NAME:ARYAN PATANKAR **ROLL.NO:34** **CLASS:D15A**

Case Study

14. Continuous Integration with Simple Code Analysis

- **Concepts Used**: Jenkins, AWS Cloud9, and SonarQube.
- **Problem Statement**: "Set up a Jenkins pipeline using AWS Cloud9 to perform a simple code analysis on a JavaScript file using SonarQube."
- **Tasks**:
  - Create a Jenkins job using AWS Cloud9.
  - Configure the job to integrate with SonarQube for basic code analysis.
  - Run the Jenkins job with a JavaScript file and review the analysis report.

---

1. Introduction

Case Study Overview: This case study focuses on setting up a **Continuous Integration (CI) pipeline** using Jenkins and SonarQube on an **AWS EC2 instance**. This setup ensures automated testing and code quality analysis during software development. Due to limitations in **Cloud9** availability, we used an **EC2 instance** to host Jenkins and SonarQube.

Key Feature and Application: The main feature of this case study is **automating the build process** with Jenkins, combined with SonarQube for **code quality analysis**. This pipeline helps detect errors early and ensures that the code meets high standards before deployment.

Third-Year Project Integration (Optional): If applicable, explain how your third-year project relates to the case study.

The **E-Mart project** directly relates to the case study on **Continuous Integration with Simple Code Analysis** by demonstrating the practical application of **Jenkins**, **AWS Cloud9**, and **SonarQube** in a real-world e-commerce platform. In the case study, Jenkins is set up to automate code analysis using SonarQube to ensure high code quality, which mirrors the workflow in the E-Mart project for maintaining the integrity of a scalable e-commerce solution.

While the E-Mart project is a full-fledged e-commerce platform with features like **collaborative shopping** and **social impact initiatives**, integrating a **CI/CD pipeline** as outlined in the case study enhances the development process by automatically detecting code quality issues, improving reliability, and ensuring efficient deployments. This integration of automated code analysis supports the E-Mart project in ensuring that the code remains robust, secure, and well-optimized as the platform scales and incorporates new features.

NAME:ARYAN PATANKAR **ROLL.NO:34** **CLASS:D15A**

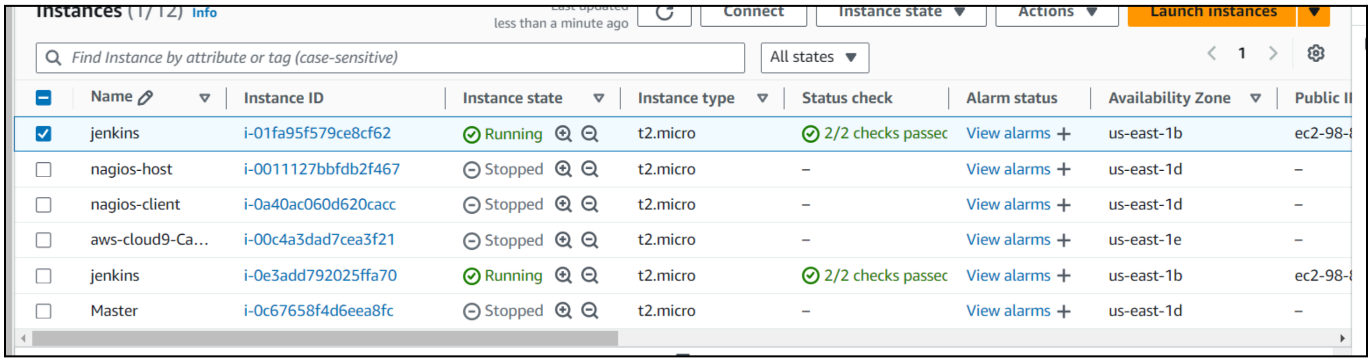
2. Step-by-Step Explanation

Step 1: Initial Setup and Configuration

- **Launch AWS EC2 Instance** for both **jenkins** and **sonarqube** :
    1. Create an AWS account if you haven't.
    2. Launch a **t2.medium** EC2 instance with **Ubuntu 20.04**.
    3. SSH into the instance using a terminal with the command

Allow the following inbound rules:

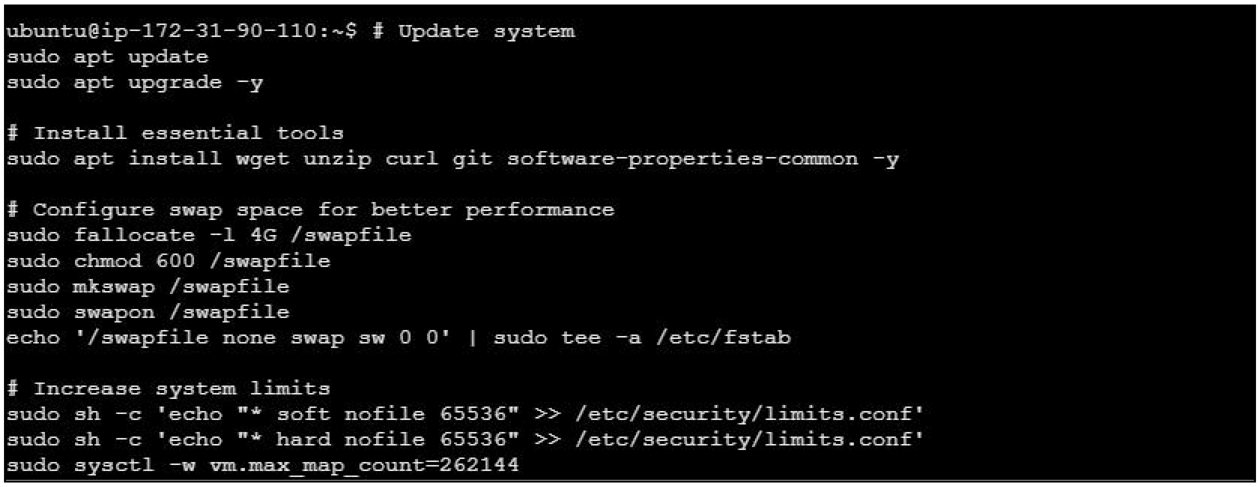
- **HTTP (port 80)**: For accessing Jenkins.
- **SSH (port 22)**: For secure shell access.
- **Custom TCP (port 8080)**: For accessing Jenkins.



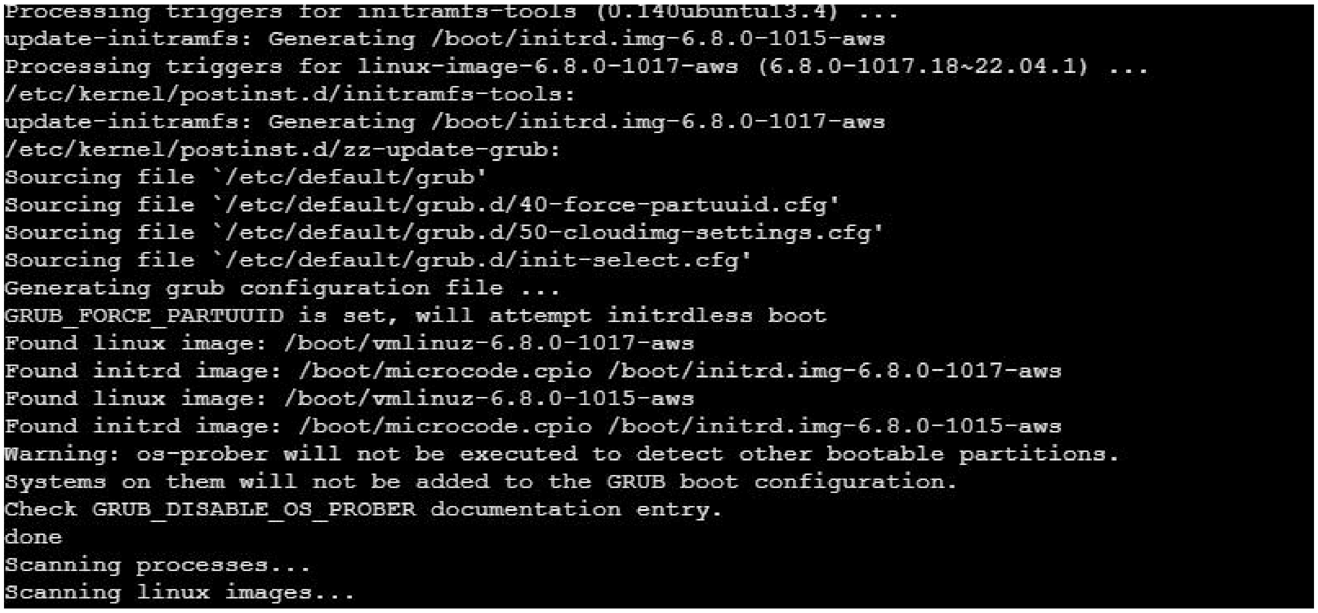
Instances (1/12) Info
Last updated less than a minute ago
Connect
Instance state
Actions
Launch instances
Find Instance by attribute or tag (case-sensitive)
All states
1

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public II
✓ jenkins	i-01fa95f579ce8cf62	Running	t2.micro	2/2 checks passec	View alarms +	us-east-1b	ec2-98-
nagios-host	i-0011127bbfdb2f467	Stopped	t2.micro	–	View alarms +	us-east-1d	–
nagios-client	i-0a40ac060d620cacc	Stopped	t2.micro	–	View alarms +	us-east-1d	–
aws-cloud9-Ca…	i-00c4a3dad7cea3f21	Stopped	t2.micro	–	View alarms +	us-east-1e	–
jenkins	i-0e3add792025ffa70	Running	t2.micro	2/2 checks passec	View alarms +	us-east-1b	ec2-98-
Master	i-0c67658f4d6eea8fc	Stopped	t2.micro	–	View alarms +	us-east-1d	–

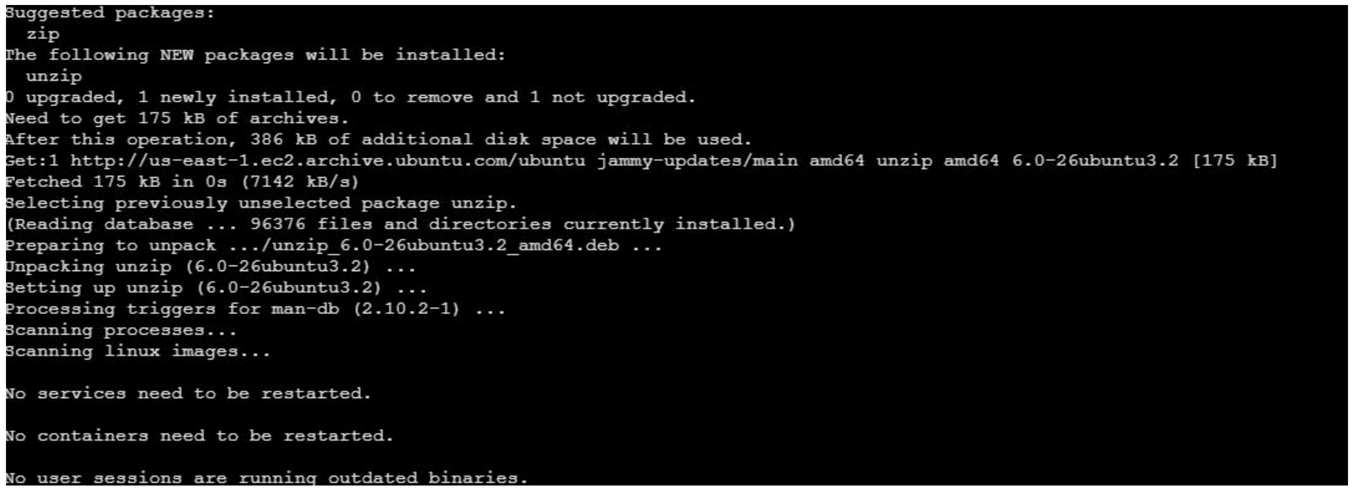
Step 2:Updating the system and installing essential tools



```
ubuntu@ip-172-31-90-110:~$ # Update system
sudo apt update
sudo apt upgrade -y

# Install essential tools
sudo apt install wget unzip curl git software-properties-common -y

# Configure swap space for better performance
sudo fallocate -l 4G /swapfile
sudo chmod 600 /swapfile
sudo mkswap /swapfile
sudo swapon /swapfile
echo '/swapfile none swap sw 0 0' | sudo tee -a /etc/fstab

# Increase system limits
sudo sh -c 'echo "* soft nofile 65536" >> /etc/security/limits.conf'
sudo sh -c 'echo "* hard nofile 65536" >> /etc/security/limits.conf'
sudo sysctl -w vm.max_map_count=262144
```



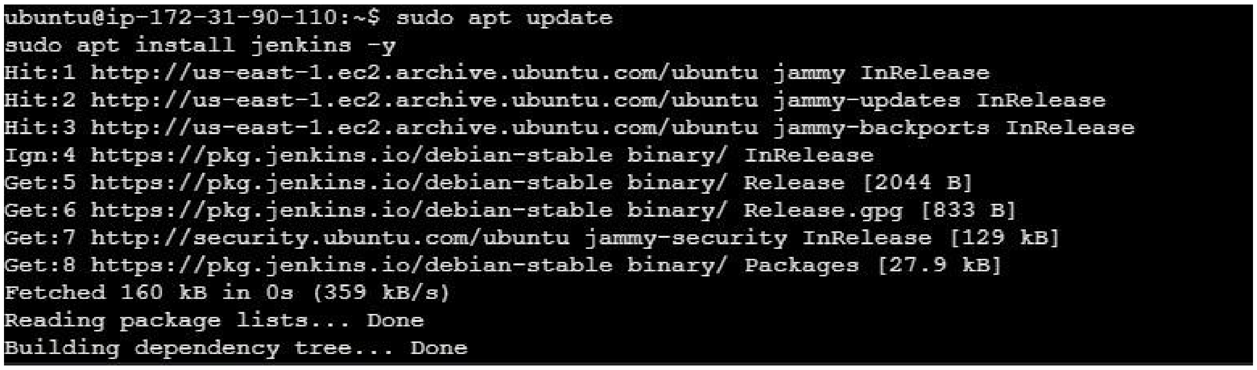
```
Processing triggers for initramfs-tools (0.140ubuntu13.4) ...
update-initramfs: Generating /boot/initrd.img-6.8.0-1015-aws
Processing triggers for linux-image-6.8.0-1017-aws (6.8.0-1017.18~22.04.1) ...
/etc/kernel/postinst.d/initramfs-tools:
update-initramfs: Generating /boot/initrd.img-6.8.0-1017-aws
/etc/kernel/postinst.d/zz-update-grub:
Sourcing file `/etc/default/grub'
Sourcing file `/etc/default/grub.d/40-force-partuuid.cfg'
Sourcing file `/etc/default/grub.d/50-cloudimg-settings.cfg'
Sourcing file `/etc/default/grub.d/init-select.cfg'
Generating grub configuration file ...
GRUB_FORCE_PARTUUID is set, will attempt initrdless boot
Found linux image: /boot/vmlinuz-6.8.0-1017-aws
Found initrd image: /boot/microcode.cpio /boot/initrd.img-6.8.0-1017-aws
Found linux image: /boot/vmlinuz-6.8.0-1015-aws
Found initrd image: /boot/microcode.cpio /boot/initrd.img-6.8.0-1015-aws
Warning: os-prober will not be executed to detect other bootable partitions.
Systems on them will not be added to the GRUB boot configuration.
Check GRUB_DISABLE_OS_PROBER documentation entry.
done
Scanning processes...
Scanning linux images...
```



```
Suggested packages:
  zip
The following NEW packages will be installed:
  unzip
0 upgraded, 1 newly installed, 0 to remove and 1 not upgraded.
Need to get 175 kB of archives.
After this operation, 386 kB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 unzip amd64 6.0-26ubuntu3.2 [175 kB]
Fetched 175 kB in 0s (7142 kB/s)
Selecting previously unselected package unzip.
(Reading database ... 96376 files and directories currently installed.)
Preparing to unpack .../unzip_6.0-26ubuntu3.2_amd64.deb ...
Unpacking unzip (6.0-26ubuntu3.2) ...
Setting up unzip (6.0-26ubuntu3.2) ...
Processing triggers for man-db (2.10.2-1) ...
Scanning processes...
Scanning linux images...

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.
```

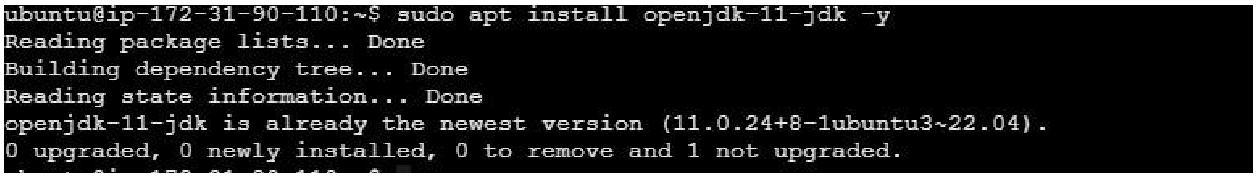
Step 3: Install Jenkins on EC2 (Ubuntu)

- ssh -i path/to/your-key.pem ubuntu@<your-EC2-IP>
- sudo apt update



```
ubuntu@ip-172-31-90-110:~$ sudo apt update
sudo apt install jenkins -y
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Ign:4 https://pkg.jenkins.io/debian-stable binary/ InRelease
Get:5 https://pkg.jenkins.io/debian-stable binary/ Release [2044 B]
Get:6 https://pkg.jenkins.io/debian-stable binary/ Release.gpg [833 B]
Get:7 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Get:8 https://pkg.jenkins.io/debian-stable binary/ Packages [27.9 kB]
Fetched 160 kB in 0s (359 kB/s)
Reading package lists... Done
Building dependency tree... Done
```

- sudo apt install fontconfig openjdk-17-jre

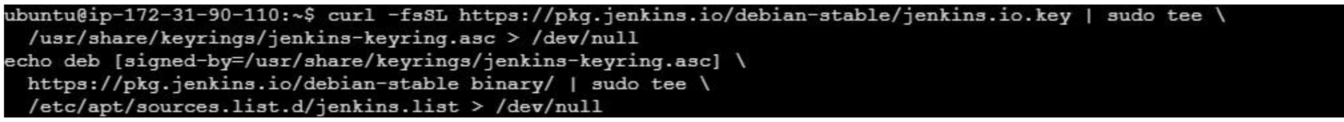


```
ubuntu@ip-172-31-90-110:~$ sudo apt install openjdk-11-jdk -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
openjdk-11-jdk is already the newest version (11.0.24+8-1ubuntu3~22.04).
0 upgraded, 0 newly installed, 0 to remove and 1 not upgraded.
```

- java -version

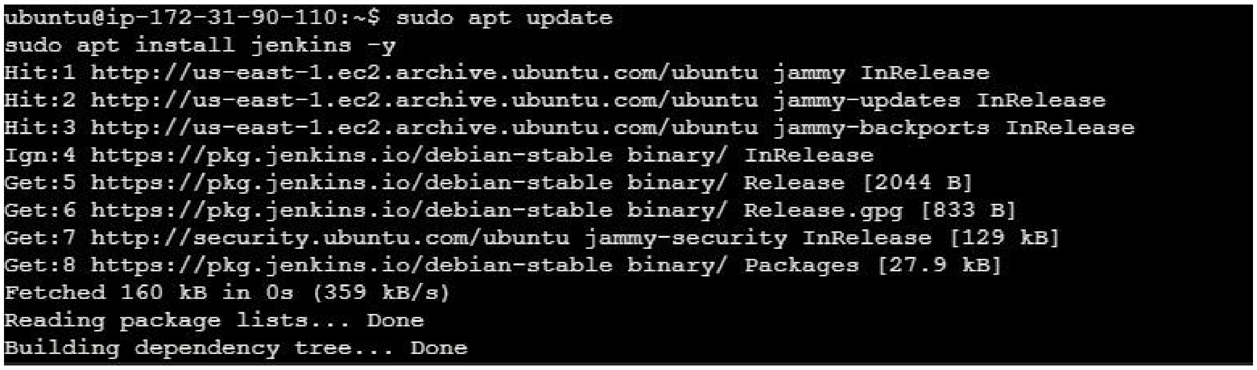
<u># Add the Jenkins repository</u>

- sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \
  https://pkg.jenkins.io/debian/jenkins.io-2023.key



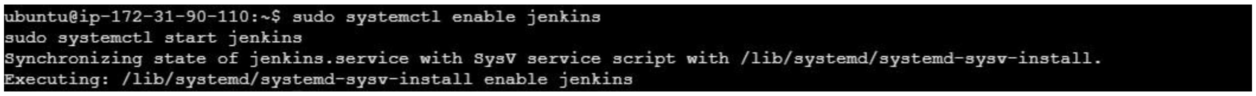
```
ubuntu@ip-172-31-90-110:~$ curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo tee \
  /usr/share/keyrings/jenkins-keyring.asc > /dev/null
echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
  https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
  /etc/apt/sources.list.d/jenkins.list > /dev/null
```

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



```
ubuntu@ip-172-31-90-110:~$ sudo apt update
sudo apt install jenkins -y
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Ign:4 https://pkg.jenkins.io/debian-stable binary/ InRelease
Get:5 https://pkg.jenkins.io/debian-stable binary/ Release [2044 B]
Get:6 https://pkg.jenkins.io/debian-stable binary/ Release.gpg [833 B]
Get:7 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Get:8 https://pkg.jenkins.io/debian-stable binary/ Packages [27.9 kB]
Fetched 160 kB in 0s (359 kB/s)
Reading package lists... Done
Building dependency tree... Done
```

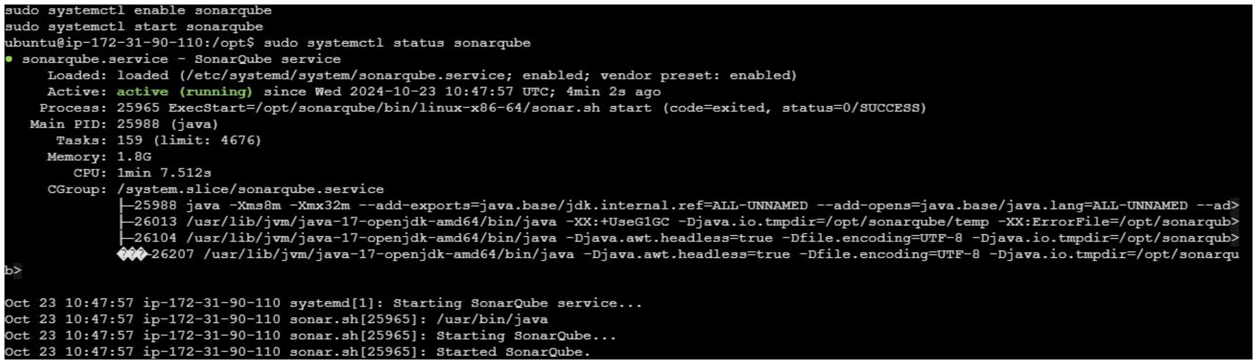
- sudo systemctl start jenkins
- sudo systemctl enable jenkins



```
ubuntu@ip-172-31-90-110:~$ sudo systemctl enable jenkins
sudo systemctl start jenkins
Synchronizing state of jenkins.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable jenkins
```

NAME:ARYAN PATANKAR ROLL.NO:34 CLASS:D15A

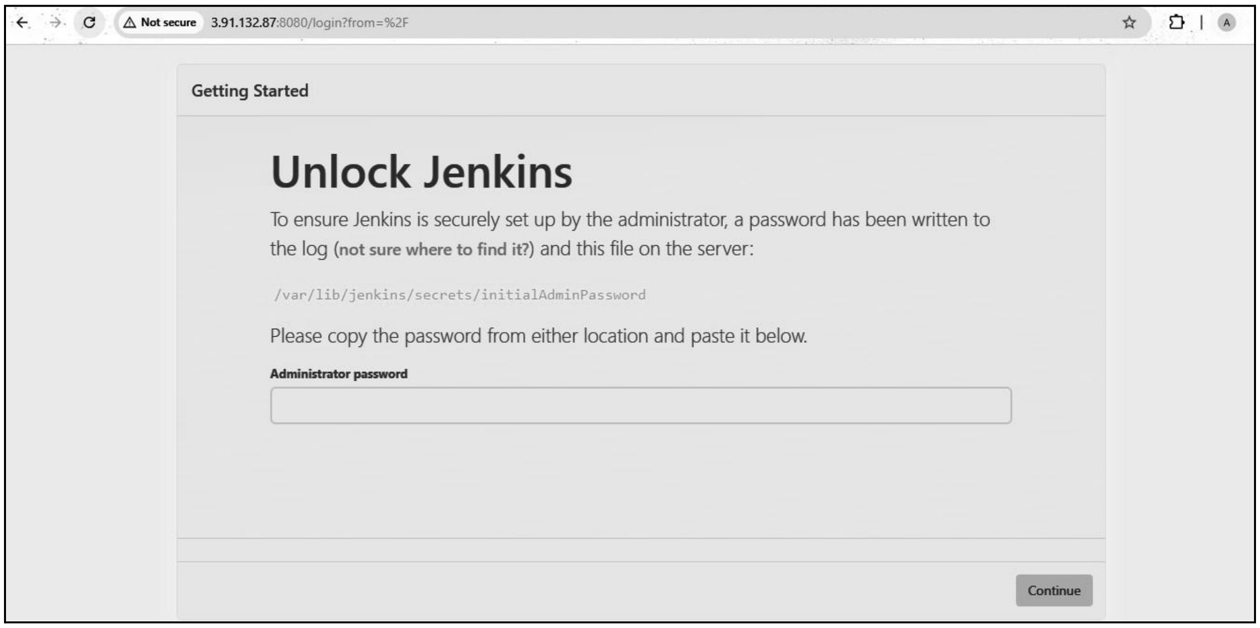
- sudo systemctl status jenkins



```
sudo systemctl enable sonarqube
sudo systemctl start sonarqube
ubuntu@ip-172-31-90-110:/opt$ sudo systemctl status sonarqube
● sonarqube.service - SonarQube service
     Loaded: loaded (/etc/systemd/system/sonarqube.service; enabled; vendor preset: enabled)
     Active: active (running) since Wed 2024-10-23 10:47:57 UTC; 4min 2s ago
    Process: 25965 ExecStart=/opt/sonarqube/bin/linux-x86-64/sonar.sh start (code=exited, status=0/SUCCESS)
   Main PID: 25988 (java)
      Tasks: 159 (limit: 4676)
     Memory: 1.8G
        CPU: 1min 7.512s
     CGroup: /system.slice/sonarqube.service
             ├─25988 java -Xms8m -Xmx32m --add-exports=java.base/jdk.internal.ref=ALL-UNNAMED --add-opens=java.base/java.lang=ALL-UNNAMED --ad>
             ├─26013 /usr/lib/jvm/java-17-openjdk-amd64/bin/java -XX:+UseG1GC -Djava.io.tmpdir=/opt/sonarqube/temp -XX:ErrorFile=/opt/sonarqub>
             ├─26104 /usr/lib/jvm/java-17-openjdk-amd64/bin/java -Djava.awt.headless=true -Dfile.encoding=UTF-8 -Djava.io.tmpdir=/opt/sonarqub>
             ���26207 /usr/lib/jvm/java-17-openjdk-amd64/bin/java -Djava.awt.headless=true -Dfile.encoding=UTF-8 -Djava.io.tmpdir=/opt/sonarqu
b>

Oct 23 10:47:57 ip-172-31-90-110 systemd[1]: Starting SonarQube service...
Oct 23 10:47:57 ip-172-31-90-110 sonar.sh[25965]: /usr/bin/java
Oct 23 10:47:57 ip-172-31-90-110 sonar.sh[25965]: Starting SonarQube...
Oct 23 10:47:57 ip-172-31-90-110 sonar.sh[25965]: Started SonarQube.
```

Open a browser and navigate to [`http://<your-EC2-IP>:8080`](http://<your-EC2-IP>:8080).

Jenkins status Active



Getting Started

Unlock Jenkins

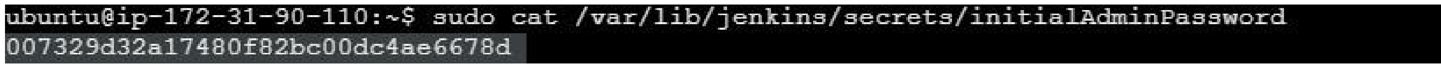
To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:

/var/lib/jenkins/secrets/initialAdminPassword

Please copy the password from either location and paste it below.

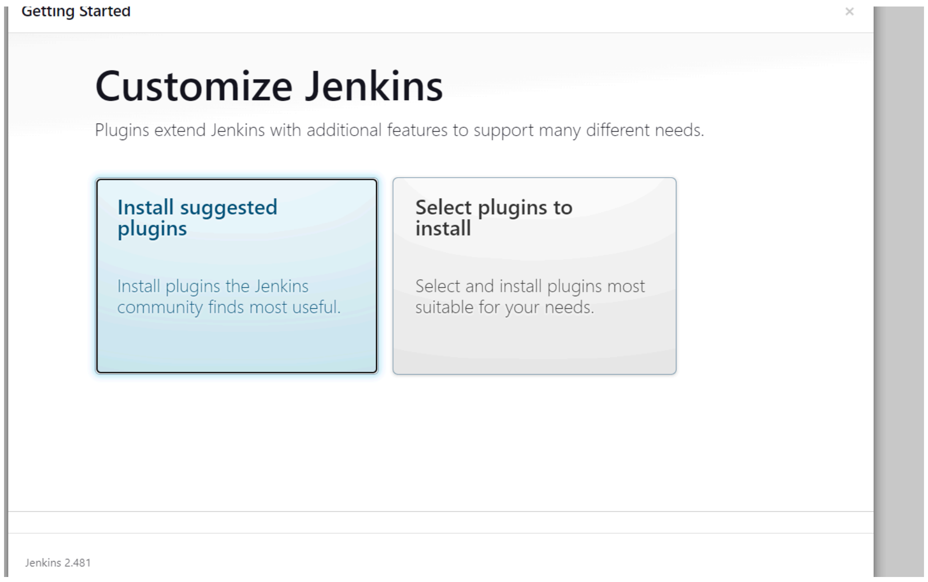
Administrator password

`sudo cat /var/lib/jenkins/secrets/initialAdminPassword` to get Administrator Password



```
ubuntu@ip-172-31-90-110:~$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword
007329d32a17480f82bc00dc4ae6678d
```

NAME:ARYAN PATANKAR **ROLL.NO:34** **CLASS:D15A**


Getting Started

Customize Jenkins

Plugins extend Jenkins with additional features to support many different needs.

Install suggested plugins

Install plugins the Jenkins community finds most useful.

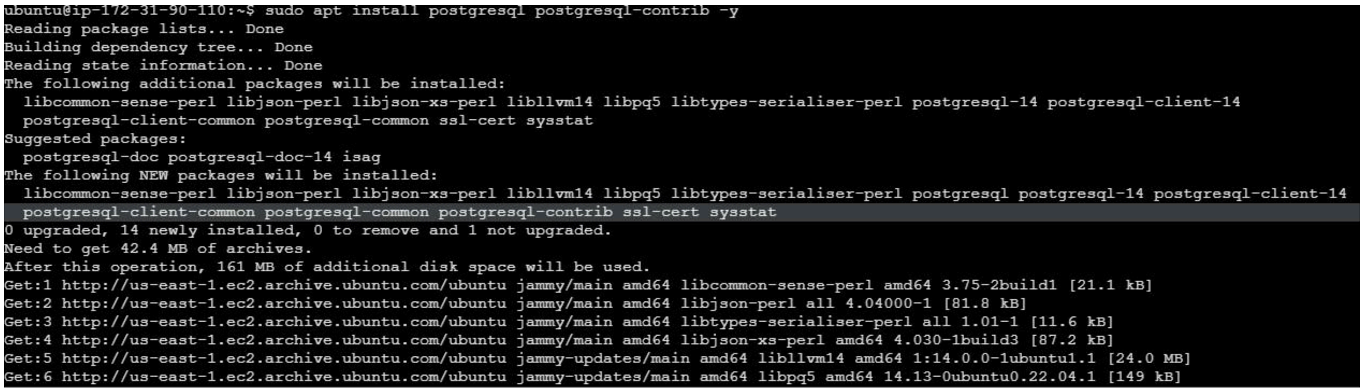
Select plugins to install

Select and install plugins most suitable for your needs.

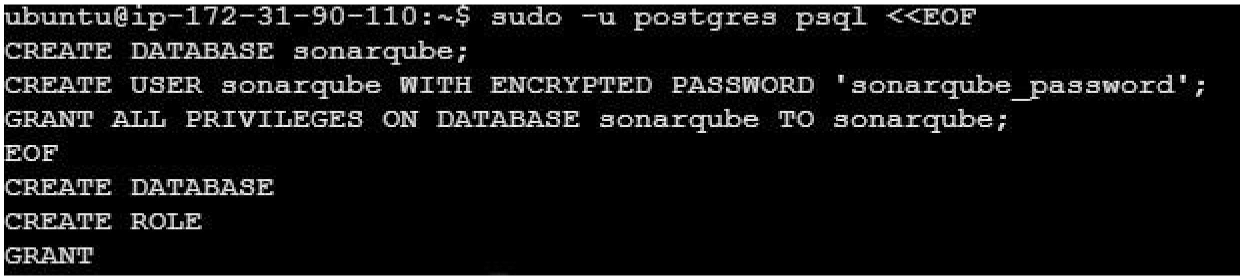
Jenkins 2.481

Step 3: To setup sonarqube in ec2

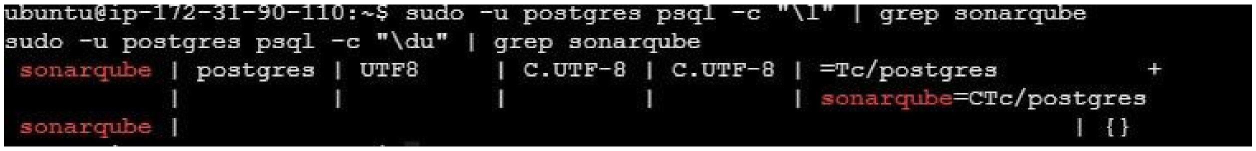
- **Install PostgreSQL**


```
ubuntu@ip-172-31-90-110:~$ sudo apt install postgresql postgresql-contrib -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libcommon-sense-perl libjson-perl libjson-xs-perl libllvm14 libpq5 libtypes-serialiser-perl postgresql-14 postgresql-client-14
  postgresql-client-common postgresql-common ssl-cert sysstat
Suggested packages:
  postgresql-doc postgresql-doc-14 isag
The following NEW packages will be installed:
  libcommon-sense-perl libjson-perl libjson-xs-perl libllvm14 libpq5 libtypes-serialiser-perl postgresql postgresql-14 postgresql-client-14
  postgresql-client-common postgresql-common postgresql-contrib ssl-cert sysstat
0 upgraded, 14 newly installed, 0 to remove and 1 not upgraded.
Need to get 42.4 MB of archives.
After this operation, 161 MB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libcommon-sense-perl amd64 3.75-2build1 [21.1 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libjson-perl all 4.04000-1 [81.8 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libtypes-serialiser-perl all 1.01-1 [11.6 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libjson-xs-perl amd64 4.030-1build3 [87.2 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libllvm14 amd64 1:14.0.0-1ubuntu1.1 [24.0 MB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libpq5 amd64 14.13-0ubuntu0.22.04.1 [149 kB]
```

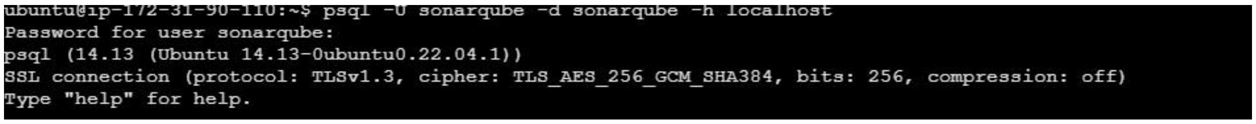
- **Create SonarQube database and user**


```
ubuntu@ip-172-31-90-110:~$ sudo -u postgres psql <<EOF
CREATE DATABASE sonarqube;
CREATE USER sonarqube WITH ENCRYPTED PASSWORD 'sonarqube_password';
GRANT ALL PRIVILEGES ON DATABASE sonarqube TO sonarqube;
EOF
CREATE DATABASE
CREATE ROLE
GRANT
```

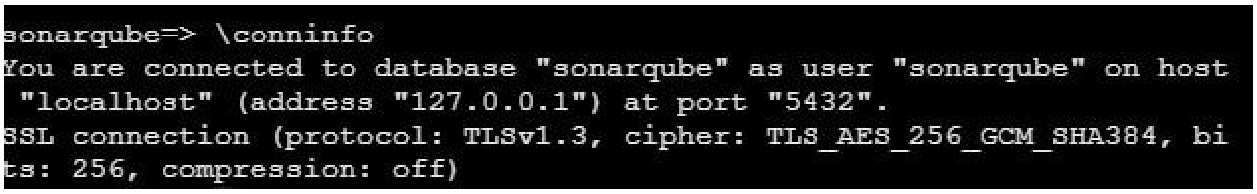
NAME:ARYAN PATANKAR **ROLL.NO:34** **CLASS:D15A**



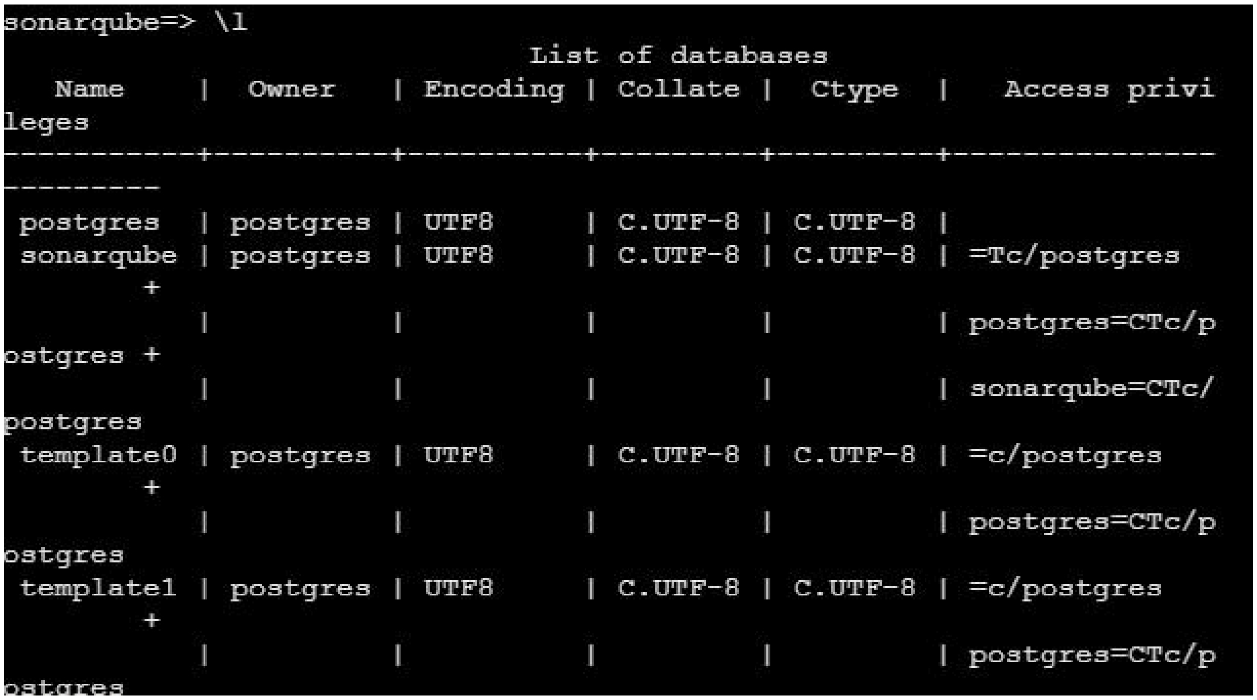
```
ubuntu@ip-172-31-90-110:~$ sudo -u postgres psql -c "\l" | grep sonarqube
sudo -u postgres psql -c "\du" | grep sonarqube
 sonarqube | postgres | UTF8     | C.UTF-8 | C.UTF-8 | =Tc/postgres         +
           |          |          |         |         | sonarqube=CTc/postgres
 sonarqube |                                                      | {}
```



```
ubuntu@ip-172-31-90-110:~$ psql -U sonarqube -d sonarqube -h localhost
Password for user sonarqube:
psql (14.13 (Ubuntu 14.13-0ubuntu0.22.04.1))
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, bits: 256, compression: off)
Type "help" for help.
```

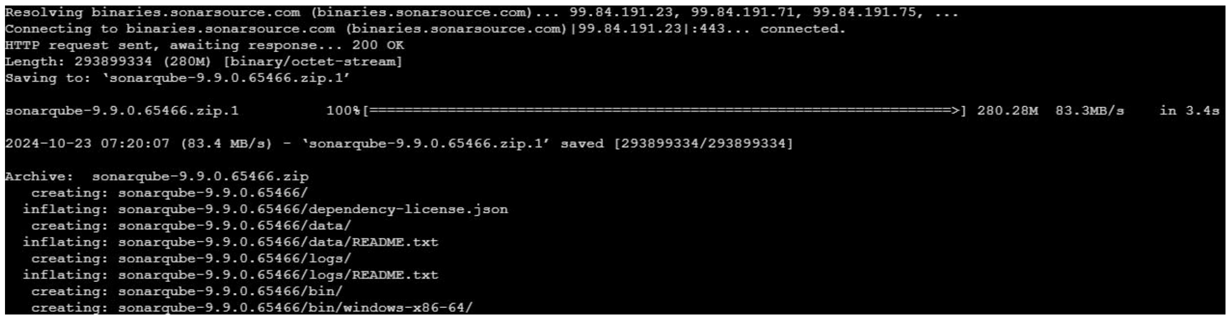


```
sonarqube=> \conninfo
You are connected to database "sonarqube" as user "sonarqube" on host
 "localhost" (address "127.0.0.1") at port "5432".
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, bi
ts: 256, compression: off)
```



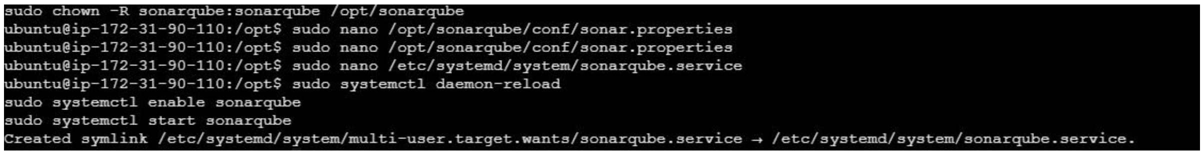
```
sonarqube=> \l
                              List of databases
   Name    |  Owner   | Encoding | Collate |  Ctype  |   Access privi
leges
-----------+----------+----------+---------+---------+---------------
---------
 postgres  | postgres | UTF8     | C.UTF-8 | C.UTF-8 |
 sonarqube | postgres | UTF8     | C.UTF-8 | C.UTF-8 | =Tc/postgres
        +
           |          |          |         |         | postgres=CTc/p
ostgres +
           |          |          |         |         | sonarqube=CTc/
postgres
 template0 | postgres | UTF8     | C.UTF-8 | C.UTF-8 | =c/postgres
        +
           |          |          |         |         | postgres=CTc/p
ostgres
 template1 | postgres | UTF8     | C.UTF-8 | C.UTF-8 | =c/postgres
        +
           |          |          |         |         | postgres=CTc/p
ostgres
```

- **Install SonarQube**

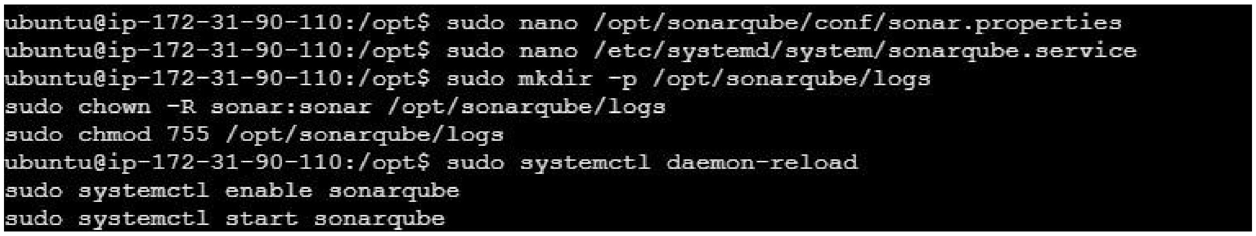


```
Resolving binaries.sonarsource.com (binaries.sonarsource.com)... 99.84.191.23, 99.84.191.71, 99.84.191.75, ...
Connecting to binaries.sonarsource.com (binaries.sonarsource.com)|99.84.191.23|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 293899334 (280M) [binary/octet-stream]
Saving to: 'sonarqube-9.9.0.65466.zip.1'

sonarqube-9.9.0.65466.zip.1         100%[===================================================================>] 280.28M  83.3MB/s    in 3.4s

2024-10-23 07:20:07 (83.4 MB/s) - 'sonarqube-9.9.0.65466.zip.1' saved [293899334/293899334]

Archive:  sonarqube-9.9.0.65466.zip
   creating: sonarqube-9.9.0.65466/
  inflating: sonarqube-9.9.0.65466/dependency-license.json
   creating: sonarqube-9.9.0.65466/data/
  inflating: sonarqube-9.9.0.65466/data/README.txt
   creating: sonarqube-9.9.0.65466/logs/
  inflating: sonarqube-9.9.0.65466/logs/README.txt
   creating: sonarqube-9.9.0.65466/bin/
   creating: sonarqube-9.9.0.65466/bin/windows-x86-64/
```

NAME:ARYAN PATANKAR **ROLL.NO:34** **CLASS:D15A**

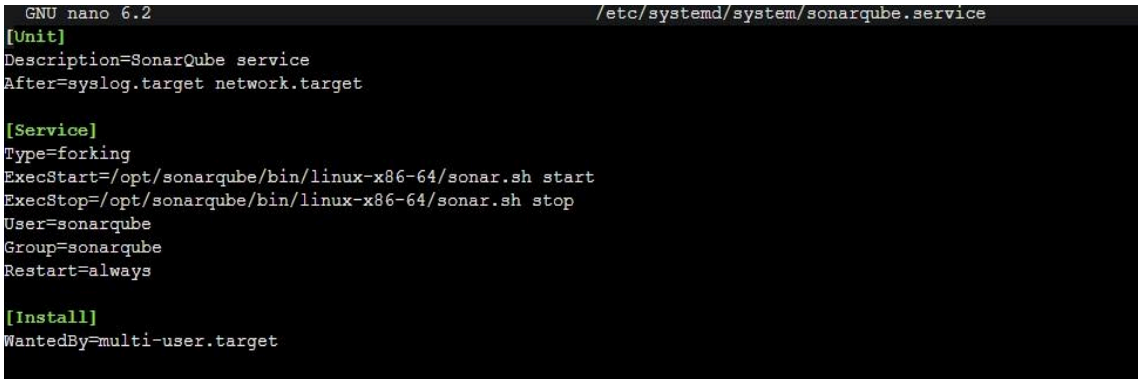


```
sudo chown -R sonarqube:sonarqube /opt/sonarqube
ubuntu@ip-172-31-90-110:/opt$ sudo nano /opt/sonarqube/conf/sonar.properties
ubuntu@ip-172-31-90-110:/opt$ sudo nano /opt/sonarqube/conf/sonar.properties
ubuntu@ip-172-31-90-110:/opt$ sudo nano /etc/systemd/system/sonarqube.service
ubuntu@ip-172-31-90-110:/opt$ sudo systemctl daemon-reload
sudo systemctl enable sonarqube
sudo systemctl start sonarqube
Created symlink /etc/systemd/system/multi-user.target.wants/sonarqube.service → /etc/systemd/system/sonarqube.service.
```



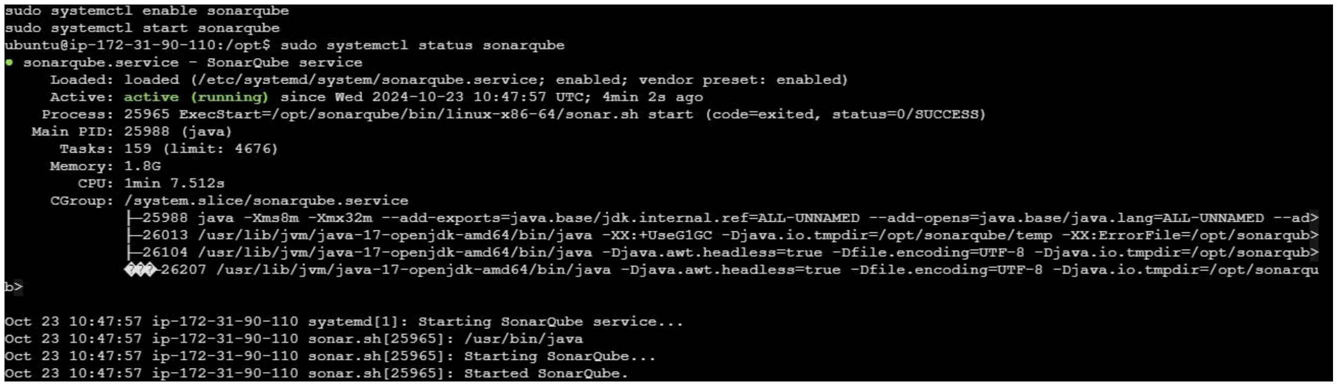
```
ubuntu@ip-172-31-90-110:/opt$ sudo nano /opt/sonarqube/conf/sonar.properties
ubuntu@ip-172-31-90-110:/opt$ sudo nano /etc/systemd/system/sonarqube.service
ubuntu@ip-172-31-90-110:/opt$ sudo mkdir -p /opt/sonarqube/logs
sudo chown -R sonar:sonar /opt/sonarqube/logs
sudo chmod 755 /opt/sonarqube/logs
ubuntu@ip-172-31-90-110:/opt$ sudo systemctl daemon-reload
sudo systemctl enable sonarqube
sudo systemctl start sonarqube
```

- **Configure SonarQube**



```
  GNU nano 6.2                                           /etc/systemd/system/sonarqube.service
[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=sonarqube
Group=sonarqube
Restart=always

[Install]
WantedBy=multi-user.target
```

- **Check for sonarqube status,here we see the sonarqube status as running**

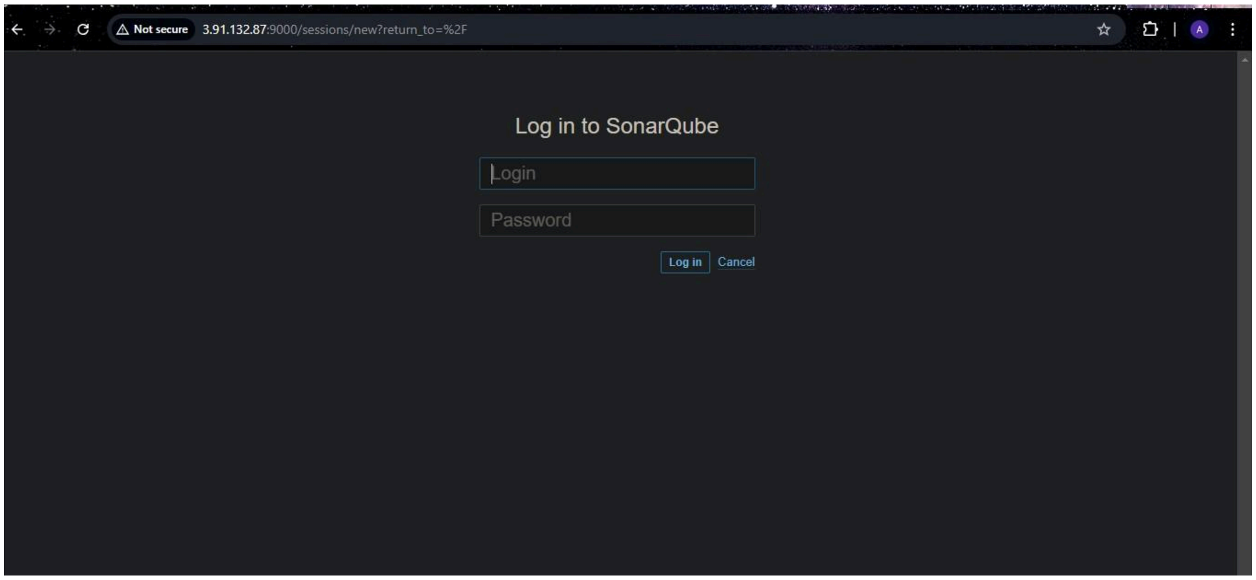


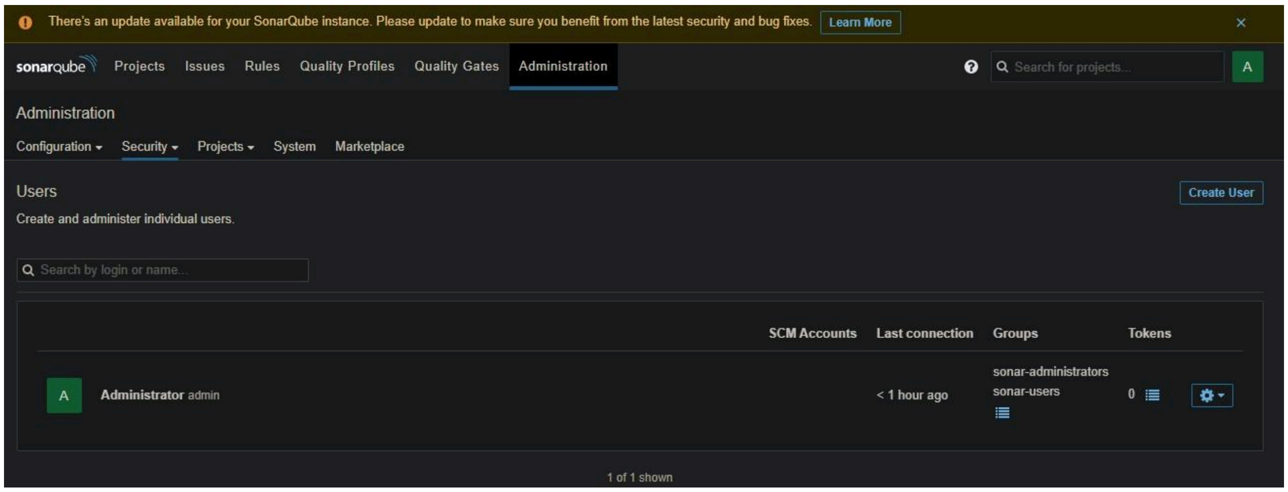
```
sudo systemctl enable sonarqube
sudo systemctl start sonarqube
ubuntu@ip-172-31-90-110:/opt$ sudo systemctl status sonarqube
● sonarqube.service - SonarQube service
     Loaded: loaded (/etc/systemd/system/sonarqube.service; enabled; vendor preset: enabled)
     Active: active (running) since Wed 2024-10-23 10:47:57 UTC; 4min 2s ago
    Process: 25965 ExecStart=/opt/sonarqube/bin/linux-x86-64/sonar.sh start (code=exited, status=0/SUCCESS)
   Main PID: 25988 (java)
      Tasks: 159 (limit: 4676)
     Memory: 1.8G
        CPU: 1min 7.512s
     CGroup: /system.slice/sonarqube.service
             ├─25988 java -Xms8m -Xmx32m --add-exports=java.base/jdk.internal.ref=ALL-UNNAMED --add-opens=java.base/java.lang=ALL-UNNAMED --ad>
             ├─26013 /usr/lib/jvm/java-17-openjdk-amd64/bin/java -XX:+UseG1GC -Djava.io.tmpdir=/opt/sonarqube/temp -XX:ErrorFile=/opt/sonarqub>
             ├─26104 /usr/lib/jvm/java-17-openjdk-amd64/bin/java -Djava.awt.headless=true -Dfile.encoding=UTF-8 -Djava.io.tmpdir=/opt/sonarqub>
             ���-26207 /usr/lib/jvm/java-17-openjdk-amd64/bin/java -Djava.awt.headless=true -Dfile.encoding=UTF-8 -Djava.io.tmpdir=/opt/sonarqu
b>

Oct 23 10:47:57 ip-172-31-90-110 systemd[1]: Starting SonarQube service...
Oct 23 10:47:57 ip-172-31-90-110 sonar.sh[25965]: /usr/bin/java
Oct 23 10:47:57 ip-172-31-90-110 sonar.sh[25965]: Starting SonarQube...
Oct 23 10:47:57 ip-172-31-90-110 sonar.sh[25965]: Started SonarQube.
```

Open a browser and navigate to `http://<your-EC2-IP>:9000`.

NAME:ARYAN PATANKAR **ROLL.NO:34** **CLASS:D15A**

Now go to the sonarqube website and login with credentials as admin and admin for both username and password.Create a JavaScript project called as Javascript-Test




There's an update available for your SonarQube instance. Please update to make sure you benefit from the latest security and bug fixes. Learn More

Users

Create and administer individual users.

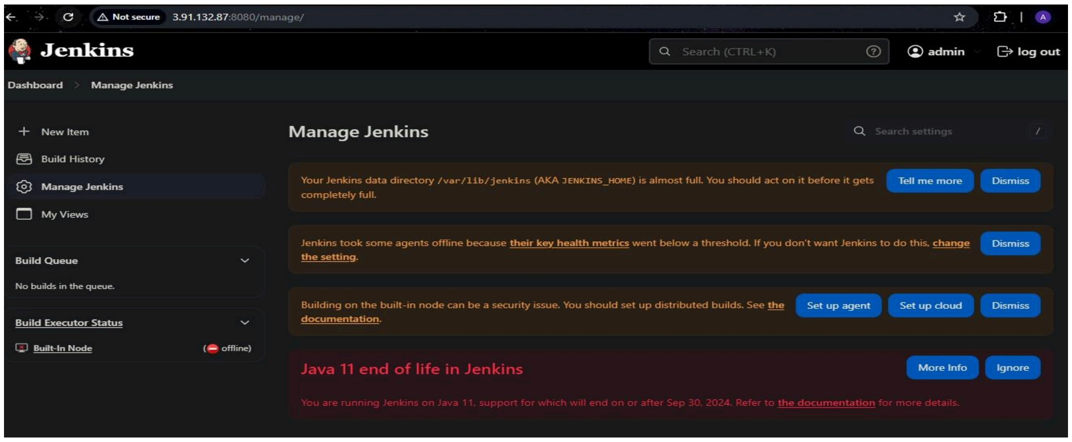
	SCM Accounts	Last connection	Groups	Tokens
Administrator admin		< 1 hour ago	sonar-administrators sonar-users	0

1 of 1 shown

NAME:ARYAN PATANKAR **ROLL.NO:34** **CLASS:D15A**

Step 4: Integrate Jenkins with SonarQube

1. **Install SonarQube Scanner Plugin in Jenkins**:
   - Go to **Manage Jenkins** → **Manage Plugins**.



3.91.132.87:8080/manage/

Dashboard > Manage Jenkins

Manage Jenkins

Your Jenkins data directory /var/lib/jenkins (AKA JENKINS_HOME) is almost full. You should act on it before it gets completely full.

Jenkins took some agents offline because their key health metrics went below a threshold. If you don't want Jenkins to do this, change the setting.

Building on the built-in node can be a security issue. You should set up distributed builds. See the documentation.

Java 11 end of life in Jenkins

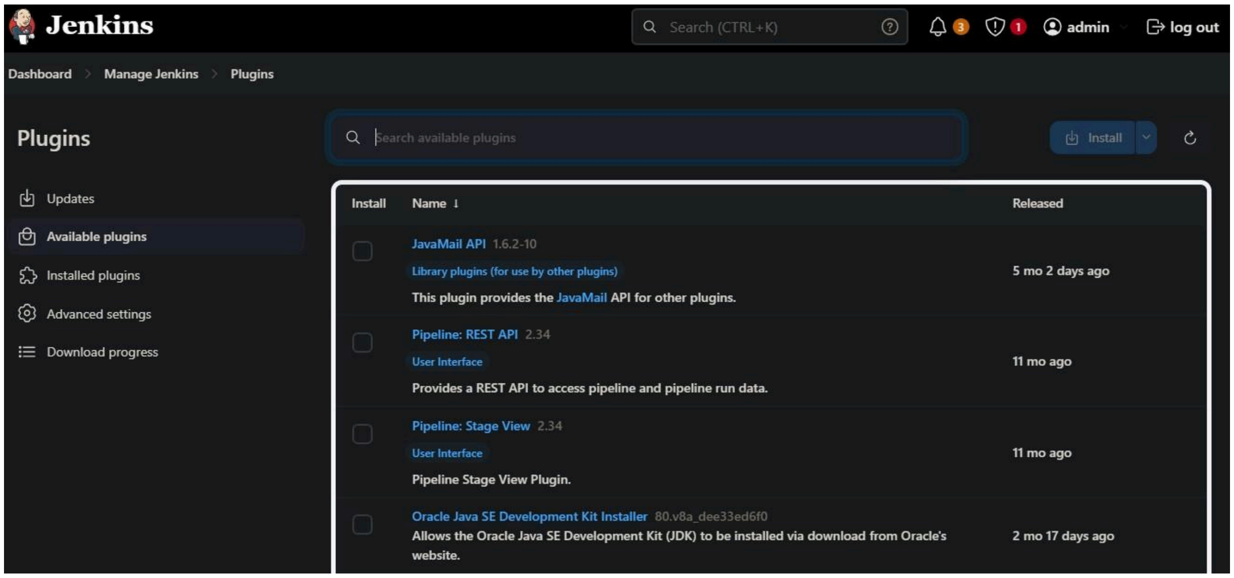
You are running Jenkins on Java 11, support for which will end on or after Sep 30, 2024. Refer to the documentation for more details.

Build Queue

No builds in the queue.

Build Executor Status

Built-In Node (offline)

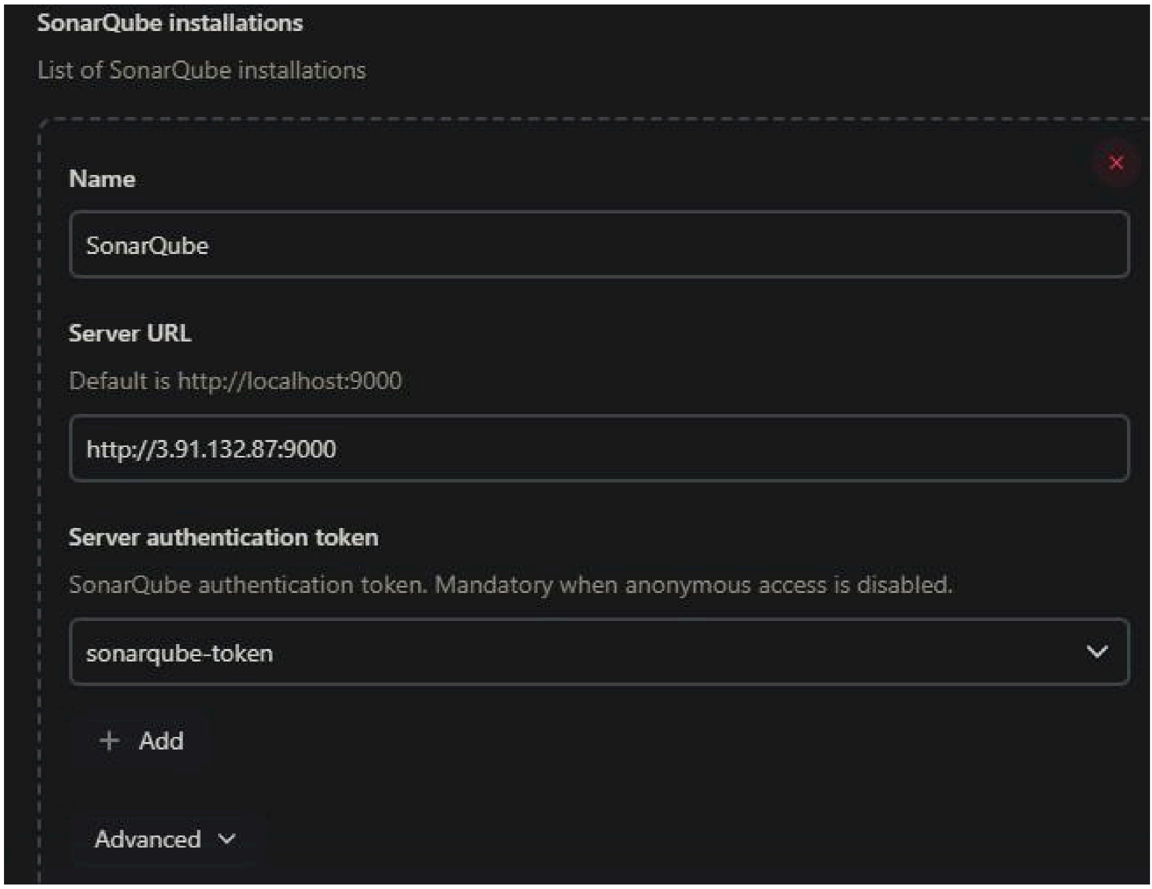


Dashboard > Manage Jenkins > Plugins

Plugins

Install	Name ↓	Released
☐	JavaMail API 1.6.2-10 Library plugins (for use by other plugins) This plugin provides the JavaMail API for other plugins.	5 mo 2 days ago
☐	Pipeline: REST API 2.34 User Interface Provides a REST API to access pipeline and pipeline run data.	11 mo ago
☐	Pipeline: Stage View 2.34 User Interface Pipeline Stage View Plugin.	11 mo ago
☐	Oracle Java SE Development Kit Installer 80.v8a_dee33ed6f0 Allows the Oracle Java SE Development Kit (JDK) to be installed via download from Oracle's website.	2 mo 17 days ago

   - Search for **SonarQube Scanner** and install it.
2. **Configure SonarQube Server in Jenkins**:
   - Go to **Manage Jenkins** → **Configure System**.
   - Find the **SonarQube servers** section and click **Add SonarQube**.
   - Enter:
     - **Name**: SonarQube
     - **Server URL**: `http://<your-local-IP>:9000` (use your local machine's IP, not `localhost`).
     - **Server authentication token**: Generate a token in SonarQube by going to **My Account** → **Security** → **Generate Tokens**.



SonarQube installations

List of SonarQube installations

Name

SonarQube

Server URL

Default is http://localhost:9000

http://3.91.132.87:9000

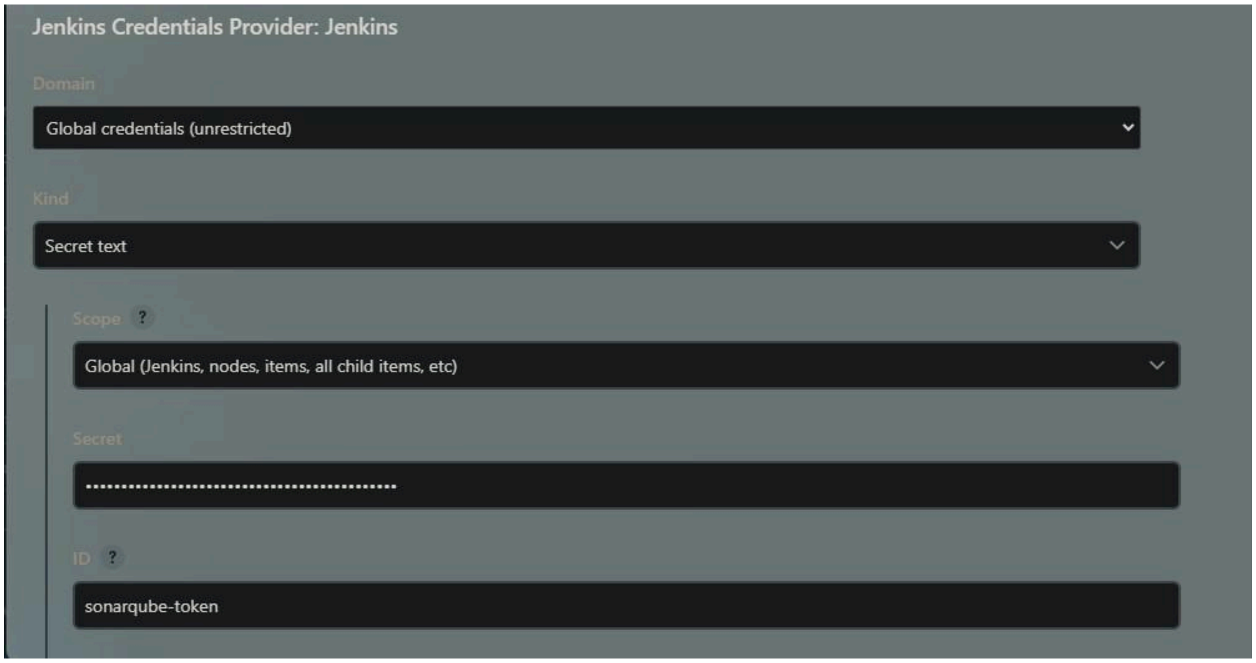
Server authentication token

SonarQube authentication token. Mandatory when anonymous access is disabled.

sonarqube-token

3. **Add Credentials in Jenkins**:
    - Go to **Manage Jenkins** → **Manage Credentials** → **Add a new credential**.
    - Add your SonarQube token as a **Secret Text** credential.

NAME:ARYAN PATANKAR ROLL.NO:34 CLASS:D15A


Jenkins Credentials Provider: Jenkins

Domain: Global credentials (unrestricted)

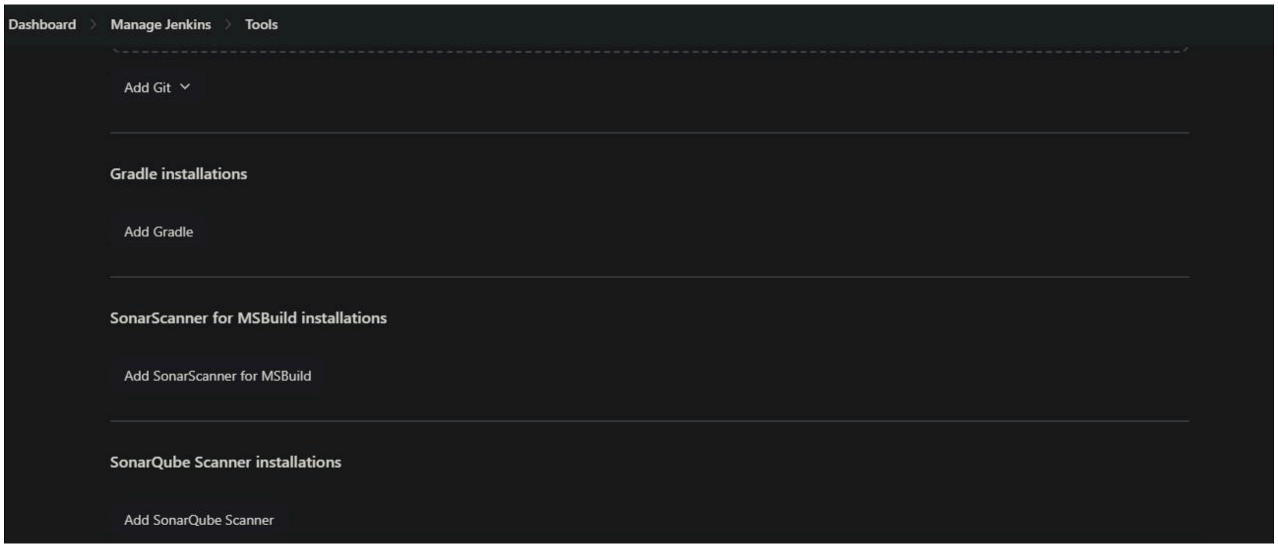
Kind: Secret text

Scope: Global (Jenkins, nodes, items, all child items, etc)

Secret: ••

ID: sonarqube-token

4. **Set sonarqube Scanner**
   Manage Jenkins → Tools


Dashboard > Manage Jenkins > Tools

Add Git

Gradle installations

Add Gradle

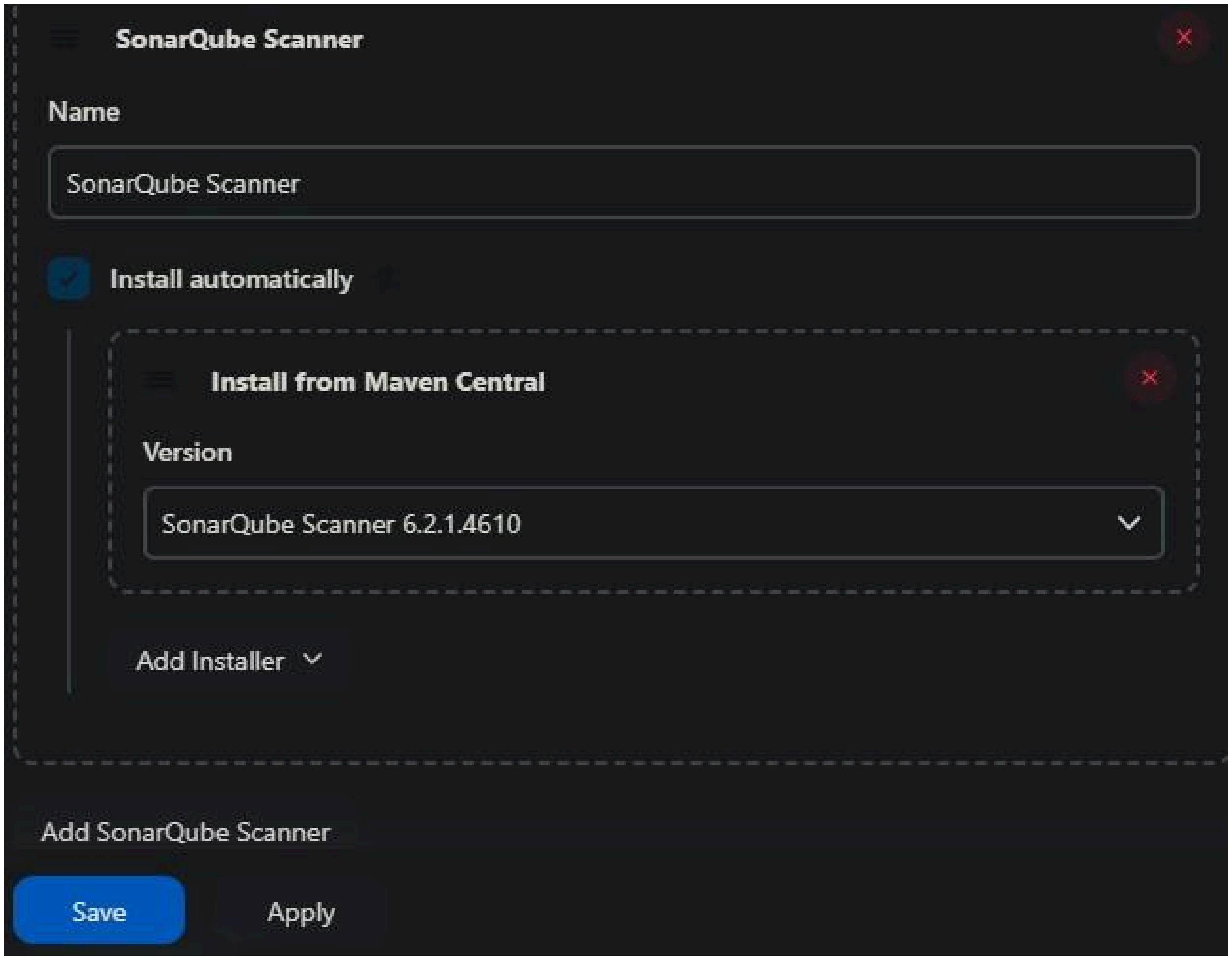
SonarScanner for MSBuild installations

Add SonarScanner for MSBuild

SonarQube Scanner installations

Add SonarQube Scanner

NAME:ARYAN PATANKAR ROLL.NO:34 CLASS:D15A



SonarQube Scanner

Name

SonarQube Scanner

Install automatically ✓

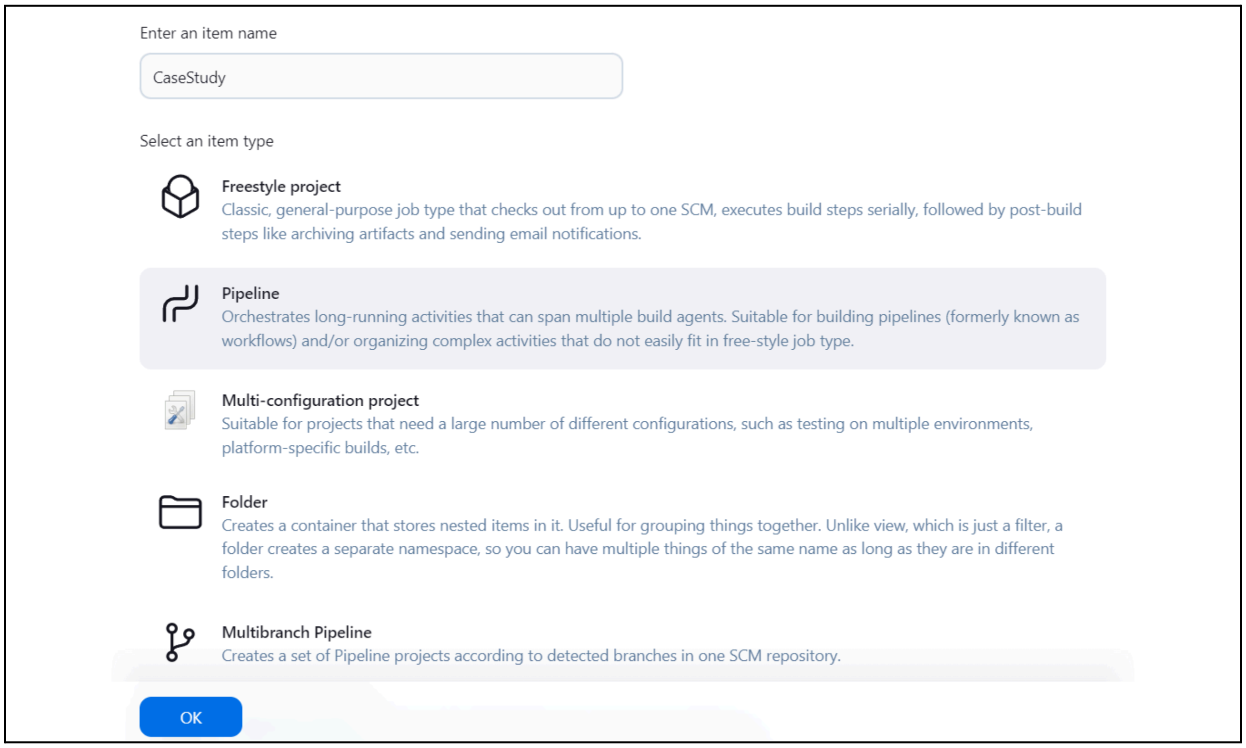
Install from Maven Central

Version

SonarQube Scanner 6.2.1.4610

NAME:ARYAN PATANKAR **ROLL.NO:34** **CLASS:D15A**

Step 5: Create Pipeline project



Enter an item name

CaseStudy

Select an item type

Freestyle project
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.

Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

Folder
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

Multibranch Pipeline
Creates a set of Pipeline projects according to detected branches in one SCM repository.

Pipeline code:

```
pipeline {
    agent any

    stages {
        stage('Checkout') {
            steps {
                dir('/home/ubuntu/test-project') {
                    sh 'pwd'
                    sh 'ls -la'
                }
            }
        }

        stage('SonarQube Analysis') {
            steps {
                withSonarQubeEnv('SonarQube') {
                    sh """
/var/lib/jenkins/tools/hudson.plugins.sonar.SonarRunnerInstallation/SonarScanner/bin/sonar-sca
nner \
                        -Dsonar.projectKey=test-project \
                        -Dsonar.projectName='Test Project' \
                        -Dsonar.sources=/home/ubuntu/test-project \
                        -Dsonar.host.url=http://3.91.132.87:9000 \
                        -Dsonar.login=squ_5666dac44e95402542731ba9143cee79b4cb64a5 \
                        -Dsonar.sourceEncoding=UTF-8 \
                        -Dsonar.javascript.node.path=/usr/bin/node \
                        -Dsonar.javascript.node.maxspace=2048
                    """
                }
            }
        }
    }
}
```

JS Code-

```
// Global variables - multiple bad practices
var globalVar = "I am global";
var anotherGlobal = "Also global";
var unused_global = "Never used";  // Unused variable

// Function with multiple issues: unused params, variables, and complex nesting
function badFunction(unusedParam1, unusedParam2) {
    var unusedVar = "never used";
    var x = 1;
    x = x;  // Self assignment

    if (true) {
        console.log("Always true");
        while (true) {  // Infinite loop
            if (x > 0) break;
        }
    } else {
        console.log("Unreachable code");
    }
    return;  // Unnecessary return
}
```

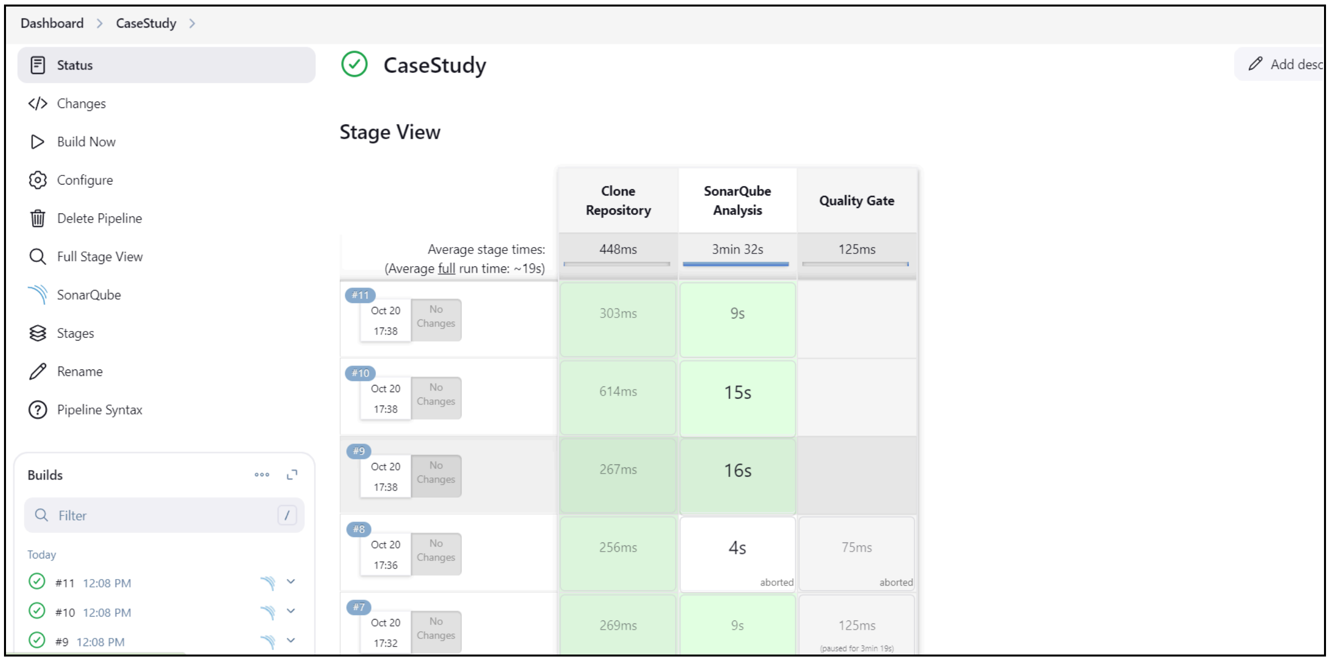
NAME:ARYAN PATANKAR ROLL.NO:34 CLASS:D15A

```
// Duplicate code blocks with slight variations
function duplicate1() {
    console.log("Start");
    for(var i = 0; i < 10; i++) {  // Using var instead of let
        console.log(i);
        console.log(i * 2);
        console.log(i * 3);
        if(i == 5) continue;  // Unnecessary continue
    }
    console.log("End");
}

function duplicate2() {
    console.log("Begin");
    for(var i = 0; i < 10; i++) {  // Using var instead of let
        console.log(i);
        console.log(i * 2);
        console.log(i * 3);
        if(i == 6) continue;  // Unnecessary continue
    }
    console.log("Finish");
}

// Multiple security issues
function securityRisk(input) {
    eval(input);  // Never use eval
    new Function(input)();  // Another dangerous eval-like construct
    document.write(input);  // XSS vulnerability
}

// Extremely complex function with high cognitive complexity
function complexFunction(a, b, c) {
    let result = 0;
    if (a > 0) {
        if (b > 0) {
            if (c > 0) {
                while (a > 0) {
                    for (let i = 0; i < b; i++) {
                        if (c > i) {
                            result += a + b + c;
                        } else {
                            result += a + b;
                        }
                    }
                    a--;
                }
            } else {
                result = a + b;
            }
        } else {
            if (c > 0) {
                result = a + c;
            } else {
                result = a;
            }
        }
    }
    return result;
}

// Multiple variable shadowing issues
function shadowingIssue() {
    let x = 5;
    let y = 10;
    {
        let x = 10;  // Shadows outer x
        {
            let x = 15;  // Shadows again
            let y = 20;  // Shadows outer y
            console.log(x, y);
        }
    }
    return x;
}

// Multiple empty catch blocks and undefined variables
try {
    undefinedFunction();  // Calling undefined function
    nonExistentVariable.property;  // Accessing undefined variable
} catch(e) {
    // Empty catch block
}

try {
    riskyOperation();  // Another undefined function
} catch(e) {
    // Another empty catch block
}
```

NAME:ARYAN PATANKAR **ROLL.NO:34** **CLASS:D15A**

```
// Magic numbers throughout function
function calculateTotal(quantity) {
    const basePrice = 24.99;  // Magic number
    const taxRate = 1.08;     // Magic number
    const discount = 0.15;    // Magic number
    return quantity * taxRate * basePrice * (1 - discount) + 4.99;  // More magic numbers
}

// Function with too many parameters
function tooManyParams(a, b, c, d, e, f, g, h, i, j) {
    return a + b + c + d + e + f + g + h + i + j;
}

// Multiple calls to problematic functions
badFunction("unused1", "unused2");
duplicate1();
duplicate2();
securityRisk("alert('xss')");
complexFunction(1, 2, 3);
shadowingIssue();
calculateTotal(5);
tooManyParams(1,2,3,4,5,6,7,8,9,10);
```

After adding pipeline : Build project by clicking **Build Now**



Dashboard > CaseStudy >

CaseStudy

Stage View

	Clone Repository	SonarQube Analysis	Quality Gate
Average stage times: (Average full run time: ~19s)	448ms	3min 32s	125ms
#11 Oct 20 17:38 No Changes	303ms	9s	
#10 Oct 20 17:38 No Changes	614ms	15s	
#9 Oct 20 17:38 No Changes	267ms	16s	
#8 Oct 20 17:36 No Changes	256ms	4s aborted	75ms aborted
#7 Oct 20 17:32 No Changes	269ms	9s	125ms (paused for 3min 19s)

Builds

Today

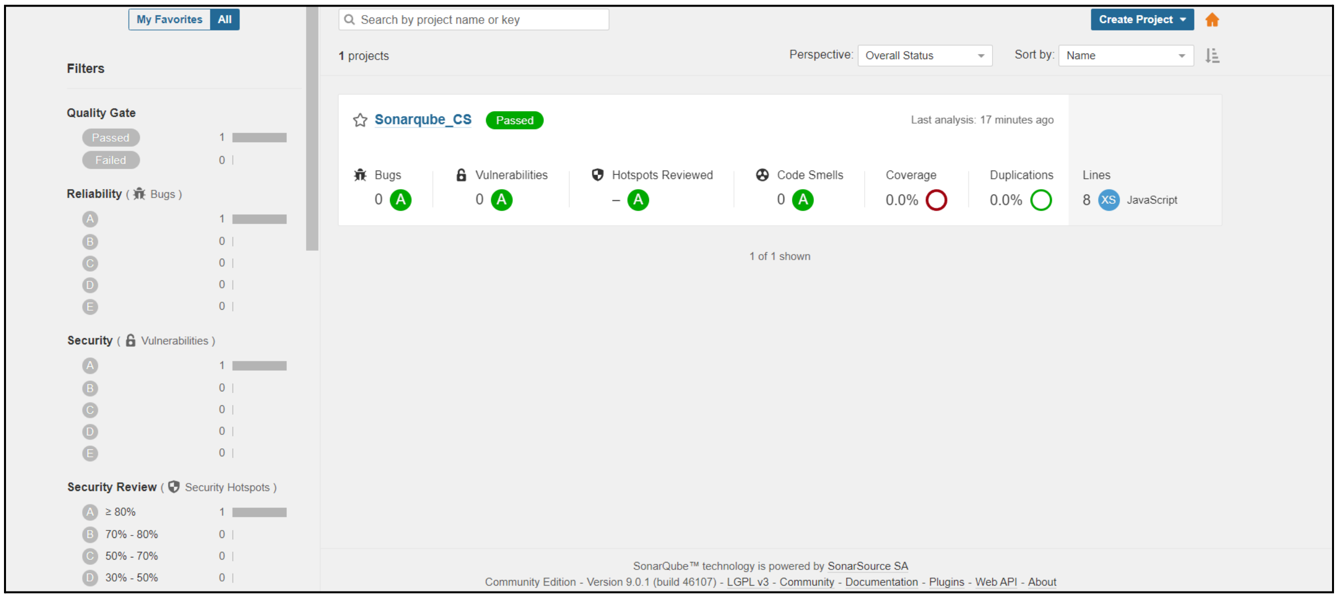
#11 12:08 PM

#10 12:08 PM

#9 12:08 PM

NAME:ARYAN PATANKAR **ROLL.NO:34** **CLASS:D15A**

Sonarqube:



My Favorites | All

Search by project name or key

Create Project

Filters

1 projects

Perspective: Overall Status Sort by: Name

Quality Gate
Passed 1
Failed 0

Reliability (Bugs)
A 1
B 0
C 0
D 0
E 0

Security (Vulnerabilities)
A 1
B 0
C 0
D 0
E 0

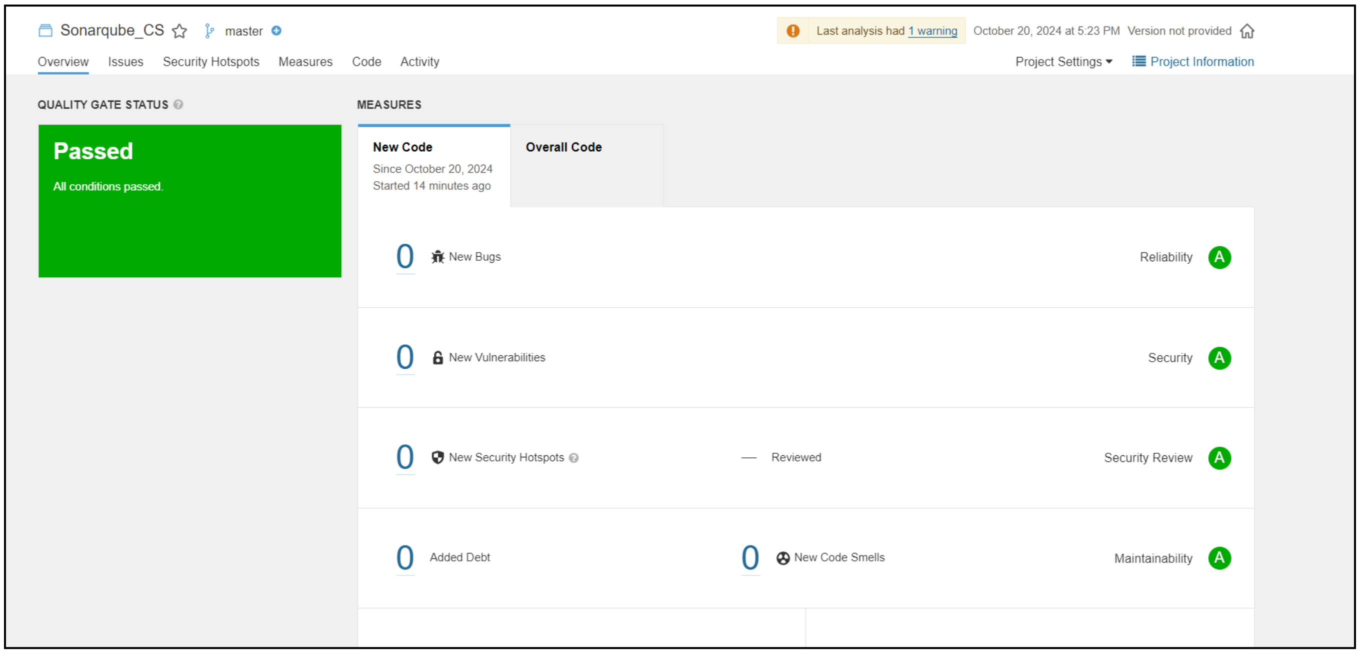
Security Review (Security Hotspots)
A ≥ 80% 1
B 70% - 80% 0
C 50% - 70% 0
D 30% - 50% 0

Sonarqube_CS Passed Last analysis: 17 minutes ago

Bugs	Vulnerabilities	Hotspots Reviewed	Code Smells	Coverage	Duplications	Lines
0 A	0 A	– A	0 A	0.0%	0.0%	8 XS JavaScript

1 of 1 shown

SonarQube™ technology is powered by SonarSource SA
Community Edition - Version 9.0.1 (build 46107) - LGPL v3 - Community - Documentation - Plugins - Web API - About



Sonarqube_CS master

Last analysis had 1 warning October 20, 2024 at 5:23 PM Version not provided

Overview Issues Security Hotspots Measures Code Activity Project Settings Project Information

QUALITY GATE STATUS

Passed
All conditions passed.

MEASURES

New Code
Since October 20, 2024
Started 14 minutes ago

Overall Code

0 New Bugs Reliability A

0 New Vulnerabilities Security A

0 New Security Hotspots — Reviewed Security Review A

0 Added Debt 0 New Code Smells Maintainability A

Conclusion:

In conclusion, this case study demonstrates the effective use of **Continuous Integration (CI)** by integrating **Jenkins**, **AWS Cloud9**, and **SonarQube** to maintain code quality in a JavaScript project. By automating code analysis, the setup ensures that potential issues like bugs, code smells, and security vulnerabilities are detected early in the development cycle, enhancing the overall reliability and maintainability of the codebase.

NAME:ARYAN PATANKAR **ROLL.NO:34** **CLASS:D15A**

Through the use of **SonarQube**, developers receive continuous feedback on their code, promoting best practices and preventing the introduction of low-quality code into production. The cloud-based environment provided by **AWS Cloud9** offers flexibility and scalability, allowing for easy collaboration and rapid development, while Jenkins automates the entire process, reducing manual effort and increasing productivity.