**Static Code Analysis using SonarQube:**

**What is SonarQube:** SonarQube collects and analyses source code, measuring quality and providing reports for your projects. It combines static and dynamic analysis tools and enables quality to be measured continuously over time.  Everything that affects our code base, from minor styling details to critical design errors, is inspected and evaluated by SonarQube, thereby enabling developers to access and track code analysis data ranging from styling errors, potential bugs, and code defects to design inefficiencies, code duplication, lack of test coverage, and excess complexity. The Sonar platform analyses source code from different aspects and hence it drills down to your code layer by layer, moving from the module level down to the class level. At each level, SonarQube produces metric values and statistics, revealing problematic areas in the source that require inspection or improvement.

**Why SonarQube?**

* As of now, CI tools does not have a plugin which would make all these plays together
* As of now, CI tools does not have plugins to provide nice drill-down features as SonarQube does
* CI plugins does not talk about overall compliance value
* CI plugins does not provide managerial perspective
* As of now there is no CI plugin for Design/Architecture issues
* It does not provide a dashboard for overall projects quality

**What is provides?**

1. Whether the coding has been done following a specific convention?
2. Whether well-known/established good practices have been followed and well-known/established bad practices have been avoided?
3. Are there any potential bugs and performance issues, security vulnerabilities?
4. Is there any duplicate code?
5. Is the code logic very complex?
6. Whether the public API has good documentation and comments?
7. Whether the code has unit tests?
8. Whether the code follows good design and architecture principles?

# **Static Code Analysis:**

Static code analysis is a collection of algorithms and techniques used to analyse source code in order to automatically find potential errors or poor coding practices.

1. Install plugins: **Sonar Gerrit** , **SonarQube Scanner** and **SonarQube Scanner for Jenkins**
2. Create **t2 medium ec2 linux** instance and install docker (minimum of 2GB of RAM, that’s why we can install sonar on t2.micro)

yum install -y docker

service docker start

systemctl enable docker

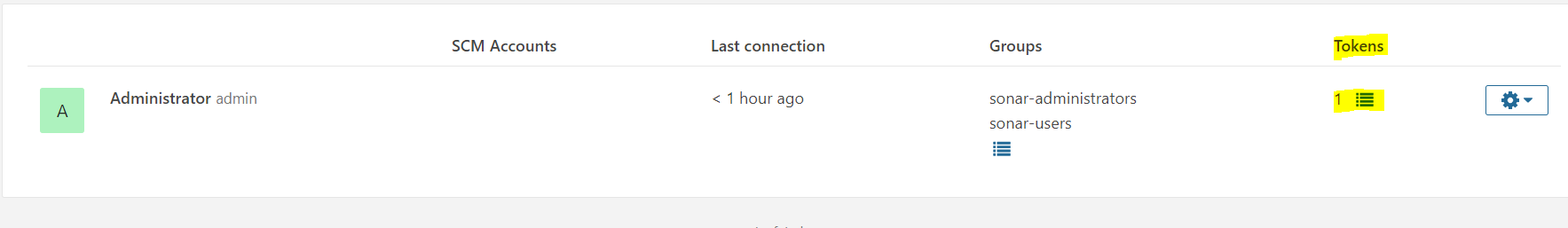
docker run -d --name sonarqube -p 9000:9000 sonarqube

1. **login to sonarqube (user: admin password: admin)**

Access SonarQube Dashboard:

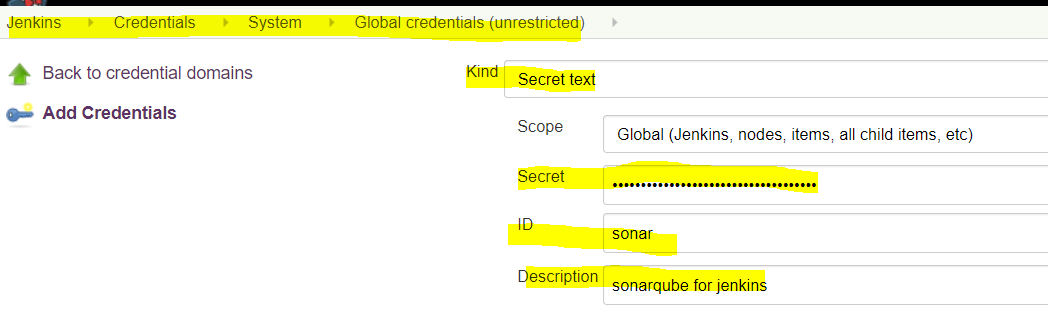
<Sonar instance public IP>:9000

**Administrator->Security->User->Token->Generate Token**



1. **Add Credential in Jenkins for SonarQube.**

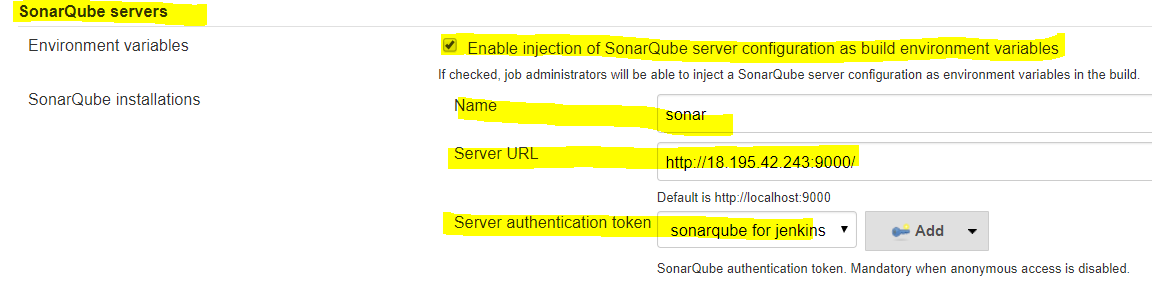
**Jenkins dashboard->Credentials-**>**Jenkins** under store scoped to Jenkins->**Global Credentials**->**Add Credentials**-> from dropdown select **Secret Text**->paste previously generated **sonar token**->Under ID section put as **sonar** -> under description put **sonarqube for Jenkins**



1. **Integrate Jenkins and SonarQube**

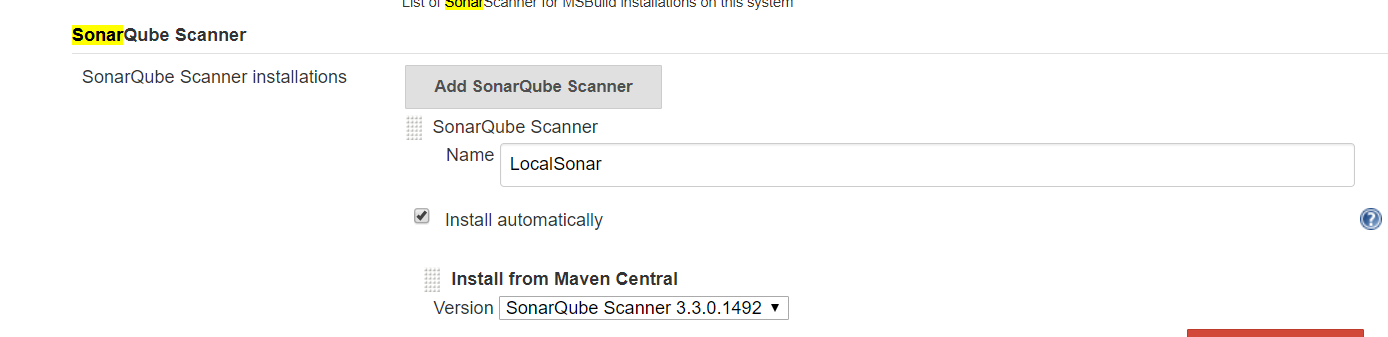
Jenkins dashboard->Manage Jenkins->Configure System:

Check for SonarQube servers "enjection option" and put name as "sonar”, sonar url and select server authentication token which you generated in previous step.



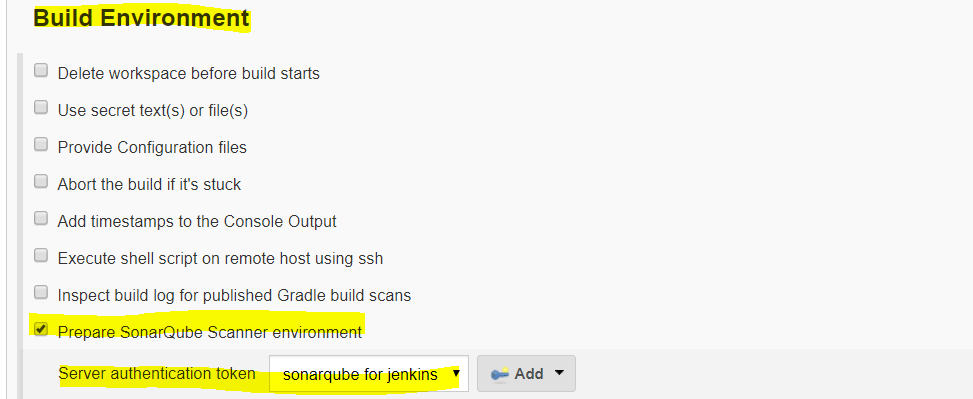
1. **Configure SonarQube for Jenkins job**

**Manage Jenkins->Global tool Configuration->Sonarqube Scanner – add sonar**



Create Jenkins Job:

In Build Environment section: Select Prepare SonarQube Scanner:



In Build Section:



Build

Enjoy 😊

**Pipeline Job with SonarQube:**

Create Pipeline Job:

<https://github.com/prakashk0301/maven-project>

(Select Branch name: **ci-cd-pipeline-sonar)**

Build

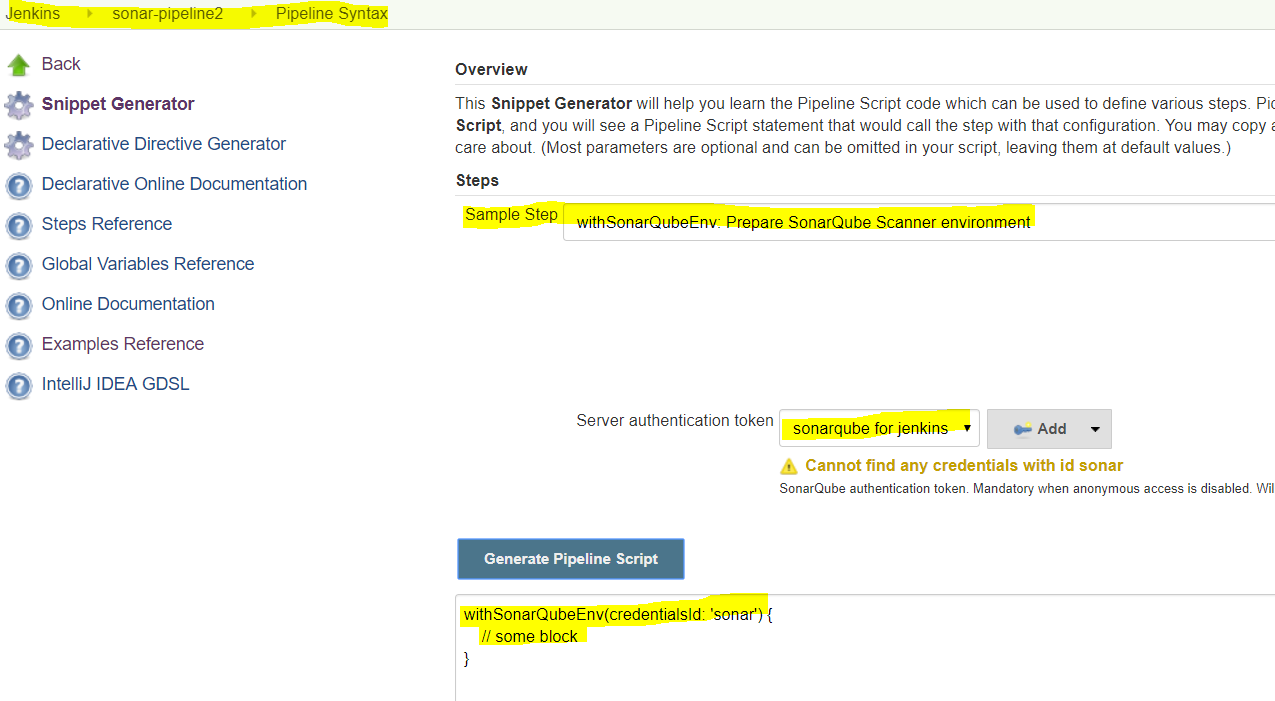
Done ☺

**How to generate Jenkins file script for Sonarqube:**

From **Pipeline Syntax**, look for **withSonarQubeEnv**

Choose your sonarqube **server authentication token**

Then generate Pipeline script.



Then add stage in Jenkins file.

stage ('build && SonarQube analysis') {

steps {

withSonarQubeEnv('sonar') {

withMaven(maven : 'LocalMaven') {

sh 'mvn clean package sonar:sonar'

} } } }

Some other projects for labs

<https://github.com/prakashk0301/sonarqubemaven>