

Sonar Qube

Code quality checking tool

SonarQube is an open-source tool/platform for inspection of code quality and security

Basically using this tool, we can identify the mistakes or issues in the code. As a DevOps engineer, we identify the bugs and we give them to Developer and they will fix them

--> It is developed by using Java language

--> SonarQube supports most of the programming languages to perform Code review (30+ programming languages definitely will support).

--> With the help of this platform, we can generate Code review report. That report can be given to Developers to fix the code.

Note: Code review is a part of Project Build Process

Sonar issues

--> SonarQube server will identify the following issues in the report

- > Bugs
- > Vulnerabilities (Security hotspots)
- > Duplicate code blocks (repeated code)
- > Code smells (Weak design in application)
- > Code coverage (How many lines of code is tested in Unit testing)

We can add these things in Sonar Quality Profiles

- > Set of rules to be considered during code review

1:07

Launch a new EC2 instance

It seems like you may be new to launching instances in EC2. Take a walkthrough to learn about EC2, how to launch instances and about best practices

Launch an instance [Info](#)

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags [Info](#)

Name

[Add additional tags](#)

▼ **Application and OS Images (Amazon Machine Image) [Info](#)**

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Description

Amazon Linux 2023 (kernel-6.1) is a modern, general purpose Linux-based OS that comes with 5 years of long term support. It is optimized for AWS and designed to provide a secure, stable and high-performance execution environment to develop and run your cloud applications.

Amazon Linux 2023 AMI 2023.7.20250623.1 x86_64 HVM kernel-6.1

Architecture

64-bit (x86)

Boot mode

uefi-preferred

AMI ID

ami-0f9cb75652314425a

Publish Date

2025-06-20

Username

ec2-user

Verified provider

▼ Instance type

Info | Get advice

Instance type

t2.medium

Family: t2 2 vCPU 4 GiB Memory Current generation: true

On-Demand Ubuntu Pro base pricing: 0.0547 USD per Hour On-Demand RHEL base pricing: 0.08 USD per Hour

On-Demand SUSE base pricing: 0.1512 USD per Hour On-Demand Windows base pricing: 0.0692 USD per Hour

On-Demand Linux base pricing: 0.0512 USD per Hour

All generations

Compare instance types

Additional costs apply for AMIs with pre-installed software

▼ Key pair (login)

Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - required

DevOpsMar30

Create new key pair

▼ Network settings

Info

Edit

Network

Info

vpc-0a752647f0a021f2e | default_VPC

Subnet

Info

No preference (Default subnet in any availability zone)

Auto-assign public IP

Info

Enable

Additional charges apply when outside of free tier allowance

Firewall (security groups)

Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group

Select existing security group

Common security groups

Info

Select security groups

DevOps-sg sg-031a081efd38c0e3a

VPC: vpc-0a752647f0a021f2e

Compare security group rules

Security groups that you add or remove here will be added to or removed from all your network interfaces.

Launch Instance

```
[ec2-user@ip-172-31-29-86 ~]$ sudo yum update -y
Amazon Linux 2023 Kernel Livepatch repository
kB 00:00
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-29-86 ~]$ sudo yum install docker -y
```

160 kB/s | 17

```
[ec2-user@ip-172-31-29-86 ~]$ sudo service docker start
[ec2-user@ip-172-31-29-86 ~]$ sudo service docker start
Redirecting to /bin/systemctl start docker service
[ec2-user@ip-172-31-29-86 ~]$ sudo usermod -aG docker ec2-user
```

```
[ec2-user@ip-172-31-29-86 ~]$ docker -v
Docker version 25.0.8, build 0bab007
```

```
[ec2-user@ip-172-31-29-86 ~]$ docker run -d --name sonarqube -p 9000:9000 -p 9092:9092 sonarqube:lts-community
```

```
[ec2-user@ip-172-31-29-86 ~]$ sudo docker run -d --name sonarqube -p 9000:9000 -p 9092:9092 sonarqube:lts-community
```

```
See 'docker run --help'.
[ec2-user@ip-172-31-29-86 ~]$ sudo docker run -d --name sonarqube -p 9000:9000 -p 9092:9092 sonarqube:lts-community
Unable to find image 'sonarqube:lts-community' locally
lts-community: Pulling from library/sonarqube
e735f3a6b701: Pull complete
25706e9e719c: Pull complete
5006e0852115: Pull complete
e412a6aa107b: Pull complete
a023b21e1d61: Pull complete
9ab0fd257737: Pull complete
036eda256d37: Pull complete
4f4fb700ef54: Pull complete
Digest: sha256:c4995e9521940286579290379b66010082d69508c8ca304f6c33bb74ad0a2ba5
Status: Downloaded newer image for sonarqube:lts-community
0fb55be786fa902f7a6fa3ca382c87ab417d40538a1309439cde3e41f11a30ef
[ec2-user@ip-172-31-29-86 ~]$
```

Rule ID	Rule Name	Type	Protocol	Port	Scope	Action
sgr-0e5cef937ec2e0b52		All TCP	TCP	0 - 65535	Custom	0.0.0/0 X Delete
sgr-07d1078f0fa10518a		HTTP	TCP	80	Custom	0.0.0/0 X Delete
		Custom TCP	TCP	9000	Anywhere...	0.0.0/0 X Delete

Add rule

Open: http://3.99.184.80:9000/sessions/new?return_to=%2F

SonarQube is running on port 9000

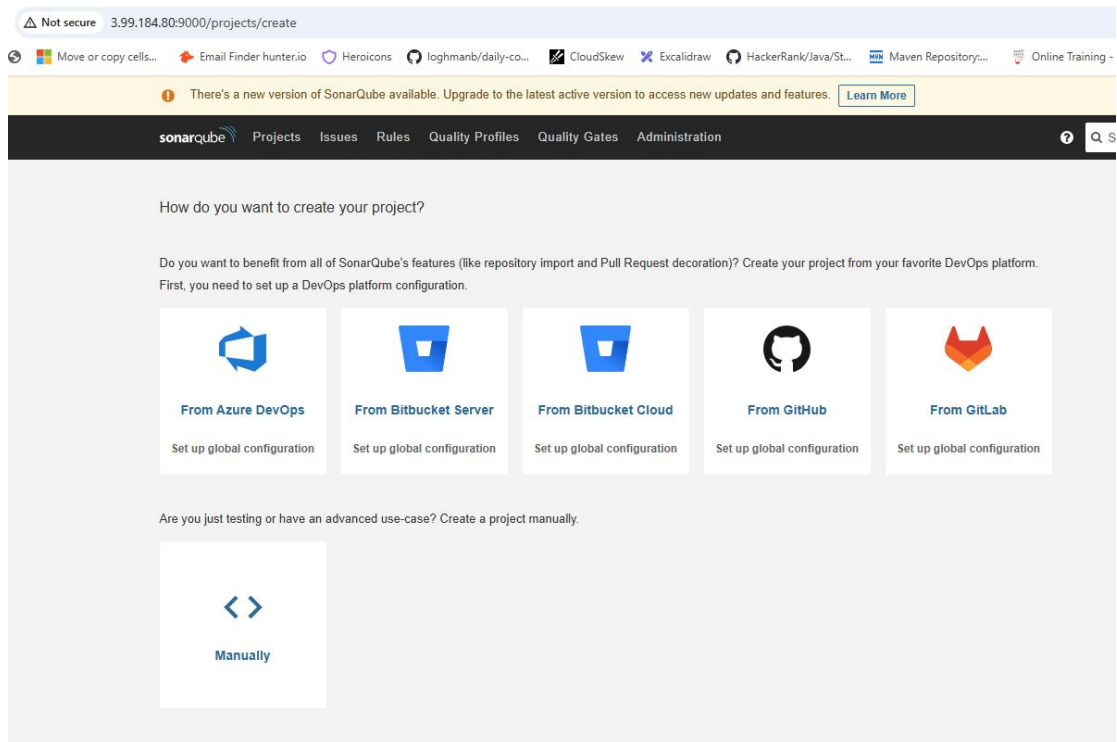
Log in to SonarQube

Login

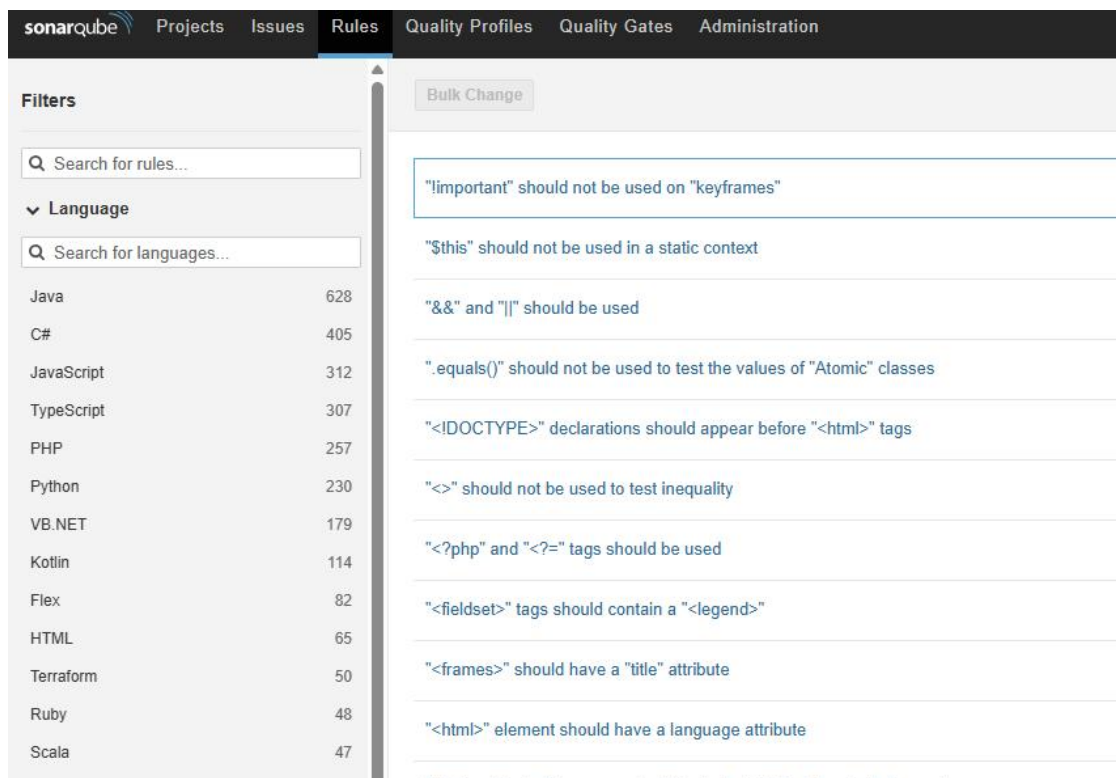
Password

Log in Cancel

Default username and password is admin, admin



<http://3.99.184.80:9000/projects/create>



sonarqube Projects Issues Rules **Quality Profiles** Quality Gates Administration

Quality Profiles

Quality profiles are collections of rules to apply during an analysis.
For each language there is a default profile. All projects not explicitly assigned to some other profile will be analyzed with the default.
Ideally, all projects will use the same profile for a language. [Learn More](#)

Filter profiles by:

C#, 1 profile(s)	Projects ?	Rules	Updated	Used
Sonar way BUILT-IN	DEFAULT	260	3 hours ago	Never

CSS, 1 profile(s)	Projects ?	Rules	Updated	Used
Sonar way BUILT-IN	DEFAULT	23	3 hours ago	Never

CloudFormation, 1 profile(s)	Projects ?	Rules	Updated	Used
Sonar way BUILT-IN	DEFAULT	26	3 hours ago	Never

If you get Quality Gate passed, it is safe to deploy the application

Quality Profiles **Quality Gates** Administration ?

Sonar way **BUILT-IN**

This quality gate complies with Clean as You Code

This quality gate complies with the [Clean as You Code](#) methodology, so that you benefit from the most efficient approach to delivering Clean Code. It ensures that:

- No new bugs are introduced
- No new vulnerabilities are introduced
- All new security hotspots are reviewed
- New code has limited technical debt
- New code has limited duplication
- New code is properly covered by tests

Conditions ?

Conditions on New Code

Metric	Operator	Value
--------	----------	-------

Go to Git

<https://github.com/Haider7214/SpringApp/blob/main/pom.xml>

We can replace the IP with my SonarQube public IP

```
</tag>
<url/>
</scm>
<properties>
  <java.version>17</java.version>
  <sonar.host.url>http://65.2.187.244:9000</sonar.host.url>
  <sonar.login>sqa_3cfe9d4e8765e7c2a4a6c80451500dae4ad262e</sonar.login>
</properties>
<dependencies>
```

```

PS C:\Users\saito\source\repos\DevOpsWithAWS_Course\SonarQube> git --version
git version 2.46.0.windows.1
PS C:\Users\saito\source\repos\DevOpsWithAWS_Course\SonarQube>
PS C:\Users\saito\source\repos\DevOpsWithAWS_Course\SonarQube>
PS C:\Users\saito\source\repos\DevOpsWithAWS_Course\SonarQube> git clone https://github.com/Haider7214/SpringApp.git
Cloning into 'SpringApp'...
remote: Enumerating objects: 47, done.
remote: Counting objects: 100% (47/47), done.
remote: Compressing objects: 100% (34/34), done.
remote: Total 47 (delta 8), reused 30 (delta 1), pack-reused 0 (from 0)
Receiving objects: 100% (47/47), 13.73 KiB | 226.00 KiB/s, done.
Resolving deltas: 100% (8/8), done.
PS C:\Users\saito\source\repos\DevOpsWithAWS_Course\SonarQube> ls -l

```

Go to SonarQube “My Account” --> Security

A Administrator

[Profile](#)
[Security](#)
[Notifications](#)
[Projects](#)

Tokens

If you want to enforce security by not providing credentials of a real SonarQube user to run your code scan or to invoke web services, you can provide a User Token as a replacement of the user login. This will increase the security of your installation by not letting your analysis user's password going through your network.

Generate Tokens

Name

Type

Expires in

Select Token Type

30 days

Generate

Name	Type	Project	Last use	Created	Expiration
No tokens					

Enter a new password

All fields marked with * are required

Old Password *

New Password *

Confirm Password *

Update

Global Analysis Token

Click Generate

A Administrator

[Profile](#)
[Security](#)
[Notifications](#)
[Projects](#)

Tokens

If you want to enforce security by not providing credentials of a real SonarQube user to run your code scan or to invoke web services, you can provide a User Token as a replacement of the user login. This will increase the security of your installation by not letting your analysis user's password going through your network.

Generate Tokens

Name

Type

Expires in

Global Analysis Token

30 days

Generate

Name	Type	Project	Last use	Created	Expiration
No tokens					

Token: sqq_89db2d71077985fb78fa1d176ba4abae93a98dd2

Open Pom file and enter Token and Public IP

```

        <tag/>
        <url/>
    </scm>
    <properties>
        <java.version>17</java.version>
        <sonar.host.url>http://3.99.184.80/:9000/</sonar.host.url>
        <sonar.login>sqa_89db2d71077985fb78fa1d176ba4abae93a98dd2</sonar.login>
    </properties>
    <dependencies>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>

```

Open project in Eclipse --> right click --> Run As --> Maven Build --> Goals: sonar:sonar
 Edit configuration and launch.

Name: SpringApp

Main JRE Refresh Source Environment Common

Base directory: \${project_loc:FirstSpringWebApp}

Workspace... File System... Variables...

Goals: sonar:sonar

Profiles:

User settings:

Workspace... File System... Variables...

☐ Offline ☐ Update Snapshots

☐ Debug Output ☐ Skip Tests ☐ Non-recursive

☐ Resolve Workspace artifacts

Threads: 1 Color Output: Auto

Parameter Name	Value

Add... Edit... Remove

Maven Runtime: EMBEDDED (3.9.7/3.9.700.20240602-2313) Configure...

Revert Apply

Run Close


```

[INFO] 16:25:22.596 ANALYSIS SUCCESSFUL, you can find the r
[INFO] 16:25:22.596 Note that you will be able to access th
[INFO] 16:25:22.596 More about the report processing at htt
[INFO] 16:25:22.626 Analysis total time: 20.418 s
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 35.600 s
[INFO] Finished at: 2025-07-06T16:25:22-04:00
[INFO] -----

```

Go back to SonarQube --> Projects

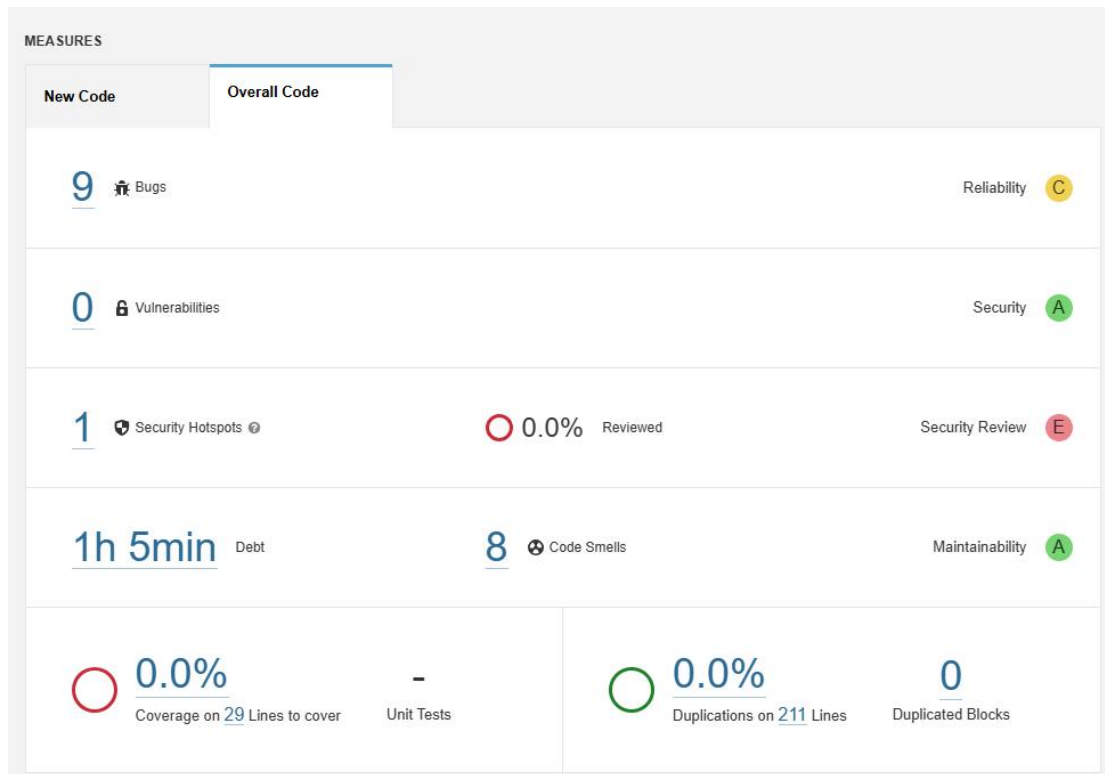
The screenshot shows the SonarQube 'Projects' page. On the left, there are filters for 'Quality Gate' (Passed: 1, Failed: 0) and 'Reliability' (A rating: 0, B rating: 0, C rating: 1, D rating: 0, E rating: 0). The main area shows 1 project(s) with a search bar and filters. The project 'FirstSpringWebApp' is listed with a 'Passed' status. Below the project name, there are metrics: Bugs (9, C), Vulnerabilities (0, A), Hotspots Reviewed (0.0%, E), Code Smells (8, A), Coverage (0.0%), Duplications (0.0%), and Lines (211, XS). The last analysis was 2 minutes ago.

FirstSpringWebApp project is Passed

The screenshot shows the detailed view of the 'FirstSpringWebApp' project. The 'Overview' tab is selected. At the top, there is a banner indicating that the project is 'Passed'. Below this, the 'QUALITY GATE STATUS' section shows 'Passed' with the message 'All conditions passed.' The 'MEASURES' section shows two tabs: 'New Code' and 'Overall Code'. Under 'New Code', there are 9 Bugs and 0 Vulnerabilities. Under 'Overall Code', there are 9 Bugs and 0 Vulnerabilities.

9 bugs are there

It will take Developers 1h 5min to resolve the issues



For Quality Profile

sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration

Quality Profiles

Quality profiles are collections of rules to apply during an analysis.
For each language there is a default profile. All projects not explicitly assigned to some other profile will be analyzed with the default.
Ideally, all projects will use the same profile.

Filter profiles by: Select...

C#, 1 profile(s)
Sonar way BUILT-IN


CSS, 1 profile(s)
Sonar way BUILT-IN

CloudFormation, 1 profile(s)
Sonar way BUILT-IN

Docker, 1 profile(s)
Sonar way BUILT-IN


New Quality Profile

What type of profile do you want to create?




Extend an existing quality profile

Create a child quality profile inheriting its parent's active rules.
Changes to the parent profile will impact the child profile.



Copy an existing quality profile

Create a new quality profile as a replica of the copied quality profile.
The two profiles will then evolve independently.



Create a blank quality profile

Create a new quality profile with no active rules by default.

All fields marked with * are required

Language *
Select...

Profile to extend *
Select...

Name *

Click on Activate More

Quality Profiles / Java

Java_QP

Updated: 20 se

Rules	Active	Inactive
Total	0	628
Bugs	0	150
Vulnerabilities	0	33
Code Smells	0	408
Security Hotspots	0	37

[Activate More](#)

Sonar way rules not included 479

Inheritance

Java_QP 0 active rules 0 overridden

Projects

You must activate at least 1 rule before you can assign projects to this profile.

Permissions

[Bulk Change](#) 1 / 628 rules

"equals()" should not be used to test the values of "Atomic" classes	Java Bug multi-threading	Activate
"==" should not be used instead of "!="	Java Bug	Activate
"==" and "!=" should not be used when "equals" is overridden	Java Code Smell cert, cwe, suspicious	Activate
"@CheckForNull" or "@Nullable" should not be used on primitive types	Java Code Smell	Activate
"@Controller" classes that use "@SessionAttributes" must call "setComplete" on their "SessionStatus" objects	Java Bug spring	Activate
"@Deprecated" code marked for removal should never be used	Java Code Smell cert, cwe, obsolete	Activate

1 / 628 rules

"equals()" should not be used to test the values of "Atomic" classes	Java Bug multi-threading	Deactivate
"==" should not be used instead of "!="	Java Bug	Deactivate
"==" and "!=" should not be used when "equals" is overridden	Java Code Smell cert, cwe, suspicious	Deactivate
"@CheckForNull" or "@Nullable" should not be used on primitive types	Java Code Smell	Activate
"@Controller" classes that use "@SessionAttributes" must call "setComplete" on their "SessionStatus" objects	Java Bug spring	Activate

Java, 2 profile(s)	Projects	Rules	Updated	Used
Java_QP	0	3	53 minutes ago	Never
Sonar way BUILT-IN	DEFAULT	479	6 hours ago	1 hour ago

2 Java profiles are there, default has 479 rules

Set my rules as default

Java, 2 profile(s)	Projects ⓘ	Rules	Updated	Used
Java_QP	0	3	53 minutes ago	Never
Sonar way BUILT-IN	DEFAULT	479	6 hours ago	1 h
<div> Activate More Rules Back up Compare Extend Copy Rename Set as Default Delete </div>				
JavaScript, 1 profile(s)	Projects ⓘ	Rules	Updated	
Sonar way BUILT-IN	DEFAULT	219	6 hours ago	
Kotlin, 1 profile(s)	Projects ⓘ	Rules	Updated	Used

Maven Build once again in Eclipse

Name: SpringApp (2)

Main JRE Refresh Source Environment Common

Base directory:
\${project_loc:FirstSpringWebApp}

Workspace... File System... Variables...

Goals: sonar:sonar

Profiles:

User settings:

Workspace... File System... Variables...

☐ Offline ☐ Update Snapshots

☐ Debug Output ☐ Skip Tests ☐ Non-recursive

☐ Resolve Workspace artifacts

Threads: 1 Color Output: Auto

Parameter Name	Value

Add... Edit... Remove

Now we can see 0 bugs based on my 8 rules

sonarqube

ProjectsIssuesRulesQuality ProfilesQuality GatesAdministration

?

Search for projects

FirstSpringWebApp

main

Last analysis of this Branch had 1 warning

July 6, 2025 at 5:46 PM

Ver

OverviewIssuesSecurity HotspotsMeasuresCodeActivity

Project Settings

To benefit from more of SonarQube's features, set up analysis in your favorite CI.

QUALITY GATE STATUS

Passed

All conditions passed.

MEASURES

New Code

Since July 6, 2025

Started 1 hour ago

Overall Code

0

New Bugs

0

New Vulnerabilities

0

New Security Hotspots

Reviewed

St

0

Added Debt




0

New Code Smells

Now 0 bugs


Project Information



Project Settings ▾  Project Information

	<h3>Project Information</h3>
	Description Demo project for Spring Boot  No tags ▾
Reliability	Lines of Code (Main branch) 211 
Security	Quality Gate used (Default) Sonar way Quality Profiles used (Java) Java_QP (JSP) Sonar way (XML) Sonar way
Security Review	Project Key <div>com.telusko:FirstSpringWebA  Copy</div>
Maintainability	Get project badges Set notifications

tions on **0** New Lines

211 lines are there and Java_QP is being used

sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration 

Quality Gates   Sonar way BUILT-IN

Sonar way DEFAULT BUILT-IN



This quality gate ensures that the code is clean and free of errors.


- No new bugs
- No new vulnerabilities
- All new security issues are resolved
- New code is covered by tests
- New code is free of errors

Create Quality Gate

All fields marked with * are required

Name *

 Save  Cancel

Conditions 

Conditions on New Code

Metric	Operator	Value
Coverage	is less than	80.0%
Duplicated Lines (%)	is greater than	3.0%

demo_quality_gate Rename Copy Set as Default Delete

This quality gate complies with Clean as You Code

This quality gate complies with the [Clean as You Code](#) methodology, so that you benefit from the most efficient approach to delivering Clean Code. It ensures that:

- No new bugs are introduced
- No new vulnerabilities are introduced
- All new security hotspots are reviewed
- New code has limited technical debt
- New code has limited duplication
- New code is properly covered by tests

Conditions ⓘ

Conditions on New Code

Metric	Operator	Value	
Coverage	is less than	80.0%	
Duplicated Lines (%)	is greater than	3.0%	
Maintainability Rating	is worse than	A (Technical debt ratio is less than 5.0%)	
Reliability Rating	is worse than	A (No bugs)	
Security Hotspots Reviewed	is less than	100%	

We can change the Conditions as well

SonarQube setup

1 --> Setup a Linux VM (Amazon Linux AMI --> t2.medium) min 2 GB RAM is required
Connect to that VM once created

2 --> Install Docker in Linux VM
`sudo yum update -y`
`sudo yum install docker -y`
`sudo service docker start`
`sudo usermod -aG docker ec2-user`
`exit`

Reconnect back to Linux VM

3 --> Verify the Docker installation

`docker -v`

4 --> Run SonarQube using Docker image of SonarQube

`docker run -d --name sonarqube -p 9000:9000 -p 9092:9092 sonarqube/lts-community`
 Enable 9000 port number in Linux VM security group

5 --> Access SonarQube public IP of Linux VM:port number (9000)

After accessing default username and password : admin and admin

Change it to your own

Integrate Sonar Server with Java Maven project

1 --> Generated Sonar token for integration

Sonar Dashboard --> My Account --> Security --> Generate Token

2 --> Install Docker in Linux VM <https://github.com/Haider7214/SpringApp/tree/main>

pom.xml under properties

```
<properties>
    <java.version>17</java.version>
    <sonar.host.url>http://3.99.184.80:9000</sonar.host.url>
    <sonar.login>sqa_89db2d71077985fb78fa1d176ba4abae93a98dd2</sonar.login>
</properties>
```


3 --> mvn clean package --> Open Git bash from a folder where Project is cloned
--> mvn sonar:sonar --> Goal will build project

Sonar Quality Profiles

--> Set of rules to perform code review
--> For every programming language one Quality profile with set of rules to perform Code review would be available (default/built in)
--> Java project --> It has its own Java quality profile --> Java set of rules to perform code review
--> Whenever we perform code review using Sonarqube it will identify our project developed using which language based on that it will execute the language specific quality profile and perform code review
--> We can create our own Quality Profile based on project requirement
- Name: Demo_App_QP
- Language: Java
- Parent: None

We can make our quality profile as default profile then it would be applicable for all projects which get reviews under this Sonar server

Sonar Quality Gate

--> Quality Gate represents the overall project code quality whether it is passed or failed, which will decide if project can be deployed or not

Note: If code quality is failed, we should not deploy that particular code or application

In Sonar, we will have default Quality Gate and every project must be passed in order to deploy
If required, we can create our own quality gate as per need or requirement

Summary Note:

--> If project quality gate is failed then we must not accept that code for deployment as a DevOps engineer
--> if we found out any Sonar issues then Development team is responsible to fix all issues and as a DevOps engineer we will perform code review and send code review report to development team

SonarQube --> Sonar setup on Linux VM --> Sonar integration in Maven project --> Sonar token generation --> Server issue types, Quality profiles --> Quality gates