**STEP 7: Môi trường vận dụng**

**(05/10/2015-09/10/2015)**

Trong phần này mình sẽ đi tìm hiểu về 2 phần đó là

**Phần 1: CI Tool**

**1.CI là gì ? 継続インテグレーション**

CI là viết tắt của Continuous Integration hay còn gọi là tích hợp liên tục.

Nghe đến cái tên CI – Continuous Integration(tích hợp liên tục) là 1 cái gì đó rất cao siêu và hoành tráng, nhưng ở bài viết này, mình sẽ giải thích 1 cách đơn giản nhất về CI.

Theo IBM thì: “CI – Tích hợp liên tục là **phương pháp phát triển phần mềm đòi hỏi các thành viên trong nhóm tích hợp công việc thường xuyên**. Mỗi ngày, các thành viên đều phải theo dõi và phát triển công việc của họ ít nhất 1 lần.Việc này được 1 nhóm khác kiểm tra tự động, nhóm này sẽ tiến hành kiểm thử truy hồi để phát hiện lỗi nhanh nhất có thể.Cả nhóm thấy rằng phương pháp tiếp cận này giúp giảm bớt các vấn đề về tích hợp hơn và cho phép phát triển phần mềm gắn kết nhanh hơn”



Ngày xửa ngày xưa, Nam còn là sinh viên ngành IT của trường B. Mỗi lần code, làm bài tập nhóm đối với Nam là một cực hình. Cả team ngồi phác thảo ra từng module nho nhỏ, sau đó chia ra, mạnh ai về nhà code. Cuối tuần, cả nhóm hẹn nhau ra quán cafe để “ráp code”, cóp các phần đã làm qua USB, bỏ vào một project chung. Mỗi lần “ráp code”, chương trình không build được, một núi lỗi xuất hiện, cả nhóm phải  hì hục mất nguyên buổi chiều để sửa. “Ráp code” trở thành một cơn ác mộng đối với Nam và các bạn trong nhóm.

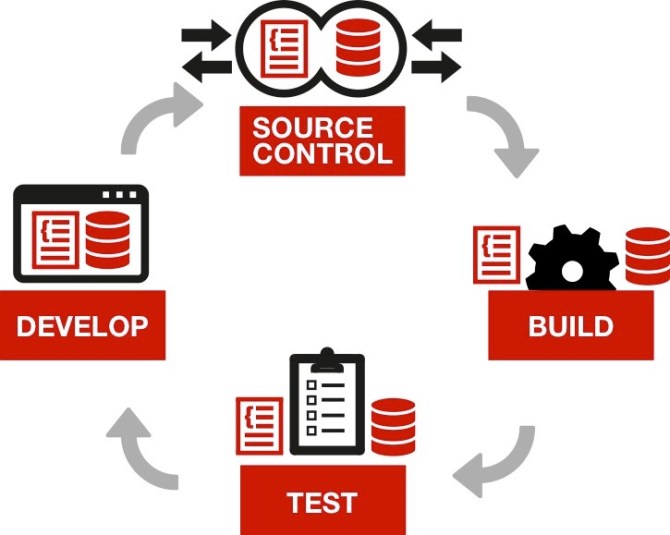
Ra trường rồi đi làm, Nam đi phỏng vấn xin việc, được nhận vào công ty phần mềm C. Ở đây, mọi người đã biết dùng SVN nên không còn cảnh phải copy code qua lại nữa. Từ “ráp code” cũng biến mất mà thay vào đó là từ “tích hợp”. Tuy nhiên, team của Nam vẫn còn làm việc khá ẩu. Mỗi sáng, các thành viên trong team update code từ SVN/Git về, code say sưa, sau đó commit code lên trước khi về nhà.

Đôi khi code không build được, cả team lại nháo nhào “truy tìm thủ phạm”: anh Phạm Văn A sửa code mà quên commit file mới lên, chị Lê Thị B sửa connection string, … Đời nhiều khi còn trái ngang hơn, anh A sửa code, chị B sửa code làm phần của Nam chạy bị lỗi, thế là Nam lãnh đủ. Mỗi cuối tuần, cả team lại phải OT để tích hợp và sửa lỗi.

[](https://toidicodedao.files.wordpress.com/2015/08/fail.jpg)

Phần mềm không bán được vì quá nhiều lỗi, release chậm, công ty cũ phá sản, Nam phải đi tìm việc mới. Với khả năng của mình, Nam được nhận vào công ty C. Công ty này khá chuyên nghiệp, có áp dụng Agile, CI – tích hợp liên tục. Nhờ có CI, team của Nam hiện tại không còn gặp những rắc rối khi tích hợp:

* Mỗi khi có người commit code, **hệ thống CI sẽ tự lấy code từ SVN, thực hiện build. Hệ thống sẽ gửi mail thông báo cho toàn bộ thành viên nếu như build bị lỗi. Cả nhóm chỉ việc đọc mail, xem ai là người commit revision đó, và “nắm đầu” thủ phạm, bắt hắn sửa lỗi**.
* Project của Nam được viết Unit Test rất cẩn thận, đầy đủ. Khi anh A, chị B sửa code, commit lên, hệ thống sẽ build và chạy lại toàn bộ các Unit Test. Nếu có case nào bị fail, cả team sẽ nhận được thông báo, và anh A, chị B phải nhanh chóng fix code để các case này pass.
* **Việc tích hợp được diễn ra hàng ngày, nhiều lần trong ngày. Mỗi khi ai đó commit code làm hư build hoặc gây lỗi, cả team có thể giải quyết vấn đề NGAY LẬP TỨC**. “Tích hợp” không còn là nỗi ác mộng mà trở thành chuyện thường ngày đối với Nam.

[](https://toidicodedao.files.wordpress.com/2015/08/app1.jpg)

Nói tóm lại, **CI – tích hợp liên tục là phương pháp mà các team Agile(Agile là 1 mô hình phát triển phần mềm linh hoạt) sử dụng để đảm bảo code của toàn dự án luôn build được, luôn chạy đúng (Pass toàn bộ các test case)**. Hiện tại, hầu như các công ty phần mềm đều áp dụng CI, thông qua một số framework như: TFS, TeamCity, Hudson, Jenkin, Travis, … Tự trang bị cho mình những kiến thức về CI sẽ khiến bạn có giá hơn trong mắt nhà tuyển dụng nhé.

Nếu muốn hiểu chuyên sâu về Continous Integration và áp dụng, bạn nên xem một bài viết chi tiết. cụ thể về Continous Integration của Martin Folwer:

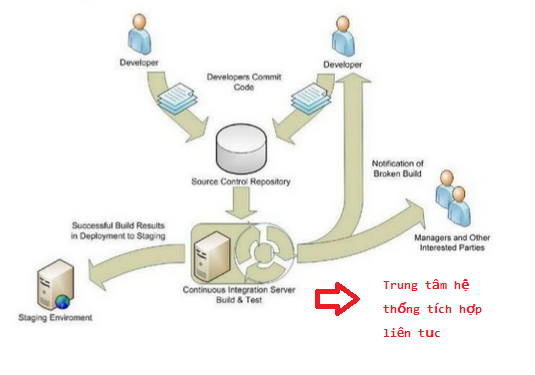
<http://www.martinfowler.com/articles/continuousIntegration.html>

**2.Hệ thống tích hợp liên tục**

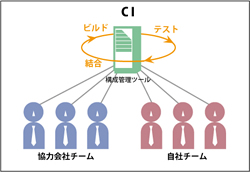
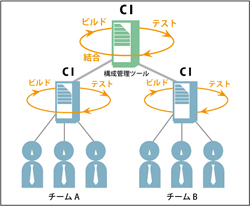
2.1.Định nghĩa hệ thống tích hợp liên tục(Continuous Integration System)

Tích hợp liên tục là 1 hoạt động phát triển phần mềm mà ở đó các thành viên của 1 nhóm sẽ tích hợp công việc của họ với nhau liên tục, thường thì các thành viên sẽ tích hợp công việc ít nhất 1 lần trong ngày. Mỗi sự tích hợp sẽ được kiểm tra bằng một công cụ build tự động(bao gồm cả test) để phát hiện ra lỗi tích hợp càng sớm càng tốt.

Như vậy 1 hệ thống phát triển theo mô hình agile(mô hình phát triển phần mềm nhanh) và có tích hợp CI thì được gọi là 1 hệ thống tích hợp hợp liên tục



Sử dụng CI để quản lý chất lượng và tiến độ phát triển ứng dụng

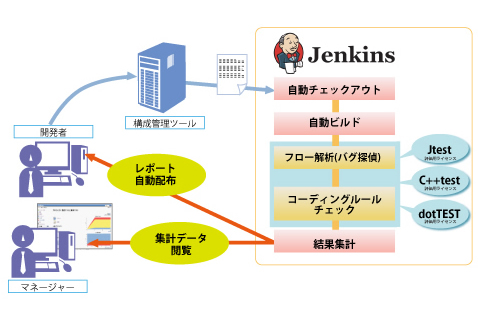
 

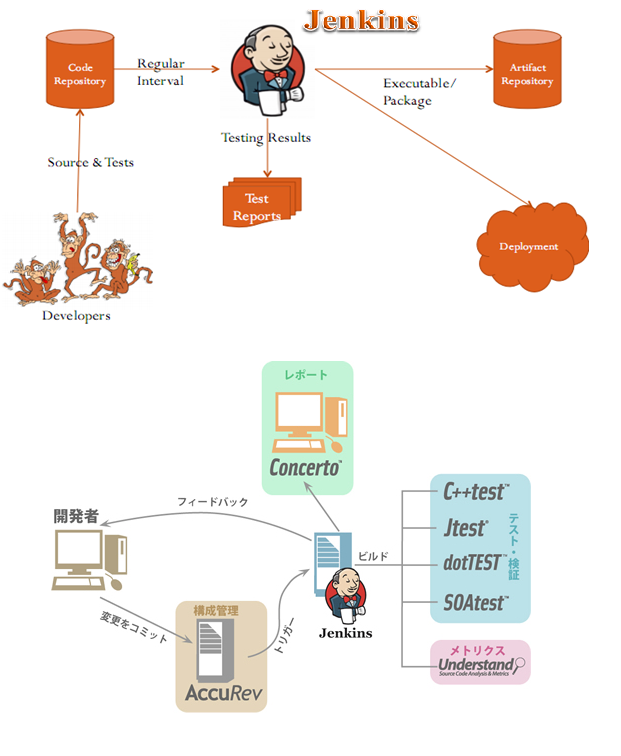
**3.Giới thiệu Jenkins**

3.1. Sơ lược về Jenkins

Jenkins là CI Tool.Nó đại diện cho khái niệm CI

Jenkins là 1 ứng dụng web application mã nguồn mở, đóng vai trò máy chủ build và test của hệ thống tích hợp liên tục.Jenkins là CI Server, giữ vị trí trung tâm cho mọi hoạt động của hệ thống





CI(継続的インテグレーション)とは、コンパイル、ビルド、テスト、デプロイなどを繰り返し行ってコードの品質改善や納期短縮を目指す開発手法です。そして、JenkinsはCIのジョブを定期的に実行し、実行結果をフィードバックしてくれる便利なツールなのです。

Tạm dịch:

CI là kỹ thuật phát triển nhằm rút ngắn thời gian deploy và cải thiện chất lượng code bằng cách tiến hành lặp đi lặp lại việc compiler, build , test , deploy. Jenkins là 1 tool tiện lợi cho việc thực thi và feedback kêt quả thực thi định kì các công việc trong CI.

Đặc điểm của Jenkins:

+ Dễ cài đặt và sử dụng

+ Đa nền tảng và hỗ trợ nhiều công nghệ phát triển phần mềm

+ Dễ dàng liên kết với các công cụ khác trong hệ thống tích hợp liên tục thông qua các plugin

+ Miễn phí , mã nguồn mở

Như vậy ta có thể hiểu khái quát về việc sử dụng Jenkins trong hệ thống phát triển phần mềm

Khi developer commit code của họ lên trên trình quản lý code(như là Github, SVN,..) thì trình quản lý code sẽ thực hiện tự động build và test bằng cách sử dụng Jenkins.Jenkins sau khi build và test sẽ gửi lại kết quả đến developer và manager. Phát sinh lỗi sẽ được phát hiện và yêu cầu sửa chữa ngay lập tức. Như vậy quá trình đó sẽ lặp đi lặp lại trước khi Jenkins thông báo ko có lỗi và cho phép deploy sản phẩm.



3.2.Cài đặt Jenkins trong CentOS

Ta sẽ tham khảo 1 số bài viết để thực thi CI bằng Jenkins cho Rails App

Tham khảo :

+ <http://www.maepachi.com/blog/entry/144>

+ <http://qiita.com/pakiran/items/458fae106566c6c3c963>

**+B1**: Cài đặt java. Để có thể sử dụng Jenkins thì việc cài đặt Java 6 trở lên là điều quan trọng. Vì vậy trước hết ta sẽ kiểm tra xem java đã được cài đặt hay chưa, nếu được cài đặt rồi mà phiên bản thấp hơn java6 thì ta sẽ cài đặt lại.

Test java version

>sudo yum list installed

#=> java-1.7.0-openjdk.x86\_64

Nếu không ta sẽ remove bản java cũ và cài đặt lại bản mới như sau:

>sudo yum remove java

>sudo yum install java-1.7.0-openjdk java-1.7.0-openjdk-devel

**+B2**: Ta cài đặt Jenkins

Thực thi các lệnh sau:

>sudo wget -O /etc/yum.repos.d/jenkins.repo http://pkg.jenkins-ci.org/redhat/jenkins.repo

>sudo rpm --import http://pkg.jenkins-ci.org/redhat/jenkins-ci.org.key

>sudo yum install Jenkins

**+B3**: Khởi động Jenkins

>sudo service jenkins start

Check status xem Jenkins có đang hoạt động hay không

>sudo service jenkins status

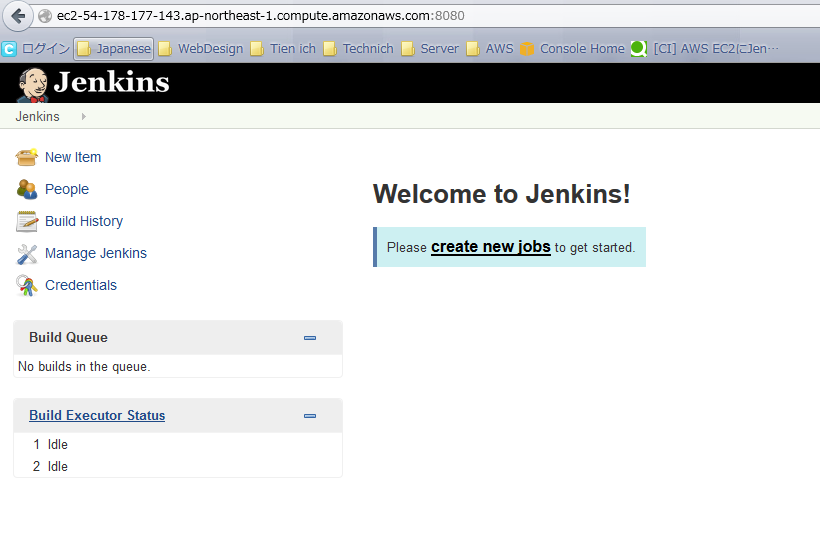
Bây giờ muốn test ta sẽ truy cập vào link sau:

<http://xx.xx.xx.xx/8080>

Trong đó xx.xx.xx.xx là địa chỉ ip của server đã cài đặt Jenkins. Nếu ta sử dụng trong AWS EC2 thì đó chính là public DNS.

VD: <http://ec2-54-178-177-143.ap-northeast-1.compute.amazonaws.com:8080/>

Kết quả:



3.3.Sử dụng Jenkins với Rails App

Tham khảo: <http://www.buildinsider.net/enterprise/jenkins/>

**Bài 1**: <http://www.buildinsider.net/enterprise/jenkins/01>

Bài này hướng dẫn tạo 1 job trên Jenkins.

**Bài 2**: <http://www.buildinsider.net/enterprise/jenkins/02>

Bài này hướng dẫn cài đặt Ruby với rvm và tạo rails app

**3.3.1.Tạo rails app**

>rails new circle

>rails generate scaffold user name:string age:integer

>rake db:create

>rake db:migrate

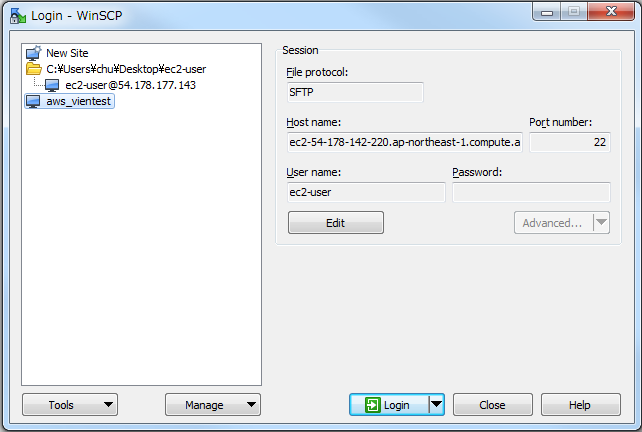
>rails server –b 0.0.0.0

Như vậy là ta đã có 1 rails app chạy trên host của AWS

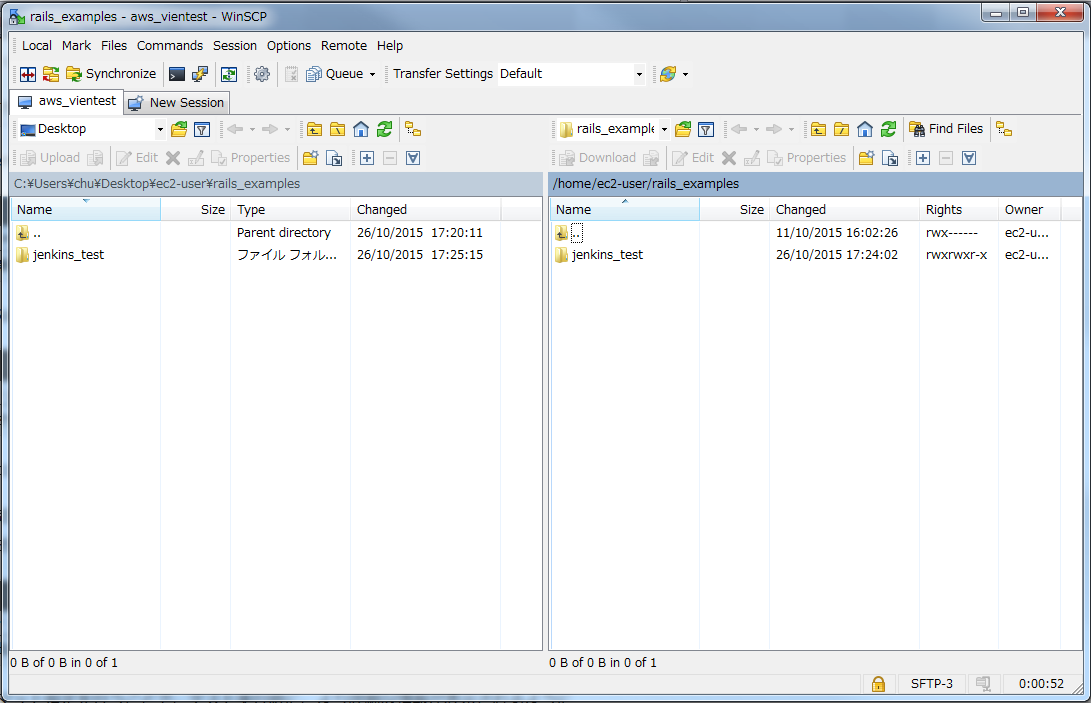
VD: <http://public_dns_on_aws/users/>

P/s: Sử dụng WinSCP để đồng bộ code giữa môi trường máy ảo và Windows

Down và cài đặt WinSCP. Tạo 1 site để có thể đăng nhập vào máy ảo thông qua giao thức ssh



Trong đó hostname chính là địa chỉ IP hoặc DNS, username thì ta để mặc định là ec2-user. Giao thức SSH sử dụng cổng 22 để truy cập vào máy ảo. Trong phần Edit ta thiết lập đường dẫn của file ppk đã được tạo khi sử dụng Putty. Vậy là ta có thể sử dụng rồi.



Để đồng bộ thì ta tạo 1 folder và kéo những folder bên trong máy ảo vào bên phía của windows(máy ảo phía bên tay phải, môi trường win bên trái). Còn ngược lại, khi đã code trên win và muốn test trong máy ảo thì ta chỉ cần click vào folder muốn đồng bộ sau đó chọn upload, nhớ chú ý đường dẫn tương ứng với thư mục trong máy ảo.

**3.3.2.Cài đặt gem rspec**

Để có thể cài đặt gem này, ta làm như sau

>gem install rspec-rails

Sửa file gem

==============Gemfile====================

group :development, :test do

gem 'rspec-rails', '~> 2.0'

end

=========================================

>bundle install

Sau đó ta vào trong project vừa tạo và thực thi:

>rails generate rspec:install #Cài đặt gem vào project

Trong file .rspec vừa sinh ra ta sửa nội dung như sau:

--color --format documentation # Khi thực thi test thì sẽ hiển thị nội dung thông tin cụ thể

**3.3.1.Chỉnh sửa application và thực thi unit test**

Bây giờ ta sẽ chỉnh sửa project rails vừa tạo và thực hiện các unit test

Tài liệu tham khảo:

【Tiếng Nhật】

<https://www.techmatrix.co.jp/quality/cisolution/cisolution2.html>

<https://www.techmatrix.co.jp/quality/cisolution/cisolution1.html>

<http://www.techmatrix.co.jp/quality/cisolution/service/>

<http://itpro.nikkeibp.co.jp/article/COLUMN/20130927/507264/>

<http://softwaretest.jp/quality/>

<http://www.slideshare.net/SumiKoichiro/ci-continuous-integration-servicematome>

<http://www.buildinsider.net/enterprise/jenkins/02>

【Tiếng Việt】

<https://toidicodedao.wordpress.com/2015/08/27/giai-thich-don-gian-ve-ci-continuous-integration-tich-hop-lien-tuc/>

<http://www.slideshare.net/MinhTriLam/jenkins-30241736>