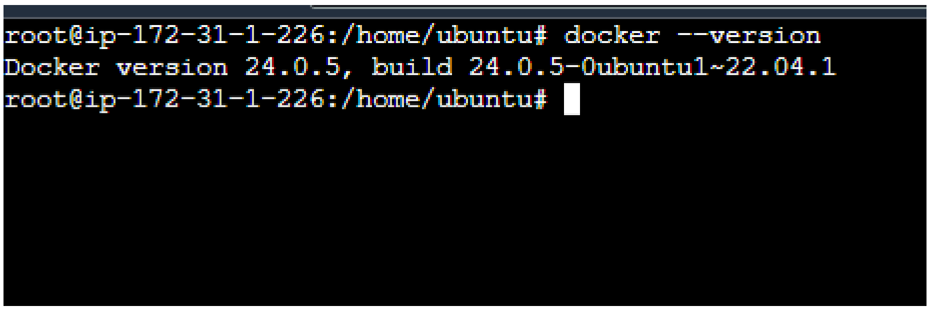
Step 2:



```
root@ip-172-31-1-226:/home/ubuntu# apt install docker.io -y
```

Here we install docker in ec2 instance with the above code

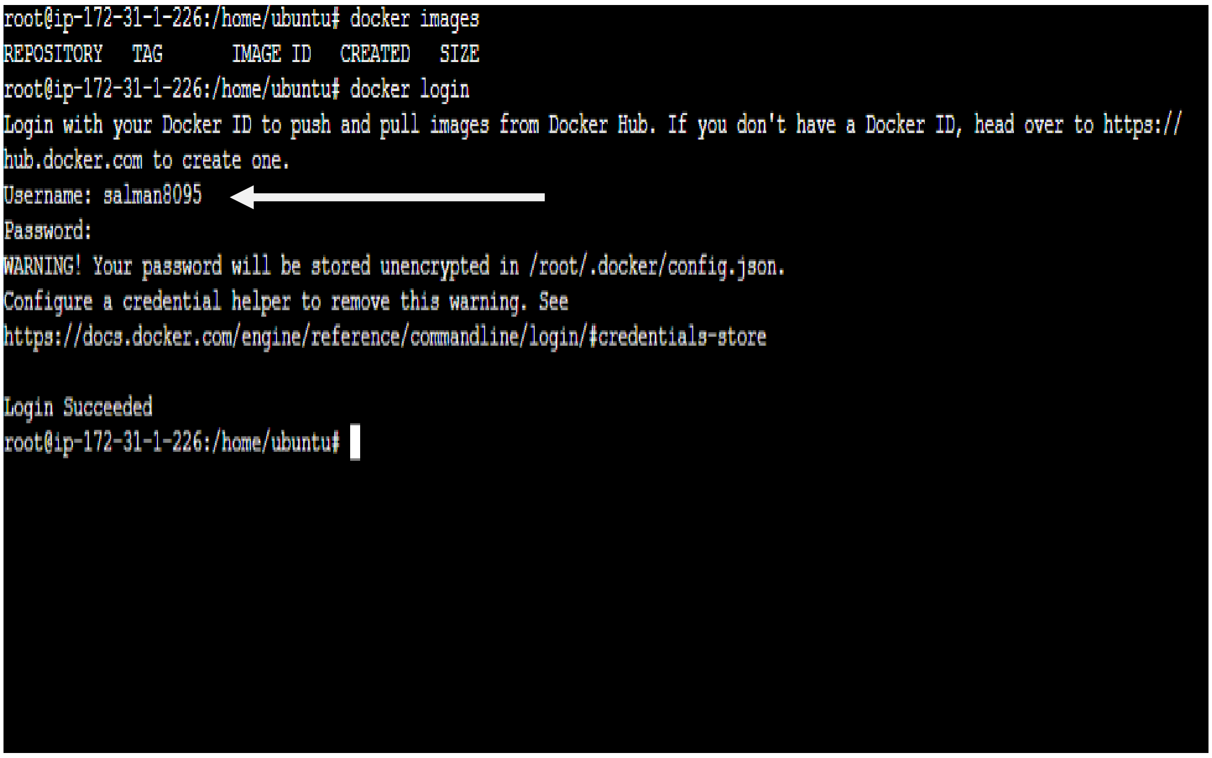
Step 3:



```
root@ip-172-31-1-226:/home/ubuntu# docker --version
Docker version 24.0.5, build 24.0.5-0ubuntu1~22.04.1
root@ip-172-31-1-226:/home/ubuntu#
```

Docker has been successfully installed

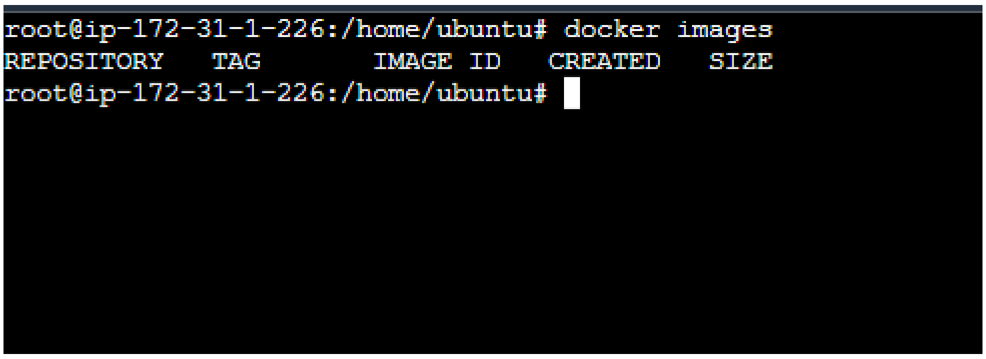
Step 4:



```
root@ip-172-31-1-226:/home/ubuntu# docker images
REPOSITORY   TAG       IMAGE ID   CREATED   SIZE
root@ip-172-31-1-226:/home/ubuntu# docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://
hub.docker.com to create one.
Username: salman8095
Password:
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
root@ip-172-31-1-226:/home/ubuntu#
```

Here we login into docker hub using credentials using the terminal

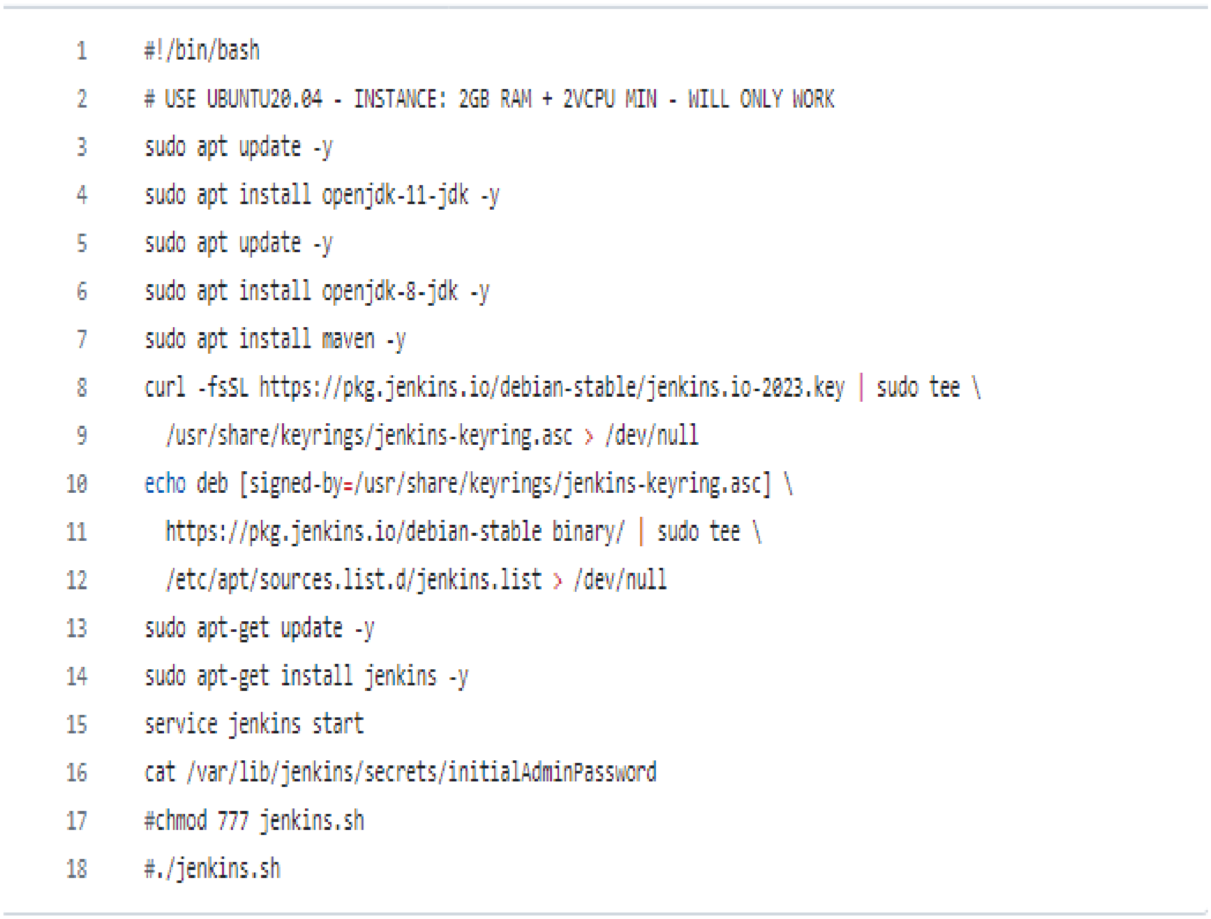
Step 5:



```
root@ip-172-31-1-226:/home/ubuntu# docker images
REPOSITORY    TAG       IMAGE ID    CREATED    SIZE
root@ip-172-31-1-226:/home/ubuntu#
```

we use this command to check docker images

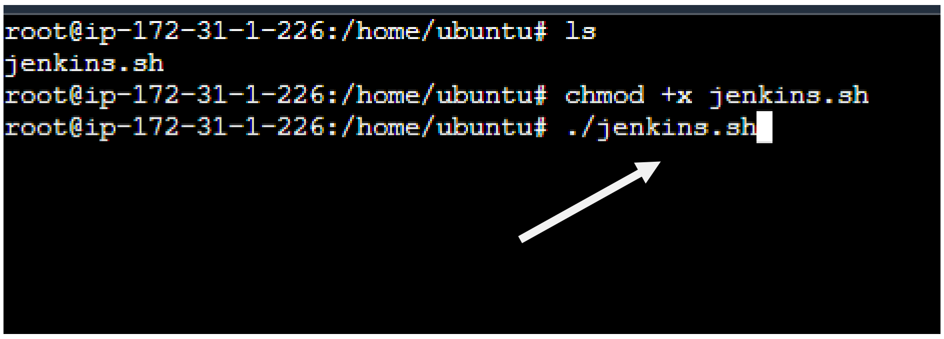
Step 6:



```
#!/bin/bash
# USE UBUNTU20.04 - INSTANCE: 2GB RAM + 2VCPU MIN - WILL ONLY WORK
sudo apt update -y
sudo apt install openjdk-11-jdk -y
sudo apt update -y
sudo apt install openjdk-8-jdk -y
sudo apt install maven -y
curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key | sudo tee \
  /usr/share/keyrings/jenkins-keyring.asc > /dev/null
echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
  https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
  /etc/apt/sources.list.d/jenkins.list > /dev/null
sudo apt-get update -y
sudo apt-get install jenkins -y
service jenkins start
cat /var/lib/jenkins/secrets/initialAdminPassword
#chmod 777 jenkins.sh
#./jenkins.sh
```

We install Jenkins using the above code in Ec2 instance

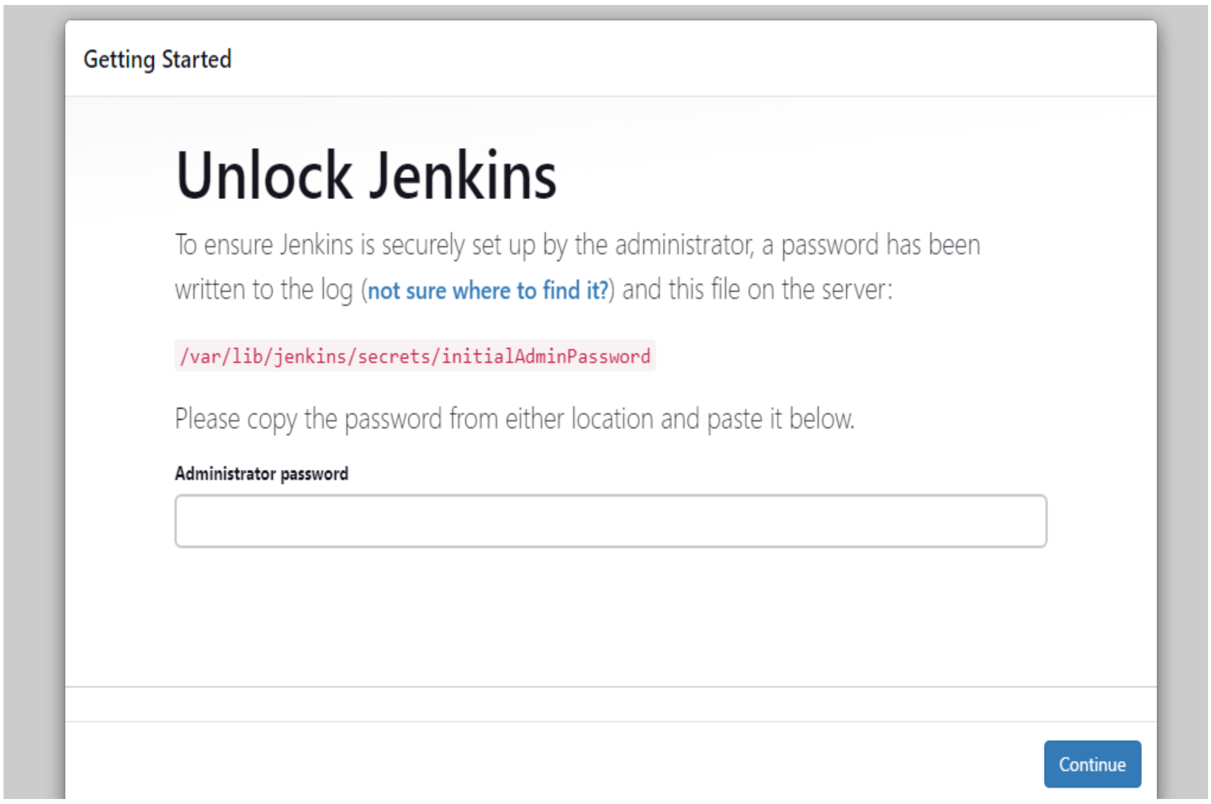
Step 7:



```
root@ip-172-31-1-226:/home/ubuntu# ls
jenkins.sh
root@ip-172-31-1-226:/home/ubuntu# chmod +x jenkins.sh
root@ip-172-31-1-226:/home/ubuntu# ./jenkins.sh
```

Here we download and install Jenkins using shell script

Step 8:



Getting Started

Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:

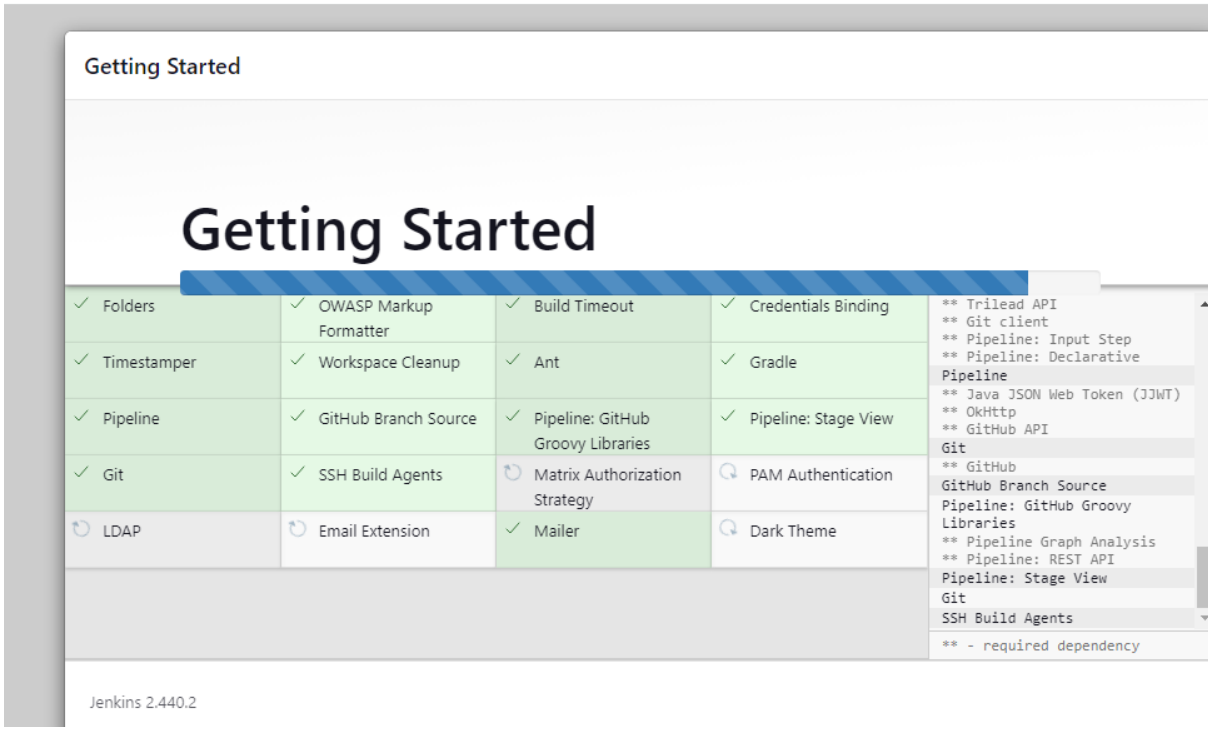
`/var/lib/jenkins/secrets/initialAdminPassword`

Please copy the password from either location and paste it below.

Administrator password

Jenkins has been Successfully installed in server on port 8080

Step 9:


Getting Started

Getting Started

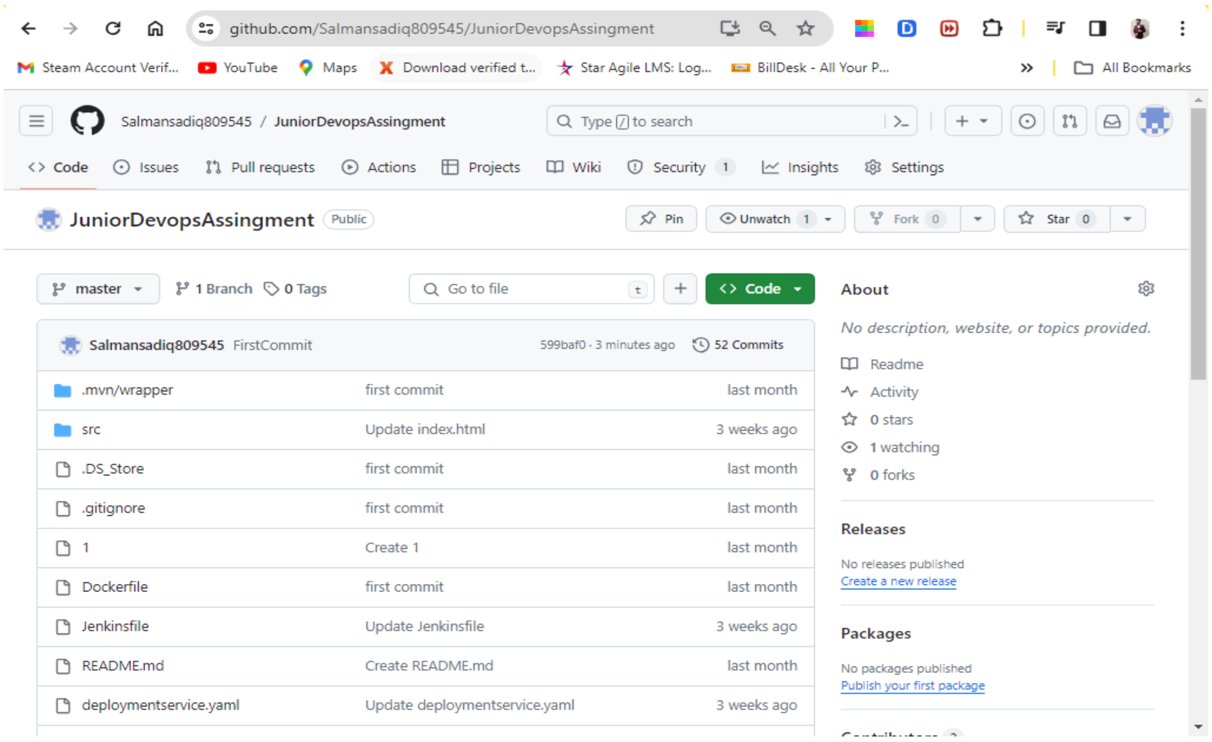
✓ Folders	✓ OWASP Markup Formatter	✓ Build Timeout	✓ Credentials Binding
✓ Timestamper	✓ Workspace Cleanup	✓ Ant	✓ Gradle
✓ Pipeline	✓ GitHub Branch Source	✓ Pipeline: GitHub Groovy Libraries	✓ Pipeline: Stage View
✓ Git	✓ SSH Build Agents	Matrix Authorization Strategy	PAM Authentication
LDAP	Email Extension	✓ Mailer	Dark Theme

```
** Trilead API
** Git client
** Pipeline: Input Step
** Pipeline: Declarative
Pipeline
** Java JSON Web Token (JJWT)
** OkHttp
** GitHub API
Git
** GitHub
GitHub Branch Source
Pipeline: GitHub Groovy
Libraries
** Pipeline Graph Analysis
** Pipeline: REST API
Pipeline: Stage View
Git
SSH Build Agents
** - required dependency
```

Jenkins 2.440.2

Some Necessary Jenkins extensions are being installed

Step 10:



github.com/Salmansadiq809545/JuniorDevopsAssingment

Salmansadiq809545 / JuniorDevopsAssingment

JuniorDevopsAssingment Public

master · 1 Branch · 0 Tags

Salmansadiq809545 FirstCommit 599baf0 · 3 minutes ago 52 Commits

.mvn/wrapper	first commit	last month
src	Update index.html	3 weeks ago
.DS_Store	first commit	last month
.gitignore	first commit	last month
1	Create 1	last month
Dockerfile	first commit	last month
Jenkinsfile	Update Jenkinsfile	3 weeks ago
README.md	Create README.md	last month
deploymentservice.yaml	Update deploymentservice.yaml	3 weeks ago

About

No description, website, or topics provided.

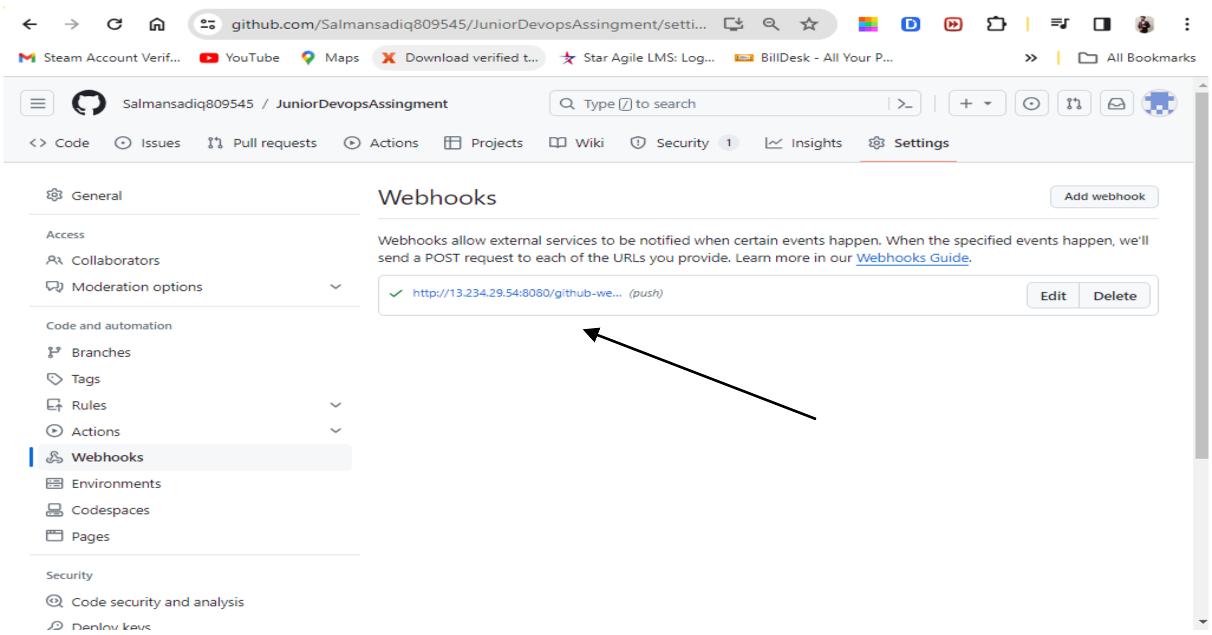
Readme · Activity · 0 stars · 1 watching · 0 forks

Releases: No releases published. Create a new release

Packages: No packages published. Publish your first package

This is the Github Repo where code is kept

Step 11:



github.com/Salmansadiq809545/JuniorDevopsAssingment/setti...

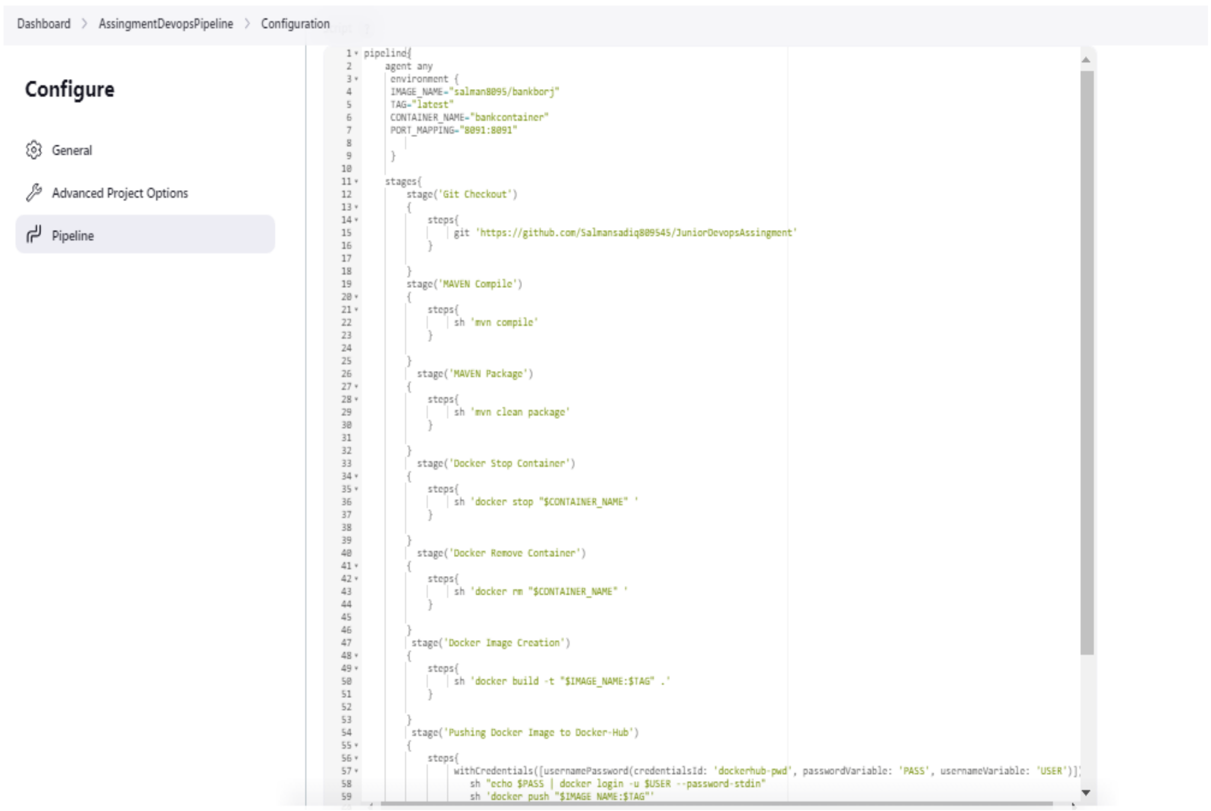
Webhooks

Webhooks allow external services to be notified when certain events happen. When the specified events happen, we'll send a POST request to each of the URLs you provide. Learn more in our Webhooks Guide.

✓ http://13.234.29.54:8080/github-we... (push) Edit Delete

Github webhook has been configured

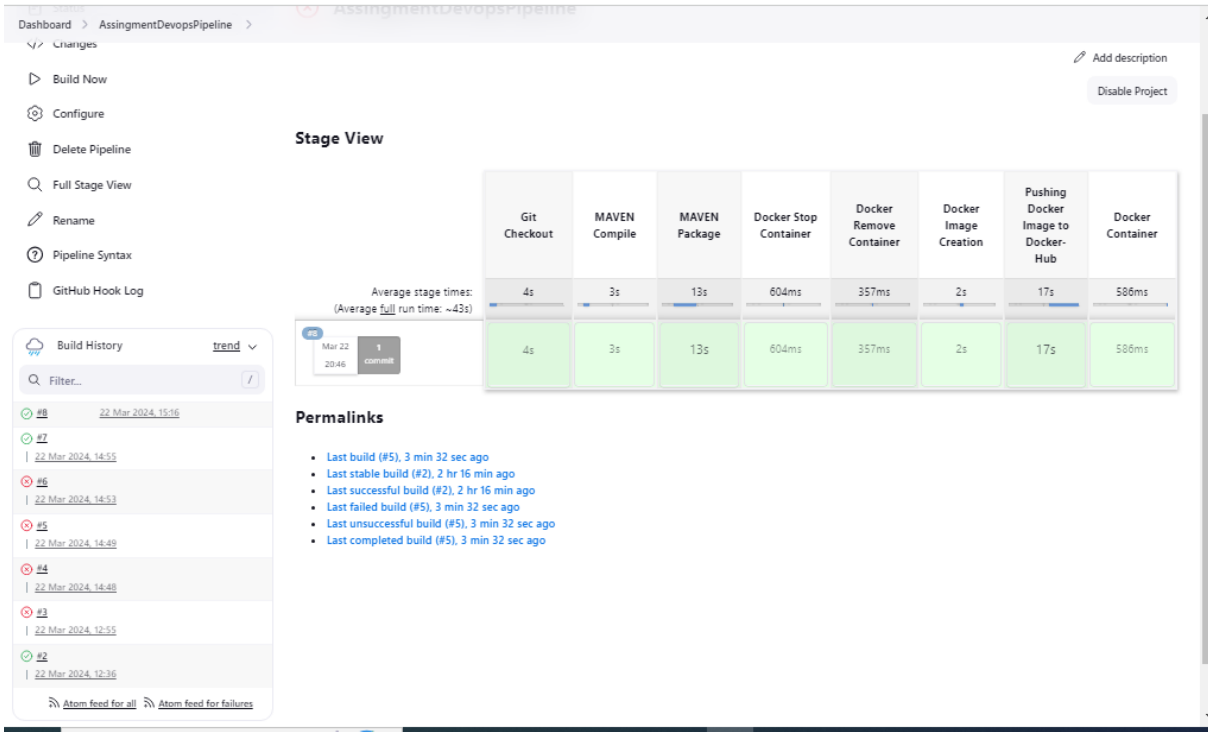
Step 12:



```
pipelind{
    agent any
     environment {
     IMAGE_NAME="salman8095/bankborj"
     TAG="latest"
     CONTAINER_NAME="bankcontainer"
     PORT_MAPPING="8091:8091"

     }

    stages{
        stage('Git Checkout')
        {
            steps{
                git 'https://github.com/Salmansadiq809545/JuniorDevopsAssingment'
            }

        }
        stage('MAVEN Compile')
        {
            steps{
                sh 'mvn compile'
            }

        }
          stage('MAVEN Package')
        {
            steps{
                sh 'mvn clean package'
            }

        }
          stage('Docker Stop Container')
        {
            steps{
                sh 'docker stop "$CONTAINER_NAME" '
            }

        }
          stage('Docker Remove Container')
        {
            steps{
                sh 'docker rm "$CONTAINER_NAME" '
            }

        }
         stage('Docker Image Creation')
        {
            steps{
                sh 'docker build -t "$IMAGE_NAME:$TAG" .'
            }

        }
         stage('Pushing Docker Image to Docker-Hub')
        {
            steps{
                withCredentials([usernamePassword(credentialsId: 'dockerhub-pwd', passwordVariable: 'PASS', usernameVariable: 'USER')])
                   sh "echo $PASS | docker login -u $USER --password-stdin"
                   sh 'docker push "$IMAGE_NAME:$TAG"'
```

This is the Jenkins pipeline which does Continuous integration and continuous deployment

Step 13:



Dashboard > AssingmentDevopsPipeline >

Changes

Build Now

Configure

Delete Pipeline

Full Stage View

Rename

Pipeline Syntax

GitHub Hook Log

Add description

Disable Project

Stage View

	Git Checkout	MAVEN Compile	MAVEN Package	Docker Stop Container	Docker Remove Container	Docker Image Creation	Pushing Docker Image to Docker-Hub	Docker Container
Average stage times: (Average full run time: ~43s)	4s	3s	13s	604ms	357ms	2s	17s	586ms
#8 Mar 22 20:46 1 commit	4s	3s	13s	604ms	357ms	2s	17s	586ms

Permalinks

- Last build (#5), 3 min 32 sec ago
- Last stable build (#2), 2 hr 16 min ago
- Last successful build (#2), 2 hr 16 min ago
- Last failed build (#5), 3 min 32 sec ago
- Last unsuccessful build (#5), 3 min 32 sec ago
- Last completed build (#5), 3 min 32 sec ago

Build History trend

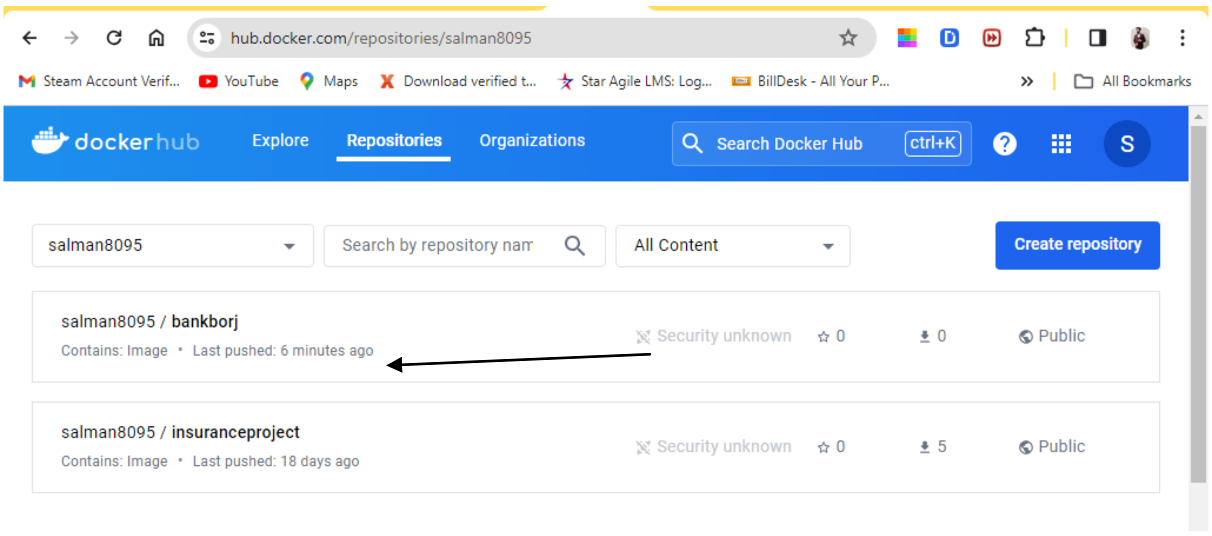
Filter...

✓ #8	22 Mar 2024, 15:16
✓ #7	22 Mar 2024, 14:55
⊗ #6	22 Mar 2024, 14:53
⊗ #5	22 Mar 2024, 14:49
⊗ #4	22 Mar 2024, 14:48
⊗ #3	22 Mar 2024, 12:55
✓ #2	22 Mar 2024, 12:36

Atom feed for all   Atom feed for failures

Pipeline has been successful and it has deployed the container on port 8091 and pushed the docker images to docker hub

Step 14:

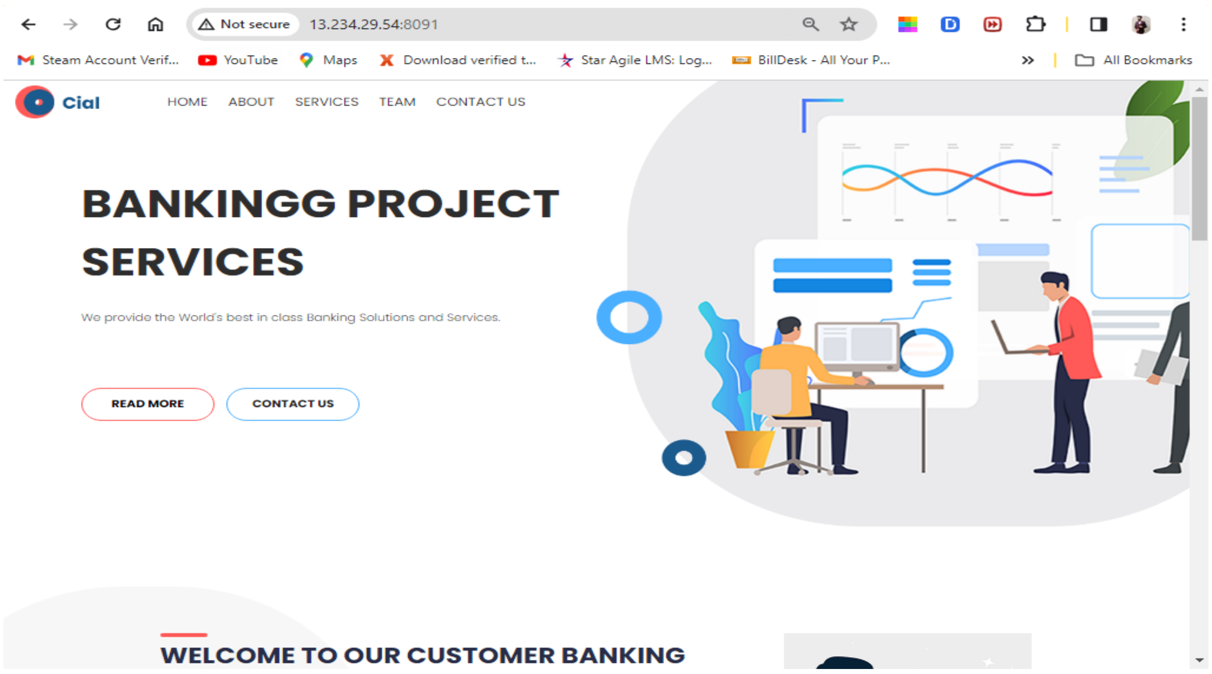

hub.docker.com/repositories/salman8095

salman8095 / bankborj
Contains: Image • Last pushed: 6 minutes ago
Security unknown ☆ 0 ⤓ 0 Public

salman8095 / insuranceproject
Contains: Image • Last pushed: 18 days ago
Security unknown ☆ 0 ⤓ 5 Public

Here is the docker hub where docker image has been pushed using Jenkins ci/cd pipeline

Step 15:


Not secure 13.234.29.54:8091

BANKINGG PROJECT SERVICES

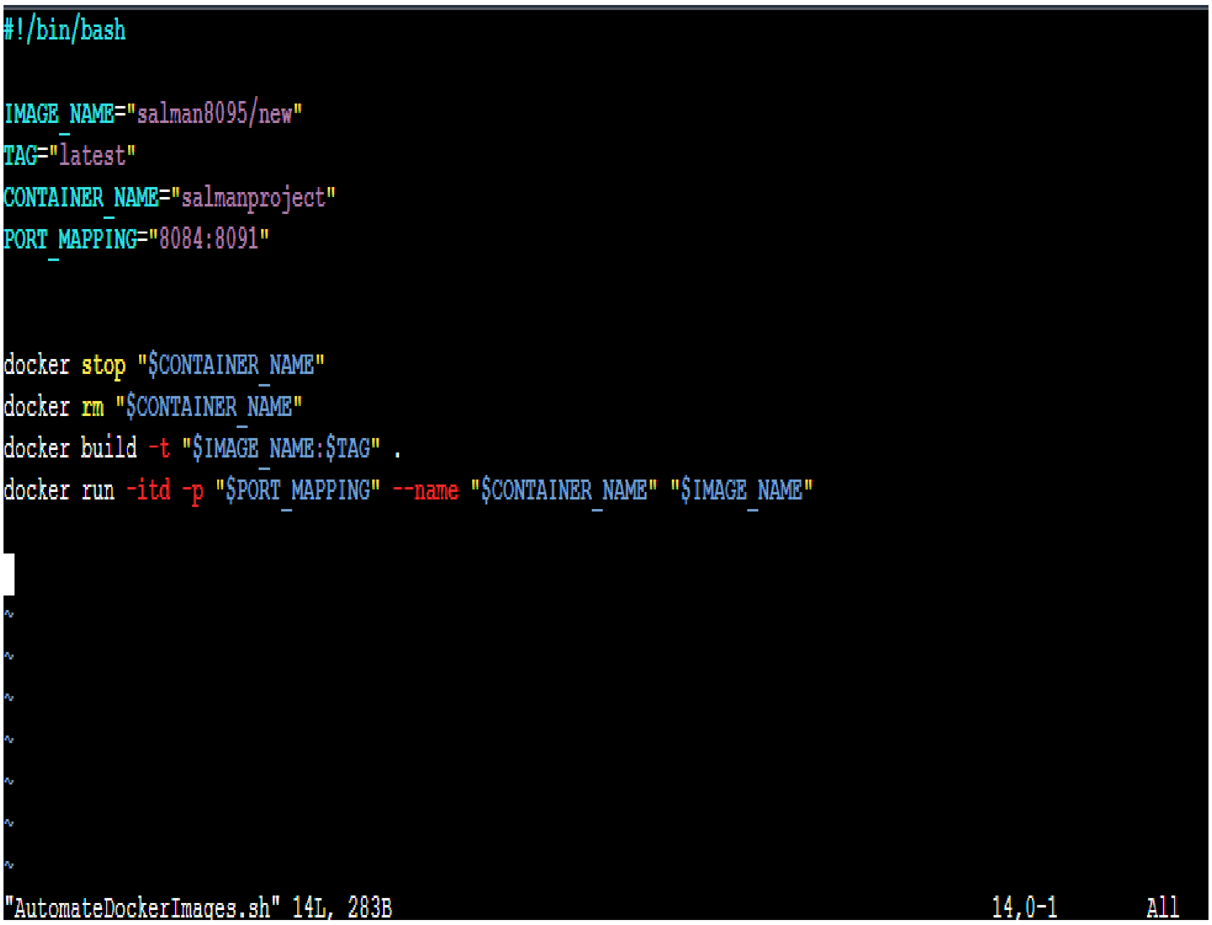
We provide the World's best in class Banking Solutions and Services.

WELCOME TO OUR CUSTOMER BANKING

Here is the docker container deployed at port 8091 and its running

Step 16:

Bash Scripting



```
#!/bin/bash

IMAGE_NAME="salman8095/new"
TAG="latest"
CONTAINER_NAME="salmanproject"
PORT_MAPPING="8084:8091"


docker stop "$CONTAINER_NAME"
docker rm "$CONTAINER_NAME"
docker build -t "$IMAGE_NAME:$TAG" .
docker run -itd -p "$PORT_MAPPING" --name "$CONTAINER_NAME" "$IMAGE_NAME"


~
~
~
~
~
~
~
"AutomateDockerImages.sh" 14L, 283B                                      14,0-1         All
```

Here is the bash scripting to automate the process of building docker image and docker container

THANK YOU