DEVOPS TOOLCHAIN IMPLEMENTATION

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI - WORK INTEGRATED LEARNING PROGRAM DIVISION

Asst. Prof. Sonika Rathi
sonika.rathi@pilani.bits-pilani.ac.in

TABLE OF CONTENTS

ENVIRONMENT SETUP

MAVEN ENVIRONMENT SETUP

PREREQUISITE:

JDK	1.7 or above.
Memory	No minimum requirement.
Disk Space	No minimum requirement.
Operating System	No minimum requirement.

VERIFY JAVA INSTALLATION:

OS	Task	Command
Windows	Open Command Console	c:\> java -version
Linux	Open Command Terminal	$ java -version
Mac	Open Terminal	$ java -version

SET JAVA ENVIRONMENT:

Set the **JAVA_HOME** environment variable to point to the base directory location where Java is installed on your machine

OS	Output
Windows	Set the environment variable JAVA_HOME= [%JAVA_HOME%] *Note the path should contain location of JDK installer at your local system* Example JAVA_HOME=C:\Program Files\Java\jdk1.7.0_60
Linux	export JAVA_HOME=/usr/local/java-current
Mac	export JAVA_HOME=/Library/Java/Home

Append Java compiler location to System variable Path,

Windows : Append the string ";C:\Program Files\Java\jdk1.7.0.60\bin" to the end of the system variable, Path
Linux: export PATH=$PATH:$JAVA_HOME/bin/
Mac: It is not required

DOWNLOAD MAVEN:

Download Maven latest version from https://maven.apache.org/download.cgi

OS	Archive name
Windows	apache-maven-3.6.3-bin.zip
Linux	apache-maven-3.6.3-bin.tar.gz
Mac	apache-maven-3.6.3-bin.tar.gz

EXTRACT THE MAVEN ARCHIVE & SET MAVEN ENVIRONMENT VARIABLE:

Add M2_HOME (Your Preferable Term) to environment variables.

OS	Output
Windows	Set the environment variables using system properties. M2_HOME= [%M2_HOME %] *Note the path should contain location of Maven installer at your local system* Example: M2_HOME=C:\Program Files\Apache Software Foundation\apache-maven-3.3.1
Linux	Open command terminal and set environment variables. Export M2_HOME=/usr/local/apache-maven/apache-maven-3.3.1
Mac	Open command terminal and set environment variables. export M2_HOME=/usr/local/apache-maven/apache-maven-3.3.1

Add Maven bin directory location to System Path:

Windows : Append the string ";C:\Program Files\Apache Software Foundation\apache-maven-3.3.1\bin" to the end of the system variable, Path
Linux: export PATH=$PATH:$M2_HOME/bin/
Mac: export PATH=$PATH:$M2_HOME/bin/

GRADLE ENVIRONMENT SETUP

DOWNLOAD GRADLE:

Download the latest version of Gradle from https://gradle.org/install/
In the reference page, click on the Complete Distribution link. This step is common for any platform. After this you will get the complete distribution file into your Downloads folder.

EXTRACT THE GRADLE ARCHIVE & SET ENVIRONMENT VARIABLE FOR GRADLE:

OS	Output
Windows	Extract the downloaded zip file named gradle-2.11-all.zip and copy the distribution files from Downloads\gradle-2.11\ to C:\gradle\ or your preferred location. Set the environment variables using system properties. GRADLE_HOME=C:\gradle
Linux & Mac	Extract the downloaded zip file named gradle-2.11-all.zip then you will find an extracted file named gradle-2.11 You can Move the distribution files from Downloads/gradle-2.11/ to /opt/gradle/ location or your preferred location export GRADLE_HOME = /opt/gradle

Add Gradle bin directory location to System Path:

Windows: Append the string ";C:\gradle\bin\" to the end of the system variable Path
Linux & Mac: $ export PATH=$PATH:/opt/gradle/gradle-6.8.3/bin

INSTALLING SELENIUM:

SELENIUM SETUP

Selenium setup is quite different from the setup of other commercial tools. To use Selenium in your automation project you need to install the language bindings libraries for your language of choice. In addition, you will need WebDriver binaries for the browsers you want to automate and run test on. Refer the link below for installing selenium libraries based on language of choice.

https://www.selenium.dev/documentation/en/selenium_installation/installing_selenium_libraries/

DOWNLOAD THE WEBDRIVER:

Selenium WebDriver refers to both the language bindings and the implementations of the individual browser controlling code.
Refer the link for installing selenium WebDriver based on browser

https://www.selenium.dev/downloads/

INSTALL SONAR

Navigate: https://www.sonarqube.org/downloads/
Download the Community Edition, Extract the zip folder to install. If the installed folder is customized then add the appropriate path to the PATH as environment variable

RUN SONAR

OS	Output
Windows	Execute the script bin/windows-x86-64/StartSonar.bat
Linux	Execute the script bin/linux-x86-64/sonar.sh start
Mac	Execute the script bin/macosx-universal-64/sonar.sh start

You can now browse SonarQube at http://localhost:9000 (the default System administrator credentials are admin/admin).

JENKINS INSTALLATION:

JENKINS SETUP STEPS:

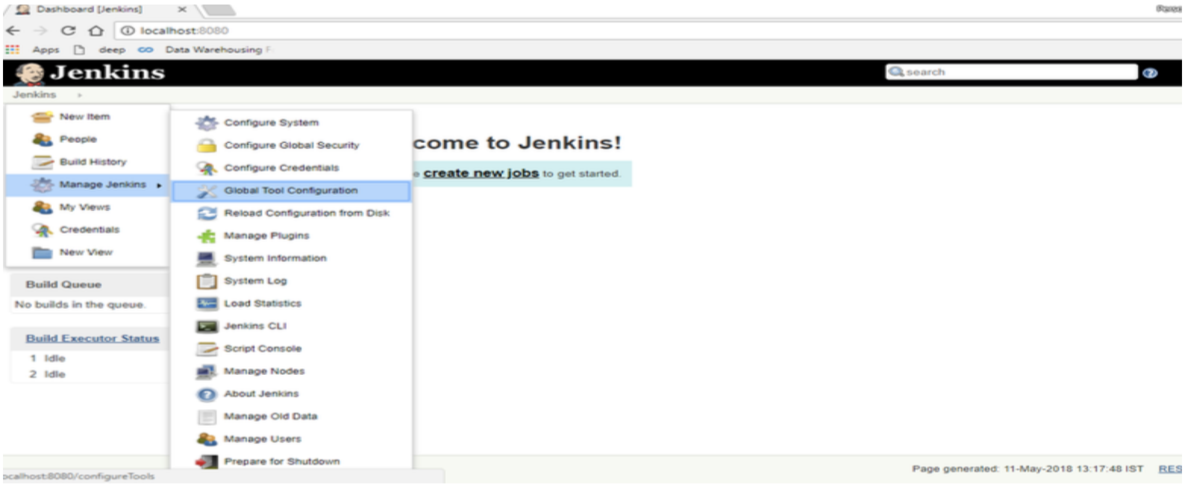
OS	Link
Windows	https://www.jenkins.io/doc/book/installing/windows/
Linux	https://www.jenkins.io/doc/book/installing/linux/
Mac	https://www.jenkins.io/doc/book/installing/macos/

SET UP FOR PLUGINS: INTEGRATION WITH JENKINS

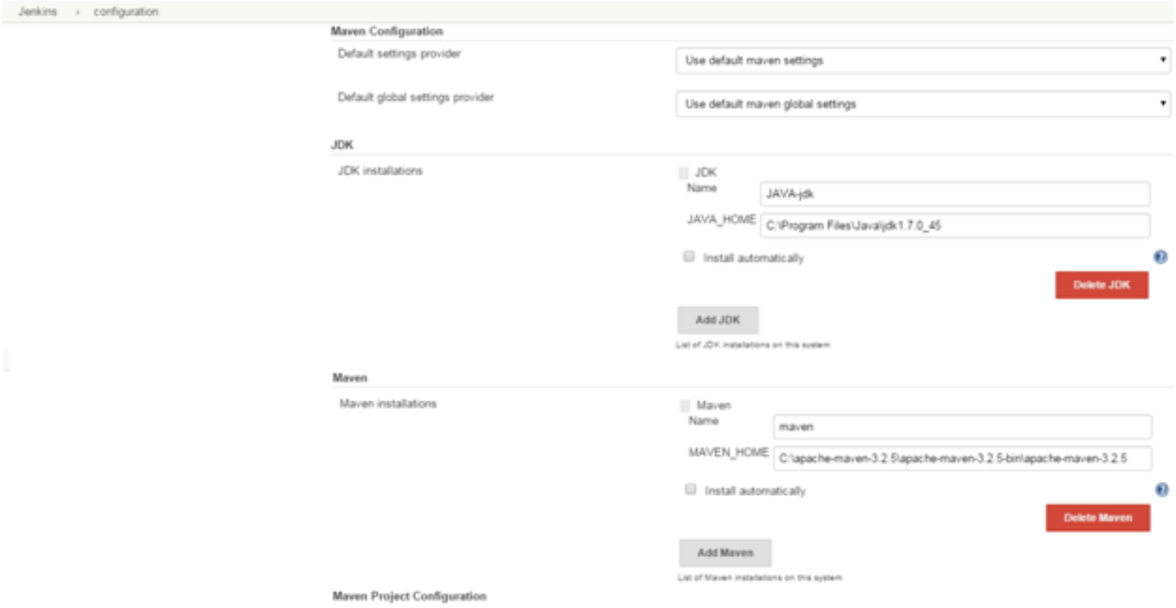
MAVEN INTEGRATION:

Steps:

1) Navigate: Jenkins-> Manage Jenkins -> Global Configuration ->



2) Configure JDK: Add JDK and provide the JDK name which you have given for system variable
3) Configure Maven: Click add Maven and provide Name and the installation path of the Maven in the appropriate fields



Maven Configuration

Default settings provider: Use default maven settings

Default global settings provider: Use default maven global settings

JDK

JDK installations

JDK

Name: JAVA-jdk

JAVA_HOME: C:\Program Files\Java\jdk1.7.0_45

□ Install automatically

List of JDK installations on this system

Maven

Maven installations

Maven

Name: maven

MAVEN_HOME: C:\apache-maven-3.2.5\apache-maven-3.2.5-bin\apache-maven-3.2.5

□ Install automatically

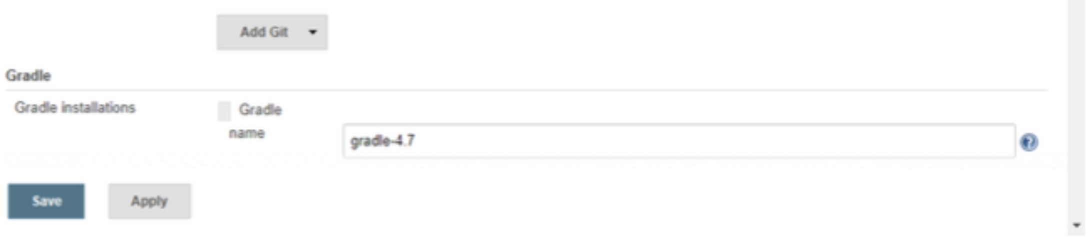
List of Maven installations on this system

Maven Project Configuration

GRADLE INTEGRATION:

Steps:

1) Navigate: Jenkins-> Manage Jenkins -> Global Configuration
2) Configure Gradle -> provide Name and the installation path of the Maven in the appropriate fields



Gradle

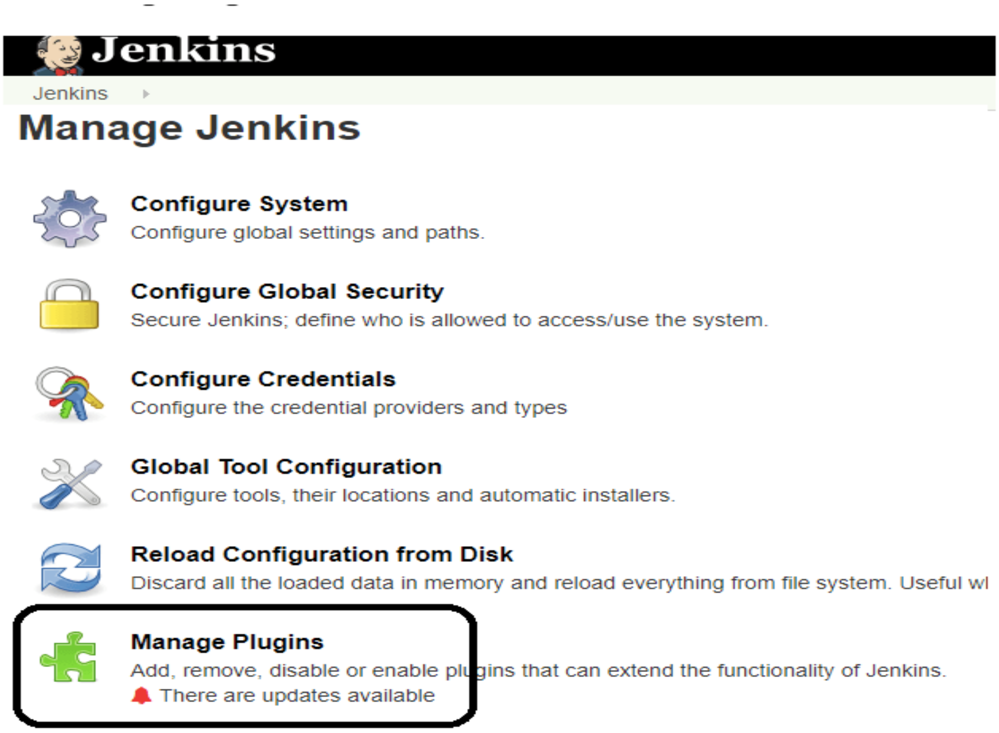
Gradle installations

Gradle name: gradle-4.7

GIT INTEGRATION:

Steps:

1) Navigate: Jenkins -> Manage Jenkins -> Manage Plugins



Jenkins

Jenkins ▸

Manage Jenkins

Configure System
Configure global settings and paths.

Configure Global Security
Secure Jenkins; define who is allowed to access/use the system.

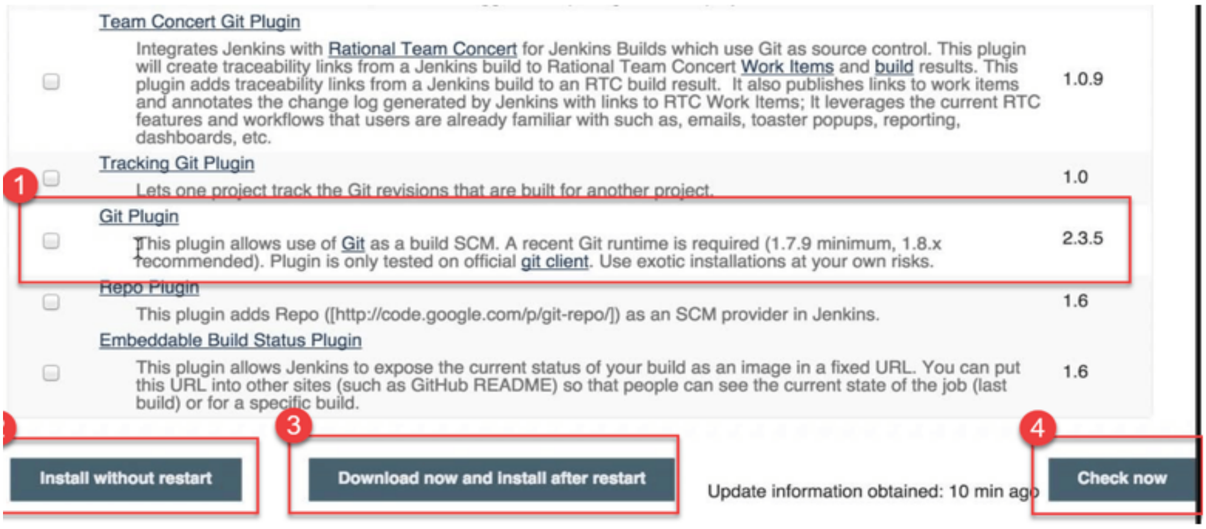
Configure Credentials
Configure the credential providers and types

Global Tool Configuration
Configure tools, their locations and automatic installers.

Reload Configuration from Disk
Discard all the loaded data in memory and reload everything from file system. Useful wl

Manage Plugins
Add, remove, disable or enable plugins that can extend the functionality of Jenkins.
🔔 There are updates available

2) Select the GIT Plugin
3) Click on Install without restart. The plugin will take a few moments to finish downloading depending on your internet connection, and will be installed automatically
4) You can also select the option Download now and Install after restart button. In which plugin is installed after restart
5) You will be shown a "No updates available" message if you already have the Git plugin installed



☐ Team Concert Git Plugin — 1.0.9
Integrates Jenkins with Rational Team Concert for Jenkins Builds which use Git as source control. This plugin will create traceability links from a Jenkins build to Rational Team Concert Work Items and build results. This plugin adds traceability links from a Jenkins build to an RTC build result. It also publishes links to work items and annotates the change log generated by Jenkins with links to RTC Work Items; It leverages the current RTC features and workflows that users are already familiar with such as, emails, toaster popups, reporting, dashboards, etc.

☐ Tracking Git Plugin — 1.0
Lets one project track the Git revisions that are built for another project.

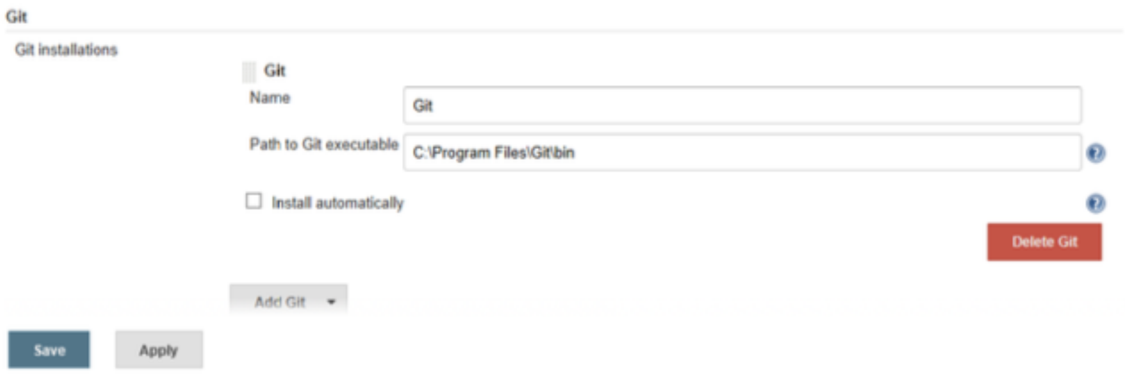
☐ Git Plugin — 2.3.5
This plugin allows use of Git as a build SCM. A recent Git runtime is required (1.7.9 minimum, 1.8.x recommended). Plugin is only tested on official git client. Use exotic installations at your own risks.

☐ Repo Plugin — 1.6
This plugin adds Repo ([http://code.google.com/p/git-repo/]) as an SCM provider in Jenkins.

☐ Embeddable Build Status Plugin — 1.6
This plugin allows Jenkins to expose the current status of your build as an image in a fixed URL. You can put this URL into other sites (such as GitHub README) so that people can see the current state of the job (last build) or for a specific build.

Update information obtained: 10 min ago

6) Navigate: Jenkins-> Manage Jenkins -> Global Configuration
7) Enter the name of the GIT installer and the path where its discoverable in your system under appropriate fields as given below:



Git

Git installations

Git

Name: Git

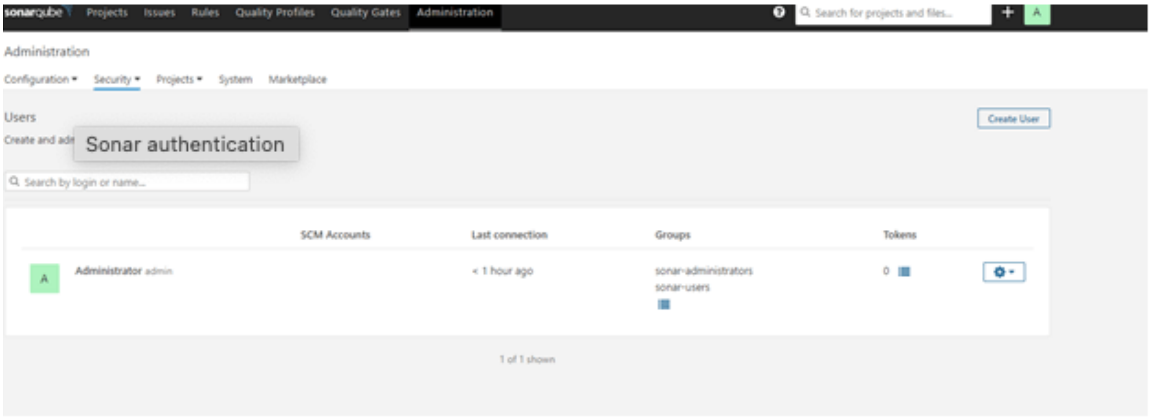
Path to Git executable: C:\Program Files\Git\bin

□ Install automatically

INTEGRATING SONARQUBE:

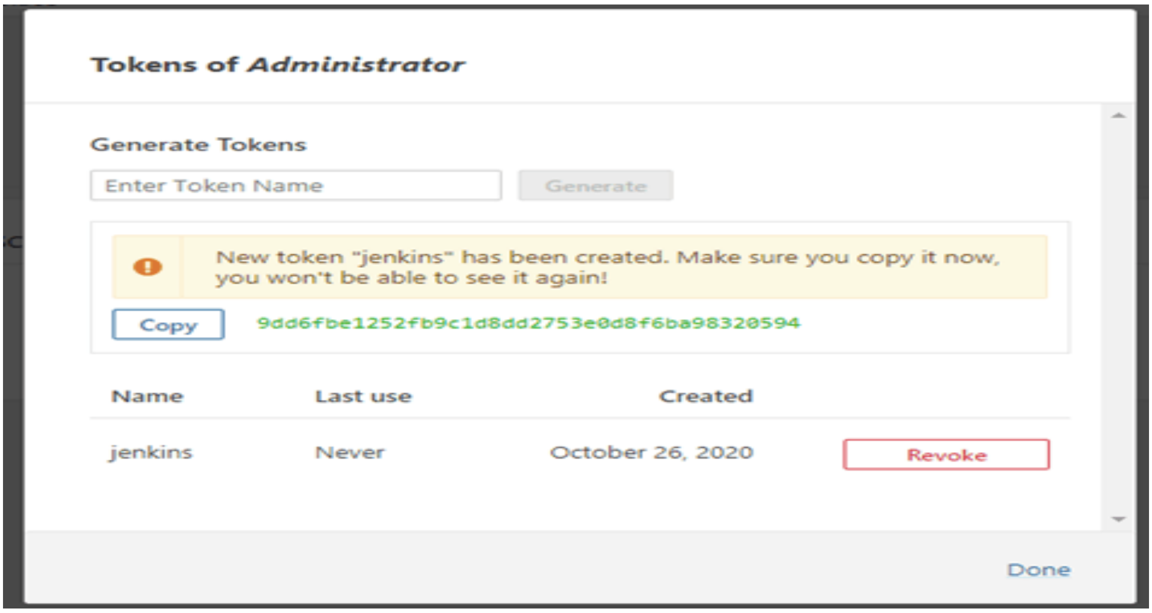
Steps:

1) To Integrate the Jenkins DevOps environment with SonarQube installation is to generate an access token.
2) Start the SonarQube server (See Run SonarQube step form set SonarQube environment)
3) Hit http://localhost:9000, and Login using username- admin, password – admin
4) Now generate token with an appropriate name, which is under Administration/Security/Users/Tokens
5) This generated token will be later used in Jenkins for Sonar authentication.



	SCM Accounts	Last connection	Groups	Tokens
A Administrator admin		< 1 hour ago	sonar-administrators sonar-users	0

1 of 1 shown



Tokens of *Administrator*

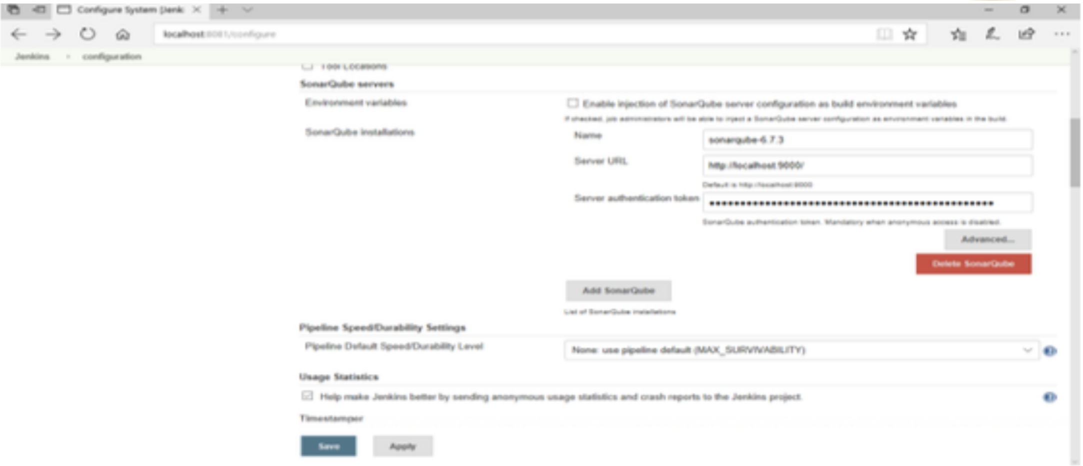
Generate Tokens

New token "jenkins" has been created. Make sure you copy it now, you won't be able to see it again!

9dd6fbe1252fb9c1d8dd2753e0d8f6ba98320594

Name	Last use	Created	
jenkins	Never	October 26, 2020	Revoke

6) Navigate: Manage Jenkins->Manage Plugins-> Available tab
7) Find the SonarQube plugin and install it
8) Go to Manage Jenkins-> Configure System->SonarQube servers section
9) Click on the Add SonarQube server. Provide Server URL and credentials. Get a Server authentication token**(Use here the generated key)** and provide it in Jenkins.



□ Tool Locations

SonarQube servers

Environment variables □ Enable injection of SonarQube server configuration as build environment variables

If checked, job administrators will be able to inject a SonarQube server configuration as environment variables in the build.

SonarQube installations

Name: sonarqube-6.7.3

Server URL: http://localhost:9000/

Default is http://localhost:9000

Server authentication token: ••

SonarQube authentication token. Mandatory when anonymous access is disabled.

List of SonarQube installations

Pipeline Speed/Durability Settings

Pipeline Default Speed/Durability Level: None: use pipeline default (MAX_SURVIVABILITY)

Usage Statistics

☑ Help make Jenkins better by sending anonymous usage statistics and crash reports to the Jenkins project.

Timestamper