**Installation of Jenkins plugins**

* [Eclipse Temurin installerVersion1.5](https://plugins.jenkins.io/adoptopenjdk)
* [SonarQube ScannerVersion2.16.1](https://plugins.jenkins.io/sonar)
* [Sonar Quality GatesVersion1.3.1](https://plugins.jenkins.io/sonar-quality-gates)
* [Quality GatesVersion2.5](https://plugins.jenkins.io/quality-gates)
* [NodeJS Version1.6.1](https://plugins.jenkins.io/nodejs)
* Docker Commons
* [Docker Pipeline](https://plugins.jenkins.io/docker-workflow)
* [Docker APIVersion3.3.4-86.](https://plugins.jenkins.io/docker-java-api)
* [Docker-build-stepVersion2.10](https://plugins.jenkins.io/docker-build-step)

**Configure tools in Jenkins -> Tools**

* Nodejs Installation
  + Name: nodejs16
  + Install from nodejs.org
  + v16.2.0
* JDK installation
  + Name: jdk17
  + Install from Adoptium.net (Eclipse)
  + v17.0.8.1+1
* Docker Installation
  + Name: docker
  + Install from docker.com
  + vLatest
* SonarQube installation
  + Name: sonar-scanner
  + Install automatically
  + Install from Maven central
  + v5.0.1.3006

**Configure SonarQube and integrate SonarQube with Jenkins**

* Generate sonar-Jenkins-token
  + SonarQube->Administration->security-> users
  + Name: sonar-jenkins-token
  + Token: squ\_07ca920f36da1b650dddf3e7ba0488b9a82e98cf
* Add sonar-Jenkins-token in Jenkins-> credentials
  + Kind: secret text
  + Secret text: squ\_07ca920f36da1b650dddf3e7ba0488b9a82e98cf
  + ID: SonarQube-Token
  + Create
* Add SonarQube server in Jenkins
  + Goto Manage Jenkins -> systems
  + SonarQube installation
  + Name: SonarQube-Server
  + <http://sq-private-ip:9000>
  + Server auth token: SonarQube-Token
* Create quality gates in SonarQube
  + Goto SonarQube dashboard
  + Quality Gates
  + Create
  + Name: SonarQube-Quality-Gates
  + Done
* Create webhook between Jenkins and SonarQube
  + Goto SonarQube dashboard
  + Administration -> Configuration-> Webhooks-> Create
  + Name: jenkins
  + URL: http://jenkins-private-IP:8080/sonarqube-webhook/
  + Create

**Create Jenkins pipeline to build and push docker image to Docker hub**