**What is??**

* Bamboo é um servidor CI desenvolvido puramente em Java pela empresa informática Atlassian (que desenvolveu o Jira, Bitbucket, e muitos mais programas).
* É usada para o implantação automática de aplicações, usando triggers de maneira a simplificar o trabalho da equipa de software. O Bamboo disponibiliza três tipos de triggers: Commits, intervalos de tempo e manualmente.
* Ainda em relação ao seu lançamento, foi lançada há pouco tempo, em Maio deste ano, o que revela um bocado de imaturidade, com alguns habituais bugs de lançamento. (imaturidade, atualizações constantes, bugs iniciais).
* Sendo desenvolvido em Java (cross-platform), é possível instalar a ferramenta nos 3 mais usados sistemas operativos, Linux, Windows e Mac OS.
* Em relação às linguagens que suporta, esta suporta diversas linguagens de programação, como Java,.NET, PHP, JavaScript, etc.
* Em relação ao seu uso, infelizmente não é open-source pelo que se tem de pagar para usufruir da ferramenta.
* Apesar de na sua versão beta a ferramenta podia ser usada em CLOUD, a Atlassian decidiu descontinuar este serviço. Sendo que assim apenas é disponibilizada em modo Servidor, tendo as empresas que arranjar os servidores.

**How it works??**

Isto funciona da seguinte maneira:

Primeiro começamos pelo mais básico: as **Tasks**.

As **Task’s** consistem em pequenas execuções de código (testes, implantação, execução de scripts). Estas são realizadas em sequência numa ordem especificada pelo utilizador. Caso uma falhe, as seguintes deixam de ser executas.

(Exemplo de Tasks: “Teste unitários, Code Checkout, Script”)

Os **Jobs** permitem organizar as **Tasks** (das dependentes entre si, e as independentes) e ao contrário destas, os Jobs podem ser executados em paralelo, sendo cada um atribuído a um agente. Um Job processa um conjunto de tarefas e têm o propósito de dar a possibilidade ao utilizador a execução de diferentes tarefas, como testes funcionais e unitários.

(Exemplo de Jobs: “Integration Tests, UI tests”)

As **Stages** possibilitam organizar os Jobs que são independentes entre si e que como consequência podem correr diferentes Jobs em paralelo. Basicamente é um estado do projeto, sendo que só se deve mudar de **Stage** só depois do anterior **Stage** terminar. Uma **Stage** termina, só depois de todos os seus Jobs terminarem.

(Exemplo de uma Stage: “Package Wenapp, Deploy to QA, Run Tests, Deploy to Production, Smoke Test Production”)

Um **Plano** contém um conjunto de **Stages**, e este tem uma hierarquia. Ou seja, um plano pai pode ter vários planos filhos, e só depois do plano pai terminar com sucesso é que os planos filhos podem iniciar. Um plano usa sempre o mesmo repositório, em que deverá ser configurado pelo utilizador logo no início.

Um utilizador pode iniciar um plano através de um commit, manualmente ou até mesmo calendarizar um (normalmente, em alturas em que a equipa não está a usar o servidor).

**// Agente**: Um agente, semelhante a CPU só que virtual, tem o objetivo de correr um trabalho. Pode ser local, na máquina hospedeira do Bamboo, ou Remoto que pode ser uma outra máquina.

*Conclusão: Um projeto pode ter, um ou vários planos. Um plano tem várias stages que correm sequencialmente. Uma Stage tem vários Jobs que correm em paralelo. Um Job contém várias task que correm sequencialmente.*

//Na criação do plano pudemos definir as permissões de cada utilizador, como ver, editar, build, e administração.

//Developer’s experientes recomendam para equipas grandes uma utilizador dedicado apenas ao servidor SI, gerindo os builds.

**Pricing Slides:**

Os preços do Bamboo baseiam-se nos agentes, em vez dos habituais utilizadores. Quanto mais agentes, mais processos podem correr concorrentemente, seja localmente ou remotamente, e por isso mais performance temos e menores serão os tempos de build/implantação.

Apesar do número de agentes locais ser ilimitado, visto que estão todos a ser corridos na mesma Java Virtual Machine que o Bamboo, quanto maior o número de agentes for, pior será a performance da máquina. Segundo a Atlassian, os limites para a maioria das máquinas é entre 30 e 40 agentes.

Dividir o Jobs em outros Jobs, e corre-los em paralelo distribuindo em agentes é uma boa técnica de aumentar o tempo de build.

Infelizmente não é possível responder quantos agentes o servidor do cliente pode aguentar. A Atalassian recomenda que sejam feitos diferentes testes no servidor ao correr múltiplos agente em paralelo, obtendo-se sempre o tempo de build. Após a recolha dos tempos, a equipa deve concluir qual o limite máximo do servidor.

*No fim de contas, qual dos 2 pacotes devo escolher?*

Se se tratar de uma equipa com menos de 20 elementos recomenda-se escolher o small teams packets. O melhor é iniciar com o primeiro pacote, e consoante as necessidades, faz se upgrades.

*E quais os critérios que devo usar para escolher o número de agentes? 1 é pouco?*

Infelizmente, só com a realização de testes manuais nos servidores dos clientes é que se pode responder à pergunta. Mas para isso temos a versão Free-Trial para ambos os pacotes.

*Conclusão, recomendam a ferramenta?*

Depende, com base no estudo desta ferramenta, parece-me que é bastante útil para grandes empresas de software, com recursos de topo de gama, especialmente em empresas que já usem outros serviços da Atlassian como Jira, Hipchat, Bitbucket e Cloverage (que é para testes). Para equipas pequenas, ou ainda em solo, não me parece viável devido ao seu preço e complexidade.

Notifications: Permite notificar através de SMS, Email e muitos mais (RSS, IM).

**Table**

IDE Integration: Existem plugins para Eclipse e IntelliJ – em que permite monitorizar builds.

Version Control: Yup!

Hosted:Only!

Open Source: No!

Documentation: Yes.

**À parte**, a atlassian recentemente providenciava o CI Bambo em Clouds, contudo ainda em Abril deste ano informou a todos os seus clientes, do termino deste serviço. ISTO É MAU! “Devido à complexidade deste serviço e à incapacidade da empresa em fornece-lo com um mínimo de qualidade, a Atlassian decidiu focar-se apenas em melhorar o Bambo Server para grandes equipas“