



Integración Continua con Jenkins usando CppCheck

*Adriana Eusebio León
Moisés Lodeiro Santiago*

Laboratorio de Desarrollo y Herramientas

Índice

- 1. ¿Que es Jenkins?**
- 2. Instalación de Jenkins**
- 3. Instalación de los Paquetes necesarios.**
- 4. Creación de un Proyecto y configuración de GitHub y Cppcheck.**
- 5. Conclusión**

¿Qué es Jenkins?

En ocasiones nos encontramos con una mala gestión de un proyecto debido a que la fuente no está centralizada y se trabaja en local. De éste modo las carpetas se dispersan y es muy difícil hacer una buena integración de un proyecto libre de errores y bien programado.

Para ello se ofrece la solución de la integración continua, una metodología que va verificando errores y aciertos de un código a medida que se va desarrollando. En especial está diseñado para trabajar con repositorios (en éste caso con GitHub).

Jenkins es un servidor de integración continua Open Source. Su uso más difundido es en la generación de instancias en aplicaciones a intervalos de tiempo para ejecutar una serie de pruebas sobre los mismos de forma automática.

Se llama “de integración continua” porque permite que el código nuevo se vaya integrando y probando de a poco. Ésto permite , aparte de tener revisiones del código, el detectar temprano fallos en las aplicaciones de manera fácil y con un menor coste a la hora de solucionar problemas.

De ésta manera podremos contar con una aplicación de mejor calidad. A diferencia de otro tipo de metodología que sigue el esquema tradicional de versiones de desarrollo en hitos alejados en el tiempo ya que acumulan muchas modificaciones de código que luego insumen más tiempo de *pruebas* y corrección de errores.

Instalación de Jenkins.

Antes que nada, hay que indicar que estamos trabajando en un entorno Linux 64 bits en una distribución basada en Debian. Para poder descargar Jenkins podemos hacerlo por medio de dos vías:

La primera y más fácil es el uso de los repositorios de Linux. Iniciamos una consola y tecleamos el siguiente comando:

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

Eso, automáticamente, nos instalará todo lo necesario para ejecutar Jenkins. En el paquete viene incluida la aplicación, un servidor propio basado en Tomcat y las librerías necesarias de Java (pues el programa funciona bajo dicho lenguaje) para su correcto funcionamiento. Posiblemente la aplicación se inicie sola, en caso contrario deberemos ingresar en la misma consola el siguiente comando:

```
sudo services jenkins start
```

Para verificar que todo haya salido bien, debemos iniciar un navegador web y entrar a puerto 8080 (por defecto) en la URL <http://localhost:8080> que es la ruta por defecto de la plataforma.

El otro modo, más artesanal, es acceder a la página oficial de la plataforma Jenkins (<http://www.jenkins-ci.org>) donde deberemos ir al apartado de descargas y bajar la última versión (actualmente la 1.494).

Una vez descargado el paquete (http://pkg.jenkins-ci.org/debian/binary/jenkins_1.494_all.deb), descomprimos y ejecutamos el archivo ejecutable. Esto nos abrirá una nueva ventana donde debemos pulsar sobre “instalar”. Una vez finalice el asistente ya tenemos nuestra plataforma Jenkins instalada. Cabe destacar que la misma página de Jenkins tiene multitud de paquetes para distintas distribuciones de Linux como Debian, FreeBSD, Fedora, RedHat, etc. Incluso para MacOS y, como no, para Windows.

Nota: Suponemos que el usuario ya tiene instalado CppCheck en su equipo y que está completamente funcional. En otro caso consulte el manual correspondiente a la instalación del cppcheck.

Instalación de los paquetes necesarios.

Una vez estemos en nuestra aplicación dentro del navegador accedemos a la sección "Administrar Jenkins". Nos encontraremos con una pantalla como la siguiente:

Administrar Jenkins



[Configurar el Sistema](#)

Configurar variables globales y rutas.



[Configure Global Security](#)

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



[Actualizar configuración desde el disco duro.](#)

Descartar todos los datos cargados en memoria y actualizar todo nuevamente desde los ficheros del sistema. /til cuando se



[Administrar Plugins](#)

Añadir, borrar, desactivar y activar plugins que extienden la funcionalidad de Jenkins. **(Existen actualizaciones disponibles)**



[Información del sistema](#)

Muestra información del entorno que puedan ayudar a la solución de problemas.

Nosotros nos centraremos en la opción de "Administrar Plugins". Al acceder debemos ir a la pestaña "Todos los Plugins" y seleccionar los siguientes:

- Jenkins Cppcheck Plug-in
- Github Plugin
- Git Server Plugin
- Github Authentication Plugin

Una vez seleccionados hacemos click en la parte inferior de la pantalla donde pone "Descargar ahora e instalar después de reiniciar". Veremos una pantalla con unas esferas que se van marcando activas a medida que se van descargando. Para reiniciar Jenkins, desde una consola, tecleamos:

sudo services jenkins restart

El proceso de reinicio puede tardar. Una vez reiniciado accedemos a la página web y veremos que Jenkins está instalando los paquetes. Si no existe ningún error, podremos continuar al siguiente punto.

Creación de un Proyecto y configuración de GitHub y Cppcheck.

Si ya tenemos instalados los plugins, procedemos a crear un nuevo proyecto. Para ello, en la pantalla principal de Jenkins, haremos click en "Nueva tarea" seleccionando como parámetros el nombre del proyecto que queramos y como tipo de proyecto el "Crear un proyecto de estilo libre".

En la nueva pantalla que nos aparece dejaremos todas las configuraciones como están salvo las siguientes que hay que modificar:

Suponiendo que el usuario ya dispone de una cuenta en Github, un repositorio creado ya en la plataforma de Github cuya URL, a partir de ahora, haremos referencia con GITURL que apunta al repositorio que almacena archivos de C/C++.

En 'Configurar el origen del código fuente', seleccionaremos "Git" e ingresaremos la URL HTTPS de sólo lectura de nuestro repositorio GitHub. En Disparadores de ejecuciones, hacemos click sobre la opción 'Build when a change is pushed to GitHub'.


En ejecutar, seleccionamos en el desplegable "Ejecutar línea de comandos" y en el campo de comando, ponemos:

cppcheck --xml --errorlist *.c > cppcheck-result.xml

En la opción 'Acciones para ejecutar después' elegimos “Publish Cppcheck results” ingresando como comando:

*****/cppcheck-result.xml***

Una vez seguidos éstos pasos hacemos click en Guardar.

Si todo ha ido bien podremos hacer click en la pantalla principal sobre el botón  de nuestro proyecto para que empiece a ejecutar. Veremos una barra en la parte izquierda que nos indicará cuándo ha terminado. Si ahora entramos a nuestro proyecto podremos ver una gráfica de errores (en caso de que haya) y podremos acceder a los resultados de CPP en la opción “Cppcheck Results”.

Conclusión

Jenkins es una herramienta muy poderosa para la integración continua de proyectos pues nos permite ir verificando periódicamente y avanzar en la corrección de errores entre otras cosas. Además se encarga de trabajar también con repositorios online de modo que no hace falta estar trabajando con carpetas en local.

Además nos permite integrar muchos plugins para ir revisando calidad (como pudiera ser el Sonar), errores (Cppecheck para C/C++ y otros). Es una herramienta altamente recomendable que debería ser introducida en más sitios con el fin de mejorar la calidad.