

Integració contínua d'un projecte Canigó

A qui va dirigit

Aquest document va dirigit principalment a arquitectes d'aplicacions Canigó que vulguin utilitzar la metodologia d'integració contínua.

Versió de Canigó

Els passos descrits en aquest document són d'aplicació a totes les versions 2.x de Canigó.

Introducció

La integració contínua (*Continuous Integration*, CI) és una metodologia informàtica que consisteix en fer integracions automàtiques d'un projecte, tant sovint com sigui possible amb la finalitat de detectar possibles errades. S'entén per integració la compilació (build) i l'execució de tests de tot un projecte.

El procés s'executa cada cert temps (hores, dies) i consisteix en descarregar-se els fonts del gestor de versions (ja sigui CVS, Subversion, etc.), compilar-los, executar els tests i generar informes.

Sovint, la integració contínua està associada amb les metodologies d'eXtreme Programming (XP) i desenvolupament àgil.

Context

Un projecte de desenvolupament de software amb control de versions i automatització de la fase de compilació té la necessitat d'integrar el codi amb freqüència per a reduir els riscos pel que fa a problemes d'integració final del projecte.

Problema

La integració de codi en el treball de tot un equip de desenvolupament després de cicles llargs resulta una tasca feixuga. Integracions poc freqüents poden generar retards en la planificació i problemes de qualitat en el producte. Sense integracions periòdiques no és possible certificar diàriament l'estabilitat del producte.

Aspectes a tenir en compte:

- els problemes d'integració idealment s'han de detectar en el moment en què s'insereixen en el codi que pot generar-los
- els desenvolupadors no tenen l'hàbit de pujar el codi al controlador de versions amb freqüència
- el procés de compilació pot retardar-se i arribar a ser un impediment per a la construcció i integració de codi

Solució

La integració contínua.

El model ideal d'integració contínua permet que la construcció i execució de proves es realitzi cada cop que el codi canvia o és enviat al gestor de versions.

El procés de build diari minimitza els riscos d'integració perquè els problemes es detecten contínuament. També es redueix el risc de baixa qualitat de codi, ja que un conjunt de proves bàsic permet que les funcionalitats principals de l'aplicació siguin avalades constantment.

El procés d'integració contínua es fa efectiu quan el build és generat amb èxit. En el cas que el procés de compilació falli, la prioritat serà la correcció dels errors. Generalment, un error en aquesta fase té lloc quan aquest no compila o no passa les proves automatitzades.

El procés funciona de la manera següent:

Integració contínua d'un projecte Canigó

- els desenvolupadors fan modificacions en el codi font, compilen i executen les proves unitàries automatitzades i pugem el codi al controlador de versions.
- L'eina d'integració contínua, periòdicament, verifica si hi ha hagut canvis en la darrera versió.
- Si es detecten modificacions, l'eina d'integració contínua extreu tot el codi font i compila.
- Si compila, l'eina d'integració contínua executa les tasques que tingui definides: compilar, executar proves unitàries, generació de resultats de les proves...
- L'eina d'integració contínua actualitza les dades a la web del projecte amb tots els resultats de l'execució
- L'eina té la possibilitat d'enviar missatges a l'equip informant del resultat del procés.

Integració contínua amb projectes Canigó

En la pròpia Oficina Tècnica de Canigó es fa integració contínua de Canigó fent servir el software *Continuum*, desenvolupada per *Apache*.

Actualment hi ha una iniciativa del CTTI per oferir una plataforma d'integració contínua basada en una altra eina de *Integració Contínua* anomenada *Hudson*. Aquesta iniciativa no està pensada inicialment per dur a terme la integració en fase de desenvolupament.

En aquest document es descriu com es pot dur a terme la integració contínua del projecte en tems de desenvolupament. L'eina triada es *Continuum*, tot i que la majoria de conceptes són d'aplicació per Hudson.

Continuum es una eina que ens proporciona els principis bàsics de la integració contínua:

1. Detecció de canvis al controlador de versions
2. Compilació del projecte programada o immediata
3. Execució de tests
4. Notificació del succés o fracàs de la compilació

Totes aquestes funcionalitats fan de *Continuum* una eina més que capaç d'oferir-nos una bona integració contínua al nostre projecte, a més a més, cal sumar-li que es tracta d'una eina pensada per a projectes de tipus *Maven*, com és el cas del projectes Canigó.

Per afegir un projecte al *Continuum* haurem d'importar el pom i aquest l'associarem a un grup.

Add Maven 2.0+ Project

POM Url:	<input type="text"/>
Username:	<input type="text" value="admin"/>
Password:	<input type="password" value="*****"/>
	<input type="checkbox"/> Use SCM Credentials Cache, if available
	Enter the URL to the Maven 2 POM. Provide the username and password if it is a secured resource.
	OR
Upload POM:	<input type="text"/> <input type="button" value="Examinar..."/>
	Enter the local filename of the Maven 2 POM to upload (works only for a single project without modules).
Project Group:	<input type="text" value="Defined by POM"/>
	<input type="checkbox"/> For multi modules project, load only root as recursive build
Build Definition Template:	<input type="text" value="Default"/>
	<input type="button" value="Add"/> <input type="button" value="Cancel"/>

Pantalla per afegir un projecte Maven a Continuum a partir d'un pom

Integració contínua d'un projecte Canigó

Com podem observar a la imatge anterior i amb un requadre vermell, podem deixar que el mateix pom faci un grup nou (o l'afegeixi a un existent) o el podem afegir a un grup nosaltres mateixos seleccionant-lo a la llista.

Un cop em importat el pom, obtenim una pantalla semblant a la següent que conté la informació dels grups disponibles, en aquest cas parlarem d'un grup anomenat *test*.

Name	Group Id				Total
test	test	0	1	0	1

Grup de projectes al Continuum

Les icones que podem veure a les columnes de la dreta ens indiquen ràpidament l'estat de l'última execució, si el projecte s'ha construït correctament, amb avisos o ha fallat.

Si seleccionem el grup, se'ns mostra una altra pantalla amb la informació i accions que podem fer amb aquest juntament amb els projectes que el formen:

Project Group Summary | Members | Build Definitions | Notifiers

Project Group Informations

Project Group Name: test

Project Group Id: test

Description: test

Group Actions

Default Build Definition Add M2 Project

Project Group Last Build Result Overview

Success : 0 Errors : 0 Failed : 1

Member Projects

<input type="checkbox"/>	Project Name	Version	Build	Last Build Date					
<input type="checkbox"/>	Connector Root	2.3.6-SNAPSHOT	29	sep 30, 2009 05:49:14 PM CEST					

Default Build Definition

Informació del grup

La pantalla està organitzada per pestanyes:

- *Project Group Summary*: Informació i accions del grup
- *Members*: Usuaris i permisos per aquest grup
- *Build Definitions*: Goals de Maven a executar
- *Notifiers*: Notificadors de quan un projecte s'ha construït bé o malament

Nosaltres treballarem bàsicament amb la primera pestanya que és un petit resum de les altres. Si seleccionem un dels projectes (en el nostre cas només hi ha un però pot haver-hi més), veurem dos seccions diferenciades:

Integració contínua d'un projecte Canigó

1. Configuració de la connexió al control de versions:

Continuum Project	
Project Name:	Connector Root
Version:	2.3.6-SNAPSHOT
SCM Url:	scm:cvs pserver cvs @localhost /opt/cvsroot canigo2009/connectors
SCM Branch/Tag:	
Group:	test
Last Build Date:	Sep 30, 2009 05:49:14 PM CEST
<input type="button" value="Edit"/> <input type="button" value="Build Now"/>	

Configuració de la connexió al control de versions

Ho configurarem segons la configuració del nostre servidor.

2. Goals de Maven a executar:

Build Definitions										
Goals	Arguments	Build File	Schedule	Profile	From	Build Fresh	Default	Description	Type	
clean source:jar install	--batch-mode -Dmaven.test.skip=true	pom.xml	DEFAULT_SCHEDULE		GROUP	false	true	default maven2 buildDefinition	maven2	
<input type="button" value="Add"/>										

Goals de Maven a executar

Veiem que, a més a més de la comanda Maven (goals), també podem definir els paràmetres que li passarem a Maven.

També cal remarcar la columna *Schedule*, es tracta de quan volem que s'executi la comanda que podrem configurar a l'apartat *Schedules* del menú *Administration*. Actualment a Canigó tenim els següents:

Schedules						
Name	Description	Quiet Period	Cron Expression	Max Job Time	Active	
DEFAULT_SCHEDULE	Run hourly	0	0 0 * * * ?	3600	true	
EVERY_DAY	Fire at 12pm (noon) every day	0	0 0 12 * * ?	0	true	
EVERY_NIGHT_1am	Fire at 1:00 every day	0	0 0 1 * * ?	0	true	
EVERY_NIGHT_2am	Fire at 2:00 every day	0	0 0 2 * * ?	0	true	
EVERY_NIGHT_3am	Fire at 3:00 every day	0	0 0 3 * * ?	0	true	
EVERY_NIGHT_5am	Fire at 5:00 every day	0	0 0 5 * * ?	0	true	
NO_SCHEDULE	Nomes execucio manual	0	0 0 12 1 1 ? 2099	0	true	
<input type="button" value="Add"/>						

Programacions d'execucions dels projectes

Com podem veure, les programacions es basen en expressions de Cron de **NIX*, a l'apartat de referència trobareu un enllaç amb més informació al respecte.

Un cop hem configurat el projecte amb les configuracions del nostre entorn i els *goals* que volem que s'executin, tenim dos opcions, executar el *goal* manualment o esperar a que s'executi mitjançant el

Integració contínua d'un projecte Canigó

disparador que hem configurat. En tots dos casos, i si la configuració ha estat correcta, s'executaran els següents passos:

1. Comprovació i actualització de la copia de treball amb la última del control de versions
2. Execució del *goal* de Maven
3. Execució dels tests en el cas de que no s'hagi indicat el contrari
4. Notificació de l'estat de l'execució als membres indicats

Referències

Continuum: <http://continuum.apache.org>

Maven: <http://maven.apache.org/>

Hudson: <https://hudson.dev.java.net/>

Expressions Cron: http://en.wikipedia.org/wiki/CRON_expression