



July 22 – 24, 2007 • Portland, Oregon

# Ruby on Rails on Ubuntu

Bradley Taylor  
Rails Machine, LLC  
[bradley@railsmachine.com](mailto:bradley@railsmachine.com)

Co-presented by

O'REILLY  CANONICAL

## Bio

- Owner of Rails Machine
- Virtual appliance hosting
- Author of `mongrel_cluster` and `railsmachine` gems
- Savannah, GA
- [blog.railsmachine.com](http://blog.railsmachine.com)

# Deploying Rails

- The Stack
- Installation
- Integration
- Application configuration
- Best Practices

## The Stack

- Web tier
- Application tier
- Database tier
- Same or different servers
- Independently scalable
- Alternatives for each tier

## Web Tier

- `Apache 2.2`
- `mod_proxy_balancer`
- `mod_rewrite`
- `Worker MPM`
- `Server static files`
- `Proxies and load balances dynamic requests`

## Application Tier

- `Ruby 1.8.5+` (language)
- `Ruby Gems` (libraries)
- `Ruby on Rails 1.2+` (web framework)
- `Mongrel 1.0.1` (application server)
- `mongrel_cluster 1.0.2+` (mongrel process manager)

## Database Tier

- MySQL 5.0
- MySQL native ruby gem
- Others: PostgreSQL, SQLite3

# Installation

- Web
  - `apt-get install apache2-mpm-worker`
  - `a2enmod rewrite`
  - `a2enmod proxy_balancer`
- App
  - `apt-get install build-essential ruby1.8 ruby1.8-dev ri1.8 rdoc 1.8 rubygems`
  - `gem update --system`
  - `gem install mongrel mongrel_cluster rails`

## Installation (2)

- `mkdir /etc/mongrel_cluster`
- `cp /usr/lib/ruby/gems/1.8/gems/mongrel_cluster-1.0.2/resources/mongrel_cluster /etc/init.d/`
- `chmod +x /etc/init.d/mongrel_cluster`
- `update-rc.d -f mongrel_cluster defaults`
- Database
  - `sudo apt-get install mysql-server libmysqlclient15-dev`
  - `sudo gem install mysql`

# Integration

- Deployment user
  - `adduser rails`
- Application directory
  - `mkdir /var/www/rails`
  - `chown rails:rails /var/www/rails`

# Configuration

- Apache

- `/etc/apache2/sites-available/your_app`

```
<Proxy balancer://your_app_cluster>
  BalancerMember http://127.0.0.1:8000
  BalancerMember http://127.0.0.1:8001
</Proxy>
```

```
RewriteEngine On
# Rewrite index to check for static
RewriteRule ^/$ /index.html [QSA]
# Rewrite to check for Rails cached page
RewriteRule ^([\^.]*)$ $1.html [QSA]
# Redirect all non-static requests to cluster
RewriteCond %{DOCUMENT_ROOT}/%{REQUEST_FILENAME} !-f
RewriteRule ^/(.*)$ balancer://your_app_cluster%{REQUEST_URI}
[P, QSA, L]
```

- `a2ensite`  
`your_app`

## Configuration (2)

- application directory
    - `mkdir /var/www/rails/your_app`
    - `chown rails:rails /var/www/rails/your_app`
  - `mongrel_cluster`
    - `/etc/mongrel_cluster/your_app`
- ```
---  
cwd: /var/www/rails/your_app  
port: "8000"  
environment: production  
address: 127.0.0.1  
pid_file: /var/run/mongrel_cluster/your_app.pid  
servers: 2  
user: rails  
group: rails
```

## Configuration (3)

- `config/database.yml`

```
production:
  adapter: mysql
  database: your_app
  username: your_app
  password: your_passwd
```

- MySQL database and user

- `CREATE DATABASE your_app;`

- `GRANT ALL PRIVILEGES ON your_app.* TO 'your_app'@'localhost' IDENTIFIED BY 'your_password_here';`

## Best Practices

- Revision control (subversion)
- Automated deployment (capistrano)
- Multiple stages (virtualization)

# Revision control

- Manage changes to application
- Multiple developers
- Ease deployment to multiple servers
- trunk - current version
- tag - released version
- branch - future version

# Automated Deployment

- Capistrano
  - Automates code updates and app restarts via SSH
  - tasks and recipes - not just for Rails!
  - <http://capify.org/>
- Puppet
  - Automates system administration
  - <http://reductivelabs.com/trac/puppet>

## Deployment Staging

- Don't test on production server!
- Test new code, libraries, schema, and packages.
- Staging - newest code (trunk)
- Testing - acceptance
- Virtualization
  - Many virtual server for 1 physical server
  - Throw away servers for testing

Questions?