

Zomato Web Application – DevSecOps CI/CD Project, Docker, SonarQube & Trivy using Jenkins

1. Project Overview

This project demonstrates the implementation of a complete CI/CD pipeline using Jenkins, integrated with SonarQube for static code analysis, Trivy for security scanning, and Docker for containerization.

The pipeline automates code checkout, quality checks, vulnerability scanning, Docker image build, and application deployment.

2. Objectives

- Automate application build and deployment using Jenkins
 - Perform static code analysis using SonarQube
 - Perform file system and Docker image vulnerability scanning using Trivy
 - Build and deploy application using Docker containers
 - Implement a real-world DevOps CI/CD workflow

3. Infrastructure Setup

- Cloud Platform: AWS
 - EC2 Instance Type: t2.large
 - Operating System: Amazon Linux
 - Tools Installed:

- Jenkins

○ Git Installed

```
Last metadata expiration check: 0:00:17 ago on Fri Feb  6 05:29:45 2026.
Dependencies resolved.
=====
Package          Architecture Version      Repository  Size
=====
Installing:
git              x86_64      2.50.1-1.amzn2023.0.1    amazonlinux 53 k
Installing dependencies:
git-core          x86_64      2.50.1-1.amzn2023.0.1    amazonlinux 4.9 M
git-core-doc      noarch      2.50.1-1.amzn2023.0.1    amazonlinux 2.8 M
perl>Error        noarch      1:0.17029-5.amzn2023.0.2    amazonlinux 41 k
perl>File-Find     noarch      1.37-477.amzn2023.0.7   amazonlinux 25 k
perl>Git           noarch      2.50.1-1.amzn2023.0.1    amazonlinux 41 k
perl>TermReadKey   x86_64      2.38-9.amzn2023.0.2   amazonlinux 36 k
perl-lib          x86_64      0.65-477.amzn2023.0.7  amazonlinux 15 k
Transaction Summary
=====
Install 8 Packages

Total download size: 7.9 M
Installed size: 41 M
Downloading Packages:
(1/8): git-2.50.1-1.amzn2023.0.1.x86_64.rpm          1.3 MB/s | 53 kB  00:00
(2/8): perl>Error-0.17029-5.amzn2023.0.2.noarch.rpm    1.7 MB/s | 41 kB  00:00
(3/8): git-core-doc-2.50.1-1.amzn2023.0.1.noarch.rpm   34 MB/s | 2.8 MB  00:00
(4/8): git-core-2.50.1-1.amzn2023.0.1.x86_64.rpm       43 MB/s | 4.9 MB  00:00
(5/8): perl>File-Find-1.37-477.amzn2023.0.7.noarch.rpm 481 kB/s | 25 kB  00:00
(6/8): perl>Git-2.50.1-1.amzn2023.0.1.noarch.rpm       1.2 MB/s | 41 kB  00:00
(7/8): perl>TermReadKey-2.38-9.amzn2023.0.2.x86_64.rpm 1.4 MB/s | 36 kB  00:00
(8/8): perl-lib-0.65-477.amzn2023.0.7.x86_64.rpm      631 kB/s | 15 kB  00:00
Total                                         42 MB/s | 7.9 MB  00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing :                                                 1/1
  Installing : git-core-2.50.1-1.amzn2023.0.1.x86_64      1/8
  Installing : git-core-doc-2.50.1-1.amzn2023.0.1.noarch   2/8
  Installing : perl-lib-0.65-477.amzn2023.0.7.x86_64      3/8
  Installing : perl>TermReadKey-2.38-9.amzn2023.0.2.x86_64 4/8
  Installing : perl>File-Find-1.37-477.amzn2023.0.7.noarch 5/8
```

○ Docker

```
Complete!
Last metadata expiration check: 0:00:20 ago on Fri Feb  6 05:29:45 2026.
Dependencies resolved.
=====
Package          Architecture Version      Repository  Size
=====
Installing:
docker           x86_64      25.0.14-1.amzn2023.0.1    amazonlinux 46 M
Installing dependencies:
container-selinux  noarch      4:2.242.0-1.amzn2023      amazonlinux 58 k
containerd        x86_64      2.1.5-1.amzn2023.0.4      amazonlinux 23 M
iptables-libs    x86_64      1.8.8-3.amzn2023.0.2      amazonlinux 401 k
iptables-nft     x86_64      1.8.8-3.amzn2023.0.2      amazonlinux 183 k
libcgroup         x86_64      3.0-1.amzn2023.0.1      amazonlinux 75 k
libnetfilter_conntrack x86_64  1.0.8-2.amzn2023.0.2      amazonlinux 58 k
libnftnl          x86_64      1.0.1-19.amzn2023.0.2     amazonlinux 30 k
libnftnl          x86_64      1.2.2-2.amzn2023.0.2      amazonlinux 84 k
pigz              x86_64      2.5-1.amzn2023.0.3      amazonlinux 83 k
runc              x86_64      1.3.4-1.amzn2023.0.1      amazonlinux 3.9 M
Transaction Summary
=====
Install 11 Packages

Total download size: 74 M
Installed size: 281 M
Downloading Packages:
(1/11): container-selinux-2.242.0-1.amzn2023.noarch.rpm 1.5 MB/s | 58 kB  00:00
(2/11): iptables-libs-1.8.8-3.amzn2023.0.2.x86_64.rpm   8.5 MB/s | 401 kB  00:00
(3/11): iptables-nft-1.8.8-3.amzn2023.0.2.x86_64.rpm    4.3 MB/s | 183 kB  00:00
(4/11): libcgroup-3.0-1.amzn2023.0.1.x86_64.rpm        2.2 MB/s | 75 kB  00:00
(5/11): libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64.rpm 2.0 MB/s | 58 kB  00:00
(6/11): libnftnl-1.0.1-19.amzn2023.0.2.x86_64.rpm      1.2 MB/s | 30 kB  00:00
(7/11): libnftnl-1.2.2-2.amzn2023.0.2.x86_64.rpm      2.6 MB/s | 84 kB  00:00
(8/11): pigz-2.5-1.amzn2023.0.3.x86_64.rpm            3.3 MB/s | 83 kB  00:00
(9/11): containerd-2.1.5-1.amzn2023.0.4.x86_64.rpm    47 MB/s | 23 MB  00:00
(10/11): runc-1.3.4-1.amzn2023.0.1.x86_64.rpm        15 MB/s | 3.9 MB  00:00
(11/11): docker-25.0.14-1.amzn2023.0.1.x86_64.rpm    49 MB/s | 46 MB  00:00
Total                                         73 MB/s | 74 MB  00:01
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
```

○ Trivy

```
root@ip-172-31-0-124:~ % + ~
Last login: Fri Feb  6 05:22:38 2026 from 205.254.163.232
[ec2-user@ip-172-31-0-124 ~]$ trivy -v
-bash: trivy: command not found
[ec2-user@ip-172-31-0-124 ~]$ sudo yum install -y wget
wget https://github.com/aquasecurity/trivy/releases/latest/download/trivy_0.49.1_Linux-64bit.tar.gz
tar -xvf trivy_0.49.1_Linux-64bit.tar.gz
sudo mv trivy /usr/local/bin/
trivy --version
Last metadata expiration check: 0:44:23 ago on Fri Feb  6 05:35:02 2026.
Package wget-1.21.3-1.amzn2023.0.4.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
--2026-02-06 06:19:25-- https://github.com/aquasecurity/trivy/releases/latest/download/trivy_0.49.1_Linux-64bit.tar.gz
Resolving github.com (github.com)... 20.207.73.82
Connecting to github.com (github.com)|20.207.73.82|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://github.com/aquasecurity/trivy/releases/download/v0.69.1/trivy_0.49.1_Linux-64bit.tar.gz [following]
--2026-02-06 06:19:25-- https://github.com/aquasecurity/trivy/releases/download/v0.69.1/trivy_0.49.1_Linux-64bit.tar.gz
Reusing existing connection to github.com:443.
HTTP request sent, awaiting response... 404 Not Found
2026-02-06 06:19:26 ERROR 404: Not Found.

tar: trivy_0.49.1_Linux-64bit.tar.gz: Cannot open: No such file or directory
tar: Error is not recoverable: exiting now
mv: cannot stat 'trivy': No such file or directory
-bash: trivy: command not found
[ec2-user@ip-172-31-0-124 ~]$ trivy --version
-bash: trivy: command not found
[ec2-user@ip-172-31-0-124 ~]$ sudo tee /etc/yum.repos.d/trivy.repo <<'EOF'
[trivy]
name=Trivy repository
baseurl=https://aquasecurity.github.io/trivy-repo/rpm/releases/$releasever/$basearch/
gpgcheck=1
enabled=1
gpgkey=https://aquasecurity.github.io/trivy-repo/rpm/public.key
EOF

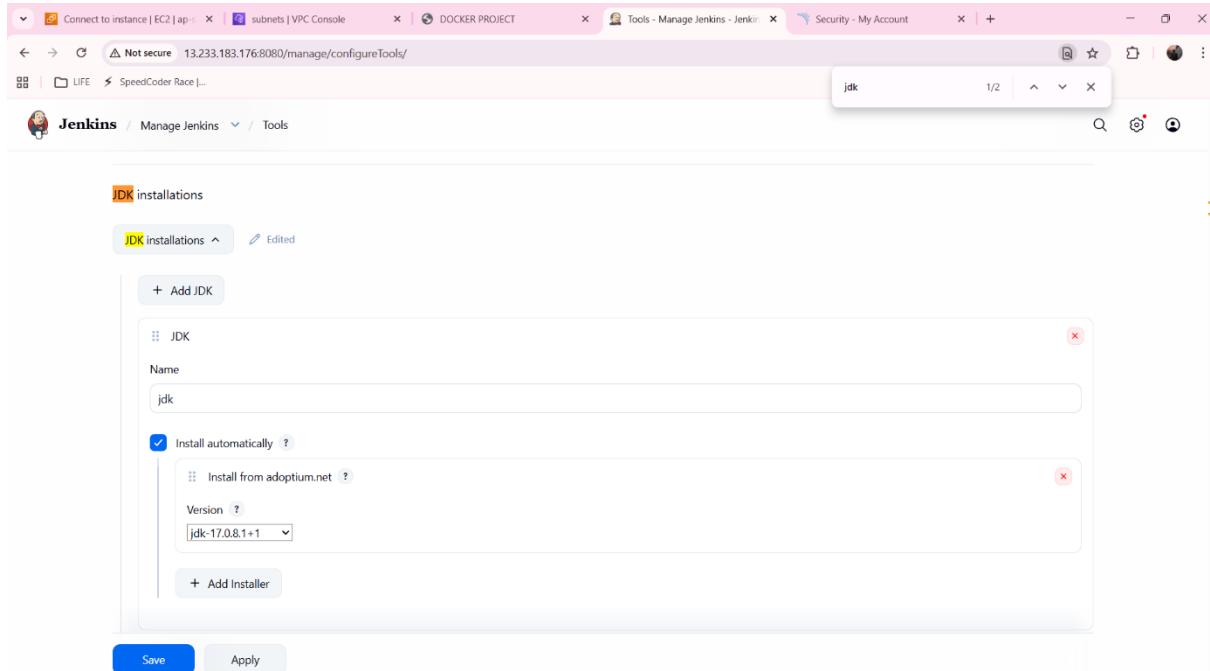
sudo yum install -y trivy
trivy--version
[trivy]
name=Trivy repository
baseurl=https://aquasecurity.github.io/trivy-repo/rpm/releases/$releasever/$basearch/
gpgcheck=1
```

4. Jenkins Installation

Jenkins is installed on the EC2 instance using the official Jenkins repository.

After installation, Jenkins service is started and enabled to run automatically on system boot.

Jenkins dashboard is accessed using port 8080 from the browser.



5. Git and Docker Installation

- **Git is installed to pull application source code from GitHub.**
- **Docker is installed and configured to build and run application containers.**
- **Docker service is enabled and started.**

```
root@ip-172-31-0-124: ~ + \ Complete!
Last metadata expiration check: 0:00:20 ago on Fri Feb  6 05:29:45 2026.
Dependencies resolved.
=====
      Package          Architecture      Version       Repository      Size
=====
Installing:
  docker           x86_64          25.0.14-1.amzn2023.0.1   amazonlinux    46 M
Installing dependencies:
  container-selinux      noarch        4.242.0-1.amzn2023
  containerd           x86_64          1.2.1.5-1.amzn2023.0.4   amazonlinux    58 k
  iptables-libs         x86_64          1.8.8-3.amzn2023.0.2    amazonlinux    23 M
  iptables-nft          x86_64          1.8.8-3.amzn2023.0.2    amazonlinux    401 k
  libcgroup            x86_64          3.0-1.amzn2023.0.1     amazonlinux    183 k
  libnetfilter_conntrack x86_64          1.0.8-2.amzn2023.0.2    amazonlinux    75 k
  libnftnl              x86_64          1.0.1-19.amzn2023.0.2   amazonlinux    58 k
  libnftnl              x86_64          1.2.2-2.amzn2023.0.2    amazonlinux    30 k
  pigz                 x86_64          2.5-1.amzn2023.0.3     amazonlinux    84 k
  runc                x86_64          1.3.4-1.amzn2023.0.1    amazonlinux    83 k
Transaction Summary
=====
Install 11 Packages
Total download size: 74 M
Installed size: 281 M
Downloading Packages:
(1/11): container-selinux-2.242.0-1.amzn2023.noarch.rpm      1.5 MB/s | 58 kB  00:00
(2/11): iptables-libs-1.8.8-3.amzn2023.0.2.x86_64.rpm      8.5 MB/s | 401 kB  00:00
(3/11): iptables-nft-1.8.8-3.amzn2023.0.2.x86_64.rpm      4.3 MB/s | 183 kB  00:00
(4/11): libcgroup-3.0-1.amzn2023.0.1.x86_64.rpm            2.2 MB/s | 75 kB  00:00
(5/11): libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64.rpm 2.0 MB/s | 58 kB  00:00
(6/11): libnftnl-1.2.2-2.amzn2023.0.2.x86_64.rpm            1.2 MB/s | 38 kB  00:00
(7/11): libnftnl-1.2.2-2.amzn2023.0.2.x86_64.rpm            2.6 MB/s | 84 kB  00:00
(8/11): pigz-2.5-1.amzn2023.0.3.x86_64.rpm                3.3 MB/s | 83 kB  00:00
(9/11): containerd-2.1.5-1.amzn2023.0.1.x86_64.rpm          47 MB/s | 23 MB  00:00
(10/11): runc-1.3.4-1.amzn2023.0.1.x86_64.rpm              15 MB/s | 3.9 MB  00:00
(11/11): docker-25.0.14-1.amzn2023.0.1.x86_64.rpm          49 MB/s | 46 MB  00:00
Total                                         73 MB/s | 74 MB  00:01
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.

root@ip-172-31-0-124: ~ + \ Last metadata expiration check: 0:00:17 ago on Fri Feb  6 05:29:45 2026.
Dependencies resolved.
=====
      Package          Architecture      Version       Repository      Size
=====
Installing:
  git               x86_64          2.50.1-1.amzn2023.0.1   amazonlinux    53 k
Installing dependencies:
  git-core           x86_64          2.50.1-1.amzn2023.0.1   amazonlinux    4.9 M
  git-core-doc        noarch        2.50.1-1.amzn2023.0.1   amazonlinux    2.8 M
  perl-Error          noarch        1:0.17029-5.amzn2023.0.2   amazonlinux    41 k
  perl-File-Find       noarch        1.37-477.amzn2023.0.7    amazonlinux    25 k
  perl-Git             noarch        2.50.1-1.amzn2023.0.1   amazonlinux    41 k
  perl-TermReadKey     x86_64          2.38-9.amzn2023.0.2    amazonlinux    36 k
  perl-lib             x86_64          0.65-477.amzn2023.0.7    amazonlinux    15 k
Transaction Summary
=====
Install 8 Packages
Total download size: 7.9 M
Installed size: 41 M
Downloading Packages:
(1/8): git-2.50.1-1.amzn2023.0.1.x86_64.rpm      1.3 MB/s | 53 kB  00:00
(2/8): perl-Error-0.17029-5.amzn2023.0.2.noarch.rpm 1.7 MB/s | 41 kB  00:00
(3/8): git-core-doc-2.50.1-1.amzn2023.0.1.noarch.rpm 34 MB/s | 2.8 MB  00:00
(4/8): git-core-2.50.1-1.amzn2023.0.1.x86_64.rpm    43 MB/s | 4.9 MB  00:00
(5/8): perl-File-Find-1.37-477.amzn2023.0.7.noarch.rpm 481 kB/s | 25 kB  00:00
(6/8): perl-Git-2.50.1-1.amzn2023.0.1.noarch.rpm   1.2 MB/s | 41 kB  00:00
(7/8): perl-TermReadKey-2.38-9.amzn2023.0.2.x86_64.rpm 1.4 MB/s | 36 kB  00:00
(8/8): perl-lib-0.65-477.amzn2023.0.7.x86_64.rpm   631 kB/s | 15 kB  00:00
Total                                         42 MB/s | 7.9 MB  00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing   :
  Installing  : git-core-2.50.1-1.amzn2023.0.1.x86_64        1/1
  Installing  : git-core-doc-2.50.1-1.amzn2023.0.1.noarch      1/8
  Installing  : perl-lib-0.65-477.amzn2023.0.7.x86_64        2/8
  Installing  : perl-TermReadKey-2.38-9.amzn2023.0.2.x86_64  3/8
  Installing  : perl-File-Find-1.37-477.amzn2023.0.7.noarch  4/8
  Installing  : perl-Git-2.50.1-1.amzn2023.0.1.noarch        5/8
```

6. Trivy Installation

Trivy is installed manually by downloading the binary from the official GitHub repository. After extraction, the Trivy binary is moved to /usr/local/bin and added to the system PATH to allow global usage.

```

root@ip-172-31-0-124:~ + ~
Last login: Fri Feb  6 05:22:38 2026 from 205.254.163.232
[ec2-user@ip-172-31-0-124 ~]$ trivy -v
-bash: trivy: command not found
[ec2-user@ip-172-31-0-124 ~]$ sudo yum install -y wget
wget https://github.com/aquasecurity/trivy/releases/latest/download/trivy_0.49.1_Linux-64bit.tar.gz
tar -xvf trivy_0.49.1_Linux-64bit.tar.gz
sudo mv trivy /usr/local/bin/
trivy --version
Last metadata expiration check: 0:44:23 ago on Fri Feb  6 05:35:02 2026.
Package wget-1.21.3-1.amzn2023.0.4.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
--2026-02-06 06:19:25-- https://github.com/aquasecurity/trivy/releases/latest/download/trivy_0.49.1_Linux-64bit.tar.gz
Resolving github.com (github.com)... 20.207.73.82
Connecting to github.com (github.com)|20.207.73.82|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://github.com/aquasecurity/trivy/releases/download/v0.69.1/trivy_0.49.1_Linux-64bit.tar.gz [following]
--2026-02-06 06:19:25-- https://github.com/aquasecurity/trivy/releases/download/v0.69.1/trivy_0.49.1_Linux-64bit.tar.gz
Reusing existing connection to github.com:443.
HTTP request sent, awaiting response... 404 Not Found
2026-02-06 06:19:26 ERROR 404: Not Found.

tar: trivy_0.49.1_Linux-64bit.tar.gz: Cannot open: No such file or directory
tar: Error is not recoverable: exiting now
mv: cannot stat 'trivy': No such file or directory
-bash: trivy: command not found
[ec2-user@ip-172-31-0-124 ~]$ trivy --version
-bash: trivy: command not found
[ec2-user@ip-172-31-0-124 ~]$ sudo tee /etc/yum.repos.d/trivy.repo <<'EOF'
[trivy]
name=Trivy repository
baseurl=https://aquasecurity.github.io/trivy-repo/rpm/releases/$releasever/$basearch/
gpgcheck=1
enabled=1
gpgkey=https://aquasecurity.github.io/trivy-repo/rpm/public.key
EOF

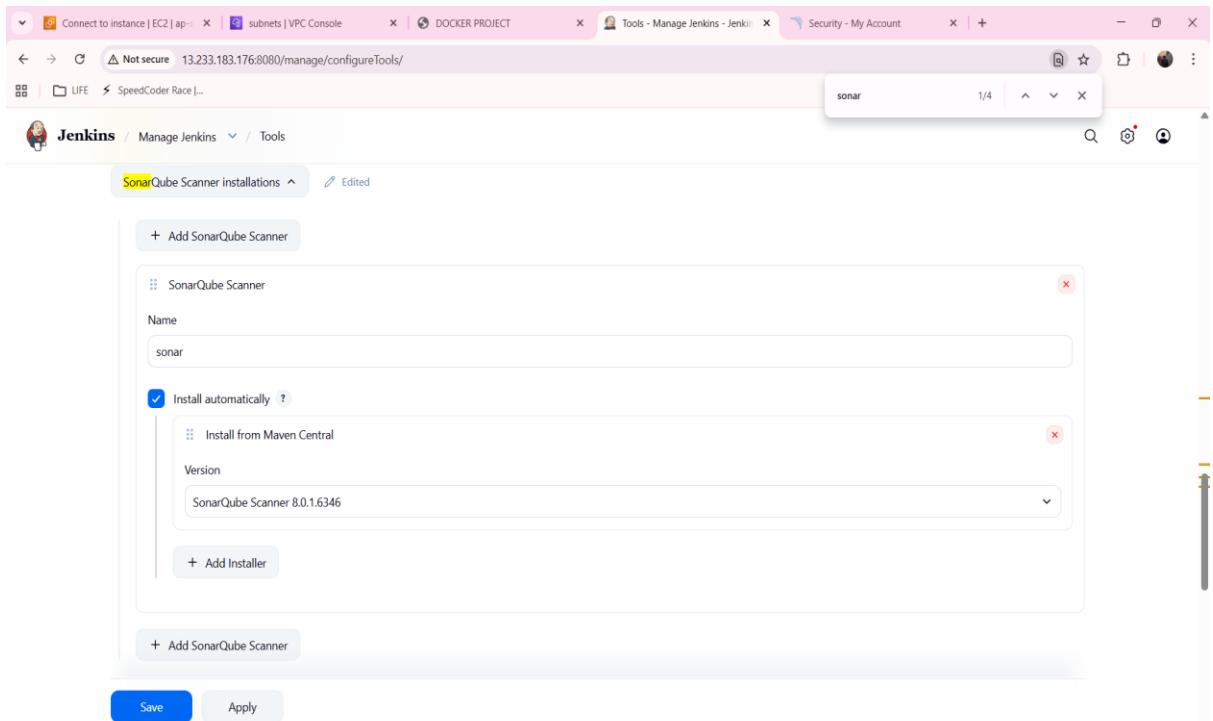
sudo yum install -y trivy
trivy --version
[trivy]
name=Trivy repository
baseurl=https://aquasecurity.github.io/trivy-repo/rpm/releases/$releasever/$basearch/
gpgcheck=1

```

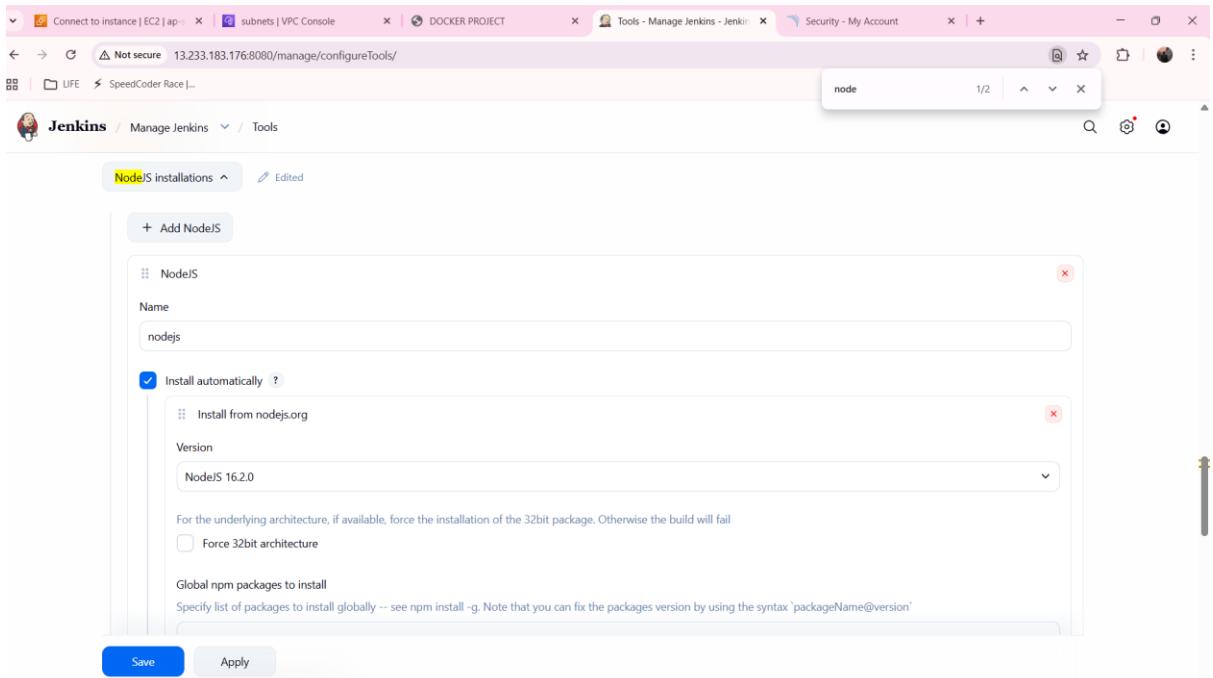
7. Jenkins Plugin Configuration

The following Jenkins plugins are installed to support the pipeline:

- SonarQube Scanner

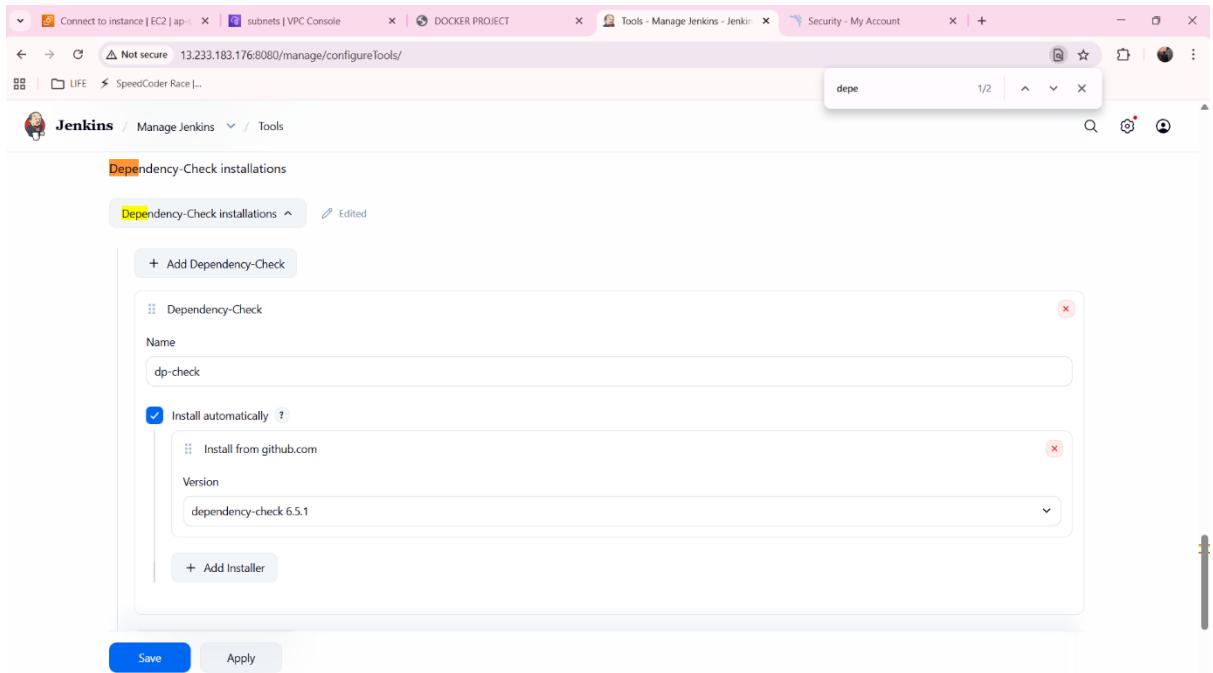


- **NodeJS**



The screenshot shows the Jenkins 'Tools' configuration page for 'NodeJS installations'. A new installation is being created with the name 'nodejs'. The 'Install automatically' checkbox is checked, and the source is set to 'Install from nodejs.org'. The version is specified as 'NodeJS 16.2.0'. There is an option to force a 32-bit architecture, which is currently unchecked. Global npm packages to install are listed as 'dependency-check@6.5.1'. The 'Save' button is highlighted in blue.

- **OWASP Dependency-Check**



The screenshot shows the Jenkins 'Tools' configuration page for 'Dependency-Check installations'. A new installation is being created with the name 'dp-check'. The 'Install automatically' checkbox is checked, and the source is set to 'Install from github.com'. The version is specified as 'dependency-check@6.5.1'. The 'Save' button is highlighted in blue.

- **Docker Pipeline**

```
pipeline {
```

```
    agent any
```

```
    tools {
```

```
jdk 'jdk'
nodejs 'nodejs'

}

environment {
    SCANNER_HOME = tool 'sonar'
}

stages {

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

    stage('Checkout Code') {
        steps {
            git 'https://github.com/sonugupta4166/zomato-project1.git'
        }
    }

    stage('SonarQube Analysis') {
        steps {
            withSonarQubeEnv('sonar') {
                sh """
${SCANNER_HOME}/bin/sonar-scanner \
-Dsonar.projectName=zomato \
-Dsonar.projectKey=zomato
"""
            }
        }
    }
}
```

```
        }
    }

stage('Install Dependencies') {
    steps {
        sh 'npm install'
    }
}

stage('Dependency Check') {
    steps {
        dependencyCheck additionalArguments: '--scan ./ --disableYarnAudit -- disableNodeAudit',
        odcInstallation: 'dp-check'
        dependencyCheckPublisher pattern: '**/dependency-check-report.xml'
    }
}

stage('Docker Build') {
    steps {
        sh 'docker build -t container436/zomato:v1 .'
    }
}

stage('Trivy FS Scan') {
    steps {
        sh 'trivy fs . --severity HIGH,CRITICAL --exit-code 0 > trivyfs.txt'
    }
}

stage('Trivy Image Scan') {
```

```
steps {
    sh 'trivy image container436/zomato:v1 --severity HIGH,CRITICAL'
}

}

stage('Docker Push') {
    steps {
        script {
            withDockerRegistry(credentialsId: 'ad4fcd89-3746-4c33-9450-47e0078295fc') {
                sh 'docker push container436/zomato:v1'
            }
        }
    }
}

stage('Deploy Container') {
    steps {
        sh """
        docker rm -f c1 || true
        docker run -d --name c1 -p 3000:3000 container436/zomato:v1
        """
    }
}
```

The screenshot shows the Jenkins Pipeline configuration page for a job named "Zomato". The "Pipeline" tab is selected in the sidebar. The main area contains a Groovy script block:

```
15      cleanWs()
16    }
17  }
18  stage('code') {
19    steps {
20      git 'https://github.com/sonugupta4166/zomato-project1.git'
21    }
22  }
23  stage('QAT') {
24    steps{
25      withSonarQubeEnv('sonar') {
26        sh """$SCANNER_HOME/bin/sonar-scanner \
27          -Dsonar.projectName=zomato \
28          -Dsonar.projectKey=zomato"""
29    }
30  }
```

Below the script, there is a checkbox labeled "Use Groovy Sandbox" with a question mark icon.

The screenshot shows the Jenkins Pipeline configuration page for the same "Zomato" job. The "Pipeline" tab is selected. The main area contains a Groovy script block:

```
61  script {
62    withDockerRegistry(credentialsId: 'ad4fcdb89-3746-4c33-9450-47e0078295fc') {
63      sh 'docker push container436/zomato:v1'
64    }
65  }
66  }
67  }
68  }
69  }
70  }
71  stage("deploy-container"){
72    steps{
73      sh 'docker run -itd --name c1 -p 3000:3000 container436/zomato:v1'
74    }
75  }
```

Below the script, there is a checkbox labeled "Use Groovy Sandbox" with a question mark icon.

The screenshot shows two instances of the Jenkins Pipeline configuration page. Both instances have the 'Pipeline' section selected in the left sidebar.

Script Example 1:

```

15     cleanWS()
16   }
17 }
18 stage('code') {
19 steps {
20   git 'https://github.com/sonugupta4166/zomato-project1.git'
21 }
22 }
23 }
24 stage('QAT') {
25 steps{
26 withSonarQubeEnv('sonar') {
27   sh """$SCANNER_HOME/bin/sonar-scanner \
28 -Dsonar.projectName=zomato \
29 -Dsonar.projectKey=zomato"""

```

Script Example 2:

```

1 pipeline {
2   agent any
3   tools{
4     jdk "jdk"
5     nodejs "nodejs"
6   }
7 environment{
8   SCANNER_HOME = tool "sonar"
9 }
10 }
11 stages {
12   stage('cleanWS') {
13     steps {
14       sh "rm -rf ./.scannerXXXXXX"
15     }
16   }
17 }
18 }
19 
```

Both screenshots include a 'Use Groovy Sandbox' checkbox and a 'Pipeline Syntax' link. Below the script editor, there are 'Save' and 'Apply' buttons.

- **Eclipse Temurin Installer**

Each plugin is installed and configured from Manage Jenkins → Plugins.

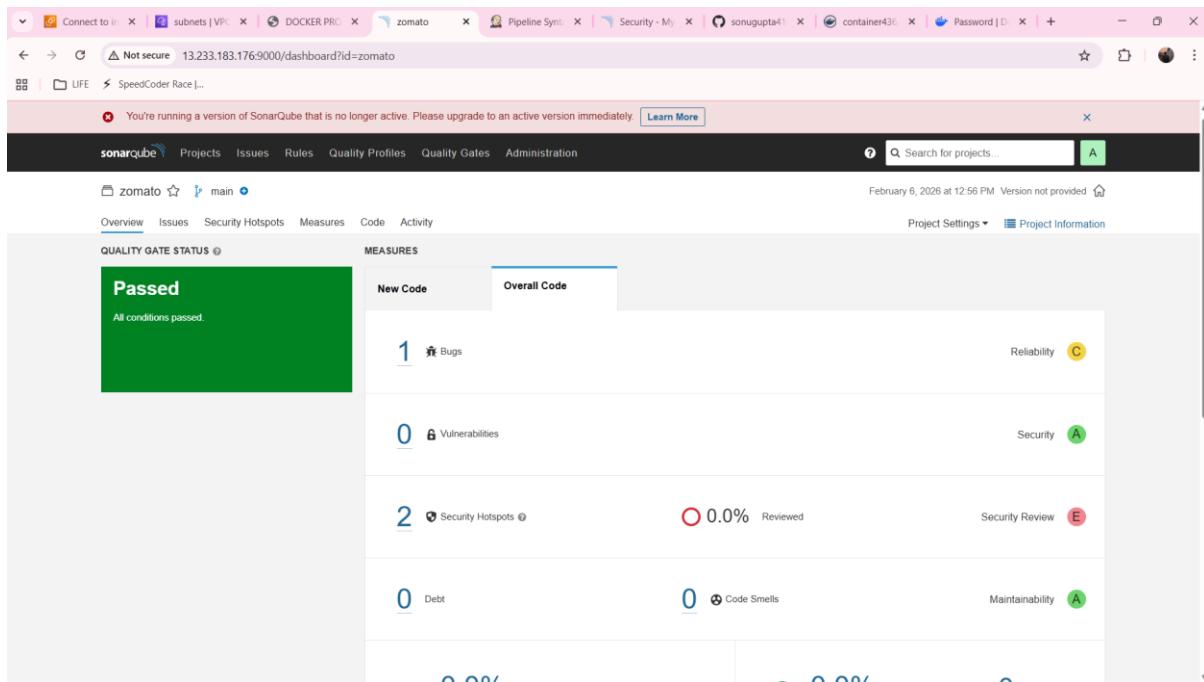
8. SonarQube Setup Using Docker

SonarQube is deployed as a Docker container using the official SonarQube LTS image. The container exposes port 9000, which is used to access the SonarQube dashboard.

`docker run -d \`

`--name sonarqube \`

```
-p 9000:9000 \
-v sonarqube_data:/opt/sonarqube/data \
-v sonarqube_logs:/opt/sonarqube/logs \
-v sonarqube_extensions:/opt/sonarqube/extensions \
sonarqube:its-community
```



Initial login is performed using default credentials, and a new password is set.

9. SonarQube Token Generation

A SonarQube authentication token is generated from the SonarQube dashboard under user security settings.

This token is later used to authenticate Jenkins with SonarQube.

10. Jenkins Credentials Configuration

The SonarQube token is added to Jenkins as a Secret Text credential.

This allows Jenkins to securely communicate with the SonarQube server during pipeline execution.

11. SonarQube Server Configuration in Jenkins

SonarQube server details are configured under Manage Jenkins → System. The server name and authentication token are mapped for pipeline usage.

12. Tool Configuration in Jenkins

The following tools are configured under Manage Jenkins → Tools:

- **JDK (Java 17)**

This screenshot shows the 'JDK installations' configuration page in Jenkins. A search bar at the top right contains the text 'jdk'. The main form is titled 'JDK installations' and shows one entry named 'jdk'. The entry includes fields for 'Name' (set to 'jdk'), 'Install automatically?' (checked), 'Install from adoptium.net?' (unchecked), 'Version?' (set to 'jdk-17.0.8.1+1'), and a 'Save' button.

- **NodeJS**

This screenshot shows the 'NodeJS installations' configuration page in Jenkins. A search bar at the top right contains the text 'node'. The main form is titled 'NodeJS installations' and shows one entry named 'nodejs'. The entry includes fields for 'Name' (set to 'nodejs'), 'Install automatically?' (checked), 'Install from nodejs.org?' (checked), 'Version' (set to 'NodeJS 16.2.0'), and a checkbox for 'Force 32bit architecture' (unchecked). There is also a section for 'Global npm packages to install' with a note about specifying package names and versions. Buttons for 'Save' and 'Apply' are at the bottom.

- **SonarQube Scanner**

The screenshot shows the Jenkins interface for managing tools. The URL is `13.233.183.176:8080/manage/configureTools/`. The page title is "Jenkins / Manage Jenkins / Tools". Under "SonarQube Scanner installations", there is one entry named "sonar". The "Name" field contains "sonar". The "Install automatically" checkbox is checked. Under "Install from Maven Central", the "Version" is set to "SonarQube Scanner 8.0.1.6346". There are "Save" and "Apply" buttons at the bottom.

- **OWASP Dependency-Check**

The screenshot shows the Jenkins interface for managing tools. The URL is `13.233.183.176:8080/manage/configureTools/`. The page title is "Jenkins / Manage Jenkins / Tools". Under "Dependency-Check installations", there is one entry named "dp-check". The "Name" field contains "dp-check". The "Install automatically" checkbox is checked. Under "Install from github.com", the "Version" is set to "dependency-check 6.5.1". There are "Save" and "Apply" buttons at the bottom.

Automatic installation is enabled for all tools.

13. Quality Gate Configuration

A Quality Gate is configured in SonarQube to enforce minimum code quality standards.
A webhook is created to notify Jenkins about analysis results.

14. Jenkins Declarative Pipeline

Screenshot of Jenkins Pipeline configuration for a 'Zomato' job.

The pipeline script is defined as follows:

```
script {
    withDockerRegistry(credentialsId: 'ad4fcdb9-3746-4c33-9450-47e0078295fc') {
        sh 'docker push container436/zomato:v1'
    }
}
stage('deploy-container'){
    steps{
        sh 'docker run -itd --name c1 -p 3000:3000 container436/zomato:v1'
    }
}
```

The 'Pipeline Syntax' section shows the Groovy code for the pipeline.

Advanced settings are visible at the bottom of the configuration page.

Screenshot of Jenkins Pipeline configuration for a 'Zomato' job.

The pipeline script is defined as follows:

```
cleanWs()
stage('code') {
    steps {
        git 'https://github.com/sonugupta4166/zomato-project1.git'
    }
}
stage('QAT') {
    steps{
        withSonarQubeEnv('sonar') {
            sh """$SCANNER_HOME/bin/sonar-scanner \
                -sonar.projectName=zomato \
                -Sonar.projectKey=zomato"""
        }
    }
}
```

The 'Pipeline Syntax' section shows the Groovy code for the pipeline.

Advanced settings are visible at the bottom of the configuration page.

The screenshot shows the Jenkins Pipeline configuration page for a job named 'Zomato'. The left sidebar has tabs for General, Triggers, Pipeline (which is selected), and Advanced. The main area contains a 'Script' editor with the following Groovy code:

```
1< pipeline {  
2     agent any  
3     tools{  
4         jdk "jdk"  
5         nodejs "nodejs"  
6     }  
7     environment{  
8         SCANNER_HOME = tool "sonar"  
9     }  
10    stages {  
11        stage('cleanWS') {  
12            steps {  
13                echo "Cleaning workspace..."  
14            }  
15        }  
16    }  
17}
```

Below the script editor is a checkbox for 'Use Groovy Sandbox' which is checked. At the bottom are 'Save' and 'Apply' buttons.

A Declarative Jenkins Pipeline is written to automate the complete workflow. The pipeline includes multiple stages to perform different DevOps tasks.

15. Workspace Cleanup Stage

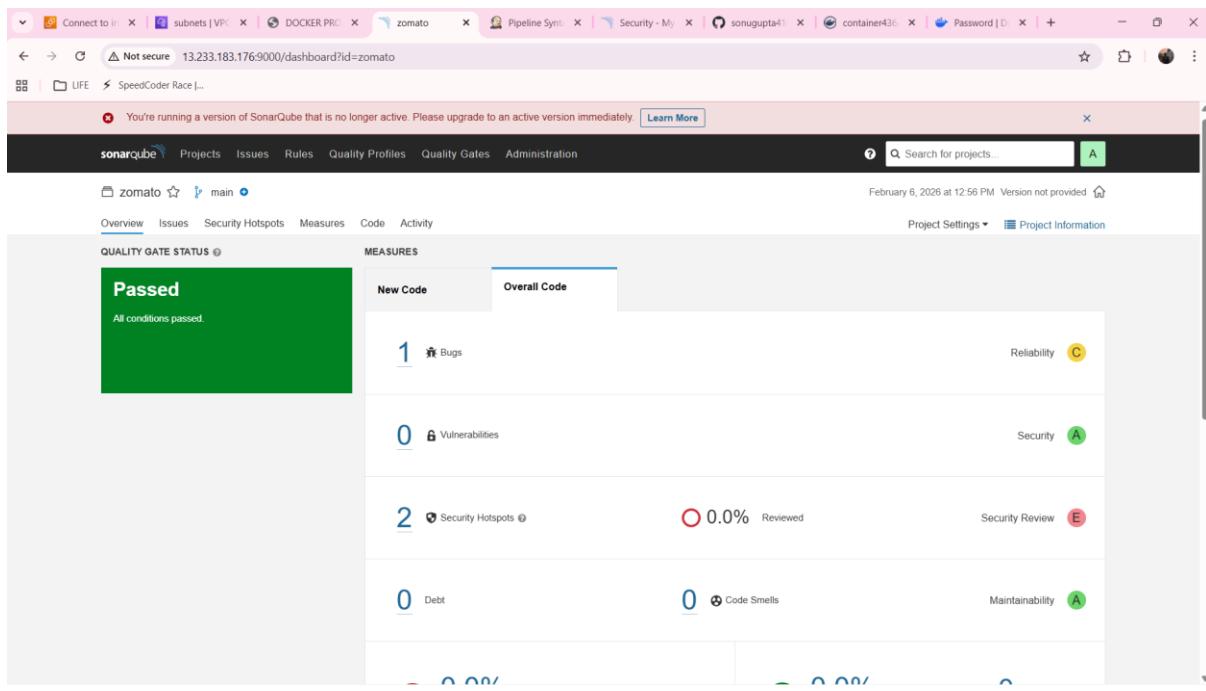
The pipeline starts by cleaning the Jenkins workspace to avoid conflicts from previous builds.

16. Source Code Checkout Stage

Source code is cloned from the GitHub repository using Git. This ensures the latest version of the application is used for the pipeline.

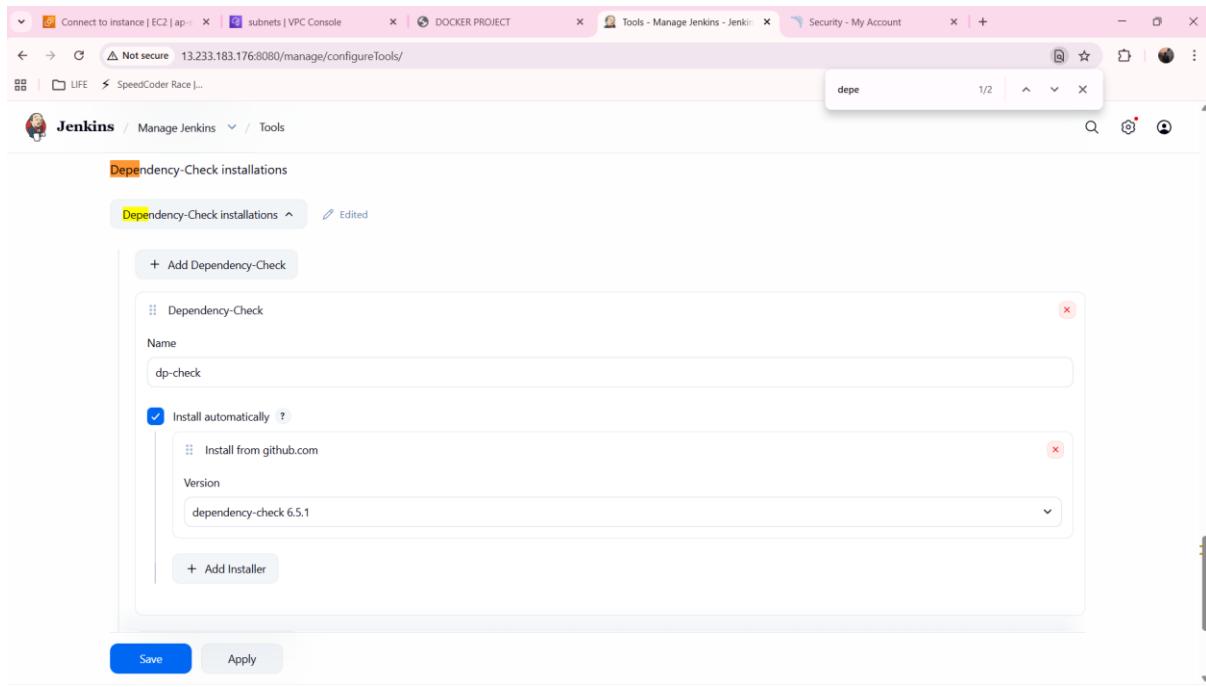
17. SonarQube Code Analysis Stage

SonarQube scanner is executed to perform static code analysis. The analysis checks code quality, bugs, and security issues and publishes results to the SonarQube dashboard.



18. Dependency Vulnerability Scan Stage

OWASP Dependency-Check is used to scan project dependencies.
A vulnerability report is generated and published in Jenkins.



19. Docker Image Build Stage

Docker is used to build an application image from the Dockerfile.
The image is tagged with a version number for identification.

20. Trivy File System Scan Stage

Trivy is used to scan the project file system for vulnerabilities before image creation.
The scan report is stored for security review.

21. Trivy Docker Image Scan Stage

The built Docker image is scanned using Trivy to identify OS and library vulnerabilities.
This ensures the container image is secure before deployment.

22. Docker Image Push Stage

The Docker image is pushed to Docker Hub using secure credentials stored in Jenkins.
This allows the image to be reused or deployed on other systems.

23. Application Deployment Stage

The Docker container is deployed using the pushed image.
The application is exposed on the required port and verified through browser access.

24. Verification

- **Jenkins pipeline execution is verified**
 - **SonarQube analysis results are reviewed**
 - **Trivy vulnerability reports are checked**
 - **Application accessibility is confirmed**
-

25. Key Learnings

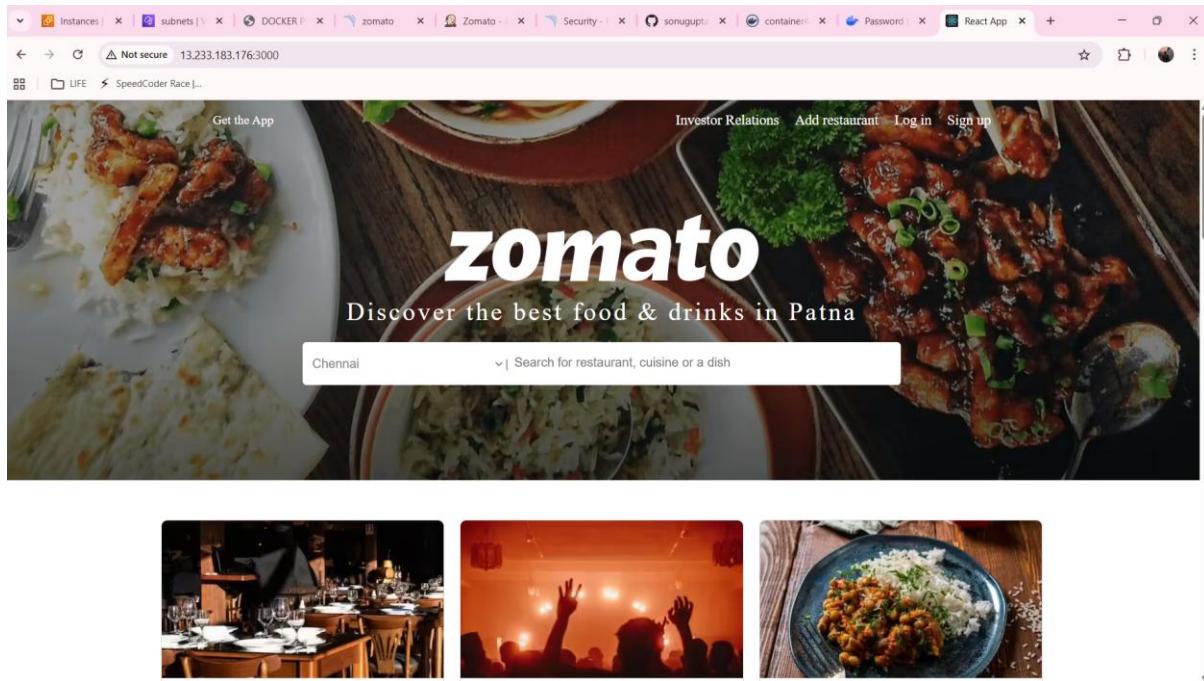
- **CI/CD automation using Jenkins**
 - **Static code analysis with SonarQube**
 - **Container security scanning using Trivy**
 - **Docker-based application deployment**
 - **Real-world DevOps pipeline implementation**
-

26. Conclusion

This project demonstrates a real-time DevOps CI/CD pipeline integrating code quality, security, and containerization.

It reflects industry-standard DevOps practices and provides strong hands-on experience in automation and cloud-native deployments.

Deployed Application



A screenshot of the Zomato footer. It includes links for 'ABOUT ZOMATO', 'ZOMAVERSE', 'FOR RESTAURANTS', 'LEARN MORE', and 'SOCIAL LINKS'. There are also links for 'Popular restaurant types near me' and 'Top Restaurant Chains'. The footer is pink at the bottom and contains a small legal notice about terms of service and trademarks.

Instances | subnets | DOCKER | zomato | Zomato | Security | sonugupt | container | Password | React App | + | - | X

← → ⌂ Not secure 13.233.183.176:3000

LIFE SpeedCoder Race |...

Navrangpura > 302 Places

Vastrapur > 217 Places

Thaltej > 222 Places

Prahald Nagar > 181 Places

C G Road > 94 Places

See more ▾

Get the Zomato app

We will send you a link, open it on your phone to download the app

Email Phone

Email Share App Link

Download app from

 GET IT ON Google Play  Downloaded on the App Store

Instances | subnets | DOCKER | zomato | Zomato | Security | sonugupt | container | Password | React App | + | - | X

← → ⌂ Not secure 13.233.183.176:3000

LIFE SpeedCoder Race |...

Collection

Explore curated lists of top restaurants, cafes, pubs, and bars in Ahmedabad, based on trends

All collection in Ahmedabad ▾



Popular localities in and around Ahmedabad

Bodakdev
345 Places

Setellite
336 Places

Gurukul
83 Places