Semester: V
Academic Year: 2025-26
Class / Branch: TE IT-C3
Subject: ADL Lab
Name of Instructor: Prof.Vishal Badgujar

Name of Student:Shreya Sonawale
Student ID:23104134

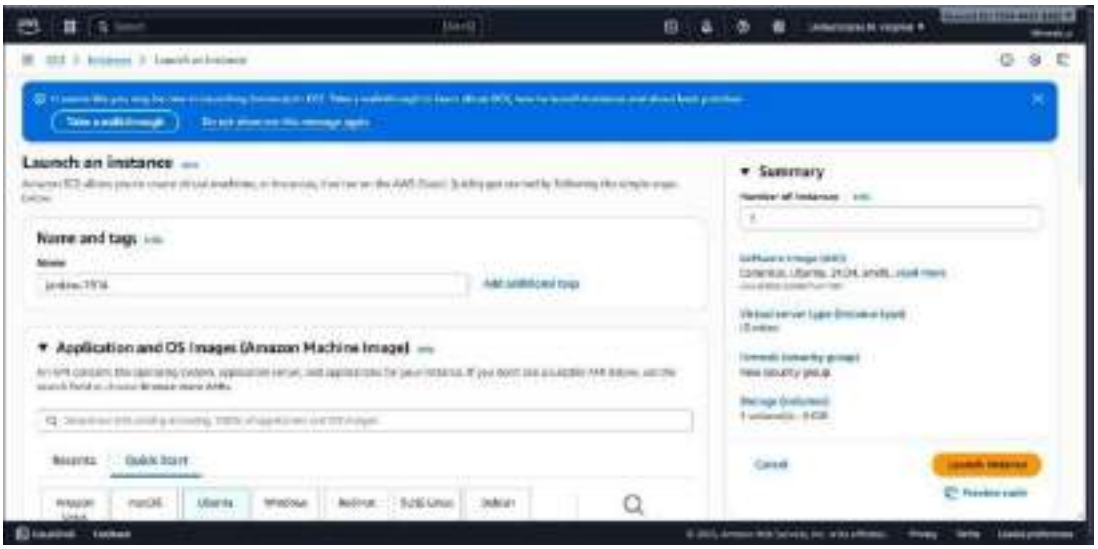
EXPERIMENT NO. 08

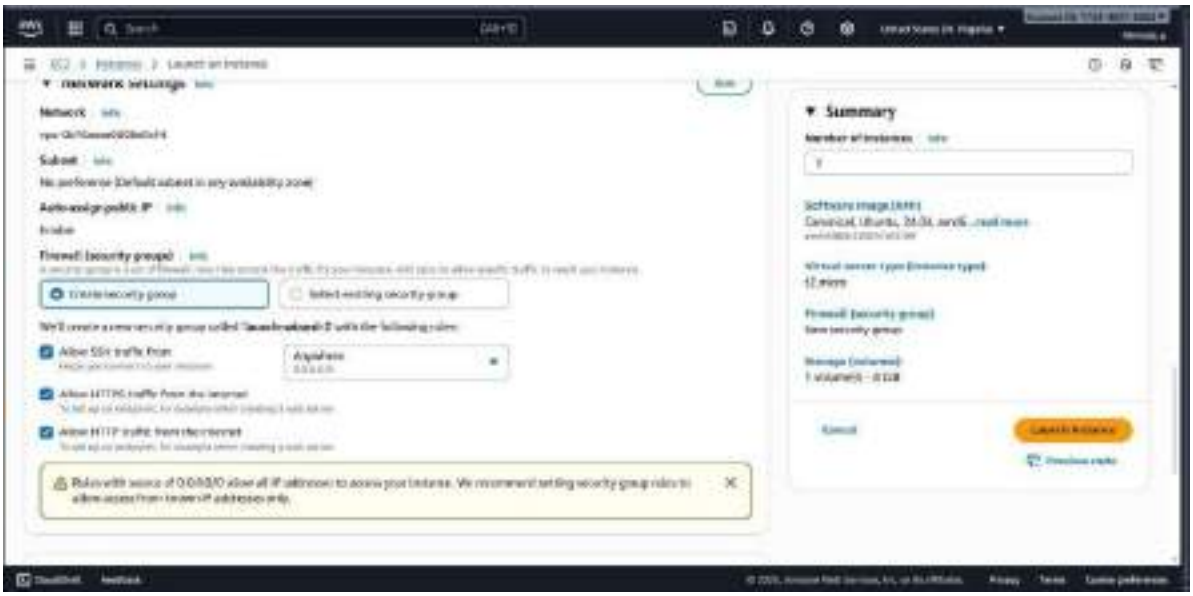
Aim: Create a Jenkins CICD Pipeline with SonarQube / GitLab Integration to perform a static analysis of the code to detect bugs, code smells, and security vulnerabilities on a sample Java application.

Steps:

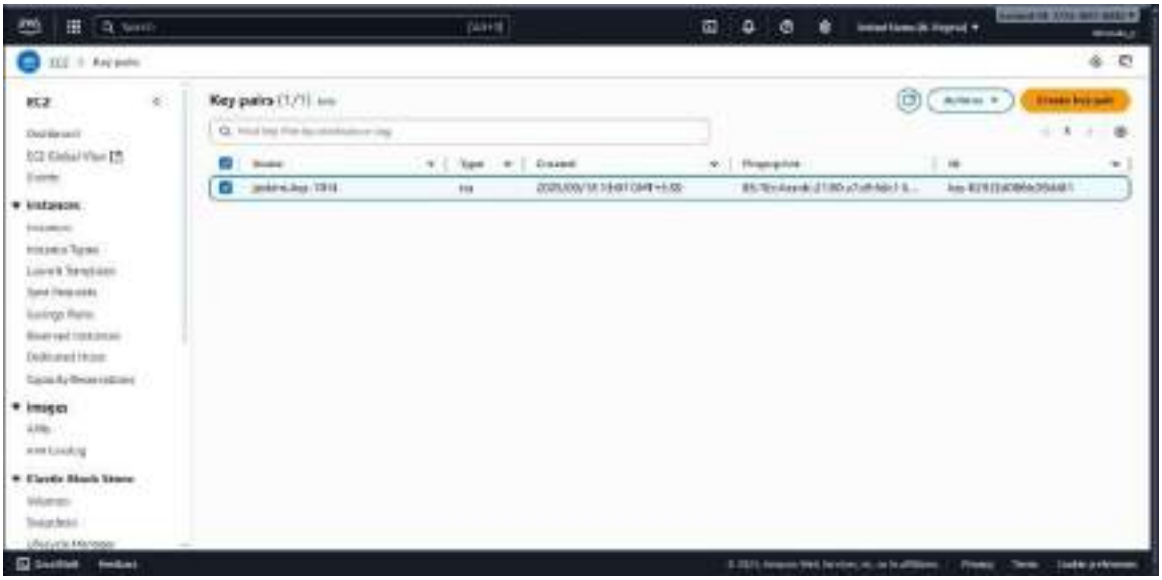
JENKINS Install on AWS instance:

1. Create EC2 Instance -> Ubuntu/t2.micro/enable all network settings

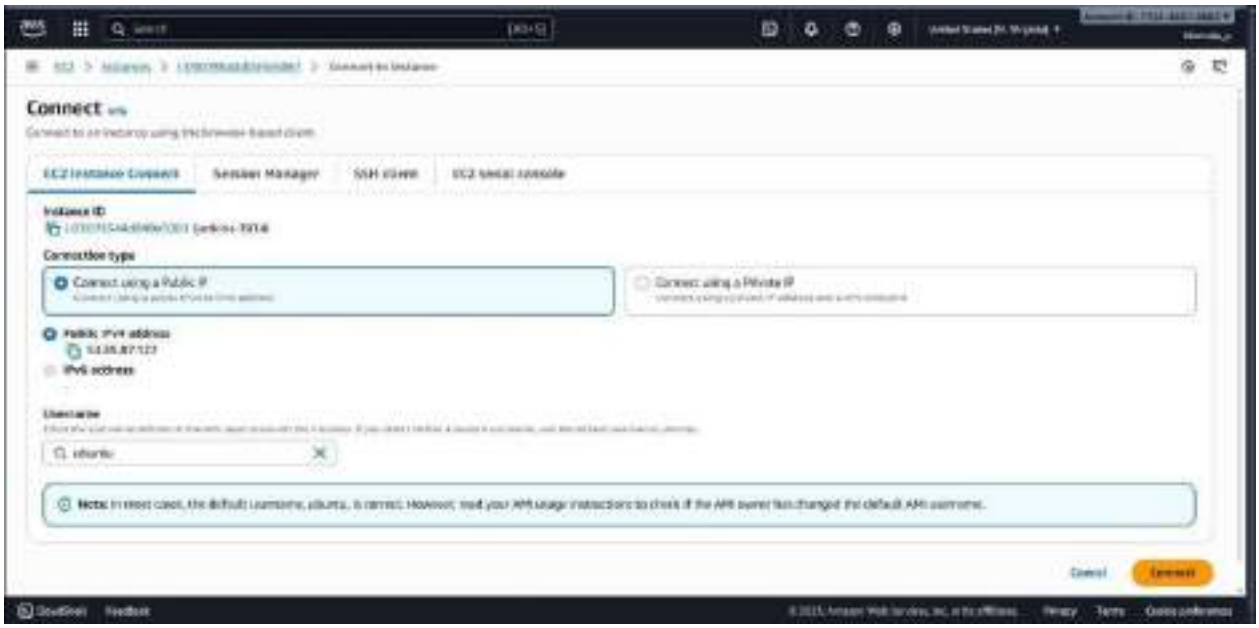




2. Create Key-Value pair

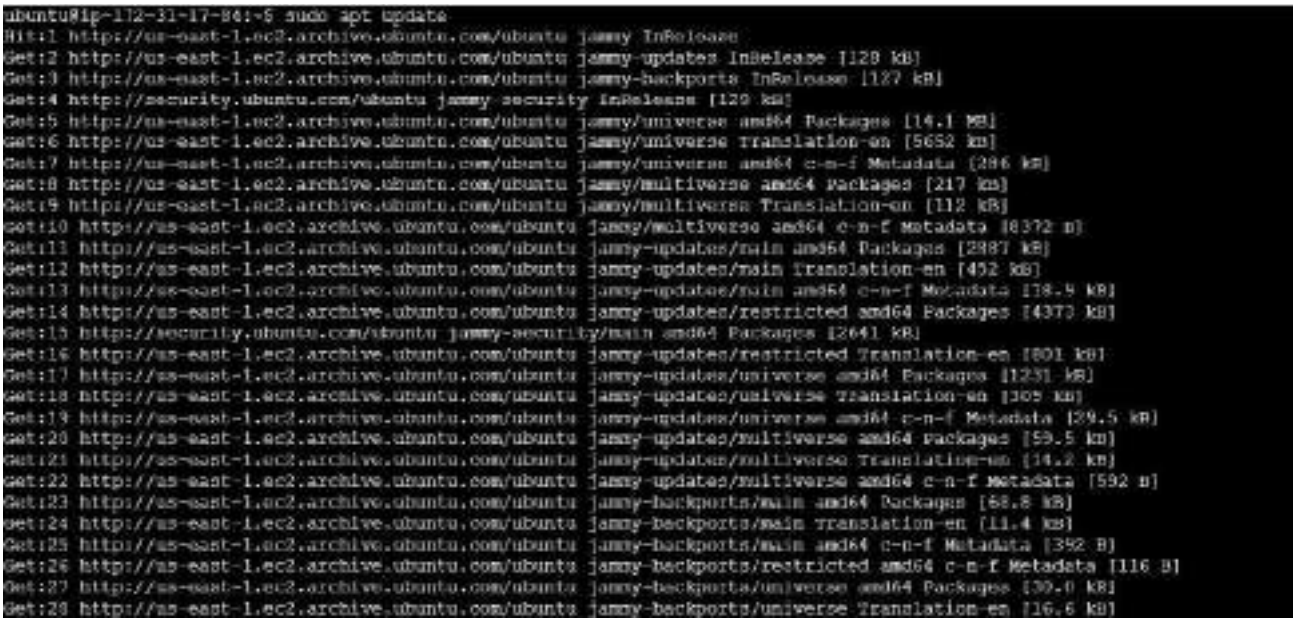


3. Connect to EC2 instance



4. Run Commands on Jenkins terminal:

$ sudo apt update



$ sudo apt install openjdk-11-jdk –y

$ java --version

5. Run commands:

- Use debian-stable

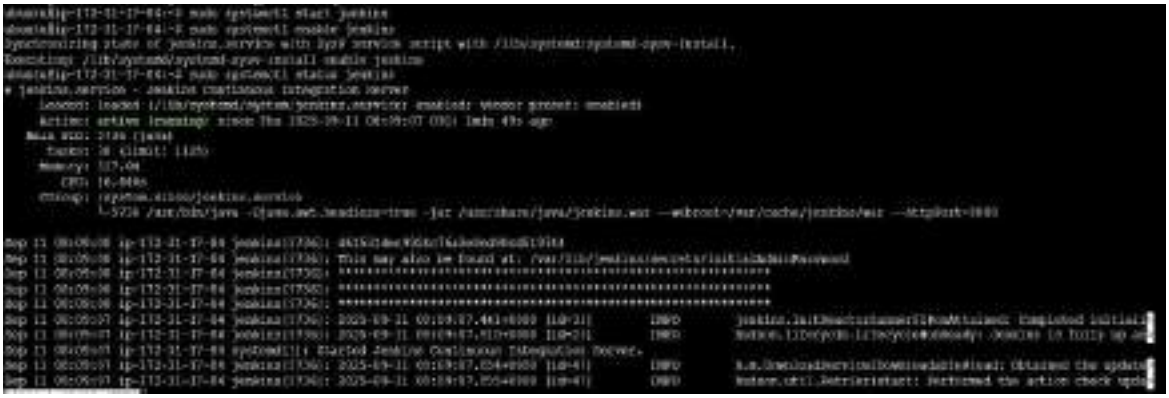
  Run the following commands: -

```
sudo wget -O /etc/apt/keyrings/jenkins-keyring.asc \
  https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
```

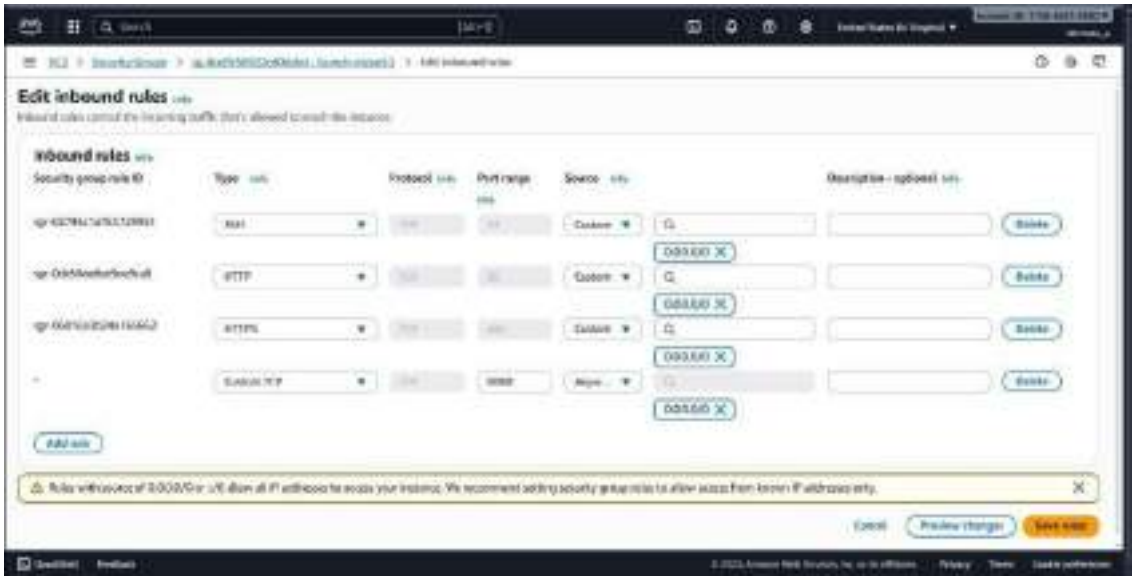
```
echo "deb [signed-by=/etc/apt/keyrings/jenkins-keyring.asc]" \
  https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
  /etc/apt/sources.list.d/jenkins.list > /dev/null
```

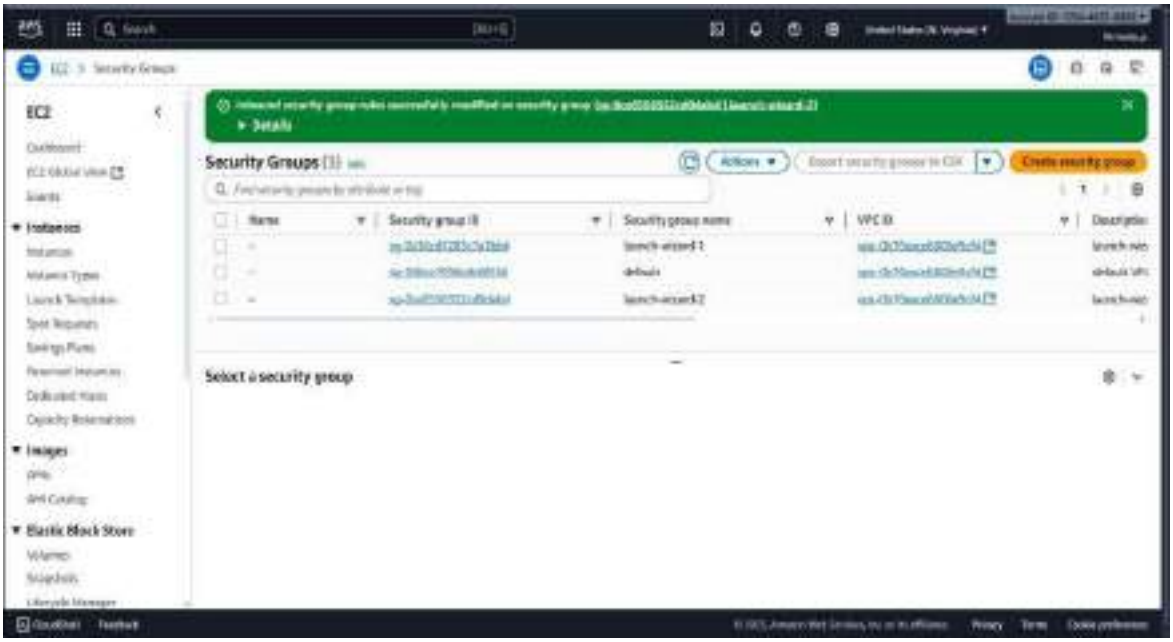
```
sudo apt-get update
sudo apt-get install fontconfig openjdk-17-jre
sudo apt-get install jenkins
```

6. Then run following commands: -
   sudo systemctl start jenkins
   sudo systemctl enable jenkins
   sudo systemctl status jenkins



7. Edit inbound rules -> add rule -> port 8080





8. Go to instance and browse using IPV4 address

9. For password -> Copy path on webpage you browsed and run command **sudo cat <path>**

```
ubuntu@ip-172-31-17-84:~$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword
d61531dec9924c76a3e8ed90cd519744
ubuntu@ip-172-31-17-84:~$
```

- Copy the password and paste
- Choose Install suggested plugins

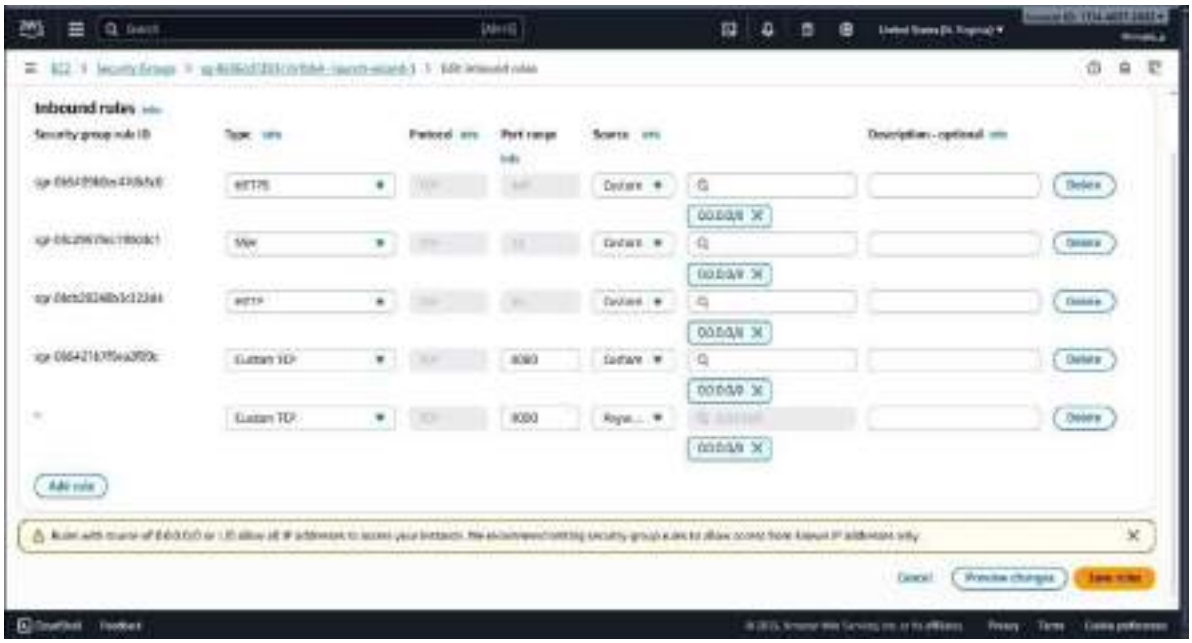
- Fill the details

- Click 'Save and Finish'
- Start using jenkins

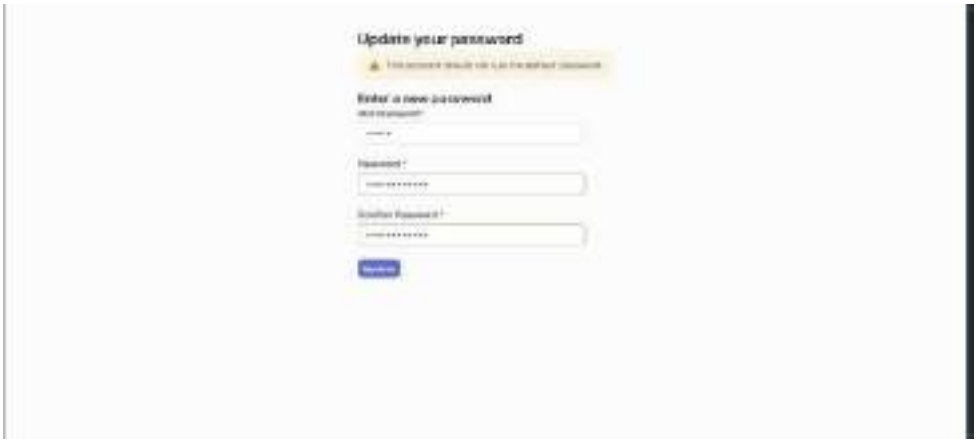
10. Go to Jenkins Dashboard -> manage Jenkins -> Available Plugins -> Install Plugin

11. Go to jenkins dashboard -> manage jenkins -> system configuration -> tools -> Sonarqube scanner -> save

12. Edit inbound rules -> add port 9000



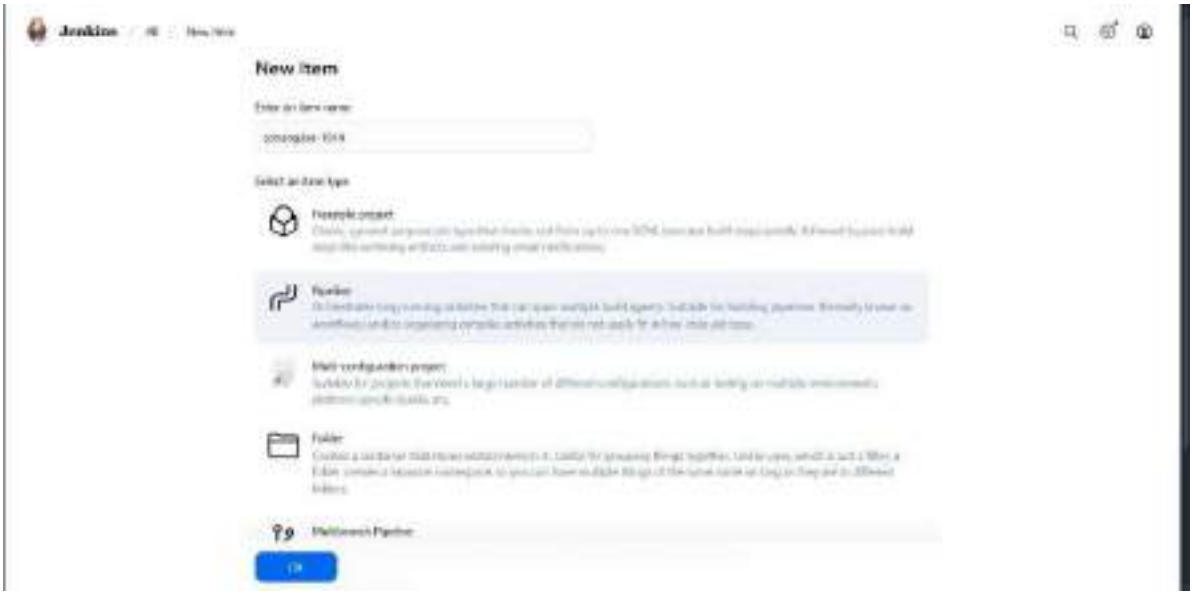
13. Go to browser -> http://localhost:9000
    - type username and pasword as 'admin'
    - Set new password



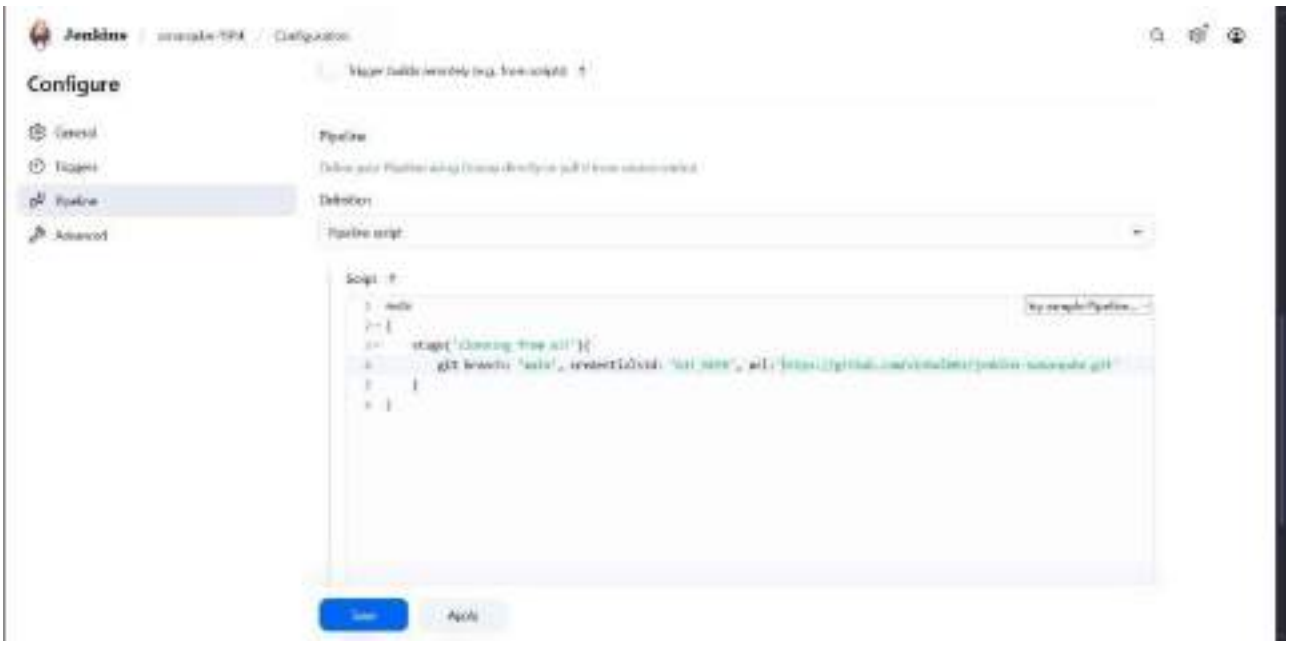
14. Change system configuration in jenkins Add server url -> 'http://localhost://9000'

15. To create pipeline

- Go to jenkins Dashboard -> new item -> select pipeline

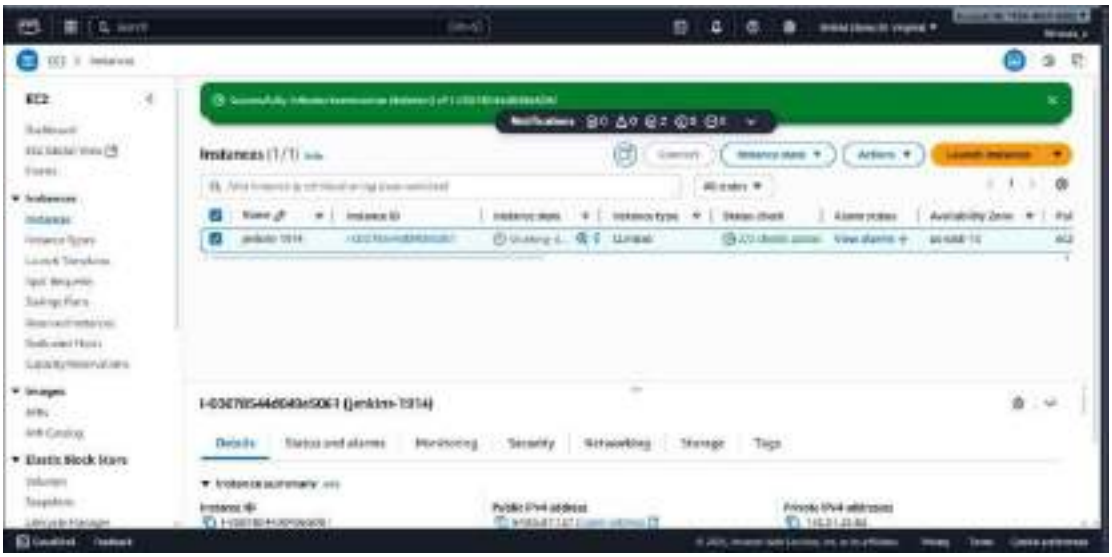


- Add script



Go to Console Output:

16. Delete everything in end
    - Key-value pair
    - Instance
    - Security groups



CONCLUSION:

In this experiment, I understood how SAST helps to find security issues in the source code before running the program. By connecting Jenkins with SonarQube, we can

automatically check the code in the pipeline and fix problems early. This makes the code safer, better in quality, and reduces risks during development.