

Continuous Integration with Hudson

Creating More Robust Code,
Developing More Rapidly,
Improving Testing,
Deploying More Reliably,
Gaining Confidence,
and
Saving Money

2010-01-20

Chris Shenton

chris@koansys.com

<http://koansys.com/tech/hudson/>

What's Continuous Integration?

- Martin Fowler sez:

Continuous Integration is a software development practice where members of a team integrate their work frequently, usually each person integrates at least daily - leading to multiple integrations per day. Each integration is verified by an automated build (including test) to detect integration errors as quickly as possible. Many teams find that this approach leads to significantly reduced integration problems and allows a team to develop cohesive software more rapidly.

Continuous Integration: Some Free Systems

- Cruise Control
- BuildBot
- Hudson

How Developers Code

```
while ( not done ) {  
    write some code;  
    test by poking at it tediously;  
    fix the bugs you see;  
}  
deploy;  
cross your fingers nobody finds new bugs;  
return;
```



Isn't There Enough World Suffering??



How to Improve It?

- What if we could hire a new developer?
- Make them test all day long
- Every time we make a code change
- Track and report progress
- Give metrics, graphs on improvements
- Report failures quickly

That New Guy is Hudson

- Doesn't draw a salary
- Doesn't get bored
- Tells you about problems it finds
- Doesn't make fun of you or your bugs



Basic Continuous Integration

- Check the code repo for changes every few minutes
- Build or compile the code
- Run your tests: unit, regression, etc
 - of course you have to have tests to run!
- Alert if problems

Hudson Benefits

- Never gets bored doing builds and tests
- Catches problems fast
- Alerts developers while code is fresh in their minds
- Prevents bugs from propagating downstream
- Cheaper to fix bugs earlier, before QA or Deployment

Expensive, Hard to Install -- NOT!

- Free download (13 seconds)
 - `$ curl -L -O http://hudson.gotdns.com/latest/hudson.war`
- Run it (16 seconds until ready)
 - `$ java -jar hudson.war`
- Configure though the web!
 - `http://localhost:6543/`

Alerts by Mail Alerts and Commands by IM

10:01 ChrisShenton
!status

10:01 SMD Hudson
status of all projects:
sunra-trunk: last build: 218 (2 min 16 sec ago): SUCCESS: <http://10.9.8.7:6543/job/sunra-trunk/218/>

10:01 Away Message: Working: 1 out of 2 executors are busy.

10:03 SMD Hudson came back

11:48 SMD Hudson
status of all projects:
sunra-trunk: last build: 220 (33 min ago): SUCCESS: <http://10.9.8.7:6543/job/sunra-trunk/220/>
webcompare: last build: 3 (20 min ago): SUCCESS: <http://10.9.8.7:6543/job/webcompare/3/>

11:50 ChrisShenton
!health

11:53 SMD Hudson
health of all projects:
sunra-trunk: Health [Cobertura Coverage: 100% (0/0) Conditionals (100%), Build stability: No recent builds failed.
(100%): <http://10.9.8.7:6543/job/sunra-trunk/220/>
webcompare: Health [Build stability: 1 out of the last 3 builds failed.(66%): <http://10.9.8.7:6543/job/webcompare/3/>

Helpful Plugins

- Subversion: code repo checkout
- Cobertura: test coverage reporter
- Maven: many plugins available
- Selenium: functional, click-testing
- Monitor for Android: status on your phone
- Chuck Norris: motivational help :-)

Immediate Benefits

- Easy to put projects under Hudson
- Find build problems, quicker to fixed
- Find test breakage, quicker fixed
- Reports on increasing test coverage
- Motivates us to increase test coverage

Critical Prerequisites: Automated Build

- No human intervention allowed
 - ant, maven
 - setup.py, pip, buildout
 - (every language has one)

Critical Prerequisites: Code that Tests Your Code

- Faster than clicking around your app
- Can run without firing up the whole stack (web framework, security, ...)
 - unit tests: junit, nose, ...
 - functional tests: selenium, abott, ...
- Run locally before checking in your code:
“don’t break the build”

This Simple Test Saved Me HOURS!

```
class TestUrlManglers(unittest.TestCase):

    def test_normalize_url(self):

        from webcompare import Walker
        w = Walker("fakeurl", "fakeurl", ignoreres=['\\?.*', '#.*', '/RSS.*', '<bound.*'])

        self.assertEquals(w._normalize_url("http://example.com/something"),
                          "http://example.com/something")

        self.assertEquals(w._normalize_url("http://example.com/something/RSS"),
                          "http://example.com/something")

        self.assertEquals(w._normalize_url("http://example.com?queryString"),
                          "http://example.com")

        self.assertEquals(w._normalize_url("http://example.com#fragment"),
                          "http://example.com")
```

Recap

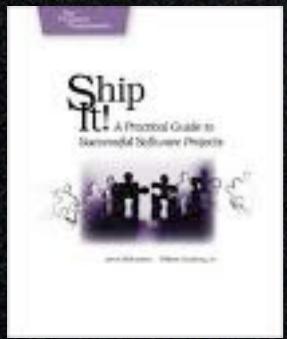
- Hudson watches the code repo for commits
 - Checks out code
 - Builds it with your build ‘script’
 - Runs your tests
 - Reports problems by email and IM
 - Tracks metrics, if available

These Books are Helpful

- They're not specifically about Hudson, SVN, Java, Python, ColdFusion, .net, ...
- They cover real programmer hands-on practice with code repos, tests, continuous integration



- Pragmatic Project Automation: How to Build, Deploy, and Monitor Java Applications (Mike Clark)
- Ship It! A Practical Guide to Successful Software Projects (Richardson, Gwaltney) [excellent for SW project managers]



Danger! Live Demo!

[Back to Dashboard](#)

[Status](#)

[Changes](#)

[Workspace](#)

[Build Now](#)

[Delete Project](#)

[Configure](#)

[Coverage Report](#)

[Subversion Polling Log](#)

Build History [\(trend\)](#)

#	Build	Date
239	#239	Jan 19, 2010 1:43:31 PM
233	#233	Jan 19, 2010 1:16:19 PM
228	#228	Jan 19, 2010 11:23:14 AM
225	#225	Jan 15, 2010 5:06:03 PM
224	#224	Jan 15, 2010 2:50:00 PM
223	#223	Jan 15, 2010 2:40:32 PM
222	#222	Jan 15, 2010 1:06:03 PM
221	#221	Jan 15, 2010 12:06:03 PM
220	#220	Jan 15, 2010 11:15:30 AM
216	#216	Jan 14, 2010 6:26:00 PM

Project sunra-trunk

[add description](#)

Chuck Norris's keyboard doesn't have a Ctrl key because nothing controls Chuck Norris.

Code Coverage

Packages 100% Files 100% Classes 100% Lines 90% Conditionals

Build	Classes	Conditionals	Files	Lines
#204	100%	100%	100%	100%
#205	100%	100%	100%	100%
#207	100%	100%	100%	100%
#208	100%	100%	100%	100%
#209	100%	100%	100%	100%
#210	100%	100%	100%	100%
#211	100%	100%	100%	100%
#212	100%	100%	100%	100%
#213	100%	100%	100%	100%
#214	100%	100%	100%	100%
#215	100%	100%	100%	100%
#216	100%	100%	100%	100%
#220	100%	100%	100%	100%
#221	100%	100%	100%	100%
#222	100%	100%	100%	100%
#223	100%	100%	100%	100%
#224	100%	100%	100%	100%
#225	100%	100%	100%	100%
#228	100%	100%	100%	100%
#239	100%	100%	100%	100%

[Coverage Report](#)

[Workspace](#)

[Recent Changes](#)

Live Demo: walk through a project

Hudson » sunra-trunk ENABLE AUTO REFRESH

[Back to Dashboard](#)

[Status](#)

[Changes](#)

[Workspace](#)

[Build Now](#)

[Delete Project](#)

[Configure](#)

[Coverage Report](#)

[Subversion Polling Log](#)

[Build History \(trend\)](#)

[#242 Jan 20, 2010 9:06:34 AM](#)

Changes

#242 (Jan 20, 2010 9:06:34 AM)

8387. White space cleanup — [Daniel Greenfeld / detail](#)

#241 (Jan 19, 2010 6:05:52 PM)

8381. Enabled django-db-log
About time to roll this in: now we get logs nicely batched in the database — [Chris Adams / detail](#)

8380. Left side nav: fixed generated HTML
The original logic from mptt.utils generated an extra level of list closure, which was harmless but broke HTML validation. — [Chris Adams / detail](#)

8379. missing left nav image for Earth Mission — [wkeeter / detail](#)

#240 (Jan 19, 2010 5:05:52 PM)

Coverage Breakdown by File

Name	Classes	Lines
sunra.apps.blog.search_indexes.py	100% 1/1	86% 19/22
nasascience.apps.imageoftheday.models.py	100% 1/1	62% 23/37
sunra.apps.contact.urls.py	100% 1/1	100% 7/7
sunra.apps.cms.extensions.page_sorting.py	100% 1/1	89% 25/28
sunra.apps.custommedia.tests.py	100% 1/1	100% 67/67
sunra.apps.cms.search_indexes.py	100% 1/1	81% 39/48

Live Demo: Create New Hudson Instance

Job name

Monitor an external job
This type of job allows you to record the execution of a process run outside Hudson, even on a remote machine. This is designed so that you can use Hudson as a dashboard of your existing automation system. See [the documentation for more details](#).

Build a free-style software project
This is the central feature of Hudson. Hudson will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

Build multi-configuration project (alpha)
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

Build a maven2 project
Build a maven2 project. Hudson takes advantage of your POM files and drastically reduces the configuration.

Copy existing job
Copy from