

About



关于我

- (1), 80后
- (2), !女
- (3), 毕业于 [cs1g](#)
- (4), 乡下人进城
- (5), 目前依靠 [ruby on rails](#) 在北京混口饭
- (6) , ,

关于本站

本站始建于2010年儿童节,修改开源[typo5.4.4](#)(rails)搭建,架在[Linode](#) VPS上,主要记录在工作和生活中一些片段,个人认为 , 好记性不如烂笔记,留若干年后回来看看,嘿嘿。。域名 来源于小生名字的简写

希望和志同者 交换连接~

capistrano + nginx + thin + subversion 自动部署



以前部署app还是用很老套的ssh客户端, windows我使用 SSH Secure Shell Client, linux 上使用 Filezilla, 传代码非常之慢.今天体验下rails capistrano 自动部署, 确实挺方便的, 实现了ssh远程登录操作的一系列操作, capistrano 的实现原理是, 本地客户端 执行命令, 实质进行了一系列远程操作. 部署同步的根本是check 版本控制系统的最新版, 版本控制系统可以是subversion, git等, 我用的是subversion, 步骤:

我的环境:nginx + thin + capistrano 2.5.19 (客户端) + centos 5.5 + subversion

1, 安装 capistrano

```
gem install capistrano
```

2,rails app加入 capistrano 配置文件

```
capify .
```

3,配置 config 下的deploy.rb ,最核心的东西都在这个文件里

```
set :application, "wxianfeng_com" # 工程名, 随便写了
set :repository, "svn://173.230.155.150/wxianfeng_com" # svn 地址
set :scm, :subversion # 使用 suversion版本控制
# Or: `accurev`, `bzzr`, `cvs`, `darcs`, `git`, `mercurial`, `perforce`, `subversion` or `none`
set :scm_username, 'wxianfeng' # svn 用户名
set :scm_password, '123456' #svn 密码
set :checkout, 'export' # 使用export方式, svn版本信息不需要 track下来
```

```

role :web, "173.230.155.150" # 这三个填你服务器的ip地址
role :app, "173.230.155.150" #
role :db, "173.230.155.150", :primary => true #
# role :db, "your slave db-server here"

set :deploy_to, "/usr/local/system/www/wxianfeng_com" # 部署到remote 服务器路径
set :deploy_via, :remote_cache # 通过 remote_cache 方式部署 ,还有一种本地copy的方式
set :user, "root" # remote 服务器的用户名
set :runner, "root" # 同上
set :password, "123456" # 服务器的 root 密码

set :rake, "/usr/bin/rake" # 服务器上rake 命令path
default_run_options[:shell] = false
default_run_options[:pty] = true
set :use_sudo, true # 允许使用 sudo

namespace :deploy do

  task :start, :roles => :app do
    invoke_command "cd .." # invoke_command 是连到 服务器上执行的命令
    invoke_command "thin start -C /etc/thin/thin.yml"
  end

  task :stop, :roles => :app do
    invoke_command "killall thin" # or "thin stop -C /etc/thin/thin.yml"
  end

  task :restart, :roles => :app, :except => { :no_release => true } do
    #run "#{try_sudo} touch #{File.join(current_path,'tmp','restart.txt')}"
    invoke_command "cd .."
    invoke_command "thin restart -C /etc/thin/thin.yml -O" # -O 是onebyone(关一个, 启一个) 启动thin ,注意是 大写
  end

  desc "Symlink shared configs and folders on each release."
  task :symlink_shared do
    run "rm -rf #{release_path}/public/files" # 创建 软连接, 保证服务器上 图片的 同步
    run "ln -fs #{shared_path}/public/files #{release_path}/public/files" # or current_path run 等同于 invoke_command
  end

end

after 'deploy:update_code', 'deploy:symlink_shared' # 在 deploy:update_code 执行后 callback deploy:symlink_shared

```

以上当你执行 `cap deploy` 后 大体执行过程是 先 `update_code` 然后 `deploy:symlink_shared` 最后 `deploy:restart` 服务器 , 上面有一个thin的配置 `/etc/thin/thin.yml`,如果没有的话, 可以到服务器上生成:

```
thin config -C /etc/thin/thin.yml -c /usr/local/system/www/wxianfeng_com/current -s 3 -e production -p 3000
```

4.部署

第一次部署需要运行下面的命令:

```
cap deploy:setup
```

报错:

```
sh: sudo: command not found
```

解决:

```

>vi /etc/sudoers
# Disable "ssh hostname sudo <cmd>", because it will show the password in clear.
# You have to run "ssh -t hostname sudo <cmd>".
#
#Defaults    requiretty #将这行注释掉

```

这个命令会在服务器上配置好相关的目录结构.然后再运行:

```
cap deploycheck
```

检查通过

第一次部署用的命令是:

```
cap deploycold
```

以后部署都可以直接用:

```
cap deploy
```

修改 nginx 配置:

```
root    /usr/local/system/www/wxianfeng_com/current/public/;
```

修改 root 根目录为 capistrano 的 current_path 的 public 目录

ok, 你以后更新代码 只需要 cap deploy 就ok了, 前提是你先 commit 代码到 svn 服务器上, 用capistrano好处还有一个可以 备份代码, 因为每一次部署 其实就是相当于 track了svn版本里的最新代码

See:

<http://gautamrege.wordpress.com/2009/11/10/capistrano-nginx-thin-deployment-on-linode/>

<http://weekface.info/2010/03/15/rails-git-capistrano>

<http://rustammamedov.wordpress.com/2009/12/27/capistrano-nginx-mongrel-deployment/>

mysql 导入大量数据



想让你的本地mysql数据库某个表迅速产生大量数据吗, 例如 500 万条, 来进行性能测试, 我就遇到了这个情况, 导入过程如下:

测试环境: windows xp + mysql 5.1

myisam 引擎的表导入:

表结构如下:

```
members(id,name,password,city,created_at,updated_at)
```

id是int型,name,password,city 都是varchar型,created_at,updated_at 是 datetime类型,其中 city字段有索引

1,关闭表的索引

```
ALTER TABLE members DISABLE KEYS;
```

关闭keys, 会在导入过程中对新的record不建索引, 加快导入速度

2,导入数据

```
load data infile 'E:/members.txt' into table members fields terminated by "," optionally enclosed by "'" lines terminated by "\r\n" (name,password,city,created_at,updated_at);
```

fields terminated by "," 字段分隔符, lines terminated by "\r\n" 行分隔符, 注意windows下的默认行换行符是 \r\n ,UNIX 下的换行符是 \n , optionally enclosed by "'" 对于 字符型字段 加双引号, 不是指字符型的, 例如int 则不加双引号

members.txt 文件内容形式:

```
"wxianfeng","wxianfeng","beijing","2010-06-30 00:07:45","2010-06-30 00:07:48"
"kaili","kaili","suzhou","2010-06-30 00:07:45","2010-06-30 00:07:48"
```

最后 (name,password,city,created_at,updated_at) 指定依次导入的字段

那么 members.txt 又是怎么来的呢, 你可以导出一个表瞧瞧就知道了, 导出方法如下:

```
select * from members into outfile 'E:/members.txt' fields terminated by "," optionally enclosed by '';
```

那么下面就是你有什么办法快速生成上面的 members.txt 了, 我用ruby 写了个脚本生成了 500 万条数据:

```
# members(name,password,city,created_at,updated_at)

citys = %w(beijing shanghai guangzhou nanjing wuhan xihan hangzhou dalian jinan hefei)

arr = ('0'..'9').to_a + ('a'..'z').to_a

File.open("E:/members_1.txt","a+") do |file|
  2_000_000.times do |ele|
    p "generate --- #{ele}"
    str = ''
    5.times { |t| str << arr[rand(arr.size)] }
    time = Time.now.strftime("%Y-%m-%d %H:%M:%S")
    file.write("#{str}\",\"#{str}\",\"#{citys[rand(citys.size)]}\\\", \"#{time}\\\", \"#{time}\\\"\\n")
  end
end
```

最后生成 500 万条数据后, 导入的速度非常之快, 据官方介绍 load data infile 的方式是sql导入速度的20倍, 快的根本原因我想应该还是生成的文件小, 纯粹数据的形式, 容易快速加载到内存里

3, 再开启 KEYS

```
ALTER TABLE members ENABLE KEYS;
```

innodb 引擎表的导入:

1, 导入前关闭表的唯一性校验

```
SET UNIQUE_CHECKS=0;
```

2, 关闭自动提交

innodb型的表是事务安全的, 每一条sql执行后都会自动提交, 关闭后将会提交导入速度

```
SET AUTOCOMMIT=0;
```

3, 导入数据

和之前 myisam 表的导入方式相同, 但是有一点不同的是, 生成members.txt 数据的时候, 把id也生成, 因为innodb类型的表是按照主键顺序保存的, 对于给定有序的文本, 比没有顺序的要快的多

4, 开启之前的 唯一性和事务自动提交功能

```
SET UNIQUE_CHECKS=1;
SET AUTOCOMMIT=1;
```

最主要的还是要使用 load data infile 的导入方式, 无论怎样, 你不要使用 单条 insert, 一条一条的插入, 这样网络io开销会巨大, 速度很慢, 实在不行, 你应该构造 一个insert into 多个value的方式

来导入，也会快很多，这个就是默认的mysqldump导出表数据所用的方法，还可以发现 在导入之前，它进行了锁表：

```
LOCK TABLES `members` WRITE;
insert into members values (.....),(.....),(....)
UNLOCK TABLES;
```

ref: 参考了 网易出的那本mysql书籍

centos 5.5 安装 php + phpmyadmin



这几天真是和phpmyadmin有缘啊，老是和他接触，今天又在我的linode VPS上搭建了个phpmyadmin，用来管理mysql数据库，装好后才猛然发现我的数据库中中文全部是乱码，但是rails读出来是中文，也就是大家看到我现在的这个blog是中文，但是mysql里是乱码，看了下所有的编码都是utf-8形式，诡异的问题，这个问题比较棘手，必须解决。但是今天写的是 centos 5.5 上如何搭建phpmyadmin，和其他系统的安装的大同小异

1, 安装php

```
yum install php --enablerepo=centosplus
```

注意 - 和 enablerepo 之间有个空格，加上 --enablerepo=centosplus 为了安装到5.*的php，如果直接 yum install php 的话，则安装的是php 4.* 系列

以后更新php 可以执行下面的命令：

```
yum update php --enablerepo=centosplus
```

结束后 可以用 php -v 查看安装的版本信息

2,安装phpmyadmin

到phpmyadmin官网下载 phpmyadmin，注意下载的版本 必须注意支持的mysql 和 php 版本

我用的是 2.1 版,安装和配置过程：

```
>cd /usr/local/system # 进入到该目录下，这个一般是我的工作目录，你可以任意指定
>wget http://downloads.sourceforge.net/project/phpmyadmin/phpMyAdmin/2.11.10/phpMyAdmin-2.11.10-all-languages.tar.gz?use_mirror=cdnetworks-kr-1&27689465 # 下载 phpmyadmin
>tar -zxvf phpMyAdmin-2.11.9.5-all-languages.tar.gz # 解压
>mv phpMyAdmin-2.11.9.5-all-languages phpmyadmin # rename
>cp config.sample.inc.php config.inc.php
>vi config.inc.php
Scfg['blowfish_secret'] = 'TypeAnything_for_Secure'; # 找到 Scfg['blowfish_secret'] 值为后面的值
>vi /etc/httpd/conf.d/phpmyadmin.conf
Alias /phpmyadmin /usr/local/system/phpmyadmin # 添加该代码
```

3,配置apache的端口

```
>vi /etc/httpd/conf/httpd.conf
Listen:8080 # 修改Listen 端口为 8080
```

4,重启 apache

```
service httpd restart
```

>netstat -antup # 可以查看端口启动情况

5,访问phpmyadmin

<http://localhost:8080/phpmyadmin>

报错, php 不能加载 mysql 模块 , 因为 php 没有安装 mysql 扩展,
安装之:

>yum install php-mysql - enablerepo=centosplus

再次访问 phpmyadmin .OK 成功了

我的phpmyadmin地址:

<http://wxianfeng.com:8080/phpmyadmin>

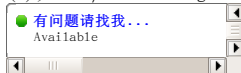
See: <https://www.seven1.net/blog/how-to-install-lamp-phpmyadmin-and-vsftp-on-centos-5-3-using-yum/>

<http://imthi.com/blog/linux/install-or-update-php-5-on-centos.php>

Contact



(1),Gmail/Gtalk: wang.fl1429@gmail.com



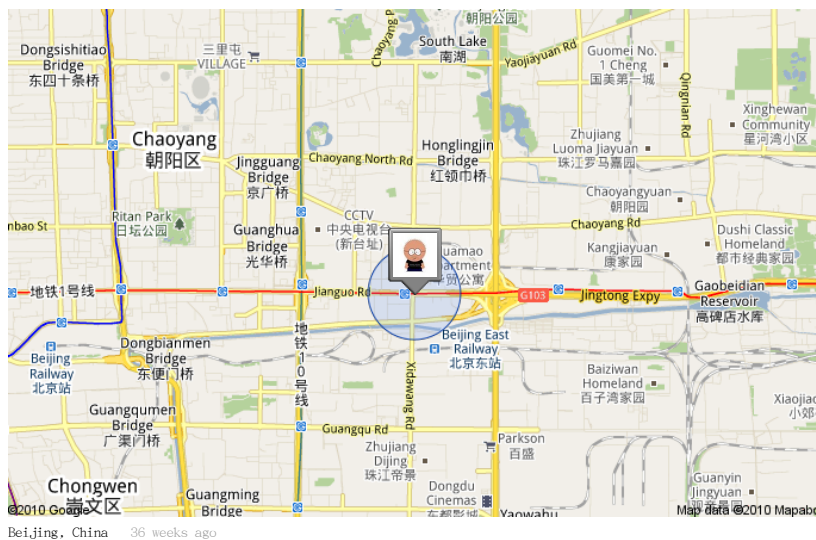
(2),QQ : 522096432



(3),Twitter : <http://twitter.com/fl1429>

(4),Google Profile : <http://www.google.com/profiles/Wang.FL1429>

(5),Google Latitude



google.com/latitude

centos 5.5 install svn server



为了个人的项目代码的同步，例如我在公司写的代码，回家后还能继续同步写下去，在我的VPS上搭了个svn server，搭建步骤：

1, 安装subversion

```
yum update
yum install subversion
```

3, 创建和配置代码仓库

```
cd /usr/local/system
svnadmin create vcs

# 配置工程
vim vcs/conf/svnserve.conf
# 取消掉如下两行的注释
# auth-access = write
# password-db = passwd

# 配置该工作的用户与密码
vim vcs/conf/passwd
[users]
wxianfeng = 12345
```

4, 启动svn server

```
cd /usr/local/system
svnserve -d -r vcs
```

5, 导入工程

```
svn import wxianfeng_com svn://173.230.155.150/wxianfeng_com -m "first import" --username wxianfeng --password 12345 # 远程 或本机导入
svn import wxianfeng_com file:///localhost/wxianfeng_com # 本机导入
```

前面的wxianfeng_com 是本地的文件夹，后面的wxianfeng_com 是svn仓库的project name，或者你可以借助netbeans, eclipse带的svn客户端直接导入netbeans中或eclipse中的project到svn仓库中

6, checkout项目

```
svn co svn://173.230.155.150/wxianfeng_com
```

以上就可以做到一个仓库，多个项目的目的了，当有另外一个project时，直接import即可，如果想删除仓库中的某个工程，可以执行delete命令

```
svn delete svn://173.230.155.150/wxianfeng_com -m "delete project"
```

See:

<http://www.tonyspencer.com/2007/03/02/setup-a-subversion-server-in-4-minutes/>
<http://blog.itmem.com/archives/1207>

Projects



2010-05月 管理mysql数据库 <http://wxianfeng.com:8080/phpmyadmin/>
php 5 + phpmyadmin 3.3 + apache 2.2 配置

2010-06月 管理邮件收发(webmail)
<http://wxianfeng.com:8080/webmail> SquirrelMail 1.4.8 + php 5 +
apache 2.2 配置

2010-06月 flickr mashup <http://wxianfeng.com/mashups/flickr>
rails 2.3.5 + galleria js + flickr API 开发

2010-07月 短地址服务 <http://url.wxianfeng.com> mongodb 1.4.4 +
ruby 1.8.7 + haml 3.0.13 + sinatra 1.0 开发

go on ...

rails + mongodb 入门demo



NoSQL 渐渐火了起来，最近同事也在讨论这个问题，一不做，二不休，就学习了下，mongodb是采用C++写的，开源，免费，目前已经有公司(10gen)在背后提供商业支持，文件之小，windows下载二进

制包解压后是45M左右，linux下的就更小了，读写操作性能之高，即使是在大数据量，大开发的情况下，也能保持良好的性能，可以索性，例如可以结合memcached, sphinx使用，哇。。。cool。。。这两天利用mongodb + sinatra 做了个短地址服务网址，当然只是自娱自乐一把而已，domain太ugly了，地址是 <http://url.wxianfeng.com>，这编来介绍

rails中使用mongodb的入门教程，是railscasts上的那篇教程，但是有许多注意点，在此重点指出。

在开始这个demo前，如果你对mongodb还不熟的话，建议先学习下mongodb，就想先学习下mysql 一样，掌握一些基本的操作过程
我的环境:windows xp + rails 2.3.5 + mongodb 1.4.4 +
mongo_mapper 0.8.2

1, 安装需要的gem包

```
gem install mongo_mapper
```

如果顺利的话，可以安装成功，并且包依赖的其他gem也都同时安装上了，例如我在美国的linode服务器上，直接执行上面命令，就安装成功了，但是在大陆无法安装成功，至少我是经过下面的安装步

骤成功的:

```
gem install jnunemaker-validatable --source http://gemcutter.org
gem install bson --source http://rubygems.org
gem install mongo --source http://gemcutter.org
gem install plucky --source http://gemcutter.org
gem install mongo_mapper --source http://gemcutter.org
```

2, 新建rails工程,修改你的enviroment.rb


```
config.gem "mongo_mapper",:source=>"http://gemcutter.org"
config.frameworks -= [:active_record] #- 代表不加载 active_record
```

在这里需要注意, 当你需要使用scaffold生成代码时, 必须把 `—` 给去掉, 来加载active_record, 因为scaffold默认需要使用activerecord, 当你开启server时, 再把 `—` 号给加上

4, 修改 database.yml

```
development:
  adapter: mysql
  encoding: utf8
  reconnect: false
  database: trade_supplier_development
# pool: 5
  username: root
  password: root
# socket: /var/run/mysql/mysql.sock
  host: localhost
```

需要保证mysql的username, 和 password正确, 还是为了能使用 scaffold

5, 添加 配置, 在 config/initializers 下新建 mongo_config.rb

```
MongoMapper.database = "todo-#{Rails.env}"
```

配置连接时的db name, start server时不会建立 db, 当插入数据时会同时建立 db 和 collection

6, 修改Model

```
class Project
  include MongoMapper::Document
  key :name, String, :required => true
  key :priority, Integer
  many :tasks
end

class Task
  include MongoMapper::Document
  key :project_id, ObjectId
  key :name, String
  key :completed, Boolean
  belongs_to :project
end
```

核心部分就是Model部分, db中的字段是根据 key来动态改变的, 没有shama, 没有migrate, 当你添加新的key时, 原有的数据不会改变, 上面的两个model是1对多的关系, 例如mongo——mapper 可以来

实现 1对1, 1对多, 多对多的关系, 并且和 activerecord操作非常之相像

7, 启动 mongod 进入mongod bin 目录下

```
mongod.exe --dbpath E:/mongod/data
```

8, 启动server

```
thin start
```

这一步会提示安装 bson_ext gem包, 我在linux中安装成功了, 在windows无法顺利安装, 貌似只能在UNIX中安装, 需要make编译, 但是只要你安装bson包, 程序都可以顺利跑起来, BSON是json的binary

形式, 这样会节省存储空间, 最重要的是减少网络消耗

9, ok

<http://localhost:3000/projects> 瞧瞧

DEMO源码下载:

<http://www.uushare.com/user/f11429/file/3274129>

See:

<http://asciicasts.com/episodes/194-mongodb-and-mongomapper>

Ruby 两数组分组求和



环境：ruby 1.8.7

今天我有一个这样的需求，例如，a=[1,2,1]，b=[10,20,30] 数组b中的项是a项对应的值，现在要求出a数组中相同项的和（值对应b的数组），例如数组a，b的最后结果是 {1=>40,2=>20}

在ruby的Array的方法中 没有找到对应的方法，就自己写了个，用ruby已有的方法堆砌起来的，自认为比较繁琐，望高人指点

```
# DEMO : [1,2,3].sum => 6
#       [1,2,3].sum { |s| s*2 } => 12
class Array
  # def sum
  #   inject { |sum,n| sum + n } # self.reduce(&:+)
  # end

  # Rails 里Array的sum方法源码
  def sum(identity = 0, &block)
    if block_given?
      map(&block).sum(identity)
    else
      inject { |sum, element| sum + element } || identity
    end
  end
end
```

```
class Hash
  # demo :
  # a = [1,2,1,3]
  # b = [10,20,30,50]
  # Hash.group_sum(a,b) => {1=>40,2=>20,3=>50}
  def self.group_sum(arr1,arr2)
    hsh = {}
    hsh2 = {}
    arr1.uniq.each do |ele|
      arr = []
      arr1.each_with_index do |val,index|
        arr << index if val == ele
      end
      hsh[ele] = arr
    end
    values = []
    hsh.each do |k,v|
      sum = v.inject(0) { |s,x| s+arr2[x] }
      values << sum
    end
    hsh2 = Hash[arr1.uniq.zip(values)]
    hsh2
  end
end
```

windows ssh client



如果你是命令行爱好者,我想ssh应该是你用的最多的工具了,那么在windows有好的ssh client吗,个人用过下面三个client,总结下来,力荐 secureCRT ,

1, putty

download:

<http://www.newhua.com/soft/2186.htm>

比较简单, 小巧, 免费, 支持语法高亮, 单tab, 值得推荐, 适合服务器 机器不多的情况下使用, 如果你需要同时连接 多个 电脑的话, 那么用起来就麻烦了

2, ssh secure shell client

download:

<http://www.onlinedown.net/soft/20089.htm>

免费, 也是单标签, 默认没有语法高亮, 同时 带了 ssh 文件上传功能, 如果你需要上传文件, 又需要 命令行管理服务器, 那么 ssh secure shell client 还是不错的

3,SecureCRT

download:

http://www.onlinedown.net/softdown/46646_2.htm

最后推荐secureCRT, 默认不支持 语法高亮, 杀手锏的是支持多标签, 可以保存会话状态(密码), 还有就是不免费, 但是晚上破解版的到处都是, 下载后 更改 一些配置, 例如, 语法高亮等, 配色, 用起来还是很爽的。。。

centos 中的支持VIM高亮配置 和 中文乱码解决:

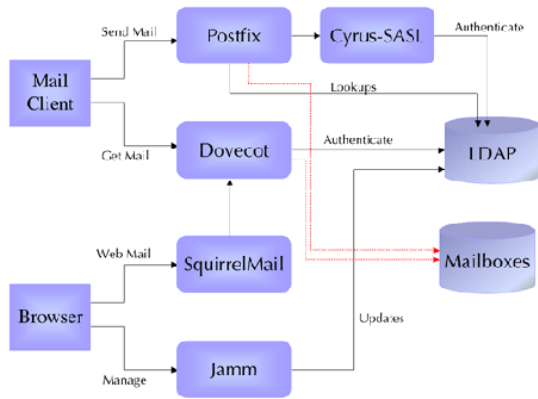
(1),在SecureCRT中设置 选项→会话选项→终端→仿真→终端: Linux, 选中ANSI Color,全局设置的话, 选择 选项→ 全局选项 设置即可

(2), 选项 → 会话选项 → 外观 修改字符编码为 UTF-8

(3),

```
>vim /etc/vimrc
syntax on # 添加语法高亮 开启
set encoding=utf-8 # 修改编码为 utf-8
```

centos install mail server



环境:centos 5.5 + postfix + dovecot +SASL + Squirrelmail

postfix 提供smtp服务,dovecot 提供pop3,imap服务, sasl提供smtp认证服务, Squirrelmail 是webmail 用来收发邮件之用, 最近需要发送email的服务. 本想使用gmail的smtp的, 后来索性就自己搭了个, 以便熟悉mail服务器的搭建过程, 一路下来, 发现还真不是一件容易的事, 这个还是比较简单的, 还有加密传输, 反垃圾邮件处理等等东西没弄, 搭建过程中遇到的问题比较多, 问题最大的地方是 例如我用 gmail 发送邮件给 wxianfeng@wxianfeng.com, 邮件收不到, gmail中也没有显示退信信息, 说明邮件还是发出去了, 最后发现是 postfix 的配置文件 mynetworks 配置错误

上面的是我mail服务器搭建的选择, 当然你也可以选择别的mail服务器, 例如 smtp你可以选择sendmail, 或者你可以看看别人的mail服务器是什么, eg:

```

[root@l1165-150 ~]# telnet vic360.com 25      # 或者 telnet vic360.com smtp
Trying 221.6.104.142...
Connected to vic360.com.
Escape character is '^['.
220 mail.vic360.com Kerio MailServer 6.3.1 ESMTP ready
quit
221 2.0.0 SMTP closing connection
Connection closed by foreign host.
[root@l1165-150 ~]# telnet vic360.com pop3 # 或者 telnet vic360.com 110
Trying 221.6.104.142...
Connected to vic360.com.
Escape character is '^['.
+OK Kerio MailServer 6.3.1 POP3 server ready <12927.1280636072@mail.vic360.com>

```

可以看出 vic360 的mail服务器选择的是 Kerio MailServer, 25 一般是 smtp的port, 110 是 pop3 的port

```

[root@l1165-150 ~]# telnet mail.beebuyer.com smtp
Trying 218.202.225.66...
Connected to mail.beebuyer.com.
Escape character is '^['.
220 mobase.cn ESMTP MDaemon 9.5.1; Sun, 01 Aug 2010 12:21:07 +0800
quit
221 See ya in cyberspace
Connection closed by foreign host.
[root@l1165-150 ~]# telnet mail.beebuyer.com pop3
Trying 218.202.225.66...
Connected to mail.beebuyer.com.
Escape character is '^['.
+OK mobase.cn POP MDaemon 9.5.1 ready <MDAEMON-F201008011221.AA2142193MD8611@mobase.cn>

```

可以看出 beebuyer.com 的 mail server 是 MDaemon 9.5.1

okay, 下面介绍 我的 mail server的搭建过程:

1, 安装 postfix, 卸载sendmail

```

yum update
yum remove sendmail
yum install postfix

```

2,配置 postfix

```

>vim /etc/postfix/main.cf
myhostname = mail.wxianfeng.com
mydomain = wxianfeng.com
myorigin = Smydomain
inet_interfaces = Smyhostname, localhost, 173.230.155.150 # 173.230.155.150 是我服务器IP
mydestination = Smyhostname,localhost.Smydomain, localhost, Smydomain,mail.Smydomain,www.Smydomain
mynetworks = 0.0.0.0/0
home_mailbox = Maildir/ # email的存放地址, 默认是存放在file中, 你还可以配置mysql, 存到mysql 中

```

3,安装配置SASL + TLS

```

>yum install cyrus-sasl
>vim /etc/postfix/main.cf
smtpd_sasl_auth_enable = yes
smtpd_sasl_local_domain = wxianfeng.com
smtpd_recipient_restrictions = permit_mynetworks,permit_sasl_authenticated,reject_unauth_destination
smtpd_sasl_security_options = noanonymous

```

```
smtpd_sasl_type = dovecot
smtpd_sasl_path = private/auth
broken_sasl_auth_clients = yes
```

4, 安装配置 Dovecot

```
>yum install dovecot
>vim /etc/dovecot.conf
protocols = pop3 pop3s imap imaps
mail_location = maildir:~/Maildir/ # email的存放地址
pop3_uidl_format = %08Xu%08Xv
auth default {
mechanisms = plain login
passdb pam {
}
userdb passwd {
}
socket listen {
client {
path = /var/spool/postfix/private/auth
mode = 0660
user = postfix
group = postfix
}
}
}
```

5, 安装配置 Squirrelmail

```
>yum install squirrelmail
>vim /etc/httpd/conf.d/squirrelmail.conf
Alias /webmail /usr/share/squirrelmail # 配置apache
>/usr/share/squirrelmail/config/conf.pl
2 -> 1 Domain : wxianfeng.com # 配置域名
10 -> 2 Default Charset : utf-8 # 配置编码
```

6, 重启所有服务

```
/etc/init.d/postfix restart
/etc/init.d/dovecot restart
/etc/init.d/saslauthd restart
service httpd restart
```

7, 测试安装是否成功

<http://wxianfeng.com:8080/webmail/src/configtest.php>

<http://wxianfeng.com:8080/webmail> 访问, 可以 收发email

8, 创建mail用户

```
>adduser wxianfeng
>passwd 12345
```

9, 本地测试收发email

smtp发:

```
>telnet localhost smtp
Trying 127.0.0.1...
Connected to localhost.localdomain (127.0.0.1).
Escape character is '^]'.
220 mail.wxianfeng.com ESMTP Postfix
>ehlo localhost
250-mail.wxianfeng.com
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-ETRN
250-ENHANCEDSTATUSCODES
250-8BITMIME
250 DSN
>mail from:wxianfeng
250 2.1.0 Ok
>rct to:wxianfeng
250 2.1.5 Ok
>data
354 End data with <CR><LF>.<CR><LF>
test
. # 以 . 结束输入
250 2.0.0 Ok: queued as 9729067C17
quit
221 2.0.0 Bye
Connection closed by foreign host.
```

查看 发送队列

```
[root@mail ~]# cd /home/wxianfeng/Maildir/new
[root@mail new]# ls
1185669817.Vfd00118012M795756.mail.wxianfeng.com
[root@mail new]# cat 1185669817.Vfd00118012M795756.mail.wxianfeng.com
```

pop3 收 测试:

```
[root@1165-150 ~]# telnet localhost pop3
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
+OK Dovecot ready.
user wxianfeng
+OK
pass 12345
+OK Logged in.
list
```

```
+OK 11 messages:
1 483
2 453
3 809
4 853
5 2029
6 2671
7 2721
8 2870
9 3178
10 2749
11 2702
.
retr 1
+OK 483 octets
Return-Path: <wxianfeng@wxianfeng.com>
X-Original-To: wxianfeng@wxianfeng.com
Delivered-To: wxianfeng@wxianfeng.com
Received: from localhost (localhost [127.0.0.1])
    by 11165-150.wxianfeng.com (Postfix) with ESMTP id D31F91331F
    for <wxianfeng@wxianfeng.com>; Thu, 29 Jul 2010 03:35:50 -0400 (EDT)
Message-Id: <20100729073619.D31F91331F@11165-150.wxianfeng.com>
Date: Thu, 29 Jul 2010 03:35:50 -0400 (EDT)
From: wxianfeng@wxianfeng.com
To: undisclosed-recipients::;
```

```
test
.
```

10 , 域名 设置

必须保证你的 域名设置那有 mail.wxianfeng.com 指向你的 ip , 或者你有 *.wxianfeng.com 的 泛域名 解析

11, 添加 MX record

在你的域名商那添加 MX record ,例如我的在godaddy, 什么是mx record: 邮件交换记录(MX record)是一个DNS资源记录类型, 它指出哪个主机能够处理一个特定域的e-mail。把godaddy默认的mx record删除掉, 我的默认是下面的这两个,删除之:

Priority	Host	Goes To	TTL
10	●	mailstore1.secureserver.net	1 Hour
0	●	smtp.secureserver.net	1 Hour

添加自己的:

Priority	Host	Goes To	TTL
10	●	mail.wxianfeng.com	1 Hour

设置好后 可能过段时间才生效,测试生效方法,..下面结果说明已经生效:

```
[root@11165-150 ~]# host -t mx wxianfeng.com
wxianfeng.com mail is handled by 10 mail.wxianfeng.com.
```

12.解决Squirrelmail 内中文email的乱码

在option (选项) 那设置显示语言为 中文简体即可

13.如果你的Squirrelmail 不可以 收发邮件,可以直接远程telnet 测试

```
>telnet mail.wxianfeng.com 25
mail from:wxianfeng
rcpt to:wang.fl1429@gmail.com
.....
```

```
>telnet mail.wxianfeng.com 110
user wxianfeng
pass 12345
list
retr 1
.....
```

如果telnet 可以发送成功 , 就没 问题….

14.查看错误

如果过程中出现错误, 可以随时查看log

```
>tail -f /var/log/maillog
```

that' s all , just do it

See :

<http://www.mysql-apache-php.com/mailserver.htm>
<http://www.linuxmail.info/postfix-smtp-server-howto-centos-5/>

douban blog 认领标识:

doubanclaim5c7c4eaf84713af2

shell 监控网站



写了个shell脚本 监控网站 是否能ping通 (前提是你iptables设置允许ping)，还有测试 访问网站首页 返回的状态 是否是 200，如果出现异常会发邮件 和 短信，ping域名不通了，说明你的网关到ISP的网络不通，ping IP 地址不通，说明你的网关到你的服务器的网络有问题了，ping 域名不通，ping IP地址通，那就应该是DNS解析的问题了。

```
#!/bin/sh

if [ $# -ne 1 ] # -ne 不等于
then
    echo "Usage: $0 <URL>"; # $0 得到当前文件名
    exit;
fi

if ping -c 2 $1 # $1 得到第一个参数
then
    echo "server is running"
else
    echo "$1 is down" | mail -s "$1 is down" wang.fl1429@gmail.com
    curl "http://sms.api.bz/fetion.php?username=15001100000&password=12345&sendto=15001100000&message=$1%20is%20down"
fi

response=$(curl -s -I -L $1 | grep HTTP);

status=${response##* }; # Strip off characters up to the first space
status=${status:0:3}; # Just use the 3 digit status code

if [ "$status" != "200" ]
then
    echo "Error fetching $1. Status code $status" | mail -s "$1 response code is not 200" wang.fl1429@gmail.com;
    curl "http://sms.api.bz/fetion.php?username=15001100000&password=12345&sendto=15001100000&message=$1%20response%20status%20code%20is%20$status"
fi
```

```
#!/bin/sh

if [ $# -ne 1 ] # -ne 不等于
then
    echo "Usage: $0 <URL>"; # $0 得到当前文件名
    exit;
fi

ping -c 2 $1 &&/dev/null # $1 得到第一个参数, -c 2 是发送两次数据包, &&/dev/null 输出内容null

if [ $? = 0 ] # $? 输出命令退出代码: 0为命令正常执行, 1-255为有出错
then
    echo "server is running"
else
    echo "$1 is down" | mail -s "$1 is down" wang.fl1429@gmail.com
    curl "http://sms.api.bz/fetion.php?username=15001100000&password=12345&sendto=15001100000&message=$1%20is%20down"
fi

response=$(curl -s -I -L $1 | grep HTTP);

status=${response#* }; # Strip off characters up to the first space
status=${status:0:3}; # Just use the 3 digit status code

if [ "$status" != "200" ]
then
    echo "Error fetching $1. Status code $status" | mail -s "$1 response code is not 200" wang.fl1429@gmail.com;
    curl "http://sms.api.bz/fetion.php?username=15001100000&password=12345&sendto=15001100000&message=$1%20response%20status%20code%20$status"
fi
```

```
chmod +x monitor domain.sh
```

```
>crontab -e
# m h d mon dow command
*/10 * * * * /usr/local/system/shell/monitor domain.sh "wxianfeng.com" # 每10分钟运行一次
```

可以设置邮件 短信提醒，同时收到邮件 和 短信，不错，
我的centos 5.5 在执行 mail 命令时提示找不到，需要安装mailx 库
解决：

```
yum install mailx
```

see:

<http://www.coderholic.com/server-status-checker-shell-script-2/>
http://blog.s135.com/fetion_api/

Ebooks



Ruby & Rails

1,Refactoring Ruby Edition (Ruby 重构)

英文版: <http://www.uushare.com/user/f11429/file/3378004>

中文版(前三章): <http://www.uushare.com/user/f11429/file/3377937>

2,Agile.Web.Development.With.Rails.rar (rails敏捷开发)

第二版(en版): <http://www.uushare.com/user/f11429/file/1914412>

第二版(zh版): <http://www.verycd.com/topics/202070/>

第三版(en版): <http://www.uushare.com/user/f11429/file/1914425>

3,Refactoring.In.Ruby

en version : <http://www.uushare.com/user/f11429/file/3377947>

4,《Ruby on Rails Web 开发之旅 (高清中文PDF版)》

<http://www.verycd.com/topics/2764557/>

5,《Web开发大全: Ruby on Rails版》

<http://www.verycd.com/topics/2755648/>

6,《Ruby on Rails辅助开发教程》

<http://www.verycd.com/topics/156833/>

7,Head.First.Rails

<http://www.verycd.com/topics/2744888/>

8,《Ruby编程语言(涵盖Ruby 1.8和1.9)中文高清PDF版》

<http://www.verycd.com/topics/2798470/>

9,《面向对象的脚本语言Ruby+中英文教程》

<http://www.verycd.com/topics/113543/>

10,《Ruby不完全读书指南》

<http://www.verycd.com/topics/141282/>

11,《Ruby on Rails 开发工具及教程合集》

<http://www.verycd.com/topics/136859/>

12,《vtr ruby on rails 教程》
<http://www.verycd.com/topics/91982/>

13,Wrox.Professional.Ruby.on.Rails.Feb.2008.pdf
<http://www.uushare.com/user/f11429/file/3378095>

14,Rails4days
<http://www.uushare.com/user/f11429/file/3378094>

15,The Rails Way
<http://www.uushare.com/user/f11429/file/3378157>

16,OReilly Advanced Rails
<http://www.uushare.com/user/f11429/file/3378155>

17,Agile Web Development With Rails Fourth Edition
<http://www.uudisc.com/user/f11429/file/3613566>

go on

G1 安装apk程序



环境: XP + android 1.6 版本 + G 1手机

在中关村入手G1 (1500元)已经一个星期了, 另外在taobao购买了34元购得一个2g TF卡, 今天测试朋友做的android应用程序, g1 通过usb连到电脑上死活没反应, 折腾了会, 找到了原因。。。需要重启手机就ok了。。。安装apk程序方法有很多种, 依自己选择~!

1,查看 android 版本

设置 → 固件版本

2, 下载g1 XP驱动程序

<http://www.uushare.com/user/f11429/file/3408808>

3, 插上g1 usb连接到 电脑

4, 安装驱动, 不行的话, 重启手机

5, 下载apk安装器, 下载并且安装

<http://www.uushare.com/user/f11429/file/3408824>

6,关联后 直接双击你的 apk 程序 即可完成安装~!

最后建议下载 [91助手](#) 来管理, 很好用。。。。

7, 如何将Android手机切换为U盘?

将手机和电脑通过USB线连接, 滑动轨迹球到屏幕左上边顶端的状态栏, 会有个USB的图标, 点击“USB 已连接”这个通知
点击USB图标, 选择“装载” 即可把手机的SD卡装载到手机上, 手机即可变身为U盘。

注意: 在U盘模式, 可以从电脑往手机SD卡copy文件。但无法使用PC端的APK安装器安装软件。

8, 如何关闭U盘模式?

如果想关掉U盘模式, 切换回PC连接模式, 只要在此滑动屏幕顶端的状态栏, 并点击USB图标。

然后选择卸载，即可关闭USB的U盘模式。

See:

<http://www.hiapk.com/bbs/viewthread.php?tid=5020&highlight=usb%C7%FD%B6%AF>
http://s.jbbs.zol.com.cn/1/33672_147.html

Linode mysql : Got error 28 from storage engine



环境: linode(16G磁盘) + centos 5.5

前些天一直收到 linode的 磁盘IO 警告的邮件，工作繁忙，也没有管它，殊不知 隔了一天 我的blog 不能访问了，ssh进入linode跟踪log，mysql报 Got error 28 from storage engine，后来查出是磁盘已经用完的原因，解决过程如下：

1, 查看磁盘占用情况

```
[root@l1165-150 system]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/xvda        16G   15G    0   100% /
```

发现磁盘已经占满，接下来需要找出谁占用了这么大的空间

```
[root@l1165-150 system]# cd /
[root@l1165-150 /]# du -sh *
5.4M    bin
4.0K    boot
32K     dev
13M     etc
56M     home
21M     lib
16K     lost+found
4.0K    media
4.0K    mnt
4.0K    opt
15G     var
```

发现 var 目录占用最大，继续跟踪 cd /var，du -sh * 最后跟踪到是 /var/log/maillog 占用最大，足足有 14G，删除之

```
>rm /var/log/maillog
```

此时 使用 df -h 查看空间占用情况 还是 15G，需要释放删除的空间

2,重启 centos，释放空间

登入 linode.com reboot 你的centos，就可以 释放删除文件的磁盘了，还有 就是发现 linode.com的显示的磁盘占用情况依然是 100%，实际我进入系统查看，空闲磁盘已经有 9G了

3, 重启之前已经开启的服务

```
/etc/init.d/mysqld start # 启动mysql
/usr/local/system/nginx/sbin/nginx # 启动nginx
thin start -C /etc/thin/thin.yml # 启动thin
/etc/init.d/httpd start # 启动 apache
/etc/init.d/postfix start # 启动邮件相关服务
/etc/init.d/dovecot start
/etc/init.d/saslauthd start
cd /usr/local/system/www/short_url/lib
nohup ruby mongoshort.rb -e production & # 启动sinatra
cd /usr/local/system
nohup mongod --dbpath=/usr/local/system/mongodb/data & # 启动mongodb
cd /usr/local/system
svnserv -d -r vcs # 启动svn server
```

在启mongodb的时候，报文件锁定，无法启动，需要 repair 一下

```
mongodb/bin/mongod --repair
```

4, ok, 大功告成

See:

<http://www.mongodb.org/display/DOCS/Durability+and+Repair>
<http://www.fufuok.com/mysql-got-error-28.html>

postfix 关闭open-relay



环境:centos 5.5 + postfix 2.3.3

```
[root@1165-150 ~]# postconf mail_version  
mail_version = 2.3.3
```

上次, <http://wxianfeng.com/2010/08/20/linode-mysql-got-error-28-from-storage-engine>, 刚刚解决了blog不能访问的问题, 由于磁盘占满的原因, 一天时间没到我的磁盘又被占满了, 还是maillog在一直增长的原因, 经过查找原因, 原来我的mail server变成了肉鸡, 被人拿来发送大量的垃圾邮件了, 刚开始配的时候不太懂, 没注意 open relay的情况, <http://wxianfeng.com/2010/08/01/centos-install-mail-server>, 所以, 必须关系open relay功能~!

1, 什么是open — relay ?

顾名思义, relay的转发的意思, Open-Relay (开放转发或匿名转发)是指由于邮件服务器不理睬邮件发送者或邮件接受者的是否为系统所设定的用户, 而对所有的入站邮件一律进行转发 (RELAY) 的功能。通常, 若邮件服务器的此功能开放, 则我们一般称此邮件服务器是Open-Relay的

2, 怎么测我的mail server 是不是 open relay 的 ?

```
[root@1165-150 ~]# telnet mail.wxianfeng.com 25  
Trying 173.230.155.150...  
Connected to mail.wxianfeng.com.  
Escape character is '^'.  
220 mail.wxianfeng.com ESMTP Postfix  
mail from:ss@163.com  
250 2.1.0 Ok  
rcpt to:wang.fl_1429@gmail.com  
554 5.7.1 <wang.fl_1429@gmail.com>: Relay access denied
```

上面出现了 Relay access denied, 则说明你的mail server 不是 open relay的, 如果可以成功发送邮件, 则说明 你的mail server 是 open relay的, 另外, 还可以通过以下网址测试:

<http://verify.abuse.net/relay.html>

直接输入 mail.wxianfeng.com 测试

3,open-relay 后, 被人盗寄的症状

>mailq 查看邮件队列, 会发现里面有大量的邮件等待发送

>tail -f /var/log/maillog

出现了大量的像下面这样的log:

```
2010-08-21T05:52:31.424284-04:00 11165-150 postfix/qmgr[8366]: 0C50427536: to=<xlaga@yahoo.com.tw>, relay=none, delay=141354, delays=141354/0.04/0/0, dsn=4.7.1, status=deferred  
(delivery temporarily suspended: host mx2.mail.tw.yahoo.com[203.188.197.10] refused to talk to me: 421 4.7.1 [TS03] All messages from 173.230.155.150 will be permanently  
deferred; Retrying will NOT succeed. See http://postmaster.yahoo.com/errors/421-ts03.html)  
2010-08-21T05:52:31.425279-04:00 11165-150 postfix/qmgr[8366]: 0C50427536: to=<xocy@yahoo.com.tw>, relay=none, delay=141354, delays=141354/0.04/0/0, dsn=4.7.1, status=deferred  
(delivery temporarily suspended: host mx2.mail.tw.yahoo.com[203.188.197.10] refused to talk to me: 421 4.7.1 [TS03] All messages from 173.230.155.150 will be permanently  
deferred; Retrying will NOT succeed. See http://postmaster.yahoo.com/errors/421-ts03.html)  
2010-08-21T05:52:31.426311-04:00 11165-150 postfix/qmgr[8366]: 0C50427536: to=<xup6ru4vm0@yahoo.com.tw>, relay=none, delay=141354, delays=141354/0.04/0/0, dsn=4.7.1,  
status=deferred (delivery temporarily suspended: host mx2.mail.tw.yahoo.com[203.188.197.10] refused to talk to me: 421 4.7.1 [TS03] All messages from 173.230.155.150 will be  
permanently deferred; Retrying will NOT succeed. See http://postmaster.yahoo.com/errors/421-ts03.html)
```

```
2010-08-21T05:52:31.427198-04:00 11165-150 postfix/qmgr[8366]: 0C50427536: to=<y2003625@yahoo.com.tw>, relay=none, delay=141354, delays=141354/0.05/0/0, dsn=4.7.1,
status=deferred (delivery temporarily suspended: host mx2.mail.tw.yahoo.com[203.188.197.10] refused to talk to me: 421 4.7.1 [TS03] All messages from 173.230.155.150 will be
permanently deferred; Retrying will NOT succeed. See http://postmaster.yahoo.com/errors/421-ts03.html)
2010-08-21T05:52:31.428079-04:00 11165-150 postfix/qmgr[8366]: 0C50427536: to=<yck1012.tw@yahoo.com.tw>, relay=none, delay=141354, delays=141354/0.05/0/0, dsn=4.7.1,
status=deferred (delivery temporarily suspended: host mx2.mail.tw.yahoo.com[203.188.197.10] refused to talk to me: 421 4.7.1 [TS03] All messages from 173.230.155.150 will be
permanently deferred; Retrying will NOT succeed. See http://postmaster.yahoo.com/errors/421-ts03.html)
2010-08-21T05:52:31.429149-04:00 11165-150 postfix/qmgr[8366]: 0C50427536: to=<yfjtcommu@yahoo.com.tw>, relay=none, delay=141354, delays=141354/0.05/0/0, dsn=4.7.1,
status=deferred (delivery temporarily suspended: host mx2.mail.tw.yahoo.com[203.188.197.10] refused to talk to me: 421 4.7.1 [TS03] All messages from 173.230.155.150 will be
permanently deferred; Retrying will NOT succeed. See http://postmaster.yahoo.com/errors/421-ts03.html)
2010-08-21T05:52:31.430203-04:00 11165-150 postfix/qmgr[8366]: 0C50427536: to=<yfkl34@yahoo.com.tw>, relay=none, delay=141354, delays=141354/0.05/0/0, dsn=4.7.1, status=deferred
(delivery temporarily suspended: host mx2.mail.tw.yahoo.com[203.188.197.10] refused to talk to me: 421 4.7.1 [TS03] All messages from 173.230.155.150 will be permanently
deferred; Retrying will NOT succeed. See http://postmaster.yahoo.com/errors/421-ts03.html)
2010-08-21T05:52:31.431290-04:00 11165-150 postfix/qmgr[8366]: 0C50427536: to=<yi05@yahoo.com.tw>, relay=none, delay=141354, delays=141354/0.05/0/0, dsn=4.7.1, status=deferred
(delivery temporarily suspended: host mx2.mail.tw.yahoo.com[203.188.197.10] refused to talk to me: 421 4.7.1 [TS03] All messages from 173.230.155.150 will be permanently
deferred; Retrying will NOT succeed. See http://postmaster.yahoo.com/errors/421-ts03.html)
```

从log中可以看出，你的mail server 正在发往yahoo的email，但是全部被拒绝了，

4，为什么 会被人盗寄？

网络上 会有很多人用 port scan工具扫描端口，当你的25端口，被人测出是open relay的，这些人就会充分利用你的mail server 来发垃圾邮件，发垃圾邮件的后果，可能会导致你的ip被封，被世界垃圾邮件组织把你ip 拉入黑名单，这时你的mail server 也就是废物了。

5，如果关闭 盗寄 的邮件？

我的mail server 有大量的 发送队列，怎么清除，用下面命令

```
[root@11165-150 mail]# postsuper -d ALL
postsuper: Deleted: 61178 messages
```

发现我清楚了 61178 封垃圾邮件队列，这样你的log 应该会停止了，不会再狂刷了。。。

6，如何关闭open-relay ？

配置 postfix的 mynetworks

```
mynetworks = 127.0.0.1/32 # ip/netmask
```

我的整个 postfix main.cf 配置：

```
[root@11165-150 ~]# postconf -n
alias_database = hash:/etc/aliases
alias_maps = hash:/etc/aliases
broken_sasl_auth_clients = yes
command_directory = /usr/sbin
config_directory = /etc/postfix
daemon_directory = /usr/libexec/postfix
debug_peer_level = 2
home_mailbox = Maildir/
html_directory = no
inet_interfaces = Smyhostname, localhost, 173.230.155.150 # 设置postfix服务监听的网络接口 通常是将所有的网络接口都开放，以便接收任何网络接口的邮件
mail_owner = postfix
mailq_path = /usr/bin/mailq.postfix
manpage_directory = /usr/share/man
mydestination = Smyhostname,Smydomain,localhost.Smydomain, localhost, Smydomain,mail.Smydomain,www.Smydomain # 设置可接收邮件的主机名称或域名
mydomain = wxianfeng.com # 邮件域名
myhostname = mail.wxianfeng.com # 邮件主机名
mynetworks = 127.0.0.1/32 # 设置可转发 (Relay) 哪些网络的邮件
myorigin = Smydomain # 由本机寄出的域名
newaliases_path = /usr/bin/newaliases.postfix
queue_directory = /var/spool/postfix
readme_directory = /usr/share/doc/postfix-2.3.3/README_FILES
relay_domains = Smydestination # 设置可转发哪些网络的邮件
sample_directory = /usr/share/doc/postfix-2.3.3/samples
sendmail_path = /usr/sbin/sendmail.postfix
setgid_group = postdrop
smtpd_recipient_restrictions = permit_mynetworks,permit_sasl_authenticated,reject_unauth_destination
smtpd_sasl_auth_enable = yes # SASL来完成SMTP的SMTP-AUTH功能,postfix 本身没有认证机制
smtpd_sasl_local_domain = wxianfeng.com
smtpd_sasl_path = private/auth
smtpd_sasl_security_options = noanonymous
smtpd_sasl_type = dovecot
unknown_local_recipient_reject_code = 550
virtual_mailbox_domains = Smydomain
```

See:

<http://www.raidhttpd.com/jlbb/viewtopic.php?p=82160&sid=661a10a71ded2a6280fdd5ad6ba20101>
<http://www.linuxgoo.com/2005/66127/10433644999.html>
http://blog.csdn.net/daisy_cheung/archive/2009/01/16/3795087.aspx
<http://www.cnblogs.com/newversion/articles/1490910.html>

centos install imagemagick + rmagick



环境:centos 5.5 + imagemagick 6.2.8 + rmagick 1.15.17

原本想安装最新版的rmagick的, 但是发现编译ImageMagick安装不是少这个就是少那个, 一气之下, 最后还是 yum 安装的比较快。。。。

如果你已经yum安装了, 但是想再编译安装, 需要先卸载原来的:

```
>yum remove ImageMagick
```

安装过程:

```
yum install ImageMagick
yum install ImageMagick-devel
gem install rmagick -v=1.15.17 --no-rdoc --no-ri -- --disable-htmdoc # 注意后面的参数需要加上, 没加上会报错
```

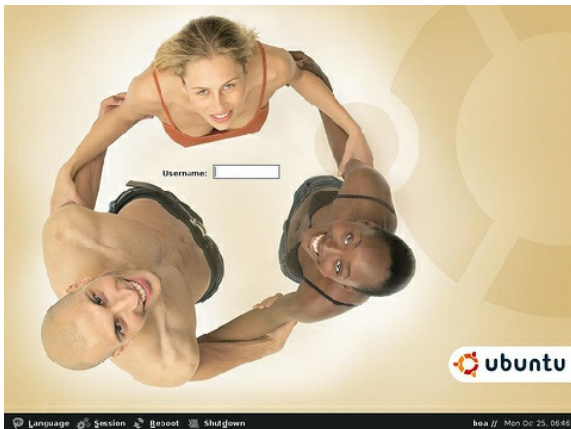
安装 rmagick时主要还有版本问题, 出错的话, 会提示你ImageMagick必须大于多少版本, 那么如何查看ImageMagick版本?

```
>convert -version
```

SEE:

<http://stackoverflow.com/questions/1254366/problem-installing-rmagick-rubygem-on-centos-5>

Linux 登录日志



环境:centos 5.5

在这之前我的服务器密码很简单, 感觉不安全, 于是改的复杂一点了, 顺手对我的登录日志分析了一下, 还真发现了很多来破解root密码的远程ip, 还记得之前在公司有一次上下班忘记打卡, 行政找到我必须找出证据, 于是从 我工作电脑中调出当天的登入登出信息 最好不过了, 记得当时不太熟悉, 还是在同事帮助下立即解决的, 另外合理的分析登录日志对你服务器的安全也有很大的作用~!

1.来看看你的服务器被哪些ip登录过（root密码错误的）

```
[root@li165-150 log]# grep "Failed password for root" /var/log/secure | awk '{print $9}' | uniq -c | sort -nr # awk分割, uniq -c 别重 + 显示重复次数, sort -nr 降序(-r) + 数值排序(-n)
3116 69.162.125.77
224 173.230.145.104
111 184.106.229.209
84 60.217.234.142
76 222.237.78.139
51 210.193.16.75
46 88.191.73.232
13 109.169.56.3
4 220.165.28.67
2 115.248.49.217
1 88.191.52.74
1 221.221.173.88
1 220.181.147.187
```

从上面可以看出有许多ip试图登录我的服务器, 其中 69.162.125.77 就有 3116次失败记录, 总共统计下:

```
[root@li165-150 log]# grep "Failed password for root" /var/log/secure | awk '{print $9}' | wc -l
13730
```

哈, 原来这么多人喜欢扫描我的服务器啊。。。。。

2, 从 系统日志里查找 root 的相关信息

```
[root@li165-150 log]# cd /var/log
[root@li165-150 log]# find ./ -type f -name "messages*" | xargs grep "root" | more
./messages:2010-08-30T12:54:25.987395-04:00 li165-150 avahi-daemon[2862]: Successfully dropped root privileges.
./messages:2010-08-30T12:54:26.023483-04:00 li165-150 avahi-daemon[2862]: Successfully called chroot().
./messages.1:2010-08-20T22:55:03.457101-04:00 li165-150 avahi-daemon[2928]: Successfully dropped root privileges.
./messages.1:2010-08-20T22:55:03.504958-04:00 li165-150 avahi-daemon[2928]: Successfully called chroot().
./messages.1:2010-08-20T22:56:22.025464-04:00 li165-150 avahi-daemon[3047]: Successfully dropped root privileges.
./messages.1:2010-08-20T22:56:22.054287-04:00 li165-150 avahi-daemon[3047]: Successfully called chroot().
./messages.1:2010-08-21T14:21:13.824695-04:00 li165-150 avahi-daemon[2816]: Successfully dropped root privileges.
./messages.1:2010-08-21T14:21:13.907610-04:00 li165-150 avahi-daemon[2816]: Successfully called chroot().
```

3, 当前登录用户的信息记录在文件/var/run/utmp 中,这是一个二进制文件 用普通的 tail, cat 等工具是无法查看的, 但是可以通过一些命令来查看文件的信息, 例如who

当前登录用户:

```
[root@li165-150 log]# who # 或者 users 命令
root    ttypl      Aug 31 12:13 (220.181.147.187)
root    ttypt2     Aug 31 12:53 (220.181.147.187)
```

```
[root@li165-150 log]# who -b # 上次启动时间
system boot  Aug 30 12:54
```

更多的who命令请>who --help

4.所有登录进入和退出纪录在文件/var/log/wtmp中, 也是二进制文件

```
[root@li165-150 log]# last -l0 root # 查看root用户最后10次的登录情况
root    ttypt2     220.181.147.187  Tue Aug 31 12:53    still logged in
root    ttypl      220.181.147.187  Tue Aug 31 12:13    still logged in
root    ttypt0     220.181.147.187  Tue Aug 31 09:06 - 13:16 (04:10)
root    ttypt0     221.221.173.88   Mon Aug 30 22:40 - 22:42 (00:01)
root    ttypt0     220.181.147.187  Mon Aug 30 12:55 - 13:11 (00:16)
root    ttypt0     220.181.147.187  Mon Aug 30 12:51 - down (00:01)
root    ttypl      221.221.8.126    Fri Aug 27 03:01 - 04:14 (01:12)
root    ttypt0     125.34.211.119   Fri Aug 27 01:47 - 04:21 (02:33)
root    ttypt0     125.34.211.119   Wed Aug 25 23:41 - 23:41 (00:00)
root    ttypt0     221.221.8.126    Wed Aug 25 21:44 - 21:46 (00:01)
```

wtmp begins Wed Jun 2 09:10:09 2010

```
[root@li165-150 log]# last # last命令往回搜索wtmp,来显示自从文件第一次创建以来登录过的用户
root    ttypt2     220.181.147.187  Tue Aug 31 12:53    still logged in
root    ttypl      220.181.147.187  Tue Aug 31 12:13    still logged in
root    ttypt0     220.181.147.187  Tue Aug 31 09:06 - 13:16 (04:10)
root    ttypt0     221.221.173.88   Mon Aug 30 22:40 - 22:42 (00:01)
root    ttypt0     220.181.147.187  Mon Aug 30 12:55 - 13:11 (00:16)
reboot  system boot  2.6.32.16-linode Mon Aug 30 12:54    (1+00:55)
root    ttypt0     220.181.147.187  Mon Aug 30 12:51 - down (00:01)
root    ttypl      221.221.8.126    Fri Aug 27 03:01 - 04:14 (01:12)
root    ttypt0     125.34.211.119   Fri Aug 27 01:47 - 04:21 (02:33)
root    ttypt0     125.34.211.119   Wed Aug 25 23:41 - 23:41 (00:00)
root    ttypt0     221.221.8.126    Wed Aug 25 21:44 - 21:46 (00:01)
root    ttypt0     125.34.211.119   Wed Aug 25 21:34 - 21:37 (00:02)
```

5, 最后一次登录记录在 /var/log/lastlog 这个文件中,可以用lastlog 查看

```
[root@li165-150 log]# lastlog
Username      Port      From      Latest
root          ttypt2    220.181.147.187  Tue Aug 31 12:53:34 -0400 2010
```

6, 清除登录log

```
>rm -f /var/log/wtmp
>cat /dev/null > /var/log/lastlog
```

另外今天学了一招历史命令的用法, 之前都是用上箭头调出, 若是调到前面第10个就需要10下, 麻烦, 可以用history命令搞定

```
[root@li165-150 log]# history 10 # 最近10条
1085 last
1086 last
1087 last
1088 lastlog
```

```
1089 lastlog -u wxianfeng
1090 lastlog
1091 last
1092 rm -f /var/log/wtmp
1093 last
1094 history 10
```

>history -c # 清除 history

SEE:

<http://zhiwei.li/text/2010/06/linux%E7%99%BB%E5%BD%95%E6%97%A5%E5%BF%97/>

mini_magick 简单处理图片



最近blog换了皮肤，以前是最大宽度，现在宽度限制死了，有的图片超过了宽度，很丑陋，于是用minimagick对所有的图片统一缩放了下，这下图片的大小刚刚好，缩放比resize的效果要好，图片不会扭曲难看，minimagick和rmagick都是调用imagemagick的ruby接口，使用起来很方便。。。。

1, 缩放（也就是我用来处理我blog里图片的脚本）

```
require 'rubygems'
require 'mini_magick'

# path = "E:/Rubyproject/wxianfeng_com/public/files/" # windows路径
path = "/usr/local/system/www/wxianfeng_com/shared/public/files/"
files = Dir.open(path).to_a.select{|x| x != '.' && x != '..' && x != '.svn' && x != 'Thumbs.db'}
imgs = files.select { |f| f != /^(thumb_|middle)/ }

imgs.each do |ele|
  p ele
  img_path = path + ele
  img = MiniMagick::Image.from_file(img_path)
  w,h = img[:width],img[:height]
  percent = ((480/w.to_f) * 100).to_i
  img.combine_options do |c|
    c.sample "#{percent}%" # 缩放
  end
  img.write(img_path)
end
```

2,resize

```
image = MiniMagick::Image.from_file("input.jpg") # or MiniMagick::Image.new("input.jpg")
image.resize "100x100" # or image.thumbnail "100x100"
image.write("output.jpg")
```

3,裁剪

```
require 'mini_magick'
img = MiniMagick::Image.from_file "1.jpg" #取得宽度和高度
w,h = img[:width],img[:height] #> [2048, 1536]
shaved_off = ((w-h)/2).round #> 256
img.shave "#{shaved_off}x0" #此处表示宽度上左右各截取256个像素，高度上截取0像素
img.write "2.jpg"
```

4,旋转

```
image = MiniMagick::Image.from_file("input.jpg")
image.combine_options do |c|
  c.rotate "-90>" # 旋转 90 度
end
image.write("input.jpg") # 同名 替换掉原来的
```

SEE:

http://github.com/probablycorey/mini_magick

<http://www.blogkid.net/archives/2154.html>

迁移Blog数据到本地



为了更好的调试，今天把我远程服务器上的blog数据同步到本地了，一切都是手动做的，要能够有自动化工具就好了，就像 capistrano 部署一样，通过ssh隧道实现远程操作，暂且这样，以后再想办法。。。

1, 备份远程数据库的数据，注意只是数据，没有表结构

```
>mysqldump -t -uroot -proot wxianfeng_com > db.txt
```

2,清空本地数据库中所有表的数据 （ruby脚本）

```
require 'rubygems'
require 'active_record'

ActiveRecord::Base.establish_connection(
  :adapter => "mysql",
  :host => "localhost",
  :database => "wxianfeng_com",
  :encoding => "utf8",
  :username => "root",
  :password => "root"
)

begin
  ActiveRecord::Base.connection.tables.each do |table|
    p "TRUNCATEing #{table}"
    ActiveRecord::Base.connection.execute("TRUNCATE #{table}")
  end
rescue
  $stderr.puts "Ooops Error"
end
```

3,导入 数据

```
>mysql -uroot -proot wxianfeng_com < db.txt
```

4,备份远程服务器上的资源文件 （图片，文件等）

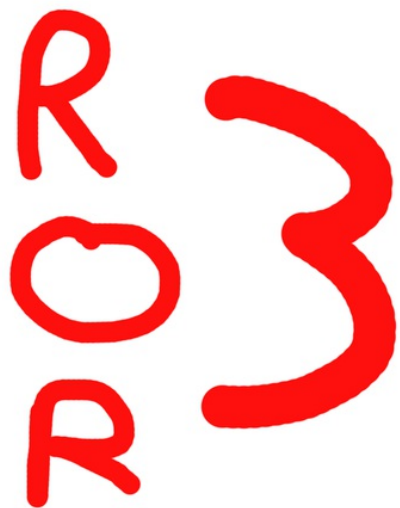
```
>tar -zcvf directory.tar.gz directory
```

5,用 windows 下ssh客户端，例如 ssh secure shell client ，把远程文件down到本地

6, ok。。。搞定～！

最近写的比较多 哈， 权当笔记了，同时加大google收录数，提升流量，为下一步 计划 打好准备.....

windows install ruby 1.9.2 + rails 3.0.0



不幸,宿舍的电脑又挂掉了,这次装了个win 7,win 7 果然气质不凡,概念版的QQ很炫的说,不过跑起来有点卡,还是XP下的qq跑得流畅,郁闷的是qq2010,安装上运行不起来,报错,这不是必win 7 用概念版的吗?!..切入正题:

最终环境:win 7 + ruby 1.9.2 + rails 3.0.0.rc2 + mysql 5.1

目前rails3的demo跑起来了,由于以前的项目都是跑在2.x系列的,所以以后还要折腾.....

1,安装 ruby 1.9.2 and install

<http://rubyforge.org/frs/download.php/72170/rubyinstaller-1.9.2-p0.exe>

注意安装上ruby 1.9.2 就不需要再安装rubygems了,这个和以前ruby 1.8.X 的有区别,ruby 1.9.2 已经自带rubygems了,可以>gem -v 查看

2,安装rails

```
F:\>gem install rails --pre --no-ri --no-rdoc
ERROR: Error installing rails:
  activemodel requires builder (~> 2.1.2, runtime)
```

接下来你只需要运行gem install rails --pre --no-ri --no-rdoc 即可,缺少什么装什么,一直装到gem install rails --pre --no-ri --no-rdoc 成功即可,例如上面错误,继续安装缺少的依赖的库

```
F:\>gem install builder --no-ri --no-rdoc
```

为什么要用--no-ri --no-rdoc 参数,因为加上去安装实在是慢,反正装了也不用,还不如不装rdoc!

3,新建rails3 project

```
>rails new rails3 -d mysql
```

4,修改database.yml

5,创建数据库

```
>rake db:create RAILS_ENV=development
```

6,启动rails3 server

```
>rails server
```

你会发现报如下错误:

```
E:\Rubyproject\rails3>rails server
Could not find gem 'mysql2 (>= 0, runtime)' in any of the gem sources.
Try running `bundle install`.
```

发现rails3.0连接mysql的adapter不再是mysql了,是mysql2,这个好办,我们还用之前的mysql

修改项目下Gemfile

```
gem 'mysql2' 改为 gem 'mysql'
```

同理修改config下的database.yml mysql2 → mysql

7,ok,你的demo已经可以跑起来了

如果你的mysql还是连不上,可能是libmysql.dll 问题,see:
<http://wxianfeng.com/2010/06/23/windows-xp-ruby-on-rails-mysql>

除此之外,还有一种更简单的方法,安装devkit,但是我没有成功,gem
install rails --pre 安装失败,see
: <http://skim.la/2010/08/21/rails-3-rc-on-windows-using-rubyinstaller-192/>

see:

<http://skim.la/2010/02/07/rails-3-beta-on-windows-using-rubyinstaller-187-rc2/>
<http://osdir.com/ml/RubyonRailsTalk/2010-09/msg00480.html>

大胡子Katz离开Rails社区



Rails社区核心人物之一Katz今天离开了engineyard, 离开了rails社区, 有点遗憾, 这个人我想rails社区的fans都应该知道, katz在kungfurails的时候来过中国, 并且作了rails3的演讲, 视频见此:
http://v.youku.com/v_show/id_XMTI4NDQ20TIw.html, katz做过很多我们耳熟能详的项目, Rails, Merb, jQuery, datamapper 等等, 并且最近刚发布的rails3主要贡献者就是katz, 把merb的想法带入了rails, Yehuda Katz 在[blog](#)里说明离开的原因和去向, 并且说明了未来三年的计划, 他的去向是去开发 [sproutcore](#)了, sproutcore是apple公司开源出来的, 然而sproutcore的创造者在今年7月8日离开了apple公司, 创办了自己的公司, 依据sproutcore开发更好的桌面体验般的webapp, sproutcore创始人准备将sproutcore打造为html5的框架, 为移动互联网, IPAD等提供良好的应用体验, 看来HTML5也是大势所趋啊....

无论怎样,Good luck Katz,必定你为rails社区贡献了很多, katz承诺虽然离开了engineyard, 但是rails3.1的开发工作还将由他主导, 说不定也因为他rails+sproutcore会很好的结合起来呢, over。。。

吼歪一下祝贺Katz进入HTML5 team

See:

<http://www.javaeye.com/news/17731-yehuda-kats-leave-engine-yard>

<http://www.infoq.com/cn/news/2009/09/sproutcore-1-0>

北京前门大街偶遇街拍MM

中秋游玩北京前门大街, 偶遇街拍MM, 话不多说, 看图:

-
-
-
-
-
-
-
-

-
-
-
-
-
-

更多pp
SEE :

<http://www.flickr.com/photos/wxianfeng/5023518682/>

常见Google搜索技巧



个人在平时使用google.com是总结出来的常见google搜索技巧:

1,简单数学计算

1.23 * 20

2,查看天气预报

beijing weather

3,查看时间

beijing time

4,搜索文件, 指定格式

rails filetype:ppt

5,查找指定站点

rails site:wxianfeng.com

6,排除站点查找

rails -site:wxianfeng.com

7,查找外链

link:www.dhgate.com

8,查找相似网站

related:www.dhgate.com

9,查看google收录数

site:wxianfeng.com

10,搜索手机号码归属地(g.cn)

13888888888

11,搜索指定的词组

"ruby on rails"

12,排除关键字搜索

-rails site:wxianfeng.com

13. 指定多个关键字

`ruby || rails`

`ruby OR rails`

14. 查询汇率

`1 dollars in RMB`

15. 查询网站title中含有指定关键字

`intitle:wxianfeng`

16. 查询url含有指定关键字的

`inurl:wxianfeng`

17. 网页内容包含的关键词

`intext:wxianfeng`

`go on`

SEE:

<http://dongxi.net/b00rT>

<http://www.williamlong.info/archives/728.html>

mysql 控制台中文乱码解决



环境:centos 5.5 + mysql 5.0

最近给我的blog生成了[pdf文档](#),发现上了服务器后生成的pdf中文都是乱码,本地都是好的,后来发现我进入mysql 的console中select出来的中文都是乱码,解决之:

查看字符编码:

```
mysql> SHOW VARIABLES LIKE 'charac%';
```

Variable_name	Value
character_set_client	utf8
character_set_connection	utf8
character_set_database	utf8
character_set_filesystem	binary
character_set_results	utf8
character_set_server	utf8
character_set_system	utf8
character_sets_dir	/usr/share/mysql/charsets/

8 rows in set (0.00 sec)

这个是正确的设置,原来查看出来的是latin1,即mysql自带的编码集,发现rails中只要你database.yml中设置了encoding:utf8就可以了,哪怕mysql的character_set_*是latin1,那么你存入数据库中的中文也不是乱码,so,rails中设置encoding非常之重要

那么如何解决mysql console中乱码,修改配置文件:

```
>vim /etc/my.cnf
```

```
[client]
default-character-set = utf8
```

mysqld 和 clcient 区域中配置上面的两句即可

然后重启mysql

```
ruby html+css->pdf
```



环境:centos 5.5 + ruby 1.8.7 + pdftk 0.4.6

最近给我的blog生成了[pdf文档](#)，每半年生成一个pdf文档，尝试不少开源组件，发现好用的不多，其中比较好的两个是[prawn](#) 和 [pdfkit](#)，

最后实验下来pdfkit 很好用，可以把html + css 转化为pdf文档，底层使用了[wkhtmltopdf](#)，而且wkhtmltopdf可以为shell直接调用，非常方便

1,安装pdfkit

```
>gem install pdfkit
```

```
>sudo pdftk --install-wkhtmltopdf
```

报 1zcat 找不到，安装之

```
>yum update
```

```
>yum install lzma
```

2,生成pdf文档的脚本

#注意<http://wxianfeng.com> 必须存在 a 链接，因为 wkhtmltopdf 可以直接对 url 抓取生成 pdf

```
require 'rubygems'
require 'pdfkit/source'
require 'pdfkit/pdfkit'
require 'pdfkit/middleware'
require 'pdfkit/configuration'
```

```
PDFKit.configure do |config|
  config.wkhtmltopdf = '/usr/local/bin/wkhtmltopdf'
end
```

```
range_t = [
    ["2009-06-30", "2009-12-12 23:59:59"],
    ["2009-12-12", "2010-06-31 23:59:59"],
    ["2010-06-31", "2010-12-12 23:59:59"]
]
```

```
path = "/usr/local/system/src/wxianfeng_com_pdf/"
exist files = Dir.open(path).to_a.select{|x| x != '.' && x != '..' && x != '.svn' && x != 'Thumbs.db'}
```

```

range_t.each do |ele|
  next if exist_files.include?("wxianfeng.com_#{ele.first}~#{ele.last.slice(/\d+~\d+\/)}.pdf")
  posts = Content.all(:conditions=>["published_at BETWEEN ? AND ?",ele.first,ele.last])
  kit , html = nil , ''
  posts.each do |i|
    p i.published_at.to_s(:db) + " " + i.title
    html << "<strong>" + i.title + "</strong><br><br>" + i.html(:all).gsub(/[\$%<br>]([a-zA-Z]:\V{1,3}\s*))/, '<a href="\1">\1</a>') + "<br><br><br><br>"
  end
  kit = PDFKit.new(html)
  # kit.stylesheets << "#{RAILS_ROOT}/themes/indholmen/stylesheets/main.css"
  # kit.stylesheets << "#{RAILS_ROOT}/themes/indholmen/stylesheets/print.css"
  # kit.stylesheets << "#{RAILS_ROOT}/themes/indholmen/stylesheets/local.css"
  kit.to_pdf
  kit.to_file "/usr/local/system/src/wxianfeng.com/pdf/wxianfeng.com_#{ele.first}~#{ele.last.slice(/\d+~\d+\/)}.pdf"
end

```

end

3. 运行

```
ruby script/runner script/tools/generate_pdf.rb
```

注意script/runner 调用的事development指定的db