

## บทที่ 3 การติดตั้งระบบ KnowledgeVolution

KnowledgeVolution เป็นซอฟต์แวร์ที่ถูกพัฒนาขึ้นมาบนพื้นฐานของเทคโนโลยี Ruby on Rails การติดตั้ง KnowledgeVolution จึงจำเป็นต้องติดตั้งซอฟต์แวร์พื้นฐานก่อน ดังรายละเอียดต่อไปนี้

### 1. ระบบปฏิบัติการสำหรับติดตั้ง

KnowledgeVolution ถูกพัฒนาขึ้นมาบนระบบปฏิบัติการ MacOS X และ Ubuntu ส่วนเครื่องแม่ข่ายที่ติดตั้งใช้งานจริง ทำงานอยู่บนระบบปฏิบัติการ FreeBSD และ Ubuntu จึงมีคำแนะนำว่าการนำ KnowledgeVolution ไปใช้งาน ควรติดตั้งบนระบบปฏิบัติการจำพวกยูนิกซ์หรือลินุกซ์จะเหมาะสมกว่า แต่ทั้งนี้สามารถติดตั้งบนระบบปฏิบัติการ Windows ได้เช่นเดียวกัน

อนึ่ง วิธีการติดตั้งทั้งหมดในเอกสารเล่มนี้ ตั้งอยู่บนสมมติฐานว่าเป็นระบบปฏิบัติการลินุกซ์ Ubuntu

### 2. ติดตั้งซอฟต์แวร์ที่จำเป็น ได้แก่

2.1 GCC ทำหน้าที่เป็นตัวแปลภาษาซี ติดตั้งโดยใช้คำสั่ง

```
$ sudo apt-get install build-essential
```

2.2 Ruby ทำหน้าที่เป็นตัวแปลภาษารูบี้ ติดตั้งโดยใช้คำสั่ง

```
$ sudo apt-get install ruby ri rdoc libyaml-ruby libzlib-ruby irb ruby1.8-dev memcached libopenssl-ruby
```

2.3 RubyGems ทำหน้าที่เป็นตัวจัดการแพ็คเกจของ Ruby ติดตั้งโดยใช้คำสั่ง

```
$ sudo apt-get install gems
```

2.4 MySQL ทำหน้าที่เป็นตัวจัดการฐานข้อมูล ติดตั้งโดยใช้คำสั่ง

```
$ sudo apt-get install mysql-server libmysqlclient12-dev libmysql-ruby1.8
```

2.5 ImageMagick ทำหน้าที่เป็นตัวการจัดเกี่ยวกับรูปภาพ ติดตั้งโดยใช้คำสั่ง

```
$ sudo apt-get install imagemagick libmagick9-dev
```

2.6 Subversion ทำหน้าที่บริหารจัดการรุ่นของซอร์สโค้ด ติดตั้งโดยใช้คำสั่ง

```
$ sudo apt-get install subversion
```

2.7 Apache ทำหน้าที่เป็นเว็บเซิร์ฟเวอร์ ติดตั้งโดยใช้คำสั่ง

```
$ sudo apt-get install apache2 apache2.2-common apache2-mpm-prefork apache2-utils
```

2.8 Rails ทำหน้าที่เป็นโครงสร้างพื้นฐานของ KnowledgeVolution ติดตั้งโดยใช้คำสั่ง

```
$ sudo gem update --system
```

```
$ sudo gem install rails --include-dependencies
```

2.9 ติดตั้งซอฟต์แวร์ที่จำเป็นอื่นๆ

```
$ sudo gem install ruby-memcache --include-dependencies
```

```
$ sudo gem install memcache-client --include-dependencies
```

```
$ sudo gem install cached_model --include-dependencies
```

```
$ sudo gem install mongrel --include-dependencies
```

```
$ sudo gem install gettext --include-dependencies
```

```
$ sudo gem install redcloth --include-dependencies
```

```
$ sudo gem install color-tools --include-dependencies
```

```
$ sudo gem install rmagick --include-dependencies
```

```
$ sudo gem install erubis --include-dependencies
```

```
$ sudo gem install capistrano --include-dependencies
```

```
$ sudo gem install termios --include-dependencies
```

หากเจอข้อผิดพลาดว่า

```
ERROR: While executing gem ... (Gem::GemNotFoundException)
```

```
Could not find ... (> 0) in any repository
```

แสดงว่าแหล่งเผยแพร่ของ Gems repository มีปัญหาชั่วคราว ให้รอสักครู่แล้วลองติดตั้งใหม่

สำหรับเครือข่ายที่ใช้พร็อกซี ต้องกำหนดค่าพร็อกซีให้ gem ด้วยรูปแบบ

```
$ sudo gem install rails --include-dependencies -p http://cache.psu.ac.th:8080
```

### 3. สร้างไคลเอนต์สำหรับ KnowledgeVolution

สามารถเลือกที่เก็บได้ตามความเหมาะสม ในที่นี้จะนำไปเก็บไว้ใน /var/www/html สร้างไดเรกทอรีโดยใช้คำสั่ง

```
$ sudo mkdir /var/www/html
```

จากนั้นเข้าไปทำงานในไดเรกทอรี html ด้วยคำสั่ง \$ cd /var/www/html

ขั้นตอนต่อไปคือการติดตั้ง KnowledgeVolution ผ่านทาง Subversion โดยใช้คำสั่ง

```
$ sudo svn checkout svn://svn.usablelabs.com/kv/trunk kv
```

#### 4. สร้างฐานข้อมูลสำหรับ KnowledgeVolution

ในกรณีนี้ใช้ MySQL เป็นตัวจัดการฐานข้อมูล การสร้างฐานข้อมูลต้องใช้คำสั่ง

```
$ mysql -u root -p
```

```
$ (กรอกรหัสผ่านของ MySQL root)
```

```
mysql> create database kv_production;
```

```
mysql> grant all privileges on kv_production.* to 'kv'@'localhost' identified by '1234567';
```

```
mysql> quit;
```

เป็นการสร้างฐานข้อมูลชื่อ 'kv\_production' จากนั้นทำการกำหนดสิทธิ์ให้กับผู้ใช้ชื่อ kv ที่มีรหัสผ่านเป็น '1234567'

#### 5. แก้ไขไฟล์ /var/www/html/kv/config/database.yml

ให้ยกเลิกส่วนของการติดตั้งแบบอื่นๆ เช่น development และ test ให้เหลือแต่ production และแก้ไขให้ส่วนของ production เป็นดังนี้

```
production:
```

```
  adapter: mysql
```

```
  database: kv_production
```

```
  username: kv
```

```
  password: 1234567
```

```
  host: localhost
```

```
  encoding: utf8
```

จากนั้นสร้างตารางเก็บข้อมูลโดยใช้คำสั่ง

```
$ cd /var/www/html/kv
```

```
$ export RAILS_ENV=production
```

```
$ rake db:migrate
```

## 6. เริ่มการทำงานของ KnowledgeVolution

```
$ cd /var/www/html/kv
```

```
$ mongrel_rails start -e production -p 8000
```

ทดสอบการทำงานโดยให้เว็บเบราว์เซอร์ชี้มาที่ <http://localhost:8000> ถ้าขึ้นหน้าโฮมเพจ KnowledgeVolution แสดงว่าการติดตั้งเสร็จสมบูรณ์

สามารถปิดการทำงานของ Mongrel ได้โดยการกดแป้น Ctrl+C

## 7. เพิ่มประสิทธิภาพด้วยการปรับแต่งให้ Mongrel Cluster ทำงานร่วมกับ Apache

แก้ไขไฟล์ `/etc/apache2/sites-available/default`

```
<VirtualHost *:80>
```

```
DocumentRoot /var/www/html/kv/public
```

```
<Directory "/var/www/html/kv/public">
```

```
Options FollowSymLinks
```

```
AllowOverride None
```

```
Order allow,deny
```

```
Allow from all
```

```
</Directory>
```

```
RewriteEngine On
```

```
RewriteCond %{DOCUMENT_ROOT}%{REQUEST_FILENAME} !-f
```

```
RewriteRule ^/(.*)$ balancer://mongrel_cluster%{REQUEST_URI} [P,QSA,L]
```

```
<Proxy *>
```

```
Order allow,deny
```

Allow from all

</Proxy>

<Proxy balancer://mongrel\_cluster>

BalancerMember http://localhost:8000

BalancerMember http://localhost:8001

BalancerMember http://localhost:8002

</Proxy>

</VirtualHost>

จากนั้นทำการสั่งให้โมดูลที่สำคัญเริ่มทำงาน ด้วยคำสั่ง

```
$ sudo a2enmod rewrite
```

```
$ sudo a2enmod proxy_http
```

```
$ sudo a2enmod proxy_balancer
```

```
$ sudo /etc/init.d/apache2 force-reload
```

จากนั้นทำการแก้ไขไฟล์ `/var/www/html/kv/config/mongrel_cluster.yml`

```
user: www-data
```

```
cwd: /var/www/html/kv
```

```
port: "8000"
```

```
environment: production
```

```
group: www-data
```

```
address: localhost
```

```
pid_file: log/mongrel.pid
```

```
servers: 3
```

เริ่มการทำงานด้วยคำสั่ง

```
$ cd /var/www/html/kv
```

```
$ mongrel_rails cluster::start
```

ยกเลิกการทำงานด้วยคำสั่ง

```
$ cd /var/www/html/kv
```

```
$ mongrel_rails cluster::stop
```

8. ปรับแต่งสำหรับใช้งานจริง โดยเปลี่ยนค่า 'localhost' ให้เป็นหมายเลขไอพี หรือชื่อโดเมนของเครื่องแม่ข่าย

โดยทำการแก้ไขไฟล์

```
/etc/httpd/conf/httpd.conf
```

```
/var/www/html/kv/config/mongrel_cluster.yml
```

9. การ deploy ด้วย Capistrano

สร้างไฟล์สองไฟล์เก็บไว้ในไดเรกทอรีสำหรับ deploy ซึ่งแยกต่างหากจาก /var/www/html/kv โดยมีรายละเอียดดังนี้

ไฟล์แรกตั้งชื่อว่า Capfile โดยเพิ่มบรรทัดต่อไปนี้

```
load 'deploy' if respond to?(:namespace) # cap2
```

```
load 'config deploy'
```

ไฟล์ที่สองตั้งชื่อว่า config\_deploy.rb โดยในไฟล์นี้จะทำการกำหนดบทบาทของเครื่องแม่ข่ายตัวอย่างเช่น

```
set :repository, "svn://svn.usablelabs.org/kv/trunk"
```

```
role :web, "demo.usablelabs.org"
```

```
role :app, "demo.usablelabs.org"
```

```
#role :db, "demo.usablelabs.org", :primary => true
```

```
set :deploy_to, "/var/www/html/kv"
```

```
set :user, "sysadmin"
```

```
ssh_options[:port] = 34
```

:repositoy คือแหล่งที่เก็บ KnowledgeVolution รุ่นล่าสุด

:web คือที่ตั้งของเครื่องแม่ข่ายที่ใช้เป็นเว็บเซิร์ฟเวอร์ โดย :web, :app, :db สามารถอยู่รวมในเครื่องเดียวกัน หรือแยกจากกันเพื่อประสิทธิภาพที่ดีขึ้นก็ได้

:deploy คือไดเรกทอรีปลายทางที่จะใช้ deploy KnowledgeVolution