SonarQube Integration with Jenkins

1. Prerequisites :

- Launch amazon linux instance with Size: **t2.medium or higher** (SonarQube needs 2 vCPU + 4GB RAM minimum)

  - Storage 20GB

- Open Security Group Ports:

  - 9000 port for SonarQube

2. Steps :

- sudo yum update -y
- sudo yum install java-17 -y
- wget https://binaries.sonarsource.com/Distribution/sonarqube/sonarqube-10.5.1.90531.zip
- unzip sonarqube-10.5.1.90531.zip
- sudo mv sonarqube-10.5.1.90531 /opt/sonarqube

3. Creating sonarqube user :

sudo adduser sonar

sudo chown -R sonar:sonar /opt/sonarqube

4. Edit the service file :

sudo nano /etc/systemd/system/sonar.service

Add : below content

[Unit]

Description=SonarQube service

After=syslog.target network.target

[Service]

Type=forking

ExecStart=/opt/sonarqube/bin/linux-x86-64/sonar.sh start

ExecStop=/opt/sonarqube/bin/linux-x86-64/sonar.sh stop

User=sonar

Group=sonar

Restart=always

LimitNOFILE=65536

LimitNPROC=4096

[Install]

WantedBy=multi-user.target

5. Starting service :

sudo systemctl daemon-reexec

sudo systemctl enable sonar

sudo systemctl start sonar

sudo systemctl status sonar

6. Verify SonarQube :

- Open browser: `http://<SonarQube-EC2-PUBLIC-IP>:9000`

- Login → `admin pwd --- admin`

- Generate a **token** for Jenkins:

  SonarQube UI → My Account → Security → Generate Token

7. Install Jenkins
8. Install plugins in Jenkins :

- In Jenkins UI → **Manage Jenkins → Plugins → Available plugins**

  Install : **SonarQube Scanner**

9. Configure SonarQube :

  Manage Jenkins → Credentials → Add Credentials

  Kind: **Secret Text**

Secret: <SONARQUBE_TOKEN_>

ID: **sonar-token**

➔ **Manage Jenkins → System → SonarQube servers**

- Add new server:
  - o Name: `sonarqube`
  - o Server URL: `http://<SonarQube-EC2-PUBLIC-IP>:9000`
  - o Credentials: `sonar-token`

10. Configure SonarQube Scanner :

Go to **Manage Jenkins → Global Tool Configuration**

Under **SonarQube Scanner**, add:

- Name: `sonar-scanner`
- Install automatically

11. Jenkins Pipeline :

```
pipeline {

  agent any

  tools {

    maven 'Maven3'

    jdk 'Java17'

  }

  stages {

    stage('Checkout') {

      steps {

        git 'https://github.com/your-org/your-repo.git'

      }

    }

    stage('Build') {

      steps {

        sh 'mvn clean install'
```

```
            }
        }
        stage('SonarQube Analysis') {
            environment {
                scannerHome = tool 'sonar-scanner'
            }
            steps {
                withSonarQubeEnv('sonarqube') {
                    sh "${scannerHome}/bin/sonar-scanner \
                        -Dsonar.projectKey=my-project \
                        -Dsonar.sources=src \
                        -Dsonar.java.binaries=target/classes"
                }
            }
        }
        stage("Quality Gate") {
            steps {
                timeout(time: 1, unit: 'HOURS') {
                    waitForQualityGate abortPipeline: true
                }
            }
        }
    }
}
```