AM

sakshi palkar's Blog

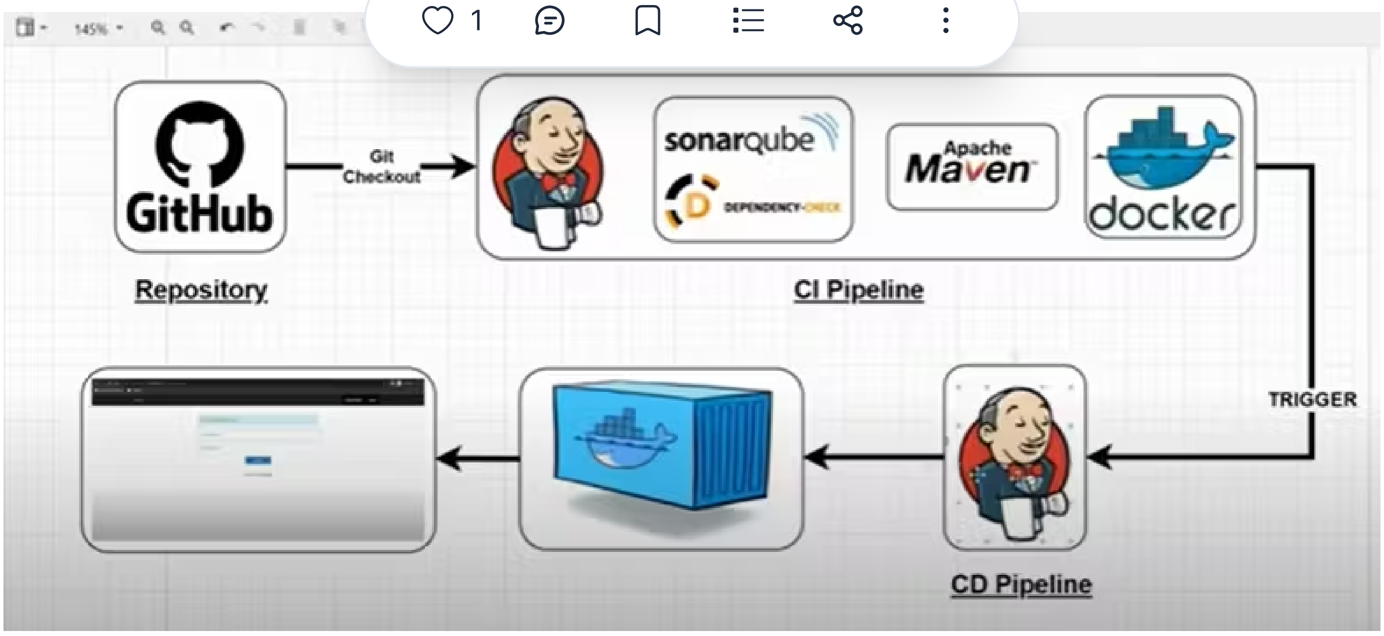
Following

Real-World CI/CD DevSecOps Pipeline for Deployment of Shopping Cart Application

SP sakshi palkar

Sep 5, 2023 · 4 min read

TABLE OF CONTENTS



Hello friends, we will be deploying a Shopping Cart Application. This is an everyday use case scenario used by several organizations. We will be using two Jenkins jobs, After completion of the CI pipeline CD pipeline will automatically triggered. Hope this detailed blog is useful.

We will be deploying our application using Docker Container.

Steps:-

Step 1 — Create an Ubuntu T2 Large Instance

Step 2 — Install Jenkins, Docker. Create a Sonarqube Container using Docker.

Step 3 — Install Plugins like JDK, Sonarqube Scanner, Maven, OWASP Dependency Check,

Step 4 — Create a Pipeline Project in Jenkins.

Step 5 — Install OWASP Dependency Check Plugins

Step 6 — Docker Image Build and Push

Step 7 — Deploy the i

Step 8 — Finally deploy on port 8070 using the Groovy pipeline script mentioned

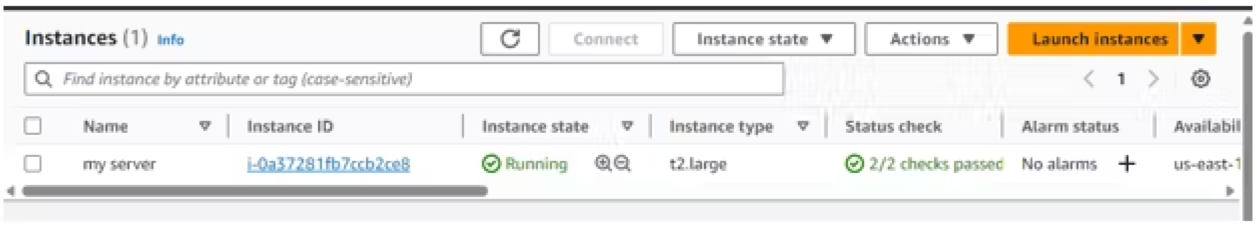
Step 9 — Access the Real World Application

Step 10 — Terminate the AWS EC2 Instance

References

Now, lets get started and dig deeper into each of these steps :-

Step 1 — Launch an AWS T2 Large Instance. Use the image as Ubuntu. You can create a new key pair or use an existing one. Enable HTTP and HTTPS settings in the Security Group.



Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availabil
my server	i-0a37281fb7ccb2ce8	Running	t2.large	2/2 checks passed	No alarms	us-east-1

Step 2 — Install Jenkins, Docker .

2A — To Install Jenkins

Connect to your console, and enter these commands to Install Jenkins

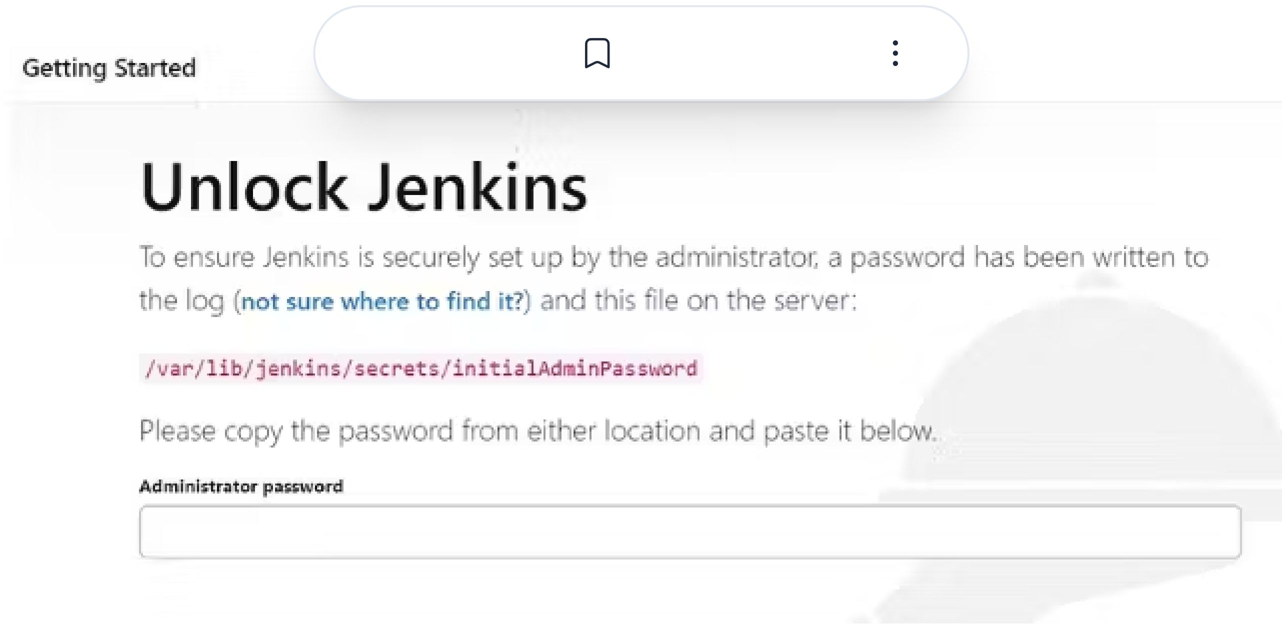
```
sudo apt-get update

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 update
sudo apt install openjdk-17-jdk
sudo apt install openjdk-17-jre

sudo systemctl enable jenkins
sudo systemctl start jenkins
sudo systemctl status jenkins

sudo cat  /var/lib/jenkins/secrets/initialAdminPassword
```

Once Jenkins is installed, you will need to go to your AWS EC2 Security Group and open Inbound Port 8080, since Jenkins works on Port 8080.

Now, grab your Public IP Address

```
<EC2 Public IP Address:8080>
sudo cat /var/lib/jenkins/secrets/initialAdminPassword
```

Unlock Jenkins using an administrative password and install the required plugins.



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.

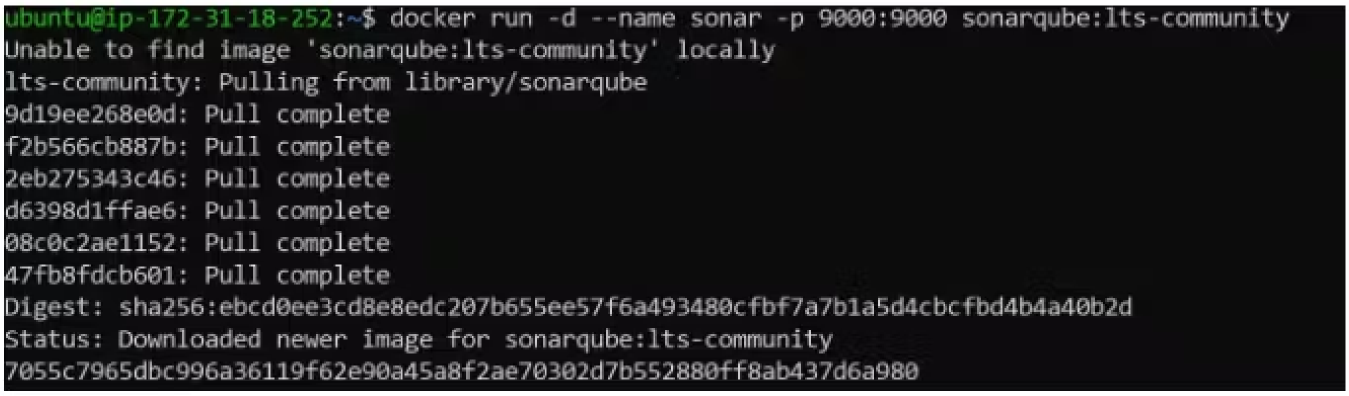
Administrator password

2B — Install Docker

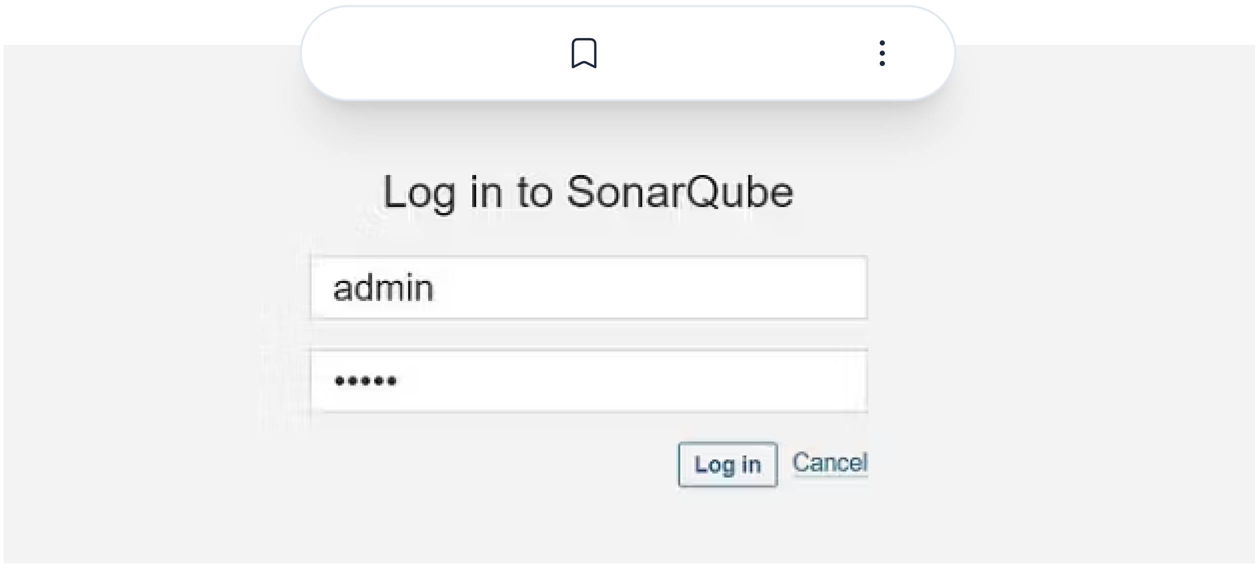
```
sudo apt-get update
sudo apt-get install docker.io -y
sudo usermod -aG docker $USER
sudo chmod 777 /var/run/docker.sock
sudo docker ps
```

After the docker installation, we create a sonarqube container (Remember added 9000 port in the security group)

```
docker run -d --name sonar -p 9000:9000 sonarqube:lts-community
```



```
ubuntu@ip-172-31-18-252:~$ docker run -d --name sonar -p 9000:9000 sonarqube:lts-community
Unable to find image 'sonarqube:lts-community' locally
lts-community: Pulling from library/sonarqube
9d19ee268e0d: Pull complete
f2b566cb887b: Pull complete
2eb275343c46: Pull complete
d6398d1ffae6: Pull complete
08c0c2ae1152: Pull complete
47fb8fdcb601: Pull complete
Digest: sha256:ebcd0ee3cd8e8edc207b655ee57f6a493480cfbf7a7b1a5d4cbcfbd4b4a40b2d
Status: Downloaded newer image for sonarqube:lts-community
7055c7965dbc996a36119f62e90a45a8f2ae70302d7b552880ff8ab437d6a980
```



Next, we will login to Jenkins and start to configure our Pipeline in Jenkins

Step 3 — Install Plugins like JDK, Sonarqube Scanner, Maven, OWASP Dependency Check,

3A — Install Plugin

Goto Manage Jenkins →Plugins → Available Plugins →

Install below plugins

1 → Eclipse Temurin Installer (Install without restart)

2 → SonarQube Scanner (Install without restart)

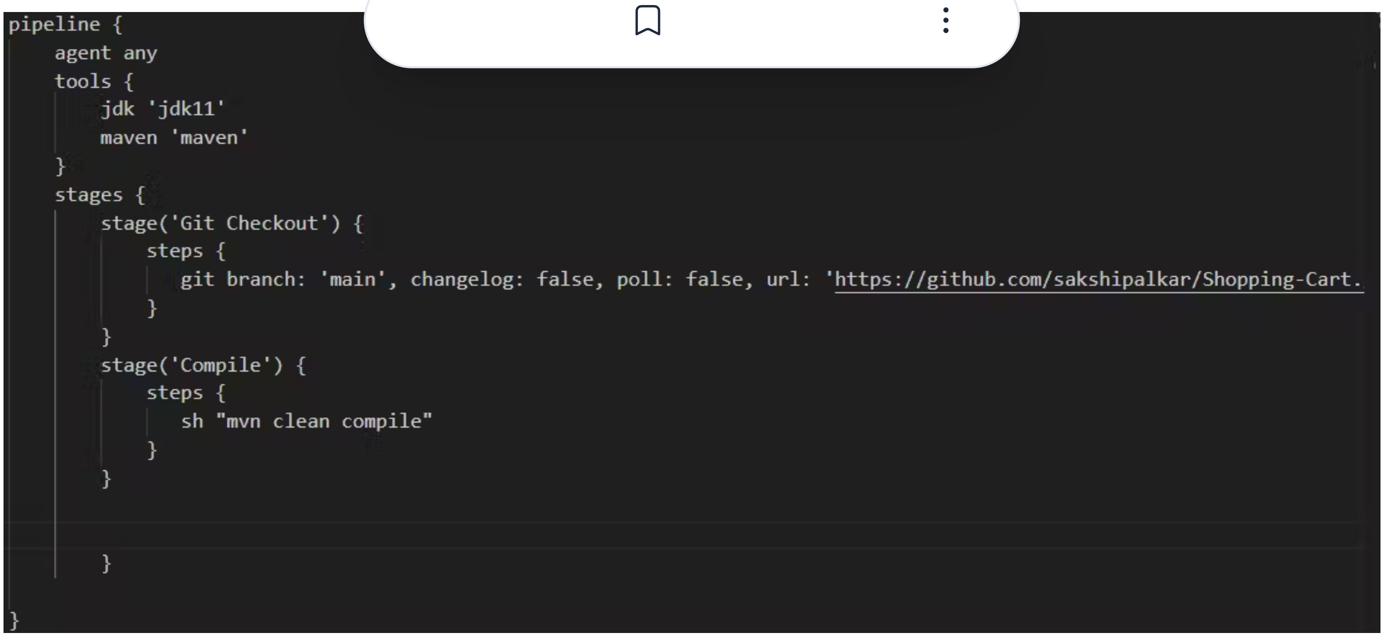
3B — Configure Java and Maven in Global Tool Configuration

Goto Manage Jenkins → Tools → Install JDK and Maven3 → Click on Apply and Save

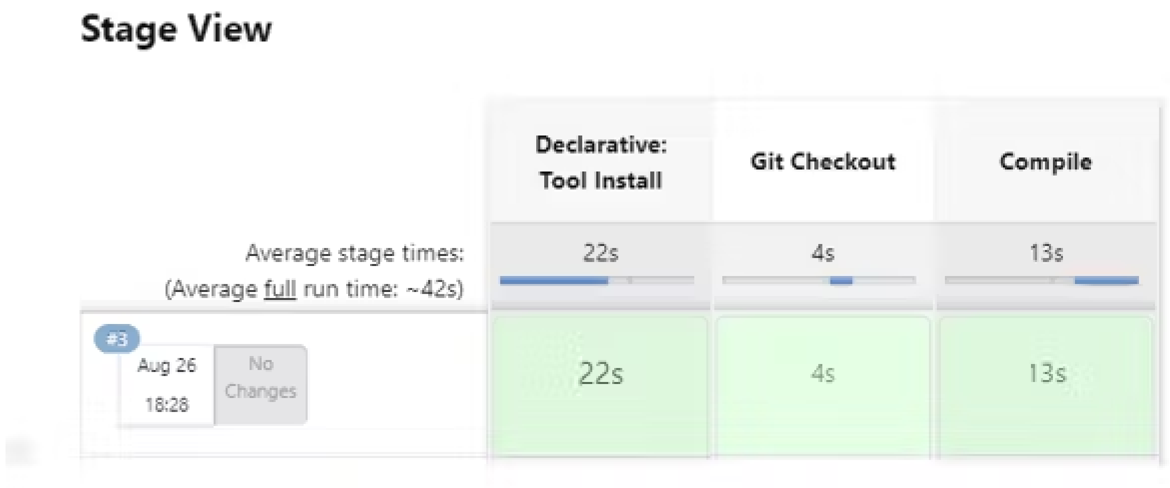
3C — Create a Job

Label it as Real-World CI-Pipeline, click on Pipeline and Ok.

Enter this in Pipeline Script,

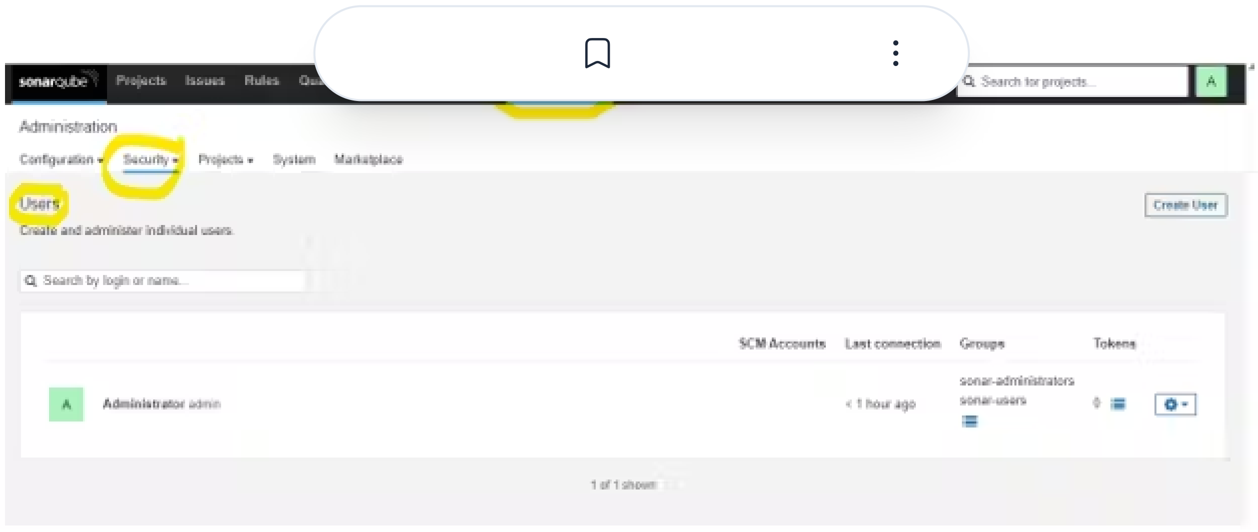

```
pipeline {
    agent any
    tools {
        jdk 'jdk11'
        maven 'maven'
    }
    stages {
        stage('Git Checkout') {
            steps {
               git branch: 'main', changelog: false, poll: false, url: 'https://github.com/sakshipalkar/Shopping-Cart.
            }
        }
        stage('Compile') {
            steps {
               sh "mvn clean compile"
            }
        }


        }

}
```

The stage view would look like this,

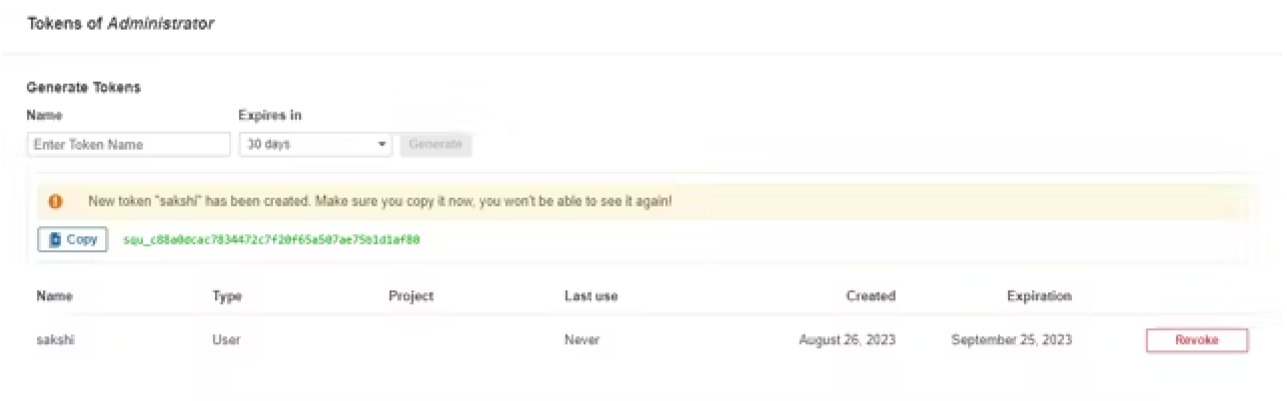


Step 4 — Configure Sonar Server in Manage Jenkins

Grab the Public IP Address of your EC2 Instance, Sonarqube works on Port 9000 , sp <Public IP>:9000. Goto your Sonarqube Server. Click on Administration → Security → Users → Click on Tokens and Update Token → Give it a name → and click on Generate Token



Click on Update Token


Tokens of *Administrator*

Generate Tokens

Name: Enter Token Name — Expires in: 30 days — Generate

New token "sakshi" has been created. Make sure you copy it now, you won't be able to see it again!

Copy squ_c88a0dcac7834472c7f20f65a507ae75b1d1af80

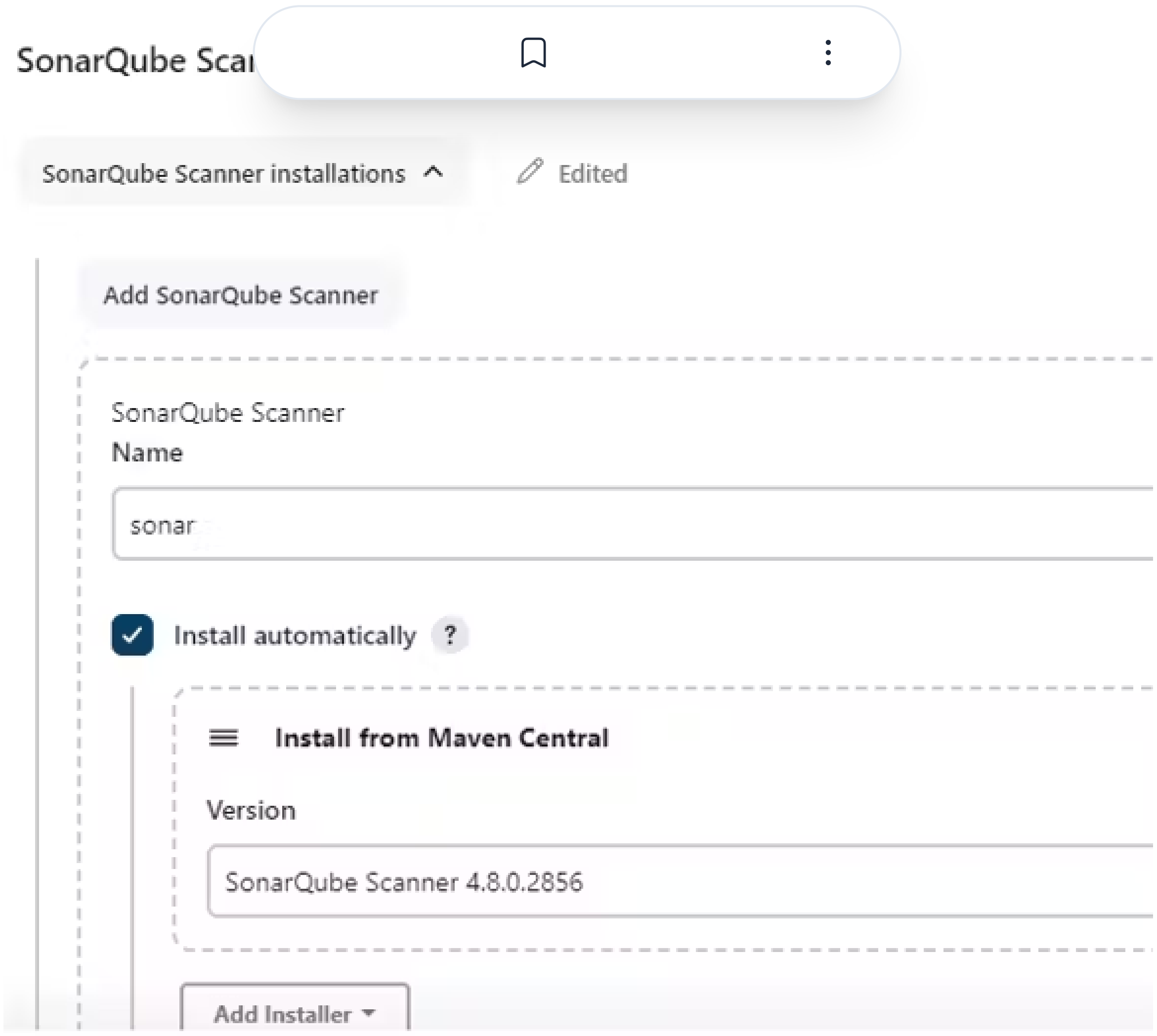
Name	Type	Project	Last use	Created	Expiration	
sakshi	User		Never	August 26, 2023	September 25, 2023	Revoke

Now, goto Dashboard → Manage Jenkins → Configure System

Configure System option is used in Jenkins to configure different server

Global Tool Configuration is used to configure different tools that we install using Plugins

We will install sonar-scanner in tools.


SonarQube Scan

SonarQube Scanner installations
Edited

Add SonarQube Scanner

SonarQube Scanner Name: sonar

Install automatically: ✓

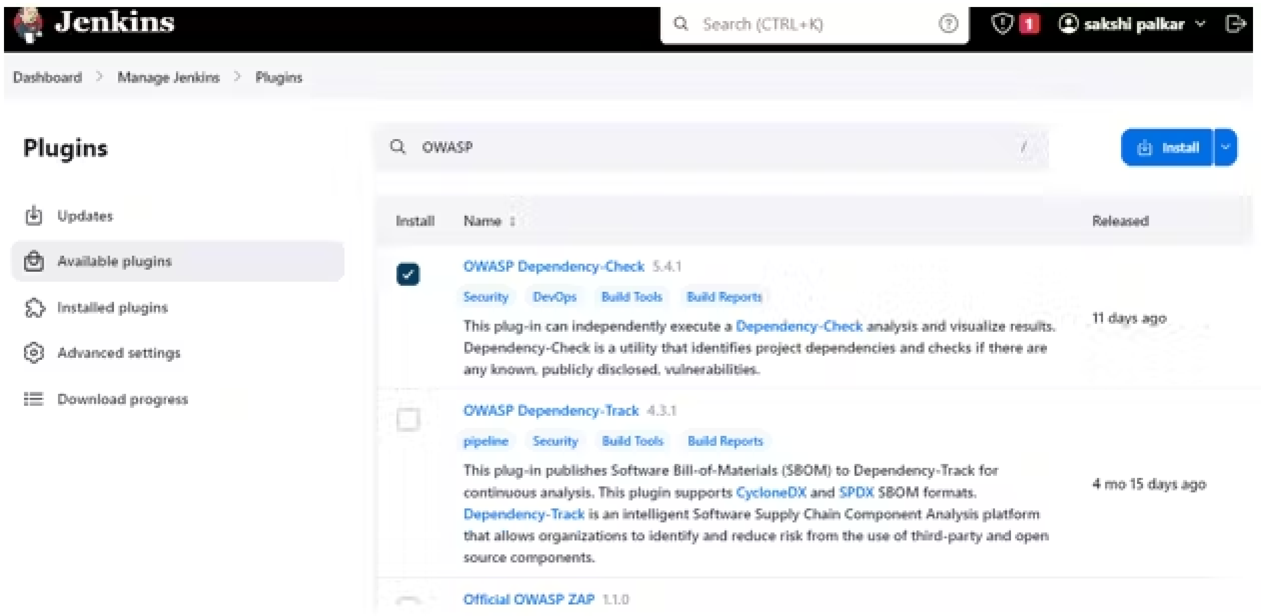
Install from Maven Central

Version: SonarQube Scanner 4.8.0.2856

Add Installer

Step 5 — Install OWASP Dependency Check Plugins

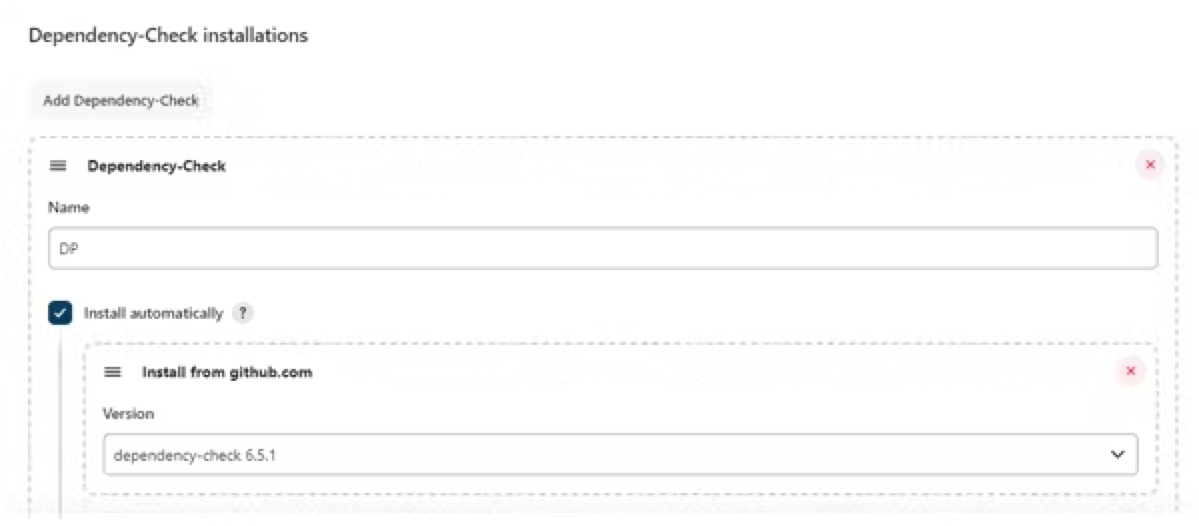
GotoDashboard → Manage Jenkins → Plugins → OWASP Dependency-Check. Click on it and install without restart.



Install	Name	Released
✓	OWASP Dependency-Check 5.4.1 Security DevOps Build Tools Build Reports This plug-in can independently execute a Dependency-Check analysis and visualize results. Dependency-Check is a utility that identifies project dependencies and checks if there are any known, publicly disclosed, vulnerabilities.	11 days ago
	OWASP Dependency-Track 4.3.1 pipeline Security Build Tools Build Reports This plug-in publishes Software Bill-of-Materials (SBOM) to Dependency-Track for continuous analysis. This plugin supports CycloneDX and SPDX SBOM formats. Dependency-Track is an intelligent Software Supply Chain Component Analysis platform that allows organizations to identify and reduce risk from the use of third-party and open source components.	4 mo 15 days ago
	Official OWASP ZAP 1.1.0	

First, we configured Pl [illegible] Tool

Goto Dashboard → Manage Jenkins → Tools →


Dependency-Check installations

Add Dependency-Check

Dependency-Check

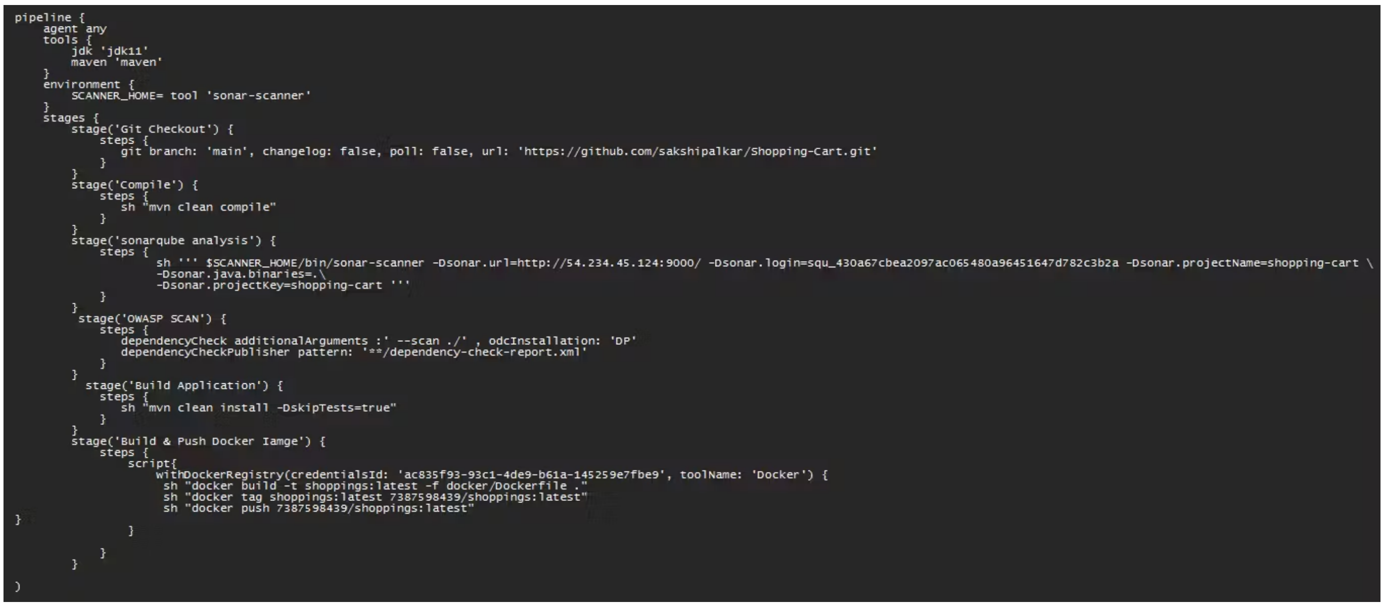
Name: DP

[✓] Install automatically

Install from github.com

Version: dependency-check 6.5.1

Now goto configure →The final pipeline would look like this,


```
pipeline {
    agent any
    tools {
        jdk 'jdk11'
        maven 'maven'
    }
    environment {
        SCANNER_HOME= tool 'sonar-scanner'
    }
    stages {
        stage('Git Checkout') {
            steps {
               git branch: 'main', changelog: false, poll: false, url: 'https://github.com/sakshipalkar/Shopping-Cart.git'
            }
        }
        stage('Compile') {
            steps {
               sh "mvn clean compile"
            }
        }
        stage('sonarqube analysis') {
            steps {
                   sh ''' $SCANNER_HOME/bin/sonar-scanner -Dsonar.url=http://54.234.45.124:9000/ -Dsonar.login=squ_430a67cbea2097ac065480a96451647d782c3b2a -Dsonar.projectName=shopping-cart \
                   -Dsonar.java.binaries=.\
                   -Dsonar.projectKey=shopping-cart '''
            }
        }
         stage('OWASP SCAN') {
            steps {
               dependencyCheck additionalArguments :' --scan ./' , odcInstallation: 'DP'
               dependencyCheckPublisher pattern: '**/dependency-check-report.xml'
            }
        }
          stage('Build Application') {
            steps {
               sh "mvn clean install -DskipTests=true"
            }
        }
        stage('Build & Push Docker Iamge') {
            steps {
                script{
                    withDockerRegistry(credentialsId: 'ac835f93-93c1-4de9-b61a-145259e7fbe9', toolName: 'Docker') {
                     sh "docker build -t shoppings:latest -f docker/Dockerfile ."
                     sh "docker tag shoppings:latest 7387598439/shoppings:latest"
                     sh "docker push 7387598439/shoppings:latest"
}
                }

            }
        }

)
```

Step 6 — Docker Image Build and Push

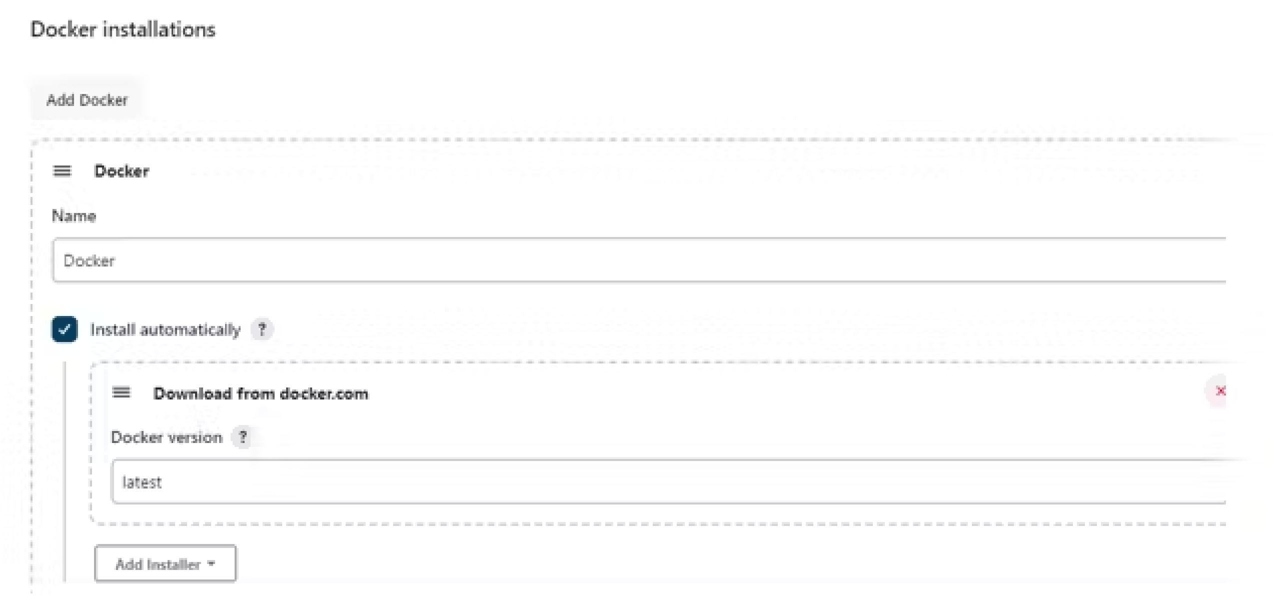
We need to install Docker tool in our system, Goto Dashboard → Manage Plugins → Available plugins → Search for Docker and install these plugins

- Docker
- Docker Commons

- Docker Pipeline
- Docker API
- docker-build-step

and click on install without restart

Now, goto Dashboard → Manage Jenkins → Tools →


Docker installations

Add Docker

Docker

Name: Docker

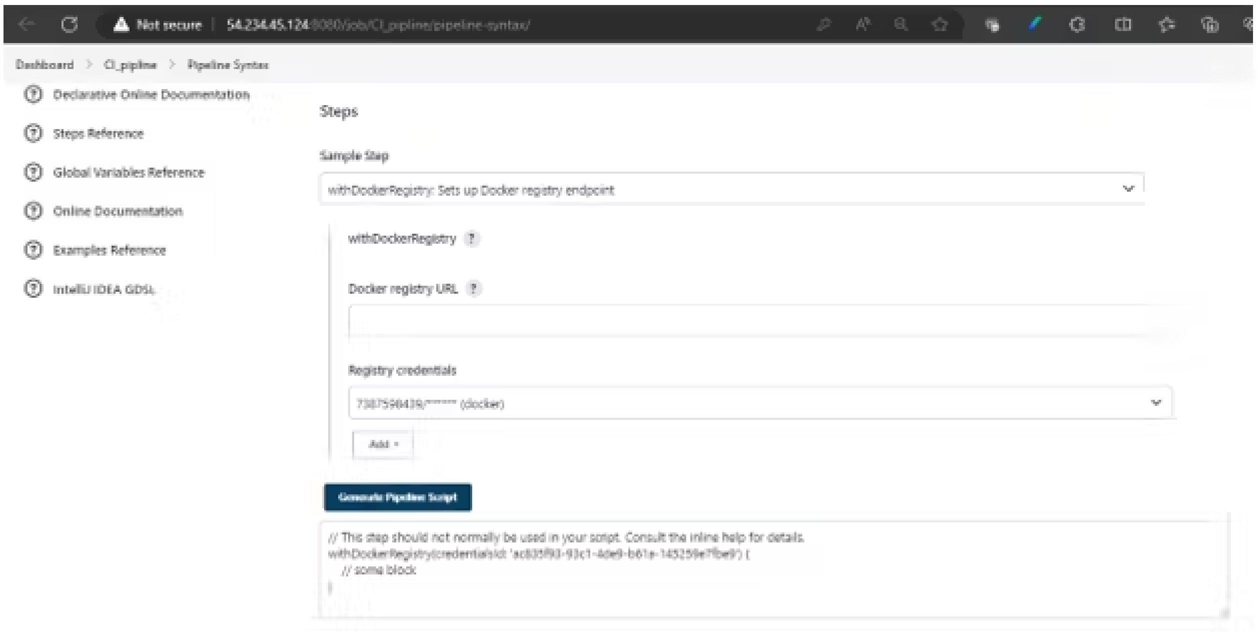
[✓] Install automatically

Download from docker.com

Docker version: latest

Add Installer

Add DockerHub Username and Password under Global Credentials


Not secure | 54.234.45.124:8080/job/CI_pipline/pipeline-syntax/

Dashboard > CI_pipline > Pipeline Syntax

- Declarative Online Documentation
- Steps Reference
- Global Variables Reference
- Online Documentation
- Examples Reference
- IntelliJ IDEA GDSL

Steps

Sample Step: withDockerRegistry: Sets up Docker registry endpoint

withDockerRegistry

Docker registry URL:

Registry credentials: 7387598439/****** (docker)

Add

Generate Pipeline Script

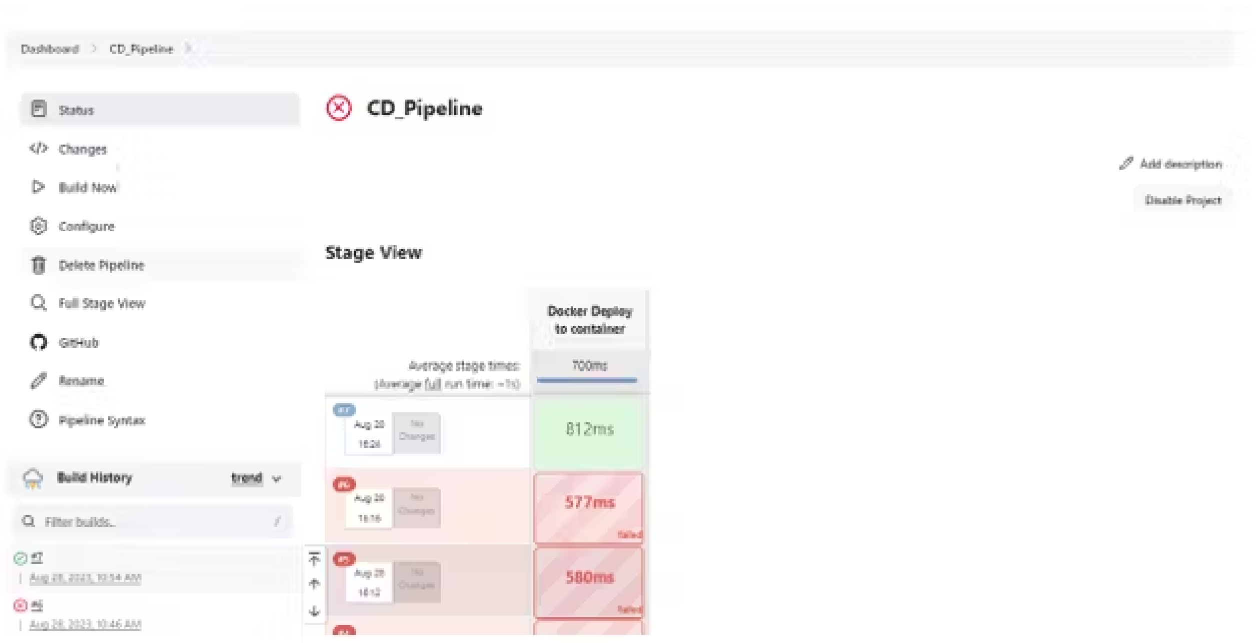
```
// This step should not normally be used in your script. Consult the inline help for details.
withDockerRegistry(credentialsId: 'ac835f93-93c1-4de9-b61a-145259e7fbe9') {
    // some block
}
```

Step 7 : Configure new [illegible] the image on docker hub.

Add this stage to your pipeline syntax

```
pipeline {
    agent any

    stages {
        stage('Docker Deploy to container') {
            steps {
              script{
                withDockerRegistry(credentialsId: 'ac835f93-93c1-4de9-b61a-145259e7fbe9', toolName: 'Docker') {
                  sh "docker run -d --name shopping-cart -p 8070:8070 7387598439/shoppings:latest"
}
            }
        }
    }
}
}
```

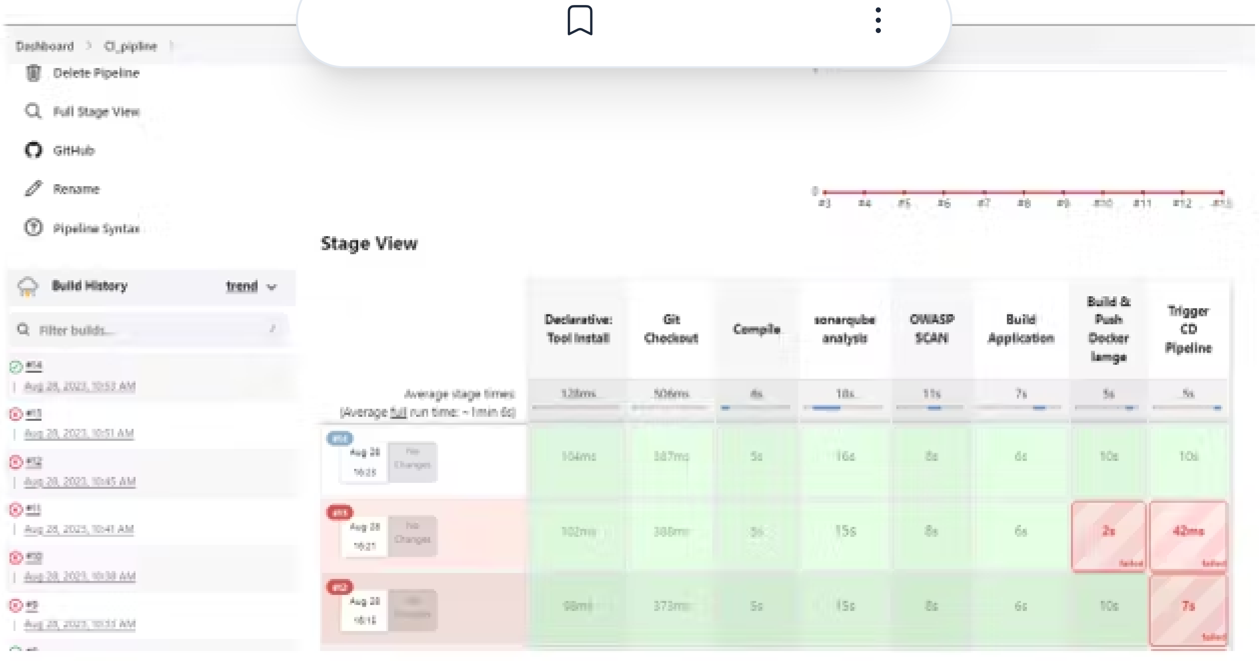
You will see the Stage View like this, for CD_pipeline.



Step 8: to triggered CD_pipeline automatically add below code in CI_pipeline.

```
stage('Trigger CD Pipeline') {
           steps {
               build job: "CD_Pipeline" , wait: true
           }
```

The final stage view would look like this.



Dashboard > CI_pipline >

- Delete Pipeline
- Full Stage View
- GitHub
- Rename
- Pipeline Syntax

Build History trend

Filter builds...

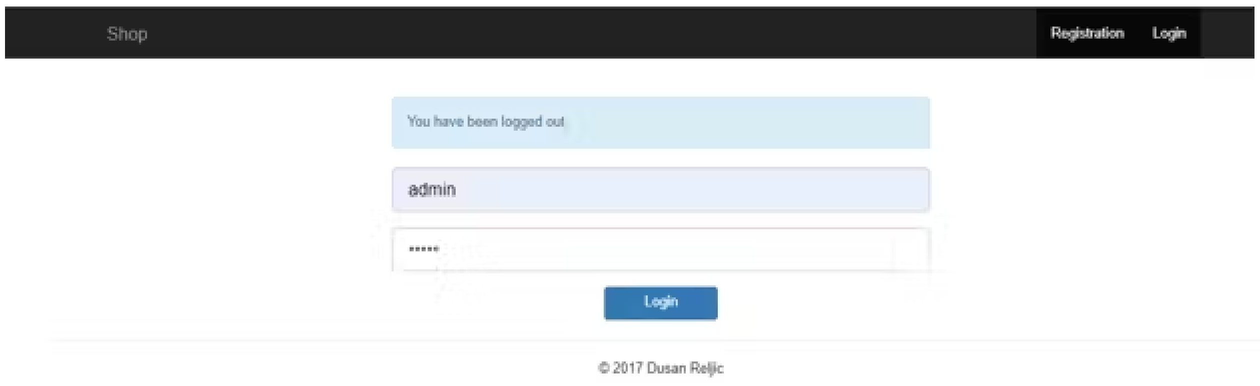
- #14 Aug 28, 2023, 10:53 AM
- #13 Aug 28, 2023, 10:51 AM
- #12 Aug 28, 2023, 10:45 AM
- #11 Aug 28, 2023, 10:41 AM
- #10 Aug 28, 2023, 10:38 AM
- #9 Aug 28, 2023, 10:33 AM

Stage View

	Declarative: Tool Install	Git Checkout	Compile	sonarqube analysis	OWASP SCAN	Build Application	Build & Push Docker Iamge	Trigger CD Pipeline
Average stage times: (Average full run time: ~1min 6s)	128ms	506ms	6s	18s	11s	7s	5s	5s
[illegible] Aug 28 16:23 No Changes	104ms	387ms	5s	16s	8s	6s	10s	10s
[illegible] Aug 28 16:21 No Changes	102ms	388ms	5s	15s	8s	6s	2s failed	42ms failed
[illegible] Aug 28 16:15 [illegible]	98ms	373ms	5s	15s	8s	6s	10s	7s failed

Step 9 :Access the Real World Application

And you can access your application on Port 8070. This is a Real World Application that has all Functional Tabs.



Shop Registration Login

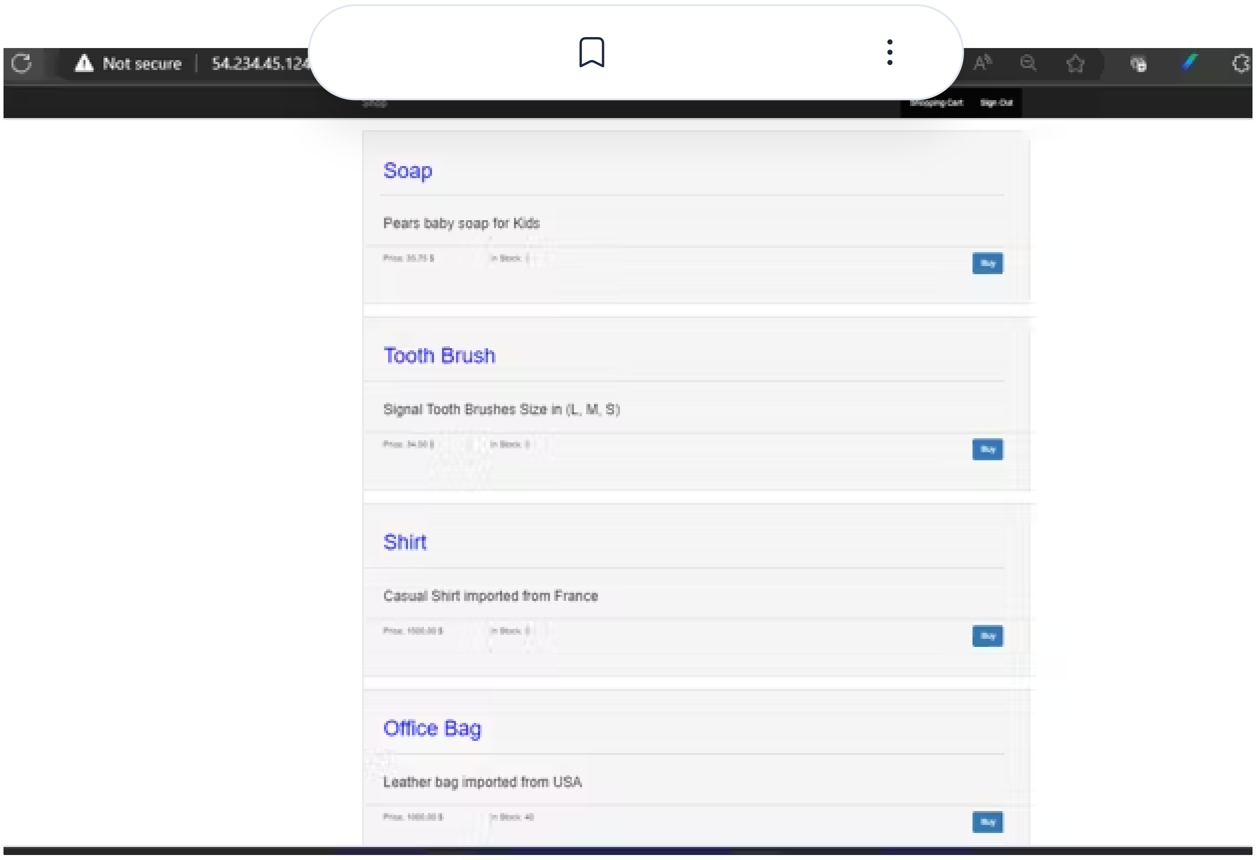
You have been logged out

admin

•••••

Login

© 2017 Dusan Reljic



Soap

Pears baby soap for Kids

Tooth Brush

Signal Tooth Brushes Size in (L, M, S)

Shirt

Casual Shirt imported from France

Office Bag

Leather bag imported from USA

Step 10 — Terminate the AWS EC2 Instance

Lastly, do not forget to terminate the AWS EC2 Instance.

Resources Used —https://www.youtube.com/watch?v=jbEF3mB7UcU&t=231s

Here is the GitHub Repo for this Project

https://github.com/sakshipalkar/Shopping-Cart

Hope you found this article useful and interesting. Thank you and Happy Learning 🙏

Subscribe to my newsletter

Read articles from dir… …to the newsletter, and

anoopkmathew4u@gmai SUBSCRIBE

Devops Jenkins #90daysofdevops #devopsproject