# Administrator Guide

**JWebSocket**

**Continuous Integration Environment for jWebSocket**

**V 1.0**

**Version history?**

1. Downloads process

Before install, configure and start the continuous integration environment it is necessary to download the following software.

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

Please give one short sentence here what these three product are about.

1. 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, 2.0, 2.1, 2.2? They differ significantly.
* MySQL server, which version?
* Apache Tomcat6 server, what about Tomcat 7.0+
* openjdk-7 virtual machine, what about Java 1.6 or Java 1.7 JDK/JRE?

1. Installation process

## Installing Subversion

Above you don’t talk about subversion! List above too?

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.

Please give a sentence what is subversion for.

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

Please give the reader some hints how a successful installation looks like. What errors can appear? Can you provide a screenshot please?

## Installing Apache Archiva

The reader will ask “What is ‘Archiva’?”, so please give a sentence here.

**Creating the ARCHIVA\_HOME folder.**

To install Apache Archiva is necessary to create the ARCHIVA\_HOME folder; there all Archiva installation files are located. This folder will be created at our home folder with the session started. (which session? Will be created or is to be created??, I don’t understand this sentence.

mkdir /home/user/ARCHIVA\_HOME

The next step is to uncompress the latest Archiva version downloaded in this folder.

Finally we can start to the Archiva installations. To do so we need to open the ARCHIVA\_HOME and execute the following line at the console.

./archiva start

How can I see if all was successful? What kind of errors could appear and what do to then?

**3.3 Installing maven2**

What is the maven tool about? For what is that needed in this environment? Please give two sentences of introduction here.

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

apt-get install maven2

**3.4 Installing Jenkins Server**

Again, some introduction desired! What is Jenkins, purpose? One sentence please.

Before we can start the Jenkins server, we need to install some prerequisites like the daemon package. To do so we need to execute the following line at the console.

apt-get install daemon

Once installed the required tools we can start with the Jenkins installation. First it is necessary to open the directory where the Jenkins install file is located and start the installtiopm. To do so we need to execute this line at the console.

dpkg -i jenkins\_1.448\_all.deb

**3.5 Installing Sonar Server**

Once more, please give some introduction sentence, what is Sonar about!

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

Shouldn’t we have a separate paragraph then for MySQL, like for all other tools?

Apt-get install mysql-server

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

mysqladmin -p create sonar

Once the data base is created, we can start with the sonar installations. The first step is to download the latest version and uncompress it in some place (??)like SONAR\_HOME.

Do we need to set any environment variables? Pease check for all above and below installations and give a checklist.

After that 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 created above. To do so, we need to find the “generic settings” and configure like this example:

generic settings  
sonar.jdbc.username: root  
sonar.jdbc.password: losmalos

Never write YOUR! passwords in documentation!!

Once the parameters needed to the configuration are set, we proceed to generate the war to deploy it at our local tomcat server. You explained the installation of all tools, but nothing about Tomat up to here. You should add this for completeness.

To generate the war is necessary open this directory SONAR\_HOME/war and execute this line at the console.

./build-war.sh

How does this shell script do that? It uses maven or Java or what? Please no books here, but the reader will want to know how that works, just a sentence.

This line generates a new .war file ready to deploy it in any Tomcat server. Now we only need to copy it into our webapps tomcat folder (/ var/lib/tomcat6/webapps) and restart the Tomcat server (really, usually Tomcat has a autodeploy per default, please check), then we can open the sonar server at this URL http://<your\_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 in which your repositories are located.

mkdir /home/user/svn

Later the repositories will be created within this folder. To do so we only need to execute this line at the console:

svnadmin create /home/user/svn/jwebsocket

Later (we or who??) 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. (Don’t understand, please check above sentence, seems to turn off read-only flag?)

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

After that we need to move the configuration file of /home/user/svn/jwebsocket.conf/authz to /home/user/svn/authz and edit it like in 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>

Also here you should never give your personal IP numbers or any credentials! Please replace by placeholders.

Now you can enter to the repository using this URL http://<your\_ server\_ip>/svn/jwebsocket/.

Finally it is necessary to configure the subversion hooks. This hook is to allow subversion to execute the Jenkins remote compiling. To do so it is necessary to edit the file /home/usuario/svn/jwebsocket/hooks/post-commit.tmpl and add the following 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 insert/append? the following source.

#!/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**

Per Default, the Archiva server is using the 8080 port, the same of Tomcat. To avoid conflicts, it is necessary to change it to another port. To do so open the ARCHIVA\_HOME/conf/jetty.xml and go to line 66 (this can be different in the future! Please give a better description where to edit/update) 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 (why is this necessary, the reader can not know this here yet) is necessary edit the file /etc/maven2/settings.xml and change the mirrors. Following you can see an example for this file: What does this example do? What’s the purpose of it, please explain.

<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>

Please never publish your private IP numbers! Always use reasonable plcaeholders as already mentioned above!

**4.4 Configuring Jenkins Server**

Per default, Jenkins is using the 8080 port, this is the same port like Tomcat uses. Because of that it is necessary to 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:

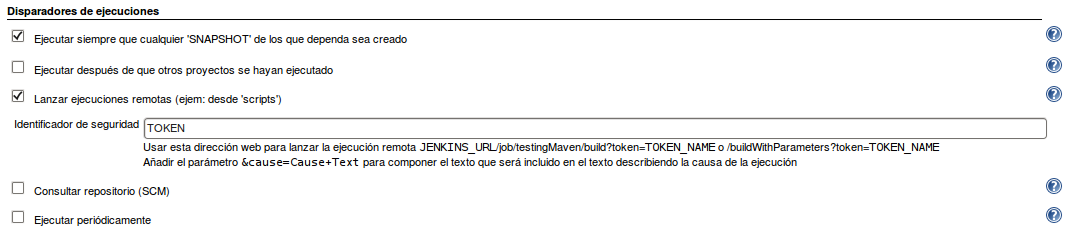
# port for HTTP connector (default 8080; disable with -1)  
HTTP\_PORT=8002

After that you need to restart the Jenkins server:

/etc/init.d/jenkins restart

Now you can access the Jenkins server using this URL: <http://<your>\_server\_ip>:8002/.

The other step is active and configure the triggers for each Jenkins projects, this is to allow remote executions. Without an explanation the reader will not know what “remote execution” is, please explain. To do so it is necessary to go to the configurations session for each project (project? Never created one in the text before, how?) and to configure the triggers like the following picture. (please replace ALL following screenshots by English version).



Picture 1.2 Example of trigger configuration in Jenkins projects.

Finally you can proceed to configure plug-in to integrate Sonar Server (the reader does not know up to here what is “Sonar”, why does he need this “Sonar Server”? with Jenkins. To start it is necessary to configure the internet access in Jenkins, in this example we use the proxy server, to do so go to the administrator menu, plug-in settings, advance settings, then you can see the view to set the proxy configurations.



Picture 1.3 Example to proxy configuration in Jenkins.

Finally you can install the Sonar plug-in (“plug-in”? Above you mention a “Sonar server”. What is it about here?). 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 to create the users. You can create a new Subversion user using the htpasswd command at the console (well, this is an apache command, right?) Where do I find that? Do I need a path to it?. Only for the first time to create a user is use the –mc parameter, the other time only use the –c parameter (why? I know –m creates a new file, but the reader does not! Too many open questions.). Now you can see an example to create the first user in Subversion.

htpasswd -mc /etc/apache2/passwords usuario (english?)

The other step is to set the security in subversion. To do so it is necessary to edit the file /home/usuario/svn/authz. The first step to configure the security is creating the group(s?). Following you can see an example to create a new group.

[groups]   
admins = carlosfeyt, vbarzana

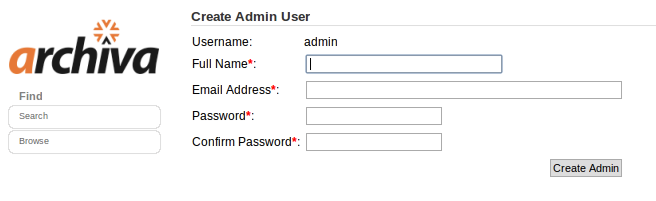
The next step is to assign privileges to the group. Here you can see an example how to do so.

[/]   
@admins = rw

I can “guess” that rw means read/write, but the reader cannot, please more details. What other options?

**5.2 Apache Archiva administration.**

When you enter into Apache Archiva the first time, Archiva shows you a view to create the administration user. Attached you see an example of this view.



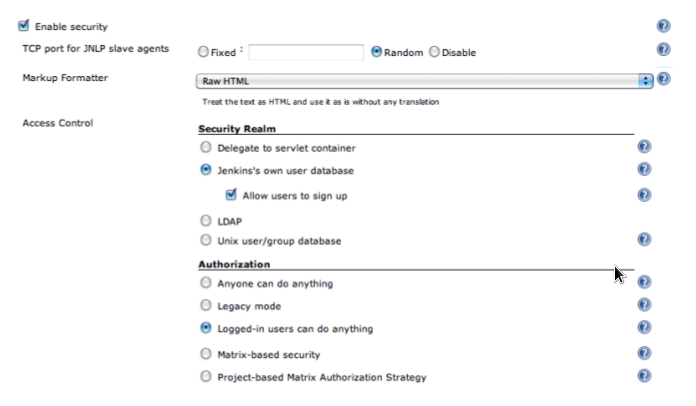
Picture 1.4 Form to create an administration user in Apache Archiva.

Ah, this screenshot is in english, much better :-)

However, what abot other users. What are the special rights of the admin. Which security aspects do I have to consider?

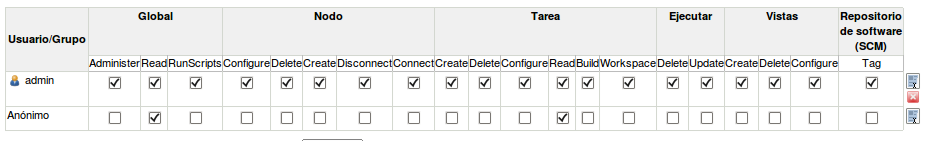
**5.3 Jenkins Administration.**

The first step to administer Jenkins is to apply security to own projects. Again, up to here we never created a project, this should be explained first. Configuring security in Jenkins is very easy. Simply open the primary configuration page in Jenkins and activate the security by clicking at the “Enable security” checkbox. The screenshot shows an example how to enable the security in Jenkins Server.



Picture 1.5 How to enable the security in Jenkins server.

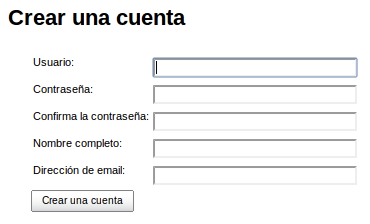
Following at the authorization session select the “strategy for the project security” and set the configuration like the following:



Picture 1.6 Example of strategy for the project security.

Why? Uff, you just give instructions but the reader wants to understand what he has to do and why. This is not sufficient for a guide like this.

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.

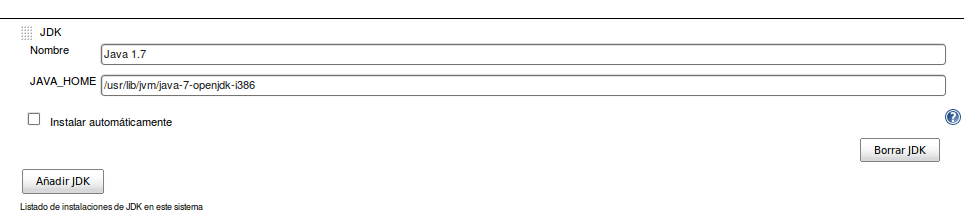


Picture 1.7 Creating a new user for Jenkins.

Can you please provide and english screenshot here?

Another important step is to specify the JDK to be used by the Jenkins Server. To do so is necessary to open the primary settings page in Jenkins and go to the JDK section and type there where is our JAVA\_HOME. Following you can see an example of this configuration.

What about the supported Java Versions?



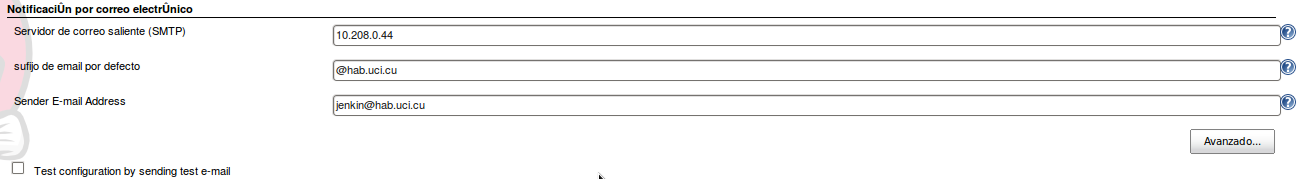
Picture 1.8 Configuring the JAVA\_HOME in Jenkins. Please English Screenshots

In the same way you confgured configured Java, you need to configure your Maven version to be used. Go to the Maven section at the primary Jenkins settings page and type there the address to the MAVEN\_HOME. Following you can see an example how to do so.



Picture 1.8 Configuring the MAVEN\_HOME in Jenkins. Please English Screenshots

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

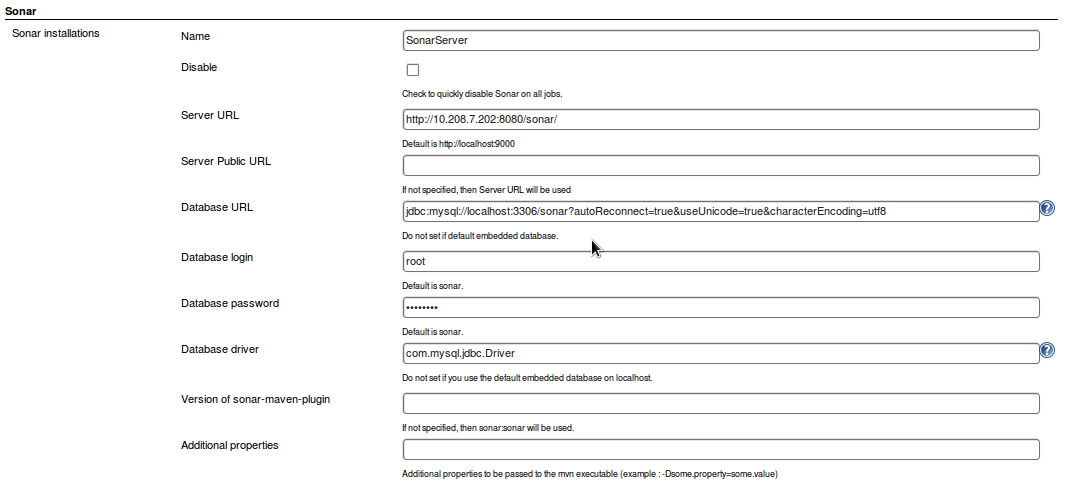


Picture 1.9 Configuring the email notification in Jenkins.

Another important step is to configure the sonar plug-in in Jenkins, this plug-in allows to integrate all Jenkins projects within the Sonar Server to get statistics like lines of code, comments, complexity, an others.

Ah, here I finally understand that “Sonar” is a Server and Jenkins has a plug-in for that, am I right here? I at least know about the structure, still don’t know yet about the purpose, importance, need and benefits.

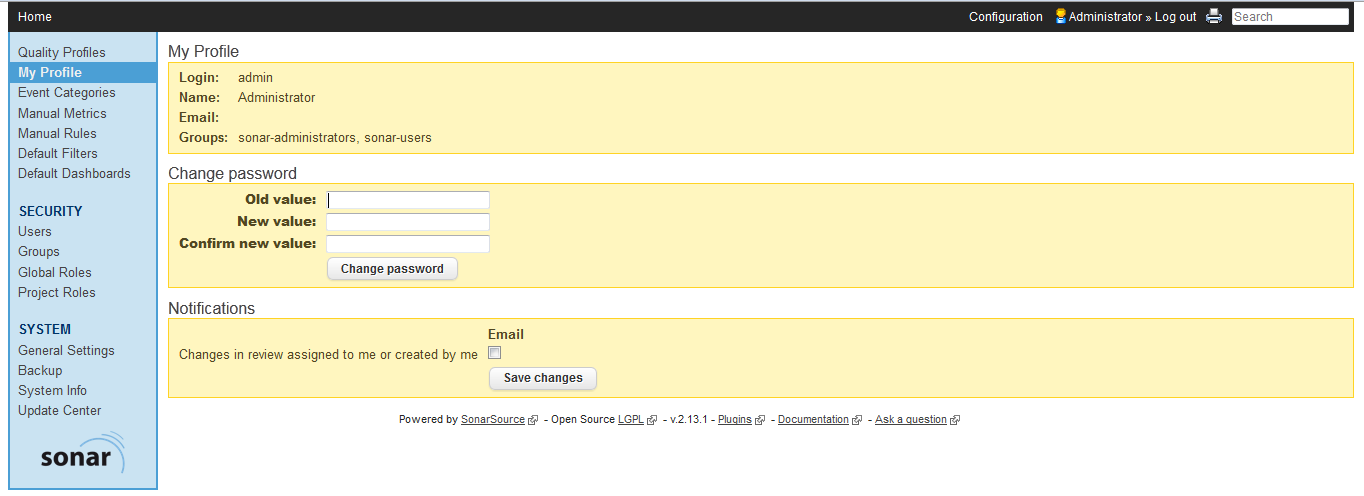
To set the Sonar plug-in configurations, go to the primary settings page at the Sonar section and type there all configurations like the following:



Picture 1.10 Configuring the Sonar Plug-in in Jenkins.Please provide English screenshot

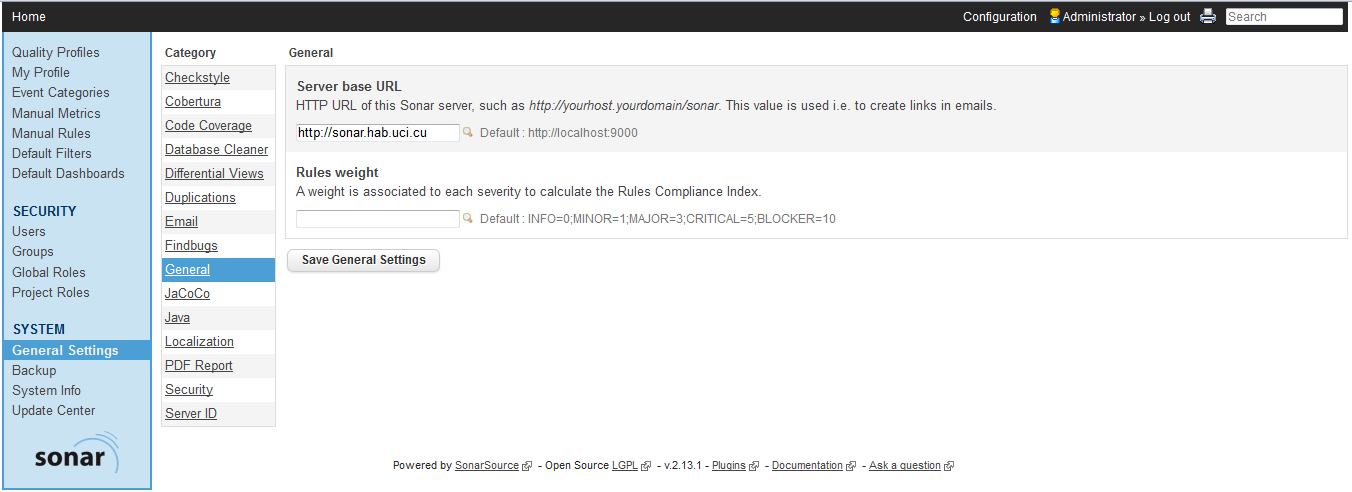
**5.4 Sonar Administration.**

Once the sonar server is installed, the first step to do is to change the default password. To do so it is necessary to start a session in sonar using the admin user with the same password (how does the reader know that? If already explained at least give a reference). Then click at the administration link, there you can change the admin password.



Picture 1.11 Change the Sonar admin profile.

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



Picture 1.12 Primary sonar settings.

The next important step is to activate the notifications in Sonar server (about what does the server inform the receiver? Pelase give a hint or example here). To do so, go to the configurations link, General Settings, Email and set the configuration like follows.

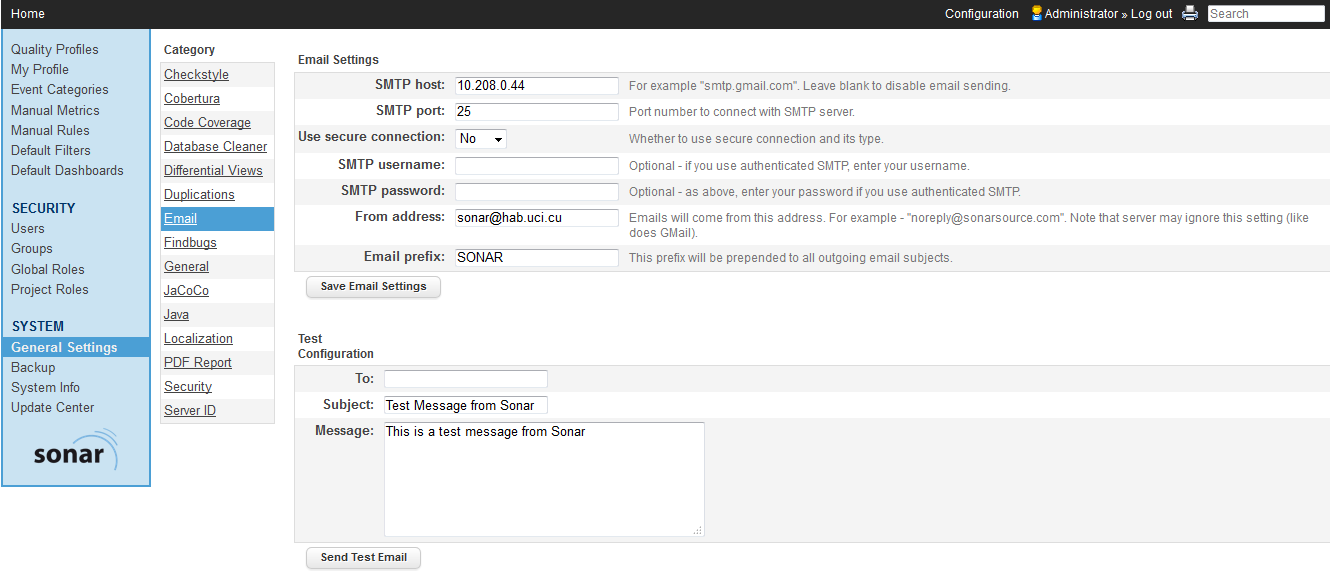


Figura 1.13 Configuring Email notifications in Sonar.

The next step is to activate the PDF report plug-in. The first step to do is to configure the internet connection. To do so, 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=<yourhost\_ip\_number>

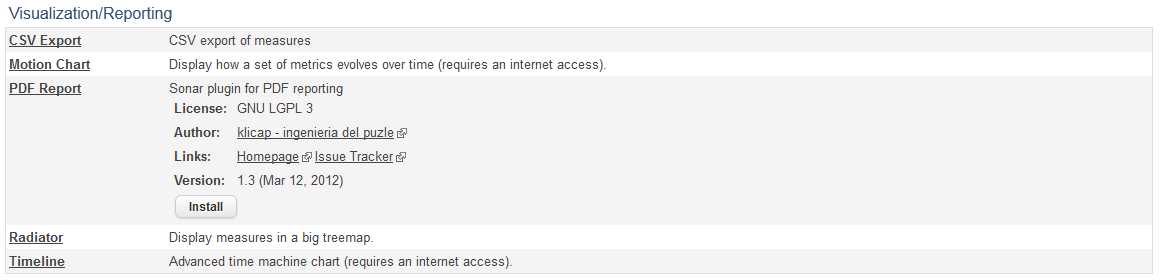
http.proxyPort=3128

# 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=<your password>

Ultimately you can install the PDF report plug-in. (What kind of reports are created, give an example here, and provide a screenshot). To do so, go to the settings menu, update center, and click at the plug-in tab, find the PDF Report plug-in and install it.



Picture 1.14. Installing PDF report plug-in in Sonar