How To

**Jenkins Installation and Setup Guide**

Table of Contents

[1 Introduction 3](#_Toc355037140)

[2 Apache Tomcat Installation 3](#_Toc355037141)

[2.1 Installing Apache Tomcat as a Windows Service 4](#_Toc355037142)

[2.2 Configure Apache Tomcat Ports and Users 7](#_Toc355037143)

[2.3 Verifying Installation 8](#_Toc355037144)

[3 Jenkins Installation 9](#_Toc355037145)

[4 Jenkins Initial Setup 10](#_Toc355037146)

[5 Jenkins Update Centre 12](#_Toc355037147)

[6 Installing and Configuring the Jenkins Sonar Plugin 13](#_Toc355037148)

[7 User Management 15](#_Toc355037149)

[7.1 Contributor Users Management 15](#_Toc355037150)

[7.2 Project Manager Users Management 16](#_Toc355037151)

[8 References 17](#_Toc355037152)

[9 Document Control 19](#_Toc355037153)

[9.1 Change History 19](#_Toc355037154)

[9.2 Open Issues 19](#_Toc355037155)

# Introduction

Jenkins is a Continuous Integration engine that enables easy configuration and management of automated builds, release and deployment management and continuous tests/quality assurance.

Jenkins is the recommended tool in Java Blueprint 3 for adopting Continuous Integration practices.

This document contains detailed insights on how to install Jenkins and setup it for first use. The companion document, **Jenkins Usage Guide**, includes information for end users on how to setup and work with Jenkins jobs.

# Apache Tomcat Installation

The preferred way of running Jenkins, for best performance and user experience, is embedded inside a Java web application container. Apache Tomcat is a well-known, reference web application container from the Apache Foundation. It is Open Source and probably the most used web container.

NOTE: The installation process below assumes that you are installing only Jenkins in this instance. If you plan for installing both Jenkins and Sonar, it is recommended to place them in different Tomcat instances. It is possible, however, to have them running in the same instance. In this case, please be aware that you should skip part of the process when the second tool is being installed.

Before installing Tomcat, we need to ensure that the Oracle Java Development Kit (JDK) is installed and that the environment variable JAVA\_HOME points to the folder where JDK is installed. The Java Blueprint 3.2 recommends using JDK 7 update 15 version. In the rest of the guide we will assume that JDK is installed in **c:\java\jdk-7** folder. Update all references to this folder in case you choose to install JDK in a different place. JDK can be downloaded from: [http://www.oracle.com/technetwork/java/javase/downloads/java-archive-downloads-javase7-521261.html#jdk-7u15-oth-JPR](http://www.oracle.com/technetwork/java/javase/downloads/java-archive-downloads-javase7-521261.html" \l "jdk-7u15-oth-JPR)

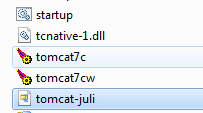
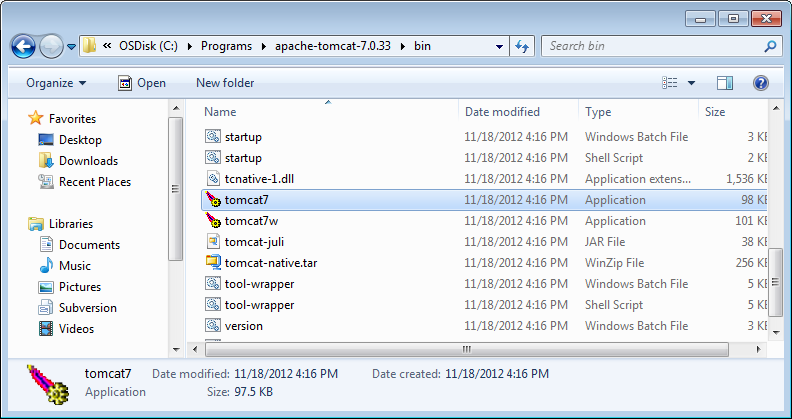
If you are installing Tomcat and Jenkins in a 64-bit box, you may choose to install the 64-bit version of JDK 6. This will enable the Tomcat process to be able to manage more than 2 GB of RAM, useful when you have Jenkins and Sonar running in the same instance or a high number of users or configured jobs.

Once JDK 7 is installed, we decompress the binary distribution of Tomcat (recommended version is 7.0.22) on any desired folder. In the rest of the guide we will assume that Tomcat is installed in **C:\Apps\apache-tomcat-7.0.22** folder. Update all references to this folder in case you choose to install Tomcat in a different place. Tomcat can be downloaded from: <http://archive.apache.org/dist/tomcat/tomcat-7/v7.0.22/bin/apache-tomcat-7.0.22.zip>

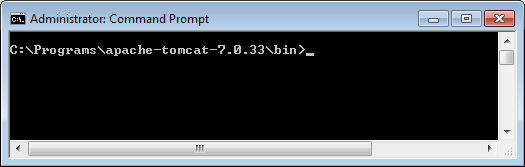
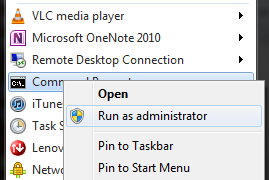
## Installing Apache Tomcat as a Windows Service

It is recommended to install Tomcat as a Windows Service. This way Tomcat (and Sonar with it) will be available automatically each time Windows is started, without the need for human intervention.

First thing to consider when installing Tomcat as a service is whether multiple Tomcat instances will be running in the same box. If this is the case, you may need to rename two files inside Tomcat **bin** folder: **tomcat7.exe** and **tomcat7w.exe**. The name of these files, **tomcat7**, is used to identify the service in Windows registry. Thus, to prevent name conflicts, you may rename both files in the different Tomcat instances, e.g. **tomcat7c.exe** and **tomcat7cw.exe**.



Once the names are correct, we proceed with the installation of the service. To do that, open a command prompt in Windows with administrator privileges and move to the Tomcat **bin** folder.



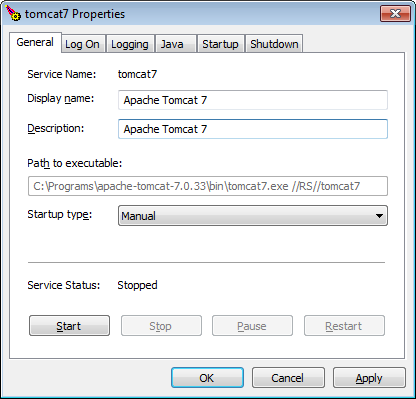
Under **bin** folder, issue the following command:

tomcat7 //IS//tomcat7 --Install=C:\Apps\apache-tomcat-7.0.22\bin\tomcat7.exe --Jvm=auto --StartMode=jvm --StopMode=jvm --StartClass=org.apache.catalina.startup.Bootstrap --StartParams=start --StopClass=org.apache.catalina.startup.Bootstrap --StopParams=stop

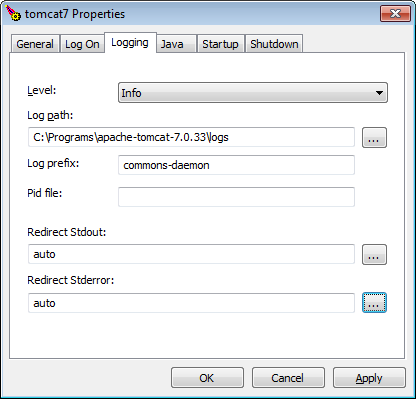
You may need to adapt the names and paths accordingly to the service name and Tomcat installation folder.

Once the service is installed, we need to configure it running the executable ending in ‘w’, e.g. **tomcat7w.exe**. A dialog with multiple tabs will appear. To complete Tomcat installation, follow these steps:

* Modify the service name and description (if needed and desired).
* Modify start-up type to Automatic (recommended).



* Modify logon account if needed, e.g. to access network resources.
* Modify logging settings:
  + Level: Error – Can be switched to info/debug if needed to troubleshoot problems
  + Path: C:\Apps\apache-tomcat-7.0.22\logs
  + Redirect: auto – In both streams



* Modify Java options:
  + JVM: c:\java\jdk-7\jre\bin\server\jvm.dll
  + Java claspath: C:\Apps\apache-tomcat-7.0.22\bin\bootstrap.jar;C:\Apps\apache-tomcat-7.0.22\bin\tomcat-juli.jar
  + Java options:

-Dcatalina.base=C:\Apps\apache-tomcat-7.0.22

-Dcatalina.home=C:\Apps\apache-tomcat-7.0.22

-Djava.endorsed.dirs=C:\Apps\apache-tomcat-7.0.22\endorsed

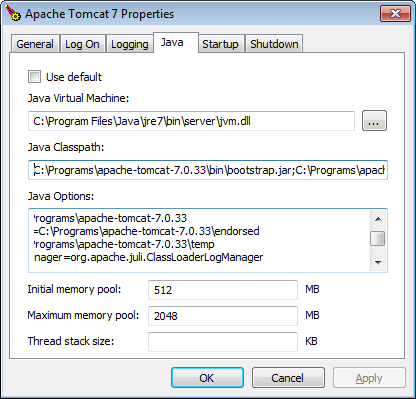
-Djava.io.tmpdir=C:\Apps\apache-tomcat-7.0.22\temp

-Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager

-Djava.util.logging.config.file=C:\Apps\apache-tomcat-7.0.22\conf\logging.properties

-XX:MaxPermSize=512m

* + Initial memory: 512
  + Maximum memory: 2048



## Configure Apache Tomcat Ports and Users

Next step is to configure ports and users for Tomcat.

Usually default ports work fine just after installation, but in the case that other software is using the same ports (or you want to setup different Tomcat instances) you may need to update ports to be used in configuration.

To update the ports, the file **conf\server.xml** needs to be edited. You will need to search for 4 different ports. By default they are:

* HTTP transport: 8080
* HTTPS transport: 8443
* Shutdown port: 8005
* AJP connector: 8009

To enable new ports, just search in the configuration file all occurrences of the ports above and replace them for the chosen ones.

<Server port="**8005**" shutdown="SHUTDOWN">

…

<Connector port="**8080**" protocol="HTTP/1.1" URIEncoding="UTF-8"

connectionTimeout="20000" redirectPort="**8443**" />

…

<Connector port="**8009**" protocol="AJP/1.3" redirectPort="**8443**" />

To update the users, the file **conf\tomcat-users.xml** needs to be edited. This step is needed to enable an administrator user to access Tomcat console and manage the server. Once opened, edit the file ensuring that a proper administrator user is configured with manager and admin roles:

<tomcat-users>

<role rolename="tomcat"/>

<role rolename="manager"/>

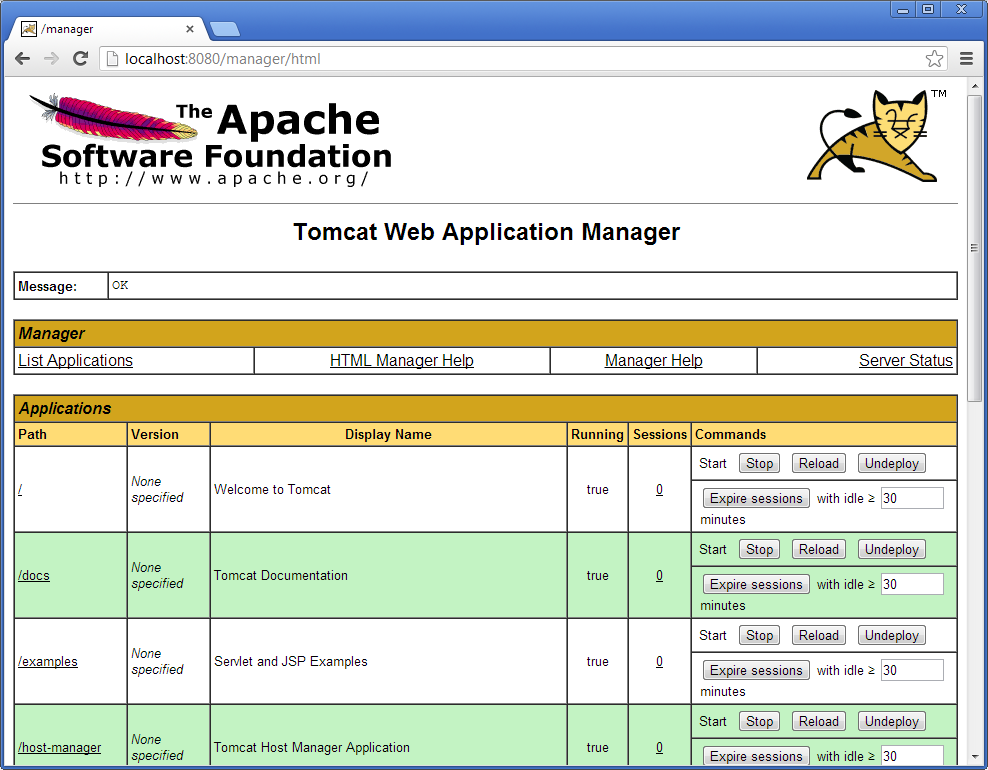
<role rolename="admin"/>

<user username="admin" password="adminadmin" roles="tomcat,manager,admin"/>

</tomcat-users>

## Verifying Installation

Once the previous steps are finished, you should verify that Tomcat is properly installed by starting the service and opening in your browser the administration console URL: <http://localhost:<port>/manager/html>



# Jenkins Installation

This section of the guide will explain how to install Jenkins inside the Apache Tomcat instance installed on previous section.

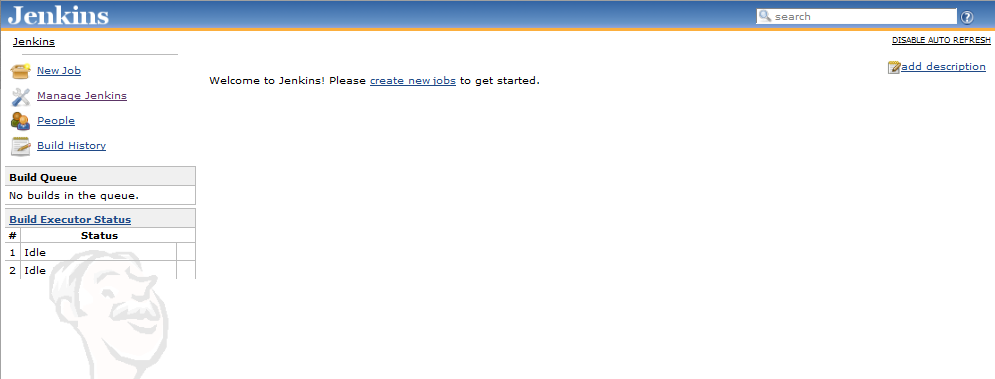
The Java Blueprint 3.2 recommends the installation of Jenkins 1.480.2 that can be downloaded from this URL: <http://mirrors.jenkins-ci.org/war-stable/1.480.2/jenkins.war>

Please get the latest stable version.

Before installing Jenkins, please ensure that the Apache Tomcat service is not running. Once stopped, copy the **jenkins.war** file to **C:\Apps\apache-tomcat-7.0.22\webapps** folder.

Jenkins stores all configuration and job execution results under a folder in the hard disk. The environment variable JENKINS\_HOME should be defined pointing to the directory where data will be stored. For example we will use **c:\Apps\jenkins\jenkins-data**. The directory should exist before Jenkins is executed.

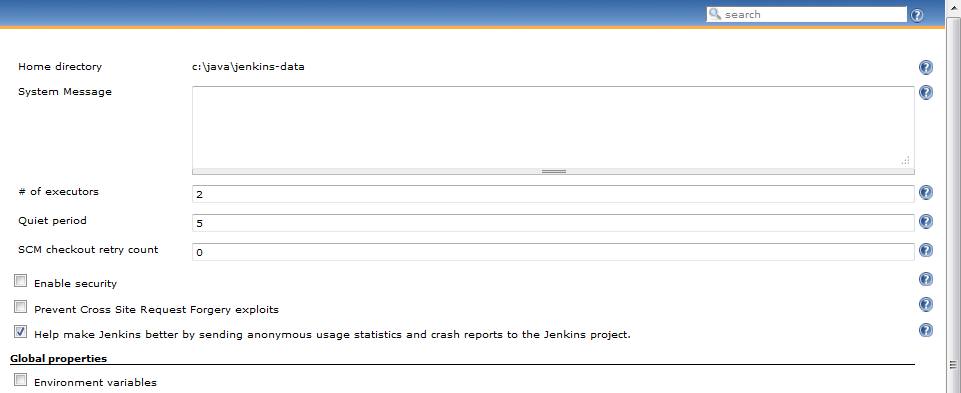
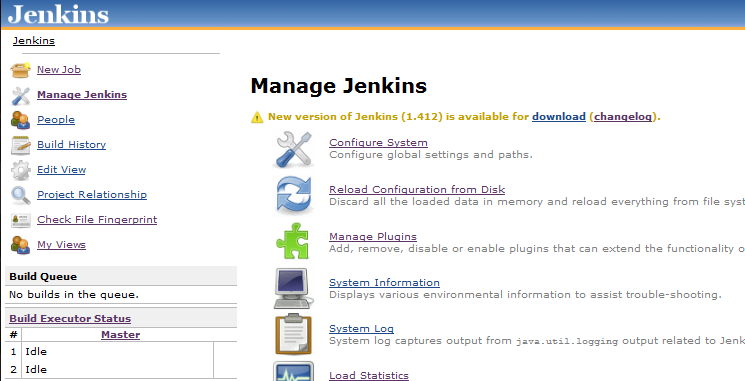
Now start the Tomcat service. After a few seconds, Tomcat should be up and running including the Jenkins engine with it. Verify that Jenkins is properly installed by opening in your browser the Jenkins dashboard URL: <http://localhost:<port>/jenkins>



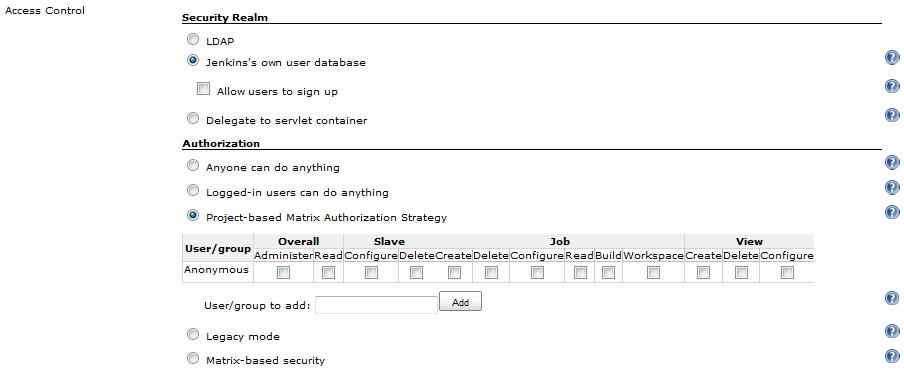
# Jenkins Initial Setup

Now that Jenkins is installed and we have access to Jenkins dashboard, we will prepare it for first use.

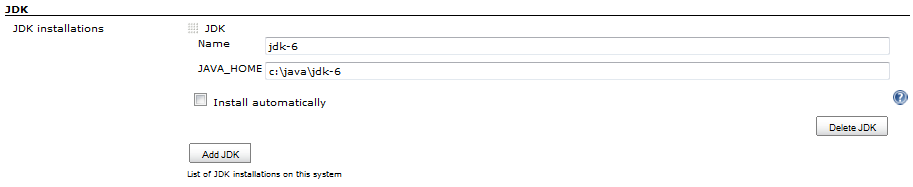
In the left menu, select **Manage Jenkins**. Select the option **Configure System** from the list of management tasks:



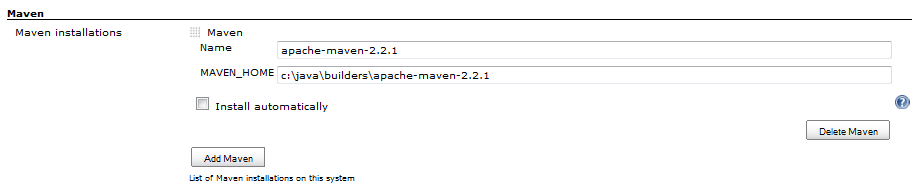
Activate the check-box **Enable Security**. In the new section that appears, under **Access Control**/**Security Realm**, select the radio-button **Jenkins's own user database** and deactivate the check-box **Allow users to sign up**. In the same section, under **Access Control**/**Authorization**, select the radio-button **Project-based Matrix Authorization Strategy**. A new section will appear to configure actions for anonymous users. Ensure that anonymous users do not have any permission granted.



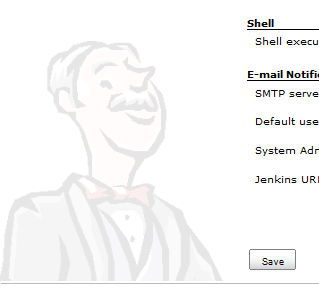
In the section JDK, click the **Add JDK** button and configure the name and location where JDK is installed. Multiple JDK versions are allowed and later any job may select which version to use when building the project and running the tests. Deactivate the check-box **Install automatically** if selected.



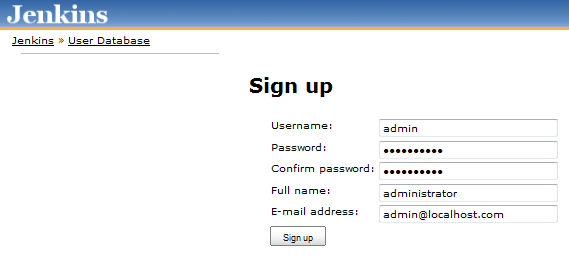
In the section Maven, click the **Add Maven** button and configure the name and location where Apache Maven is installed. Multiple Maven versions are allowed and later any job may select which version to use when building the project and running the tests. Deactivate the check-box **Install automatically** if selected.



Once these steps are completed, you can proceed to the bottom of the screen and press the **Save** button.



Now a screen appears to enter and confirm the administrator credentials:

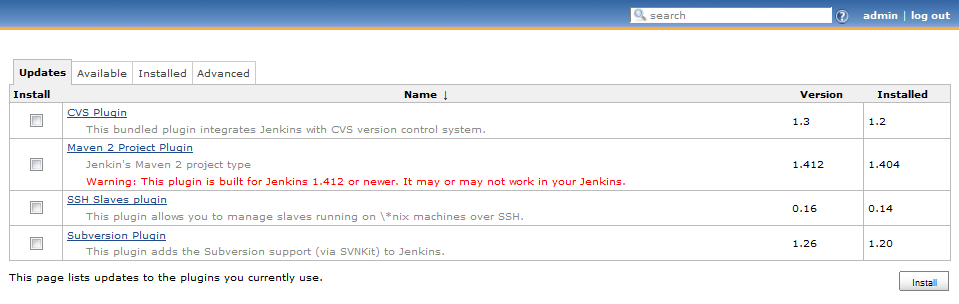
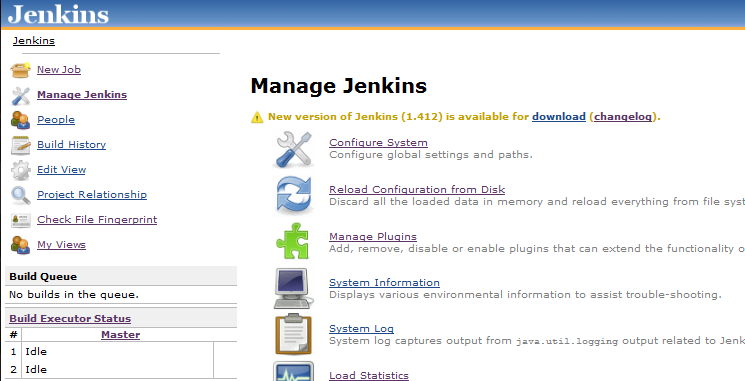


Press the **Sign up** button and Jenkins is ready for first use. For information on how to create new jobs and work with them, please refer to the companion document **Jenkins Usage Guide**. For information on how to integrate Sonar with Jenkins, please refer to the companion document **Sonar Installation and Setup Guide**.

# Jenkins Update Centre

Jenkins comes with an update centre that allows easy installation and upgrade of Jenkins engine and plugins. Although Jenkins is bundled with some core plugins that enable typical use cases – Maven integration, CVS integration, Subversion integration and Slave management – you can install new plugins any time that will extend Jenkins functionality and match your needs.

To access the update centre, in the left menu select **Manage Jenkins**. Select the option **Manage Plugins** from the list of management tasks:



In the update centre screen there are four tabs: **Updates** shows installed plugins with available updates, **Available** shows the catalogue of available plugins, **Installed** shows the list of installed plugins and **Advanced** is used to configure proxy or to manually upload plugins (e.g. you develop your own plugin and want to test it before publishing it).

Either from **Updates** or from **Available**, you can select any number of plugins to update or to install. Then, when the **Install** button at the bottom is pressed the plugins will be downloaded and stored in the Jenkins plugin folder. Once download is completed, a message will appear asking you to restart Jenkins. Once restarted, the new plugins will be available for initial configuration and use.

The plugin catalogue is huge, more than 300 and growing. There are available plugins for artefact publishing and release management, integration with external authentication systems (LDAP, OpenID), flexible notifiers (mail, Twitter, instant messaging), build report creators and integration with quality tools, integration with build tools and languages other than Java, build triggers and wrappers (VMware or VirtualBox integration), distributed builds (slaves, cloud), integration with popular trackers and dashboards (Sonar, Jira, Bugzilla), integration with source code management systems (ClearCase, TFS, Git) and many others.

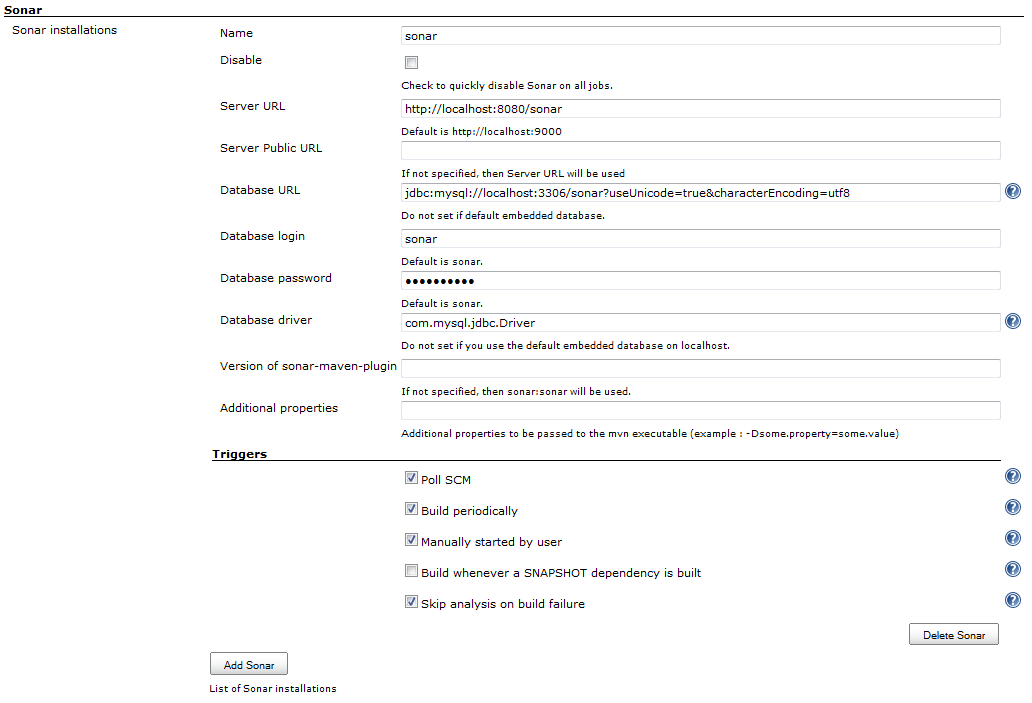
## Installing and Configuring the Jenkins Sonar Plugin

Jenkins and Sonar work closely together to provide the best of the Continuous Integration and Continuous Quality Assurance practices. Through the use of the Jenkins Sonar plugin, we can launch a Sonar analysis on selected jobs and link the Jenkins dashboard with Sonar dashboard to review analysis results.

To install the Jenkins Sonar plugin, just go to the Jenkins update centre as described in previous section and search for the Sonar plugin to install it. Restart Jenkins to finish off with the plugin installation.

Although in the rest of the section we are going to describe how to configure the plugin parameters, note that the Java Build Accelerator installer already includes Jenkins and Sonar configured to work together.

Once installed, log in to the Jenkins dashboard as administrator and in the left menu select **Manage Jenkins**. Then select **Configure System** in the list of management tasks. Now that the plugin is installed, near the end of the screen you will find a **Sonar** section. In this section we can add as many Sonar instances as needed, selecting for each one the connectivity parameters and details on how to trigger the Sonar analysis.



The connectivity parameters, and default values when using recommended setup and settings, are:

* **sonar.jdbc.url** = jdbc:mysql://localhost:3306/sonar?useUnicode=true&characterEncoding=utf8
* **sonar.jdbc.driver** = com.mysql.jdbc.Driver
* **sonar.jdbc.username** = sonar
* **sonar.jdbc.password** = sonarsonar
* **sonar.host.url** = http://localhost:8080/sonar

Don’t forget to update previous settings in the case that you are not following the recommended setup and settings, e.g. you are using a different database engine or have Jenkins and Sonar in different boxes.

The **Triggers** subsection can be used to globally enable or disable Sonar analysis from selected types of builds. For example, we may enable Sonar analysis in periodic builds but not in those triggered by an SCM poll. It is recommended, unless needed, to globally activate Sonar in all builds except when triggered by a dependency.

# User Management

Jenkins comes with a powerful security module. User’s access levels can be configured with great detail and flexibility, but usually it is not needed to enter in such detail.

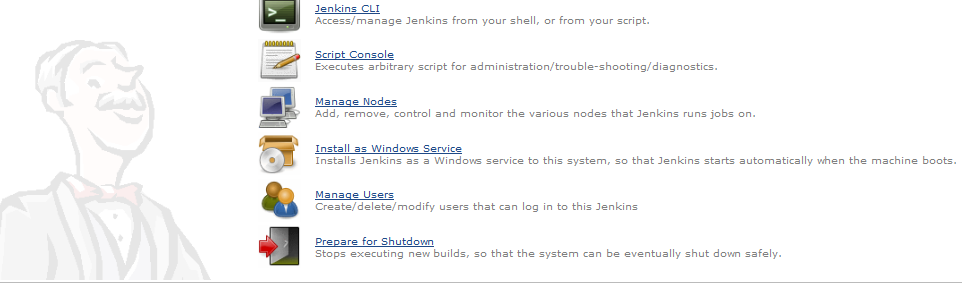
To simplify the user management in Jenkins we classify user in four groups:

* Anonymous users: do not have any permission, so they cannot access to any view in the dashboard. Specifically anonymous users cannot see what projects are configured, their status or history.
* Contributor users: they are users associated with the activity in source code management systems. Their names and domains are dependant of the configuration management engine in use by each project. These users are added by Jenkins when it reviews the activity in source code repositories. A Jenkins administrator may associate an e-mail address to each contributor user so Jenkins is able to send notifications, when a build fails, to the users that contributed to that build.
* Project manager users: they are Jenkins users with permission to create jobs, configure and manage them, organize them in views and manually launch build processes. These users are created by a Jenkins administrator upon request from the project they belong to.
* Administrator users: they have full permissions, including user creation and management and Jenkins configuration and updates. Only one administrator user will be available, with the credentials specified while doing Jenkins initial setup.

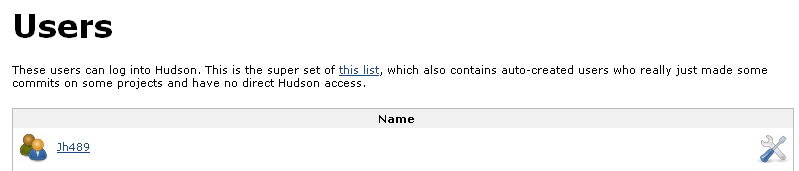
## Contributor Users Management

This section describes the process to manage contributor users.

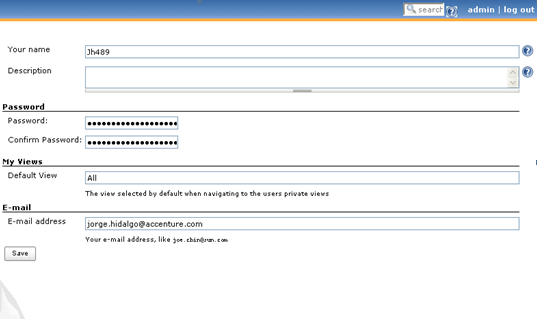
First of all, access the Jenkins dashboard URL and login with administrator credentials. In the left menu, select **Manage Jenkins**. Select the option **Manage Users** from the list of management tasks:



The next screen will show the list of users currently managed in this Jenkins instance. Select the tools icon to the right of the user that we want to manage:



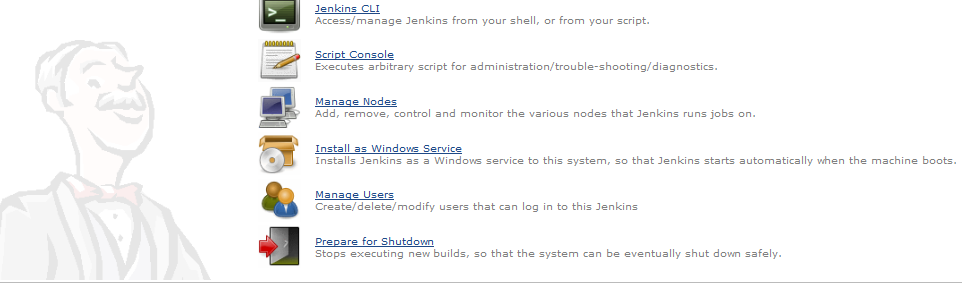
Add the e-mail address for this user and press the button **Save** at the bottom to finish:



## Project Manager Users Management

This section describes the process to manage project manager users.

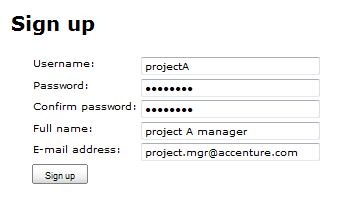
First of all, access the Jenkins dashboard URL and login with administrator credentials. In the left menu, select **Manage Jenkins**. Select the option **Manage Users** from the list of management tasks:



The next screen will show the list of users currently managed in this Jenkins instance. Select the option **Create User** from the left menu:



Add the name and password for the user – typically the name of the project or the project manager is a good choice. Add also the e-mail for the user, to enable notifications when a build fails or is unstable:



Once finished, press the **Sign up** button.

# References

Apache Tomcat: <http://tomcat.apache.org/>

Jenkins: <http://jenkins-ci.org/>

How to install Tomcat as a Windows Service: <http://tomcat.apache.org/tomcat-7.0-doc/windows-service-howto.html>

# Document Control

## Change History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | Date | Author | Approver | Comment |
| 0.1 | 01/09/2015 | Sandip Mante |  | First draft version |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

## Open Issues

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Opening Version | Date | Author | Action | Owner | Status | Closing version |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |