CohesionForce
TECHNOLOGY & SERVICES

EFD

How to use Hudson

April 16, 2014

Summary:

This presentation is an overview of the capabilities provided by the Eclipse Hudson project. We will discuss potential uses, setup, and best practices for using Hudson.

Experience Level: Beginner

This is a broad overview of the use of Hudson and the overall concept of Continuous Integration.

What is Continuous Integration?

This is the part where you have to participate with some sort of an answer.

Continuous Integration:

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.

from Martin Fowler:
http://martinfowler.com/articles/continuousIntegration.html

Hudson in a Nutshell:

Hudson is a Continuous Integration Server.

Hudson runs in a web server like Weblogic, Glassfish, Tomcat, JBoss, IIS, JRebel, or a built-in Jetty server.

Hudson executes Jobs.

Hudson provides a plug-in architecture for additional capability.

Hudson at Eclipse:

Hudson has been an Eclipse project since 2011:

http://www.eclipse.org/hudson/

There's even a free book available:

http://wiki.eclipse.org/The_Hudson_Book

Eclipse on Hudson

Eclipse is a prolific use of Hudson to continuously integrate and deliver over 250 Eclipse projects.

You can see a lot of the projects building on Hudson here:
https://hudson.eclipse.org/hudson/

Some Eclipse projects, like the Platform project have their own Hudson Instance:
https://hudson.eclipse.org/platform/

Testing with Hudson

Test results that are part of a Hudson build can be collected and reported. The Hudson analysis plugin can also report on test result trends and cause a build to fail if a user defined number of tests fail.

The test results could be from any number or supported testing frameworks, or as simple as a collection of compiler warnings.

https://hudson.eclipse.org/hudson/view/Modeling/job/emf-compare-1.3/lastCompletedBuild/testReport/

Scaling with Hudson

Hudson provides a master/slave concept where hudson builds can be distributed among a set of machines.

Machines can be connected through SSH or by JNLP.

Eclipse uses slave machines running Windows, Mac, & other OS to test target platform builds.

https://hudson.eclipse.org/hudson/computer/windows7tests/