Installation Guide

JWebSocket

Continuous Integration Environment for jWebSocket

V 1.0

1. Downloads process

Before to start to install and configure the continuous integrations environment is necessary download following software's.

- Apache Archiva (http://archiva.apache.org/)
- Jenkins (http://jenkins-ci.org/)
- Sonar (http://www.sonarsource.org/)

2. Environment installation features.

Before to start the installations you need to have the personal computer with Ubuntu Server 11.10 64 bits with the next software's installed.

- Apache2 server
- MySQL server
- Apache Tomcat6 server
- openjdk-7 virtual machine.

3. Installation process

1. Installing Subversion

Next you can see how to install Subversion server with your apache library and tools package, to do it is necessary execute this line at the console.

```
sudo apt-get install subversion libapache2-svn subversion-tools apache2-mpm-prefork
```

2. Installing Apache Archiva

Creating the ARCHIVA_HOME folder.

To install Apache Archiva is necessary create the ARCHIVA_HOME folder; there they are all Archiva installations files. This folder will be created at our home folder with the session started.

```
mkdir /home/user/ARCHIVA_HOME
```

The next step is to uncompressing the Archiva lasted version downloaded in this folder.

Finally we can start to the Archiva installations, to do it we need open the ARCHIVA_HOME and execute the following line at the console.

```
./archiva start
```

3.3 Installing maven2

To install maven2 we only need to execute the following line at the console.

```
apt-get install maven2
```

3.4 Installing Jenkins Server

Before to start the Jenkins server, we need to install some dependence like the daemon package, to do it we need to execute the following line at the console.

```
apt-get install daemon
```

Once installed the dependence we can start with the Jenkins installations. First is necessary open the directory where is the Jenkins install file and install it, to do it we need to execute this line at the console.

```
dpkg -i jenkins_1.448_all.deb
```

3.5 Installing Sonar Server

To install the Sonar server first we need to have the MySql server installed. To do it execute this line at the console:

```
Apt-get install mysql-server
```

After to install MySql server we need to create the sonar data base:

```
mysqladmin -p create sonar
```

Once create the data base, we can start with the sonar installations. The first step is download the lasted version and uncompressing it in some place that will be knower like SONAR_HOME.

After we need to edit the sonar configuration file (SONAR_HOME/conf/sonar.properties) and put there the data base configurations. This file is easy to configure, we only need to comment the three lines where is configured the database embedded.

```
#DATABASE
#Comment the 3 following lines to deactivate the default embedded database (used
only for tests and demos)
#sonar.jdbc.url:jdbc:derby://localhost:1527/sonar;create=true
#sonar.jdbc.driverClassName:org.apache.derby.jdbc.ClientDriver
#sonar.jdbc.validationQuery:values(1)
```

And uncomment the lines concerning to the MySql database.

```
#MySql
#uncomment the 3 following lines to use MySQL
sonar.jdbc.url: jdbc:mysql://localhost:3306/sonar?useUnicode=true&characterEncoding=utf8
sonar.jdbc.driverClassName: com.mysql.jdbc.Driver
sonar.jdbc.validationQuery: select 1
```

Then we need to put the configuration to connect with the sonar database above created. To do it, we need to find the “generic settings” and configured like this example:

```
generic settings
```

```
sonar.jdbc.username: root
sonar.jdbc.password: losmalos
```

Once setter the parameters needed to the configurations, we proceed to generate the war to deploy it at our local tomcat server. To generate the war is necessary open this directory SONAR_HOME/war and execute this line at the console.

```
./build-war.sh
```

This line generates a new .war file ready to deploy it in any Tomcat server. Now we only need to copy it in our webapps tomcat folder (/var/lib/tomcat6/webapps) and restart the Tomcat server, then we can open the sonar server at this URL http://server_ip:8080/sonar

4. Configurations options.

4.1 Configuring the Subversion Server

The first step to configure the Subversion server is to create the svn folder to locate this own repositories.

```
mkdir /home/user/svn
```

Later will create the repository into the folder created. To do it is we only need to execute this line at the console:

```
svnadmin create /home/user/svn/jwebsocket
```

Later will create the repository folder structure, branches, tags and trunk:

```
svn mkdir --message="Setting up the directories..."
file:///home/user/svn/jwebsocket/trunk
file:///home/user/svn/jwebsocket/tags
file:///home/user/svn/jwebsocket/branches
```

The next step is to put the www-data like the owner with the repository folder.

```
chown www-data:www-data /home/carlos/svn/jwebsocket/ -R
```

After we need to move the configurations file of /home/user/svn/jwebsocket.conf/authz to /home/user/svn/authz and edit it like this example:

```
/etc/apache2/mods-avaiable/dav_svn.conf
<Location /svn>
DAV svn
#Repository folder
SVNParentPath /home/carlos/svn/
#Authentication mode
#Name of the repository
AuthName "Subversion Repository jWebSocket"

AuthUserFile /etc/apache2/passwords
# AuthBasicProvider ldap
# AuthzLDAPAuthoritative on
# AuthLDAPURL "ldap://10.208.0.3:389/OU=Personas, DC=hab, DC=uci,
DC=cu?uid?sub?(objectClass=*)"
# AuthLDAPURL "ldap://10.0.0.3:389/DC=uci, DC=cu?uid?sub?(objectClass=*)"
Require valid-user
AuthzSVNAccessFile /home/user/svn/authz
</Location>
```

Now you can enter to the repository using this url
http://ip_server/svn/jwebsocket/.

Finally is necessary configuring the subversion hooks. This hook is to allow to subversion execute the Jenkins remote compiling. To do it is necessary edit these file /home/usuario/svn/jwebsocket/hooks/post-commit.tmpl and add this line at the file end.

```
# The first parameters is the SVN folder name and the second is the project Jenkins name
/home/usuario/svn/jwebsocket/jenikins-launch-build.sh $REPOS $REV SVNFolderName
JenkinsProjectName
```

Then we need to create the /home/usuario/svn/jwebsocket/jenikins-launch-build.sh file and put this source into this file.

```
#!/bin/bash
# This script is executed after any subversion change,
# and will notice the Jenkins server
REPOS="$1"
REV="$2"
PROJECT_NAME="$3"
JENKINS_JOB="$4"
JENKINS_USER=admin
JENKINS_PASSWORD=losmalos
JENKINS_HOST=10.208.7.201:8002
IS_PROJECT_CHANGED=`svnlook dirs-changed $REPOS --revision $REV | fgrep
$PROJECT_NAME`
if [[ -n $IS_PROJECT_CHANGED ]]; then
   wget --quiet --auth-no-challenge --no-check-certificate --http-user=$JENKINS_USER --http-
password=$JENKINS_PASSWORD
http://$JENKINS_HOST/job/$JENKINS_JOB/build?token=TOKEN
   exit 0
fi
```

4.2 Configuring the Apache Archiva

Change the Archiva port to 8888

Once installed the Archiva server it is running using the 8080 port, the same of Tomcat, for that, is necessary change it for other port. To do it open the ARCHIVA_HOME/conf/jetty.xml and go to line 66 and change the default port of 8080 to 8888, here you have an example to this configuration.

```
<Set name="port"><SystemProperty name="jetty.port" default="8888"/></Set>
```

4.3 Configuring maven2

To configure the maven2 is necessary edit the file /etc/maven2/settings.xml and change the mirrors. Following you can see an example for this file:

```
    <mirror>
          <id>archiva.default</id>
          <url>http://10.208.7.201:8888/archiva/repository/internal/</url>
          <mirrorOf>*</mirrorOf>
    </mirror>
    <mirror>
          <id>archiva.apache.snapshots</id>
          <url>http://10.208.7.201:8888/archiva/repository/snapshots/</url>
          <mirrorOf>apache.snapshots</mirrorOf>
    </mirror>
```

4.4 Configuring Jenkins Server

Once installed Jenkins it is running using the 8080 port by default, this is the same port to the Tomcat, for that is necessary change it to another port. To change the Jenkins port you need to edit the /etc/default/jenkins configuration file and change the default port of 8080 to 8002. Here you have an example:

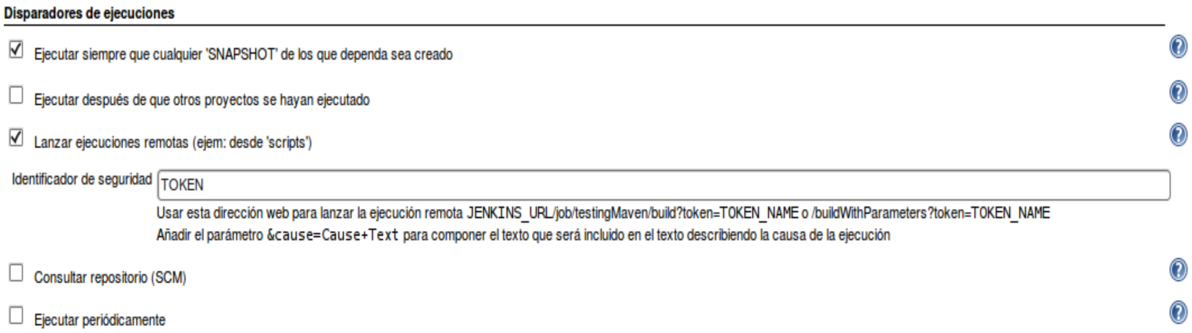
```
# port for HTTP connector (default 8080; disable with -1)
HTTP_PORT=8002
```

After you need to restart the Jenkins server:

```
/etc/init.d/jenkins restart
```

Now you can access to the Jenkins server using this url: http://ip_del_servidor:8002/.

The other step is active and configure the triggers for each Jenkins projects, this is for allows the remote executions. To do it is necessary go to the configurations session for each project and configure the Triggers like the following picture.



Disparadores de ejecuciones

☑ Ejecutar siempre que cualquier 'SNAPSHOT' de los que dependa sea creado

☐ Ejecutar después de que otros proyectos se hayan ejecutado

☑ Lanzar ejecuciones remotas (ejem: desde 'scripts')

Identificador de seguridad: TOKEN

Usar esta dirección web para lanzar la ejecución remota JENKINS_URL/job/testingMaven/build?token=TOKEN_NAME o /buildWithParameters?token=TOKEN_NAME

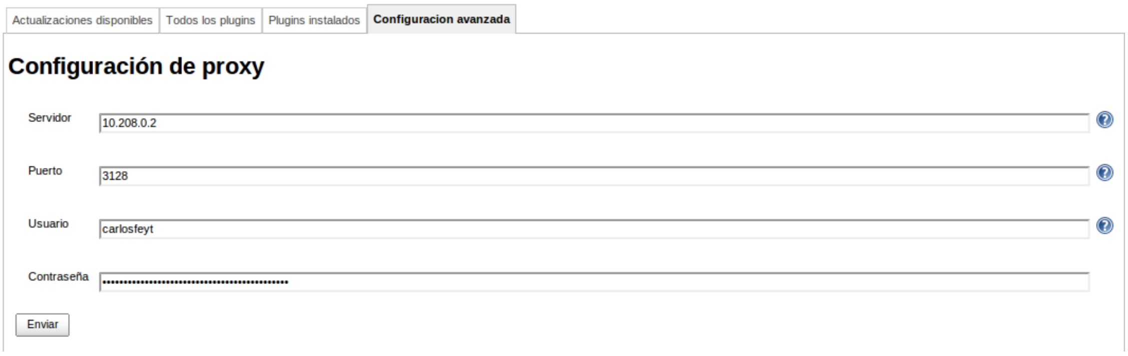
Añadir el parámetro &cause=Cause+Text para componer el texto que será incluido en el texto describiendo la causa de la ejecución

☐ Consultar repositorio (SCM)

☐ Ejecutar periódicamente

Picture 1.2 Example of trigger configuration in Jenkins projects.

Finally you can process to configure plug-in to integrate Sonar Server with Jenkins. To start is necessary configure the internet access in Jenkins, in this example we use the proxy server, to do it go to the administrator menu, plug-in settings, advance settings, then you can see the view to set the proxy configurations.



Configuración de proxy

Servidor: 10.208.0.2

Puerto: 3128

Usuario: carlosfeyt

Contraseña: •••••••••••••••••••••••••••••••••••••••

Picture 1.3 Example to proxy configuration in Jenkins.

Finally you can install the Sonar plug-in. To install it, go to the administration menu, plug-in settings, in the plug-ins tabs, find and select the sonar plug-in y press the “download new and install after restart” button.

5. Administration of the applications.

5.1 Subversion administration.

The first step of the subversion administration is the user creation. You can create a new Subversion user using the htpasswd command at the console. Only for the first time to create a user is use the –mc parameter, the other time only use the –c parameter. Now you can see an example to create the first user in Subversion.

```
htpasswd -mc /etc/apache2/passwords usuario
```

The other step is to set the security in subversion. To do it is necessary edit the /home/usuario/svn/authz file. One step to configure the security is creating the groups. Following you can see e example to create a new group.

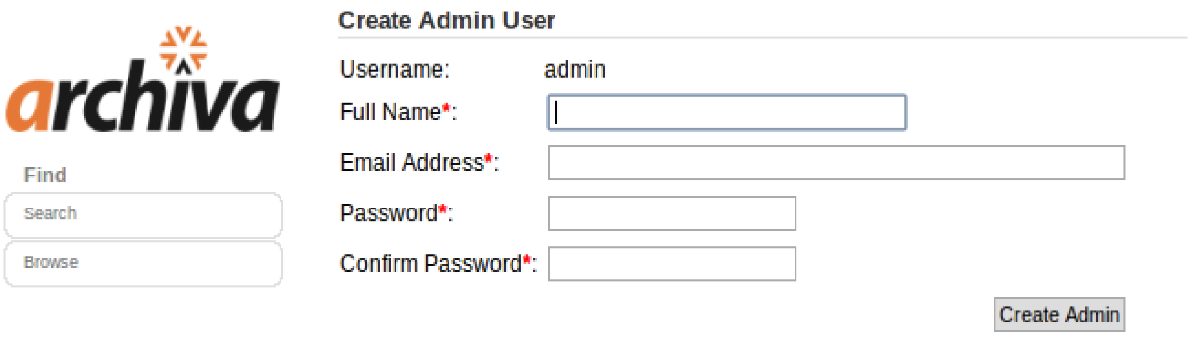
```
[groups]
admins = carlosfeyt, vbarzana
```

Other step is assign privileges to some group. Here you can see an example to do it.

```
[/]
@admins = rw
```

5.2 Apache Archiva administration.

When you enter to Apache Archiva for the first time, Archiva show you a view to create the administration user. Now you can see an example of this view.



Find

Search

Browse

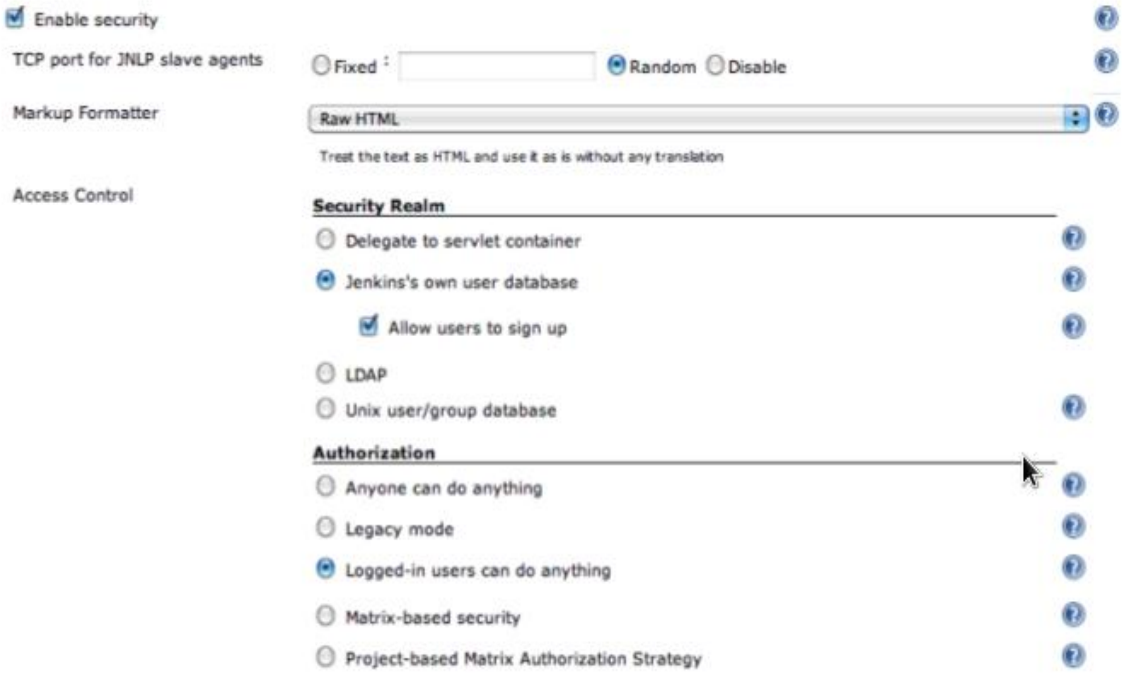
Create Admin User

Username:	admin
Full Name*:	
Email Address*:	
Password*:	
Confirm Password*:	

Picture 1.4 Form to create an administration user in Apache Archiva.

5.3 Jenkins Administration.

One the first step to administer Jenkins is to apply security to own projects. Configure the security in Jenkins is very easy is only open the primary configuration page in Jenkins and active the security clicking at the “Enable security”. Now I will show you a picture how an example to enable the security in Jenkins Server.



☑ Enable security

TCP port for JNLP slave agents ○ Fixed : ● Random ○ Disable

Markup Formatter: Raw HTML

Treat the text as HTML and use it as is without any translation

Access Control

Security Realm

○ Delegate to servlet container

● Jenkins's own user database

☑ Allow users to sign up

○ LDAP

○ Unix user/group database

Authorization

○ Anyone can do anything

○ Legacy mode

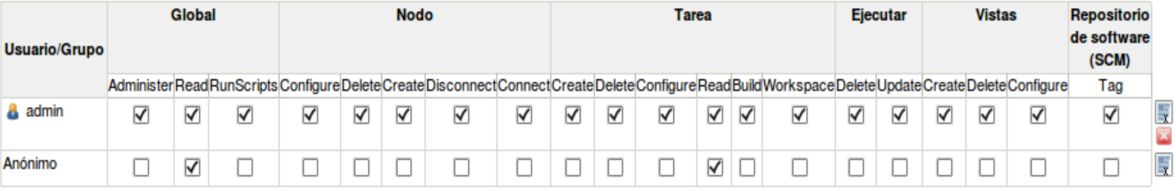
● Logged-in users can do anything

○ Matrix-based security

○ Project-based Matrix Authorization Strategy

Picture 1.5 How to enable the security in Jenkins server.

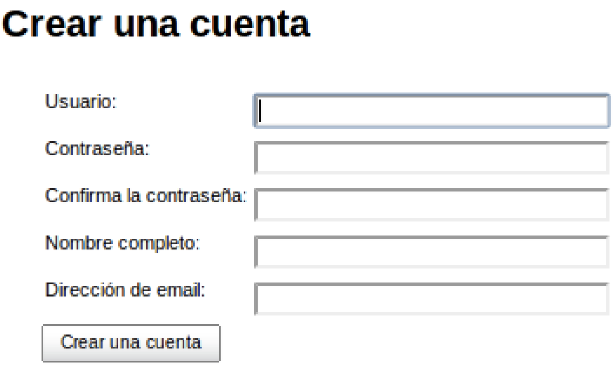
Following at the authorization session select the “strategy for the project security” and set the configuration like the following:



Usuario/Grupo	Global			Nodo					Tarea						Ejecutar		Vistas			Repositorio de software (SCM)
	Administer	Read	RunScripts	Configure	Delete	Create	Disconnect	Connect	Create	Delete	Configure	Read	Build	Workspace	Delete	Update	Create	Delete	Configure	Tag
admin	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑
Anónimo	□	☑	□	□	□	□	□	□	□	□	□	☑	□	□	□	□	□	□	□	□

Picture 1.6 Example of strategy for the project security.

The other step is create the admin user, to do it, click at the registration links located at the windows top. Following you can see the view with a form to create a new user.



Crear una cuenta

Usuario:

Contraseña:

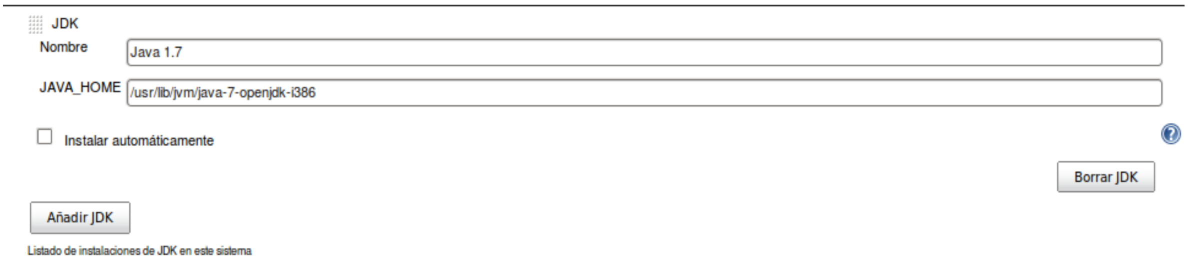
Confirma la contraseña:

Nombre completo:

Dirección de email:

Picture 1.7 Creating a new user for Jenkins.

Another important step is to define the JDK to user for Jenkins Server. To do it is necessary open the primary settings page in Jenkins and go to the JDK session and put the put there where is our JAVA_HOME. Following you can see an example to this configuration.



JDK

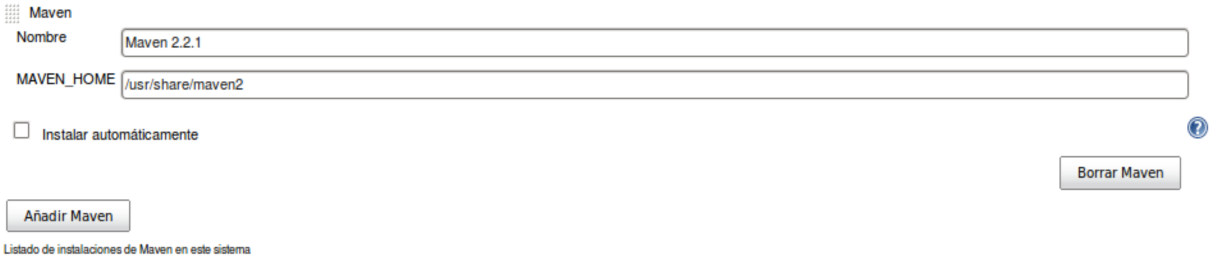
Nombre: Java 1.7

JAVA_HOME: /usr/lib/jvm/java-7-openjdk-i386

□ Instalar automáticamente

Picture 1.8 Configuring the JAVA_HOME in Jenkins.

Same how to configure the JAVA_HOME, you need to configure your Maven version to use. Go to the Maven session at the primary Jenkins settings page and put there the direction to the MAVEN_HOME. Following you can see an example to do it.



Maven

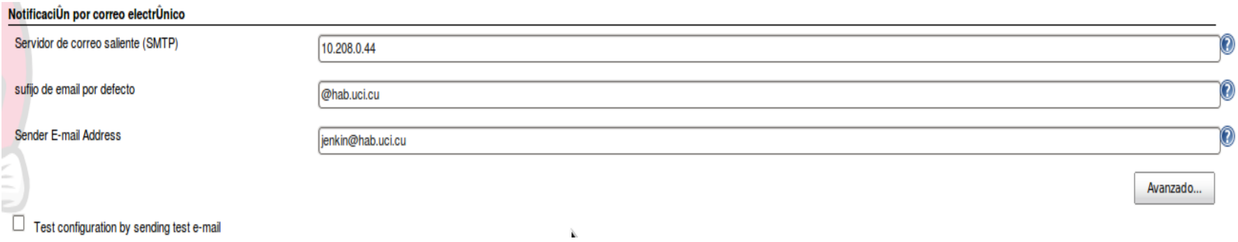
Nombre: Maven 2.2.1

MAVEN_HOME: /usr/share/maven2

□ Instalar automáticamente

Picture 1.8 Configuring the MAVEN_HOME in Jenkins.

The next step is to configure the Jenkins email notifications. To do it is necessary go to the primary settings page in Jenkins, Notifications sessions and put the same configurations like the following:



NotificaciÛn por correo electrÛnico

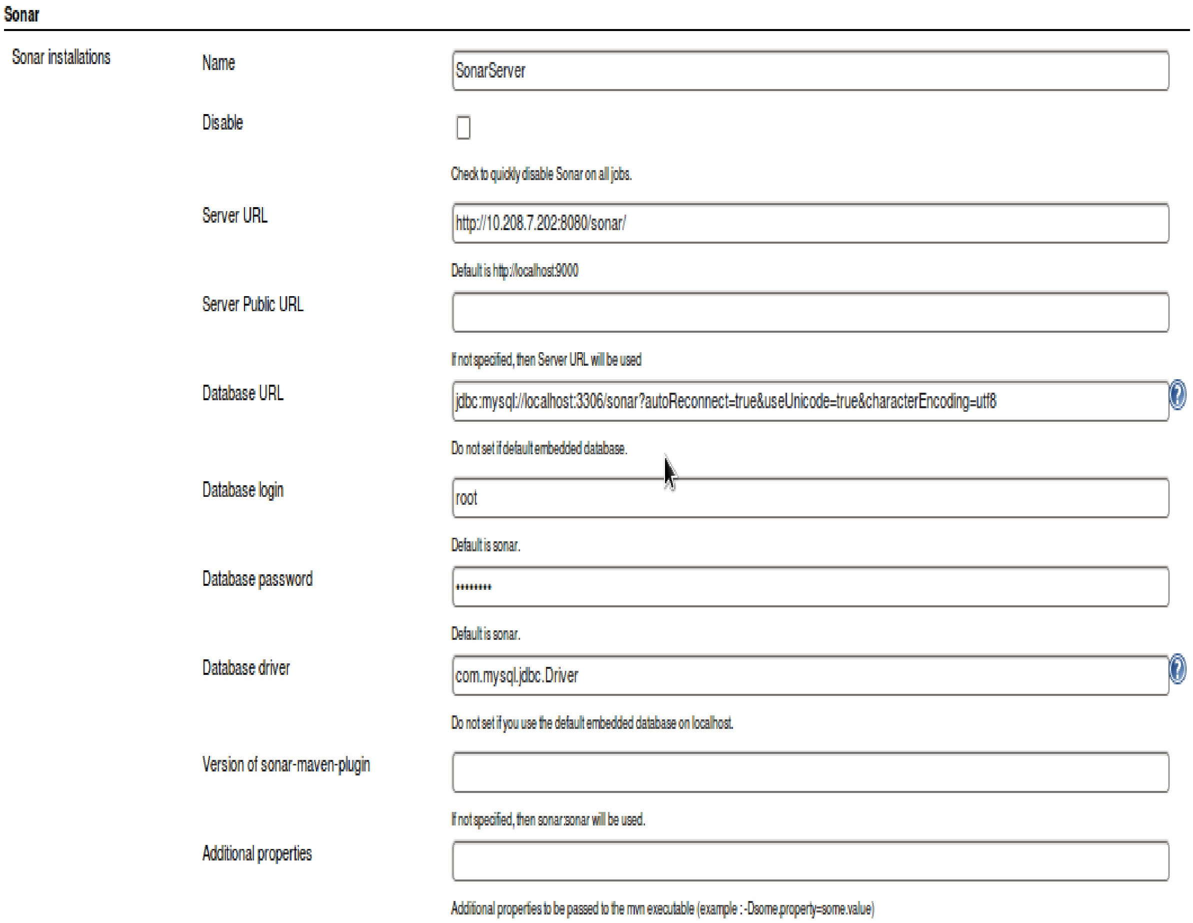
Servidor de correo saliente (SMTP)	10.208.0.44
sufijo de email por defecto	@hab.uci.cu
Sender E-mail Address	jenkin@hab.uci.cu

□ Test configuration by sending test e-mail

Picture 1.9 Configuring the email notification in Jenkins.

Other important step is to configure the sonar plug-in in Jenkins, this plug-in allow to integrate all Jenkins project with Sonar Server to get statistics like lines of code, comments, complexity, an others.

To set the Sonar plug-in configurations, go to the primary settings page at the Sonar session and put there all configurations like the following:



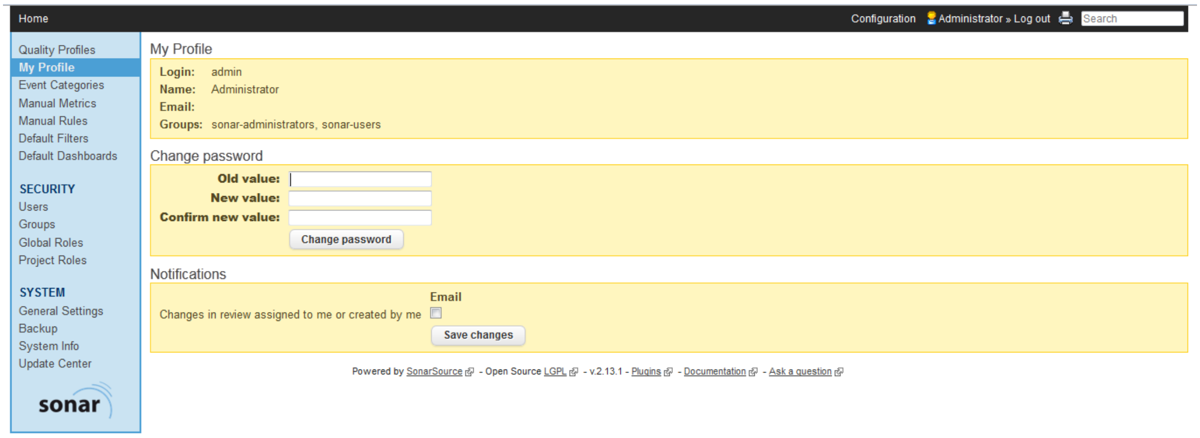
Sonar

Sonar installations	Name	SonarServer
	Disable	□ Check to quickly disable Sonar on all jobs.
	Server URL	http://10.208.7.202:8080/sonar/ Default is http://localhost:9000
	Server Public URL	 If not specified, then Server URL will be used
	Database URL	jdbc:mysql://localhost:3306/sonar?autoReconnect=true&useUnicode=true&characterEncoding=utf8 Do not set if default embedded database.
	Database login	root Default is sonar.
	Database password	•••••••• Default is sonar.
	Database driver	com.mysql.jdbc.Driver Do not set if you use the default embedded database on localhost.
	Version of sonar-maven-plugin	 If not specified, then sonar:sonar will be used.
	Additional properties	 Additional properties to be passed to the mvn executable (example : -Dsome.property=some.value)

Picture 1.10 Configuring the Sonar Plug-in in Jenkins.

5.4 Sonar Administration.

Once installed the sonar server, the first step to do is change the default password. To do it is necessary start session in sonar using the admin user with the same password. Then click at the administration link, there you can change the admin password.



My Profile

Login: admin
Name: Administrator
Email:
Groups: sonar-administrators, sonar-users

Change password

Old value:
New value:
Confirm new value:

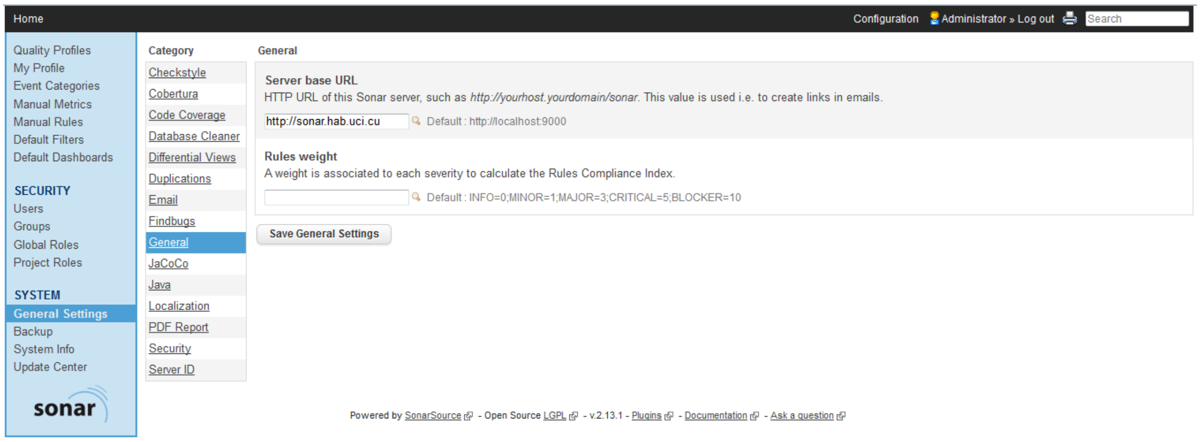
Notifications

	Email
Changes in review assigned to me or created by me	☐

Powered by SonarSource - Open Source LGPL - v.2.13.1 - Plugins - Documentation - Ask a question

Picture 1.11 Change the Sonar admin profile.

The next step to set the URL where is the Sonar server. To do it is necessary go to the settings menu, general settings, general, and then you can see the form to set the Server base URL.



General

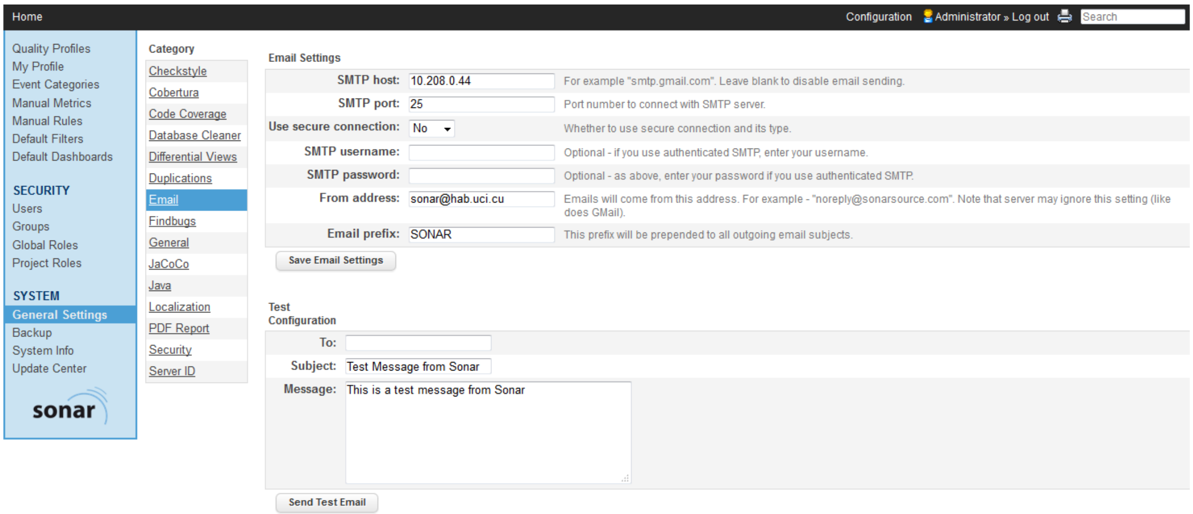
Server base URL
HTTP URL of this Sonar server, such as *http://yourhost.yourdomain/sonar*. This value is used i.e. to create links in emails.
http://sonar.hab.uci.cu Default : http://localhost:9000

Rules weight
A weight is associated to each severity to calculate the Rules Compliance Index.
Default : INFO=0;MINOR=1;MAJOR=3;CRITICAL=5;BLOCKER=10

Powered by SonarSource - Open Source LGPL - v.2.13.1 - Plugins - Documentation - Ask a question

Picture 1.12 Primary sonar settings.

The other important step is to active the notifications in Sonar server. To do it, go to the configurations link, General Settings, Email and set the configuration like the followings.



Email Settings

SMTP host:	10.208.0.44	For example "smtp.gmail.com". Leave blank to disable email sending.
SMTP port:	25	Port number to connect with SMTP server.
Use secure connection:	No	Whether to use secure connection and its type.
SMTP username:		Optional - if you use authenticated SMTP, enter your username.
SMTP password:		Optional - as above, enter your password if you use authenticated SMTP.
From address:	sonar@hab.uci.cu	Emails will come from this address. For example - "noreply@sonarsource.com". Note that server may ignore this setting (like does GMail).
Email prefix:	SONAR	This prefix will be prepended to all outgoing email subjects.

Test Configuration

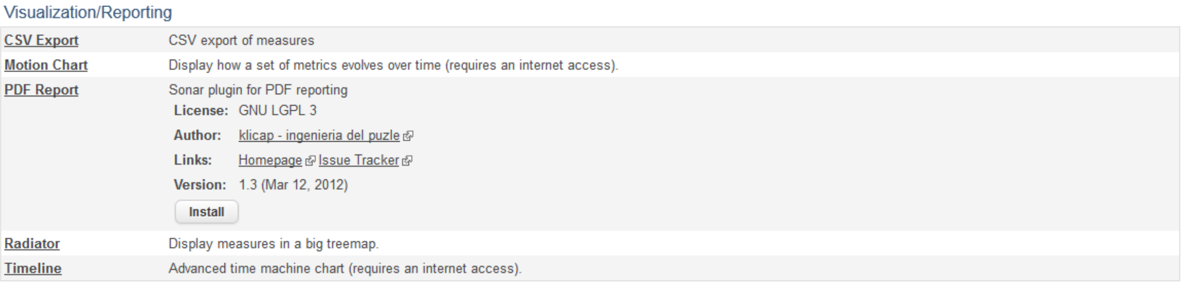
To:
Subject: Test Message from Sonar
Message: This is a test message from Sonar

Figura 1.13 Configuring Email notifications in Sonar.

The next step is to active the PDF report plug-in. The first step to do is configure the internet connection access, to do it, we need to edit the /home/usuario/SONAR_HOME/conf/sonar.properties file and set the following configuration:

```
#---------------------------------------------------------
# UPDATE CENTER
#---------------------------------------------------------
# The Update Center requires an internet connection to request http://update.sonarsource.org
# It is activated by default:
#sonar.updatecenter.activate=true
# HTTP proxy (default none)
http.proxyHost=10.208.0.2
http.proxyPort=3128
# NT domain name if NTLM proxy is used
#http.auth.ntlm.domain=
# SOCKS proxy (default none)
#socksProxyHost=
#socksProxyPort=
# proxy authentication. The 2 following properties are used for HTTP and SOCKS proxies.
http.proxyUser=usuario
http.proxyPassword=password
```

Then you can start to install the PDF plug-in report. To do it, go to the settings menu, update center, and click at the plug-in tab, find the PDF Report plug-in and install it.



Visualization/Reporting

CSV Export	CSV export of measures
Motion Chart	Display how a set of metrics evolves over time (requires an internet access).
PDF Report	Sonar plugin for PDF reporting **License:** GNU LGPL 3 **Author:** klicap - ingenieria del puzle **Links:** Homepage Issue Tracker **Version:** 1.3 (Mar 12, 2012) Install
Radiator	Display measures in a big treemap.
Timeline	Advanced time machine chart (requires an internet access).

Picture 1.14. Installing PDF report plug-in in Sonar