

[illegible]

- **sudo yum update -y**
- **sudo wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-stable/jenkins.repo**
- **sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key**
- **sudo yum install jenkins -y**
- **sudo systemctl enable jenkins**
- **sudo systemctl start jenkins**
- **sudo systemctl status jenkins**

```
[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$1$NonAttained: Completed initialization
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$onReady: Jenkins is fully up and
Oct 18 18:12:27 ip-172-31-47-120.ec2.internal systemd[1]: Started jenkins.service - Jenkins Continuous Integration Server.
Oct 18 18:12:27 ip-172-31-47-120.ec2.internal jenkins[25878]: 2024-10-18 18:12:27.436+0000 [id=47] INFO h.n.DownloadService$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

Continue

Getting Started

Jenkins is ready!

Your Jenkins setup is complete.

Start using Jenkins

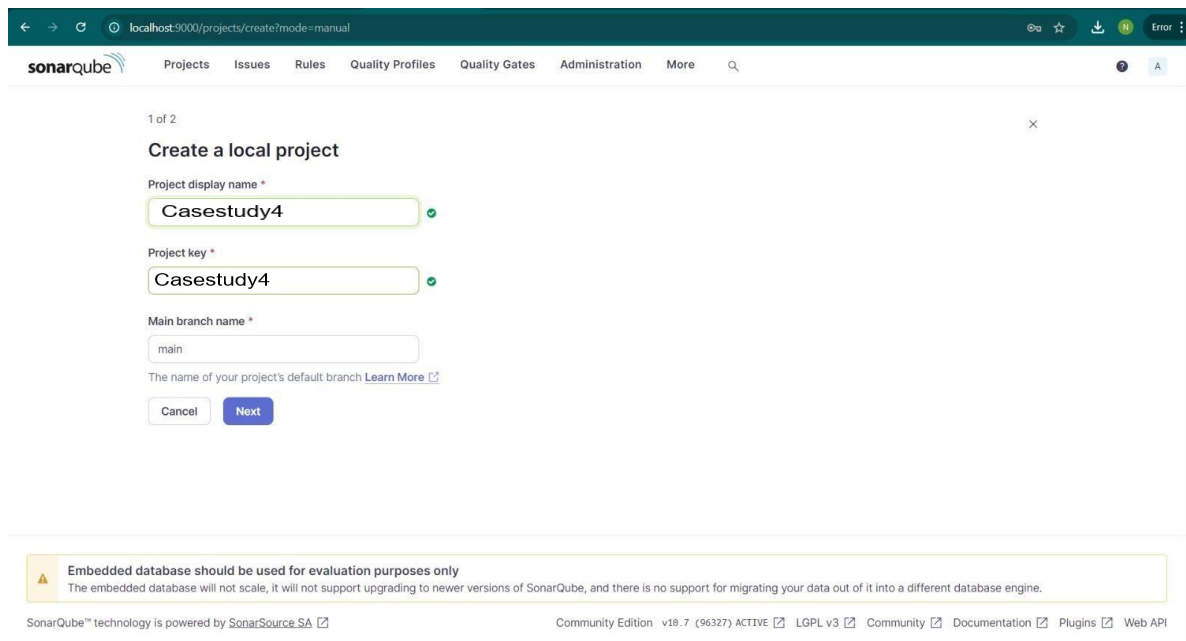
Jenkins 2.462.3

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

A] Sonarqube project:-

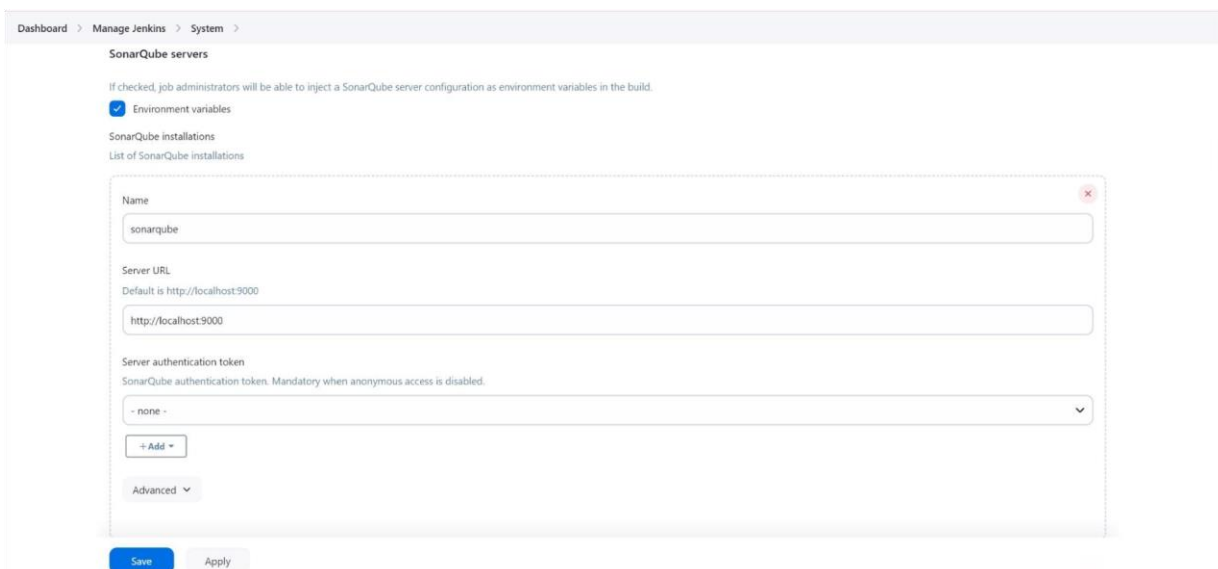
Python Project : <https://github.com/piomin/sample-java-sonar>

1) Create a sonarqube project named casestudy_24.



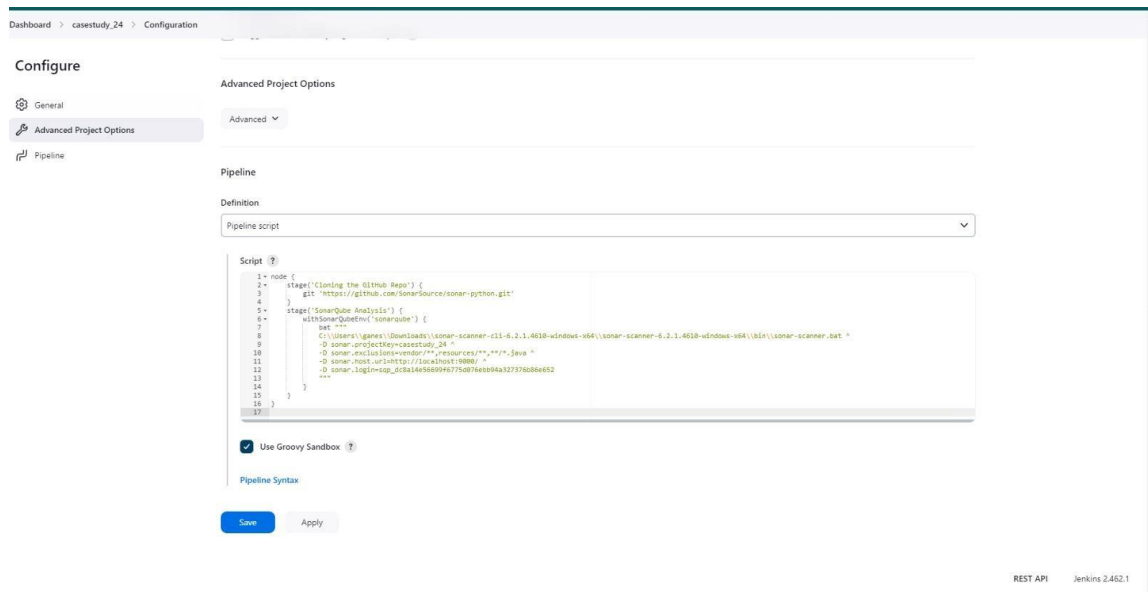
The screenshot shows the SonarQube web interface at localhost:9000/projects/create?mode=manual. The 'Create a local project' form is displayed with the following fields: 'Project display name' (Casestudy4), 'Project key' (Casestudy4), and 'Main branch name' (main). There are 'Cancel' and 'Next' buttons at the bottom of the form. A warning message at the bottom states: '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.' The footer includes 'SonarQube™ technology is powered by SonarSource SA' and 'Community Edition v10.7 (96327) ACTIVE'.

2) Configuration of SonarQube in Jenkins.



The screenshot shows the Jenkins 'SonarQube servers' configuration page. The 'Environment variables' checkbox is checked. The 'SonarQube installations' section shows a list of installations with the following details: Name (sonarqube), Server URL (http://localhost:9000), and Server authentication token (none). There are 'Add', 'Save', and 'Apply' buttons at the bottom.

3) Deploy Code on Jenkins Pipeline :



4) Pipeline Script :

Script:- node

```
{
  stage('Cloning the GitHub Repo') { git
    'https://github.com/SonarSource/sonar-python.git'
  }
  stage('SonarQube Analysis') {
    withSonarQubeEnv('sonarqube') {
      bat
      "C:\\bhagt\\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=Admin@123456 " +
        "-D sonar.projectKey=casestudy_24 " +
        "-D sonar.exclusions=vendor/**,resources/**,**/*.java
        " + "-D sonar.host.url=http://localhost:9000/"
    }
  }
}
```

5) Open Console Output on Jenkins to check whether the Output is Success or not.

Console Output

[Download](#) [Copy](#) [View as plain text](#)

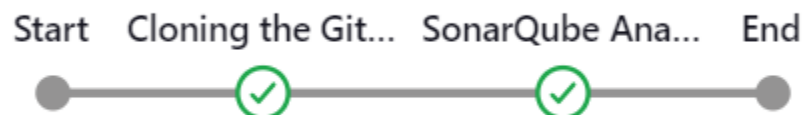
```
started by user Roshan Bhagtani
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in C:\ProgramData\Jenkins\jenkins\workspace\Case study 4
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Cloning the Github Repo)
[Pipeline] git
The recommended git tool is: NONE
no credentials specified
> git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\jenkins\workspace\Case study 4\.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 2.47.0.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 56836f0727b22c0b08f85fcd2577861cd97b75c9 (refs/remotes/origin/master)
> git.exe config core.sparsecheckout # timeout=10
> git.exe checkout -f 56836f0727b22c0b08f85fcd2577861cd97b75c9 # timeout=10
> git.exe branch -a -v --no-abbrev # timeout=10
> git.exe branch -D master # timeout=10
> git.exe checkout -B master 56836f0727b22c0b08f85fcd2577861cd97b75c9 # timeout=10
Commit message: "SonarQube-2188 Introduce the concept of unresolvedReportType (02077)"
> git.exe rev-list --no-walk 45f033ef84936087f0ef979804033932ac4ef900 # timeout=10
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (SonarQube Analysis)
[Pipeline] withSonarQubeEnv
Injecting SonarQube environment variables using the configuration: sonarqube
[Pipeline] {
[Pipeline] bat

C:\ProgramData\Jenkins\jenkins\workspace\Case study 4>"C:\Users\bhagt\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=admin@123456 -D sonar.projectkey=casestudy4 -D sonar.exclusions=vendor/**,resources/**,**/*.java -D sonar.host.url=http://localhost:9000/
21:13:12.615 INFO Scanner configuration file: C:\Users\bhagt\Downloads\sonar-scanner-cli-6.2.1.4610-windows-x64\sonar-scanner-6.2.1.4610-windows-x64\bin\..conf\sonar-scanner.properties
```

6)SonarQube assessment(Analysis):

The screenshot displays the SonarQube web interface for a project named 'casestudy4'. The top navigation bar includes links for Projects, Issues, Rules, Quality Profiles, Quality Gates, Administration, and More. The main content area shows the 'main' branch with 44k lines of code and a 'Version not provided' status. A message box prompts the user to 'Don't let issues accumulate. Discover 'Clean as You Code!'' with 'Take the Tour' and 'Not now' buttons. Below this, a 'Quality Gate' section shows a green checkmark and the word 'Passed', with a note that the last analysis was 12 minutes ago. A warning box indicates 'The last analysis has warnings. See details'. At the bottom, a 'New Code' section shows 'New Code: Since October 20, 2024 Started 1 day ago'.

#11



Thus, the Python project was successfully analyzed with SonarQube.

6. Conclusion :

This case study highlights the successful integration of Jenkins and SonarQube to automate continuous integration and static code analysis for Python and Java projects. This CI pipeline

enhances code quality and ensures compliance with security standards before deployment, fostering continuous improvement and accountability within the development team. Ultimately, this integration reduces vulnerabilities and streamlines the software delivery lifecycle.