

**!!! ACHTUNG - evtl. veraltet - ACHTUNG !!!**

Diese Seite wurde zuletzt am 9. Juli 2014 um 08:56 Uhr geändert.

Installation:

```
aptitude install sudo ruby rdoc ruby-dev make libmysqlclient-dev gettext
unzip git-core subversion nginx monit...
cd tmp; wget
http://rubyforge.org/frs/download.php/57643/rubygems-1.3.4.tgz; tar xvfz
rubygems-1.3.4.tgz; cd rubygems-1.3.4; ruby setup.rb
gem install rake rails
```

In dem (ersten) RoR-Projekt sollten in der „environment.rb“ (Rails::Initializer.run) folgende Zeilen enthalten sein:

```
[[...]]
config.gem 'rtex'
config.gem 'sqlite3-ruby', :lib => 'sqlite3'
config.gem 'mysql'
config.gem 'mongrel'
config.gem 'authlogic'
config.gem 'grosser-il8n_data', :lib => 'il8n_data', :source =>
'http://gems.github.com'
config.gem 'grosser-fast_gettext', :lib => 'fast_gettext', :source =>
'http://gems.github.com/'
config.gem 'gettext', :lib => false
config.gem 'haml'
config.gem 'mislav-will_paginate', :lib => 'will_paginate', :source =>
'http://gems.github.com'
[[...]]
```

... welche man dann ganz einfach installieren kann... und wenn man schonmal dabei ist, auch gleich ein paar nützliche Plugins dazu:

```
cd <RoR_PROJEKT_PFAD>
rake gems:install
haml --rails .
ruby script/plugin install 'rspec', :git =>
'git://github.com/dchelimsky/rspec.git'
ruby script/plugin install 'rspec-rails', :git =>
'git://github.com/dchelimsky/rspec-rails.git'
ruby script/plugin install 'exception_notifier', :git =>
'git://github.com/rails/exception_notification.git'
ruby script/plugin install 'asset_packager', :git =>
'git://github.com/sbecker/asset_packager.git'
ruby script/plugin install 'gettext_il8n_rails', :git =>
'git://github.com/grosser/gettext_il8n_rails.git'
ruby script/plugin install 'timer', :svn =>
'http://activewarehouse.rubyforge.org/svn/timer/trunk/'
```

Nun noch nen bissl konfigurieren und es kann mit groooßen Schritten losgehen!

Für eigene Zwecke hatte ich mal ein [RoR-Template](#) erstellt, welches ich ja eigentlich mal zu einem ganz einfachen Template umscripten und hier einfügen könnte!? 😊

User und Gruppe „rails“ anlegen!

config/mongrel\_cluster.yml:

```
---
address: <IP_ADRESSE_DES_(V)SERVERS>
log_file: log/mongrel.log
port: "3000"
environment: production
pid_file: tmp/pids/mongrel.pid
servers: 3
cwd: <RoR_PROJEKT_PFAD>
docroot: public
user: rails
group: rails
```

/etc/nginx/sites-available/rails

```
# Load balance to mongrels
upstream mongrel_cluster {
    server <IP_ADRESSE_DES_(V)SERVERS>>3000;
    server <IP_ADRESSE_DES_(V)SERVERS>>3001;
    server <IP_ADRESSE_DES_(V)SERVERS>>3002;
}
# Virtualhost
server {
    # HTTP Settings
    listen 80;
    server_name www.bluemetaljackets.de;
    root /var/rails/radiant/public;
    access_log /var/log/nginx/bluemetaljackets.access.log;
    error_page 500 502 503 504 /500.html;
    client_max_body_size 50M;
    # First rewrite rule for handling maintenance page
    if ( -f $document_root/system/maintenance.html ) {
        rewrite ^(.*)$ /system/maintenance.html last;
        break;
    }
    location / {
        index index.html index.htm;
        # Forward information about the client an host
        # Otherwise Radiant wouldn't have access to it
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded_For $proxy_add_x_forwarded_for;
        proxy_set_header Host $http_host;
```

```

proxy_max_temp_file_size 0;
# Directly serve static content
location ~ ^/(images|javascript|stylesheets)/ {
    expires 10y;
}
if ( -f $request_filename ) {
    break;
}
# Directly serve cached pages
if ( -f $request_filename.html ) {
    rewrite (.*) $1.html break;
}
# Otherwise let Mongrel handle the request
if ( !-f $request_filename ) {
    proxy_pass http://mongrel_cluster;
    break;
}
}
}

```

/etc/default/monit

```

[[...]]
startup=**1**
[[...]]

```

/etc/monit.d/mongrel.monitrc

```

check process mongrel_rails_3000
    with pidfile <RoR_PROJEKT_PFAD>/tmp/pids/mongrel.3000.pid
    start program = "/usr/bin/mongrel_rails cluster::start -C
<RoR_PROJEKT_PFAD>/config/mongrel_cluster.yml --clean --only 3000"
    stop program = "/usr/bin/mongrel_rails cluster::stop -C
<RoR_PROJEKT_PFAD>/config/mongrel_cluster.yml --only 3000"
    if totalmem is greater than 110.0 MB for 4 cycles then restart
    if cpu is greater than 80% for 4 cycles then restart
    if 20 restarts within 20 cycles then timeout
    group rails
check process mongrel_rails_3001
    with pidfile <RoR_PROJEKT_PFAD>/tmp/pids/mongrel.3001.pid
    start program = "/usr/bin/mongrel_rails cluster::start -C
<RoR_PROJEKT_PFAD>/config/mongrel_cluster.yml --clean --only 3001"
    stop program = "/usr/bin/mongrel_rails cluster::stop -C
<RoR_PROJEKT_PFAD>/config/mongrel_cluster.yml --only 3001"
    if totalmem is greater than 110.0 MB for 4 cycles then restart
    if cpu is greater than 80% for 4 cycles then restart
    if 20 restarts within 20 cycles then timeout
    group rails
check process mongrel_rails_3002
    with pidfile <RoR_PROJEKT_PFAD>/tmp/pids/mongrel.3002.pid
    start program = "/usr/bin/mongrel_rails cluster::start -C

```

```
<RoR_PROJEKT_PFAD>/config/mongrel_cluster.yml --clean --only 3002"
  stop program = "/usr/bin/mongrel_rails cluster::stop -C
<RoR_PROJEKT_PFAD>/config/mongrel_cluster.yml --only 3002"
  if totalmem is greater than 110.0 MB for 4 cycles then restart
  if cpu is greater than 80% for 4 cycles then restart
  if 20 restarts within 20 cycles then timeout
  group rails
```

/etc/monit.d/nginx.monitrc

```
check process nginx with pidfile /var/run/nginx.pid
  start program = "/etc/init.d/nginx start"
  stop program = "/etc/init.d/nginx stop"
  if failed host <IP_ADRESSE_DES_(V)SERVERS> port 80 then restart
  if 5 restarts within 5 cycles then timeout
```

So, das ist erstmal der Anfang = Durcheinander...

sortiert wird später! 😜

Interessante RoR-Projekte:

- <http://radiantcms.org/>
- <http://www.redmine.org/>
- <http://github.com/rubyist/aasm/tree/master>

danach conf/environment.rb folgendes hinzufuegen:

```
config.gem "rubyist-aasm", :lib => "aasm", :source =>
"http://gems.github.com"
```

- <http://github.com/technoweenie/restful-authentication/wikis/home>

danach ein einfaches

```
script/generate authenticated user sessions --aasm
```

- [http://github.com/timcharper/role\\_requirement/wikis/usage](http://github.com/timcharper/role_requirement/wikis/usage)

danach ein einfaches

```
script/generate roles Role User
```

- <http://agilewebdevelopment.com/plugins/timer>
- <http://rubyforge.org/projects/fckeditorp/>

- <http://code.google.com/p/rails-unobtrusive-date-picker/>

From:

<http://wiki.neumannsland.de/> - **Patricks DokuWiki**

Permanent link:

<http://wiki.neumannsland.de/mw2dw:ds3000-ror>

Last update: **2019/09/23 14:32**

