# Experiment No. 07

**Practical Name :** Static Analysis SAST process and learn to integrate Jenkins SAST to SonarQube/GitLab.

**Aim**: To understand Static Analysis SAST process and learn to integrate Jenkins SAST to SonarQube.

# Theory :

* **What is SAST?**

Static Application Security Testing ([SAST](https://www.microfocus.com/products/static-code-analysis-sast/overview)) is a frequently used [Application](https://www.microfocus.com/products/application-security-testing/overview) [Security](https://www.microfocus.com/products/application-security-testing/overview) (AppSec) tool, which scans an application’s source, binary, or byte code. A white-box testing tool, it identifies the root cause of vulnerabilities and helps remediate the underlying security flaws. SAST solutions analyze an application from the “inside out” and do not reed a running system to perform a scan. Static application security testing (SAST), or static analysis, is a testing methodology that analyzes source code to find security vulnerabilities that make your organization’s applications susceptible to attack. SAST scans an application before the code is compiled. It’s also known as white box testing.

# What is Jenkins?

Jenkins is a free and open source automation server. It helps automate the parts of software development related to building, testing, and deploying, facilitating continuous integration and continuous delivery. It is a server-based system that runs in servlet containers such as Apache Tomcat. It supports version control tools, including AccuRev, CVS, Subversion, Git, Mercurial, Perforce, ClearCase and RTC, and can execute Apache Ant, Apache Maven and sbt based projects as well as arbitrary shell scripts and Windows batch commands. Jenkins is a popular open source tool to perform continuous integration and build automation. Jenkins allows to execute a predefined list of steps, e.g. to compile Java source code and build a JAR from the resulting classes. The trigger for this execution can be time or event based.

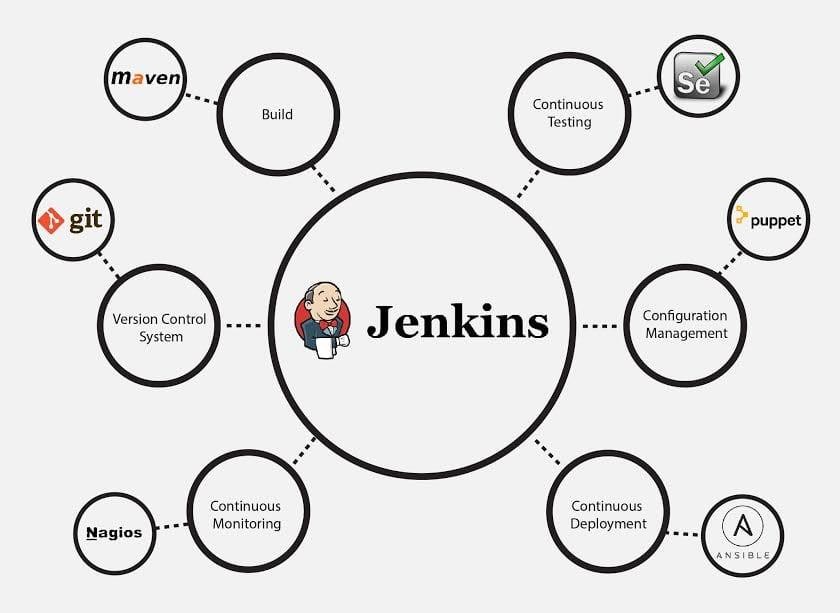
For example, you can compile your Java based application every 20 minutes or after a new

commit in the related Git repository.

Possible steps executed by Jenkins are for example:

* perform a software build using a build system like Apache Maven or Gradle
* execute a shell script
* archive a build result
* run software tests

Jenkins monitors the execution of the steps and allows to stop the process, if one of the steps fails. Jenkins can also send out notifications in case of a build success or failure. Jenkins can be extended by additional plug-ins. For example, you can install plug-ins to support building and testing Android applications.



# What is SonarQube?

SonarQube is a Code Quality Assurance tool that collects and analyzes source code, and provides reports for the code quality of your project. It combines static and dynamic analysis tools and enables quality to be measured continually over time. Everything from minor styling choices, to design errors are inspected and evaluated by SonarQube. This provides users with a rich searchable history of the code to analyze where the code is messing up and determine whether or not it is styling issues, code defeats, code duplication, lack of test coverage, or excessively complex code. The software will analyze source code from different aspects and drills down the code layer by layer, moving module level down to the class level, with each level producing metric values and statistics that should reveal problematic areas in the source code that needs improvement.

Sonarqube also ensures code reliability, Application security, and reduces technical debt by making your code base clean and maintainable. Sonarqube also provides support for 27 different languages, including C, C++, Java, Javascript, PHP, GO, Python, and much more.SonarQube also provides Ci/CD integration, and gives feedback during code review with branch analysis and pull request decoration.

# Benefits of SonarQube :

* Sustainability –

Reduces complexity, possible vulnerabilities, and code duplications, optimising the life of applications.

* Increase productivity –

Reduces the scale, cost of maintenance, and risk of the application; as such, it removes the need to spend more time changing the code

* Quality code –

Code quality control is an inseparable part of the process of software development.

* Detect Errors –

Detects errors in the code and alerts developers to fix them automatically before submitting them for output.

* Increase consistency –

Determines where the code criteria are breached and enhances the quality

* Business scaling –

No restriction on the number of projects to be evaluated

* Enhance developer skills –

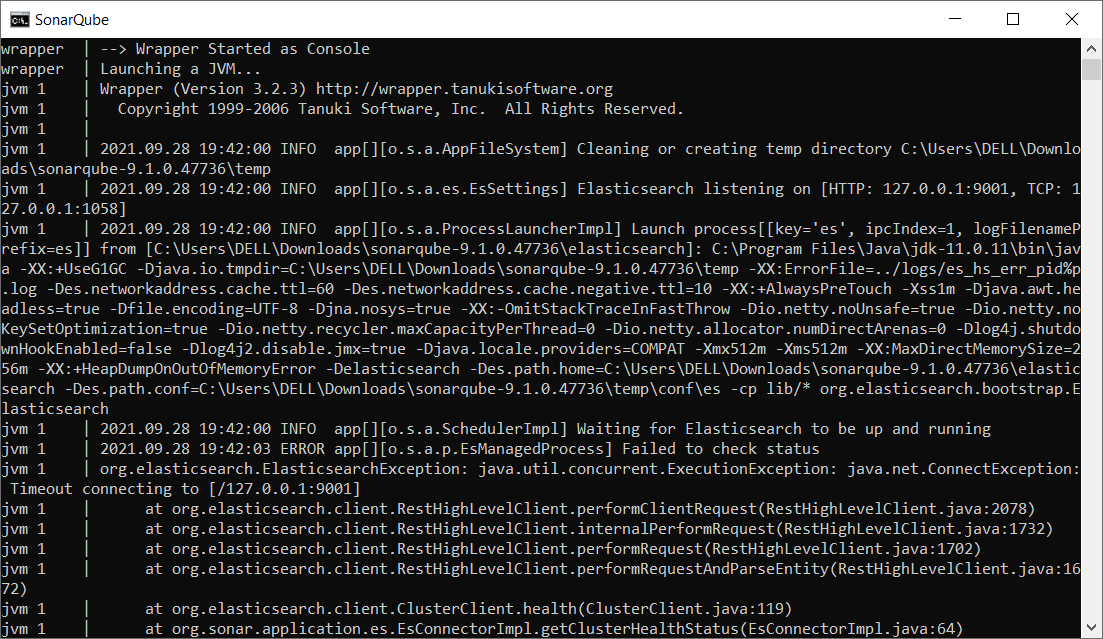
Regular feedback on quality problems helps developers to improve their coding skills

# Steps to install SonarQube on windows :

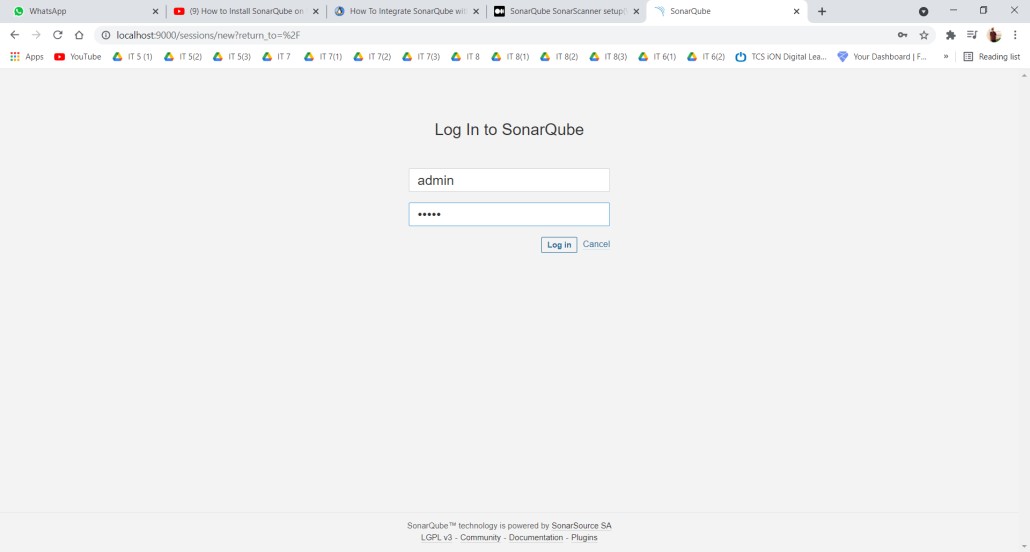
Step 1. Download Community edition from https:/[/www.sonarqube.org/downloads/](http://www.sonarqube.org/downloads/) Step 2. Extract .zip file and navigate to bin folder

Step 3. Based on your machine configuration go to folder for 32-bit OS move to windows-x86– 32 and for 64-bit OS move to windows-x86–64

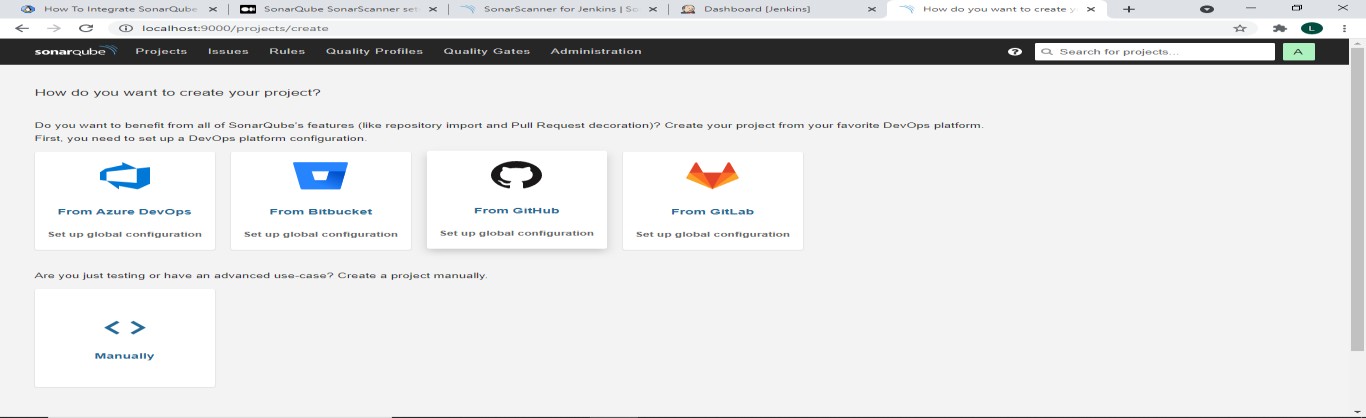
Step 4. Run StartSonar.bat and after few minute it will start your SonarQube server.



Step 5. Open browser and http://localhost:9000/ (9000 is default) you will be navigated to below window

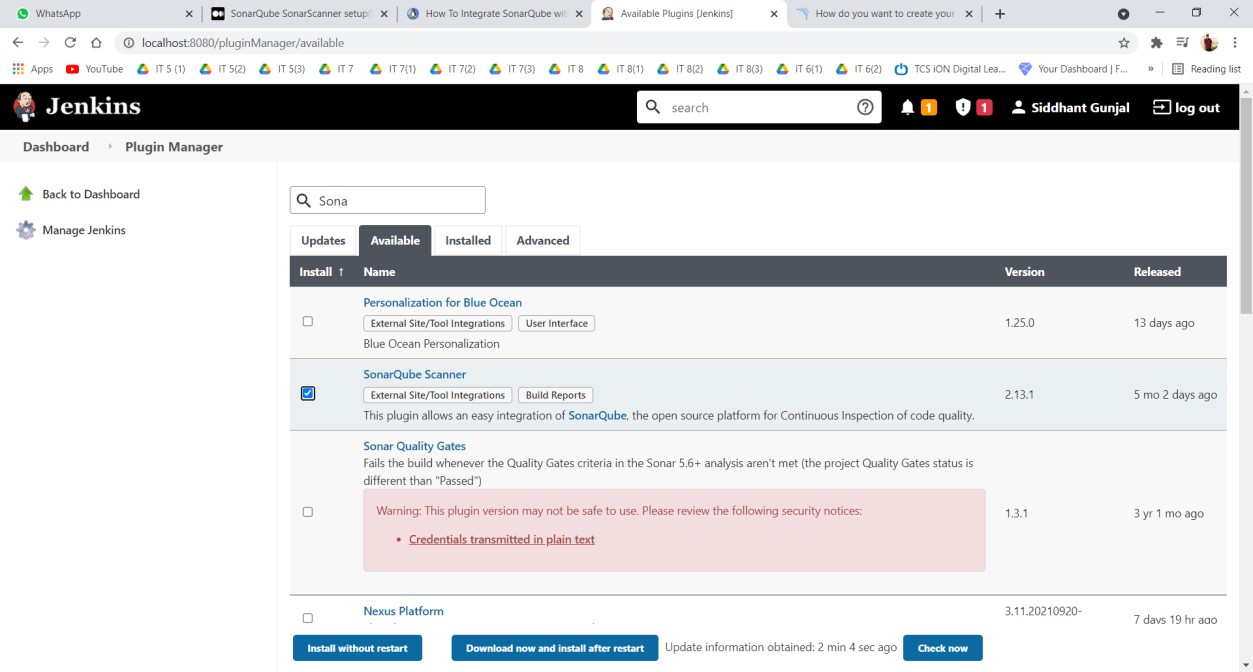
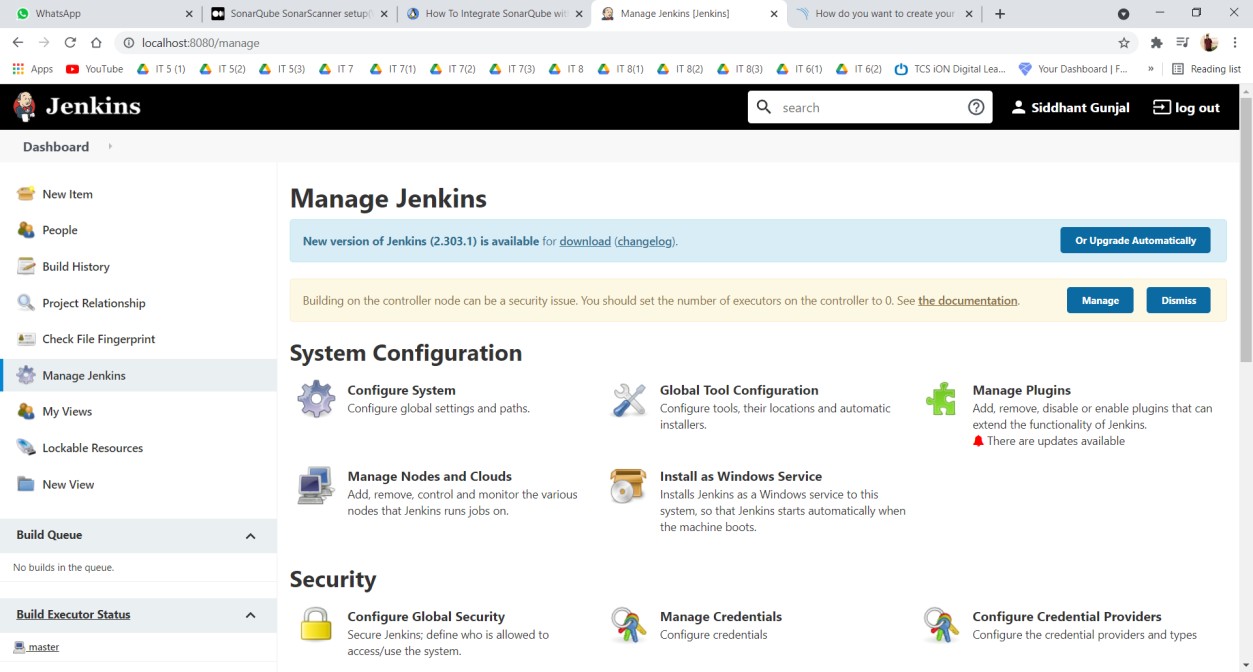


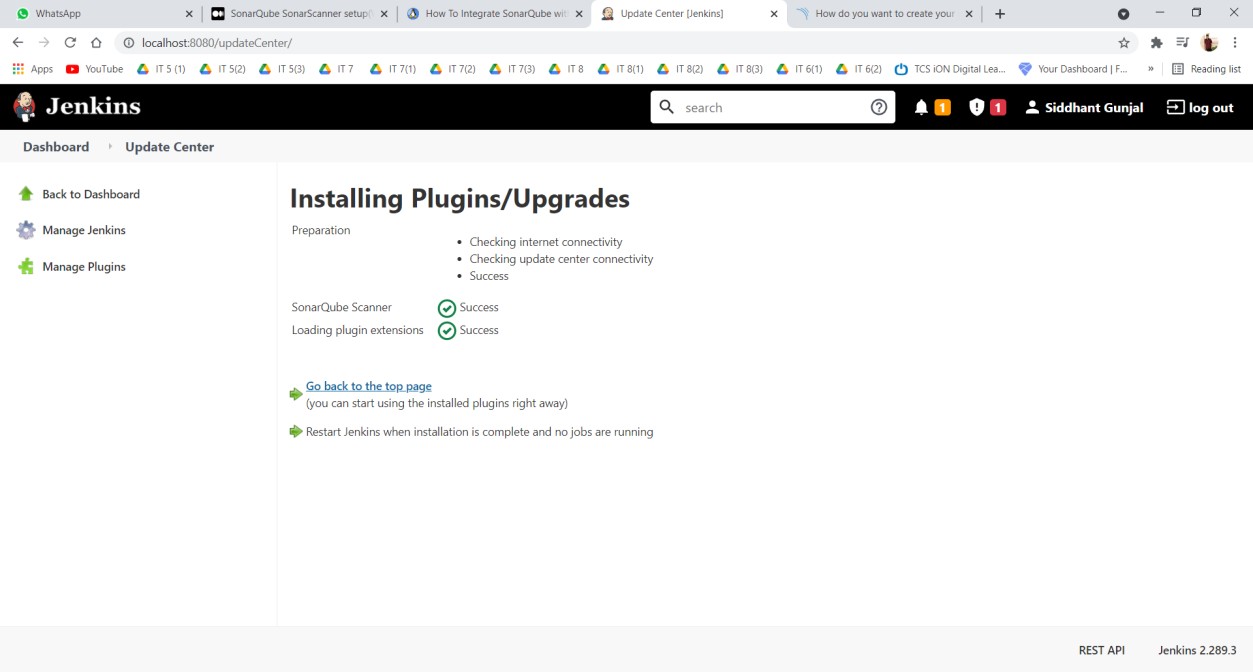
Default Login and Password is admin.



# Steps to integrate SonarQube with Jenkins.

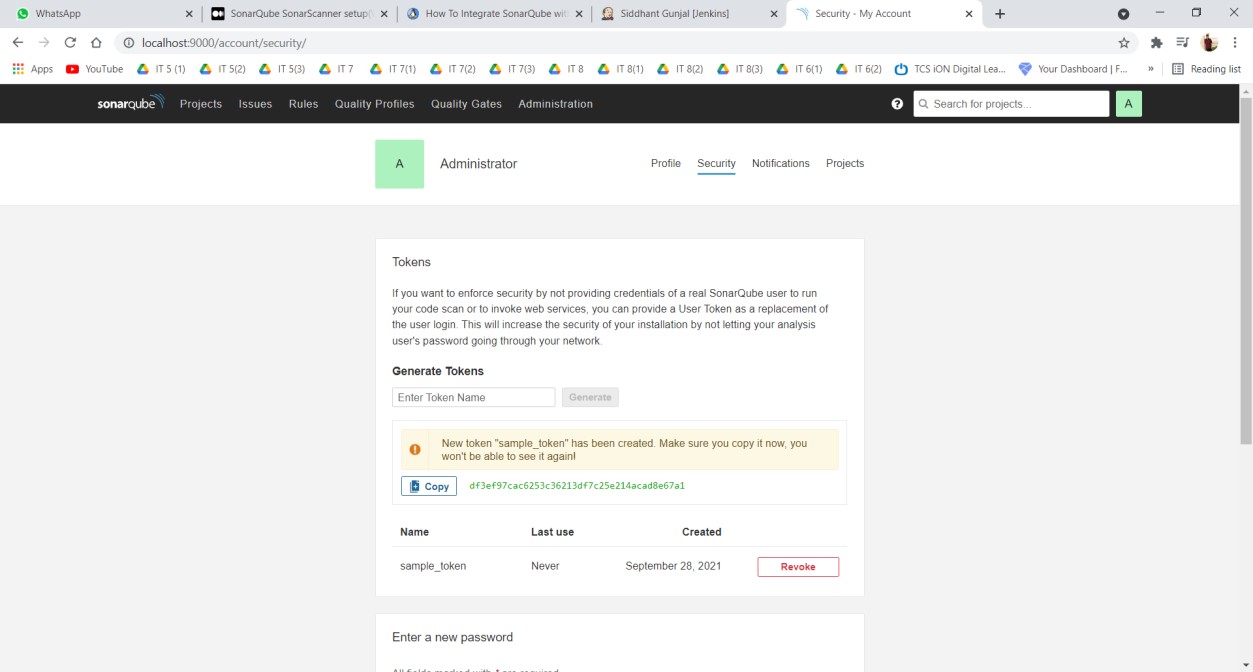
Step 1 : The first step to integrate the sonarqube installation with jenkins devops environment is to generate an access token.





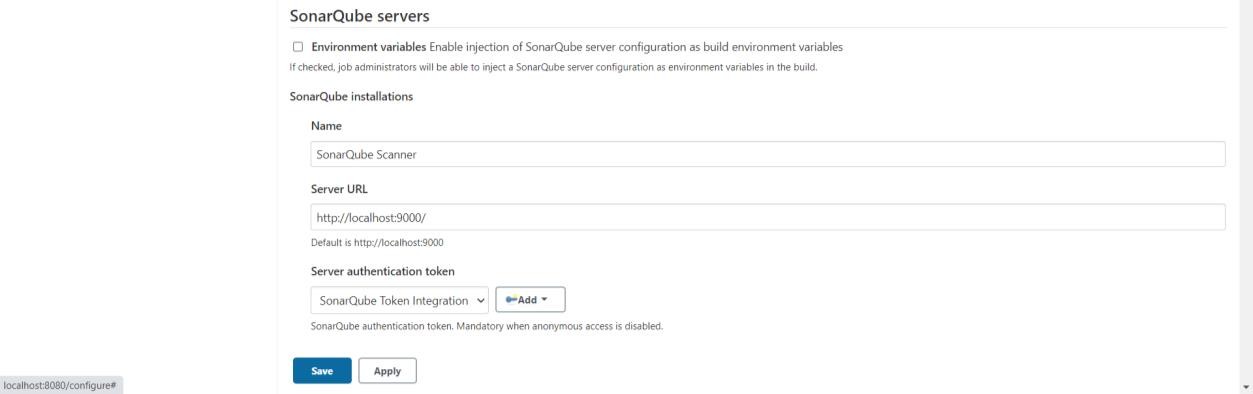
Go to Administration > Security > Users > Tokens > Generate token with some name > Copy the token

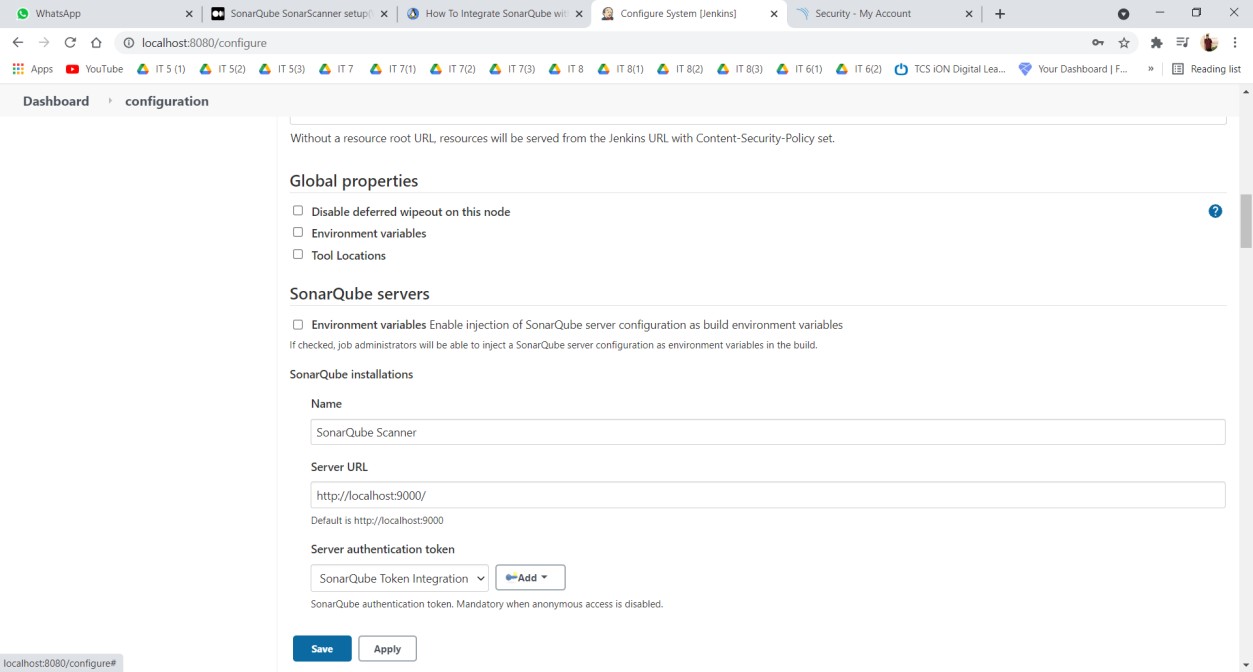
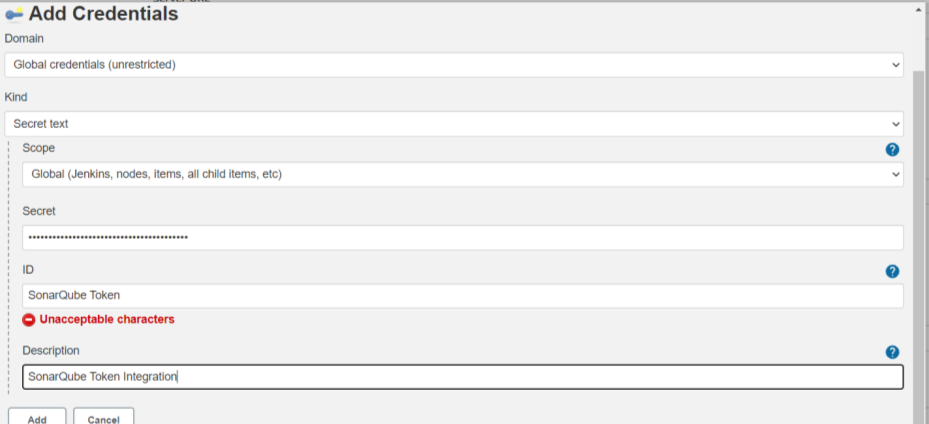
This token will be used in Jenkins for Sonar authentication



Step 2 : Then we need to configure the sonarqube installations with jenkins by using the generated access tokens. The steps to be followed in order to integrate with jenkins are,

Go to Manage Jenkins> Go to SonarQube servers section> Add SonarQube> Put a proper name(Your own choice)> Put the server URL as http://localhost:9000 if the server is running in the same server or if you install it as a separate server or if your running it on a different port you can put the respective Server URL> Click add> And select secret text> Add the generated token as secret key and save.





# Conclusion :

Hence, understood Static Analysis SAST process and successfully integrated Jenkins with SonarQube.