1. Introduction

Case Study Overview:

Case Study Overview: This project aims to implement a Continuous Integration (CI) pipeline using Jenkins, SonarQube, and AWS Cloud9 IDE. The focus is on setting up an automated static code analysis process for Java/Python applications, ensuring code quality and best practices are maintained throughout the development lifecycle.

Key Feature and Application: The unique feature of this case study is the integration of SonarQube for automated code quality checks within Jenkins. This setup streamlines the code review process, helps in identifying bugs early, and enforces coding standards.

Practical Application:
- **Jenkins:** Automates the CI pipeline, triggering builds and static code analysis whenever code changes occur.
- **SonarQube:** Performs static analysis to provide feedback on the maintainability, reliability, and security of the code.
- **AWS Cloud9:** Serves as the development environment, offering a cloud-based IDE that integrates smoothly with AWS services.

Third-Year Project Integration (EventEase Project):

The MapMyCampus project, which I developed during my third year, can benefit from the integration of this CI/CD pipeline approach to optimize the development process..

Code Quality Management:

- Integrating **SonarQube** into the CI pipeline enables continuous static code analysis, ensuring the project's codebase remains clean, secure, and adheres to best practices

Automating Deployments:

- Jenkins can automate the build, test, and deployment process for **MapMyCampus** each time code is pushed to the repository.

Team Collaboration:

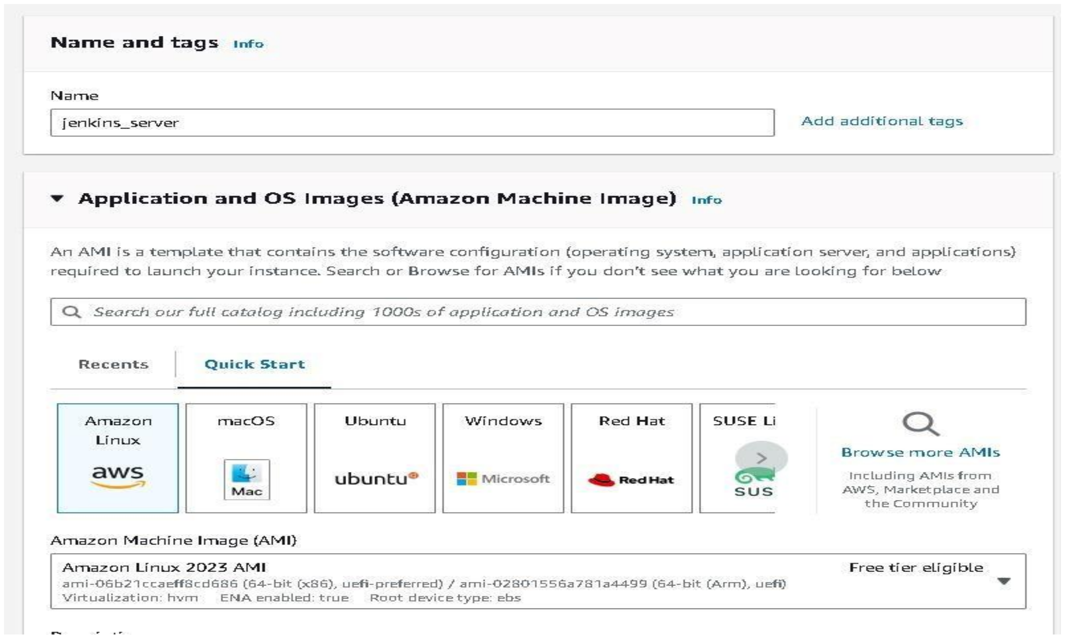
- The CI/CD setup allows your **team of four** to work efficiently with consistent feedback from SonarQube, leading to better collaboration and fewer integration issues.

- ## 2. Demonstration

Problem Statement

1] Jenkins on an EC2 instance:Resource:https://www.jenkins.io/doc/tutorials/tutorial-for-installing-jenkins-on-AWS/

1) Launch a AWS EC2 instance with a Linux OS .



Name and tags Info

Name: jenkins_server

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

Search our full catalog including 1000s of application and OS images

Recents | Quick Start

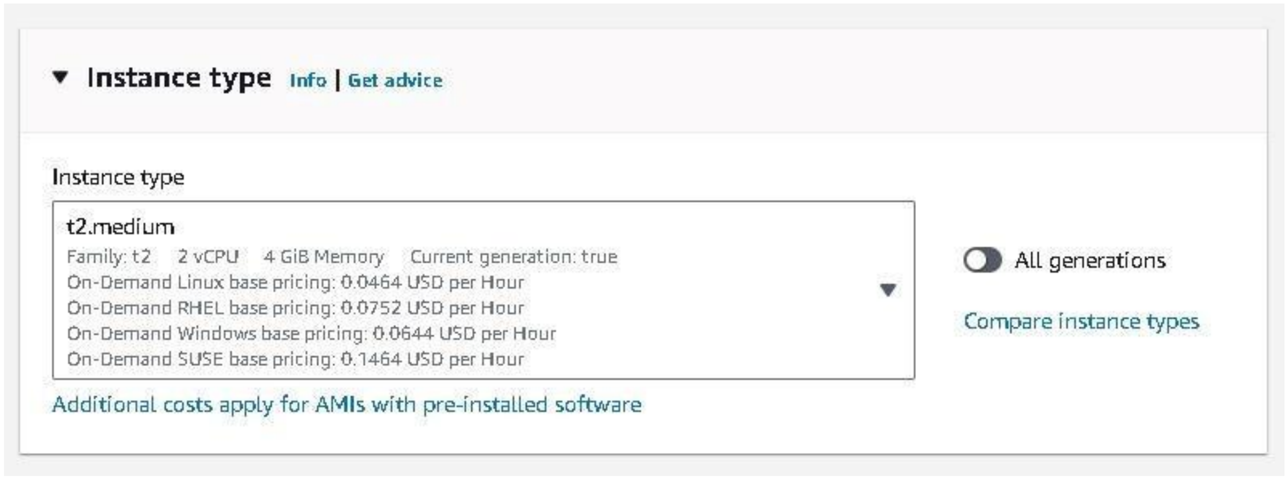
Amazon Linux | macOS | Ubuntu | Windows | Red Hat | SUSE Li

Browse more AMIs — Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Amazon Linux 2023 AMI — Free tier eligible
ami-06b21ccaeff8cd686 (64-bit (x86), uefi-preferred) / ami-02801556a781a4499 (64-bit (Arm), uefi)
Virtualization: hvm ENA enabled: true Root device type: ebs

2) instance type as t2.medium.



Instance type Info | Get advice

Instance type

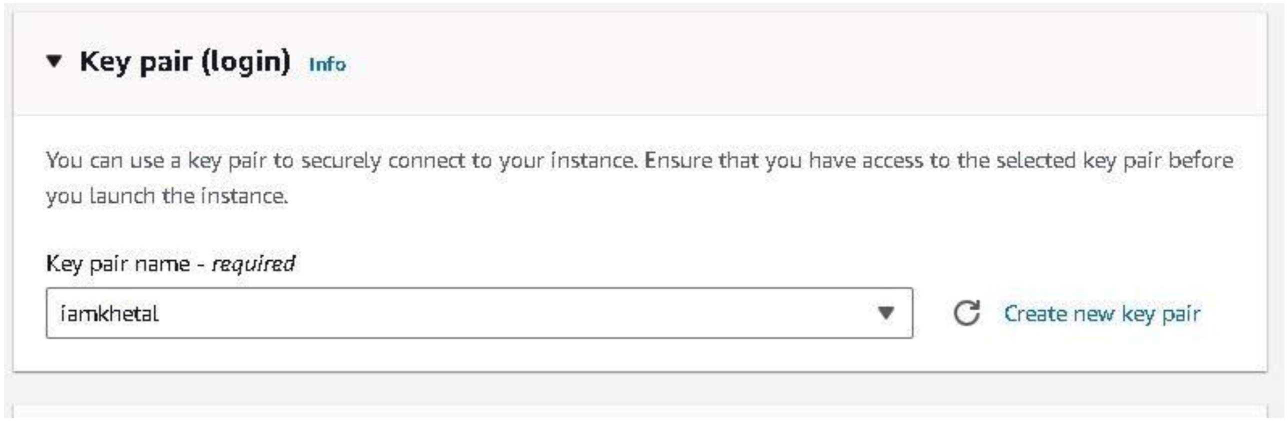
t2.medium
Family: t2 2 vCPU 4 GiB Memory Current generation: true
On-Demand Linux base pricing: 0.0464 USD per Hour
On-Demand RHEL base pricing: 0.0752 USD per Hour
On-Demand Windows base pricing: 0.0644 USD per Hour
On-Demand SUSE base pricing: 0.1464 USD per Hour

All generations

Compare instance types

Additional costs apply for AMIs with pre-installed software

3) Create a key pair for our instance



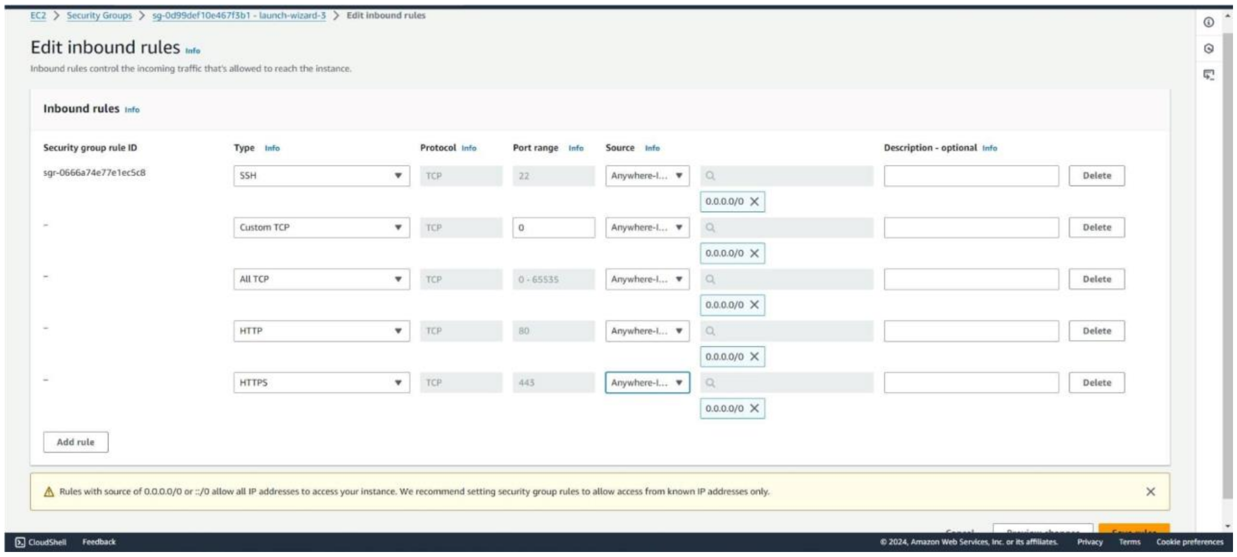
Key pair (login) Info

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*

iamkhetal

4) Allow the TCP, HTTP and HTTPS network access for all connections over the network.



EC2 > Security Groups > sg-0d99def10e467f3b1 - launch-wizard-3 > Edit inbound rules

Edit inbound rules

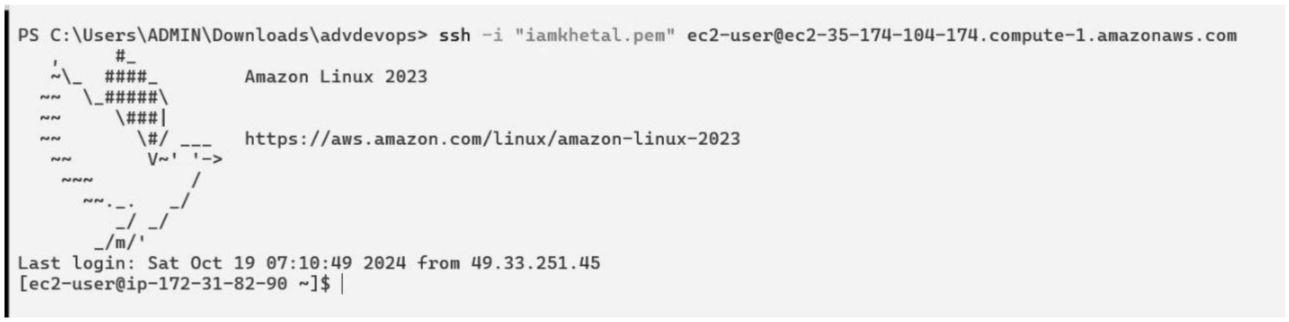
Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules

Security group rule ID	Type	Protocol	Port range	Source		Description - optional
sgr-0666a74e77e1ec5c8	SSH	TCP	22	Anywhere-I...	0.0.0.0/0	
-	Custom TCP	TCP	0	Anywhere-I...	0.0.0.0/0	
-	All TCP	TCP	0 - 65535	Anywhere-I...	0.0.0.0/0	
-	HTTP	TCP	80	Anywhere-I...	0.0.0.0/0	
-	HTTPS	TCP	443	Anywhere-I...	0.0.0.0/0	

Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

5) SSH your connection

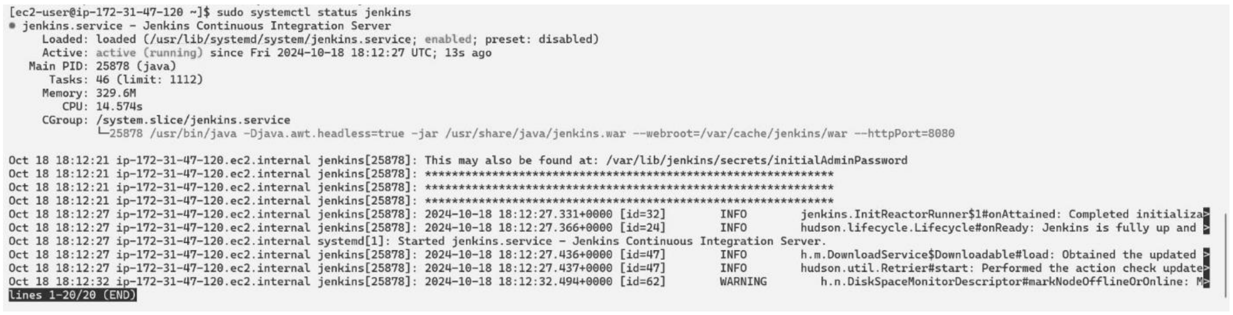


```
PS C:\Users\ADMIN\Downloads\advdevops> ssh -i "iamkhetal.pem" ec2-user@ec2-35-174-104-174.compute-1.amazonaws.com
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Sat Oct 19 07:10:49 2024 from 49.33.251.45
[ec2-user@ip-172-31-82-90 ~]$ |
```

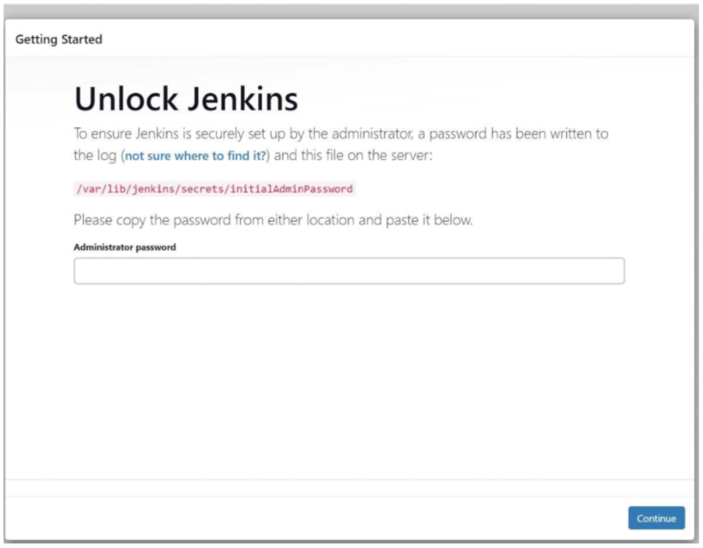
6) Execute the following commands :-

- **sudo yum update -y**: Updates all installed packages on the system to the latest available versions.

- **sudo wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-stable/jenkins.repo**: Downloads the Jenkins repository file and saves it in the system's repository folder.
- **sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key**: Imports the Jenkins GPG key to authenticate the Jenkins packages.
- **sudo yum upgrade**: Upgrades the system packages to their latest versions based on available repositories.
- **sudo yum install jenkins -y:** Installs Jenkins from the configured repository.
- **sudo systemctl enable jenkins**: Enables Jenkins to automatically start at system boot.
- **sudo systemctl start jenkins**: Starts the Jenkins service immediately.
- **sudo systemctl status jenkins**: Displays the current status of the Jenkins service.



```
[ec2-user@ip-172-31-47-120 ~]$ sudo systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
     Loaded: loaded (/usr/lib/systemd/system/jenkins.service; enabled; preset: disabled)
     Active: active (running) since Fri 2024-10-18 18:12:27 UTC; 13s ago
   Main PID: 25878 (java)
      Tasks: 46 (limit: 1112)
     Memory: 329.6M
        CPU: 14.574s
     CGroup: /system.slice/jenkins.service
             └─25878 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins/war --httpPort=8080

Oct 18 18:12:21 ip-172-31-47-120.ec2.internal jenkins[25878]: This may also be found at: /var/lib/jenkins/secrets/initialAdminPassword
Oct 18 18:12:21 ip-172-31-47-120.ec2.internal jenkins[25878]: *************************************************************
Oct 18 18:12:21 ip-172-31-47-120.ec2.internal jenkins[25878]: *************************************************************
Oct 18 18:12:21 ip-172-31-47-120.ec2.internal jenkins[25878]: *************************************************************
Oct 18 18:12:27 ip-172-31-47-120.ec2.internal jenkins[25878]: 2024-10-18 18:12:27.331+0000 [id=32]       INFO       jenkins.InitReactorRunner$1#onAttained: Completed initializa>
Oct 18 18:12:27 ip-172-31-47-120.ec2.internal jenkins[25878]: 2024-10-18 18:12:27.366+0000 [id=24]       INFO       hudson.lifecycle.Lifecycle#onReady: Jenkins is fully up and >
Oct 18 18:12:27 ip-172-31-47-120.ec2.internal systemd[1]: Started jenkins.service - Jenkins Continuous Integration Server.
Oct 18 18:12:27 ip-172-31-47-120.ec2.internal jenkins[25878]: 2024-10-18 18:12:27.436+0000 [id=47]       INFO       h.m.DownloadService$Downloadable#load: Obtained the updated >
Oct 18 18:12:27 ip-172-31-47-120.ec2.internal jenkins[25878]: 2024-10-18 18:12:27.437+0000 [id=47]       INFO       hudson.util.Retrier#start: Performed the action check update>
Oct 18 18:12:32 ip-172-31-47-120.ec2.internal jenkins[25878]: 2024-10-18 18:12:32.494+0000 [id=62]       WARNING       h.n.DiskSpaceMonitorDescriptor#markNodeOfflineOrOnline: M>
lines 1-20/20 (END)
```



Getting Started

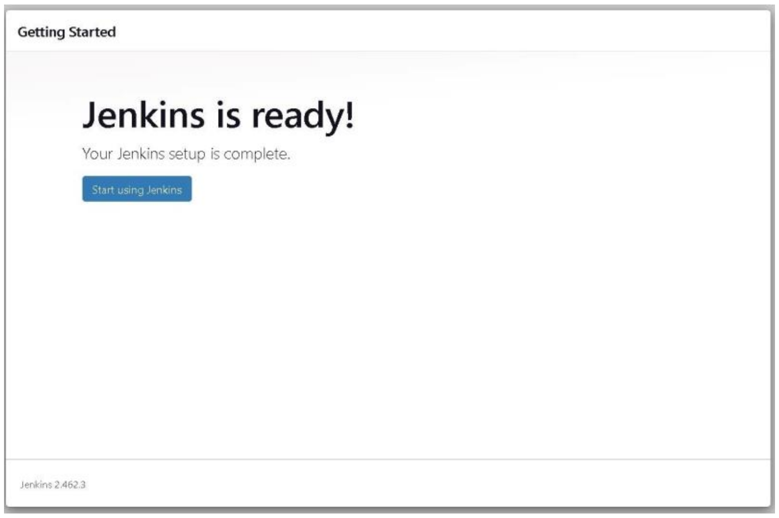
Unlock Jenkins

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

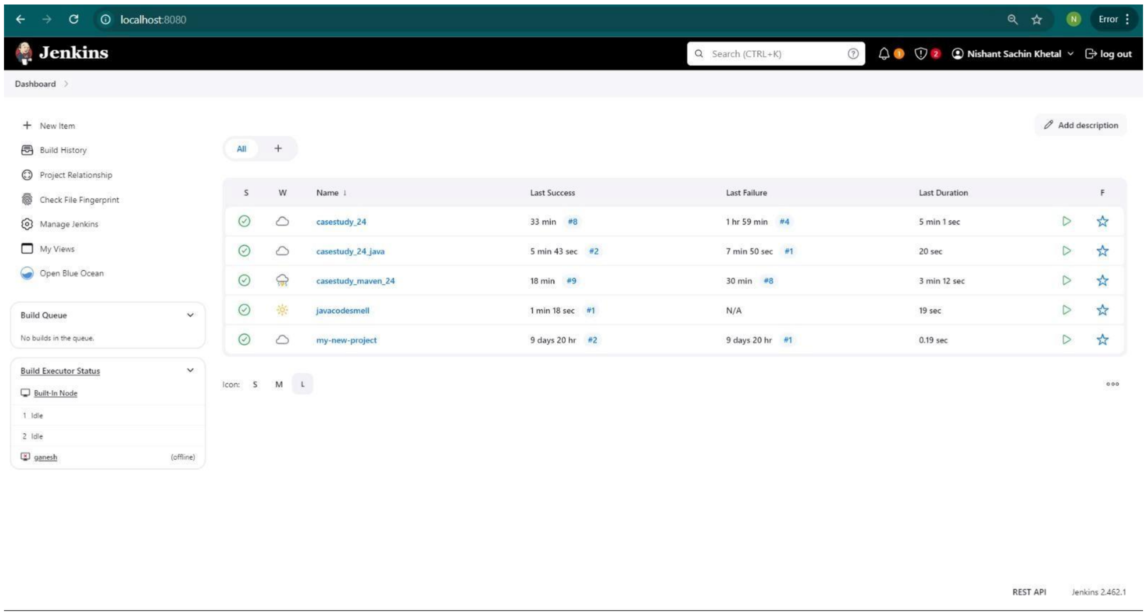


Getting Started

Jenkins is ready!

Your Jenkins setup is complete.

Jenkins 2.462.3



Build Queue

No builds in the queue.

Build Executor Status

Built-In Node

1 Idle

2 Idle

ganesh (offline)

S	W	Name ↓	Last Success	Last Failure	Last Duration		F
✓	☁	casestudy_24	33 min #8	1 hr 59 min #4	5 min 1 sec	▷	☆
✓	☁	casestudy_24_java	5 min 43 sec #2	7 min 50 sec #1	20 sec	▷	☆
✓	⛈	casestudy_maven_24	18 min #9	30 min #8	3 min 12 sec	▷	☆
✓	☀	javacodesmell	1 min 18 sec #1	N/A	19 sec	▷	☆
✓	☁	my-new-project	9 days 20 hr #2	9 days 20 hr #1	0.19 sec	▷	☆

Jenkins 2.462.1

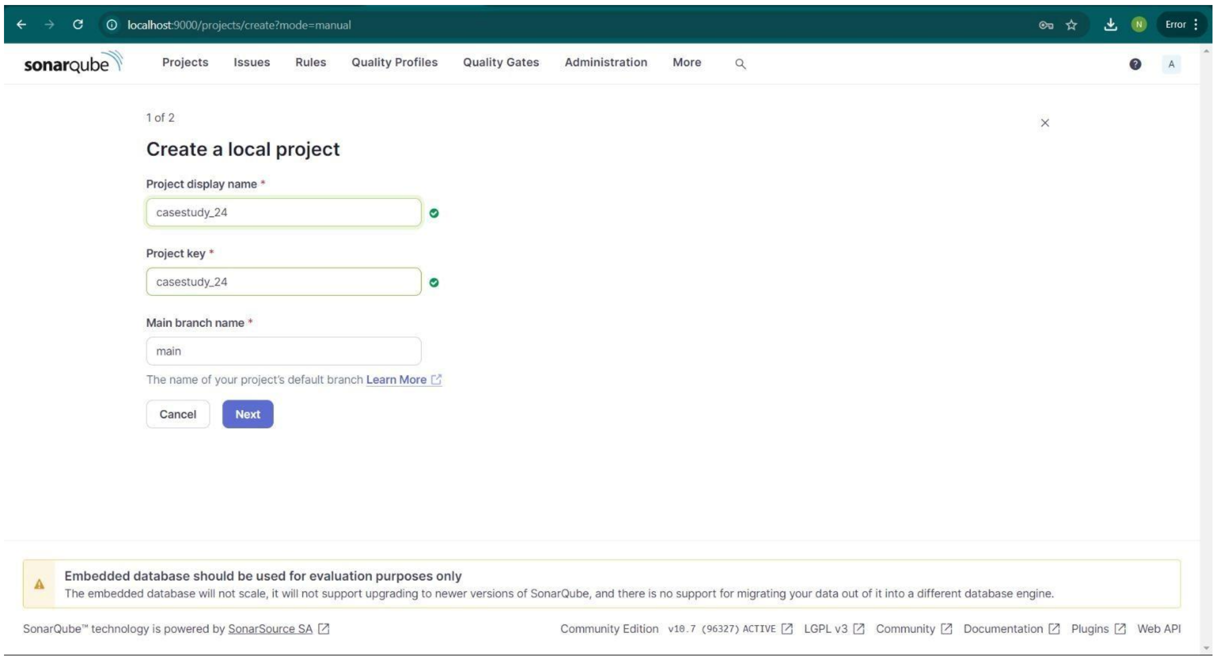
Thus, Jenkins is successfully configured on the EC2 Linux instance.

Task: SonarQube analysis of a Java/Python Project on Jenkins Pipeline:-

A] Sonarqube project:-

Python Project : https://github.com/piomin/sample-java-sonar

1) Create a sonarqube project named **casestudy_64.**


localhost:9000/projects/create?mode=manual

1 of 2

Create a local project

Project display name *: casestudy_24

Project key *: casestudy_24

Main branch name *: main

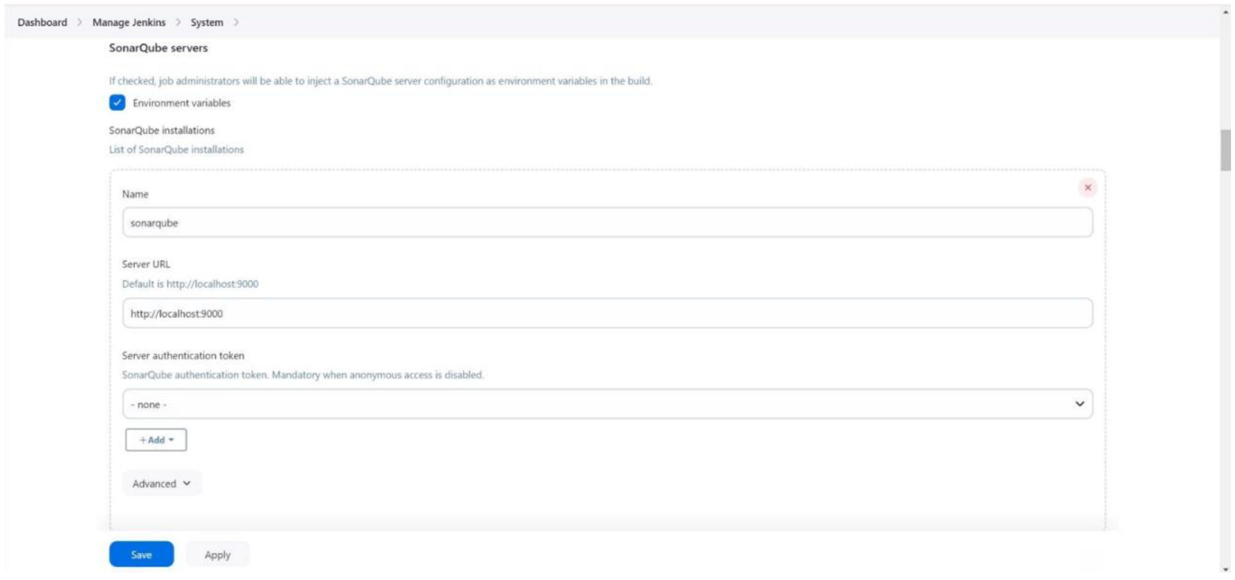
The name of your project's default branch Learn More

Embedded database should be used for evaluation purposes only
The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine.

SonarQube™ technology is powered by SonarSource SA

Community Edition v10.7 (96327) ACTIVE LGPL v3 Community Documentation Plugins Web API

2) Sonarqube configurations in jenkins:-


Dashboard > Manage Jenkins > System >

SonarQube servers

If checked, job administrators will be able to inject a SonarQube server configuration as environment variables in the build.

✓ Environment variables

SonarQube installations

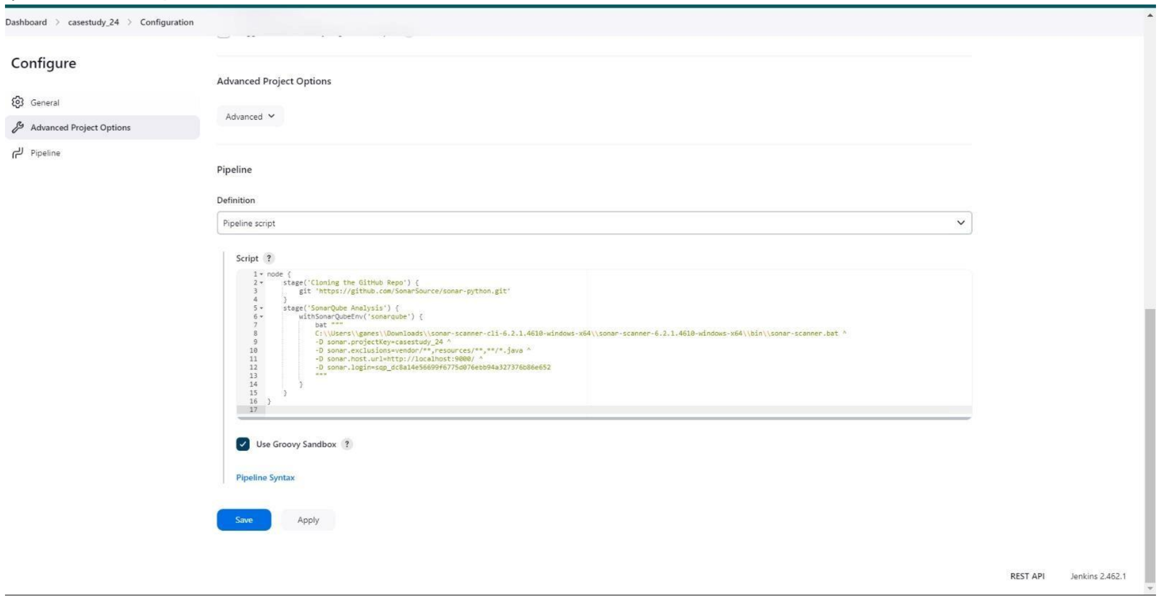
List of SonarQube installations

Name: sonarqube

Server URL
Default is http://localhost:9000
http://localhost:9000

Server authentication token
SonarQube authentication token. Mandatory when anonymous access is disabled.
- none -

3) Jenkins Pipeline:-



Pipeline

Definition: Pipeline script

Script

```
node {
    stage('Cloning the GitHub Repo') {
        git 'https://github.com/SonarSource/sonar-python.git'
    }
    stage('SonarQube Analysis') {
        withSonarQubeEnv('sonarqube') {
            bat """
            C:\\Users\\ganes\\Downloads\\sonar-scanner-cli-6.2.1.4610-windows-x64\\sonar-scanner-6.2.1.4610-windows-x64\\bin\\sonar-scanner.bat ^
            -D sonar.projectKey=casestudy_24 ^
            -D sonar.exclusions=vendor/**,resources/**,**/*.java ^
            -D sonar.host.url=http://localhost:9000/ ^
            -D sonar.login=sqp_dc8a14e56699f6775d076ebb94a327376b86e652
            """
        }
    }
}
```

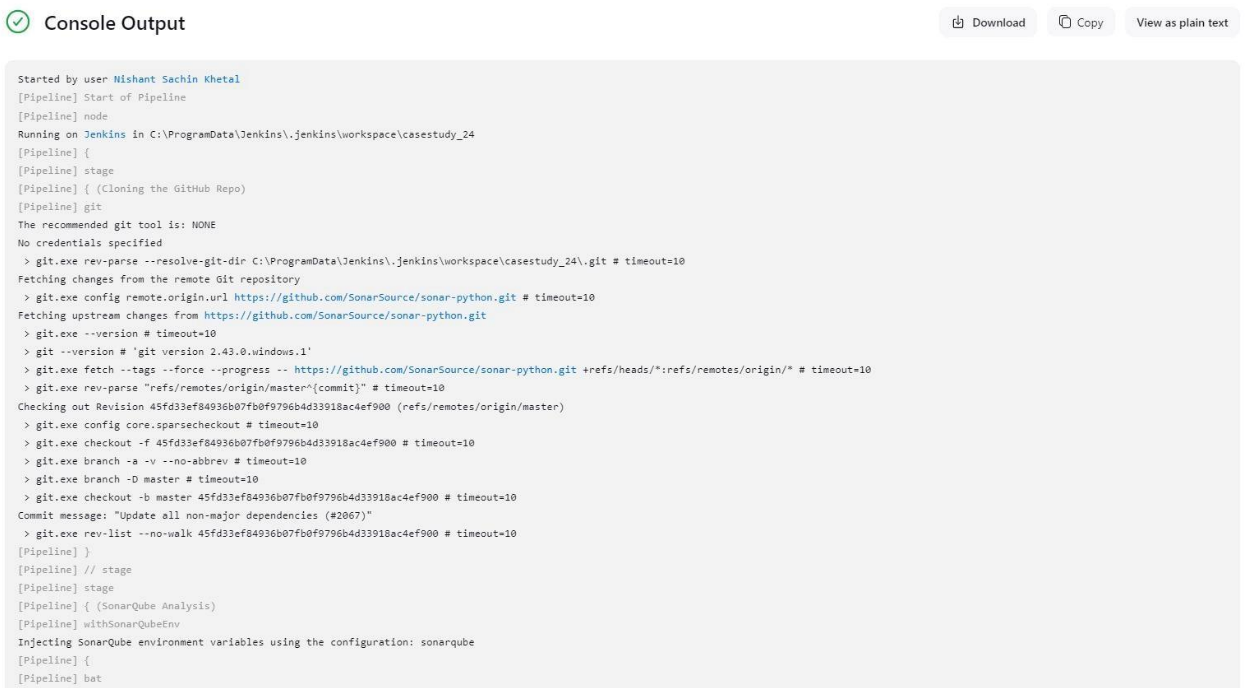
Use Groovy Sandbox ✓

Pipeline Syntax

Pipeline Script:node

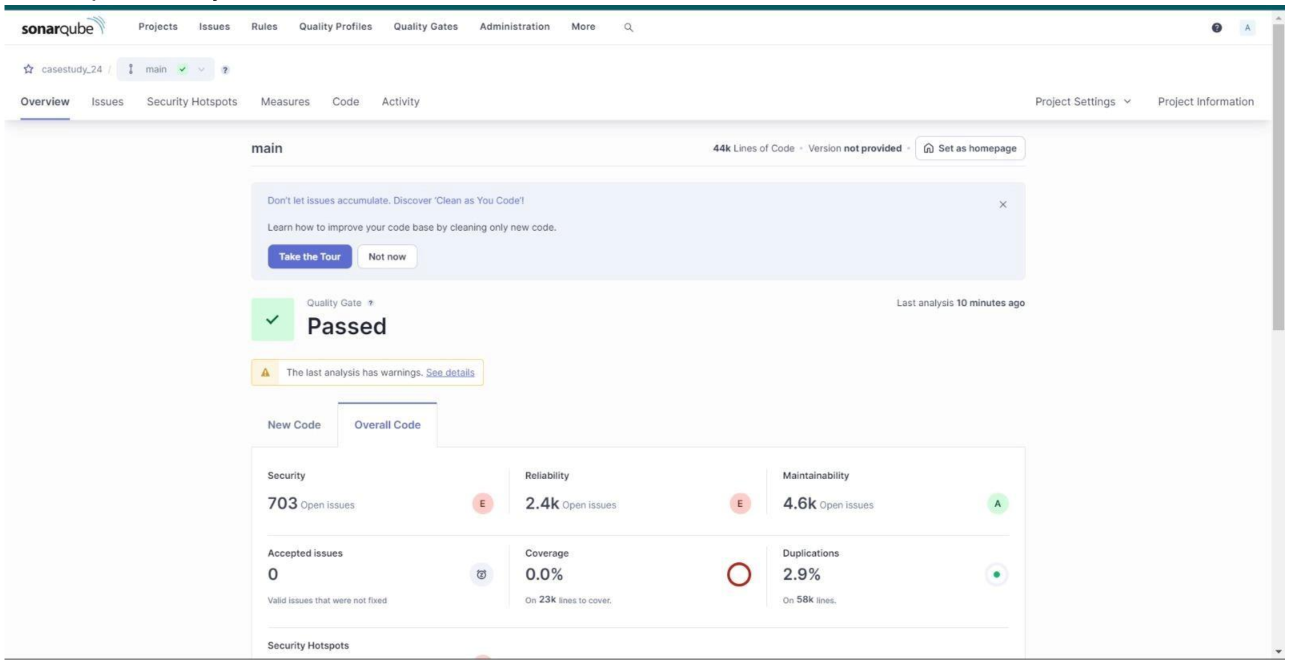
```
{
   stage('Cloning the GitHub Repo') { git
      'https://github.com/SonarSource/sonar-python.git'
   }
   stage('SonarQube Analysis') {
      withSonarQubeEnv('sonarqube') {
         bat
"C:\\Users\\nish\\Downloads\\sonar-scanner-cli-6.1.0.4477-windows-x64\\sonar-scanner-6.1.0.4
477-windows-x64\\bin\\sonar-scanner.bat " +
            "-D sonar.login=admin " +
            "-D sonar.password=HareKrishna#108 " +
            "-D sonar.projectKey=casestudy_24 " +
            "-D sonar.exclusions=vendor/**,resources/**,**/*.java " +
            "-D sonar.host.url=http://localhost:9000/"
      }
   }
}
```

4) Jenkins Output :



```
Started by user Nishant Sachin Khetal
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in C:\ProgramData\Jenkins\.jenkins\workspace\casestudy_24
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Cloning the GitHub Repo)
[Pipeline] git
The recommended git tool is: NONE
No credentials specified
 > git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\.jenkins\workspace\casestudy_24\.git # timeout=10
Fetching changes from the remote Git repository
 > git.exe config remote.origin.url https://github.com/SonarSource/sonar-python.git # timeout=10
Fetching upstream changes from https://github.com/SonarSource/sonar-python.git
 > git.exe --version # timeout=10
 > git --version # 'git version 2.43.0.windows.1'
 > git.exe fetch --tags --force --progress -- https://github.com/SonarSource/sonar-python.git +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
Checking out Revision 45fd33ef84936b07fb0f9796b4d33918ac4ef900 (refs/remotes/origin/master)
 > git.exe config core.sparsecheckout # timeout=10
 > git.exe checkout -f 45fd33ef84936b07fb0f9796b4d33918ac4ef900 # timeout=10
 > git.exe branch -a -v --no-abbrev # timeout=10
 > git.exe branch -D master # timeout=10
 > git.exe checkout -b master 45fd33ef84936b07fb0f9796b4d33918ac4ef900 # timeout=10
Commit message: "Update all non-major dependencies (#2067)"
 > git.exe rev-list --no-walk 45fd33ef84936b07fb0f9796b4d33918ac4ef900 # timeout=10
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (SonarQube Analysis)
[Pipeline] withSonarQubeEnv
Injecting SonarQube environment variables using the configuration: sonarqube
[Pipeline] {
[Pipeline] bat
```

Sonarqube Analysis:-



main — 44k Lines of Code · Version not provided · Set as homepage

Don't let issues accumulate. Discover 'Clean as You Code'!

Learn how to improve your code base by cleaning only new code.

Quality Gate: Passed — Last analysis 10 minutes ago

The last analysis has warnings. See details

Security	Reliability	Maintainability
703 Open issues (E)	2.4k Open issues (E)	4.6k Open issues (A)

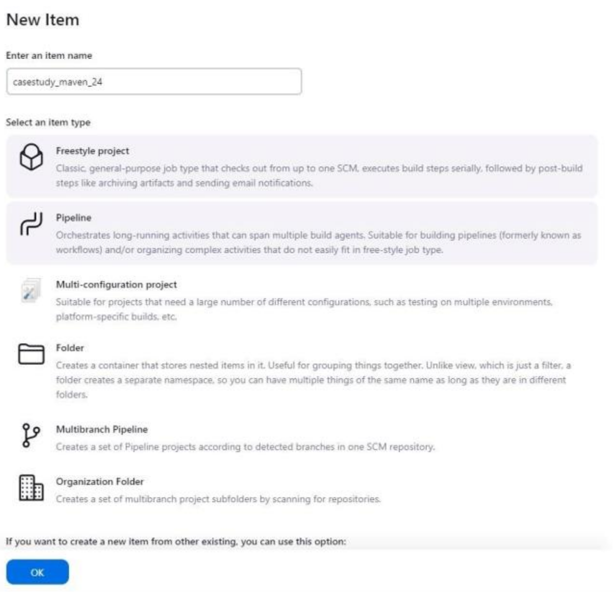
Accepted issues	Coverage	Duplications
0 — Valid issues that were not fixed	0.0% — On 23k lines to cover.	2.9% — On 58k lines.

Security Hotspots

Thus, the Python project was successfully analyzed with SonarQube.

B]For Maven Project:-

1) Make a Jenkins Pipeline.


New Item

Enter an item name

casestudy_maven_24

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.

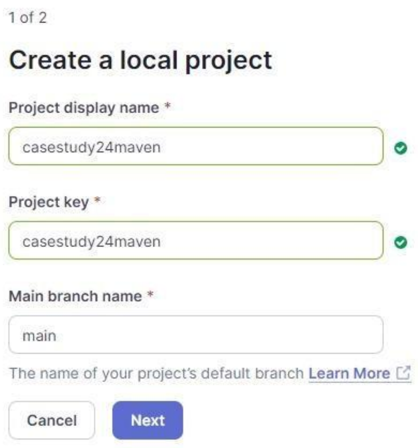
Multibranch Pipeline
Creates a set of Pipeline projects according to detected branches in one SCM repository.

Organization Folder
Creates a set of multibranch project subfolders by scanning for repositories.

If you want to create a new item from other existing, you can use this option:

Java code to be analyzed :- https://github.com/SonarSource/sonar-scanner-maven.git

2) Create a sonarqube project named **casestudy24maven.**


1 of 2

Create a local project

Project display name *: casestudy24maven

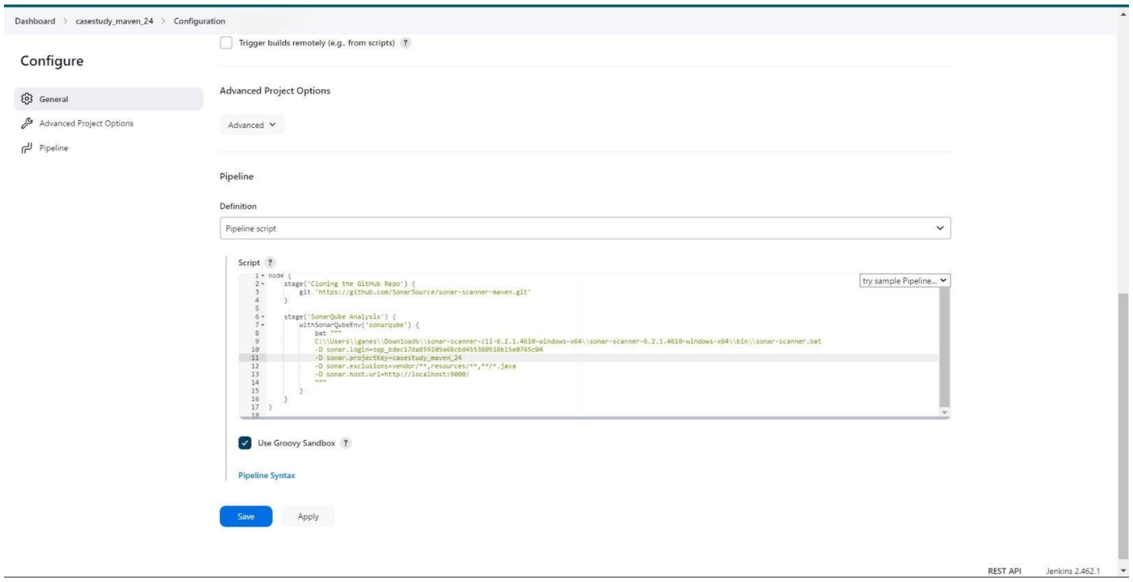
Project key *: casestudy24maven

Main branch name *: main

The name of your project's default branch Learn More

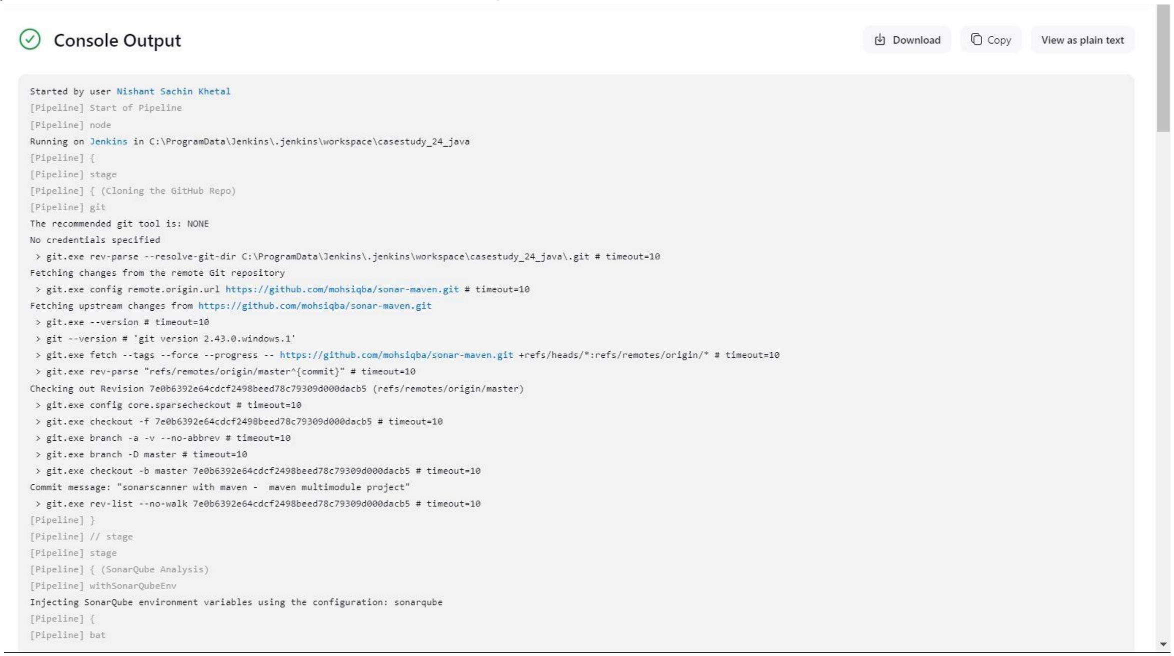
Pipeline Script:node

```
{
  stage('Cloning the GitHub Repo') {
    git 'https://github.com/SonarSource/sonar-scanner-maven.git'
  }
  stage('SonarQube Analysis') {
    withSonarQubeEnv('sonarqube') {
            bat
"C:\\Users\\nish\\Downloads\\sonar-scanner-cli-6.1.0.4477-windows-x64\\sonar-scanner-6.1.0.4
477-windows-x64\\bin\\sonar-scanner.bat " +
                "-D sonar.login=admin " +
                "-D sonar.password=HareKrishna#108 " +
                "-D sonar.projectKey=casestudy24maven " +
                "-D sonar.exclusions=vendor/**,resources/**,**/*.java " +
                "-D sonar.host.url=http://localhost:9000/"
        }
    }
}
```



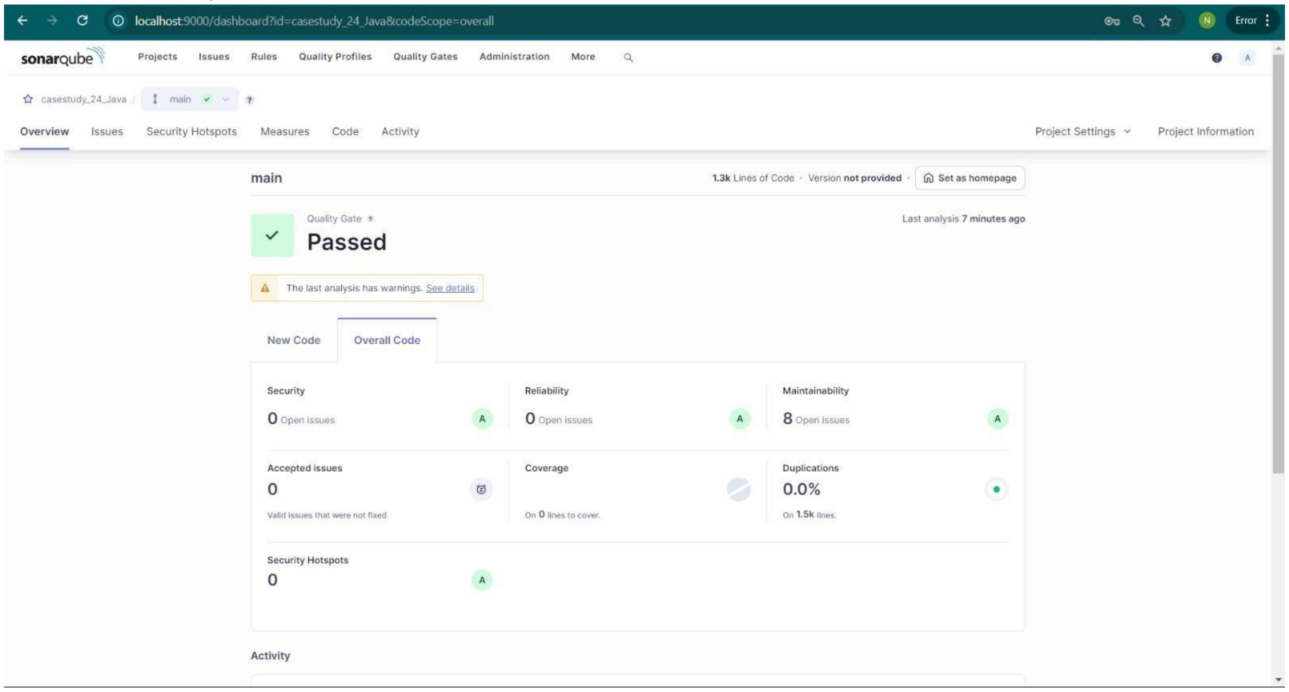
```
node {
    stage('Cloning the GitHub Repo') {
        git 'https://github.com/SonarSource/sonar-scanner-maven.git'
    }

    stage('SonarQube Analysis') {
        withSonarQubeEnv('sonarqube') {
            bat """
            C:\\Users\\ganes\\Downloads\\sonar-scanner-cli-6.2.1.4610-windows-x64\\sonar-scanner-6.2.1.4610-windows-x64\\bin\\sonar-scanner.bat
            -D sonar.login=sqp_bdec17da859205a68cbd455380518b15e0765c04
            -D sonar.projectKey=casestudy_maven_24
            -D sonar.exclusions=vendor/**,resources/**,**/*.java
            -D sonar.host.url=http://localhost:9000/
            """
        }
    }
}
```

3) Build and check its console output:-



```
Started by user Nishant Sachin Khetal
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in C:\ProgramData\Jenkins\.jenkins\workspace\casestudy_24_java
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Cloning the GitHub Repo)
[Pipeline] git
The recommended git tool is: NONE
No credentials specified
 > git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\.jenkins\workspace\casestudy_24_java\.git # timeout=10
Fetching changes from the remote Git repository
 > git.exe config remote.origin.url https://github.com/mohsiqba/sonar-maven.git # timeout=10
Fetching upstream changes from https://github.com/mohsiqba/sonar-maven.git
 > git.exe --version # timeout=10
 > git --version # 'git version 2.43.0.windows.1'
 > git.exe fetch --tags --force --progress -- https://github.com/mohsiqba/sonar-maven.git +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
Checking out Revision 7e0b6392e64cdcf2498beed78c79309d000dacb5 (refs/remotes/origin/master)
 > git.exe config core.sparsecheckout # timeout=10
 > git.exe checkout -f 7e0b6392e64cdcf2498beed78c79309d000dacb5 # timeout=10
 > git.exe branch -a -v --no-abbrev # timeout=10
 > git.exe branch -D master # timeout=10
 > git.exe checkout -b master 7e0b6392e64cdcf2498beed78c79309d000dacb5 # timeout=10
Commit message: "sonarscanner with maven -  maven multimodule project"
 > git.exe rev-list --no-walk 7e0b6392e64cdcf2498beed78c79309d000dacb5 # timeout=10
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (SonarQube Analysis)
[Pipeline] withSonarQubeEnv
Injecting SonarQube environment variables using the configuration: sonarqube
[Pipeline] {
[Pipeline] bat
```

SonarQube analysis



Attributes:-

Consistency: Ensures the code follows uniform patterns and styles throughout, making it easier to read and understand.

Intentionality: Reflects that the code is written with clear purpose and reasoning, avoiding unnecessary complexity.

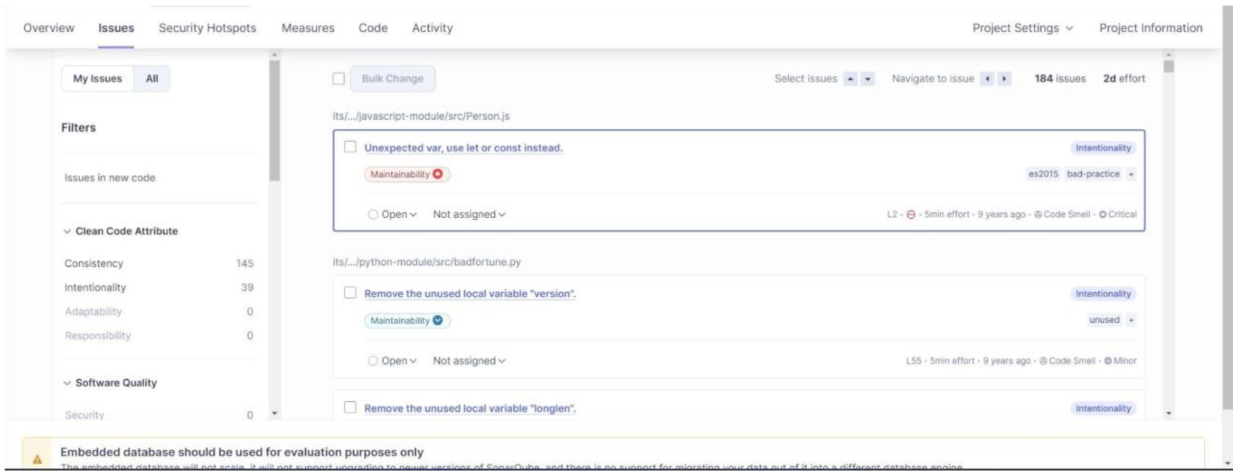
Adaptability: Indicates how flexible the code is to accommodate changes or new features without requiring major rewrites.

Responsibility: Reflects that the code is well-structured with clear ownership of functions or classes, adhering to the Single Responsibility Principle.

Security: Measures how well the code protects against vulnerabilities and malicious attacks.

Reliability: Assesses the code's ability to function correctly under different conditions and its resistance to failures.

Maintainability: Evaluates how easily the code can be modified, debugged, or enhanced for long-term use.



184 issues 2d effort

Filters

Issues in new code

Clean Code Attribute

Consistency	145
Intentionality	39
Adaptability	0
Responsibility	0

Software Quality

Security	0

its/.../javascript-module/src/Person.js

Unexpected var, use let or const instead. Intentionality
Maintainability es2015 bad-practice
Open Not assigned L2 · 5min effort · 9 years ago · Code Smell · Critical

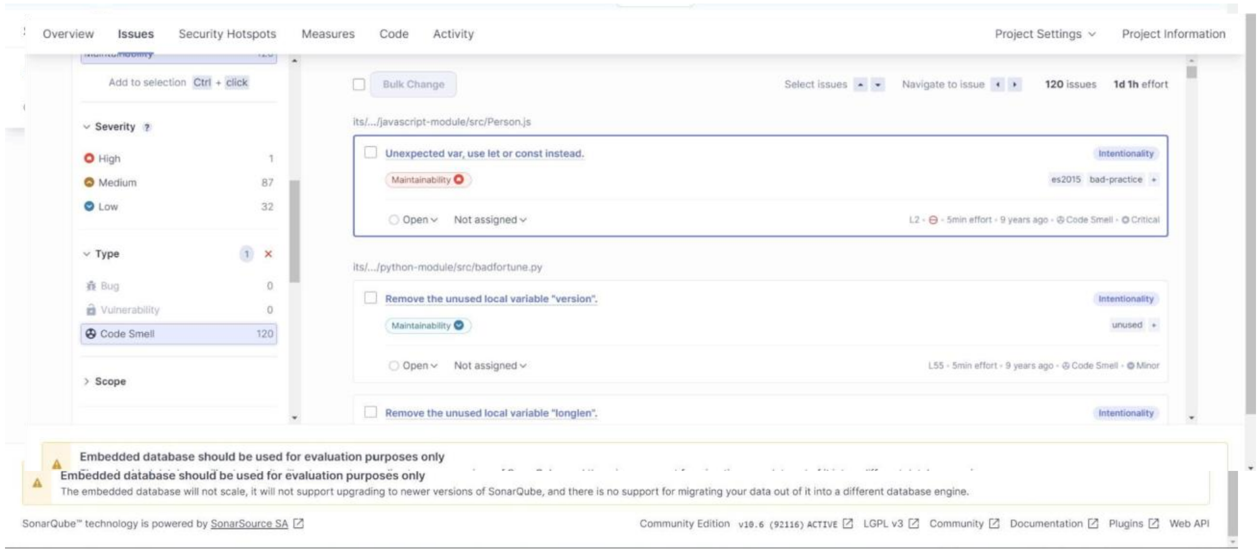
its/.../python-module/src/badfortune.py

Remove the unused local variable "version". Intentionality
Maintainability unused
Open Not assigned L55 · 5min effort · 9 years ago · Code Smell · Minor

Remove the unused local variable "longlen". Intentionality

Embedded database should be used for evaluation purposes only

} Code smells:-
A code smell is an indicator of potential design issues in code that, while not causing immediate errors, can lead to problems over time, such as maintainability and scalability issues. Common examples include large classes, long methods, and duplicated code, which increase complexity and technical debt. Identifying and addressing code smells through refactoring improves code quality and readability, ensuring long-term reliability and maintainability
ii} Security Hotspots:-



120 issues 1d 1h effort

Severity

High	1
Medium	87
Low	32

Type

Bug	0
Vulnerability	0
Code Smell	120

Scope

its/.../javascript-module/src/Person.js

Unexpected var, use let or const instead. Intentionality
Maintainability es2015 bad-practice
Open Not assigned L2 · 5min effort · 9 years ago · Code Smell · Critical

its/.../python-module/src/badfortune.py

Remove the unused local variable "version". Intentionality
Maintainability unused
Open Not assigned L55 · 5min effort · 9 years ago · Code Smell · Minor

Remove the unused local variable "longlen". Intentionality

Embedded database should be used for evaluation purposes only
Embedded database should be used for evaluation purposes only
The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine.

SonarQube™ technology is powered by SonarSource SA   Community Edition v10.6 (92116) ACTIVE LGPL v3 Community Documentation Plugins Web API

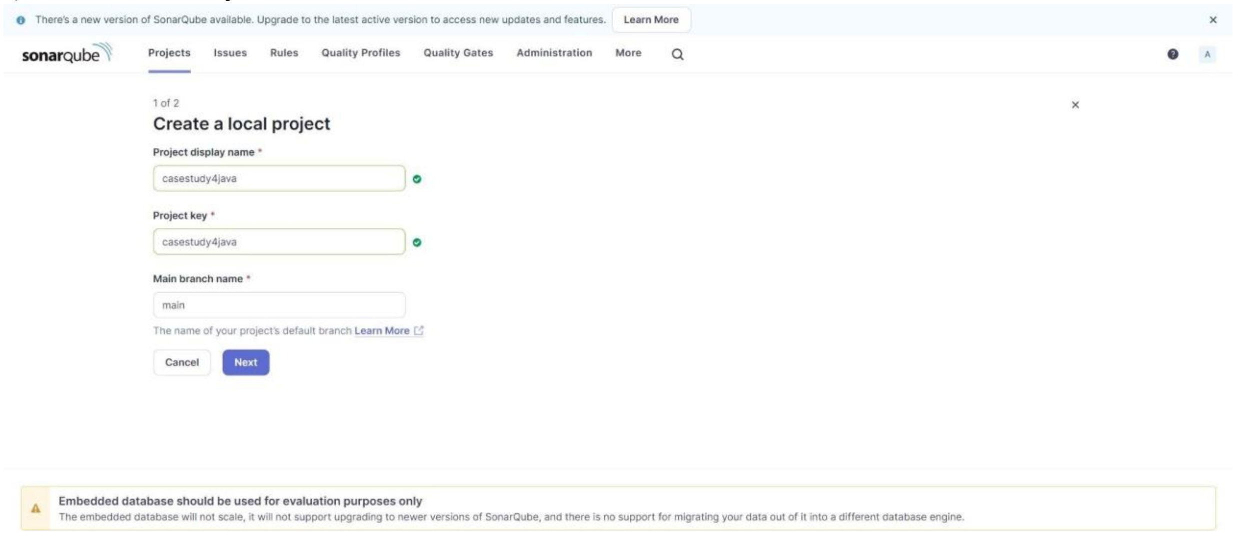
A **security hotspot** is a piece of security-sensitive code that requires review to determine if it poses a threat. Unlike vulnerabilities, which need immediate fixes, hotspots must be assessed by the developer to decide if action is required. Fixing hotspots enhances an application's resilience against attacks. SonarQube assigns review priority based on standards like OWASP Top 10 and CWE Top 25, with hotspots categorized as high, medium, or low priority. Developers review the code, assess risks, and apply necessary fixes or mark it as safe. Addressing security hotspots strengthens code security while allowing for informed, context-based decisions.

Successfully analyzed the Java Maven project using SonarQube.

3] Java Project:-

Project Link: https://github.com/mohsiqba/sonar-maven.git

1)SonarQube Project:


There's a new version of SonarQube available. Upgrade to the latest active version to access new updates and features.

1 of 2

Create a local project

Project display name *: casestudy4java

Project key *: casestudy4java

Main branch name *: main

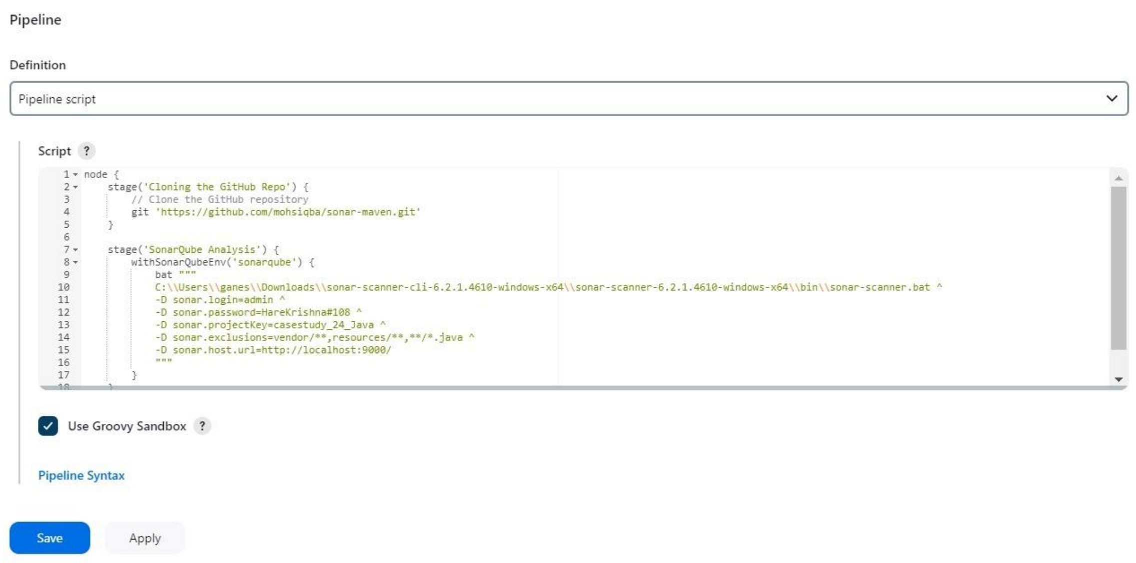
The name of your project's default branch Learn More

Embedded database should be used for evaluation purposes only

The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine.

Pipeline Script:-

```
node {
   stage('Cloning the GitHub Repo') { git
      'https://github.com/mohsiqba/sonar-maven.git'
   }
   stage('SonarQube Analysis') {
      withSonarQubeEnv('sonarqube') {
         bat
"C:\\Users\\nish\\Downloads\\sonar-scanner-cli-6.1.0.4477-windows-x64\\sonar-scanner-6.1.0.4
477-windows-x64\\bin\\sonar-scanner.bat " +
            "-D sonar.login=admin " +
            "-D sonar.password=HareKrishna#108 " +
            "-D sonar.projectKey=casestudy24java " +
            "-D sonar.exclusions=vendor/**,resources/**,**/*.java " +
            "-D sonar.host.url=http://localhost:9000/"
      }
   }
}
```



Pipeline

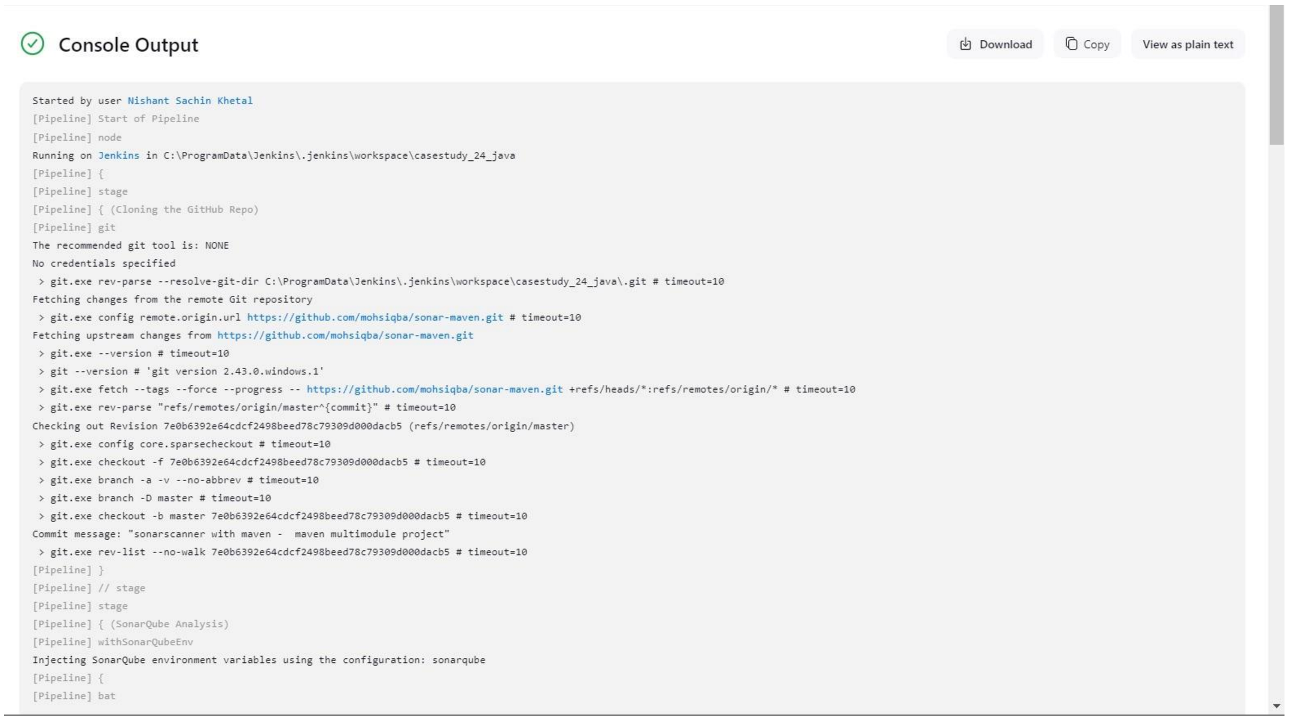
Definition

Pipeline script

Script

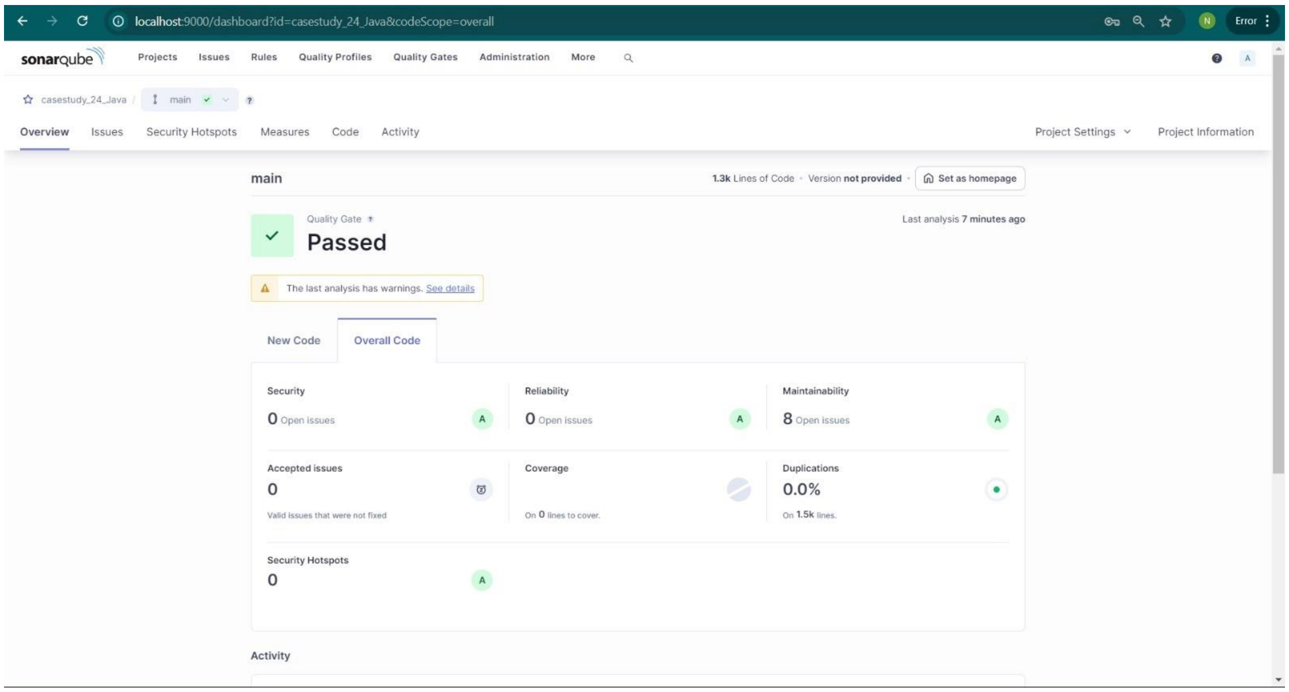
```
node {
    stage('Cloning the GitHub Repo') {
        // Clone the GitHub repository
        git 'https://github.com/mohsiqba/sonar-maven.git'
    }

    stage('SonarQube Analysis') {
        withSonarQubeEnv('sonarqube') {
            bat """
            C:\\Users\\ganes\\Downloads\\sonar-scanner-cli-6.2.1.4610-windows-x64\\sonar-scanner-6.2.1.4610-windows-x64\\bin\\sonar-scanner.bat ^
            -D sonar.login=admin ^
            -D sonar.password=HareKrishna#108 ^
            -D sonar.projectKey=casestudy_24_Java ^
            -D sonar.exclusions=vendor/**,resources/**,**/*.java ^
            -D sonar.host.url=http://localhost:9000/
            """
        }
```

✓ Use Groovy Sandbox



Console Output

```
Started by user Nishant Sachin Khetal
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in C:\ProgramData\Jenkins\.jenkins\workspace\casestudy_24_java
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Cloning the GitHub Repo)
[Pipeline] git
The recommended git tool is: NONE
No credentials specified
 > git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\.jenkins\workspace\casestudy_24_java\.git # timeout=10
Fetching changes from the remote Git repository
 > git.exe config remote.origin.url https://github.com/mohsiqba/sonar-maven.git # timeout=10
Fetching upstream changes from https://github.com/mohsiqba/sonar-maven.git
 > git.exe --version # timeout=10
 > git --version # 'git version 2.43.0.windows.1'
 > git.exe fetch --tags --force --progress -- https://github.com/mohsiqba/sonar-maven.git +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
Checking out Revision 7e0b6392e64cdcf2498beed78c79309d000dacb5 (refs/remotes/origin/master)
 > git.exe config core.sparsecheckout # timeout=10
 > git.exe checkout -f 7e0b6392e64cdcf2498beed78c79309d000dacb5 # timeout=10
 > git.exe branch -a -v --no-abbrev # timeout=10
 > git.exe branch -D master # timeout=10
 > git.exe checkout -b master 7e0b6392e64cdcf2498beed78c79309d000dacb5 # timeout=10
Commit message: "sonarscanner with maven -  maven multimodule project"
 > git.exe rev-list --no-walk 7e0b6392e64cdcf2498beed78c79309d000dacb5 # timeout=10
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (SonarQube Analysis)
[Pipeline] withSonarQubeEnv
Injecting SonarQube environment variables using the configuration: sonarqube
[Pipeline] {
[Pipeline] bat
```



main

1.3k Lines of Code · Version not provided

Quality Gate
Passed

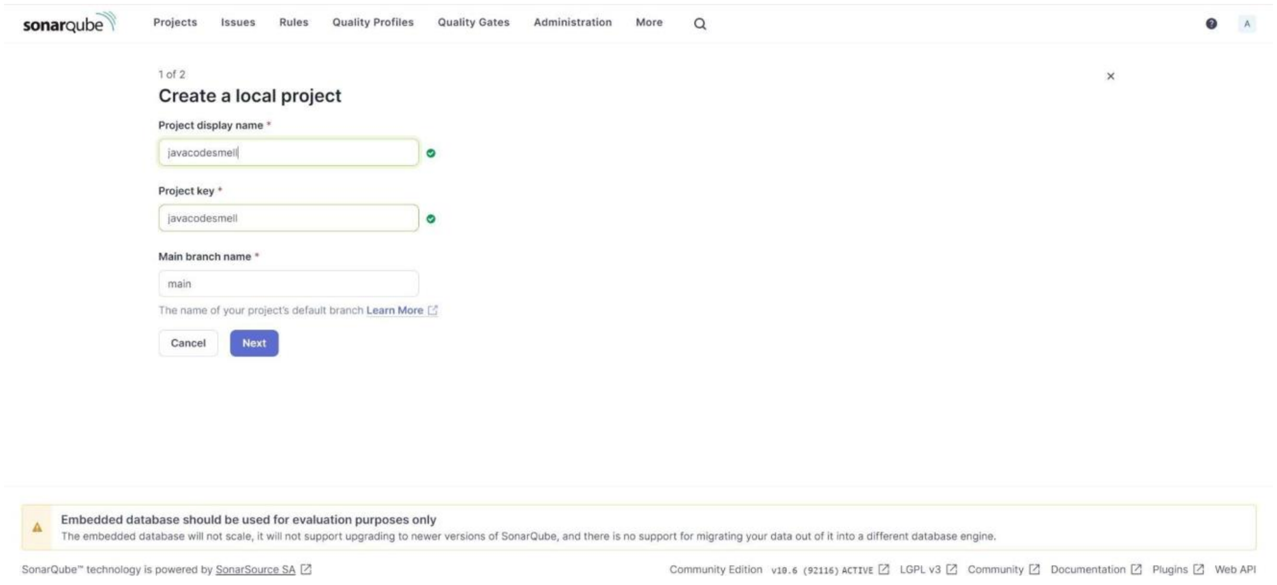
Last analysis 7 minutes ago

The last analysis has warnings. See details

Security 0 Open issues — A	Reliability 0 Open issues — A	Maintainability 8 Open issues — A
Accepted issues 0 Valid issues that were not fixed	Coverage On 0 lines to cover.	Duplications 0.0% On 1.5k lines.
Security Hotspots 0 — A		

Successfully analyzed the Java project using SonarQube. 4] Java Project:-
Project Link: https://github.com/nerdschoolbergen/code-smells.git

1)SonarQube Project:



1 of 2
Create a local project

Project display name *
javacodesmell

Project key *
javacodesmell

Main branch name *
main

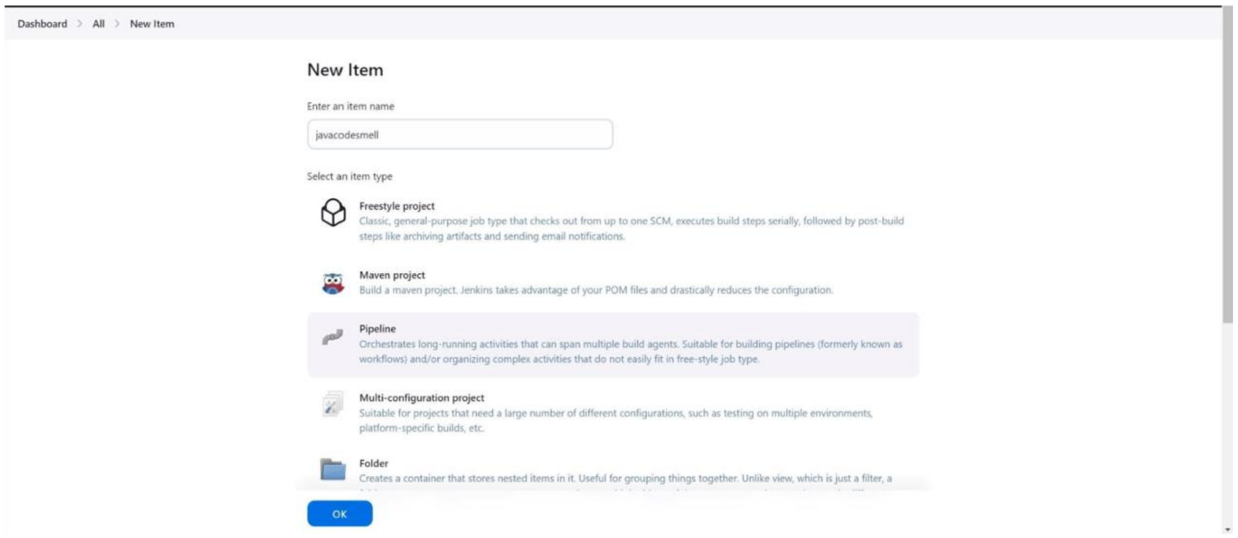
The name of your project's default branch Learn More

Embedded database should be used for evaluation purposes only
The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine.

SonarQube™ technology is powered by SonarSource SA

Community Edition v10.6 (92116) ACTIVE LGPL v3 Community Documentation Plugins Web API

2)Jenkins Pipeline:



Dashboard > All > New Item

New Item

Enter an item name

javacodesmell

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.

Maven project
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

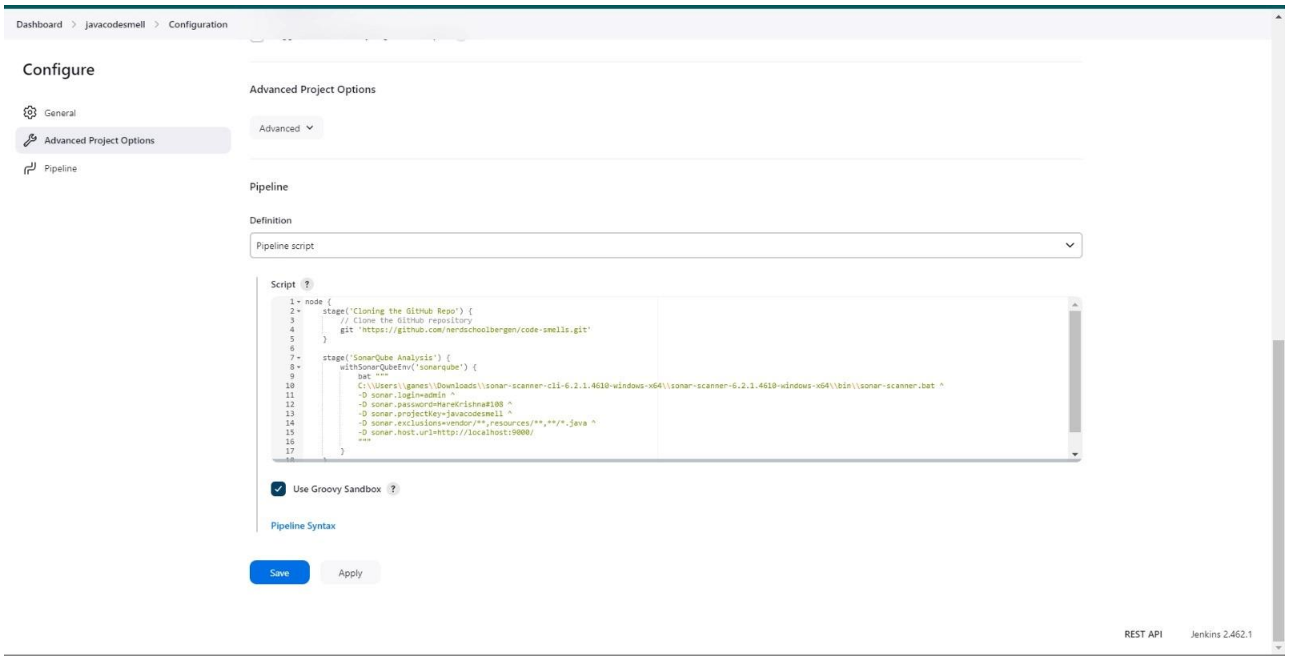
Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

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

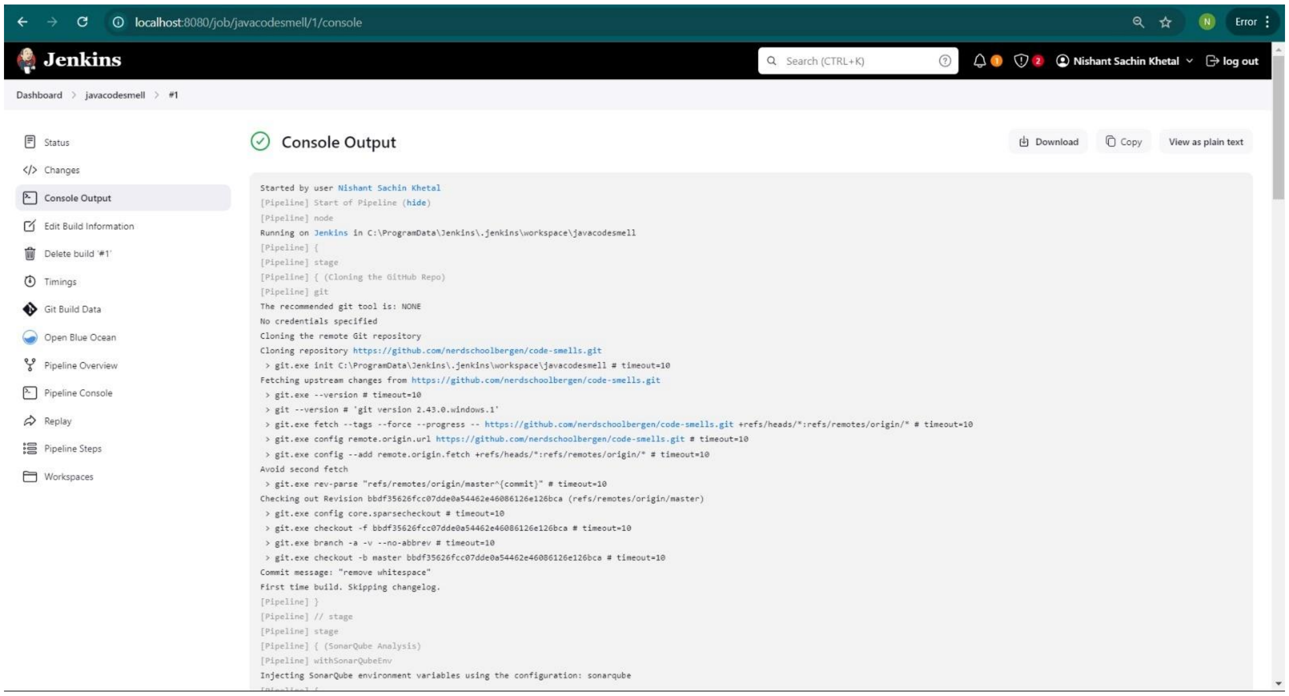
Pipeline Script:-

```
node {
   stage('Cloning the GitHub Repo') { git
      'https://github.com/nerdschoolbergen/code-smells.git'
   }
   stage('SonarQube Analysis') {
      withSonarQubeEnv('sonarqube') {
         bat
"C:\\Users\\nish\\Downloads\\sonar-scanner-cli-6.1.0.4477-windows-x64\\sonar-scanner-6.1.0.4
477-windows-x64\\bin\\sonar-scanner.bat " +
            "-D sonar.login=admin " + // Ensure there is a space before the next option
            "-D sonar.password=HareKrishna#108 " +
            "-D sonar.projectKey=javacodesmell " +
            "-D sonar.exclusions=vendor/**,resources/**,**/*.java " +
            "-D sonar.host.url=http://localhost:9000/"
      }
   }
}
```



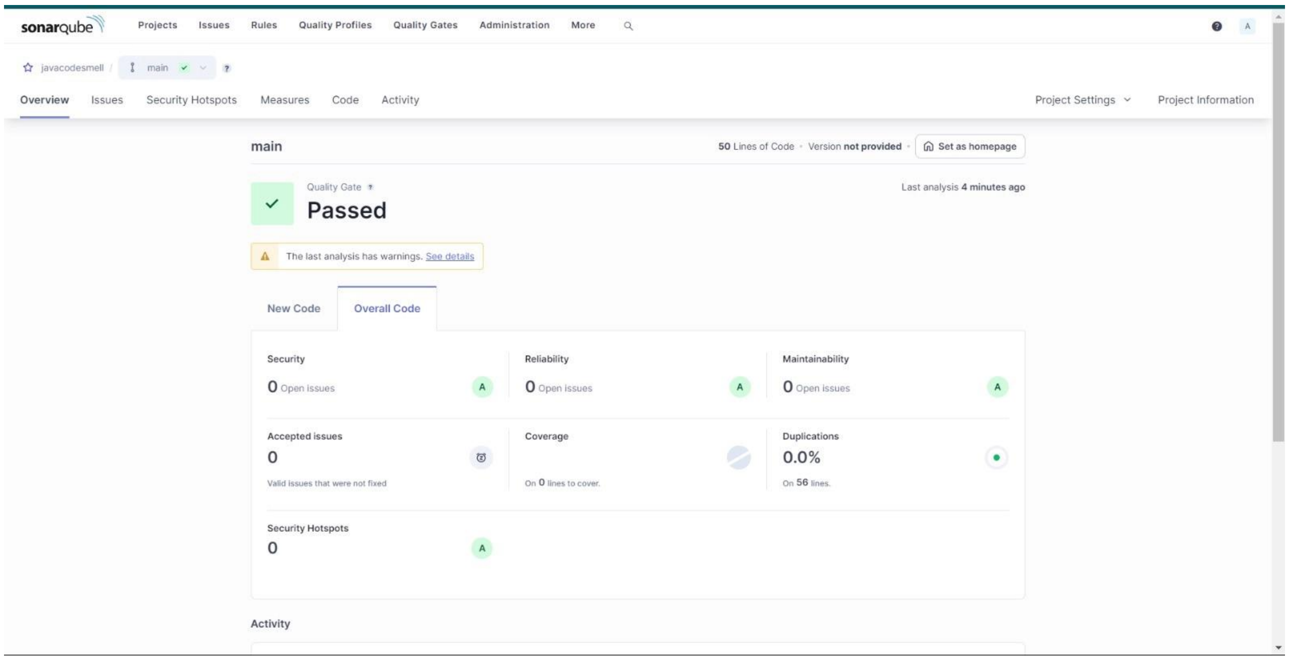
```
node {
    stage('Cloning the GitHub Repo') {
        // Clone the GitHub repository
        git 'https://github.com/nerdschoolbergen/code-smells.git'
    }

    stage('SonarQube Analysis') {
        withSonarQubeEnv('sonarqube') {
            bat """
            C:\\Users\\ganes\\Downloads\\sonar-scanner-cli-6.2.1.4610-windows-x64\\sonar-scanner-6.2.1.4610-windows-x64\\bin\\sonar-scanner.bat ^
            -D sonar.login=admin ^
            -D sonar.password=HareKrishna#108 ^
            -D sonar.projectKey=javacodesmell ^
            -D sonar.exclusions=vendor/**,resources/**,**/*.java ^
            -D sonar.host.url=http://localhost:9000/
            """
        }
```

3) Console Output:



```
Started by user Nishant Sachin Khetal
[Pipeline] Start of Pipeline (hide)
[Pipeline] node
Running on Jenkins in C:\ProgramData\Jenkins\.jenkins\workspace\javacodesmell
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Cloning the GitHub Repo)
[Pipeline] git
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/nerdschoolbergen/code-smells.git
 > git.exe init C:\ProgramData\Jenkins\.jenkins\workspace\javacodesmell # timeout=10
Fetching upstream changes from https://github.com/nerdschoolbergen/code-smells.git
 > git.exe --version # timeout=10
 > git --version # 'git version 2.43.0.windows.1'
 > git.exe fetch --tags --force --progress -- https://github.com/nerdschoolbergen/code-smells.git +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git.exe config remote.origin.url https://github.com/nerdschoolbergen/code-smells.git # timeout=10
 > git.exe config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
 > git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
Checking out Revision bbdf35626fcc07dde0a54462e46086126e126bca (refs/remotes/origin/master)
 > git.exe config core.sparsecheckout # timeout=10
 > git.exe checkout -f bbdf35626fcc07dde0a54462e46086126e126bca # timeout=10
 > git.exe branch -a -v --no-abbrev # timeout=10
 > git.exe checkout -b master bbdf35626fcc07dde0a54462e46086126e126bca # timeout=10
Commit message: "remove whitespace"
First time build. Skipping changelog.
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (SonarQube Analysis)
[Pipeline] withSonarQubeEnv
Injecting SonarQube environment variables using the configuration: sonarqube
```

SonarQube Analysis:



main

50 Lines of Code · Version not provided · Set as homepage

Quality Gate
Passed

Last analysis 4 minutes ago

The last analysis has warnings. See details

New Code | Overall Code

Security
0 Open issues — A

Reliability
0 Open issues — A

Maintainability
0 Open issues — A

Accepted issues
0
Valid issues that were not fixed

Coverage
On 0 lines to cover.

Duplications
0.0%
On 56 lines.

Security Hotspots
0 — A

Activity

Successfully analyzed the Java project using SonarQube.

Conclusion :

Through this case study, Jenkins and SonarQube were successfully integrated to automate the continuous integration and static code analysis of Python and Java projects. The CI pipeline helps maintain high code quality and ensures that code adheres to security standards before being pushed to production.