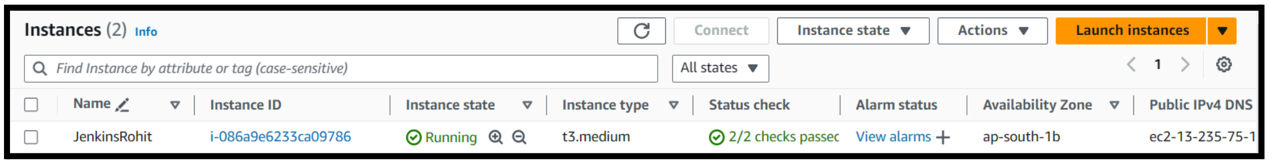
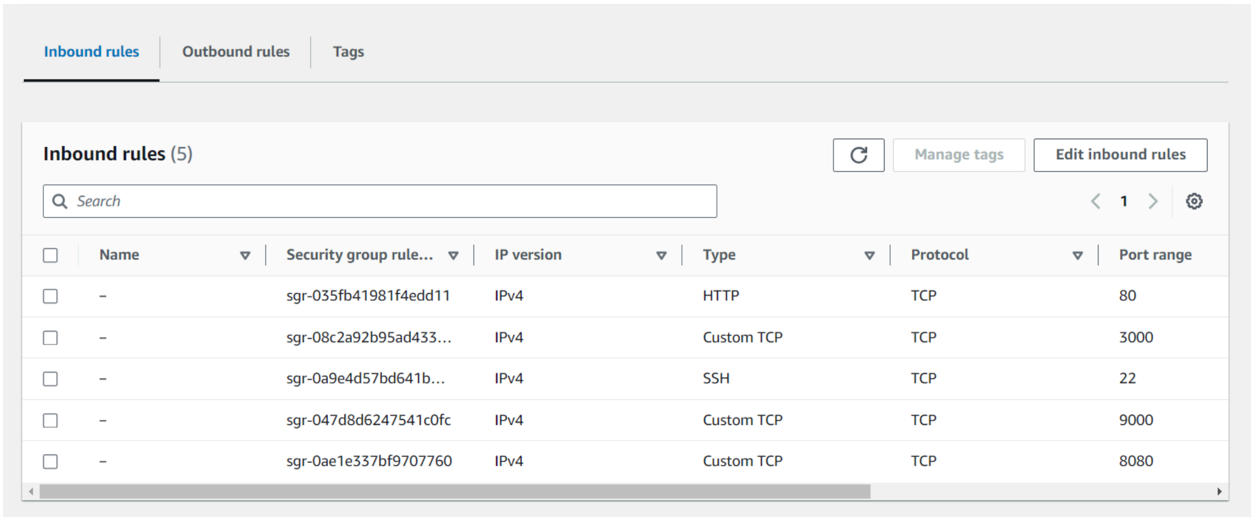
Constructing a Jenkins CI/CD Pipeline

Install Virtual Machine on AWS



Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
JenkinsRohit	i-086a9e6233ca09786	Running	t3.medium	2/2 checks passec	View alarms +	ap-south-1b	ec2-13-235-75-1

Make sure to enable these ports for access to the sonar and Jenkins

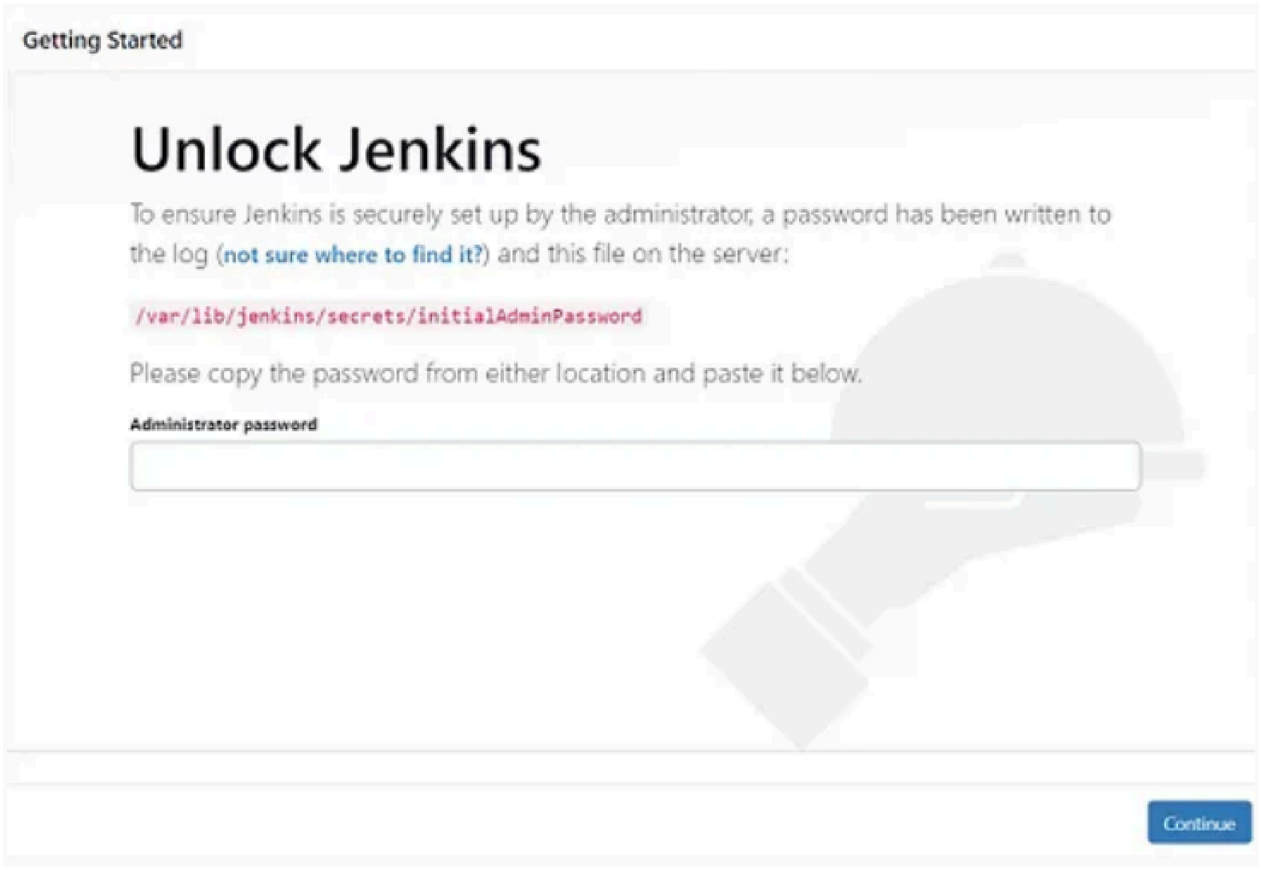


Name	Security group rule...	IP version	Type	Protocol	Port range
–	sgr-035fb41981f4edd11	IPv4	HTTP	TCP	80
–	sgr-08c2a92b95ad433…	IPv4	Custom TCP	TCP	3000
–	sgr-0a9e4d57bd641b…	IPv4	SSH	TCP	22
–	sgr-047d8d6247541c0fc	IPv4	Custom TCP	TCP	9000
–	sgr-0ae1e337bf9707760	IPv4	Custom TCP	TCP	8080

Setup Jenkins on the instance

```
sudo apt update sudo apt upgrade

sudo apt install openjdk-11-jdk
wget -q -O - https://pkg.jenkins.io/debian/jenkins.io.key | sudo apt-key add -
sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list'
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys 5BA31D57EF5975CA
sudo apt update sudo apt install jenkins
sudo systemctl start jenkins
sudo systemctl enable jenkins
```


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

Required Plugin

SonarQube Scanner for Jenkins
JaCoCo plugin
OWASP Dependency-Check Plugin
Slack Notification Plugin

Setup Sonar Qube on the instance

Install OpenJDK 17 (needed for the latest version of SonarQube (version 10.0).

```
sudo apt install -y openjdk-17-jdk
```

Add the PostgreSQL repository.

```
sudo sh -c 'echo "deb http://apt.postgresql.org/pub/repos/apt/ `lsb_release -cs`-pg
```

Add the PostgreSQL signing key.

```
wget -q https://www.postgresql.org/media/keys/ACCC4CF8.asc -O - | sudo apt-key add -
```

Install PostgreSQL.

```
sudo apt install postgresql postgresql-contrib -y
```

Enable the database server to start automatically on reboot.

```
sudo systemctl enable postgresql
```

Start the database server.

```
sudo systemctl start postgresql
```

Check the status of the database server

```
sudo systemctl status postgresql
```

Switch to the Postgres user.

```
sudo -i -u postgres
sudo useradd -d /opt/sonarqube -g ddsonar ddsonar
```

iii) Grant the sonar user access to the /opt/sonarqube directory.

```
sudo chown ddsonar:ddsonar /opt/sonarqube -R
```

Configure SonarQube

i) Edit the SonarQube configuration file.

```
sudo nano /opt/sonarqube/conf/sonar.properties
```

a) Find the following lines:

```
#sonar.jdbc.username=
#sonar.jdbc.password=
```

b) Uncomment the lines, and add the database user and Database password you created in Step 4 (xi and xii). For me, it's:

```
sonar.jdbc.username=ddsonar
sonar.jdbc.password=mwd#2%#!!#%rgs
```

c) Below these two lines, add the following line of code.

```
sonar.jdbc.url=jdbc:postgresql://localhost:5432/ddsonarqube
```

Save and exit the file.

ii) Edit the sonar script file.

```
sudo nano /opt/sonarqube/bin/linux-x86-64/sonar.sh
```

a) Add the following line

```
RUN_AS_USER=ddsonar
```

Setup Systemd service

i) Create a systemd service file to start SonarQube at system boot.

```
sudo nano /etc/systemd/system/sonar.service
```

ii) Paste the following lines to the file.

```
[Unit]
Description=SonarQube service
After=syslog.target network.target
[Service]
Type=forking
ExecStart=/opt/sonarqube/bin/linux-x86-64/sonar.sh start
ExecStop=/opt/sonarqube/bin/linux-x86-64/sonar.sh stop
User=ddsonar
Group=ddsonar
Restart=always
LimitNOFILE=65536
LimitNPROC=4096
[Install]
WantedBy=multi-user.target
```

Note: In the above script, make sure to change the User and Group section with the value you created. For me it:

```
User=ddsonar
Group=ddsonar
```

iv) Enable the SonarQube service to run at system startup.

```
sudo systemctl enable sonar
```

v) Start the SonarQube service.

```
sudo systemctl start sonar
```

vi) Check the service status.

```
sudo systemctl status sonar
```

Modify Kernel System Limits

SonarQube uses Elasticsearch to store its indices in an MMap FS directory. It requires some changes to the system defaults.

i) Edit the **sysctl** configuration file.

```
sudo nano /etc/sysctl.conf
```

ii) Add the following lines.

```
vm.max_map_count=262144
fs.file-max=65536
ulimit -n 65536
ulimit -u 4096
```

Reboot the system to apply the changes.

```
sudo reboot
```

Integrate Jenkins with GitHub

Log in to GitHub → Go to GitHub and login with your account → Go to Developer Settings → Click on your profile picture in the top right corner → Select "Settings" from the dropdown menu → In the left sidebar, click on "Developer settings." → Personal Access Tokens → In the left sidebar, click on "Personal access tokens." → Click on "Tokens (classic)" under the "Personal access tokens" section

 Generate New Token → Click the "Generate new token" button → Give your token a descriptive name in the "Note" field →
Set Scopes:

Select the scopes or permissions you want to grant this token. For Jenkins integration, you typically need the following scopes:
repo (Full control of private repositories)
admin:repo_hook (Full control of repository hooks)
user (Read all user profile data)
Generate Token:

Click the "Generate token" button at the bottom.
Copy the generated token. Store it securely, as you won't be able to see it again.
Integrate GitHub Token with Jenkins
Log in to Jenkins:

Open your Jenkins instance and log in.
Manage Credentials:

In Jenkins, go to "Manage Jenkins" > "Manage Credentials."
Select the domain (e.g., (global)).
Add Credentials:

Click on "Add Credentials."
For "Kind," select "Secret text."
Paste your GitHub personal access token into the "Secret" field.
Provide an ID and a description to identify the token.
Configure GitHub Plugin:

If you haven't already, install the GitHub Integration Plugin in Jenkins.
Go to "Manage Jenkins" > "Configure System."
Scroll down to the "GitHub" section and add the credentials you just created.

Write Pipeline for Java

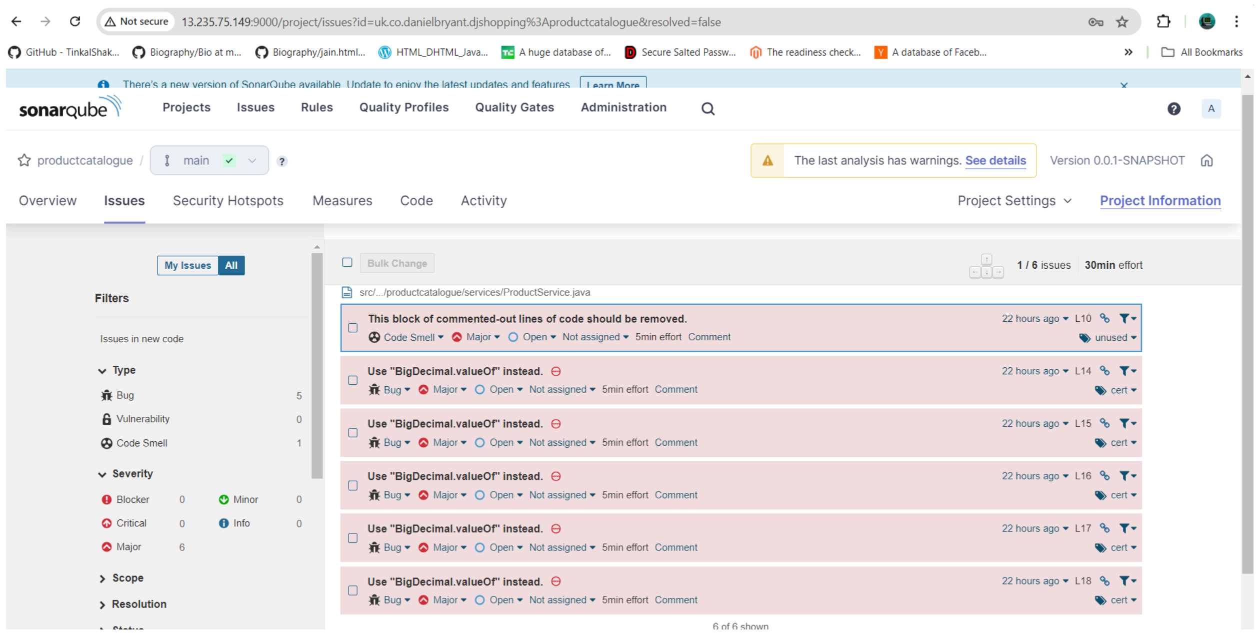
```
pipeline {
    agent any

    stages {
        stage('Checkout') {
            steps {
                git branch: 'main', credentialsId: 'GITHUB_CRED', url:
```

```
'https://github.com/01rohitjain/JavaApp.git'
            }
        }
        stage('Build') {
            steps {
                sh 'mvn install -U -P jar -DskipTests'
            }
        }
        stage('SonarQube Analysis') {
            steps {
                sh '''
                mvn clean verify sonar:sonar \
                -Dsonar.host.url=http://<IP>:9000 \
                -Dsonar.login= <SonarQubeToken> \
                -Dsonar.qualitygate.wait=true
                '''
            }
        }
        stage('Code Coverage') {
            steps {
                jacoco maximumBranchCoverage: '30',
maximumComplexityCoverage: '40', maximumInstructionCoverage: '2',
maximumLineCoverage: '40', sourcePattern: '**/src/main/java/**'
            }
        }
                stage('Security Scan') {
            steps {
                dependencyCheck additionalArguments: '''
                    -o './'
                    -s './'
                    -f 'ALL'
                    --prettyPrint''', odcInstallation: 'OWASP
Dependency-Check Vulnerabilities'
            }
        }
    }
    post {
        always {
            dependencyCheckPublisher pattern: 'dependency-check-report.xml'
        }
        success {
            emailext subject: "Build Success", body: "The build was
successful", to: 'team@example.com'
```

```
        }
        failure {
            emailext subject: "Build Failure", body: "The build failed",
to: 'team@example.com'
        }
    }
}
```

Sonar Qube Report



13.235.75.149:9000/project/issues?id=uk.co.danielbryant.djshopping%3Aproductcatalogue&resolved=false

There's a new version of SonarQube available. Update to enjoy the latest updates and features. Learn More

productcatalogue / main

The last analysis has warnings. See details   Version 0.0.1-SNAPSHOT

Overview | Issues | Security Hotspots | Measures | Code | Activity | Project Settings | Project Information

Filters

Issues in new code

Type
- Bug 5
- Vulnerability 0
- Code Smell 1

Severity
- Blocker 0 / Minor 0
- Critical 0 / Info 0
- Major 6

Scope

Resolution

Bulk Change   1 / 6 issues   30min effort

src/.../productcatalogue/services/ProductService.java

This block of commented-out lines of code should be removed.
Code Smell · Major · Open · Not assigned · 5min effort · Comment · 22 hours ago · L10 · unused

Use "BigDecimal.valueOf" instead.
Bug · Major · Open · Not assigned · 5min effort · Comment · 22 hours ago · L14 · cert

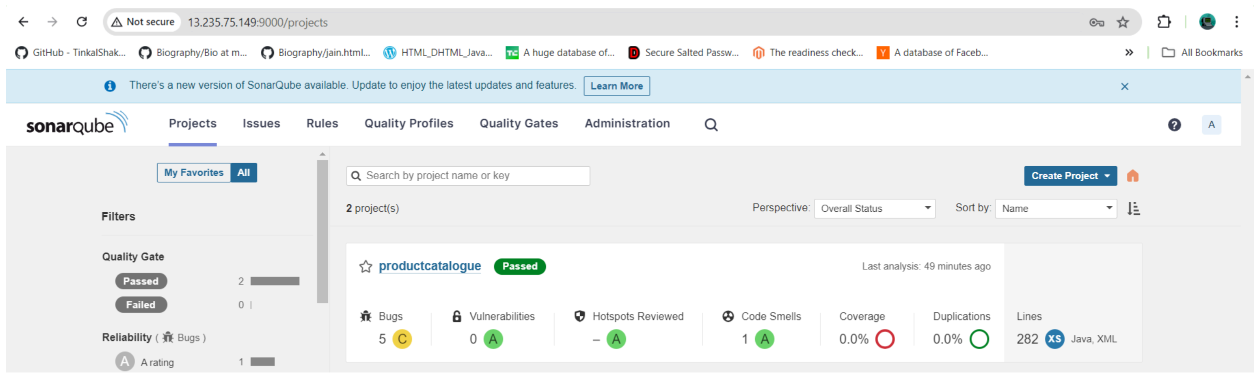
Use "BigDecimal.valueOf" instead.
Bug · Major · Open · Not assigned · 5min effort · Comment · 22 hours ago · L15 · cert

Use "BigDecimal.valueOf" instead.
Bug · Major · Open · Not assigned · 5min effort · Comment · 22 hours ago · L16 · cert

Use "BigDecimal.valueOf" instead.
Bug · Major · Open · Not assigned · 5min effort · Comment · 22 hours ago · L17 · cert

Use "BigDecimal.valueOf" instead.
Bug · Major · Open · Not assigned · 5min effort · Comment · 22 hours ago · L18 · cert

6 of 6 shown



13.235.75.149:9000/projects

There's a new version of SonarQube available. Update to enjoy the latest updates and features. Learn More

Filters

Quality Gate
- Passed 2
- Failed 0

Reliability (Bugs)
- A rating 1

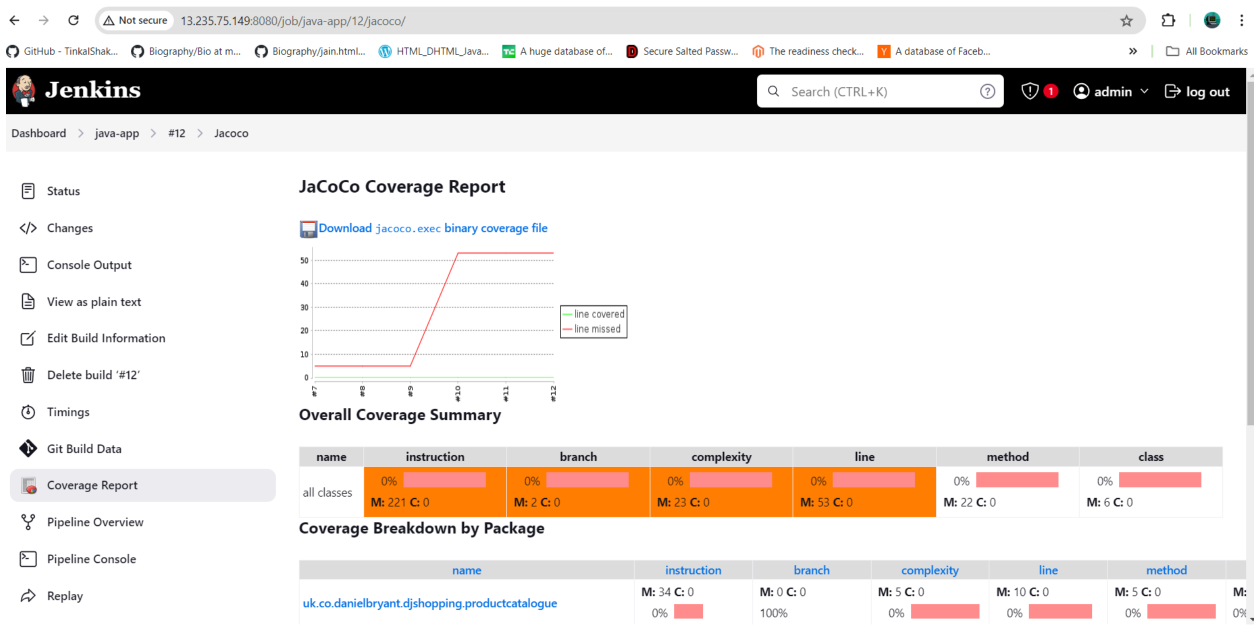
Search by project name or key   Create Project

2 project(s)   Perspective: Overall Status   Sort by: Name

productcatalogue Passed   Last analysis: 49 minutes ago

Bugs	Vulnerabilities	Hotspots Reviewed	Code Smells	Coverage	Duplications	Lines
5 C	0 A	– A	1 A	0.0%	0.0%	282 XS Java, XML

JaCoco Report



JaCoCo Coverage Report

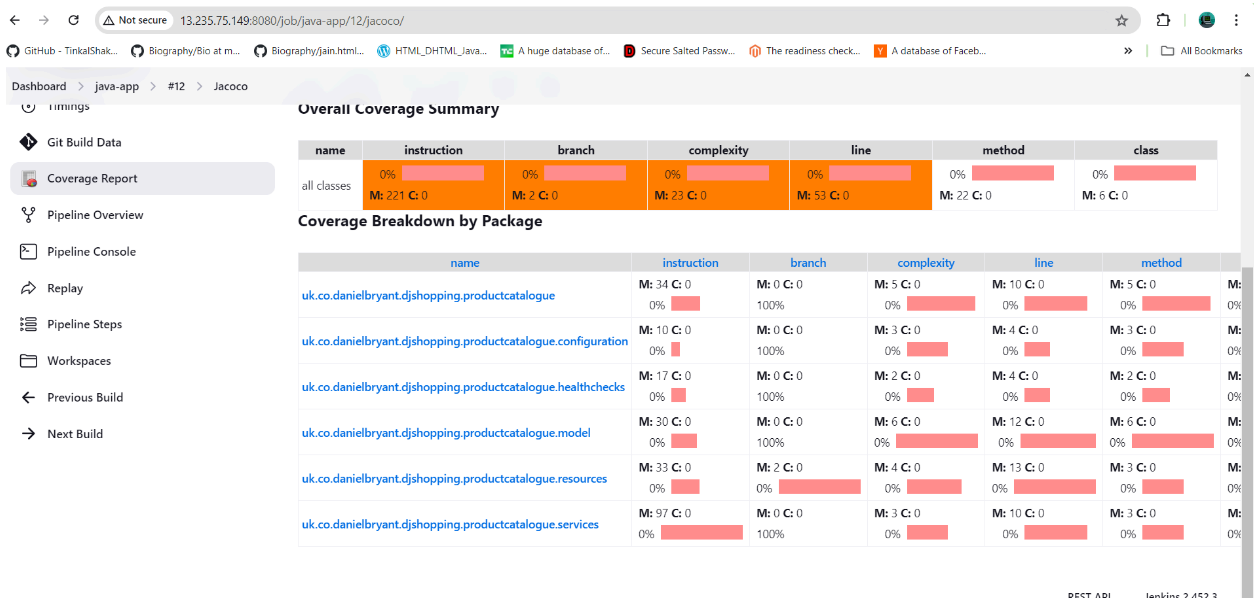
Download `jacoco.exec` binary coverage file

Overall Coverage Summary

name	instruction	branch	complexity	line	method	class
all classes	0% M: 221 C: 0	0% M: 2 C: 0	0% M: 23 C: 0	0% M: 53 C: 0	0% M: 22 C: 0	0% M: 6 C: 0

Coverage Breakdown by Package

name	instruction	branch	complexity	line	method	
uk.co.danielbryant.djshopping.productcatalogue	M: 34 C: 0 0%	M: 0 C: 0 100%	M: 5 C: 0 0%	M: 10 C: 0 0%	M: 5 C: 0 0%	M: 0%



Overall Coverage Summary

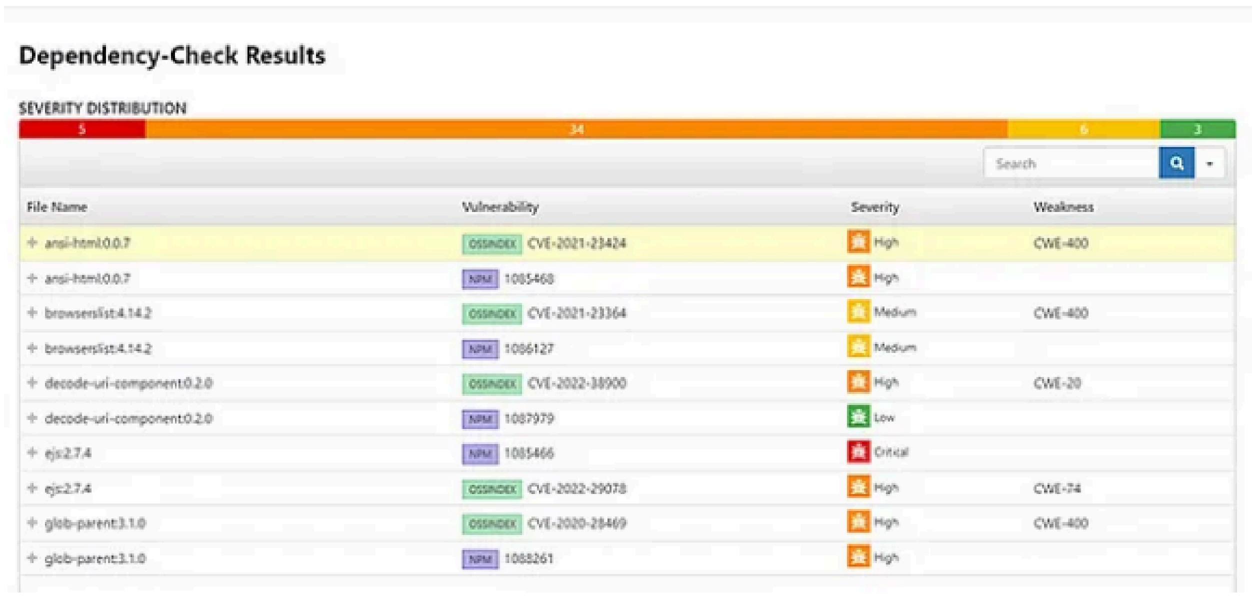
name	instruction	branch	complexity	line	method	class
all classes	0% M: 221 C: 0	0% M: 2 C: 0	0% M: 23 C: 0	0% M: 53 C: 0	0% M: 22 C: 0	0% M: 6 C: 0

Coverage Breakdown by Package

name	instruction	branch	complexity	line	method	
uk.co.danielbryant.djshopping.productcatalogue	M: 34 C: 0 0%	M: 0 C: 0 100%	M: 5 C: 0 0%	M: 10 C: 0 0%	M: 5 C: 0 0%	M: 0%
uk.co.danielbryant.djshopping.productcatalogue.configuration	M: 10 C: 0 0%	M: 0 C: 0 100%	M: 3 C: 0 0%	M: 4 C: 0 0%	M: 3 C: 0 0%	M: 0%
uk.co.danielbryant.djshopping.productcatalogue.healthchecks	M: 17 C: 0 0%	M: 0 C: 0 100%	M: 2 C: 0 0%	M: 4 C: 0 0%	M: 2 C: 0 0%	M: 0%
uk.co.danielbryant.djshopping.productcatalogue.model	M: 30 C: 0 0%	M: 0 C: 0 100%	M: 6 C: 0 0%	M: 12 C: 0 0%	M: 6 C: 0 0%	M: 0%
uk.co.danielbryant.djshopping.productcatalogue.resources	M: 33 C: 0 0%	M: 2 C: 0 0%	M: 4 C: 0 0%	M: 13 C: 0 0%	M: 3 C: 0 0%	M: 0%
uk.co.danielbryant.djshopping.productcatalogue.services	M: 97 C: 0 0%	M: 0 C: 0 100%	M: 3 C: 0 0%	M: 10 C: 0 0%	M: 3 C: 0 0%	M: 0%

REST API Jenkins 2.452.3

Owasp Scanning Report



Dependency-Check Results

SEVERITY DISTRIBUTION

5 | 34 | 6 | 3

File Name	Vulnerability	Severity	Weakness
ansi-html:0.0.7	OSSINDEX CVE-2021-23424	High	CWE-400
ansi-html:0.0.7	NPM 1085468	High	
browserslist:4.14.2	OSSINDEX CVE-2021-23364	Medium	CWE-400
browserslist:4.14.2	NPM 1086127	Medium	
decode-uri-component:0.2.0	OSSINDEX CVE-2022-38900	High	CWE-20
decode-uri-component:0.2.0	NPM 1087979	Low	
ejs:2.7.4	NPM 1085466	Critical	
ejs:2.7.4	OSSINDEX CVE-2022-29078	High	CWE-74
glob-parent:3.1.0	OSSINDEX CVE-2020-28469	High	CWE-400
glob-parent:3.1.0	NPM 1088261	High	

Troubleshoot

- I troubleshot the security group configuration to enable external access to Jenkins and SonarQube. To achieve this, we added inbound rules for ports 8080 and 9000 to the security group.
- I encountered a dependency issue with Jenkins and SonarQube due to a mismatch in Java versions. Initially, the default Java version installed was JRE 21.0. However, SonarQube requires Java version 17.x. We addressed this problem by troubleshooting and resolving the version incompatibility to ensure compatibility with SonarQube.
- I attempted to generate a cyclomatic complexity report using the Lizard tool, but I encountered difficulties setting up the tool for this purpose.