***Instalação do Jenkins e Phing para projetos PHP***

Plugins do Jenkins

Neste ponto, você pode seguir o guia do Sebastian a partir de “Required Jenkins Plugins”, ou seja, você deve instalar alguns plugins do Jenkins para que, quando nosso código chegar lá, já esteja tudo pronto – os plugins são: Checkstyle, Cloverphp, DRY, HtmlPublisher, jDepend, Plot, PMD, Violations e xUnit. Acrescente à lista o Phing – sim, vamos mudar um pouco a receita do tio Sebastian (odeio simplesmente copiar o trabalho alheio).

Ferramentas PHP

Testes Unitários – PHPUnit

Estilo de código – PHP\_CodeSniffer

Código Duplicado (o famoso Ctrl+C, Ctrl+V – que pode criar pontos de evolução distintos para a mesma funcionalidade) – PHPCPD

Bagunça no código (buscamos aqui pontos de refatoração) – PHPMD

Medição e análise estrutural – PHPLoc

Diversas Métricas de Software, adicionais ao item anterior – PHP\_Depend

Na instalação que fizemos de PHP, contemplamos a PEAR/PECL (por isso instalamos o xdebug via pecl). Caso você não tenha instalado, faça isso – não dá pra continuar sem. Bom, tio Sebastian também deixou as coisas bem adiantadas e na seção “Required PHP Tools” do link de seu Template ele tem uma maneira muito boa de instalar todos de uma vez:

# pear upgrade PEAR;

# pear config-set auto\_discover 1;

# pear install pear.phpqatools.org/phpqatools pear.netpirates.net/phpDox

Neste ponto, você deve ter todas as ferramentas citadas logo acima. Mãããããss queremos o Phing para ser nosso construtor de Projetos… Para tal, execute o seguinte:

# pear channel-discover pear.phing.info

# pear install -a phing/phing

Pronto. Phing também instalado. Não vou entrar em detalhes por enquanto sobre ele. Por hora, basta saber que ele executará tarefas de maneira automatizada.

Configurando o Projeto

O Phing, assim como seu primo javeiro Ant, funciona a partir de um arquivo de descrição de build. Tio Sebastian preparou um para Ant, e aqui está um para Phing. Crie um arquivo build.xml na raiz de seu projeto com o conteúdo deste gist.

Vale atentar para as linhas de 6 a 8 – nelas, configuramos os diretórios principais para nosso build.

Também para o target na linha 72, onde configuramos o codesniffer para o padrão de código do Symfony2 … mas este não está na lista default de padrões do phpcs, que tem entre outros os padrões PEAR E Zend). O que não nos impede de adicionar padrões – e instalei o padrão do Symfony2 usando este projeto.

Por último, mas não menos importante, estou supondo que sua suíte de testes se utilize de um bootstrap(.php) E um (phpunit.)xml de configuração. Adicionalmente ao template do Sebastian, eu coloquei uma verificação: se o xdebug estiver instalado (no caso de nossa VM está), vamos executar o teste completo com relatórios de cobertura, senão somente a execução dos testes unitários.

Usando o Template de Job (adaptação)

Realize o checkout do template no diretório de jobs do Jenkins

# cd /var/lib/jenkins/jobs

# git clone git://github.com/sebastianbergmann/php-jenkins-template.git php-template

# chown -R jenkins:nogroup php-template/

Reinicie o Jenkins

# /etc/init.d/jenkins restart

Vá para a interface web do Jenkins (se você ainda não mudou nada, deve estar em http://<hostname>.local:8080) e clique em “New Job”

Insira um nome de Job (Job Name)

Selecione “Copy Existing Job” e insira no input que aparece “php-template”

Clique em OK

Entre no projeto (se já não estiver nele) e acesse a opção “configure”

Desabilite a opção “Disable Job”

Configure o Job para seu controle de versão (Source Code Management) (há plugins dos mais variados, inclusive um plugin que utiliza um filesystem sem controle de versão – para o GIT, eu precisei instalar adicionalmente os plugins Jenkins Git e Git Parameter)

Configure o Build Trigger – por hora, use o Poll SCM (agendamento do “fetch” do SCM)

Em Build, exclua o item “ant” e em “Add Build Step” selecione “Invoke Phing Targets”, e depois selecione a Versão default do Phing

Clique em Save

Onde podem ocorrer problemas? Permissionamento – verifique se o Jenkins tem acesso ao projeto.

O Jenkins tem também muitos truques na manga. O Jogo de Integração Contínua (onde quem melhora mais o software ganha mais pontos – brincadeira que pode motivar uma equipe a melhorar o próprio trabalho), Ferramentas mil no tocante à distribuição do software, entre tantas outras coisas… você vai perder um tempo só analisando os plugins disponíveis.

Lógico que este artigo não se direciona somente a quem utiliza VMs ou a quem configurou uma pelo artigo anterior. Com poucos ajustes, você pode levantar um Jenkins em qualquer lugar.