



SONARQUBE

What is CQA
Why CQA important
SonarQube Setup
Analyze Bugs in SonarQube

SWIPE LEFT

CHARAN SURYA KILANA

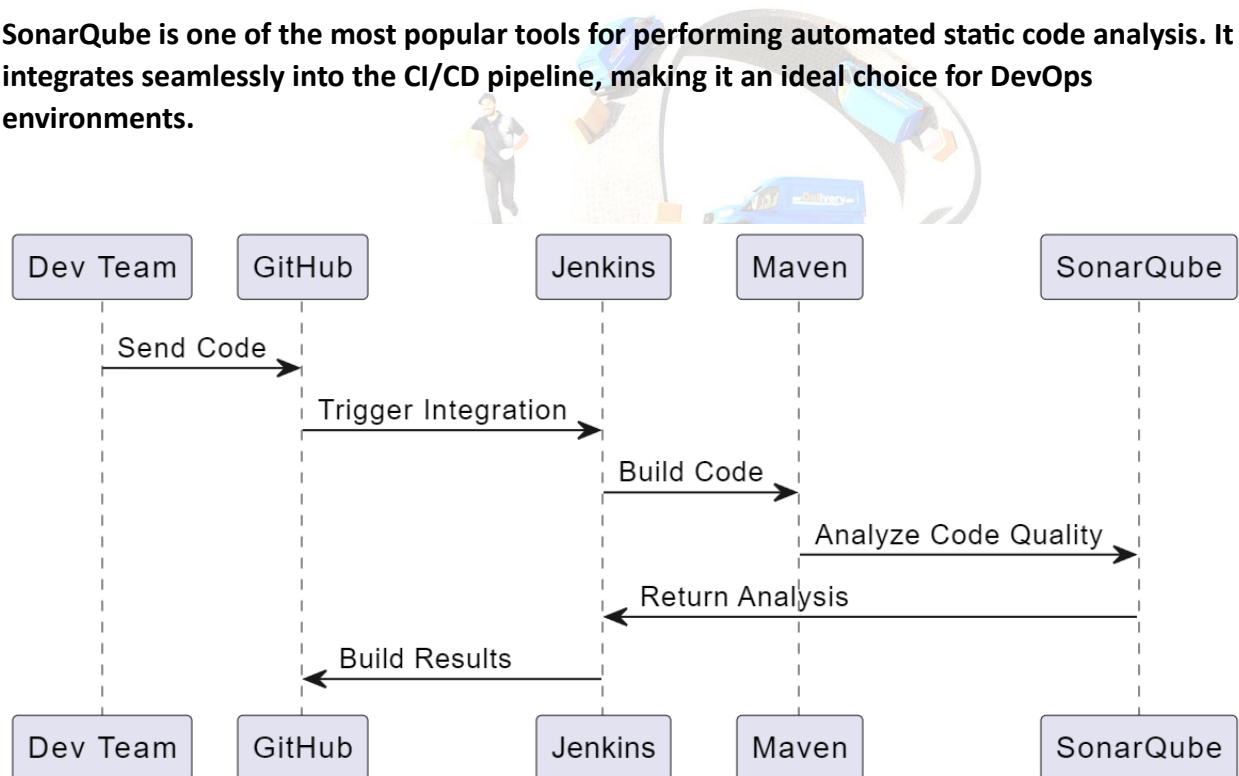
What is Code Quality Analysis?

Code Quality Analysis involves reviewing and analyzing the structure, readability, and functionality of code to identify potential bugs, vulnerabilities, and code smells. By doing so, we ensure the software is maintainable, secure, and reliable.

Why is CQA Important for DevOps Engineers?

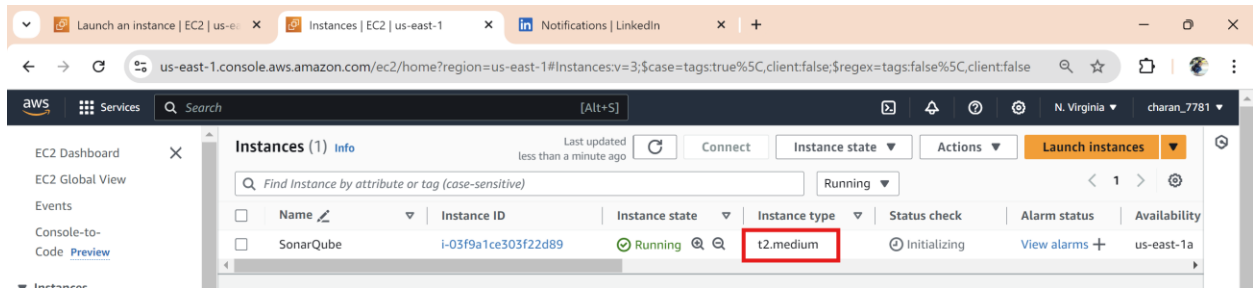
For a DevOps engineer, CQA helps ensure that all code delivered in the CI/CD pipeline adheres to best practices, meets security standards, and is free from defects that could disrupt continuous delivery. The earlier issues are identified, the faster they can be resolved, preventing critical breakdowns in production.

SonarQube is one of the most popular tools for performing automated static code analysis. It integrates seamlessly into the CI/CD pipeline, making it an ideal choice for DevOps environments.

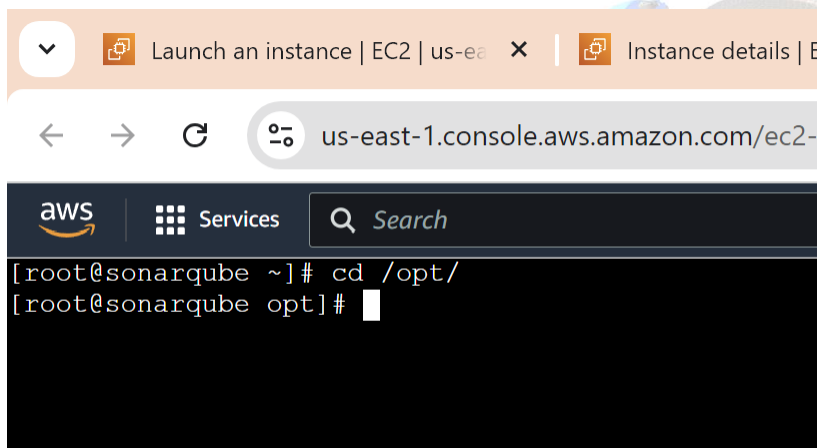


Setting Up SonarQube and Integrating it with Jenkins

STEP-1: Launch an EC2 instance, ensuring port 9000 is allowed, and select an instance type of t2.medium.

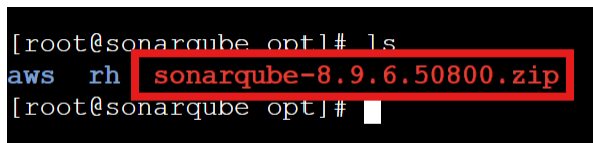
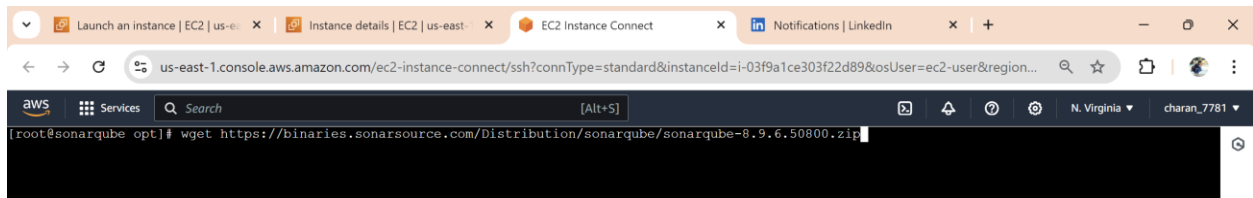


STEP-2: Change the directory to /opt/.

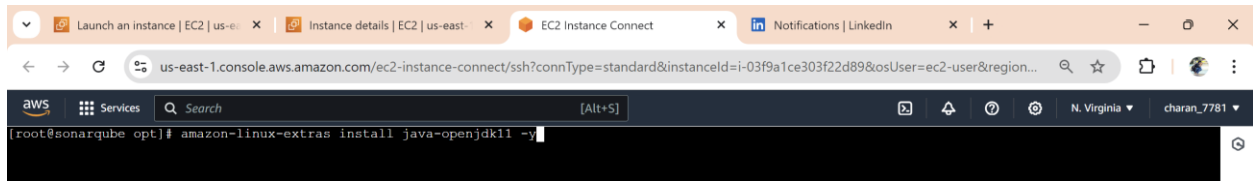


STEP-3: Download the SonarQube file from the official SonarSource website.

`wget https://binaries.sonarsource.com/Distribution/sonarqube/sonarqube-8.9.6.50800.zip`



Step-4: Install Java OpenJDK 11.

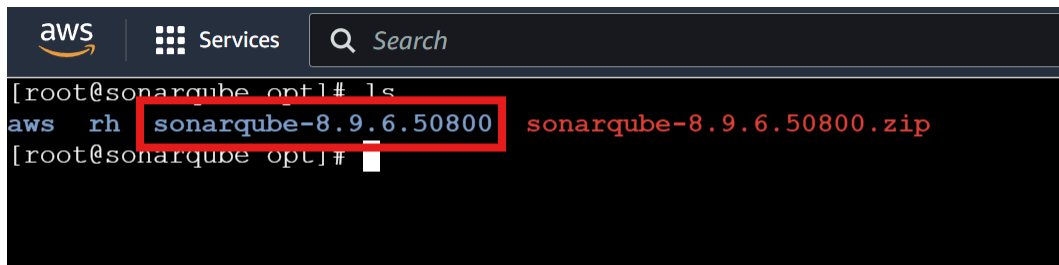


A screenshot of the AWS Management Console terminal window. The browser tabs at the top include 'Launch an instance | EC2 | us-east-1', 'Instance details | EC2 | us-east-1', 'EC2 Instance Connect', and 'Notifications | LinkedIn'. The address bar shows the URL 'us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-03f9a1ce303f22d89&osUser=ec2-user®ion=us-east-1'. The terminal window has an AWS logo and a search bar. The command being entered is 'amazon-linux-extras install java-openjdk11 -y'.

```
[root@sonarqube opt]# amazon-linux-extras install java-openjdk11 -y
```

Step-5: Unzip the file sonarqube-8.9.6.50800.zip.

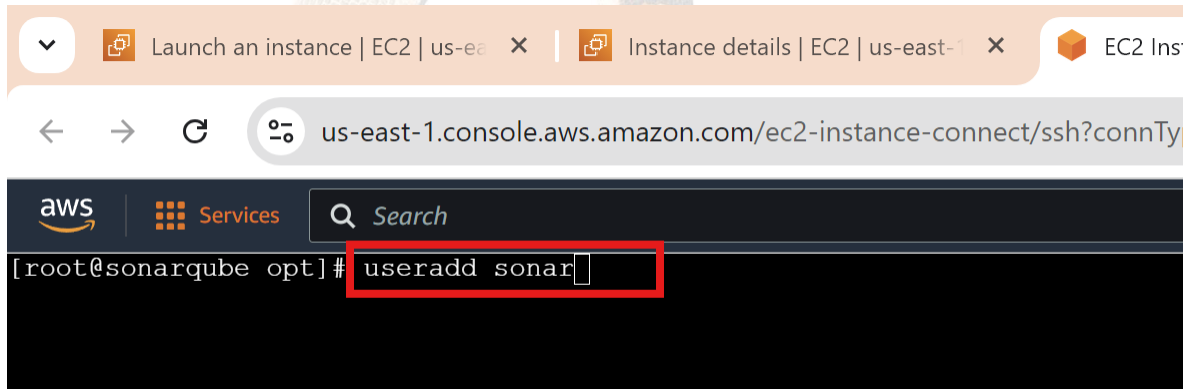
unzip sonarqube-8.9.6.50800.zip



A screenshot of the AWS Management Console terminal window. The terminal shows the command 'unzip sonarqube-8.9.6.50800.zip' being entered. The filename 'sonarqube-8.9.6.50800' is highlighted with a red box. The terminal window has an AWS logo and a search bar.

```
[root@sonarqube opt]# unzip sonarqube-8.9.6.50800.zip
```

STEP-6: Add a new user for SonarQube



A screenshot of the AWS Management Console terminal window. The terminal shows the command 'useradd sonar' being entered. The command is highlighted with a red box. The terminal window has an AWS logo and a search bar.

```
[root@sonarqube opt]# useradd sonar
```

STEP-7: Change the ownership of the SonarQube files to the newly created sonar user.

```
aws | Services | Search [Alt+S]
[root@sonarqube opt]# sudo chown -R sonar:sonar /opt/sonarqube-8.9.6.50800
[root@sonarqube opt]# ls
aws rh sonarqube-8.9.6.50800 sonarqube-8.9.6.50800.zip
[root@sonarqube opt]# ll
total 255920
drwxr-xr-x 4 root root      33 Sep  3 22:46 aws
drwxr-xr-x 2 root root      6 Aug 16  2018 rh
drwxr-xr-x 11 sonar sonar    172 Dec 20  2021 sonarqube-8.9.6.50800
-rw-r--r-- 1 root root 262060020 Feb 16  2022 sonarqube-8.9.6.50800.zip
[root@sonarqube opt]#
```

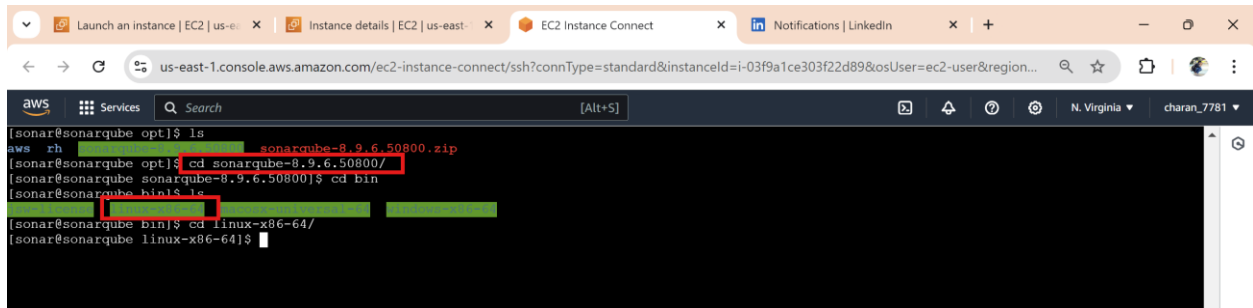
Step-8: Give full permissions to the SonarQube directory.

```
Launch an instance | EC2 | us-east-1 | Instance details | EC2 | us-east-1 | EC2 Instance C
us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=st
aws | Services | Search [Alt+S]
[root@sonarqube opt]# chmod -R 777 /opt/sonarqube-8.9.6.50800
```

STEP-9: Switch to the Sonar user and change directory to /opt/.

```
aws | Services | Search
[root@sonarqube opt]# su - sonar
[sonar@sonarqube ~]$ cd /opt/
[sonar@sonarqube opt]$
```

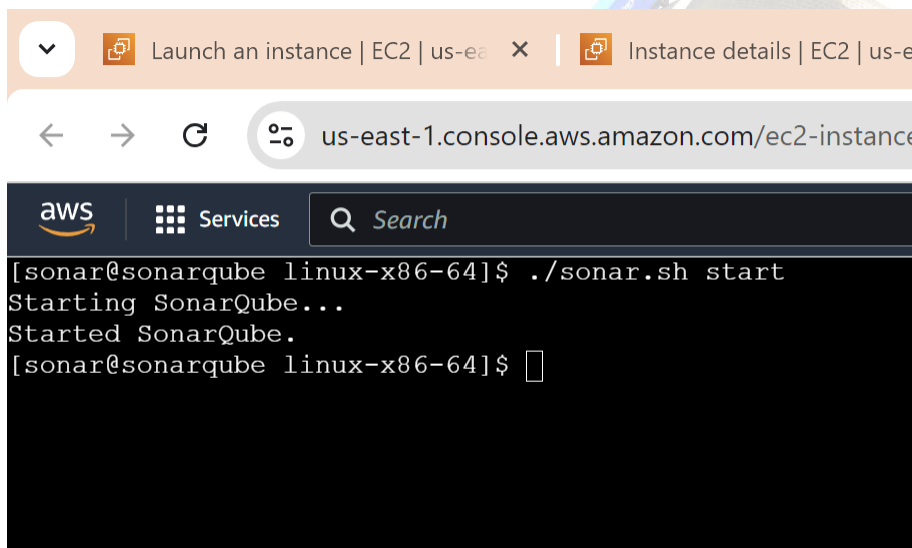
STEP-10: Navigate to the SonarQube directory and start the SonarQube server in bin folder in Linux.



A screenshot of the AWS Management Console showing an EC2 Instance Connect session. The terminal window displays the following commands and output:

```
[sonar@sonarqube opt]$ ls
aws rh [redacted] sonarqube-8.9.6.50800.zip
[sonar@sonarqube opt]$ cd sonarqube-8.9.6.50800/
[sonar@sonarqube sonarqube-8.9.6.50800]$ cd bin
[sonar@sonarqube bin]$ ls
[redacted]
[sonar@sonarqube bin]$ cd linux-x86-64/
[sonar@sonarqube linux-x86-64]$
```

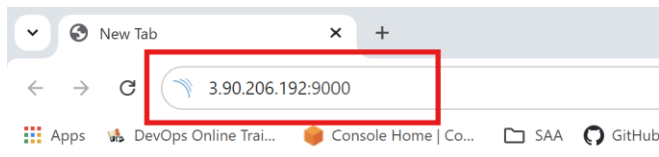
Step-11: Start the server.



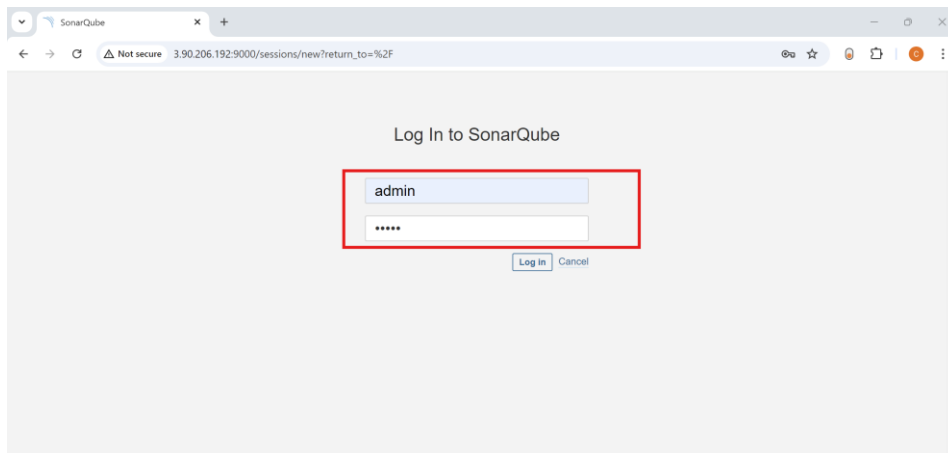
A screenshot of the AWS Management Console showing an EC2 Instance Connect session. The terminal window displays the following command and output:

```
[sonar@sonarqube linux-x86-64]$ ./sonar.sh start
Starting SonarQube...
Started SonarQube.
[sonar@sonarqube linux-x86-64]$
```

Step-12: Access the server



Step-13: Username is admin and password is admin.

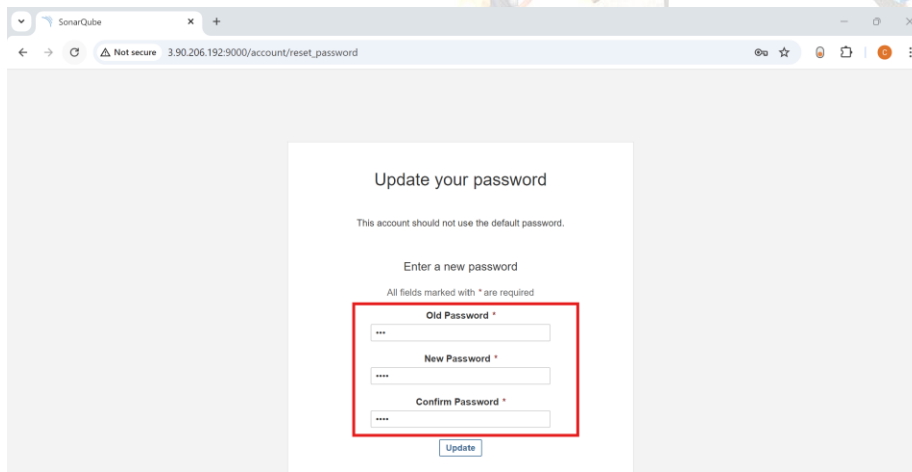


Log In to SonarQube

admin

Log In Cancel

Step-14: Set new password.



Update your password

This account should not use the default password.

Enter a new password

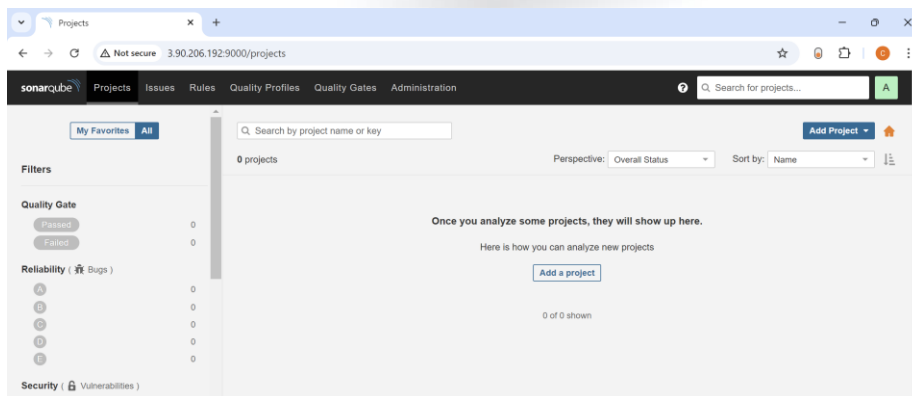
All fields marked with * are required

Old Password *

New Password *

Confirm Password *

Update



sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration

Search for projects...

Search by project name or key

0 projects

Perspective: Overall Status Sort by: Name

Once you analyze some projects, they will show up here.

Here is how you can analyze new projects

Add a project

0 of 0 shown

My Favorites All

Quality Gate

Passed 0

Failed 0

Reliability (Bugs)

0

0

0

0

0

0

Security (Vulnerabilities)

Step-15: Setup Jenkins

Step-16: Install plugin Sonar Scanner

Step-17: To get the code in the artifact repository, you need to fetch the code from GitHub, build it into a packaged format, and then store it in the artifact repository (such as Nexus).

Write a pipeline for this:

You should git and maven on Jenkins server, instead you can simply go to

Manage Jenkins > global tool configuration>> maven installation

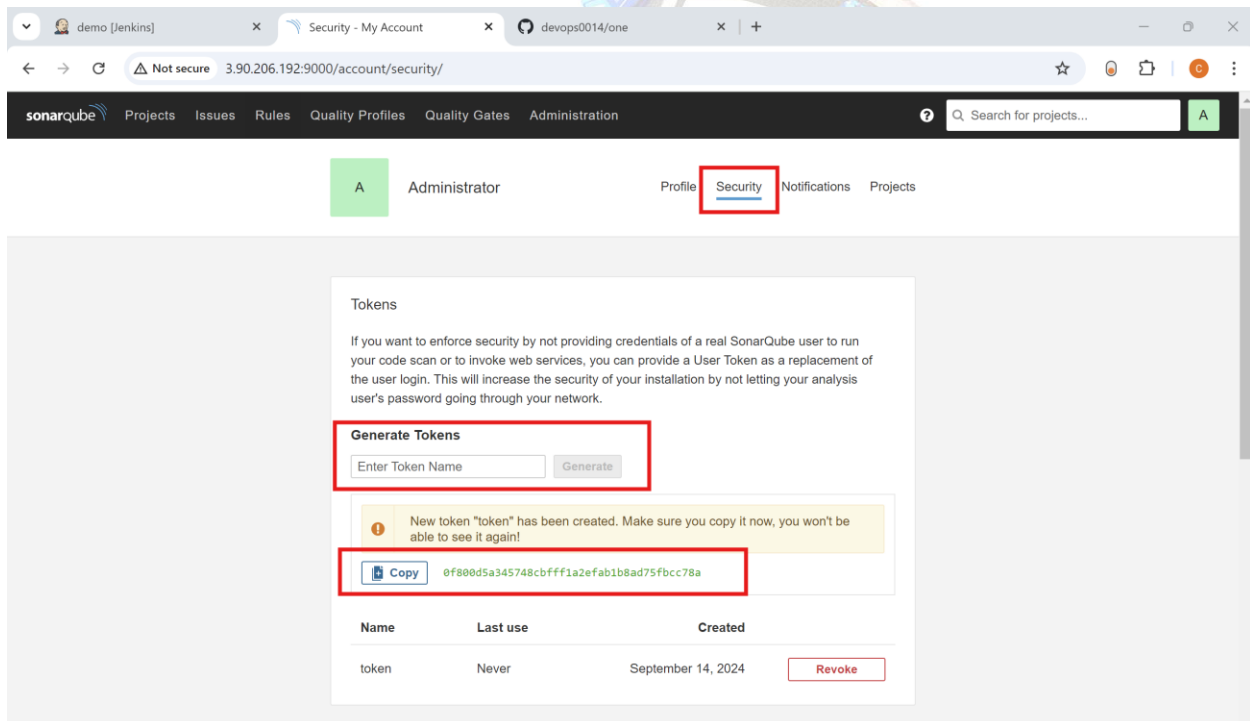
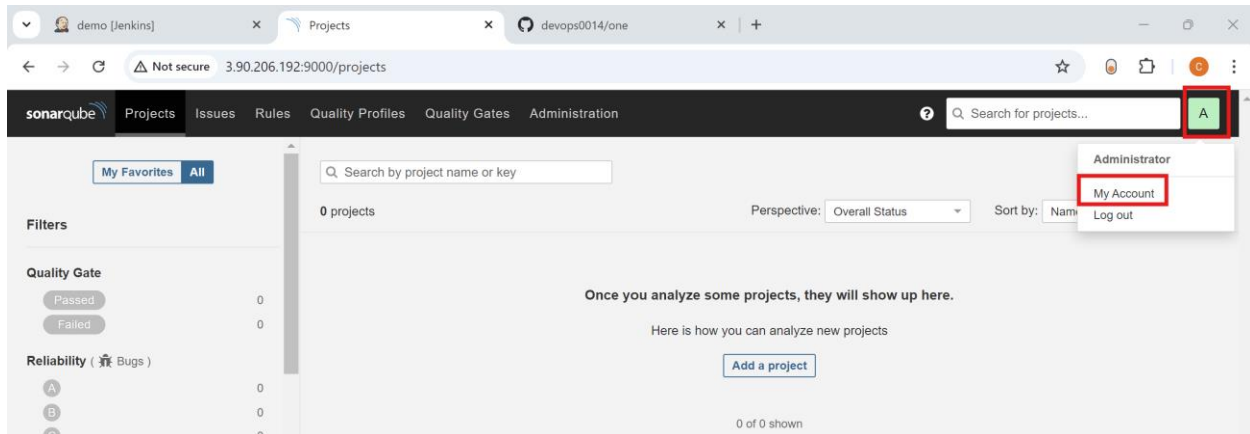
Installing Maven on a Jenkins server can result in two different Java versions: one for Jenkins and another for Maven. If not managed properly, these versions could override each other, leading to potential conflicts during builds. To avoid issues, ensure both tools use the same Java version or configure Maven to use the correct one explicitly.

The top screenshot shows the Jenkins 'Maven installations' configuration page. The 'Name' field is set to 'maven3' and the 'Version' is set to '3.8.6'. The 'Install automatically' checkbox is checked. The bottom screenshot shows the Jenkins 'Pipeline' configuration page for a job named 'demo'. The 'Definition' is set to 'Pipeline script'. The script is as follows:

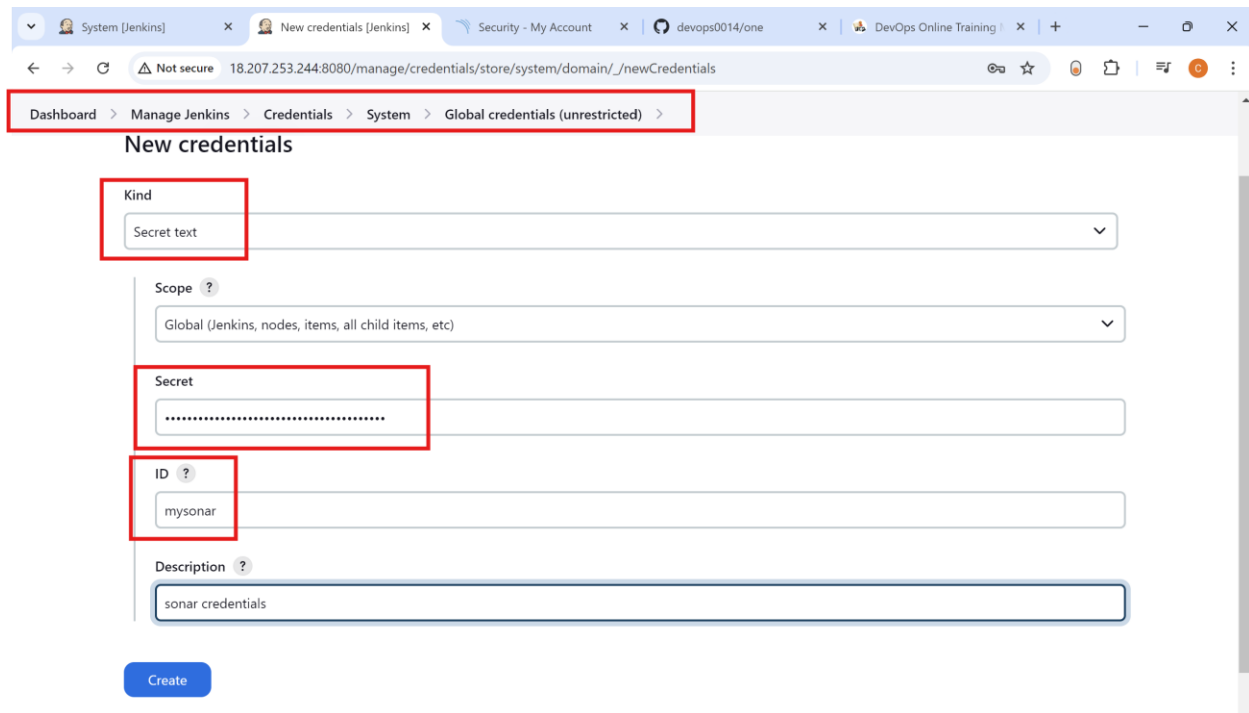
```
1 node
2 {
3   stage("Code")
4   {
5     git "https://github.com/devops0014/one.git"
6   }
7   stage("Build")
8   {
9     def mavenHome = tool name: "maven3", type: "maven"
10    def mavenCMD = "${mavenHome}/bin/mvn"
11    sh "${mavenCMD} clean package"
12  }
13 }
```

The 'Use Groovy Sandbox' checkbox is checked. The 'Save' button is highlighted.

Step-18: Go to My Account and generate token



Step-19: Go to Credentials and add sonar token.

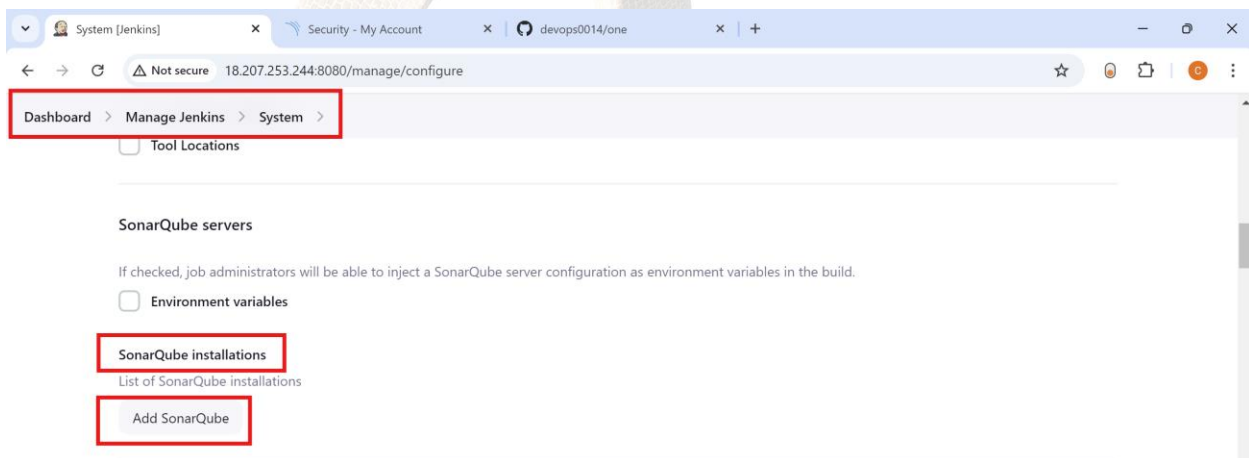


The screenshot shows the Jenkins 'New credentials' page. The breadcrumb navigation at the top is 'Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted) >', with the last part highlighted by a red box. The form contains the following fields:

- Kind:** A dropdown menu with 'Secret text' selected, highlighted by a red box.
- Scope:** A dropdown menu with 'Global (Jenkins, nodes, items, all child items, etc)' selected.
- Secret:** A text input field containing masked characters (dots), highlighted by a red box.
- ID:** A text input field with 'mysonar' entered, highlighted by a red box.
- Description:** A text input field with 'sonar credentials' entered.

A blue 'Create' button is located at the bottom left of the form.

Step-20: Go to System > SonarQube Servers and add the SonarQube server.



The screenshot shows the Jenkins 'System' configuration page. The breadcrumb navigation at the top is 'Dashboard > Manage Jenkins > System >', with the last part highlighted by a red box. The page includes the following elements:

- A checkbox for 'Tool Locations' which is currently unchecked.
- A section titled 'SonarQube servers' with a note: 'If checked, job administrators will be able to inject a SonarQube server configuration as environment variables in the build.'
- An unchecked checkbox for 'Environment variables'.
- A section titled 'SonarQube installations' (highlighted by a red box) with the text 'List of SonarQube installations' below it.
- An 'Add SonarQube' button (highlighted by a red box) located below the 'SonarQube installations' section.

System [Jenkins] x System » Global credenti... x Security - My Account x devops0014/one x DevOps Online Training x +

← → ↻ Not secure 18.207.253.244:8080/manage/configure ☆ 📄 🗑️ 🔍 ⚙️

Dashboard > Manage Jenkins > System >

SonarQube installations

List of SonarQube installations

Name

Server URL

Default is http://localhost:9000

Server authentication token

SonarQube authentication token. Mandatory when anonymous access is disabled.

+ Add ▾

Advanced ▾

Save Apply

demo Config [Jenki... x MojoExecutionEx... x System » Global c... x Projects x devops0014/one x DevOps Online Tr... x +

← → ↻ Not secure 18.207.253.244:8080/job/demo/configure ☆ 📄 🗑️ 🔍 ⚙️

Dashboard > demo > Configuration

Pipeline script ▾

Configure

⚙️ General

🔗 Advanced Project Options

📄 Pipeline

Script ?

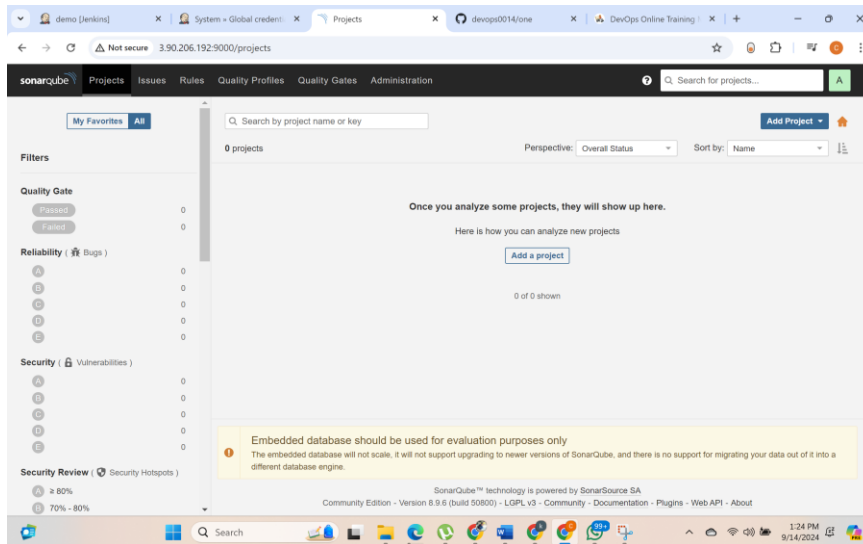
```
1 node
2 {
3     stage("Code")
4     {
5         git "https://github.com/devops0014/one.git"
6     }
7     stage("Build")
8     {
9         def mavenHome = tool name: "maven3", type: "maven"
10        def mavenCMD = "${mavenHome}/bin/mvn"
11        sh "${mavenCMD} clean package"
12    }
13    stage("CQA")
14    {
15        withSonarQubeEnv('mysonar')
16        {
17            def mavenHome = tool name: "maven3", type: "maven"
18            def mavenCMD = "${mavenHome}/bin/mvn"
19            sh "${mavenCMD} sonar:sonar"
20        }
21    }
22 }
```

☒ Use Groovy Sandbox ?

Save Apply

Step-21: Once you build the project, the results will be sent to SonarQube, where any bugs or issues will be displayed.

Before build



After Build

