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

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.

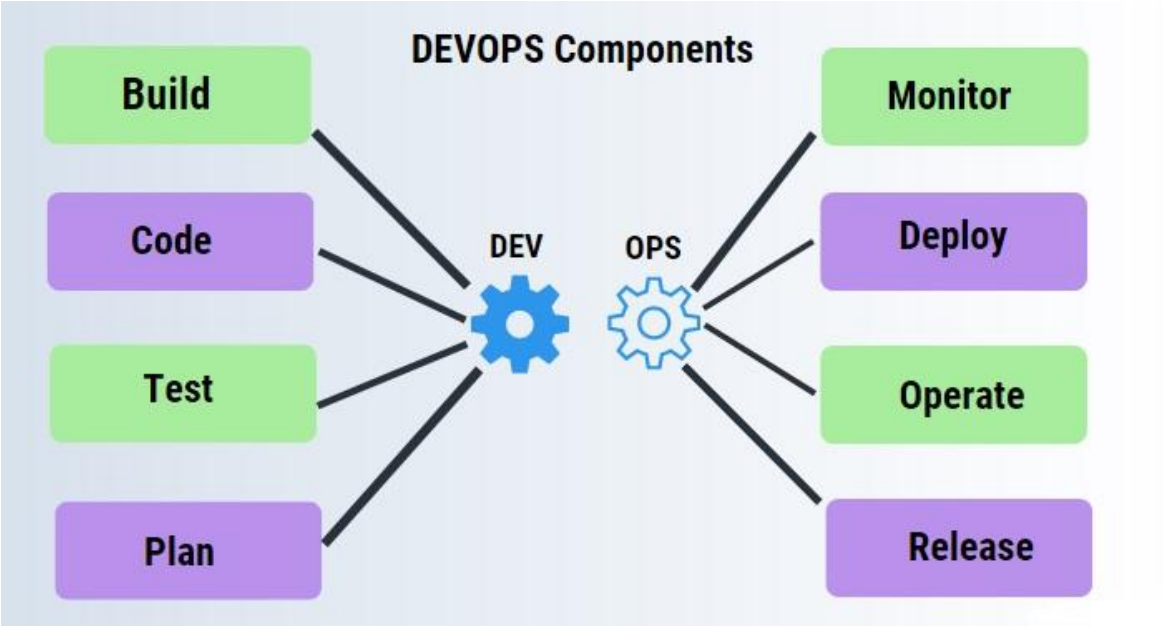
The OWASP-ZAP Top 10 is a standard awareness document for developers and web application security.

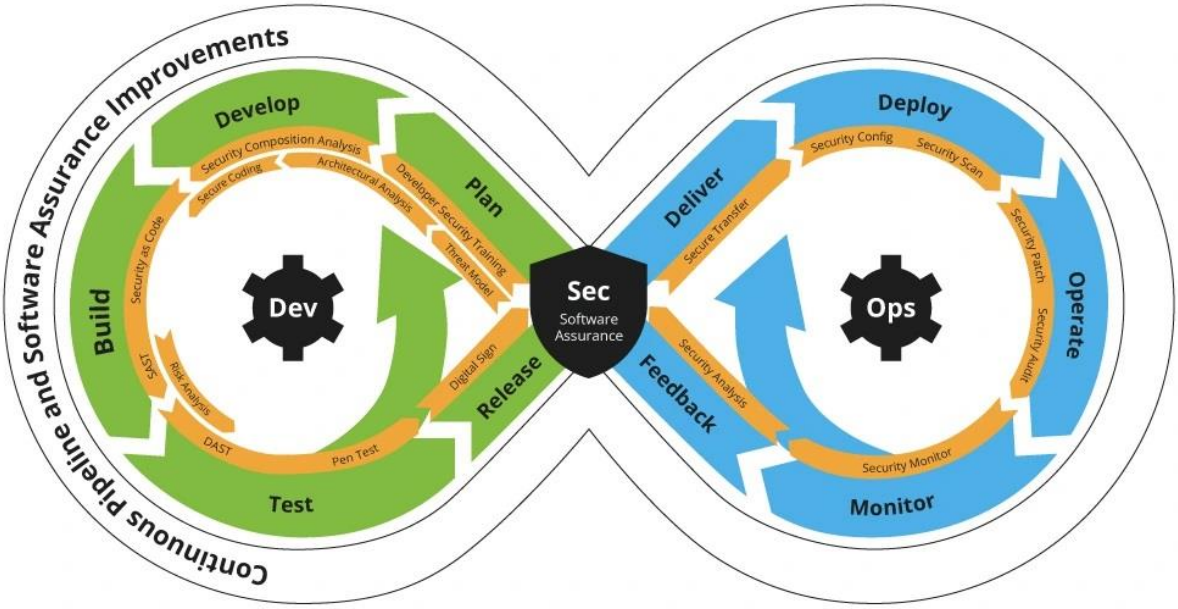
OWASP Top 10 - 2013		OWASP Top 10 - 2017
A1 – Injection	→	A1:2017-Injection
A2 – Broken Authentication and Session Management	→	A2:2017-Broken Authentication
A3 – Cross-Site Scripting (XSS)	↘	A3:2017-Sensitive Data Exposure
A4 – Insecure Direct Object References [Merged+A7]	∪	A4:2017-XML External Entities (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]	∪	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.

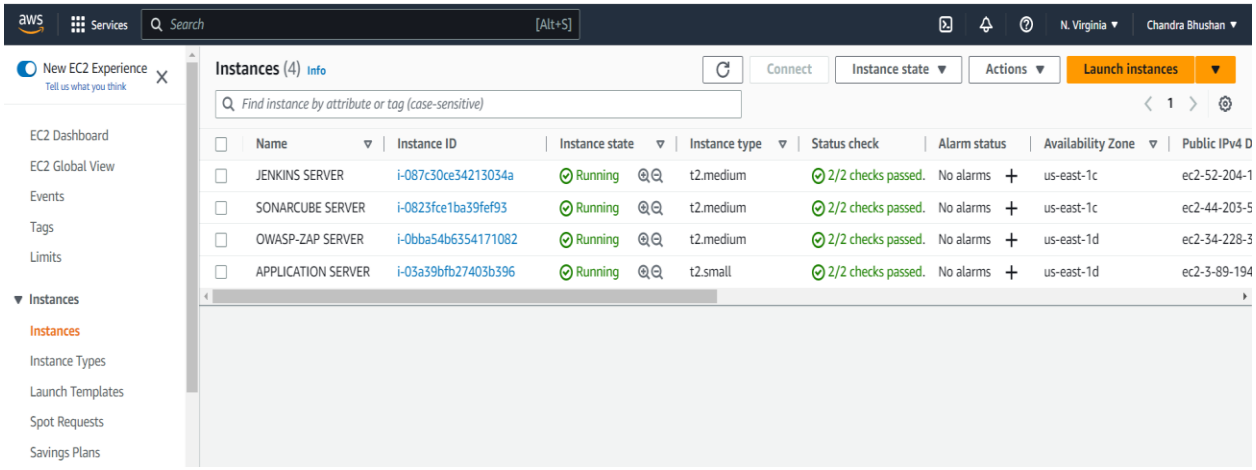
5. DevSecOps Architecture



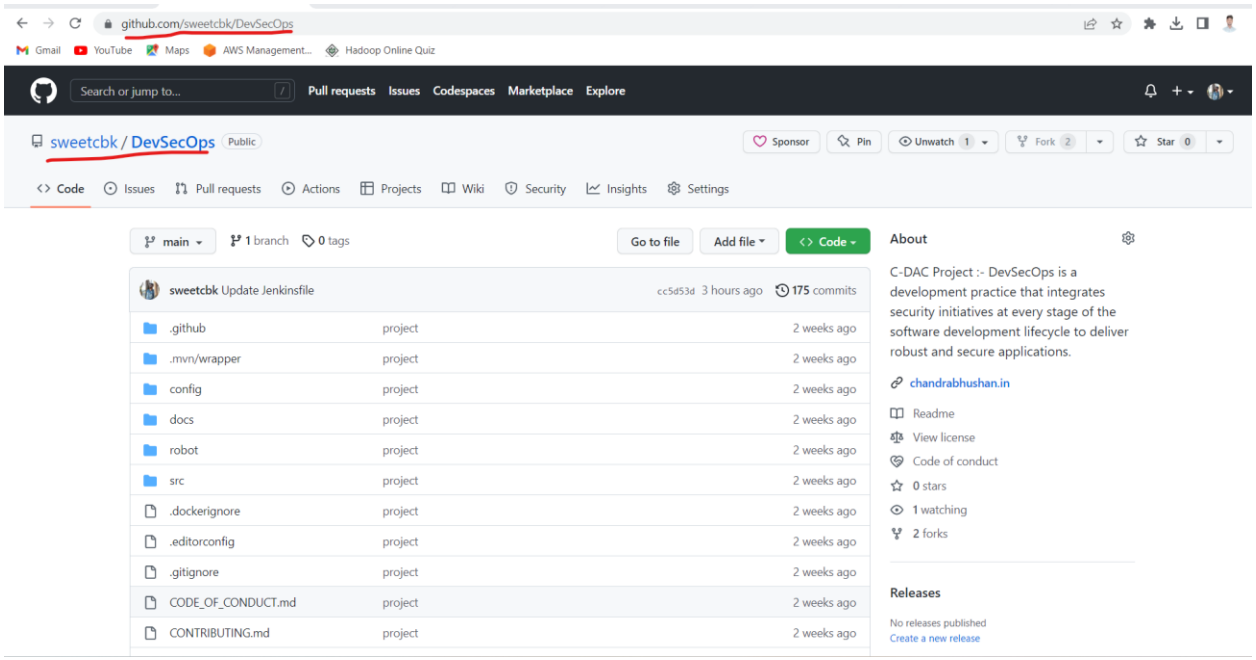


Project content

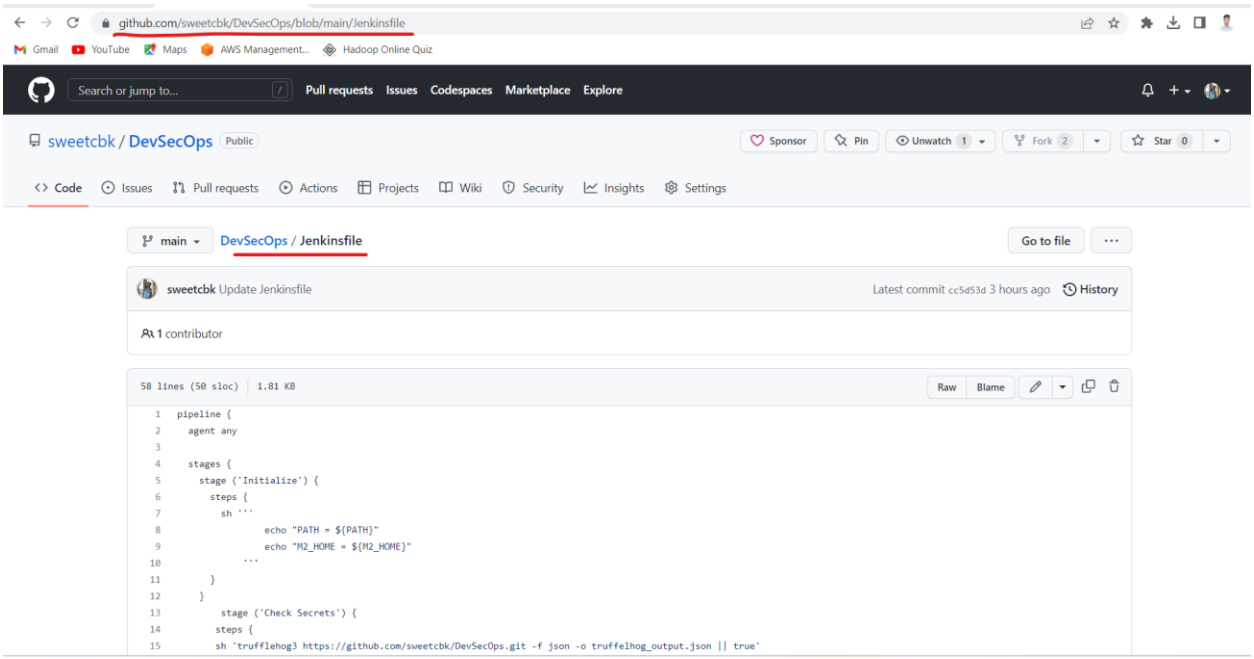
Created 4 AWS EC2 Ubuntu Instance forDevSecops



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



- **Jenkins File url :-** https://github.com/sweetcbk/DevSecOps/blob/main/Jenkinsfile

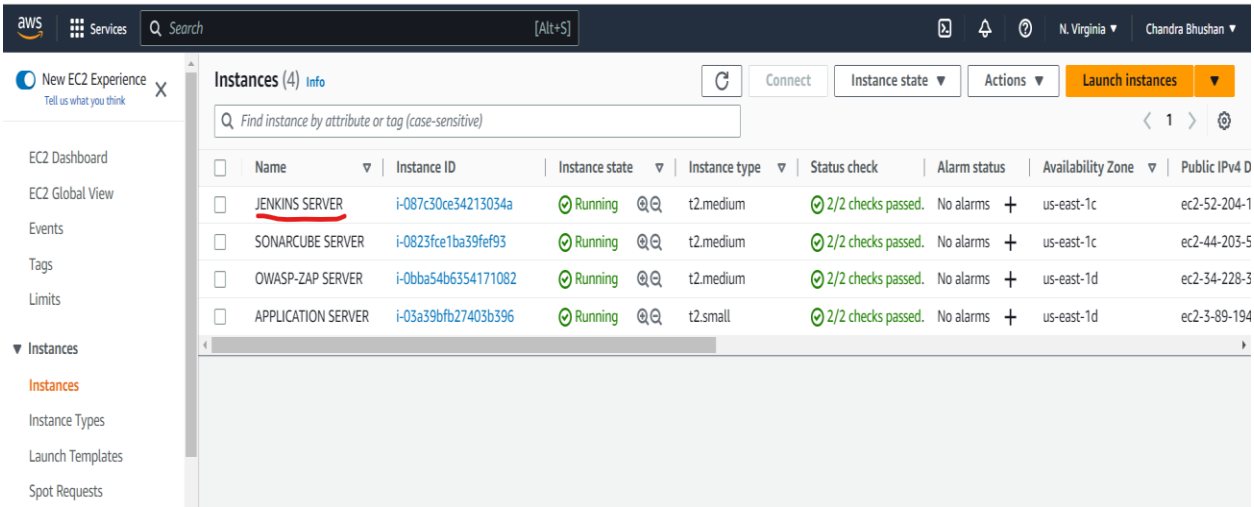


- **Jenkins File Script**

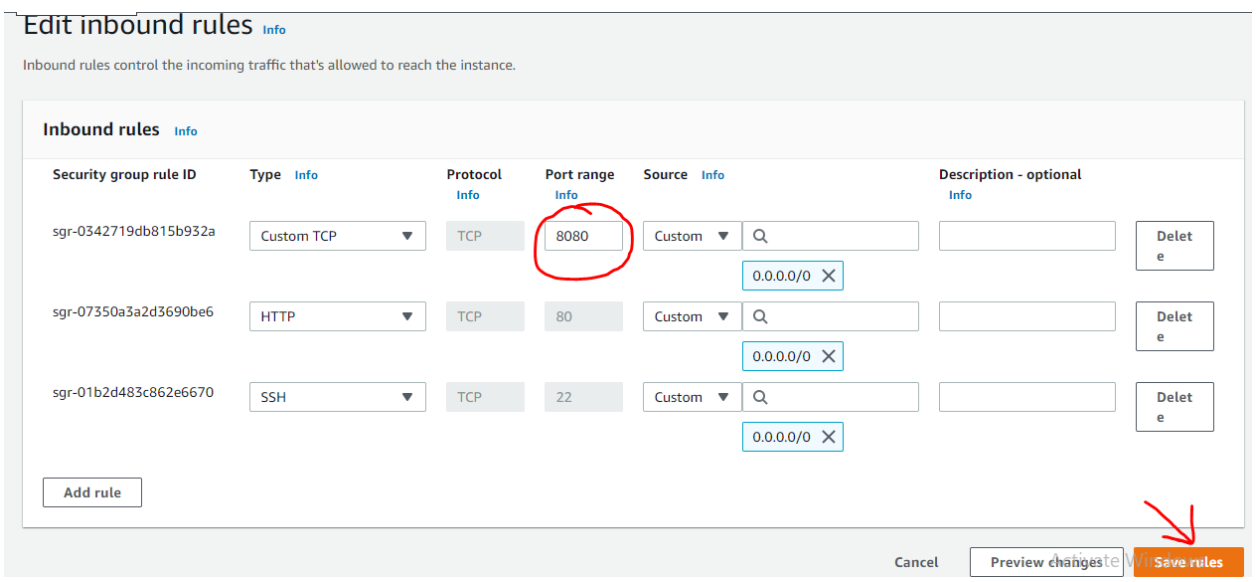
```
pipeline
{
             agent any
             stages {
               stage ('Initialize') {
                 steps {
           sh '''
                           echo "PATH = ${PATH}"
                           echo "M2_HOME = ${M2_HOME}"
                       '''
                 }
               }
                   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" '
              }
           }
    }
     }
    }
```

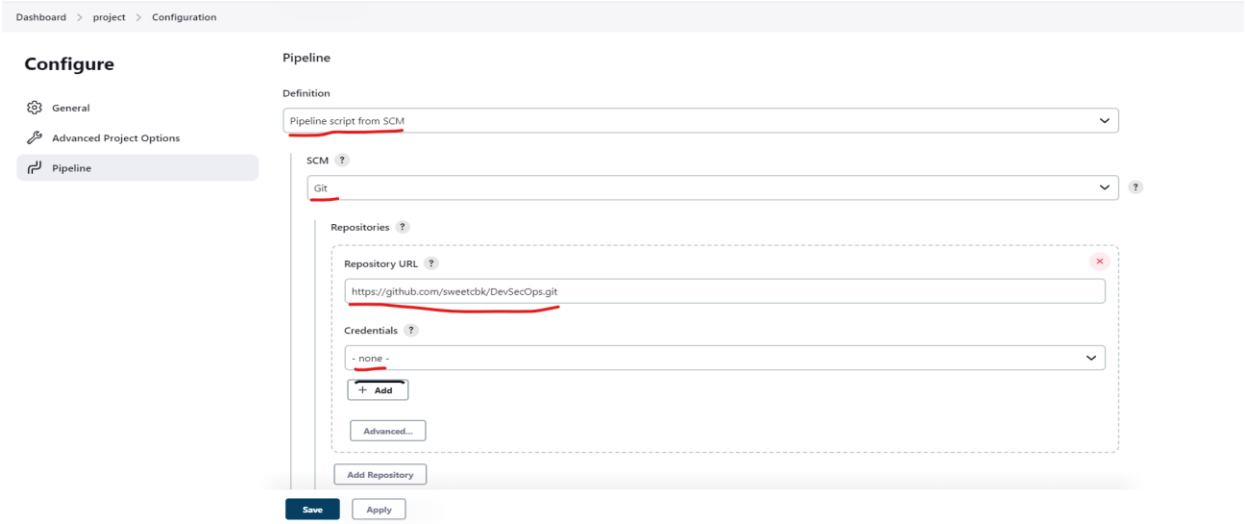
6.1 Configuration of Jenkins Server

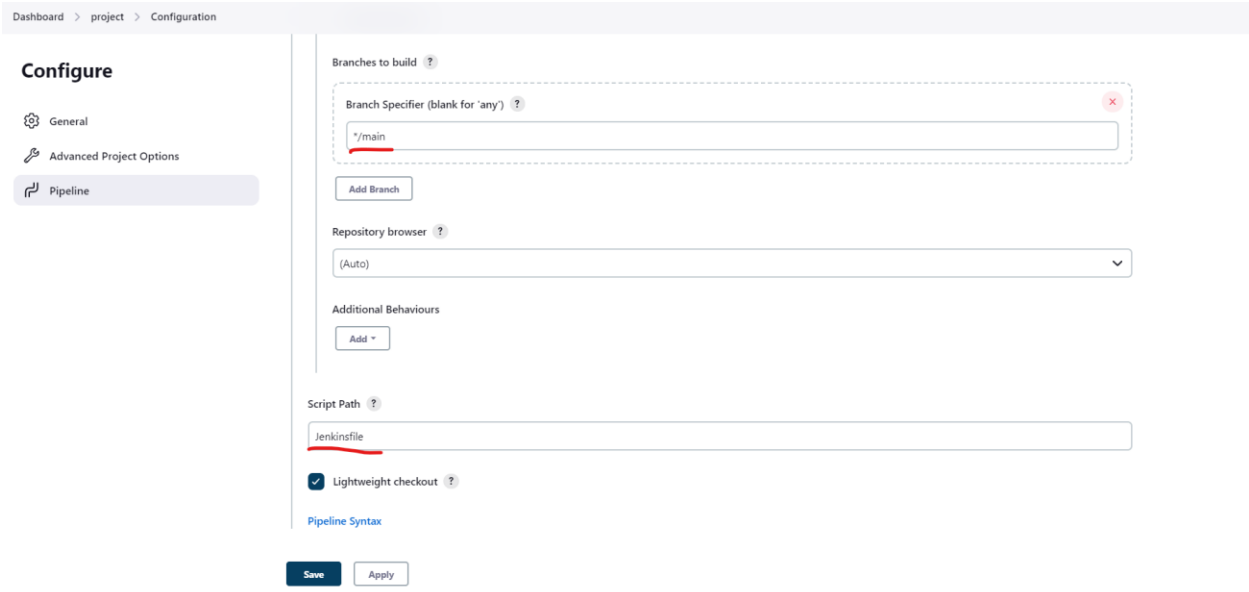


6.1.1 Add port 8080 in the Jenkins Server



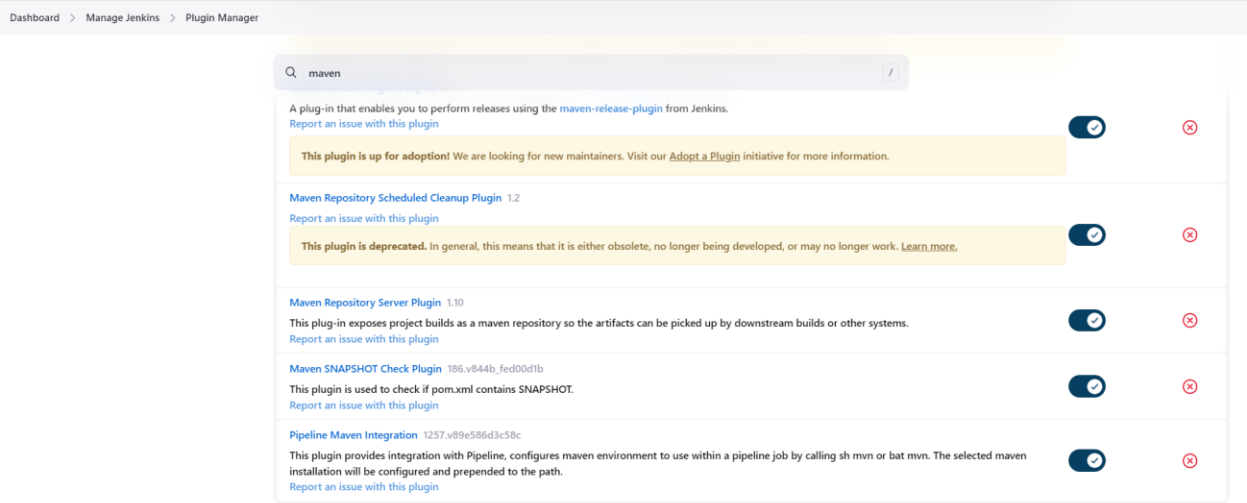
6.1.2 Jenkins Project Intigration with Github



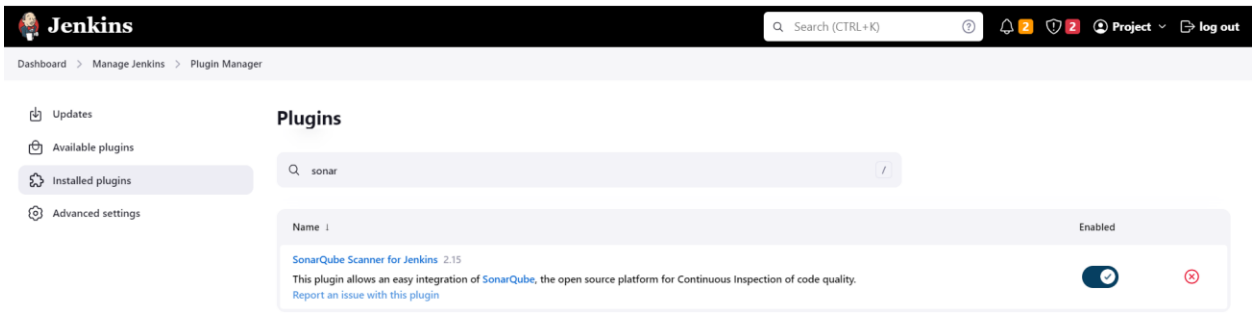


6.1.3 Install Jenkins Plugins

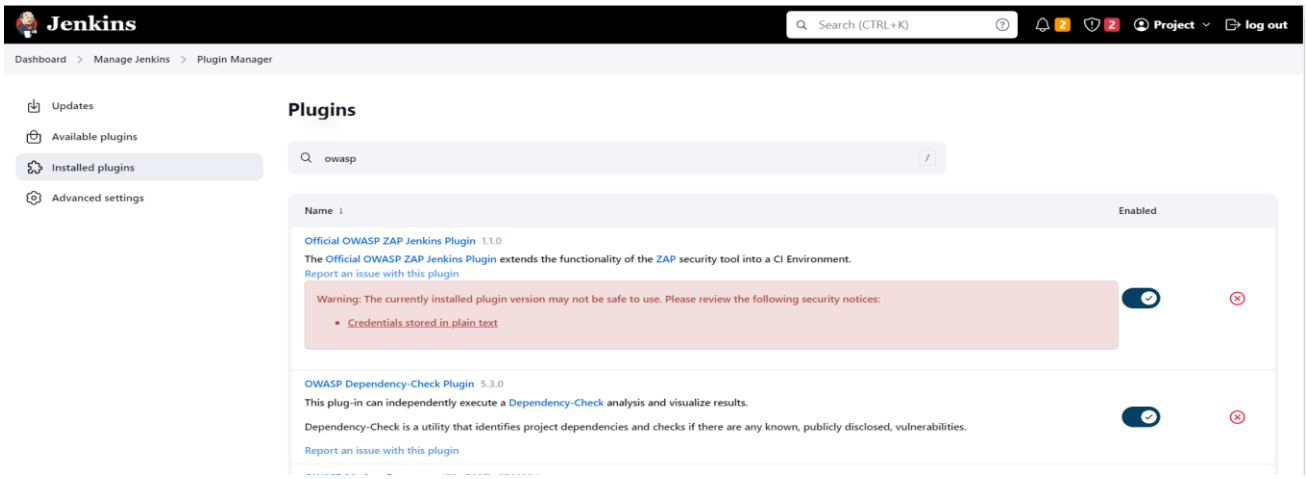
1. Maven SNAPSHOT Check Plugin
2. Pipeline Maven Integration



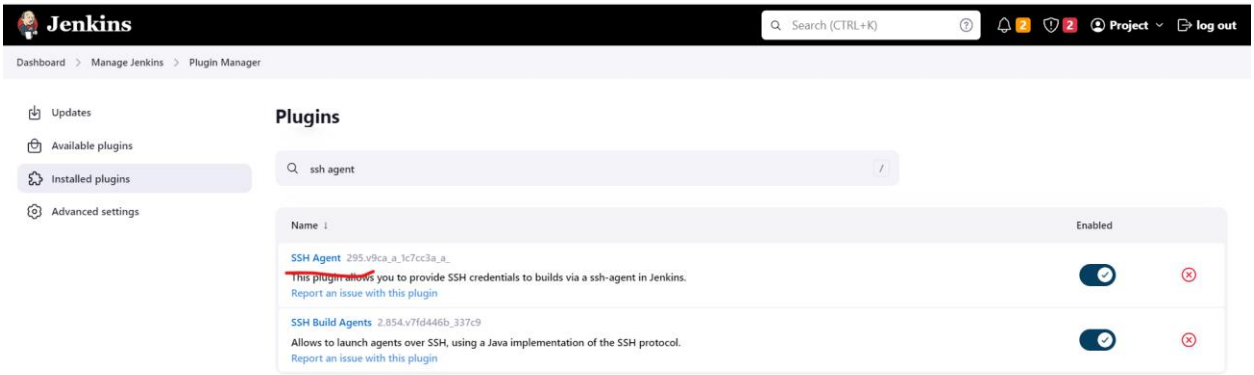
3. SonarQube Scanner for Jenkins



4. OWASP Dependency-Check Plugin

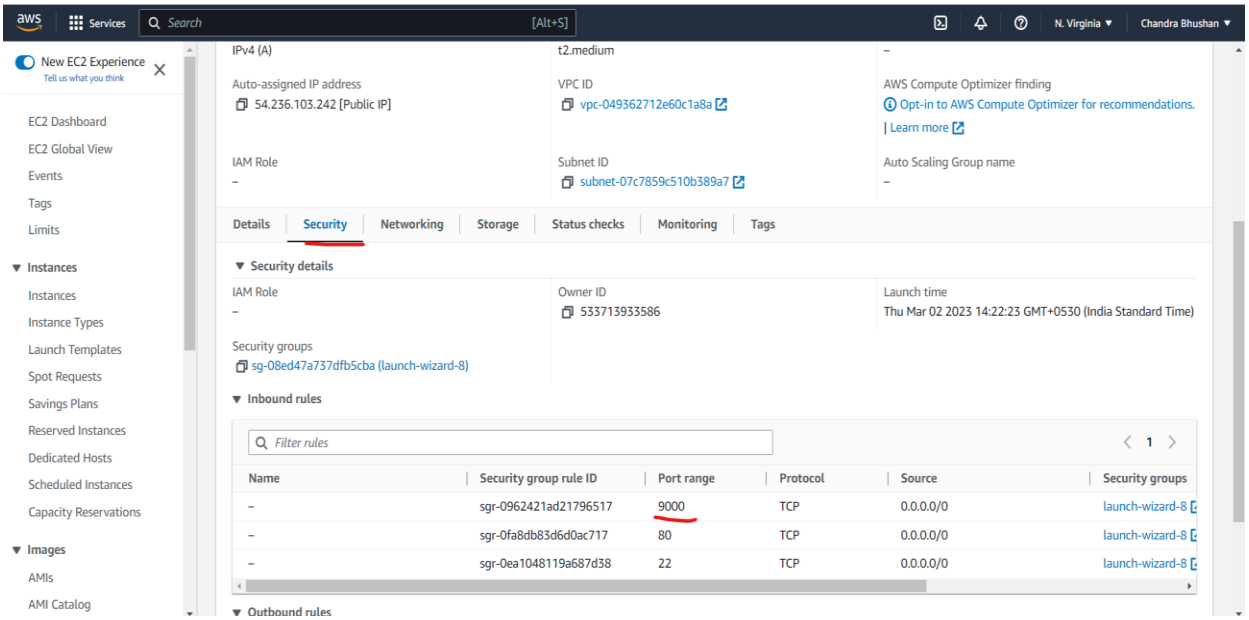


4. SSH Agent

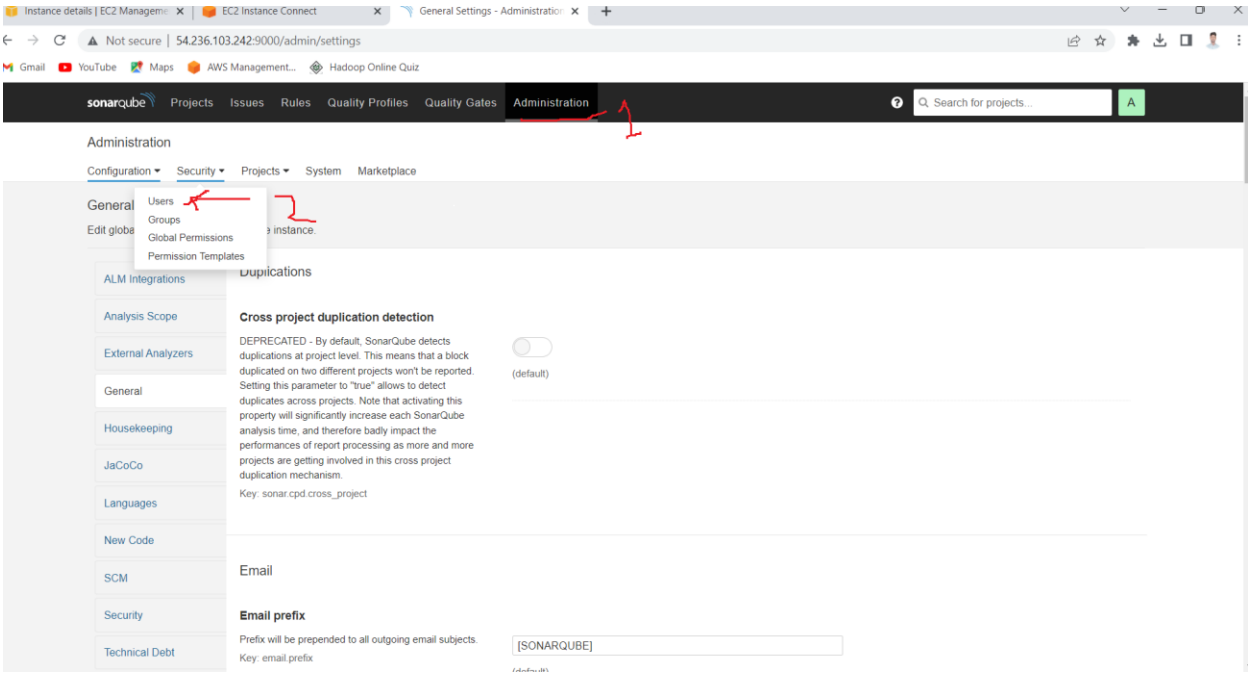


6.2 Configuration of SonarQube Server

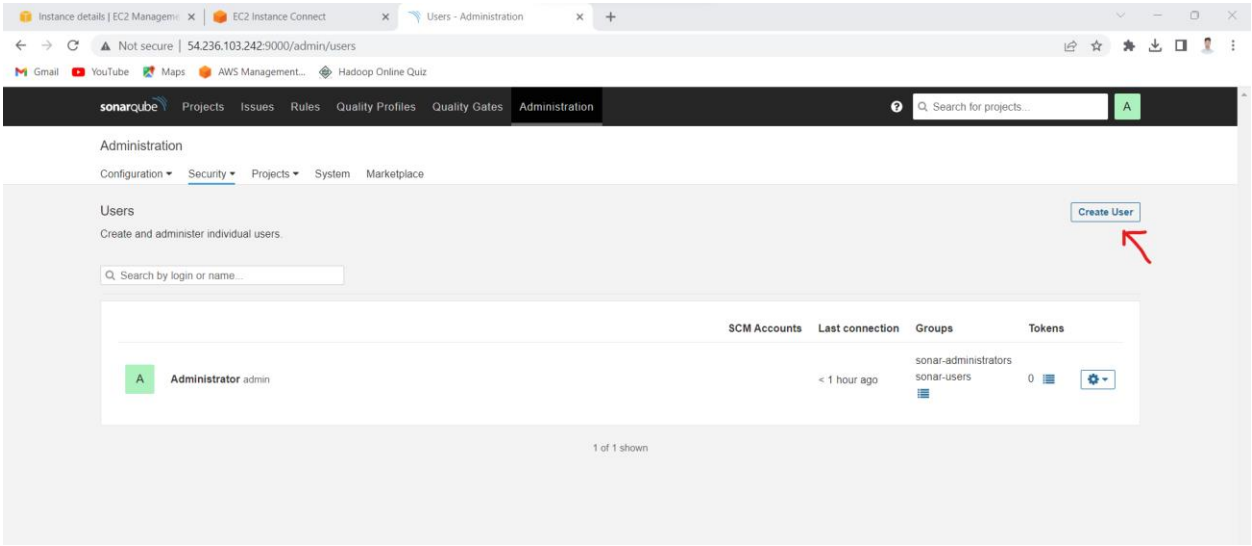
6.2.1 Add port 9000 in Sonarqube Server



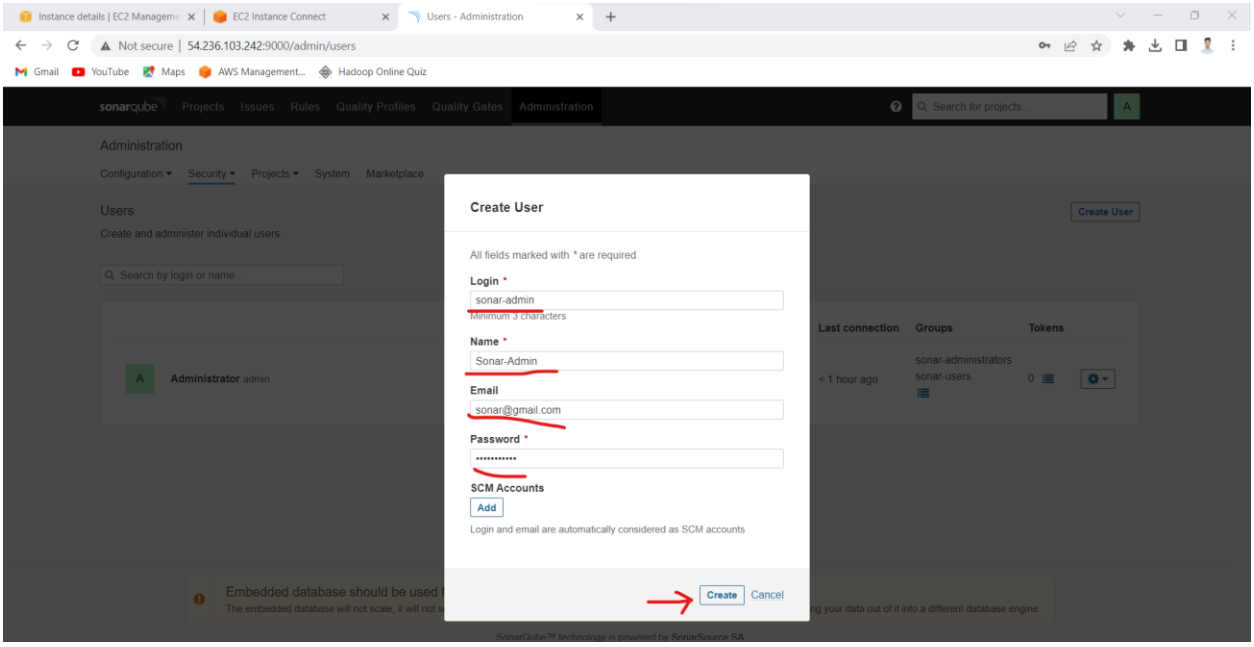
6.2.2 Add User in Sonarqube Server



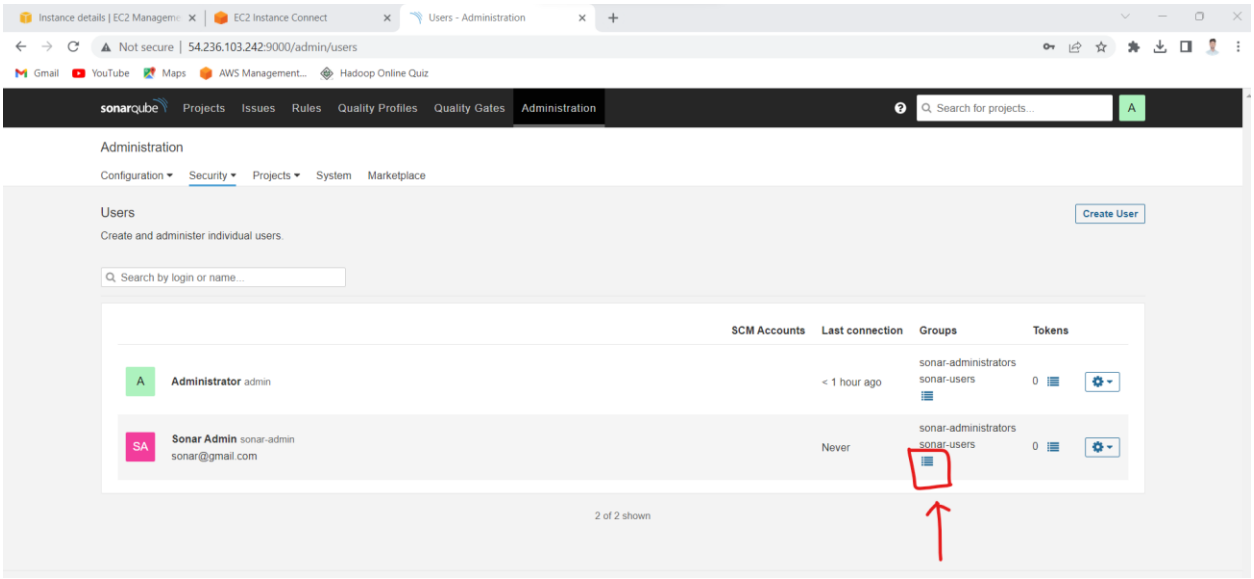
6.2.3 Create user in SonarQube Server

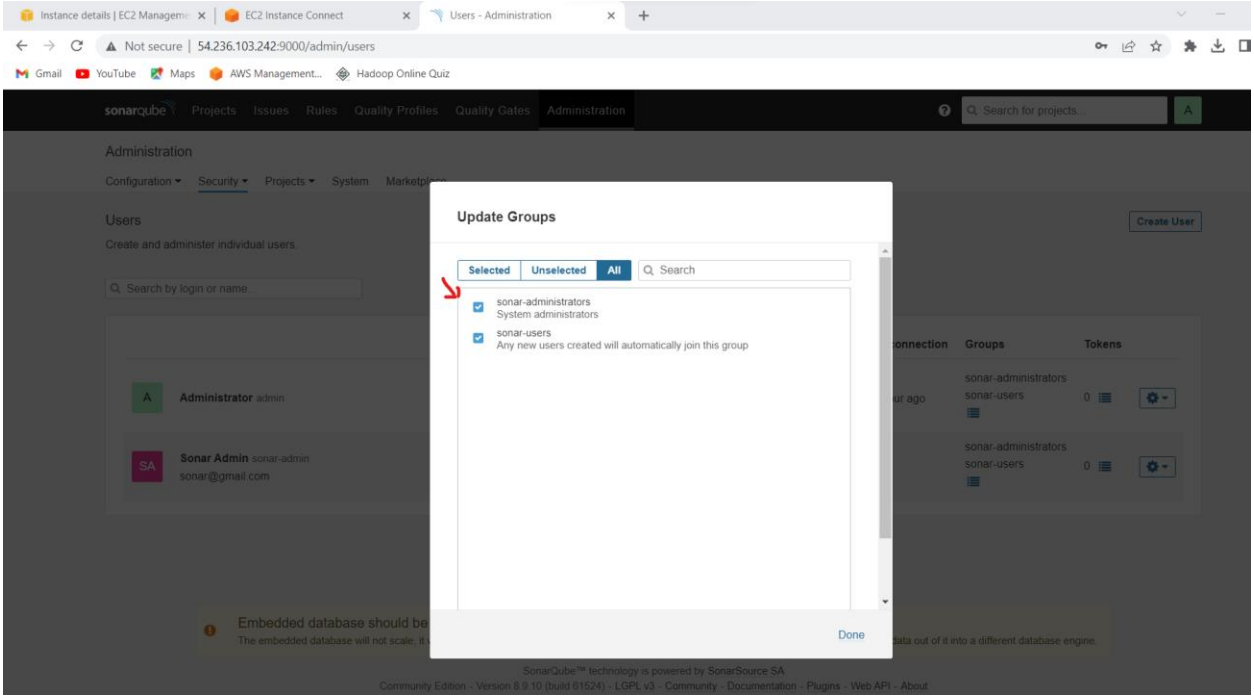


6.2.4 Set User-name and password in SonarQube Server

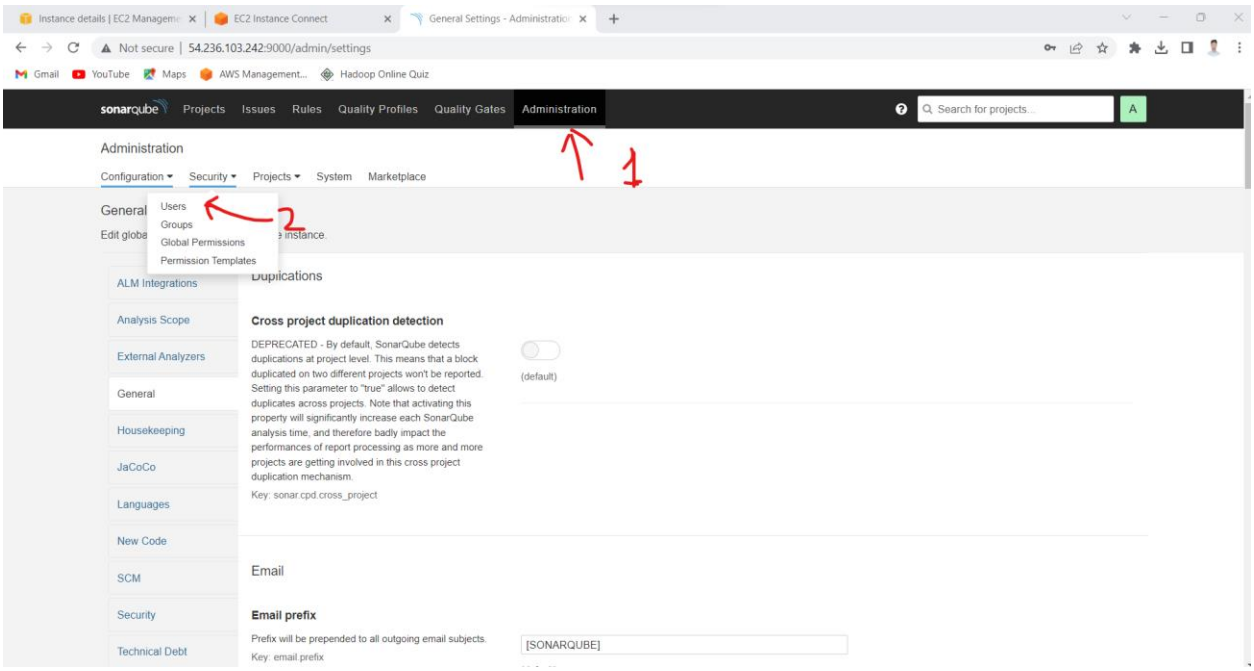


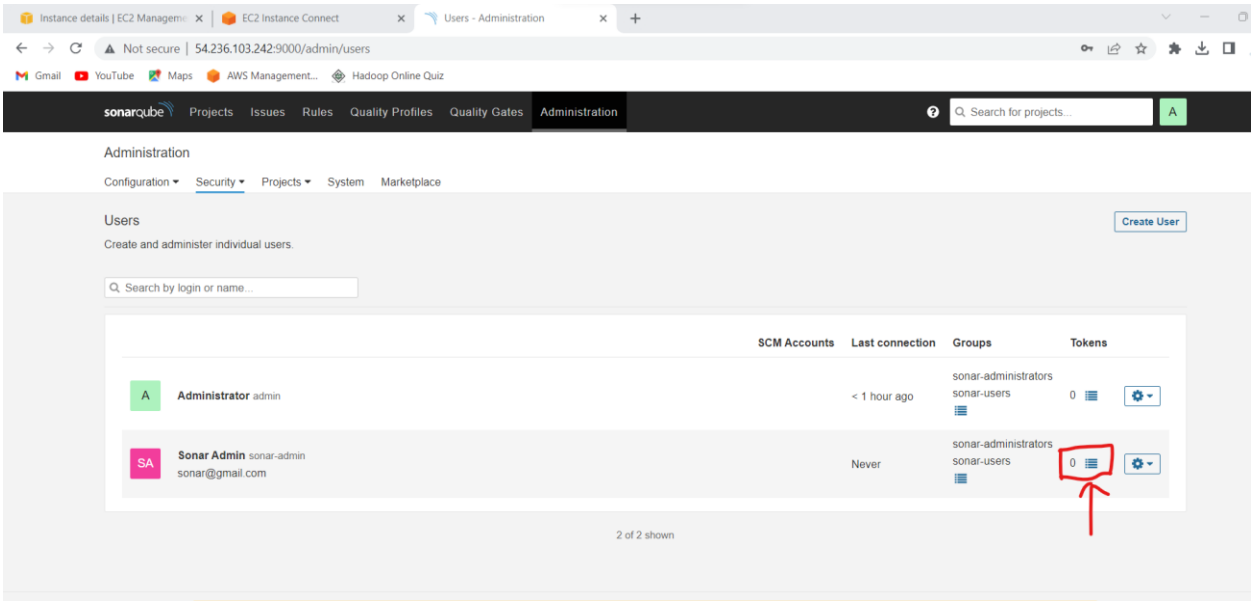
6.2.5 Set user as Adminitration in SonarQube Server

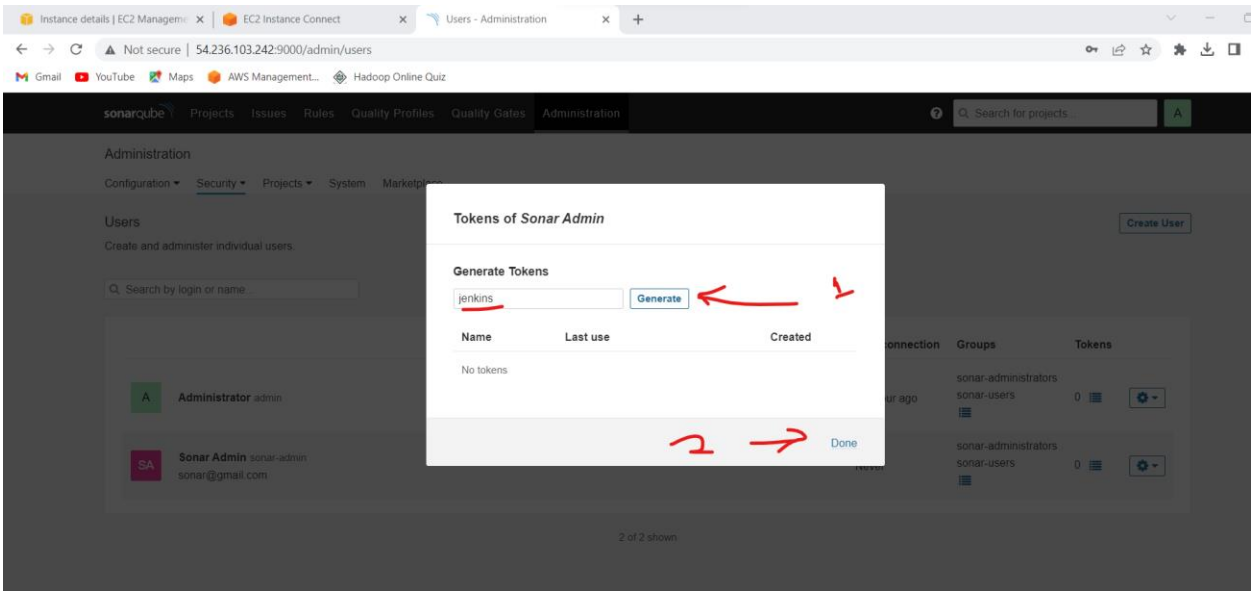


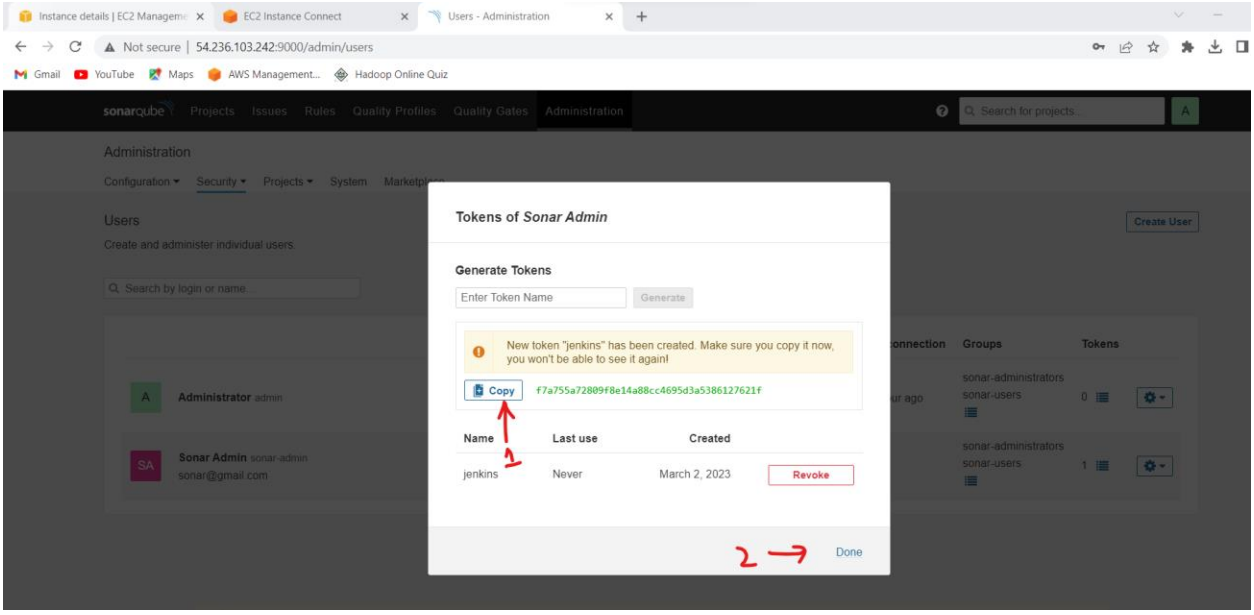


6.2.6 Create Jenkins credential for Jenkins Server in SonarQube Server

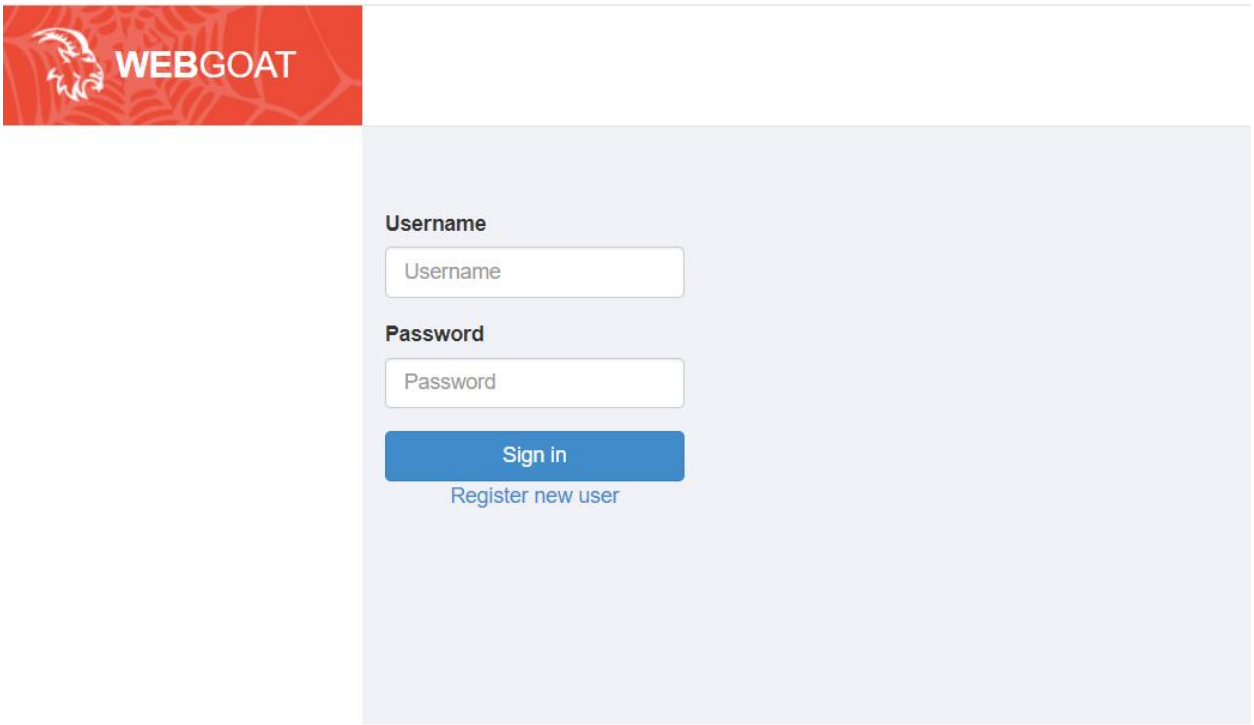



Instance details | EC2 Managem
EC2 Instance Connect
Users - Administration
Not secure | 54.236.103.242:9000/admin/users
Gmail
YouTube
Maps
AWS Management...
Hadoop Online Quiz
sonarqube
Projects
Issues
Rules
Quality Profiles
Quality Gates
Administration
Search for projects...
Administration
Configuration
Security
Projects
System
Marketplace
Users
Create and administer individual users.
Create User
Search by login or name...
SCM Accounts
Last connection
Groups
Tokens
Administrator admin
< 1 hour ago
sonar-administrators
sonar-users
Sonar Admin sonar-admin
sonar@gmail.com
Never
2 of 2 shown

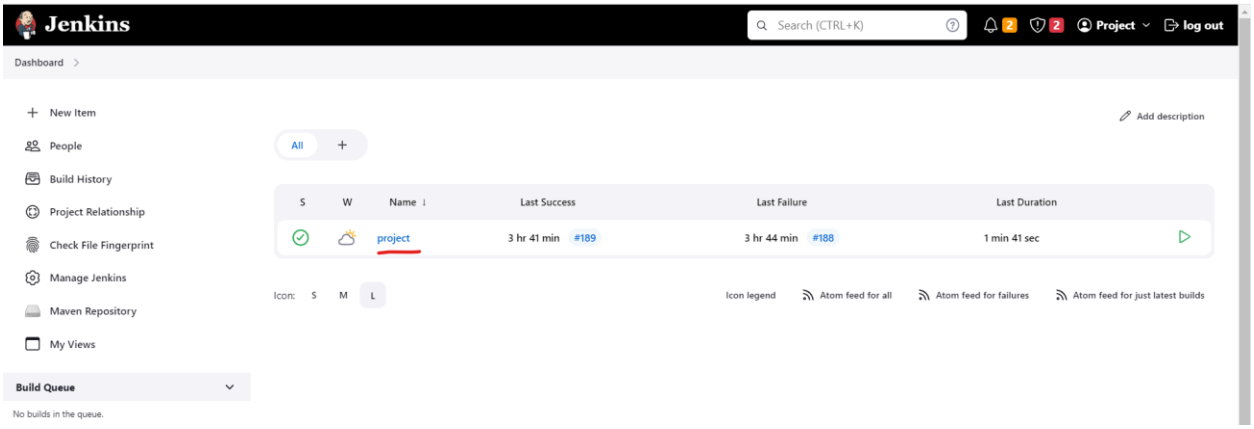

Tokens of Sonar Admin
Generate Tokens
jenkins
Generate
1
Name
Last use
Created
No tokens
2
Done

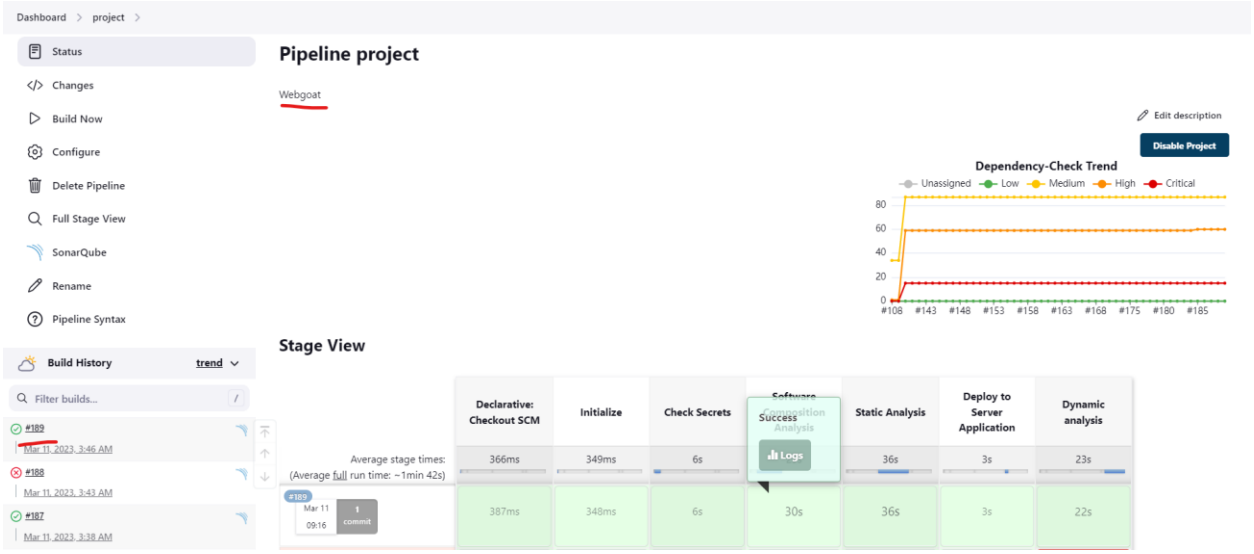


7. Dashboard of WebGoat Application

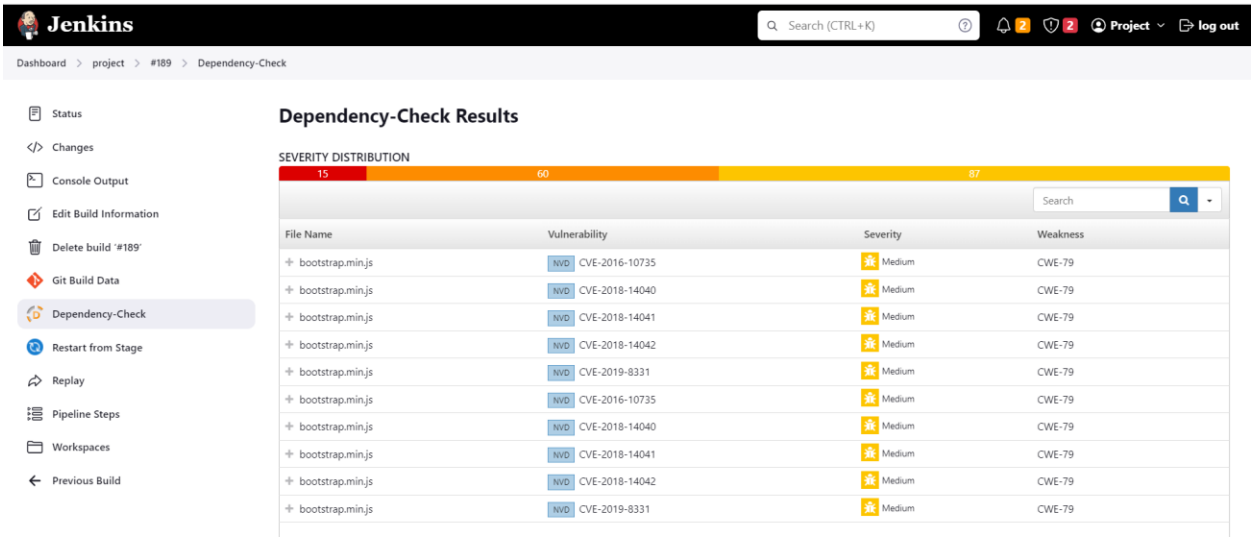


8. Dashboard of Jenkins Server after Project Build

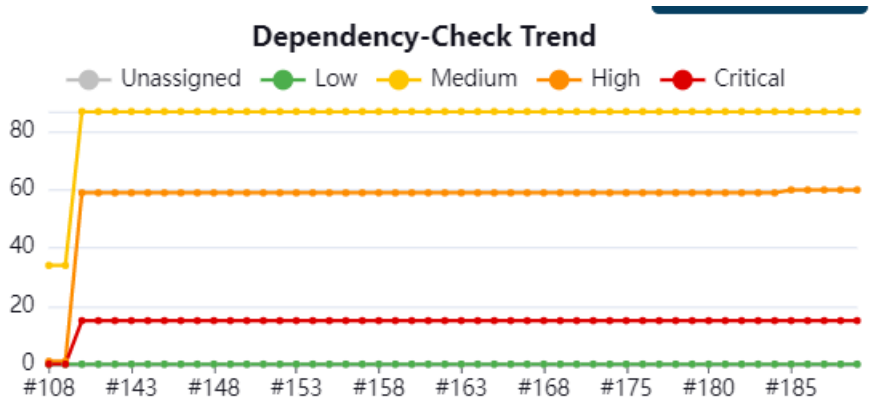




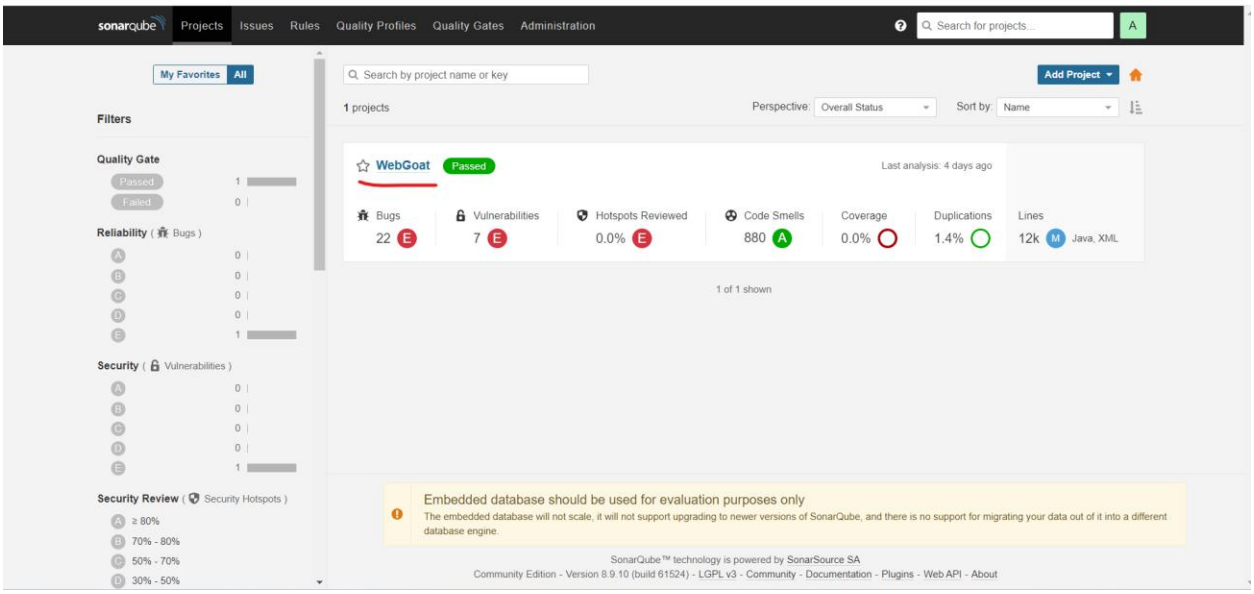
8.1 Dependency-Check Results

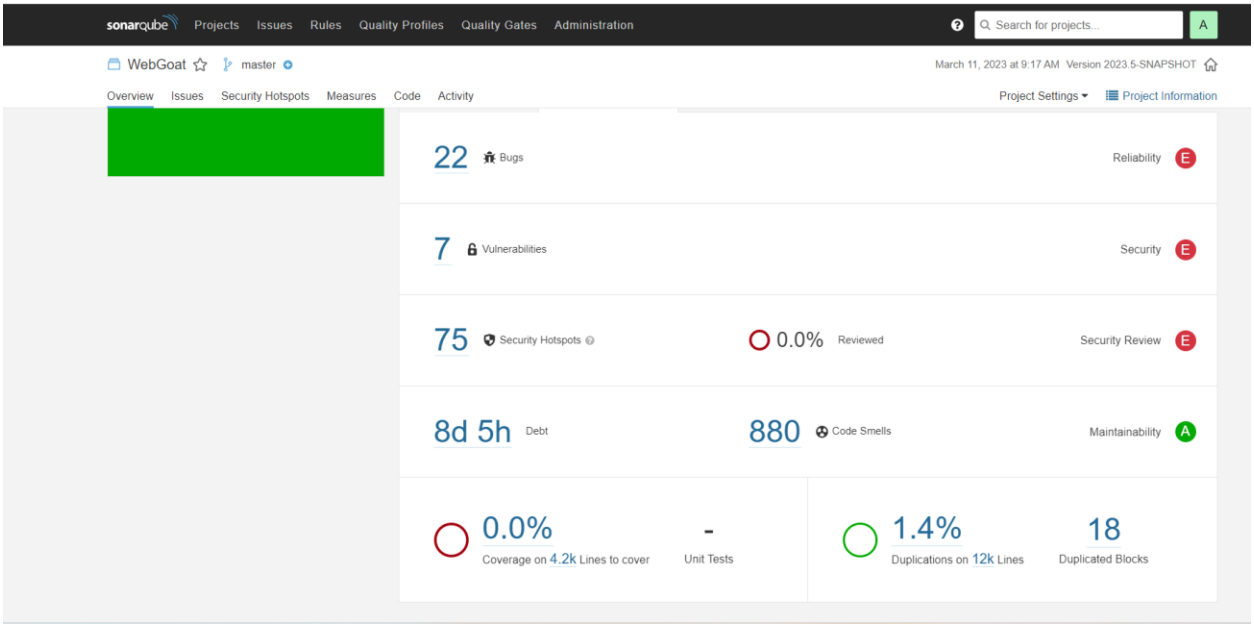


8.2 Dependency-Check Trend Status



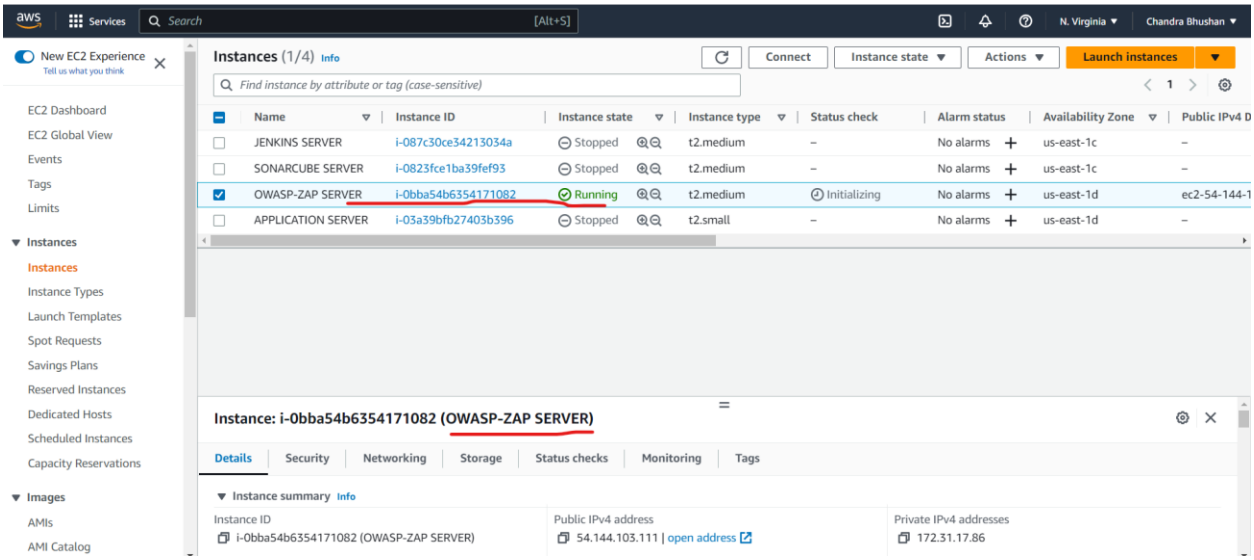
8.3 Dashboard of SonarQube after checked Bugs and Vulnerability





OWASP-ZAP Server check Scanned Report

9.1 Dashbord of OWASP-ZAP Server



9.2 Check Application Scanned Report

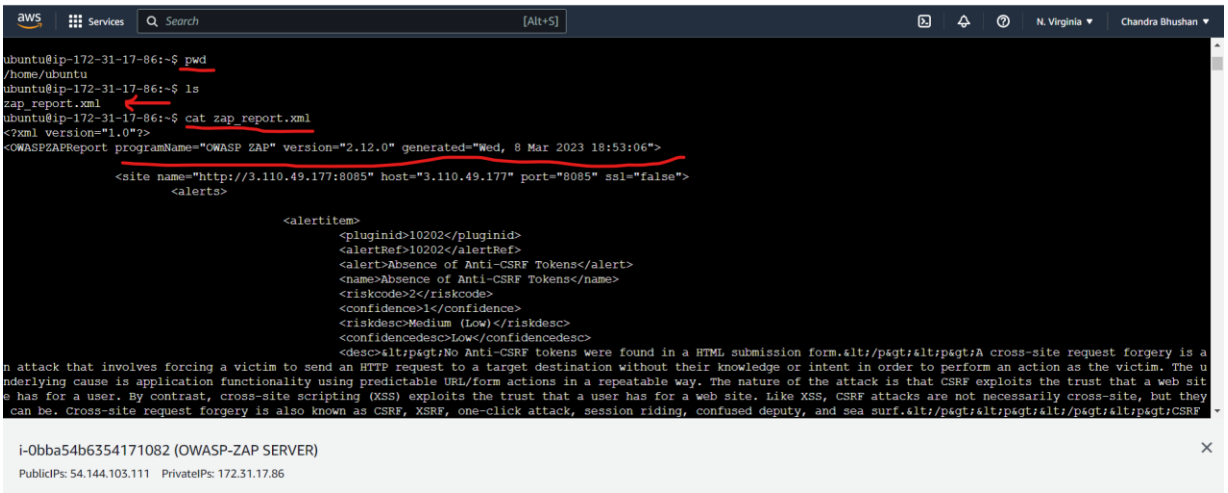
ubuntu@ip-172-31-17-86:~$ pwd

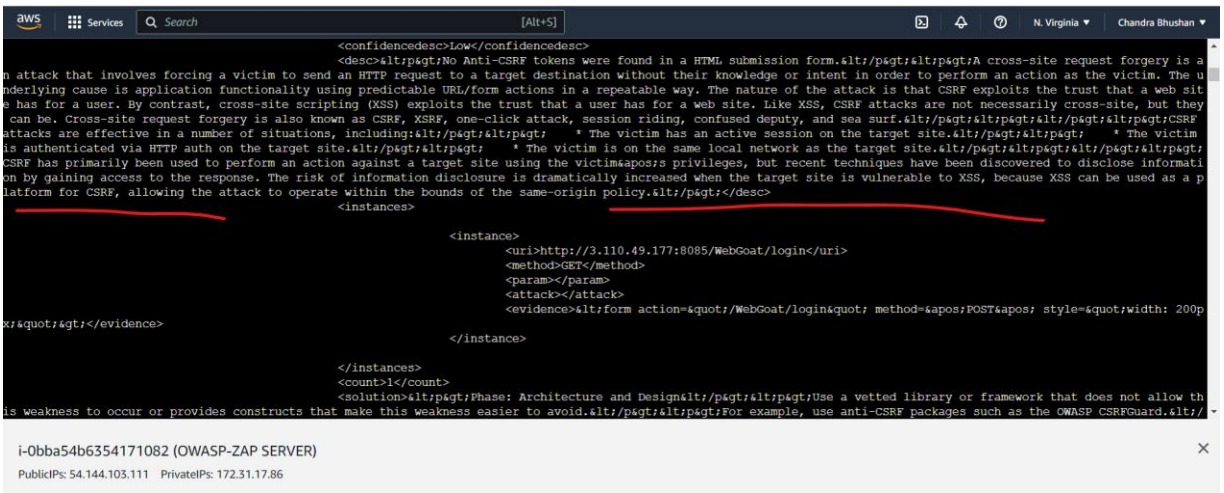
ubuntu@ip-172-31-17-86:~$ ls

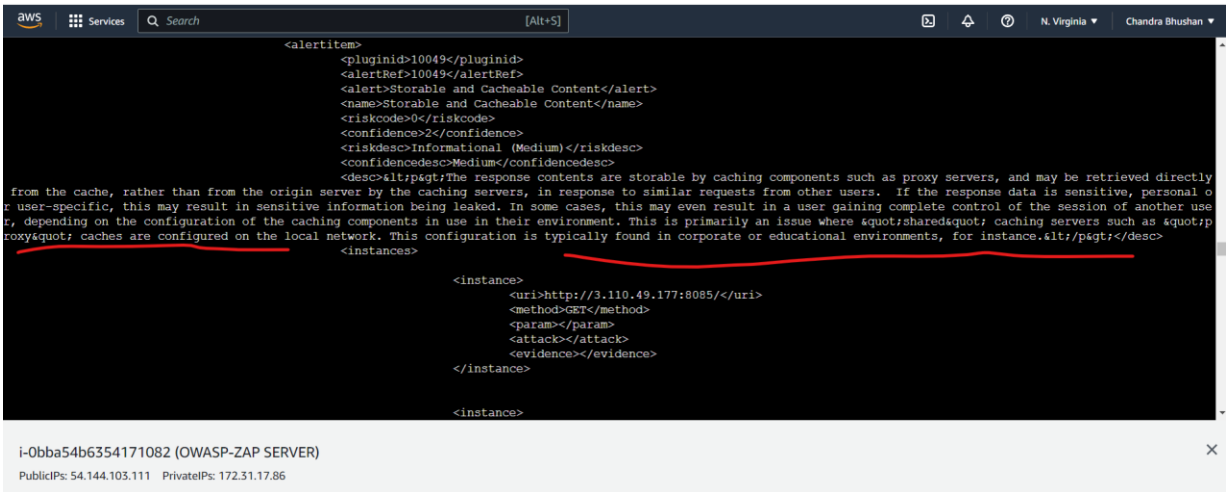
zap_report.xml

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

9.3 Screen shoot of zap_report.xml







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