

Uso de Sonar en Jenkins

Cristopher Caamana Gómez

Haridian Rodríguez Pérez

Laboratorio de Desarrollo y Herramientas

28/12/2012

Contenido

Introducción	2
Instalación de Jenkins.....	2
Configuración de Jenkins.....	3
Plugins:	3
Configuración módulos sonar.....	4
Configuración de sonar-runner:	4
Configuración Sonar:	5
Creación de la tarea.....	5
Ejecución.....	8
Resultados en el Sonar:	8
Conclusión:	9

Introducción

Jenkins es una herramienta de integración continua para el desarrollo de software. Soporta herramientas de control de versiones como CVS, Subversion, Git, Mercurial, Perforce y Clearcase y puede ejecutar proyectos basados en Apache Ant y Apache Maven, así como scripts de Shell y programas Batch de Windows.

Instalación de Jenkins

Si vamos a instalarlo en **Arch Linux** simplemente vamos a la terminal y lo descargamos del repositorio de usuarios de la comunidad (AUR) con el siguiente comando:

```
packer -S jenkins
```

En el caso de hacerlo en **Ubuntu** podemos hacerlo de dos formas:

1. Escribiendo en la terminal el comando:

```
sudo apt-get install jenkins
```

Desventaja: se encuentra desactualizado.

2. Si queremos descargar la última actualización debemos ir al siguiente enlace:

<http://pkg.jenkins-ci.org/debian/>

De donde descargamos el archivo correspondiente y lo instalamos mediante el centro de actualizaciones de Ubuntu.

El puerto por defecto del Jenkins es el localhost/8070 (Arch Linux) o localhost/8080 (Ubuntu). En el caso que queramos cambiar de puerto, modificamos el archivo `/etc/default/jenkins` con el siguiente comando:

```
sudo nano /etc/default/jenkins
```

Una vez instalado y accedido a la dirección, podemos ver el panel de administración de Jenkins, donde aparecen las tareas que han sido creadas mediante la opción *Nueva Tarea*.

El plugin de Sonar se denomina *Jenkins Sonar Plugin*. A parte de eso, instalamos los plugins de php y Github, puesto que vamos a analizar código en php y obtendremos el código a analizar de Github.

Configuración módulos sonar

Para poder utilizar sonar dentro de Jenkins, es necesario configurar los módulos de Sonar y Sonar Runner accediendo a *Administrar Jenkins > Configuración*.

Configuración de sonar-runner:

Sonar Runner

instalaciones de Sonar Runner

Sonar Runner

Name

Sonar Runner

☒

Instalar automáticamente

Install from Codehaus

Versión

Sonar Runner 2.0 ▼

Añadir un instalador ▼

Añadir Sonar Runner

Borrar un instalador

Borrar Sonar Runner

Listado de instalaciones de Sonar Runner en este sistema

Configuración Sonar:

Sonar

Instalaciones de "Sonar"	Name	<input type="text" value="sonar"/>
Desactivar	<input type="checkbox"/>	Seleccionar para desactivar Sonar en todas las tareas
URL del servidor	<input type="text" value="http://localhost:9000"/>	Por defecto es http://localhost:9000
Sonar account login	<input type="text" value="admin"/>	Sonar account used to perform analysis. Mandatory since Sonar 3.4 when anonymous access is disabled.
Sonar account password	<input type="password" value="****"/>	Sonar account used to perform analysis. Mandatory since Sonar 3.4 when anonymous access is disabled.
URL pública del servidor	<input type="text"/>	Si no se especifica se utilizará la URL del servidor
URL de la base de datos	<input type="text"/>	No poner nada si se utiliza la base de datos embebida.
Usuario de la base de datos	<input type="text" value="sonar"/>	Por defecto es "sonar".
Contraseña de la base de datos	<input type="password" value="****"/>	Por defecto es "sonar".
Controlador (Driver) para la base de datos	<input type="text"/>	No poner nada si se utiliza la base de datos embebida en "localhost"
Version of sonar-maven-plugin	<input type="text"/>	If not specified, then sonar:sonar will be used.
Propiedades adicionales	<input type="text"/>	Propiedades adicionales para pasar a maven (ejemplo: -Dsome.property=some.value)

Trigger Exclusions (only for jobs with Sonar as a post build action)

Skip if triggered by SCM Changes	<input type="checkbox"/>
Skip if triggered by the build of a dependency	<input type="checkbox"/>
Skip if environment variable is defined and set to true	<input type="text"/>

Listado de instalaciones Sonar

Desventaja: El sonar trabaja obligatoriamente con Maven dentro de Jenkins, por lo que para poder utilizarlo hace indispensable el uso de éste. Dado que no hemos trabajado con dicha herramienta, no podremos configurarla, y por lo tanto la ejecución sería errónea. En este caso, solo se utilizará Sonar Runner, y nos enviará a la página de administración de Sonar para ver los resultados.

Creación de la tarea

Para la creación de la nueva tarea en Jenkins, pinchamos en *Nueva Tarea*, el cual nos mostrará la siguiente ventana, donde rellenamos la información inicial, y pulsamos "ok":



Jenkins buscar

Jenkins > Todo

[Nueva Tarea](#)
[Personas](#)
[Historial de construcción](#)
[Administrar Jenkins](#)

Trabajos en cola
 No hay trabajos en cola

Estado de los nodos

#	Estado
1	Inactivo
2	Inactivo

Nombre del Tarea

☒ **Crear un proyecto de estilo libre**
 Esta es la característica principal de Jenkins, la de ejecutar el proyecto combinando cualquier tipo de repositorio de software (SCM) con cualquier modo de construcción o ejecución (make, ant, mvn, rake, script ...). Por tanto se podrá tanto compilar y empaquetar software, como ejecutar cualquier proceso que requiera monitorización.

☐ **Crear un proyecto maven2/3**
 Ejecuta un proyecto maven2/3. Jenkins es capaz de aprovechar la configuración presente en los ficheros POM, reduciendo drásticamente la configuración.

☐ **Crear un proyecto multi-configuración**
 Adecuado para proyectos que requieran un gran número de configuraciones diferentes, como testear en múltiples entornos, ejecutar sobre plataformas concretas, etc.

☐ **Monitorizar una tarea externa**
 Este tipo de tareas te permite registrar la ejecución de un proceso externo a Jenkins, incluso en una máquina remota. Está diseñado para usar Jenkins como un panel de control de tu sistema de automatización. Para más información consulta esta [página](#).

☐ **Copiar un Tarea existente**
 Copiar desde

Una vez hecho esto, configuramos el resto de parámetros de la siguiente ventana:

Proyecto nombre

Descripción

[Vista previa](#)

☐ Desechar ejecuciones antiguas

GitHub project

☐ Esta ejecución debe parametrizarse

☐ Desactivar la ejecución (No se ejecutará nuevamente hasta que el proyecto sea reactivado.)

☐ Lanzar ejecuciones concurrentes en caso de ser necesario

Opciones avanzadas del proyecto

Configurar el origen del código fuente

☐ CVS
☐ CVS Projectset
☒ **Git**

Repositories Repository URL

Branches to build Branch Specifier (blank for default):

Navegador del repositorio

URL

Ahora pasamos a configurar la parte específica de Sonar Runner. Para ello, en la sección *Ejecutar* añadimos un nuevo paso y seleccionamos *Invoke Standalone Sonar Analysis*.

Ejecutar

Invoke Standalone Sonar Analysis

JDK (Inherit From Job) ?
JDK to be used for this sonar analysis

Path to project properties php/sonar-project.properties ?

Project properties ?

JVM Options Borrar

Añadir un nuevo paso ▼

Acciones para ejecutar después.

Añadir una acción ▼

En el caso en que utilizemos Maven, podemos configurar el complemento de sonar añadiendo una nueva acción denominada *Sonar* en la sección *Acciones para ejecutar después*.

Acciones para ejecutar después.

Sonar ?

Rama ?
Propiedad opcional para "sonar.branch".

Language php

JDK (Inherit From Job) ?
JDK to be used for this sonar analysis

Versión de Maven asd ?

POM raíz ?
Por defecto "pom.xml"

MAVEN_OPTS ?
Variable de entorno para maven MAVEN_OPTS. Si no se especifica se utilizará el valor definido en la configuración global de Maven.

Propiedades adicionales ?
Propiedades adicionales para pasar como parámetros a maven (ejemplo: -Dsome.property=some.value).

☐ No usar disparadores globales ?

Borrar

Una vez esté todo configurado, tendremos el menú disponible desde el panel de administración.

Jenkins

Jenkins Laboratorio-php

[Volver al Panel de control](#)

[Estado](#) [Cambios](#) [Espacio de trabajo](#) [Construir ahora](#) [Borrar Proyecto](#) [Configurar](#) [GitHub](#)

Historia de tareas (tendencia)

#2	26-dic-2012 17:09:54
#1	26-dic-2012 17:08:10

[RSS para todos](#) [RSS para los fracasos](#)

Proyecto Laboratorio-php

Practica jenkins con sonar

[Espacio de trabajo](#) [Cambios recientes](#)

[Editar descripción](#) [Desactivar el Proyecto](#)

Enlaces Permanentes

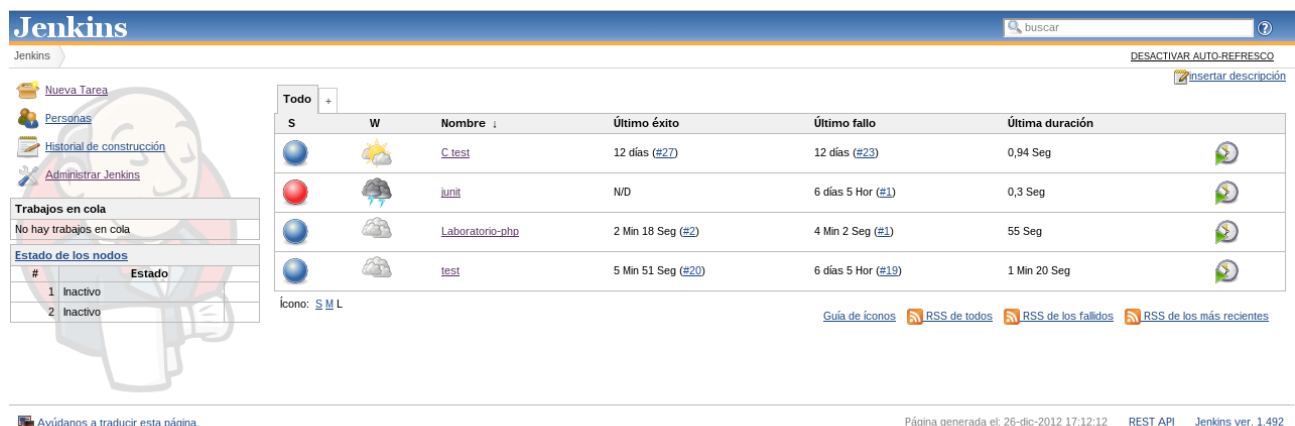
- [hace Última ejecución \(#2\), 2 Min 39 Seg](#)
- [hace Última ejecución estable \(#2\), 2 Min 39 Seg](#)
- [hace Última ejecución correcta \(#2\), 2 Min 39 Seg](#)
- [hace Última ejecución fallida \(#1\), 4 Min 23 Seg](#)
- [hace Última ejecución fallida \(#1\), 4 Min 23 Seg](#)

[Avúdanos a traducir esta página.](#)

Página generada el: 26-dic-2012 17:12:33 [REST API](#) [Jenkins ver. 1.492](#)

Ejecución

Por último, para ejecutar la tarea, simplemente pulsamos en el icono del reloj, que realizará el análisis del proyecto. Si la esfera aparece azul se ha realizado el análisis correctamente, en caso contrario, existe algún fallo.



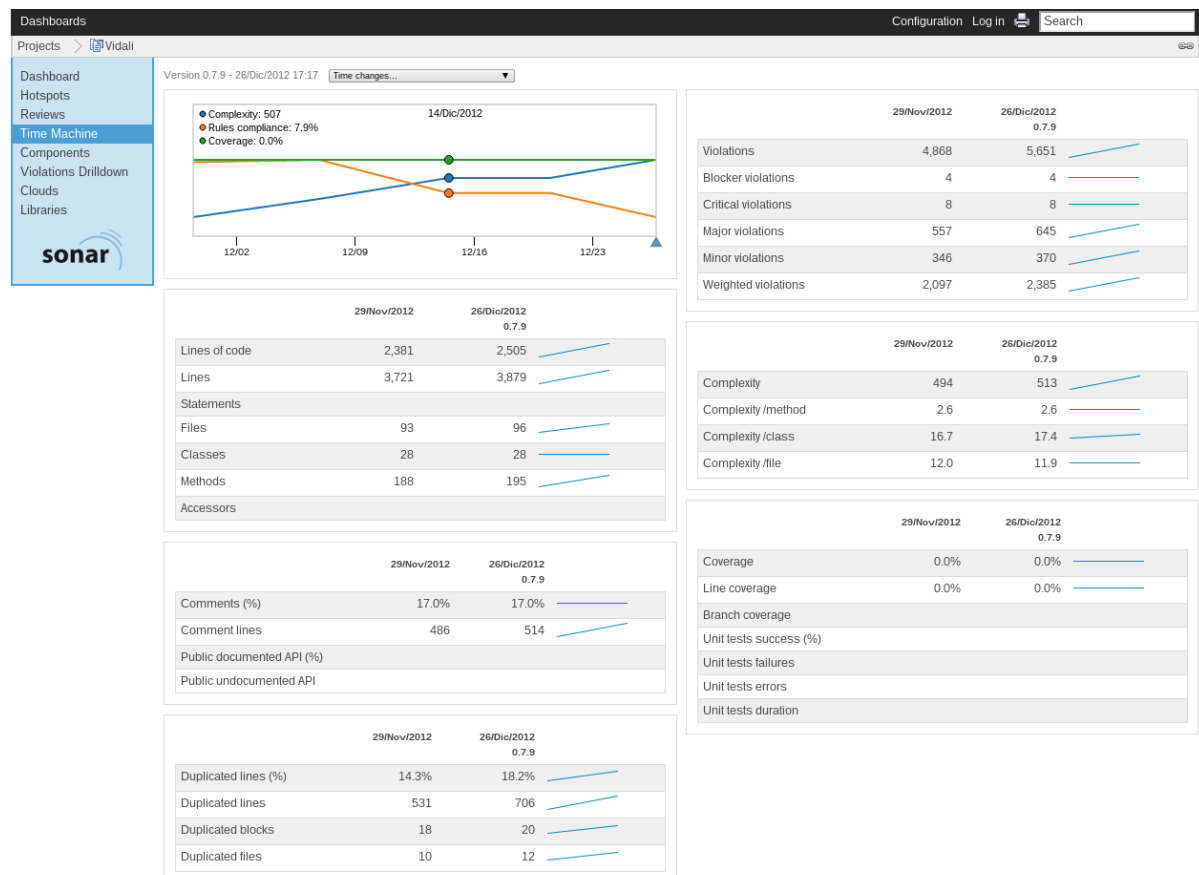
The screenshot shows the Jenkins dashboard. On the left, there's a sidebar with links like 'Nueva Tarea', 'Personas', 'Historial de construcción', and 'Administrar Jenkins'. Below this, a section titled 'Trabajos en cola' (Jobs in queue) shows 'No hay trabajos en cola' (No jobs in queue). Another section, 'Estado de los nodos' (Node status), shows two nodes: '1 Inactivo' and '2 Inactivo'. The main area displays a table of jobs with columns for 'S' (Status), 'W' (Weather icon), 'Nombre' (Name), 'Último éxito' (Last success), 'Último fallo' (Last failure), and 'Última duración' (Last duration). The jobs listed are 'C_test', 'unit', 'Laboratorio-pho', and 'test'. At the bottom, there are links for 'Guía de iconos', 'RSS de todos', 'RSS de los fallidos', and 'RSS de los más recientes'.

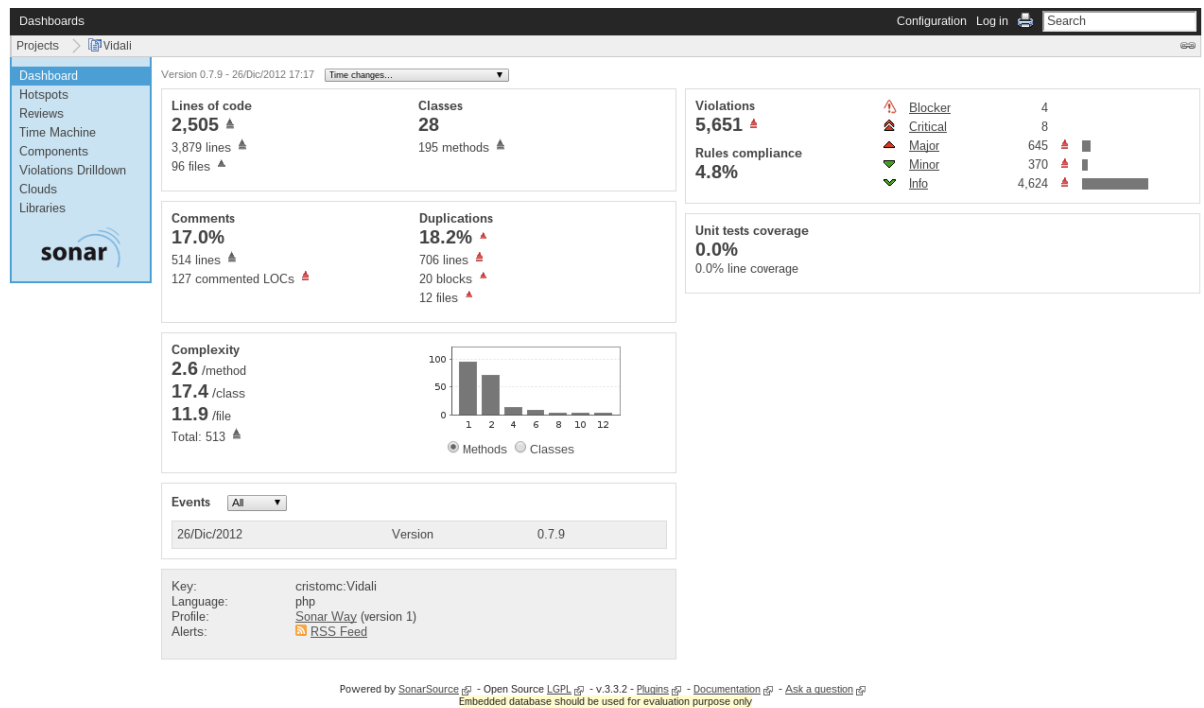
S	W	Nombre	Último éxito	Último fallo	Última duración
Blue	Sunny	C_test	12 días (#27)	12 días (#23)	0,94 Seg
Red	Cloudy	unit	N/D	6 días 5 Hor (#1)	0,3 Seg
Blue	Sunny	Laboratorio-pho	2 Min 18 Seg (#2)	4 Min 2 Seg (#1)	55 Seg
Blue	Sunny	test	5 Min 51 Seg (#20)	6 días 5 Hor (#19)	1 Min 20 Seg

Conclusiones y resultados obtenidos.

Resultados en el Sonar:

A continuación se muestran los resultados obtenidos, que nos proporciona información muy útil sobre el código analizado que nos ayuda a ver los errores cometidos en el mismo.





Conclusión:

La herramienta es muy utilizada y a lo largo del desarrollo de un proyecto nos resulta bastante útil, ya que podemos controlar el resto de las herramientas. Aunque en el caso del Sonar, se complica su funcionamiento ya que necesitamos usar Maven de manera obligatoria, puesto que sin este no podemos integrar los resultados dentro del Jenkins, lo que ocasiona instalaciones extras de las que realmente podemos necesitar.