

Project Report

On

DevSecOps



Submitted in partial fulfillment for the award of
**Post Graduate Diploma in High Performance Computing
System Administration from C-DAC ACTS (Pune)**

Guided by:
Mr. Roshan Gami

Presented by:

Ms. JADHAV TEJASWINI RAMKRUSHNA (220940127006)
Ms. SWAMI PARVATI DAYANAND (220940127013)
Ms. SONAWANE DEVYANI BHARAT (220940127020)
Mr. CHANDRA BHUSHAN KUMAR (220940127035)
Ms. KUMARI SHWETA RANI (220940127041)

ACKNOWLEDGEMENT

This project “**DevSecOps**” was a great learning experience for us and we are submitting this work to **Advanced Computing Training School (ACTS Pune)**. We all are very glad to mention the name of **Mr. Roshan Gami** for his valuable guidance to work on this project. His guidance and support helped us to overcome various obstacles and intricacies during the course of project work.

We are highly grateful to Mr. Kaushal Sharma (Manager (ACTS training Centre), CDAC), for his guidance and support whenever necessary while doing this course Post Graduate Diploma in **High Performance Computing System Administration (PGDHPCSA)** through C-DAC ACTS, Pune.

Our most heartfelt thank goes to **Ms. Swati Salunke (Course Coordinator, PGDHPCSA)** who gave all the required support and kind coordination to provide all the necessities like required hardware, internet facility and extra Lab hours to complete the project and throughout the course up to the last day here in C-DAC ACTS Pune.

From:

Ms. JADHAV TEJASWINI RAMKRUSHNA (220940127006)

Ms. SWAMI PARVATI DAYANAND (220940127013)

Ms. SONAWANE DEVYANI BHARAT (220940127020)

Mr. CHANDRA BHUSHAN KUMAR (220940127035)

Ms. KUMARI SHWETA RANI (220940127041)

TABLE OF CONTENTS

1. ABSTRACT	05
2. Introduction	06
3. Software Requirement	07
4. Technologies Used	08
4.1 Github	09
4.2 jenkins	09
4.3 Trufflehog3	09
4.4 SonarQube	09
4.5 OWASP-ZAP	09
4.6 DOCKER	10
5. DevSecOps Architecture	11
6. Project Content	12-15
6.1 Configuration of Jenkins Server	16
6.1.1 Add port 8080 in the Jenkins Server	16
6.1.2 Jenkins Project Intigration with Github	17
6.1.3 Install Jenkins Plugins	18
6.2 Configuration of SonarQube Server	19
6.2.1 Add port 9000 in Sonarqube Server	19
6.2.2 Add User in Sonarqube Server	20
6.2.3 Create user in SonarQube Server	20
6.2.4 Set User-name and password in SonarQube Server	21
6.2.5 Set user as Adminitration in SonarQube Server	21
6.2.6 Create Jenkins credential for Jenkins Server in SonarQube Server ..	22

7. Dashboard of WebGoat Application	24
8. Dashboard of Jenkins Server after Project Build	25
8.1 Dependency-Check Results	26
8.2 Dependency-Check Trend Status	26
8.3 Dashboard of SonarQube after checked Bugs and Vulnerability	27
9. OWASP-ZAP Server check Scanned Report	28
9.1 Dashbord of OWASP-ZAP Server	28
9.2 Check Application Scanned Report	28
9.3 Screen shoot of zap_report.xml	29
10. Conclusion	30
11. Bibliography	31

ABSTRACT

DevSecOps is a methodology that emphasizes the integration of security practices and tools throughout the software development life cycle, from the initial design phase to production deployment and beyond. It is a collaborative approach that seeks to eliminate silos between development, security, and operations teams, and prioritize security as a core element of the software development process. By adopting DevSecOps practices, organizations can improve the security posture of their software applications, reduce the risk of cyber attacks, and increase their ability to respond to security incidents. This approach also encourages continuous improvement and learning, and empowers DevSecOps is a methodology that emphasizes the integration of security practices and tools throughout the software development life cycle, from the initial design phase to production deployment and beyond. It is a collaborative approach that seeks to eliminate silos between development, security, and operations teams, and prioritize security as a core element of the software development process. By adopting DevSecOps practices, organizations can improve the security posture of their software applications, reduce the risk of cyberattacks, and increase their ability to respond to security incidents. This approach also encourages continuous improvement and learning, and empowers developers and security professionals to work together towards a common goal of building secure and resilient software systems.

Introduction

DevSecOps is a trending practice in application security (AppSec) that involves introducing security earlier in the Secure Software Development Life Cycle (SSDLC). It also expands the collaboration between development and operations teams to integrate security teams in the software delivery cycle. DevSecOps requires a change in culture, process, and tools across these core functional teams and makes security a shared responsibility. Everyone involved in the SSDLC has a role to play in building security into the DevOps continuous integration and continuous delivery (CI/CD) workflow.

DevSecOps (Development, Security, and Operations) is an approach to software development that incorporates security as an integral part of the entire Secure software development life cycle (SSDLC), from design to deployment and ongoing maintenance. It is an extension of the DevOps philosophy, which aims to integrate development and operations teams for faster and more efficient software development.

The key objective of DevSecOps is to incorporate security practices and controls into the software development process to ensure that security is not an afterthought, but an essential part of the software development process. This approach helps to identify and address security vulnerabilities early in the SSDLC, rather than waiting until after deployment, when fixing security issues can be much more costly and time-consuming.

DevSecOps involves a collaborative effort between developers, security teams, and operations teams to ensure that security is integrated throughout the software development process. This requires a shift in mindset and culture, with a focus on continuous testing, automation, and collaboration to achieve a more secure and resilient software development process.

Software Requirement

1. AWS ubuntu instance
2. Docker
3. Jenkins 2.375.3
4. Git
5. Trufflehog3
6. OWASP Dependency-Check
7. Maven
8. SonarQube
9. OWASP-ZAP

Technologies Used

SAST - Static application security testing

- SAST is a set of technologies designed to analyze application source code, byte code and binaries for coding and design conditions that are indicative of security vulnerabilities.
- It consists of scans performed on source code to identify the maximum number of potential vulnerabilities, before the resulting artifact could be even built.
- SAST is an essential step in the Software Development Life Cycle (SDLC) because it identifies critical vulnerabilities in an application before it's deployed to the public, while they're the least expensive to remediate.

DAST- Dynamic Application Security Testing

- It is a type of security testing that involves testing a web application in a live, running state to identify security vulnerabilities.
- It finds vulnerabilities by employing fault injection techniques on an app.
- It can also cast a spotlight on runtime problems that can't be identified by static analysis, like authentication and server configuration issues, as well as flaws visible only when a known user logs in.
- These tools typically test HTTP and HTML interfaces of web applications.
A dynamic analysis security testing tool, or a DAST test, is an application security solution that can help to find certain vulnerabilities in web applications while they are running in production.

SCA- Software Composition Analysis

- Is a process of identifying and analyzing the third-party components used in software development. It involves identifying the open source and commercial software components used in a codebase, determining the origin and licensing of these components, and assessing the potential security and compliance risks associated with using them.
- SCA tools are used to automate the process of identifying and analyzing third-party components. These tools typically scan a codebase to identify the components used and then provide information about the components, including their version, licensing, and any known vulnerabilities or security issues.

4.1 Github

GitHub is a code hosting platform for version control and collaboration. It lets you and others work together on projects from anywhere. This tutorial teaches you GitHub essentials like repositories, branches, commits, and pull requests.

4.2 Jenkins

Jenkins is an open source automation server. It helps automate the parts of software development related to building, testing, and deploying, facilitating continuous integration and continuous delivery. Jenkins is a Java-based open-source automation platform with plugins designed for continuous integration. It is used to continually create and test software projects, making it easier for developers and DevOps engineers to integrate changes to the project and for consumers to get a new build.

4.3 Trufflehog3

TruffleHog is an open-source tool that scans your environment for secrets like SSH private keys, API keys, database passwords, authentication/access tokens, cloud credentials and more. It can run scans continuously in the background every time changes are made and notify you when secrets are found.

4.4 SonarQube

Sonar is an open-source software quality platform. SonarQube saves the calculated measures in a database and showcases them in a rich web-based dashboard. Provides trends and leading indicators. Sonar uses various static & dynamic code analysis tools such as Checkstyle, PMD, FindBugs, FxCop, Gendarme, and many more to extract software metrics, which then can be used to improve software quality. Provides lots of plugins.

4.5 OWASP-ZAP

The Open Worldwide Application Security Project (OWASP-ZAP) is an online community that produces freely-available articles, methodologies, documentation, tools, and technologies in the field of web application security. The OWASP -ZAP provides free and open resources. It is led by a non-profit called The OWASP-ZAP Foundation.

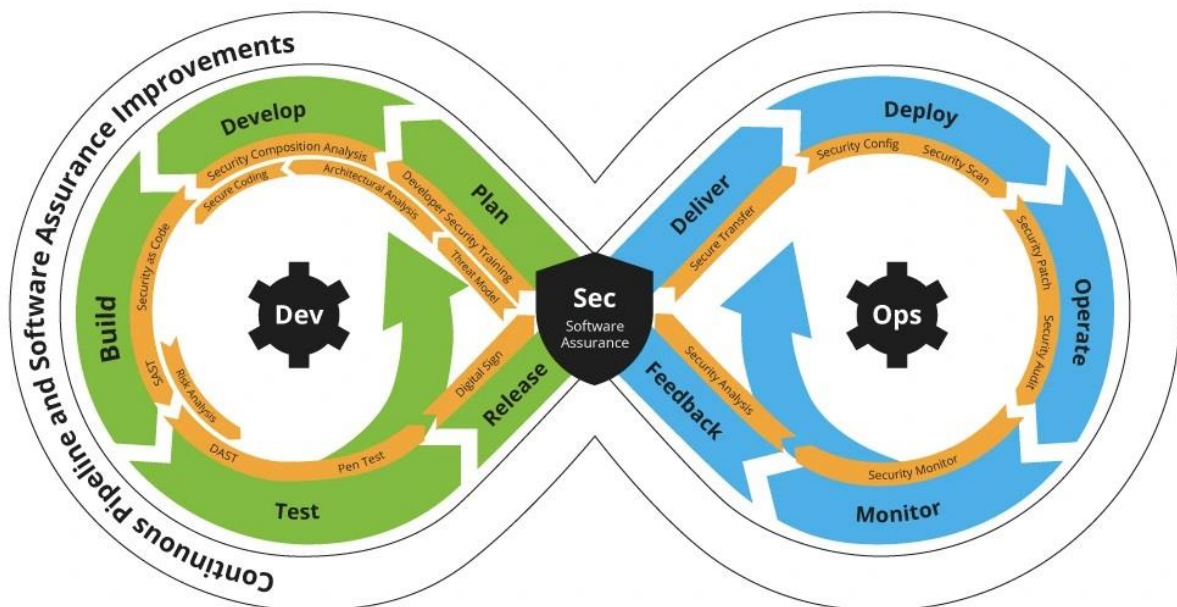
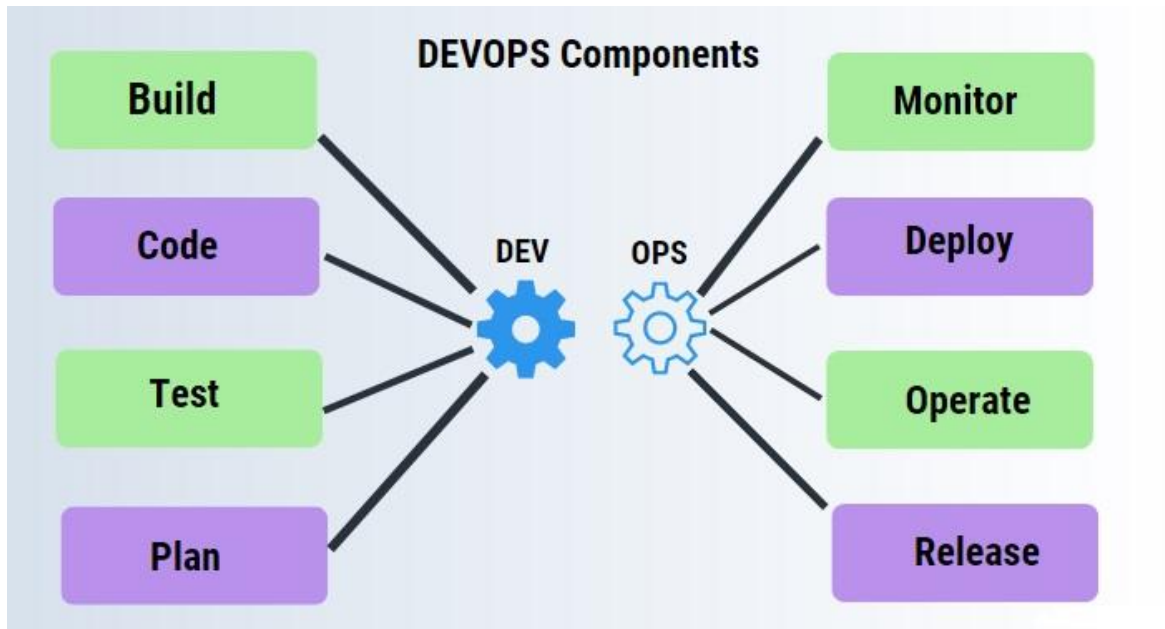
The OWASP-ZAP Top 10 is a standard awareness document for developers and web application security.

OWASP Top 10 - 2013	→	OWASP Top 10 - 2017
A1 – Injection	→	A1:2017-Injection
A2 – Broken Authentication and Session Management	→	A2:2017-Broken Authentication
A3 – Cross-Site Scripting (XSS)	↘	A3:2017-Sensitive Data Exposure
A4 – Insecure Direct Object References [Merged+A7]	U	A4:2017-XML External Entities (XXE) [NEW]
A5 – Security Misconfiguration	↘	A5:2017-Broken Access Control [Merged]
A6 – Sensitive Data Exposure	↗	A6:2017-Security Misconfiguration
A7 – Missing Function Level Access Contr [Merged+A4]	U	A7:2017-Cross-Site Scripting (XSS)
A8 – Cross-Site Request Forgery (CSRF)	⊗	A8:2017-Insecure Deserialization [NEW, Community]
A9 – Using Components with Known Vulnerabilities	→	A9:2017-Using Components with Known Vulnerabilities
A10 – Unvalidated Redirects and Forwards	⊗	A10:2017-Insufficient Logging&Monitoring [NEW,Comm.]

4.6 DOCKER

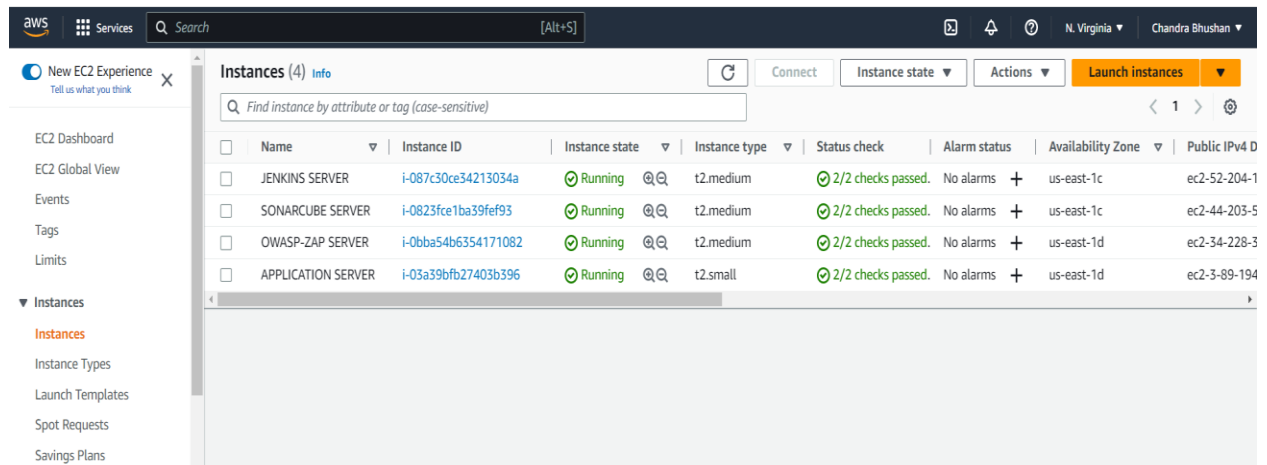
Docker is a platform that enables developers to create, deploy, and run applications in a containerized environment. Containers are lightweight, portable, and self-contained environments that bundle together all the dependencies required to run an application, including the code, libraries, and runtime.

5. DevSecOps Architecture



Project content

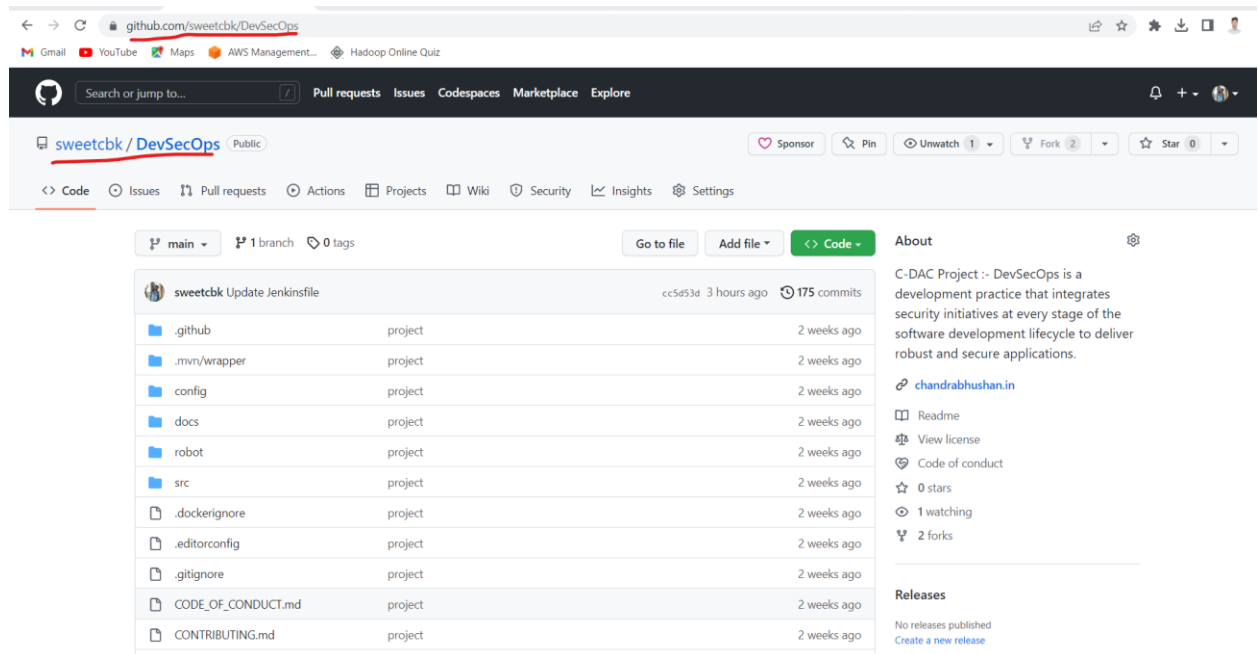
Created 4 AWS EC2 Ubuntu Instance for DevSecOps



The screenshot shows the AWS Management Console 'Instances' page. It displays a table of 4 EC2 instances, all in a 'Running' state. The instances are named JENKINS SERVER, SONARCUBE SERVER, OWASP-ZAP SERVER, and APPLICATION SERVER. Each instance has a unique Instance ID, is of type t2.medium or t2.small, and has 2/2 checks passed. The instances are located in the us-east-1c and us-east-1d Availability Zones. The left sidebar shows the navigation menu with 'Instances' selected.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 D
JENKINS SERVER	i-087c30ce34213034a	Running	t2.medium	2/2 checks passed.	No alarms	us-east-1c	ec2-52-204-1
SONARCUBE SERVER	i-0823fce1ba39fef93	Running	t2.medium	2/2 checks passed.	No alarms	us-east-1c	ec2-44-203-5
OWASP-ZAP SERVER	i-0bba54b6354171082	Running	t2.medium	2/2 checks passed.	No alarms	us-east-1d	ec2-34-228-3
APPLICATION SERVER	i-03a39bfb27403b396	Running	t2.small	2/2 checks passed.	No alarms	us-east-1d	ec2-3-89-194

Project Github url :- <https://github.com/sweetcbk/DevSecOps>



The screenshot shows the GitHub repository page for 'sweetcbk/DevSecOps'. The repository is public and has 175 commits. The file list shows various project files and folders, including .github, .mvn/wrapper, config, docs, robot, src, .dockerignore, .editorconfig, .gitignore, CODE_OF_CONDUCT.md, and CONTRIBUTING.md. The right sidebar contains information about the repository, including the README, View license, Code of conduct, 0 stars, 1 watching, and 2 forks.

Repository: sweetcbk / DevSecOps (Public)

Files and folders:

- .github (project) - 2 weeks ago
- .mvn/wrapper (project) - 2 weeks ago
- config (project) - 2 weeks ago
- docs (project) - 2 weeks ago
- robot (project) - 2 weeks ago
- src (project) - 2 weeks ago
- .dockerignore (project) - 2 weeks ago
- .editorconfig (project) - 2 weeks ago
- .gitignore (project) - 2 weeks ago
- CODE_OF_CONDUCT.md (project) - 2 weeks ago
- CONTRIBUTING.md (project) - 2 weeks ago

About: C-DAC Project :- DevSecOps is a development practice that integrates security initiatives at every stage of the software development lifecycle to deliver robust and secure applications.

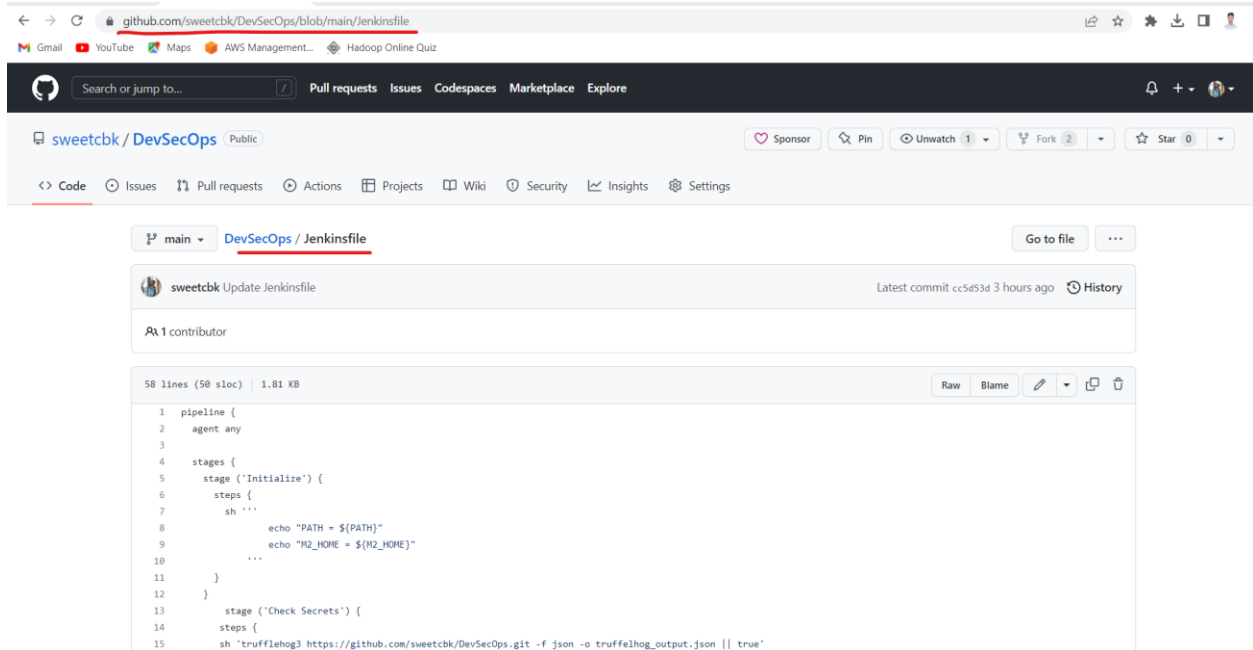
chandrabhushan.in

Readme, View license, Code of conduct, 0 stars, 1 watching, 2 forks.

Releases: No releases published. Create a new release.

- **Jenkins File url :-**

<https://github.com/sweetcbk/DevSecOps/blob/main/Jenkinsfile>



The screenshot shows a web browser displaying the GitHub repository page for 'sweetcbk/DevSecOps'. The URL in the address bar is 'https://github.com/sweetcbk/DevSecOps/blob/main/Jenkinsfile'. The page header includes the GitHub logo, a search bar, and navigation links for Pull requests, Issues, Codespaces, Marketplace, and Explore. Below the header, the repository name 'sweetcbk / DevSecOps' is shown with a 'Public' badge. A 'Sponsor' button and a 'Pin' button are visible. The file 'Jenkinsfile' is selected, and the commit history shows 'sweetcbk Update Jenkinsfile' with the latest commit 'cc5453d' from 3 hours ago. The file content is displayed in a code editor with line numbers 1 to 15. The code is a Jenkins pipeline script.

```
1 pipeline {
2   agent any
3
4   stages {
5     stage ('Initialize') {
6       steps {
7         sh '''
8           echo "PATH = ${PATH}"
9           echo "M2_HOME = ${M2_HOME}"
10          ...
11        }
12      }
13    stage ('Check Secrets') {
14      steps {
15        sh 'trufflehog3 https://github.com/sweetcbk/DevSecOps.git -f json -o trufflehog_output.json || true'
```

• Jenkins File Script

```
pipeline
{
    agent any
    stages {
        stage ('Initialize') {
            steps {
sh '''
                                echo "PATH = ${PATH}"
                                echo "M2_HOME = ${M2_HOME}"
                                ...
                                }
            }
        stage ('Check Secrets') {
            steps {
sh 'trufflehog3 https://github.com/sweetcbk/DevSecOps.git -f json -o
truffelhog_output.json || true'
            }
        }
        stage ('Software Composition Analysis') {
            steps {
dependencyCheckadditionalArguments: '''
                                -o "./"
                                -s "./"
                                -f "ALL"
                                --prettyPrint''', odcInstallation: 'owasp-dc'

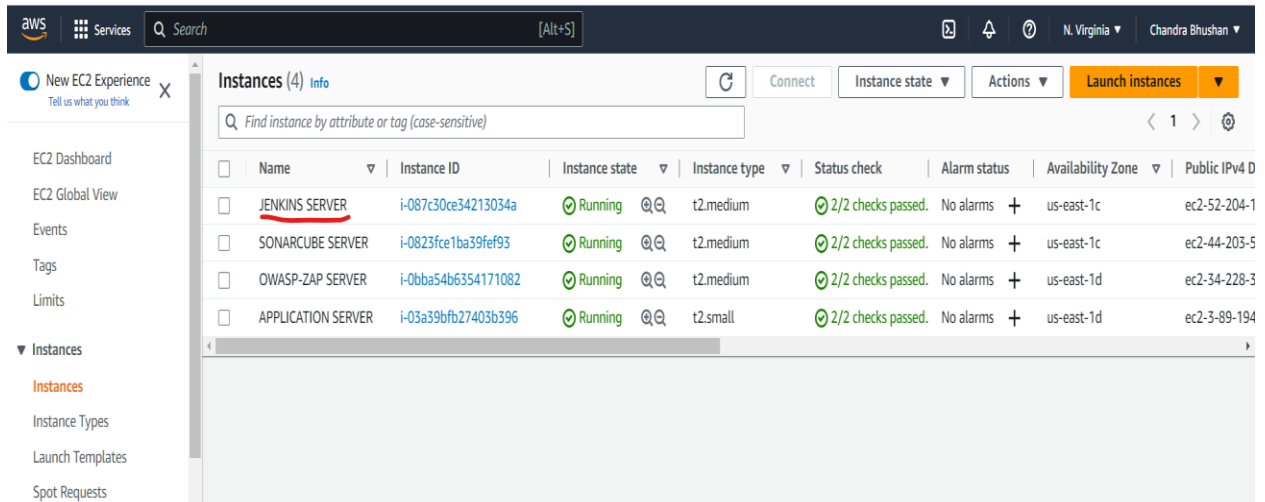
dependencyCheckPublisher pattern: 'dependency-check-report.xml'
                                }
            }
        stage ('Static Analysis') {
            steps {
withSonarQubeEnv('Sonar') {
sh 'mvnsonar:sonar'
            }
        }
        stage ('Deploy to Server Application') {
            steps {
sshagent(['server-application']) {
```

```

sh 'scp -o StrictHostKeyChecking=no
/var/lib/jenkins/workspace/project/webgoat-server-v8.2.0-SNAPSHOT.jar
ubuntu@3.89.194.15:/WebGoat'
sh 'ssh -o StrictHostKeyChecking=no ubuntu@3.89.194.15 "nohup java -jar
webgoat-server-v8.2.0-SNAPSHOT.jar --server.address=0.0.0.0 --
server.port=8080 &"'
    }
  }
}
stage ('Dynamic analysis') {
  steps {
sshagent(['application_server']) {
sh 'ssh -o StrictHostKeyChecking=no ubuntu@34.228.38.88 "sudo docker run
--rm -v /home/ubuntu:/zap/wrk/:rw -t owasp/zap2docker-stable zap-full-
scan.py -t http://3.89.194.15:8080/WebGoat -x zap_report || true" '
    }
  }
}
}
}
}

```

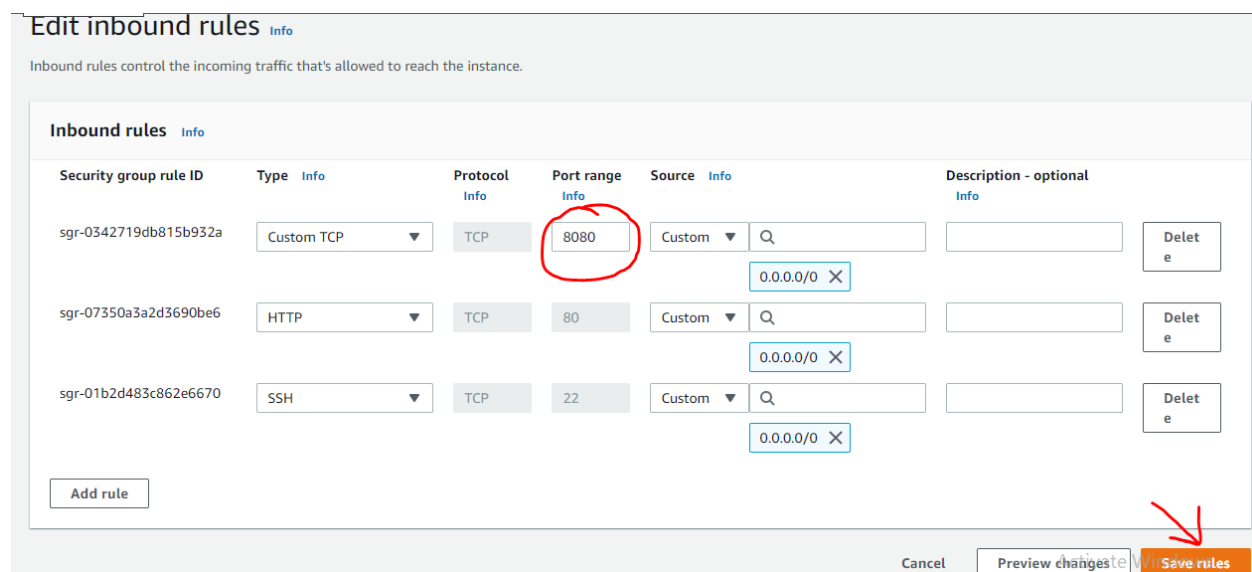
6.1 Configuration of Jenkins Server



The screenshot shows the AWS Management Console 'Instances' page. The left sidebar contains navigation links for EC2 Dashboard, EC2 Global View, Events, Tags, Limits, and Instances. The main content area displays a table of instances. The 'JENKINS SERVER' instance is highlighted with a red underline. The table columns include Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IPv4 D.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 D
JENKINS SERVER	i-087c30ce34213034a	Running	t2.medium	2/2 checks passed.	No alarms	us-east-1c	ec2-52-204-1
SONARCUBE SERVER	i-0823fce1ba39fef93	Running	t2.medium	2/2 checks passed.	No alarms	us-east-1c	ec2-44-203-5
OWASP-ZAP SERVER	i-0bba54b6354171082	Running	t2.medium	2/2 checks passed.	No alarms	us-east-1d	ec2-34-228-3
APPLICATION SERVER	i-03a39bfb27403b396	Running	t2.small	2/2 checks passed.	No alarms	us-east-1d	ec2-3-89-194

6.1.1 Add port 8080 in the Jenkins Server



The screenshot shows the 'Edit inbound rules' page in the AWS Management Console. The page title is 'Edit inbound rules' with an 'Info' link. Below the title is a description: 'Inbound rules control the incoming traffic that's allowed to reach the instance.' The main content area is titled 'Inbound rules' with an 'Info' link. It displays a table of inbound rules. The first rule has a 'Port range' of '8080', which is circled in red. The 'Save rules' button at the bottom right is highlighted with a red arrow.

Security group rule ID	Type	Protocol	Port range	Source	Description - optional	
sgr-0342719db815b932a	Custom TCP	TCP	8080	Custom 0.0.0.0/0		Delete
sgr-07350a3a2d3690be6	HTTP	TCP	80	Custom 0.0.0.0/0		Delete
sgr-01b2d483c862e6670	SSH	TCP	22	Custom 0.0.0.0/0		Delete

6.1.2 Jenkins Project Intigration with Github

Dashboard > project > Configuration

Configure

- General
- Advanced Project Options
- Pipeline

Pipeline

Definition
Pipeline script from SCM

SCM ?
Git

Repositories ?

Repository URL ?
https://github.com/sweetcbk/DevSecOps.git

Credentials ?
- none -
+ Add
Advanced...

Add Repository

Save Apply

Dashboard > project > Configuration

Configure

- General
- Advanced Project Options
- Pipeline

Pipeline

Branches to build ?

Branch Specifier (blank for 'any') ?
*/main

Add Branch

Repository browser ?
(Auto)

Additional Behaviours
Add +

Script Path ?
Jenkinsfile

☒ Lightweight checkout ?

Pipeline Syntax

Save Apply

6.1.3 Install Jenkins Plugins

1. [Maven SNAPSHOT Check Plugin](#)
2. [Pipeline Maven Integration](#)

Dashboard > Manage Jenkins > Plugin Manager

Search: maven

A plug-in that enables you to perform releases using the maven-release-plugin from Jenkins. Report an issue with this plugin	<input checked="" type="checkbox"/>	<input type="checkbox"/>
This plugin is up for adoption! We are looking for new maintainers. Visit our Adopt a Plugin initiative for more information.		
Maven Repository Scheduled Cleanup Plugin 1.2 Report an issue with this plugin	<input checked="" type="checkbox"/>	<input type="checkbox"/>
This plugin is deprecated. In general, this means that it is either obsolete, no longer being developed, or may no longer work. Learn more .		
Maven Repository Server Plugin 1.10 This plug-in exposes project builds as a maven repository so the artifacts can be picked up by downstream builds or other systems. Report an issue with this plugin	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Maven SNAPSHOT Check Plugin 186.v844b_fed00d1b This plugin is used to check if pom.xml contains SNAPSHOT. Report an issue with this plugin	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Pipeline Maven Integration 1257.v89e586d3c58c This plugin provides integration with Pipeline, configures maven environment to use within a pipeline job by calling sh mvn or bat mvn. The selected maven installation will be configured and prepended to the path. Report an issue with this plugin	<input checked="" type="checkbox"/>	<input type="checkbox"/>

3. [SonarQube Scanner for Jenkins](#)

Jenkins

Search (CTRL+K)

Dashboard > Manage Jenkins > Plugin Manager

Updates
Available plugins
Installed plugins
Advanced settings

Plugins

Search: sonar

Name	Enabled
SonarQube Scanner for Jenkins 2.15 This plugin allows an easy integration of SonarQube, the open source platform for Continuous Inspection of code quality. Report an issue with this plugin	<input checked="" type="checkbox"/>

4. [OWASP Dependency-Check Plugin](#)

Jenkins

Search (CTRL+K)

Dashboard > Manage Jenkins > Plugin Manager

Updates
Available plugins
Installed plugins
Advanced settings

Plugins

Search: owasp

Official OWASP ZAP Jenkins Plugin 1.1.0 The Official OWASP ZAP Jenkins Plugin extends the functionality of the ZAP security tool into a CI Environment. Report an issue with this plugin	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Warning: The currently installed plugin version may not be safe to use. Please review the following security notices: <ul style="list-style-type: none">Credentials stored in plain text		
OWASP Dependency-Check Plugin 5.3.0 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. Report an issue with this plugin	<input checked="" type="checkbox"/>	<input type="checkbox"/>

4. [SSH Agent](#)

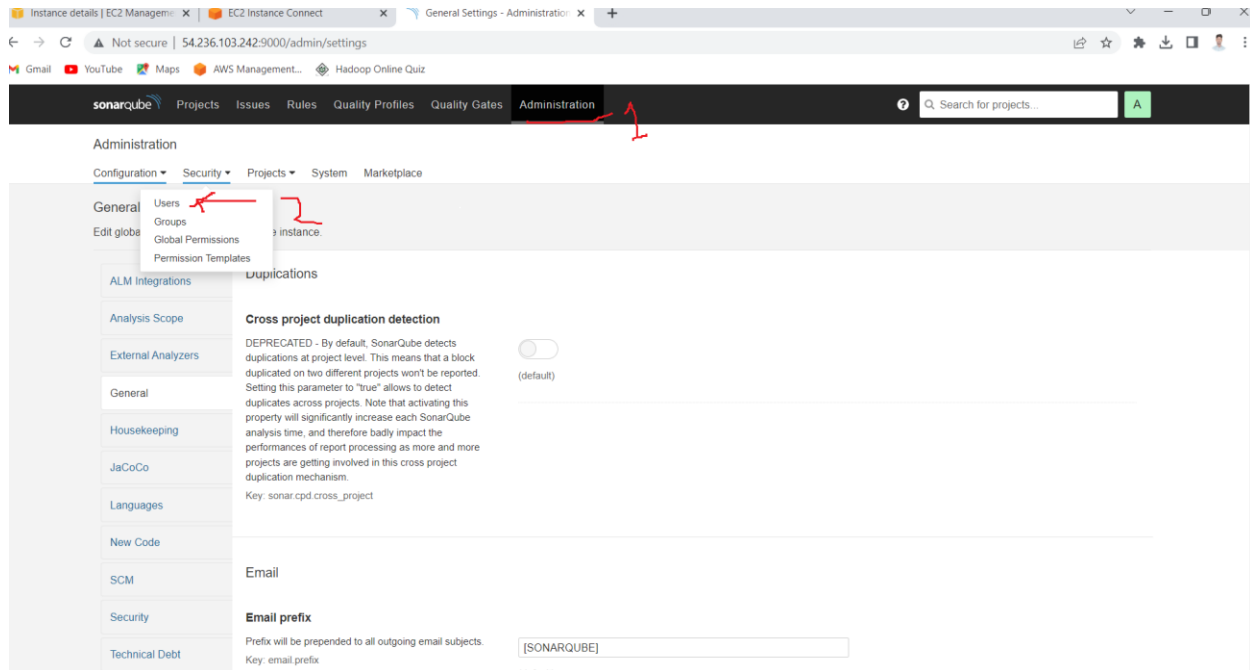
The screenshot shows the Jenkins Plugin Manager interface. On the left, there is a sidebar with navigation links: Updates, Available plugins, Installed plugins (selected), and Advanced settings. The main area is titled 'Plugins' and contains a search bar with 'ssh agent' entered. Below the search bar, there is a table of installed plugins. The first plugin is 'SSH Agent' (version 295.v9ca_a_1c7cc3a_a), which is described as 'This plugin allows you to provide SSH credentials to builds via a ssh-agent in Jenkins.' It is currently enabled, indicated by a blue toggle switch. The second plugin is 'SSH Build Agents' (version 2.854.v7fd446b_337c9), described as 'Allows to launch agents over SSH, using a Java implementation of the SSH protocol.' It is also enabled. Each plugin entry has a 'Report an issue with this plugin' link.

6.2 Configuration of SonarQube Server

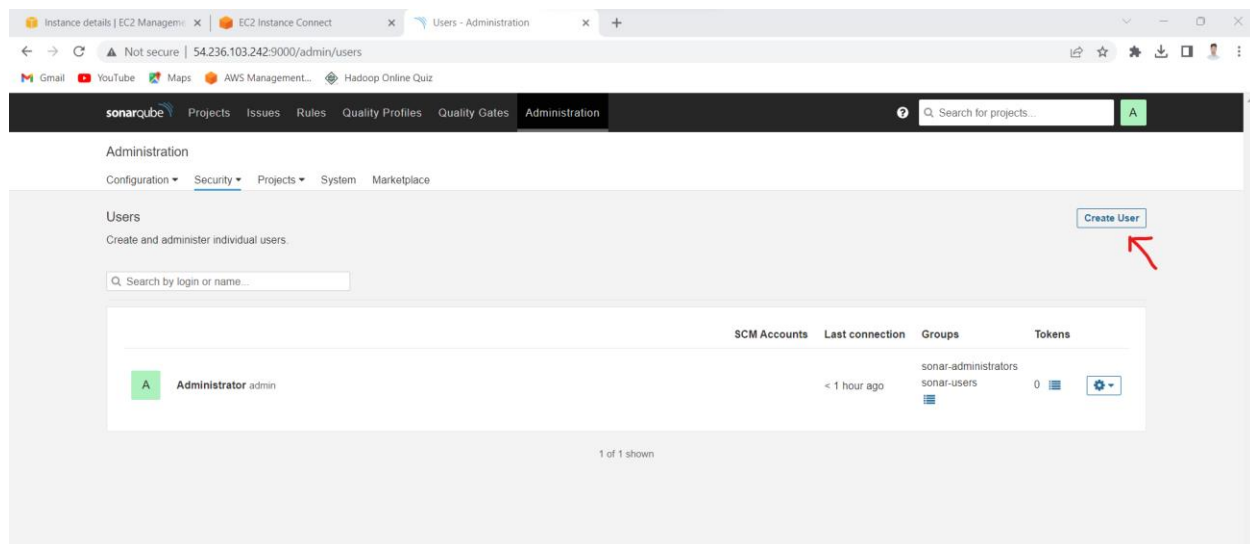
6.2.1 Add port 9000 in Sonarqube Server

The screenshot shows the AWS Management Console interface for an EC2 instance. The left sidebar contains navigation links for EC2 Dashboard, EC2 Global View, Events, Tags, Limits, Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Scheduled Instances, Capacity Reservations, Images, AMIs, and AMI Catalog. The main area displays the 'Security' tab for the instance. It shows the IAM Role, VPC ID, Subnet ID, and Launch time. Below this, there is a section for 'Inbound rules' with a table of security group rules. The table has columns for Name, Security group rule ID, Port range, Protocol, Source, and Security groups. The first rule is 'sgr-0962421ad21796517' with a port range of '9000' (highlighted with a red underline), protocol of 'TCP', and source of '0.0.0.0/0'. The second rule is 'sgr-0fa8db83d6d0ac717' with a port range of '80', protocol of 'TCP', and source of '0.0.0.0/0'. The third rule is 'sgr-0ea1048119a687d38' with a port range of '22', protocol of 'TCP', and source of '0.0.0.0/0'. All rules are associated with the 'launch-wizard-8' security group.

6.2.2 Add User in Sonarqube Server



6.2.3 Create user in SonarQube Server



6.2.4 Set User-name and password in SonarQube Server

The screenshot shows the SonarQube 'Create User' dialog box. The fields are as follows:

Field	Value
Login *	sonar-admin
Name *	Sonar-Admin
Email	sonar@gmail.com
Password *	*****

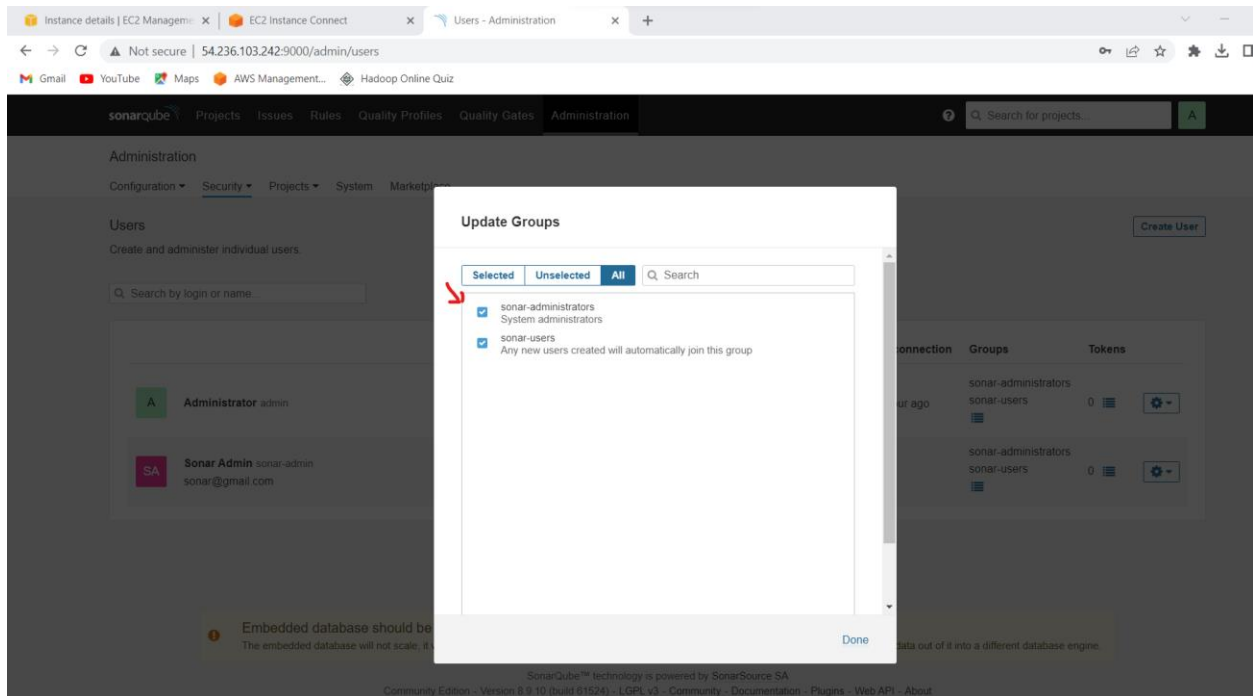
Below the fields, there is an 'Add' button under the 'SCM Accounts' section. A red arrow points to the 'Create' button at the bottom right of the dialog.

6.2.5 Set user as Administration in SonarQube Server

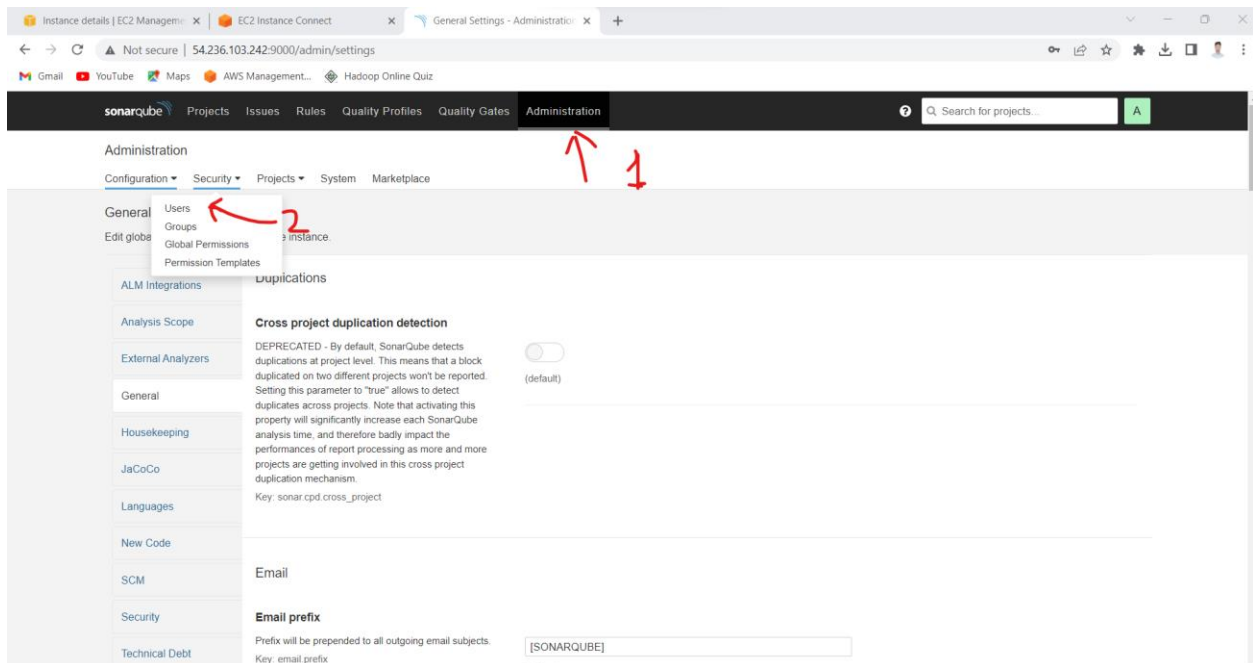
The screenshot shows the SonarQube 'Users' page. The 'Sonar Admin' user is listed in the table below:

	SCM Accounts	Last connection	Groups	Tokens
A Administrator admin		< 1 hour ago	sonar-administrators sonar-users	0
SA Sonar Admin sonar-admin sonar@gmail.com		Never	sonar-administrators sonar-users	0

A red box highlights the 'sonar-users' group for the 'Sonar Admin' user, with a red arrow pointing to it.



6.2.6 Create Jenkins credential for Jenkins Server in SonarQube Server



Instance details | EC2 Management | EC2 Instance Connect | Users - Administration

Not secure | 54.236.103.242:9000/admin/users

Gmail YouTube Maps AWS Management... Hadoop Online Quiz

sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration

Administration

Configuration Security Projects System Marketplace

Users

Create and administer individual users.

Search by login or name...

	SCM Accounts	Last connection	Groups	Tokens
A Administrator admin		< 1 hour ago	sonar-administrators sonar-users	0
SA Sonar Admin sonar-admin sonar@gmail.com		Never	sonar-administrators sonar-users	0

2 of 2 shown

Instance details | EC2 Management | EC2 Instance Connect | Users - Administration

Not secure | 54.236.103.242:9000/admin/users

Gmail YouTube Maps AWS Management... Hadoop Online Quiz

sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration

Administration

Configuration Security Projects System Marketplace

Users

Create and administer individual users.

Search by login or name...

Tokens of Sonar Admin

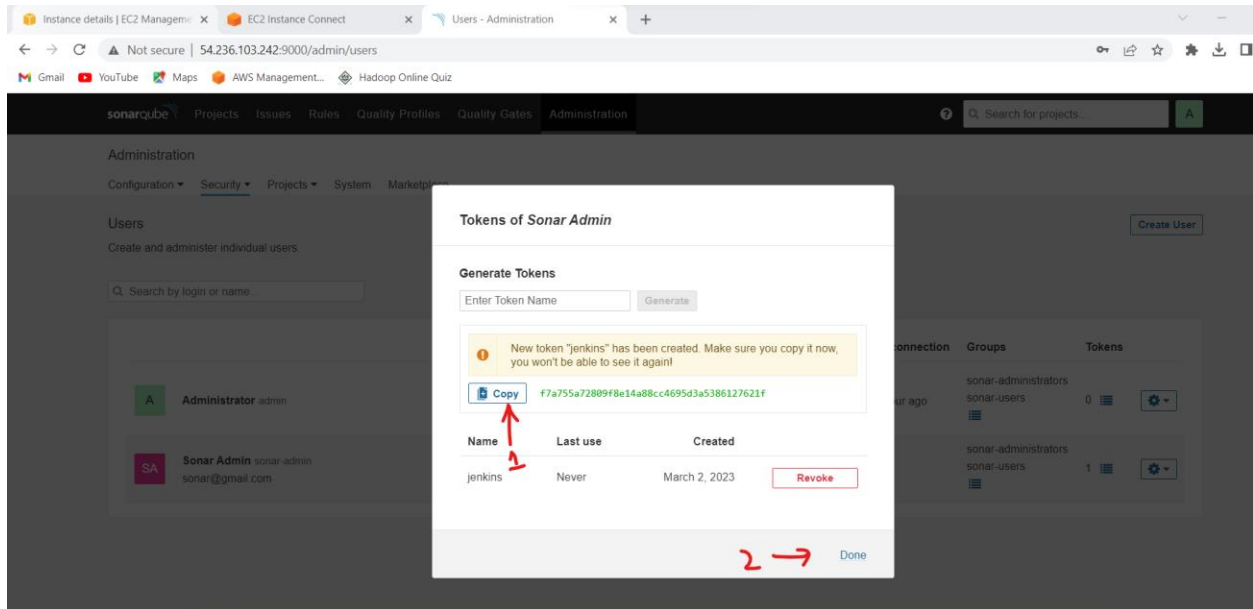
Generate Tokens

jenkins Generate

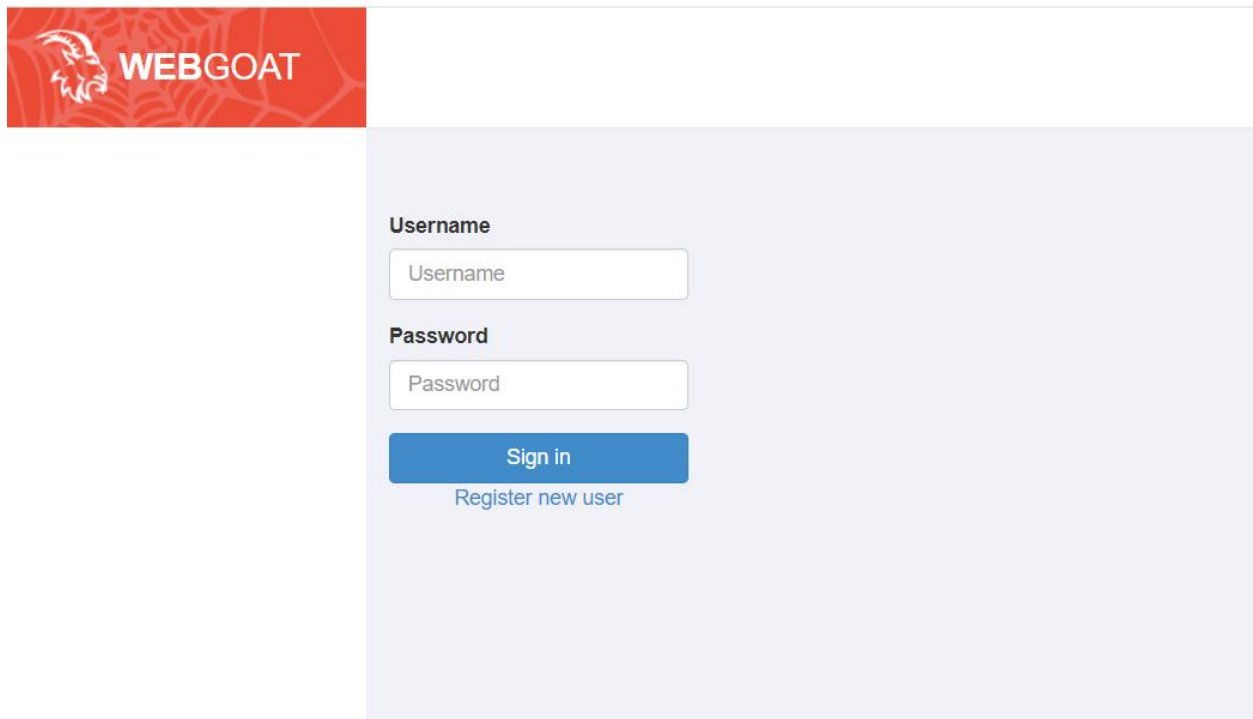
Name	Last use	Created
No tokens		

Done

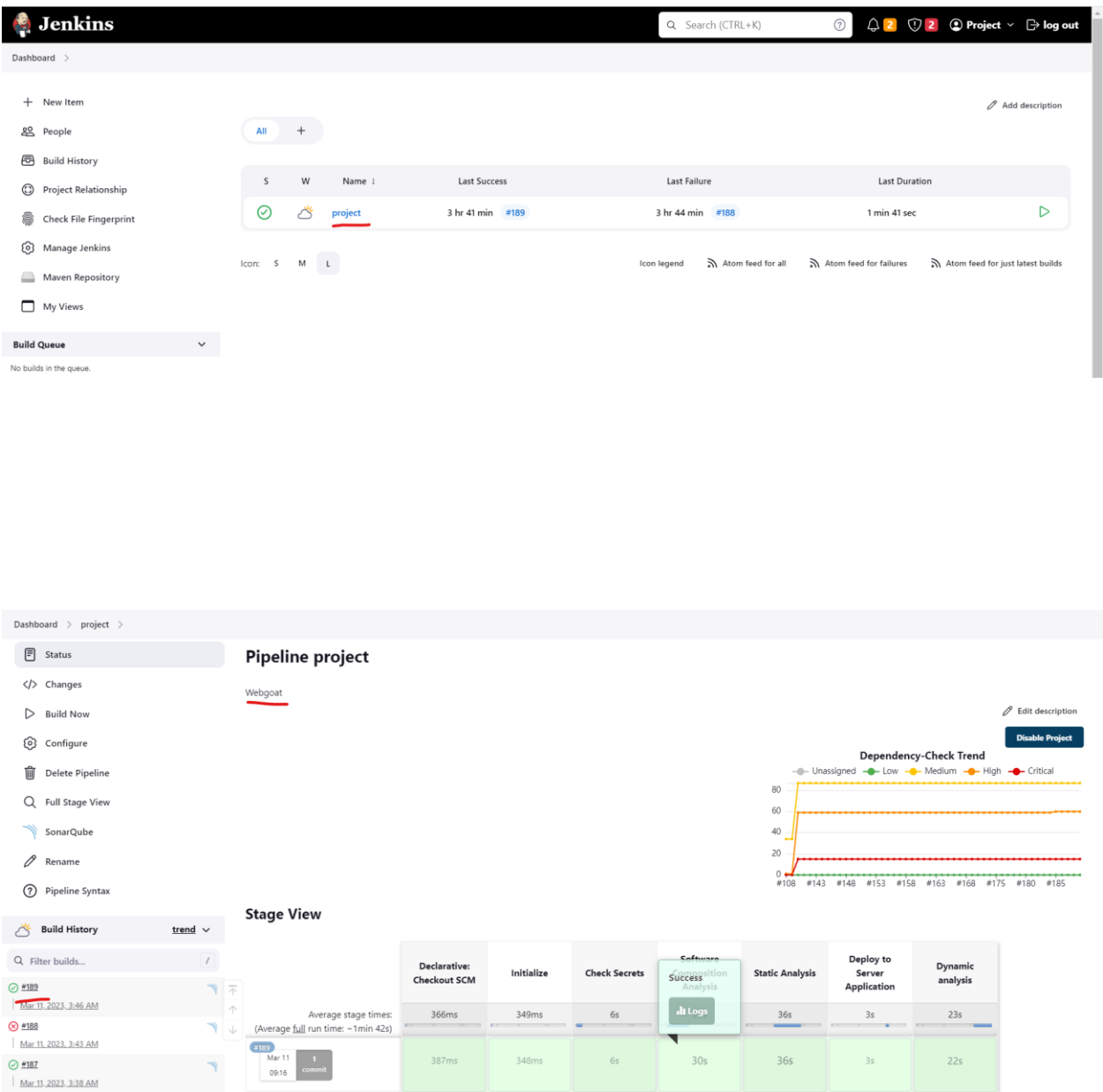
2 of 2 shown



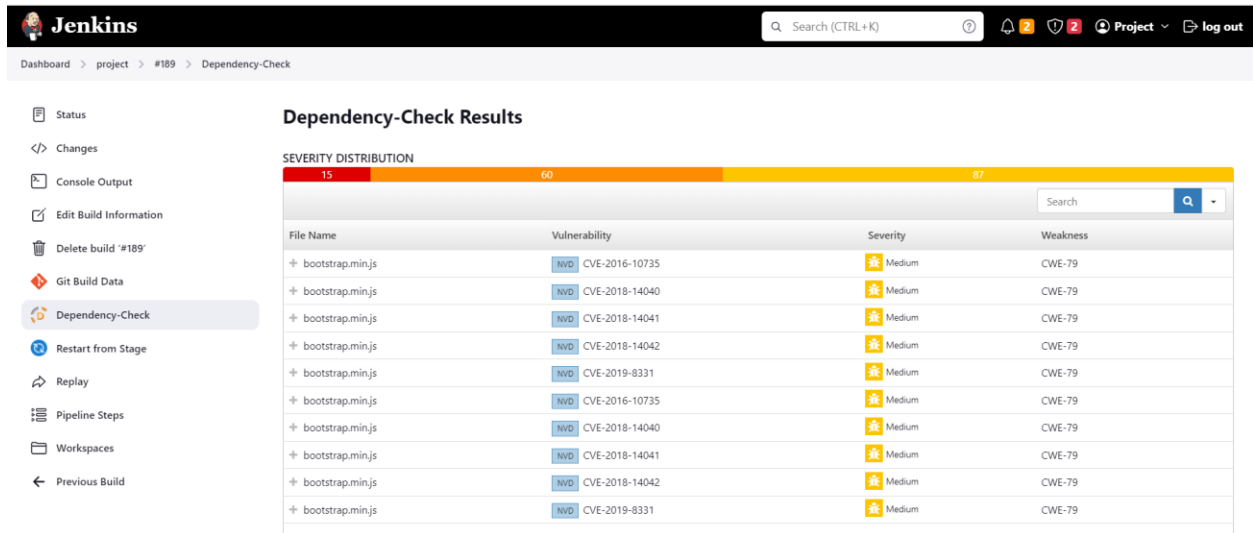
7. Dashboard of WebGoat Application



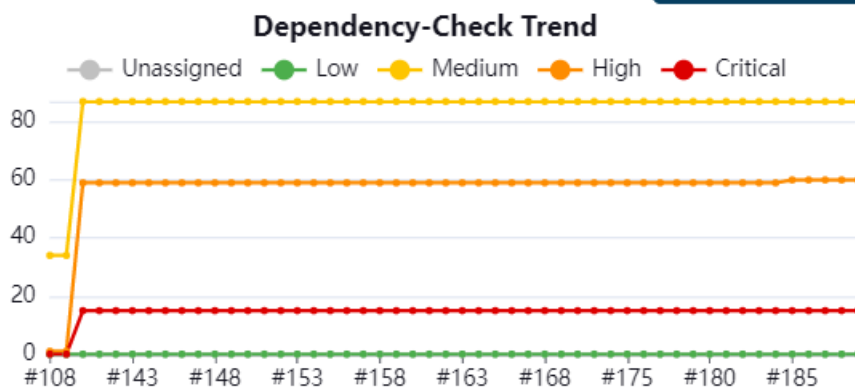
8. Dashboard of Jenkins Server after Project Build



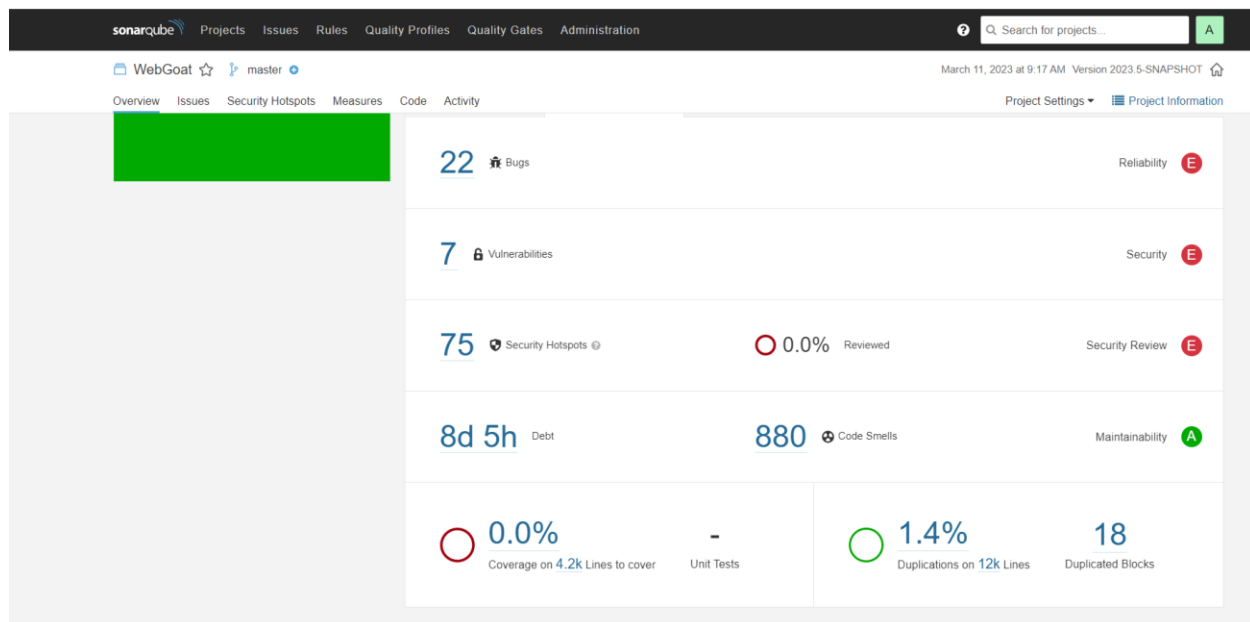
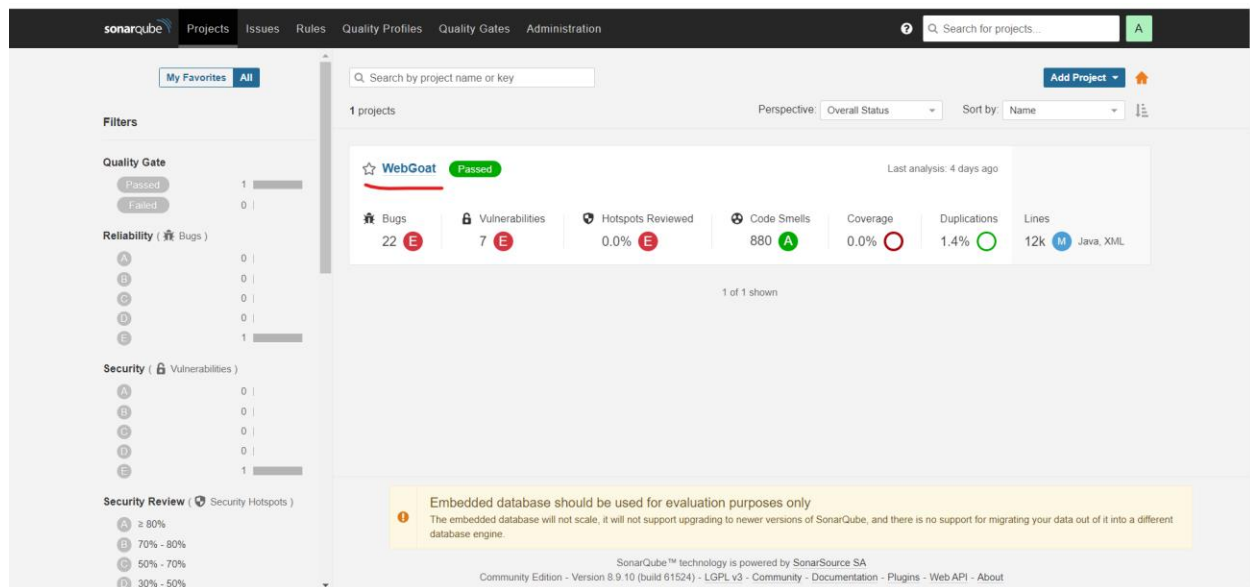
8.1 Dependency-Check Results



8.2 Dependency-Check Trend Status

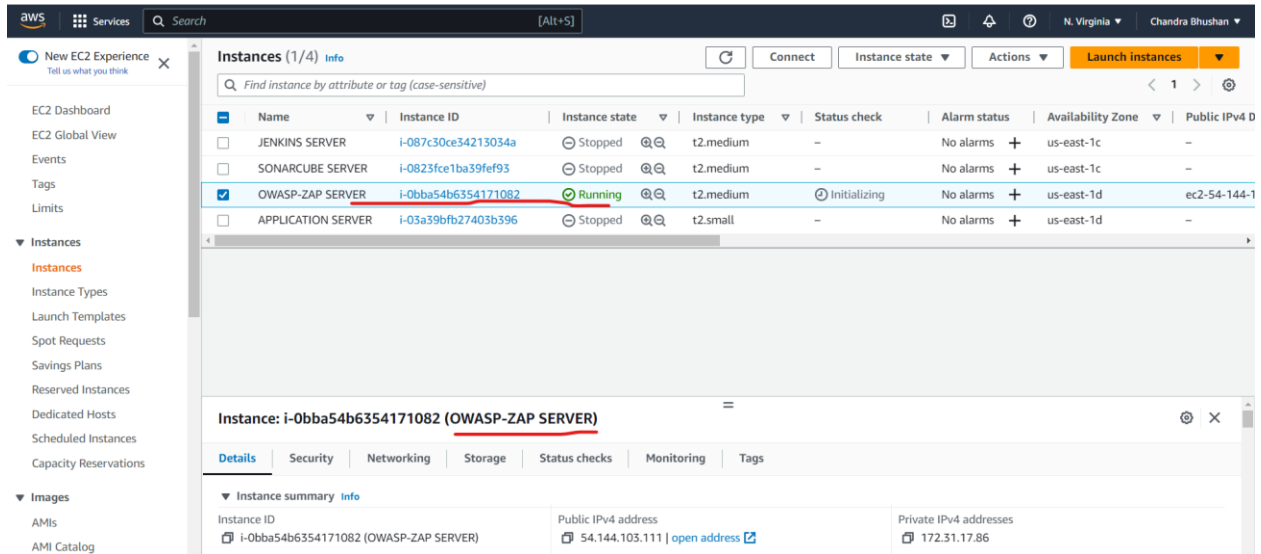


8.3 Dashboard of SonarQube after checked Bugs and Vulnerability



OWASP-ZAP Server check Scanned Report

9.1 Dashbord of OWASP-ZAP Server



The screenshot displays the AWS Management Console interface for EC2 instances. The left sidebar shows navigation options like 'EC2 Dashboard', 'Events', 'Tags', 'Limits', and 'Instances'. The main content area shows a list of instances with columns for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IPv4 D. The 'OWASP-ZAP SERVER' instance is selected and highlighted in blue. Below the table, the details for this instance are shown, including its public IP address (54.144.103.111) and private IP address (172.31.17.86).

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 D
JENKINS SERVER	i-087c30ce34213034a	Stopped	t2.medium	-	No alarms	us-east-1c	-
SONARCUBE SERVER	i-0823fce1ba39fef93	Stopped	t2.medium	-	No alarms	us-east-1c	-
OWASP-ZAP SERVER	i-0bba54b6354171082	Running	t2.medium	Initializing	No alarms	us-east-1d	ec2-54-144-1
APPLICATION SERVER	i-03a39bfb27403b396	Stopped	t2.small	-	No alarms	us-east-1d	-

Instance: i-0bba54b6354171082 (OWASP-ZAP SERVER)

Details | Security | Networking | Storage | Status checks | Monitoring | Tags

Instance summary info

Instance ID: i-0bba54b6354171082 (OWASP-ZAP SERVER)

Public IPv4 address: 54.144.103.111 | [open address](#)

Private IPv4 addresses: 172.31.17.86

9.2 Check Application Scanned Report

```
ubuntu@ip-172-31-17-86:~$ pwd
```

```
ubuntu@ip-172-31-17-86:~$ ls
```

```
zap_report.xml
```

```
ubuntu@ip-172-31-17-86:~$ cat zap_report.xml
```

9.3 Screen shoot of zap_report.xml

```
aws  Services  🔍 Search  [Alt+S]  N. Virginia  Chandra Bhushan  📌 🔔 ⚙️
```

```
ubuntu@ip-172-31-17-86:~$ pwd
/home/ubuntu
ubuntu@ip-172-31-17-86:~$ ls
zap_report.xml
ubuntu@ip-172-31-17-86:~$ cat zap_report.xml
<?xml version="1.0"?>
<OWASPZAPReport programName="OWASP ZAP" version="2.12.0" generated="Wed, 8 Mar 2023 18:53:06">

  <site name="http://3.110.49.177:8085" host="3.110.49.177" port="8085" ssl="false">
    <alerts>

      <alertitem>
        <pluginid>10202</pluginid>
        <alertRef>10202</alertRef>
        <alert>Absence of Anti-CSRF Tokens</alert>
        <name>Absence of Anti-CSRF Tokens</name>
        <riskcode></riskcode>
        <confidence></confidence>
        <riskdesc>Medium (Low)</riskdesc>
        <confidenceesc>Low</confidenceesc>
        <desc><?xml:space preserve="preserve">No Anti-CSRF tokens were found in a HTML submission form.</?xml:space><?xml:space preserve="preserve">A cross-site request forgery is an attack that involves forcing a victim to send an HTTP request to a target destination without their knowledge or intent in order to perform an action as the victim. The underlying cause is application functionality using predictable URL/form actions in a repeatable way. The nature of the attack is that CSRF exploits the trust that a web site has for a user. By contrast, cross-site scripting (XSS) exploits the trust that a user has for a web site. Like XSS, CSRF attacks are not necessarily cross-site, but they can be. Cross-site request forgery is also known as CSRF, XSRF, one-click attack, session riding, confused deputy, and sea surf.</?xml:space></desc>
      </alertitem>
    </alerts>
  </site>
</OWASPZAPReport>

```

i-Obba54b6354171082 (OWASP-ZAP SERVER)

PublicIPs: 54.144.103.111 PrivateIPs: 172.31.17.86

```
[aws] [Services] [Search] [Alt+] [Alt+] N. Virginia v Chandra Bhushan v
```

```
<confidence><desc>Low</confidence>  
<desc><!-- No Anti-CSRF tokens were found in a HTML submission form.<br/>A cross-site request forgery is a  
n attack that involves forcing a victim to send an HTTP request to a target destination without their knowledge or intent in order to perform an action as the victim. The u  
nderlying cause is application functionality using predictable URL/form actions in a repeatable way. The nature of the attack is that CSRF exploits the trust that a web sit  
e has for a user. By contrast, cross-site scripting (XSS) exploits the trust that a user has for a web site. Like XSS, CSRF attacks are not necessarily cross-site, but they  
can be. Cross-site request forgery is also known as CSRF, XSRF, one-click attack, session riding, confused deputy, and sea surf.<br/>CSRF attacks are effective in a number of situations, including:<br/>* The victim has an active session on the target site.<br/>* The victim  
is authenticated via HTTP auth on the target site.<br/>* The victim is on the same local network as the target site.<br/>CSRF has primarily been used to perform an action against a target site using the victims's privileges, but recent techniques have been discovered to disclose informati  
on by gaining access to the response. The risk of information disclosure is dramatically increased when the target site is vulnerable to XSS, because XSS can be used as a p  
latform for CSRF, allowing the attack to operate within the bounds of the same-origin policy.<br/>
```

```
aws  Services  Search  [Alt+]  N. Virginia  Chandra Bhushan

<alertitem>
  <pluginid>10049</pluginid>
  <alertRef>10049</alertRef>
  <alert>Storable and Cacheable Content</alert>
  <name>Storable and Cacheable Content</name>
  <riskcode>0</riskcode>
  <confidence>2</confidence>
  <riskdesc>Informational (Medium)</riskdesc>
  <confidenceDesc>Medium</confidenceDesc>
  <desc><!--The response contents are storable by caching components such as proxy servers, and may be retrieved directly
from the cache, rather than from the origin server by the caching servers, in response to similar requests from other users. If the response data is sensitive, personal o
r user-specific, this may result in sensitive information being leaked. In some cases, this may even result in a user gaining complete control of the session of another use
r, depending on the configuration of the caching components in use in their environment. This is primarily an issue where "shared" caching servers such as "otip
roxy" caches are configured on the local network. This configuration is typically found in corporate or educational environments, for instance.</p></desc>
  <instances>
    <instance>
      <uri>http://3.110.49.177:8085</uri>
      <method>GET</method>
      <param></param>
      <attack></attack>
      <evidence></evidence>
    </instance>
    <instance>
```

Conclusion

In conclusion, DevSecOps is a critical approach to software development in today's security-conscious environment. By incorporating security into every stage of the development process, organizations can create more secure and resilient software, reduce the time and cost associated with addressing security issues, and improve collaboration between development, security, and operations teams.

one of the main benefits of DevSecOps is that it helps to reduce the time and cost associated with addressing security issues. By catching security issues early on in the development process, organizations can avoid the high costs associated with fixing security issues in production. Additionally, by automating security testing and validation, organizations can reduce the time it takes to deploy software.

Another benefit of DevSecOps is that it helps to improve collaboration between development, security, and operations teams. By breaking down silos and bringing these teams together, organizations can create a more cohesive and efficient development process that is better equipped to address security risks.

Bibliography

- **Vulnerable project WebGoat**
<https://owasp.org/www-project-webgoat>
- **SonarqubeConfiguration**
<https://docs.sonarqube.org/latest>
- **Owasp-Zap tool Installation**
<https://www.zaproxy.org/getting-started>
- **TrufflehogScan Configuration**
<https://docs.trufflesecurity.com/docs/introduction/getting-started/index.html>
- **Jenkins Pipeline Configuration**
<https://www.jenkins.io/doc/tutorials>