**¿Qué es integración continua?**

La integración continua es un término que utilizó por primera vez Martin Fowler, y consiste en **hacer integraciones automáticas de un proyecto lo más a menudo posible.** Se entiende por integración la **compilación y ejecución de pruebas de todo un proyecto.** El proceso se lleva a cabo normalmente cada cierto tiempo y su función es la descarga de las fuentes desde el control de versiones, su posterior compilación, la ejecución de pruebas y generar informes.

Además, siempre se dispone de una versión para pruebas, demos o no lanzamientos anticipados. Se pueden ejecutar de forma inmediata las pruebas unitarias y hay una monitorización continua de las métricas de calidad del proyecto.

Jenkins es un sistema que corre en un servidor que es un contenedor de servlets, como hace también Apache Tomcat. Puede trabajar con herramientas de control de versiones como CVS, Subversion, Git, Mercurial, Perforce y Clearcase y es capaz de ejecutar proyectos basados en Apache Ant y Apache Maven, así como scripts de shell y programas batch de Windows.

Por último, destacar que Jenkins puede extenderse mediante plugins. Actualmente hay una gran cantidad de plugins que permiten cambiar su comportamiento o añadir nuevas funcionalidades. Entre los más utilizados se encuentran:

*Javadoc:* Este plugin añade soporte Javadoc a Jenkins. Esta funcionalidad solía ser una parte del núcleo, pero a partir de Jenkins 1.431, se separó en diferentes plugins.

*Mailer:* Permite configurar las notificaciones de correo electrónico con los resultados de los builds. Para ello es necesario configurar el servidor de correo.

*External-monitor-job:* Añade la posibilidad de controlar el resultado de los trabajos realizados externamente.

*Credentials:*Este plugin permite almacenar credenciales. Proporciona una API estándar para otros plugins para almacenar y recuperar diferentes tipos de credenciales.

*Ssh-slaves:*Este plugin le permite administrar los esclavos que se ejecutan en máquinas \* nix a través de SSH. Añade un nuevo tipo de método de lanzamiento de esclavos.

**¿Qué es Jenkins?**

Es una aplicación que supervisa ejecuciones de trabajos repetidos, como por ejemplo la construcción de un proyecto de software y posterior ejecución de una serie de pruebas automáticas.

Unos años después de que Oracle comprara Sun, la comunidad de Hudson decidió renombrar el proyecto a Jenkins, migrar el código a Github y continuar el trabajo desde ahí.

Cuando trabajamos con Jenkins hay múltiples formas de organizar el flujo de trabajo. Una posibilidad es programar una tarea en la que se compruebe el repositorio de control de versiones cada cierto tiempo. Jenkins soporta herramientas de control de versiones como [**CVS**](http://es.wikipedia.org/wiki/Concurrent_Versions_System), [**Subversion**](http://es.wikipedia.org/wiki/Subversion_(software)), [**Git**](http://es.wikipedia.org/wiki/Git), [**Mercurial**](http://es.wikipedia.org/wiki/Mercurial), [**Perforce**](http://es.wikipedia.org/wiki/Perforce)y [**Clearcase**](http://es.wikipedia.org/w/index.php?title=Clearcase&action=edit&redlink=1).

**Ventajas de jenkins**

* Tiene además otras características por las que es muy interesante y prácticamente un estándar en la industria del desarrollo de software.
* Desde Jenkins es posible ver el resultado de los test.
* Generar y visualizar la documentación del proyecto o incluso pasar una versión estable del software a un determinado entorno para ser probado o hacerle una demostración al cliente, o incluso pasar esa versión automáticamente a producción.
* Posee un historial de cambios realizados por build o versión, saber quién lo realizo y cuales archivos fueron  manipulados, y sus comentarios al respecto.
* **Se trata de una herramienta que se puede instalar en plataformas Windows, Linux o Mac**.
* Su instalación y posterior mantenimiento es muy sencillo, como es habitual con las herramientas desarrolladas en Java.
* No necesita ninguna base de datos. La interfaz es una interfaz web.

**¿Qué es Docker?**

Esta herramienta permite crear contenedores (caja que permite almacenar información, programas o aplicaciones) con la finalidad de poder guardar los ejecutables de una aplicación y poder ejecutarla a través de docker sin necesidad de instalarla localmente o de tener una versión especifica por aplicación.

Al instalar docker es posible ejecutar una aplicación software desde el contenedor de Docker, y dentro de él estarán todas las librerías y cosas que necesita dicha aplicación para funcionar correctamente.

Es posible compartir el contenedor para generar trabajos en aplicaciones de programación como java; anteriormente era necesario contar con las mismas versiones de cada herramienta para poder ejecutarla, por medio de docker solo es necesario compartir el contenedor, abrirlo y ejecutarlo.

Una de las funciones más evidente, es posible encapsular todas las herramientas y ejecutables necesarios para una aplicación de esta forma el programador tendrá la seguridad de estar trabajando desde su servidor local, así mismo al momento de ejecutar los cambios la configuración se realiza en el mismo entorno en el que se han hecho todas las pruebas.

**Ventajas y funciones de Docker**

Virtualizar con Docker nos ofrece una serie de ventajas respecto a hacerlo con máquinas virtuales convencionales:

Portabilidad: Todos los contenedores creados en docker son portables, por esto es posible llevarlos fácilmente de un computador a otro sin necesidad de descargar el programa o aplicación que se vaya a utilizar.

Ligereza: Al no virtualizar un sistema completo, sino solo lo necesario, el consumo de recursos es mínimo. El ahorro de recursos es de alrededor de un 80%.

Autosuficiencia: Por medio de Docker es posible realizar casi todo, por esto los contenedores solo requieren el mínimo de información, por ejemplo, aquellas librerías, archivos y configuraciones necesarias para poder realizar su función.

Un sistema de contenedores Docker se compone principalmente de 5 elementos:

*Demonio:* Es el proceso principal de la plataforma.

*Cliente:* Es el binario que constituye la interfaz y que permite al usuario interactuar con el Demonio.

*Imagen:* Plantilla utilizada para crear el contenedor para la aplicación que queremos ejecutar.

*Registros:* Directorios donde se almacenan las imágenes, tanto de acceso público como privado.

*Contenedores:* Carpetas donde se almacena todo lo necesario (librerías, dependencias, binarios, etc) para que la aplicación pueda ejecutarse de forma aislada.

La ventaja más notoria y considerable es que por medio de docker será más sencillo y ligero compartir los avances y desarrollos del software a desarrollar teniendo en cuenta que como se menciona anteriormente no será necesario que el programa contenga con la misma versión o sea compatible con un sistema operativo exclusivo, ahora solo será necesario contar con docker instalado y en el contenedor que necesita información como librería y nombre de aplicación para su utilidad.

Bibliografía

<https://www.redeszone.net/2016/02/24/docker-funciona-la-virtualizacion-contenedores/>

<http://www.javiergarzas.com/2015/07/que-es-docker-sencillo.html>

<https://openwebinars.net/blog/que-es-jenkins-introduccion/>

https://testeandosoftware.com/jenkins-servidor-de-integracion-continua-gratuito/