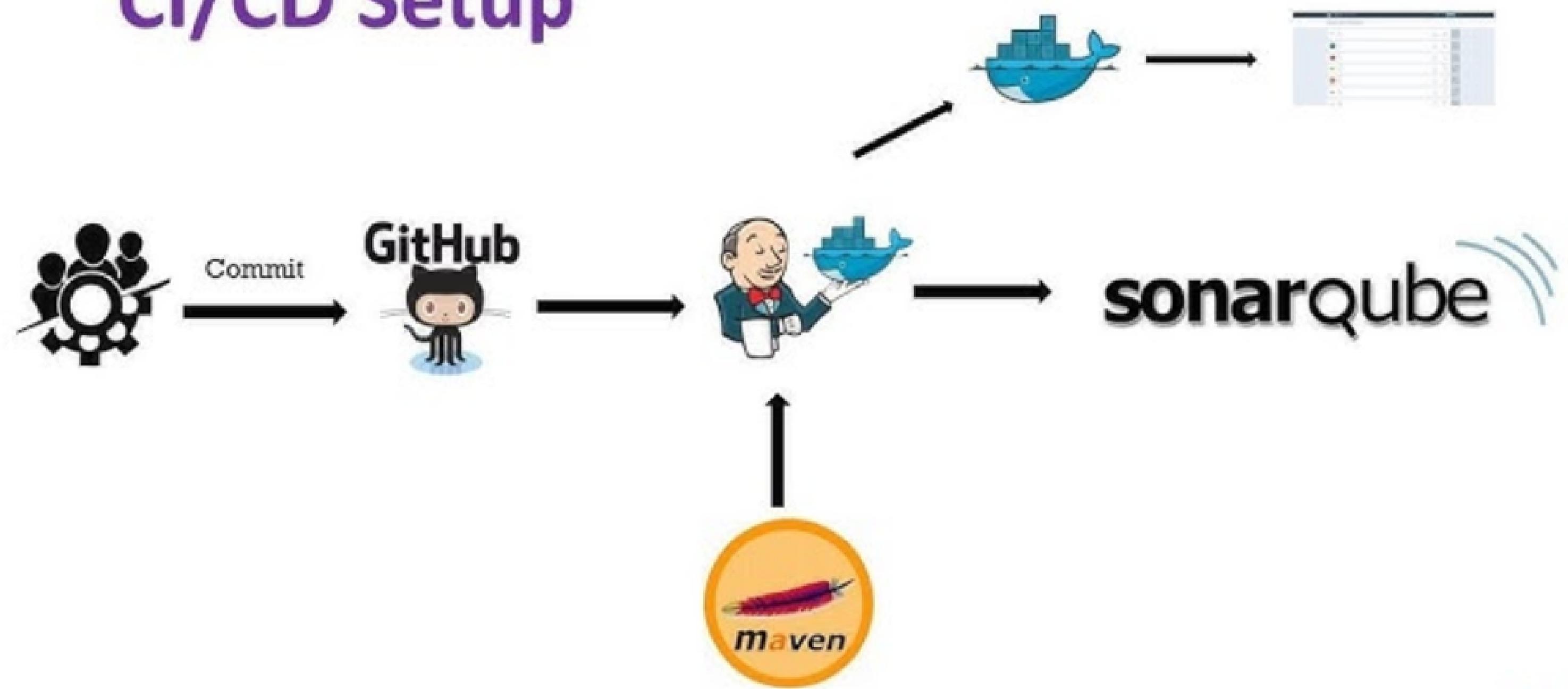


CI/CD Setup



suyash bobade

step1: Go to the aws management console and launch Ec2 instance with t2.medium size and install jenkins with the help of below commands:

EC2 > Instances > i-09e36ac554316b2a5

Instance summary for i-09e36ac554316b2a5 (jenkins02) [Info](#)

Updated 6 minutes ago

	C	Connect	Instance state ▾	Actions ▾
Instance ID	i-09e36ac554316b2a5 (jenkins02)			
IPv6 address	-			
Hostname type				
IP name:	ip-172-31-9-25.ap-south-1.compute.internal			
Answer private resource DNS name				
IPv4 (A)				
Auto-assigned IP address	15.207.248.159 [Public IP]			
Public IPv4 address	15.207.248.159 open address			
Instance state	Running			
Private IP DNS name (IPv4 only)	ip-172-31-9-25.ap-south-1.compute.internal			
Instance type	t2.medium			
VPC ID	vpc-0cbd7600121692d7b			
Private IPv4 addresses	172.31.9.25			
Public IPv4 DNS	ec2-15-207-248-159.ap-south-1.compute.amazonaws.com open address			
Elastic IP addresses	-			
AWS Compute Optimizer finding				
	Opt-in to AWS Compute Optimizer for recommendations.			
	Learn more			

#installation of java

```
sudo apt update  
sudo apt install fontconfig openjdk-17-jre  
java -version
```

#installation of jenkins

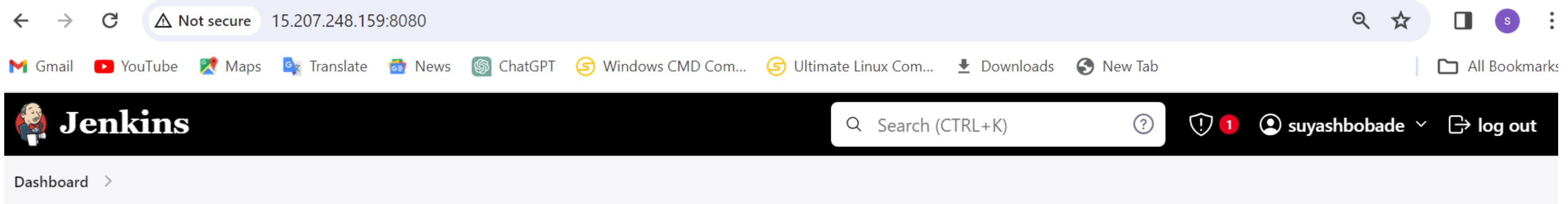
```
sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \  
https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
```

```
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
```

```
sudo apt-get install jenkins -y
```

step2: Go to chrome, access jenkin server with instance public ip and default port "http://public_ip:8080" and do the basic configuration which is required as usual.



step3: install plugins which are required for our project.

A screenshot of the Jenkins plugin management interface. It shows a list of installed plugins. The first plugin listed is 'SonarQube Scanner for Jenkins 2.17.2'. Its status is 'Enabled', indicated by a blue toggle switch with a checkmark. Below the plugin name is a brief description: 'This plugin allows an easy integration of SonarQube, the open source platform for Continuous Inspection of code quality.' and a link 'Report an issue with this plugin'. The second plugin listed is 'OWASP Dependency-Check Plugin 5.5.0'. Its status is also 'Enabled', indicated by a blue toggle switch with a checkmark. Below the plugin name is a brief description: 'This plug-in can independently execute a Dependency-Check analysis and visualize results.' and a link 'Dependency-Check is a utility that identifies project dependencies and checks if there are any known, publicly disclosed, vulnerabilities.'

[Docker Pipeline](#) 572.v950f58993843

Build and use Docker containers from pipelines.

[Report an issue with this plugin](#)



[Docker plugin](#) 1.6

This plugin integrates Jenkins with Docker

[Report an issue with this plugin](#)



[docker-build-step](#) 2.11

This plugin allows to add various docker commands to your job as build steps.

[Report an issue with this plugin](#)



[Eclipse Temurin installer Plugin](#) 1.5

Provides an installer for the JDK tool that downloads the JDK from <https://adoptium.net>

[Report an issue with this plugin](#)



step4: now go to the dashboard > manage jenkins > tool configure plugins there that we have installed.

Add JDK

JDK

Name

jdk11

Install automatically ?

Install from adoptium.net ?

Version ?

jdk8u382-b05

Add Installer ▾

Add Gradle

Gradle

name ?

gradle

Install automatically ?

Install from Gradle.org

Version

Gradle 1.6

A screenshot of the Jenkins 'Tool Configuration' page. It shows two sections: 'JDK' and 'Gradle'. In the 'JDK' section, a new entry named 'jdk11' is being configured. It includes an 'Install automatically' checkbox (checked), an 'Install from adoptium.net' option, and a dropdown menu set to 'jdk8u382-b05'. Below this is an 'Add Installer' button. In the 'Gradle' section, a new entry named 'gradle' is being configured. It also includes an 'Install automatically' checkbox (checked), an 'Install from Gradle.org' option, and a dropdown menu set to 'Gradle 1.6'. Below this is an 'Add Installer' button.

Add SonarQube Scanner

☰ SonarQube Scanner

Name

 sonar-scanner

Install automatically ?

☰ Install from Maven Central

Version

 SonarQube Scanner 5.0.1.3006

Add Maven

☰ Maven

Name

 maven

Install automatically ?

☰ Install from Apache

Version

 3.9.4

Add Dependency-Check

Dependency-Check

Name

DC

Install automatically ?

Install from github.com

Version

dependency-check 6.5.1

Add Docker

Docker

Name

docker

Install automatically ?

Download from docker.com

Docker version ?

latest

step5: Go to dashboard choose "new item" give a suitable name to your project and create pipeline project. (will recommend you to always go with "hello world" pipeline syntax)

Dashboard > All >

Enter an item name

Cbank01
» Required field

 **Freestyle project**
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.

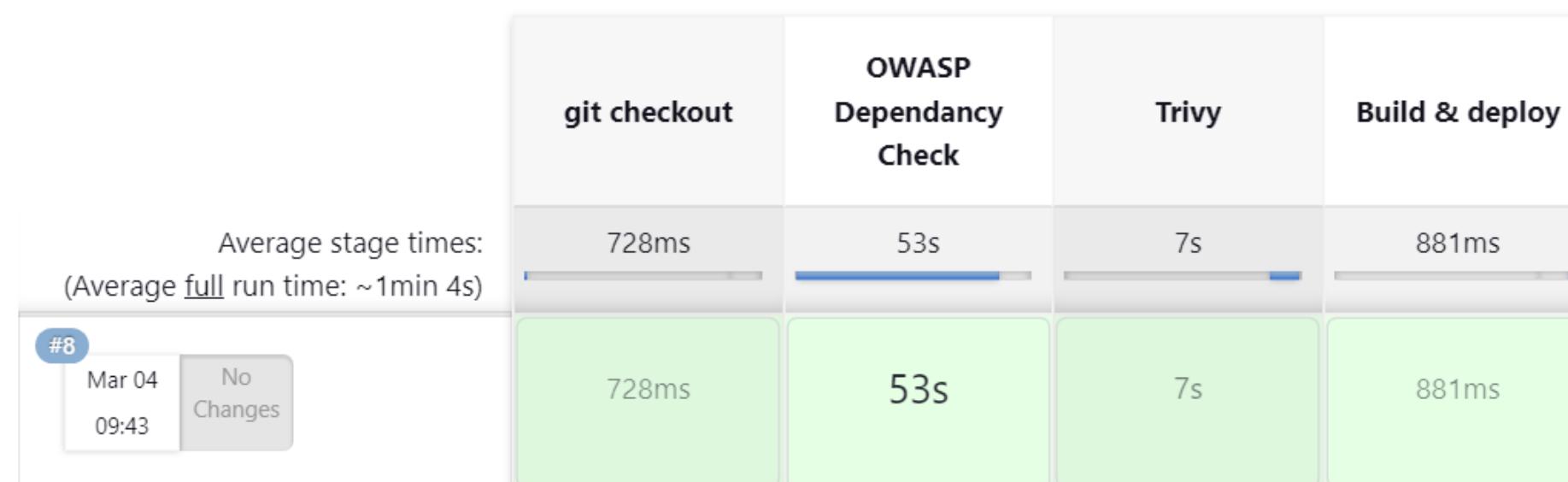
 **Maven project**
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

 **Pipeline**
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

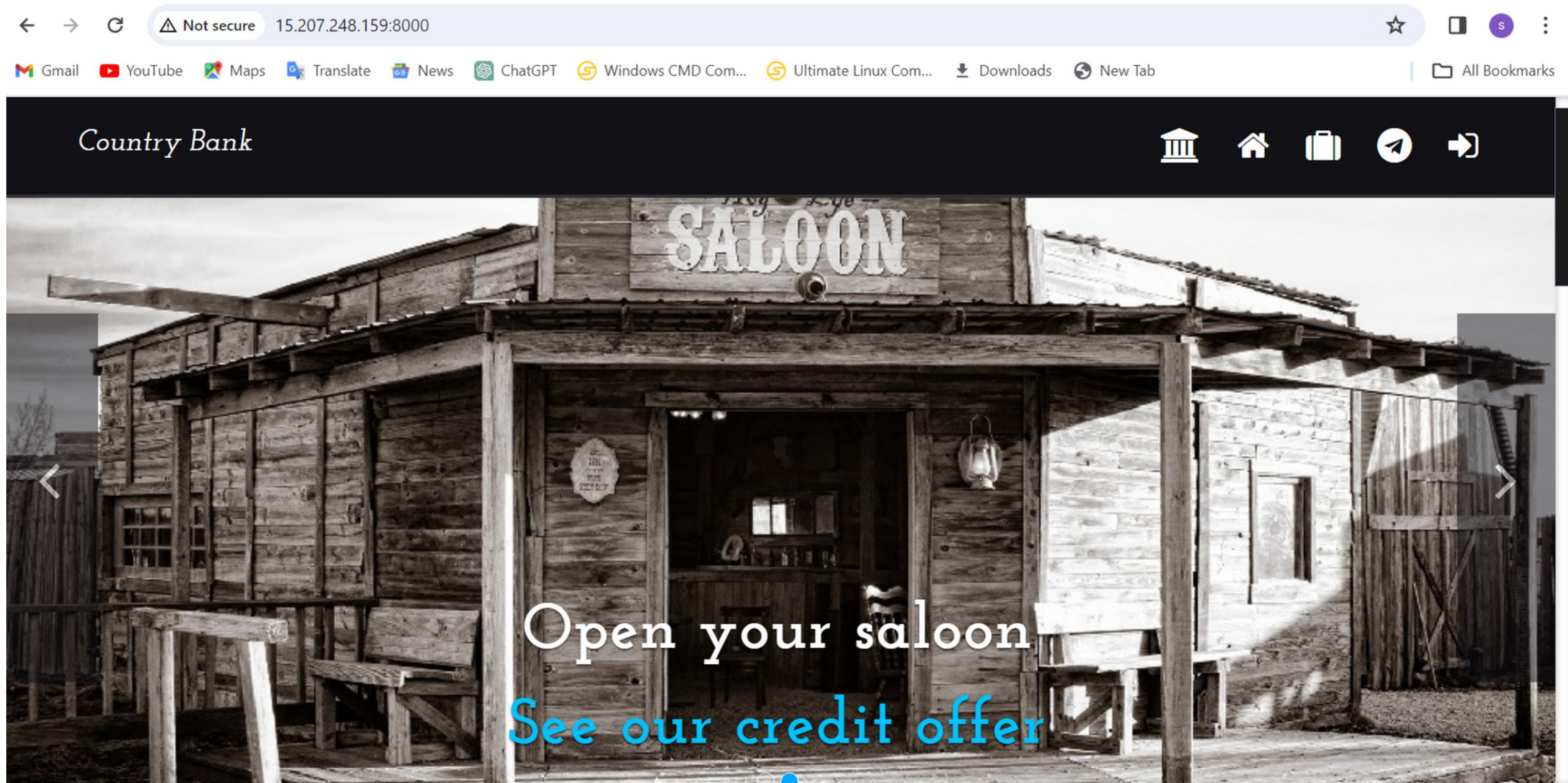
S	W	Name ↓	Last Success	Last Failure	Last Duration	
		Cbank	13 hr #8	N/A	1 min 4 sec	

#pipeline of our project (try to fix the issue if the cicd pipeline fail to execute correctly)

```
Script ?  
1 pipeline {  
2     agent any  
3  
4     stages {  
5         stage('git checkout') {  
6             steps {  
7                 git 'https://github.com/jaiswaladi246/CountryBank.git'  
8             }  
9         }  
10  
11         stage('OWASP Dependency Check') {  
12             steps {  
13                 dependencyCheck additionalArguments: '--scan ./ ', odcInstallation: 'DC'  
14                 dependencyCheckPublisher pattern: '**/dependency-check-report.xml'  
15             }  
16         }  
17  
18         stage('Trivy') {  
19             steps {  
20                 sh "trivy fs ."  
21             }  
22         }  
23  
24         stage('Build & deploy') {  
25             steps {  
26                 sh "docker-compose up -d"  
27             }  
28         }  
29     }  
30 }
```



#we have successfully deployed our application (make sure to add port '8000' in your security group to access the application)





THANK YOU !