**SONARQUBE EXPLANATION & SETUP**

**1.What is SonarQube?**

SonarQube is a **code quality and security management tool** that performs **static code analysis** to identify:

* Bugs.
* Code smells (maintainability issues).
* Security vulnerabilities.
* Compliance issues (e.g., OWASP Top 10, SANS Top 25).

It helps ensure that the codebase meets high-quality standards and is secure, reliable, and maintainable.

**2. Why is SonarQube Important for Source Code Security?**

**Security Benefits:**

* Detects **vulnerabilities** in the code that hackers can exploit, such as SQL injection, hardcoded credentials, and weak encryption.
* Flags **insecure coding practices** that may lead to future breaches.
* Assesses **third-party libraries** for known vulnerabilities (e.g., CVEs).

**Proactive Issue Detection:**

* SonarQube allows us to identify and address issues **before they reach production**, reducing risk and technical debt.

**Compliance:**

* Aligns with industry standards like OWASP, ISO/IEC 27001, and PCI DSS, ensuring code compliance.

**3. Key Features**

* **Multi-Language Support**: SonarQube supports **.NET** and 30+ other programming languages.
* **Easy Integration**: Works with CI/CD pipelines (e.g., Jenkins, Azure DevOps) to automate analysis.
* **Clear Reporting**: Provides a dashboard that visualizes code quality and security status, making it easy to track progress.
* **Continuous Improvement**: Tracks quality over time, encouraging a culture of continuous improvement.

**4. Benefits to the Business**

* **Reduced Costs**:
  + Fixing bugs and vulnerabilities early in the development cycle is much cheaper than post-production fixes.
* **Improved Reputation**:
  + Delivering secure and high-quality software increases customer confidence and trust.
* **Developer Productivity**:
  + Provides actionable feedback directly in the code, enabling developers to learn and fix issues efficiently.
* **Scalable Solution**:
  + Can handle small to large-scale codebases and multiple projects, making it suitable for long-term use.

**5. How Does SonarQube Work for Source Code Security Checks?**

1. **Static Analysis**:
   * Scans the source code to identify bugs, vulnerabilities, and other issues without executing the application.
2. **Rules-Based Checks**:
   * Uses pre-configured and customizable rules to enforce coding standards and security best practices.
3. **Reports and Alerts**:
   * Generates detailed reports highlighting the severity and location of issues, prioritizing fixes.
4. **Continuous Integration**:
   * Integrates into the development workflow, performing automatic checks during the build process.

**SONARQUBE SETUP**

**System Requirements**

* **RAM**: Minimum 2GB (4GB+ recommended for smooth performance).
* **CPU**: Dual-core or better.
* **Disk Space**: 2GB for SonarQube plus additional space for projects.
* **Java**: AdoptOpenJDK 11 or higher (SonarQube doesn’t support Java 8).
* **Database**: PostgreSQL (preferred), MySQL, or SQL Server.

**Step 1: Update the Package Index**

Ensure your system is up to date:

sudo apt update && sudo apt upgrade -y

**Step 2: Install OpenJDK 17**

Ubuntu 20.04 includes OpenJDK 17 in its default repositories. You can install it using:

sudo apt install openjdk-17-jdk -y

**Step 3: Verify the Installation**

Check the installed Java version to ensure it’s OpenJDK 17:

java -version

**Step 4: Update the System**

Ensure your system is up-to-date before installing PostgreSQL:

sudo apt update && sudo apt upgrade -y

**Step 5: Install PostgreSQL**

1. Install PostgreSQL and related tools:

sudo apt install postgresql postgresql-contrib -y

1. Verify that PostgreSQL is installed and running:

sudo systemctl status postgresql

**Step 6: Secure PostgreSQL**

1. Switch to the PostgreSQL user account:

sudo -i -u postgres

1. Access the PostgreSQL prompt:

psql

**Step 7: Create a Database and User for SonarQube**

1. Create a database named sonarqube:

CREATE DATABASE sonarqube;

1. Create a user for SonarQube with a password:

CREATE USER sonar WITH ENCRYPTED PASSWORD 'your\_secure\_password';

Replace your\_secure\_password with a strong password.

1. Grant the user full privileges on the sonarqube database:

GRANT ALL PRIVILEGES ON DATABASE sonarqube TO sonar;

1. Exit the PostgreSQL prompt:

\q

1. Exit the PostgreSQL user account:

exit

**Step 8**:**Download and Install SonarQube**

wget https://binaries.sonarsource.com/Distribution/sonarqube/sonarqube-9.9.1.69595.zip

sudo apt install unzip

unzip sonarqube-9.9.1.69595.zip -d /opt

sudo mv /opt/sonarqube-9.9.1.69595 /opt/sonarqube

**Step 9**:**Create a SonarQube User**

sudo adduser --system --no-create-home --group --disabled-login sonarqube

sudo chown -R sonarqube:sonarqube /opt/sonarqube

**Step 10:Configure SonarQube** Edit the SonarQube configuration file:

sudo nano /opt/sonarqube/conf/sonar.properties

Add the following lines:

arduino

sonar.jdbc.username=sonar

sonar.jdbc.password=sonar

sonar.jdbc.url=jdbc:postgresql://localhost/sonarqube

**Step 11:Start SonarQube**

cd /opt/sonarqube/bin/linux-x86-64/

./sonar.sh start

**Step 12:Access the SonarQube Dashboard** Open your browser and go to http://<VM-IP>:9000. The default credentials are:

* Username: admin
* Password: admin

**Step 13:Sonar-Scanner Installation**

sudo wget https://binaries.sonarsource.com/Distribution/sonar-scanner-cli/sonar-scanner-cli-5.0.1.3006-linux.zip

Unzip it

**Step 14:unzip** sonar-scanner-cli-5.0.1.3006-linux.zip -d /opt

**Step 15: Export path**

echo "export PATH=$PATH:/opt/sonar-scanner-5.0.1.3006-linux/bin" >> ~/.bashrc  
echo "export SONAR\_SCANNER\_HOME=/opt/sonar-scanner-5.0.1.3006-linux" >> ~/.bashrc  
source ~/.bashrc

Run the scanner

sonar-scanner -v

**Step 15: Set Up a Python Project in SonarQube**

Log in to SonarQube:

Access the web interface of your running SonarQube instance.

Create a New Project:

Click on Projects in the top menu.

Select "Create Project"

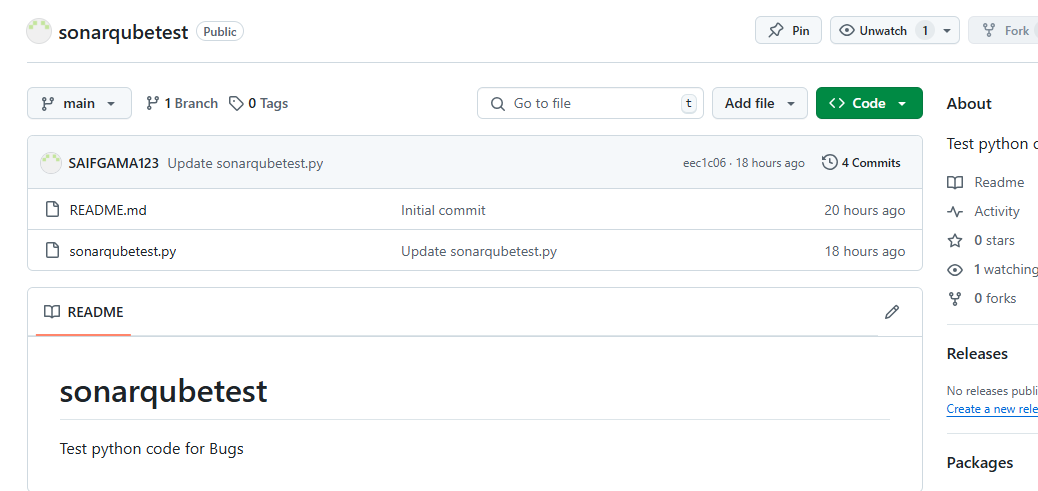
Provide a name (sonarqubetest)and a unique key for your Python project.

Choose Analysis Method: Manual

Select Manual or GitHub Integration.

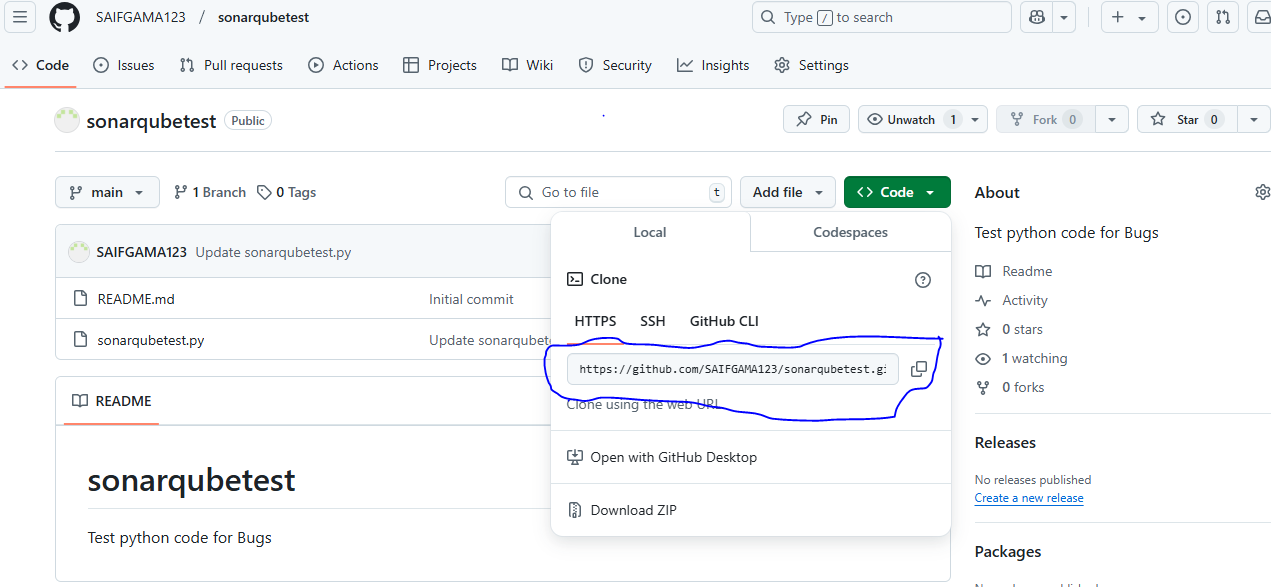
**Step16: Write the code in Visual Studio and save the file in any programming language let’s consider sonarqubetest.py**

**Step17: Create a GitHub Repository with the same name sonarqubetest and upload the python file there**



**Step18 : Clone that repository to your device or VM**

Using – git clone https://github.com/SAIFGAMA123/sonarqubetest.git



**Step 19: Run the Sonarqube Scanner**

Run the SonarScanner command with the token using the -Dsonar.login=<your\_token> option:

sonar-scanner \

-Dsonar.projectKey=<your\_project\_key> \

-Dsonar.sources=. \

-Dsonar.host.url=http://<your\_sonarqube\_url> \

-Dsonar.login=<your\_token>

Example :-

sonar-scanner\

-Dsonar.projectKey=sonarqubetest

-Dsonar.sources=. \

-Dsonar.host.url=http://192.168.1.91:9000

-Dsonar.login=sqp\_ef3508ad7010367c632cd3b57bfa1b77e0ce2baf

