**老男孩linux运维实战培训**

**老男孩教育教学核心思想6重：重目标、重思路、重方法、重实践、重习惯、重总结**

**学无止境，老男孩教育成就你人生的起点！**

**版权声明：**

本文作者为《老男孩linux运维实战培训》学生—**你的名字**

本文的所有内容均来自老男孩培训**命令总结**，未经本人及老男孩培训许可，禁止私自转发及使用。

QQ: **你的**XXXX

E-mail: **你的**XXXX

**联系方式:**

|  |  |  |  |
| --- | --- | --- | --- |
| 网站运维交流群： | | | |
| **Linux QQ交流群 385168604** | | **架构师 QQ交流群 390642196** | |
| **Python QQ交流群 29215534** | | **大数据 QQ交流群 421358633** | |
| =================================================================================== | | | |
| **老男孩linux实训联系方式** |  | |  |
| **咨询QQ** | **41117397（丹丹）** | | **70271111（歪歪）** |
|  | **80042789（飞雪）** | | **390320151（小雨）** |
|  | **41117483（冰冰）** | |  |
| **电话：** | **158-1059-0206（丹丹）** | | **189-1171-8229（歪歪）** |
|  | **135-5261-2571（飞雪）** | | **186-0046-2391（小雨）** |
| **网站:** | <http://www.etiantian.org> | | <http://www.oldboyedu.com> |
| **博客:** | http://oldboy.blog.51cto.com | | <http://blog.oldboyedu.com> |

**快捷方式说明:**

**ctrl + 1 一级标题**

**ctrl + 2 二级标题**

**ctrl + 3 三级标题**

**ctrl + 5 程序代码**

**ctrl + 6 正文**

**格式约定：**

蓝色字体：内容注释

目 录

[YUM仓库及RPM包定制课程总结 1](#_Toc512694180)

[第1章 定制RPM包 1](#_Toc512694181)

[1.1 企业面试题：上周基础运维部的小王交付给我100台服务器，我想在这100台服务器中，其中10台NFS存储，20台Mysq服务器，40台Redis服务器，20台Nginx+PHP服务器，10台反向代理服务器，你有什么方案快速部署并优化吗？需要多少时间搞定？ 1](#_Toc512694182)

[1.2 历史回顾-软件安装的方式 1](#_Toc512694183)

[1.2.1 定制rpm包 2](#_Toc512694184)

[1.3 Rpm-red hat包管理器 2](#_Toc512694185)

[1.4 FPM打包工具 2](#_Toc512694186)

[1.5 实战-定制nginx的rpm包 4](#_Toc512694187)

[1.5.1 编译安装nginx 5](#_Toc512694188)

[1.5.2 编写脚本 5](#_Toc512694189)

[1.5.3 Fpm-打包 6](#_Toc512694190)

[1.6 Fpm命令与rpmbuild命令关系 6](#_Toc512694191)

[1.6.1 查看rpm包信息 7](#_Toc512694192)

[1.7 Fpm打包注意事项 8](#_Toc512694193)

[1.8 安装rpm包 8](#_Toc512694194)

[第2章 YUM仓库搭建 9](#_Toc512694195)

[2.1 创建yum仓库目录 10](#_Toc512694196)

[2.2 安装createrepo软件 10](#_Toc512694197)

[2.3 初始化repodata索引文件 10](#_Toc512694198)

[2.4 添加新的rpm包 10](#_Toc512694199)

[2.5 提供yum服务 11](#_Toc512694200)

[2.6 6. 客户端配置 12](#_Toc512694201)

[2.7 Yum安装定制的nginx程序 13](#_Toc512694202)

[2.7.1 Yum安装nginx 13](#_Toc512694203)

YUM仓库及RPM包定制课程总结

# 定制RPM包

先抛砖引玉

## 企业面试题：上周基础运维部的小王交付给我100台服务器，我想在这100台服务器中，其中10台NFS存储，20台Mysq服务器，40台Redis服务器，20台Nginx+PHP服务器，10台反向代理服务器，你有什么方案快速部署并优化吗？需要多少时间搞定？

解决方案：

1、tar打包：先编译安装、打包、分发、解包；

2、saltstack、puppet、ansible批量部署管理；

3、定制rpm、yum仓库、yum安装

4、openstack虚拟机镜像和docker容器分发

## 历史回顾-软件安装的方式

* 编译安装

优点：定制化安装目录；

按需开启功能；

可定制（定制安装目录和软件功能）、一定情况下不需要网络、可选择软件版本。

缺点：需要朝赵并试验出合适的编译参数；

软件版本升级，参数已经取消了，并报错提示非常明智

Mysql,php编译时间长

慢、复杂、找依赖麻烦

* Yum安装软件

优点：全自动化安装

简单、便捷

不需要为依赖问题发愁

缺点：

自主性太差

需要网络、网络不好下载速度慢；

没办法定制，存放位置固定，不好迁移

* 二进制安装

优点：简单、块

缺点：不能定制、包容量大（mysql的二进制包近200M）

### 定制rpm包

根据自己的需求编译软件-->制作RPM包-->搭建内网YUM仓库-->yum安装

结合前两者的优点，暂未发现什么缺点。可能的缺点就是RPM包的通用性差，只能适用本公司的环境。

缺点：第一步编译安装复杂、打包不能再次更改

定制rpm包是互联网企业运维自动化必备技能

## Rpm-red hat包管理器

Rpm命令有五种基本功能；安装、卸载、升级、查询、验证

## FPM打包工具

|  |  |
| --- | --- |
| 常用参数 | 描述 |
| -s | 指定源类型 |
| -t | 指定目标类型，即想要制作为什么包 |
| -n | 指定包名称 |
| -v | 指定包版本号 |
| -C | 指定打包的相对路径 |
| -d | 指定依赖那些包 |
| -f | 第二次打包时目录下如果有同名安装包存在，则覆盖 |
| -p | 输出的安装包的目录，不想放在当前目录下就需要指定 |
| --post-install | 软件安装完后所需要的脚本 |
| --pre-install | 软件安装完之前所需要的脚本 |
| --post-uninstall | 软件卸载完成后所需要的脚本 |
| --pre-uninstall | 软件卸载完成之前所需要的脚本 |

支持的源类型包

DIR 将目录打包成所需要的类型,可以用于源码编译的安装包.

RPM    对rpm进行转换

gem    对rubygem包进行转换.

python    将python模块打包成相应的类型

支持的目标类型包格式:

RPM    转换为RPM包

deb     转换为deb包

solaris    转换为solaris包

puppet   转换为puppet包

安装FPM工具:

    fpm是使用ruby语言编写的,因此系统环境需要ruby这个软件,并且**版本号需要大于1.8.5**

## 安装ruby模块

|  |
| --- |
| yum  -y install ruby rubygems ruby-devel  ## 安装ruby的模块,ruby模块需要大于1.8.5 |

## 添加阿里云的rubygems仓库,移除原生的yum仓库

|  |
| --- |
| gem sources -a <http://mirrors.aliyun.com/rubygems/> ##增加阿里云的仓库  gem sources --remove  [http://rubygems.org](http://rubygems.org/)   ##移除原有的仓库 |

##安装FPM工具

|  |
| --- |
| gem install fpm -v 1.3.3     ##指定安装1.3.3 版本,我们需求不大,只要能打包就足够了,没必要追求高版本 |

Fpm是ruby编写，因此系统吸引ruby且大于1.8.5

安装ruby

[root@m01 ~]# yum -y install ruby rubygems ruby-devel

安装fpm

[root@m01 ~]# gem sources -a http://mirrors.aliyun.com/rubygems/

http://mirrors.aliyun.com/rubygems/ added to sources

[root@m01 ~]# gem sources --remove http://rubygems.org/

http://rubygems.org/ removed from sources

报错信息

[root@m01 ~]# gem install fpm -v 1.3.3

ERROR: Error installing fpm:

json requires Ruby version >= 1.9.

[root@m01 ~]# ruby -v

ruby 1.8.7 (2013-06-27 patchlevel 374) [x86\_64-linux]

遇到以上错误请先安装以下两个软件,然后继续安装fpm1.3.3:

[root@m01 ~]# gem install json -v 1.8.3

[root@m01 ~]# gem install ffi -v 1.9.18

再次安装fpm

[root@m01 ~]# gem install fpm -v 1.3.3

**快捷命令**

yum -y install ruby rubygems ruby-devel

gem sources -a http://mirrors.aliyun.com/rubygems/

gem sources --remove http://rubygems.org

gem install fpm -v 1.3.3

gem install json -v 1.8.3

gem install ffi -v 1.9.18

gem install fpm -v 1.3.3

## 实战-定制nginx的rpm包

[root@m01 ~]# cat /etc/redhat-release

CentOS release 6.7 (Final)

[root@m01 ~]# uname -r

2.6.32-573.el6.x86\_64

[root@m01 ~]# getenforce

Disabled

[root@m01 ~]# /etc/init.d/iptables status

iptables：未运行防火墙。

###设置yum缓存#####

[root@m01 tools]# cat /etc/yum.conf

[main]

cachedir=/var/cache/yum/$basearch/$releasever

keepcache=1 🡨设置为1 yum后包保存下来/var/cache/yum/x86\_64/6/base/packages/

######清空本机已经有的yum缓存

(find /var/cache/ -type f -name "\*.rpm")

[root@m01 ~]# cd /application/

[root@m01 application]# ll

[root@m01 application]# mkdir -p tools

Wget编译包

[root@m01 tools]# wget http://10.0.0.61/nginx-1.6.3.tar.gz

--2018-04-28 09:45:37-- http://10.0.0.61/nginx-1.6.3.tar.gz

正在连接 10.0.0.61:80... 已连接。

已发出 HTTP 请求，正在等待回应... 200 OK

长度：805253 (786K) [application/x-gzip]

正在保存至: “nginx-1.6.3.tar.gz”

100%[=====================================>] 805,253 --.-K/s in 0.08s

2018-04-28 09:45:42 (9.28 MB/s) - 已保存 “nginx-1.6.3.tar.gz” [805253/805253])

#####编译安装nginx (就安装我们上传的1.6.3版本)

yum -y install pcre-devel openssl-devel   #安装必备的包

[root@m01 base]# find packages/ -type f -name "\*rpm"

packages/krb5-devel-1.10.3-65.el6.x86\_64.rpm

packages/libselinux-utils-2.0.94-7.el6.x86\_64.rpm

packages/libss-1.41.12-23.el6.x86\_64.rpm

packages/openssl-1.0.1e-57.el6.x86\_64.rpm

packages/libselinux-devel-2.0.94-7.el6.x86\_64.rpm

packages/libselinux-2.0.94-7.el6.x86\_64.rpm

packages/zlib-devel-1.2.3-29.el6.x86\_64.rpm

packages/libsepol-devel-2.0.41-4.el6.x86\_64.rpm

packages/libcom\_err-1.41.12-23.el6.x86\_64.rpm

packages/keyutils-libs-devel-1.4-5.el6.x86\_64.rpm

packages/krb5-libs-1.10.3-65.el6.x86\_64.rpm

packages/pcre-devel-7.8-7.el6.x86\_64.rpm

packages/e2fsprogs-1.41.12-23.el6.x86\_64.rpm

packages/e2fsprogs-libs-1.41.12-23.el6.x86\_64.rpm

packages/libcom\_err-devel-1.41.12-23.el6.x86\_64.rpm

packages/libkadm5-1.10.3-65.el6.x86\_64.rpm

packages/openssl-devel-1.0.1e-57.el6.x86\_64.rpm

[root@m01 base]# find packages/ -type f -name "\*rpm"|xargs cp -t /tmp/

[root@m01 ~]# cd /tmp/ && tar zcf nginx\_yum.tar.gz \*.rpm

### 编译安装nginx

[root@m01 tmp]# useradd nginx -M -s /sbin/nologin

[root@m01 tmp]# cd /application/tools/

[root@m01 tools]# tar xf nginx-1.6.3.tar.gz

[root@m01 tools]# cd nginx-1.6.3

[root@m01 nginx-1.6.3]# ./configure --prefix=/application/nginx-1.6.3 --user=nginx --group=nginx --with-http\_ssl\_module --with-http\_stub\_status\_module

[root@m01 nginx-1.6.3]#make && make install

[root@m01 nginx-1.6.3]# ln -s /application/nginx-1.6.3/ /application/nginx

### 编写脚本

[root@m01 ~]# mkdir -p /server/scripts/

[root@m01 scripts]# vim nginx\_rpm.sh

#!/bin/bash

useradd nginx -M -s /sbin/nologin

ln -s /application/nginx-1.6.3/ /application/nginx

### Fpm-打包

fpm -s dir -t rpm -n nginx -v 1.6.3 -d 'pcre-devel,openssl-devel' --post-install /server/scripts/nginx\_rpm.sh -f /application/nginx-1.6.3

格式语法解释：

-s 指定目录源类型 –t 打包类型 rpm -n 包名称 –v 版本 –d ‘依赖包以逗号分隔多个’

--post-install 安装完程序后执行的脚本 -f 重名强制覆盖 ，可不带

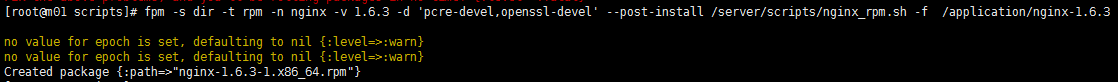
整体类似tar 苹果+框

[root@m01 scripts]# fpm -s dir -t rpm -n nginx -v 1.6.3 -d 'pcre-devel,openssl-devel' --post-install /server/scripts/nginx\_rpm.sh -f /application/nginx-1.6.3

no value for epoch is set, defaulting to nil {:level=>:warn}

no value for epoch is set, defaulting to nil {:level=>:warn}

Created package {:path=>"nginx-1.6.3-1.x86\_64.rpm"}



[root@m01 scripts]# sz nginx-1.6.3-1.x86\_64.rpm

### 封装php5.5.32

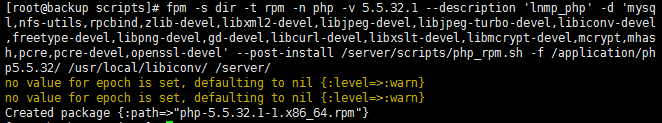
**前提是先安装编译好mysql**

*[root@backup ~]# cat /server/scripts/php\_rpm.sh*

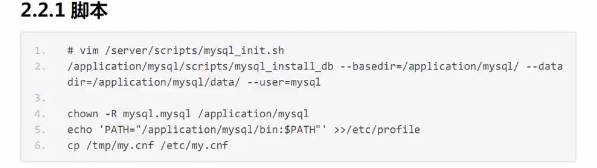
*#!/bin/bash*

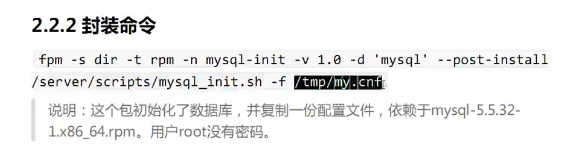
*useradd -s /sbin/noligin -M www*

*ln -s /application/php5.5.32/ /application/php*

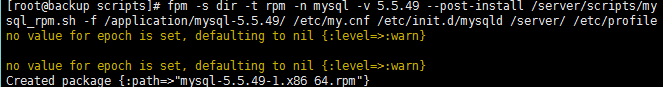


### 封装mysql



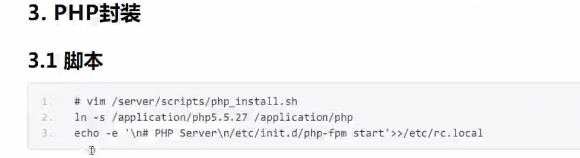


fpm -s dir -t rpm -n mysql -v 5.5.49 --post-install /server/scripts/mysql\_rpm.sh -f /application/mysql-5.5.49/ /etc/my.cnf /etc/init.d/mysqld /server/ /etc/profile

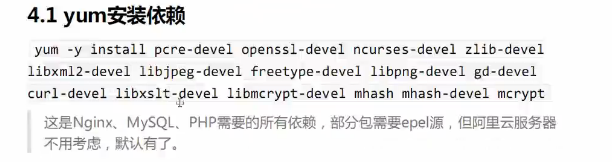


### Php封装

fpm -s dir -t rpm -n php -v 5.5.32 -d 'zlib-devel,libxml2-devel,libjpeg-devel,libjpeg-turbo-devel,libiconv-devel,freetype-devel,openssl-devel,libpng-devel,gd-devel,libcurl-devel,libxslt-devel,libmcrypt-devel,mcrypt,mhash' --post-install /server/scripts/php\_rpm.sh -f /application/php5.5.32/ /usr/local/libiconv/ /server/





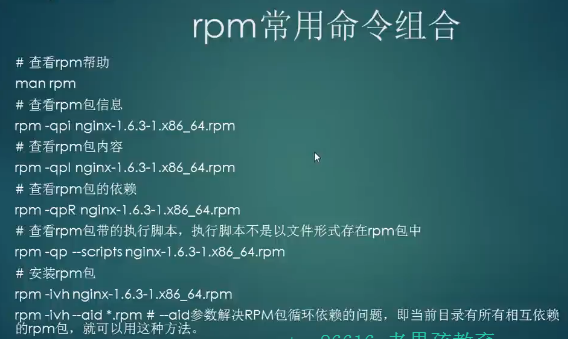


## Fpm命令与rpmbuild命令关系

Fpm命令其实是对rpmbuild命令进行封装，其实底层还是rpmbuild命令，因此系统必须要有rpmbuild命令

yum –y install rpm-build

### 查看rpm包信息



#### 查看包内容：

[root@m01 scripts]# rpm -qpl nginx-1.6.3-1.x86\_64.rpm

/application/nginx-1.6.3/conf/fastcgi.conf

/application/nginx-1.6.3/conf/fastcgi.conf.default

/application/nginx-1.6.3/conf/fastcgi\_params

/application/nginx-1.6.3/conf/fastcgi\_params.default

/application/nginx-1.6.3/conf/koi-utf

/application/nginx-1.6.3/conf/koi-win

/application/nginx-1.6.3/conf/mime.types

/application/nginx-1.6.3/conf/mime.types.default

/application/nginx-1.6.3/conf/nginx.conf

/application/nginx-1.6.3/conf/nginx.conf.default

/application/nginx-1.6.3/conf/scgi\_params

/application/nginx-1.6.3/conf/scgi\_params.default

/application/nginx-1.6.3/conf/uwsgi\_params

/application/nginx-1.6.3/conf/uwsgi\_params.default

/application/nginx-1.6.3/conf/win-utf

/application/nginx-1.6.3/html/50x.html

/application/nginx-1.6.3/html/index.html

/application/nginx-1.6.3/logs

/application/nginx-1.6.3/sbin/nginx

#### 查看包依赖

[root@m01 scripts]# rpm -qpR nginx-1.6.3-1.x86\_64.rpm

pcre-devel

openssl-devel

/bin/sh

rpmlib(PayloadFilesHavePrefix) <= 4.0-1

rpmlib(CompressedFileNames) <= 3.0.4-1

#### 查看执行脚本，执行脚本不是以文件形式存在rpm中

[root@m01 scripts]# rpm -qp --scripts nginx-1.6.3-1.x86\_64.rpm

postinstall scriptlet (using /bin/sh):

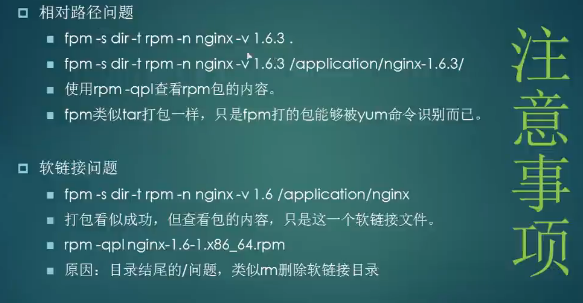
#!/bin/bash

useradd nginx -M -s /sbin/nologin

ln -s /application/nginx-1.6.3/ /application/nginx

#fpm -s dir -t fpm -n nginx -v 1.6.3 -d ‘pcer-devel,openssl-devel’ --post-install /server/scripts/nginx\_rpm.sh -f /application/nginx-1.6.3/

## Fpm打包注意事项



使用绝对路径，不要使用相对路径

软连接问题

## 安装rpm包

上传两个包依赖tar包以及nginx的rpm包

[root@m01 tools]# ll -h

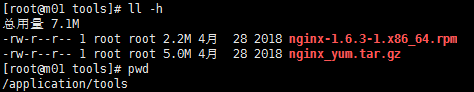
总用量 7.1M

-rw-r--r-- 1 root root 2.2M 4月 28 2018 nginx-1.6.3-1.x86\_64.rpm

-rw-r--r-- 1 root root 5.0M 4月 28 2018 nginx\_yum.tar.gz

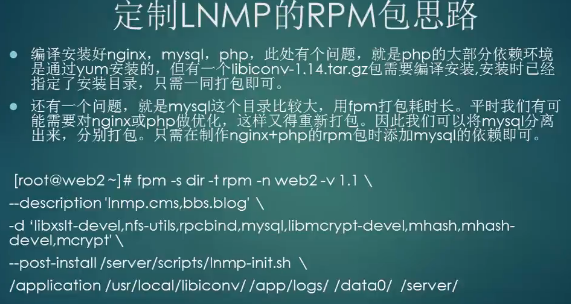
[root@m01 tools]# pwd

/application/tools



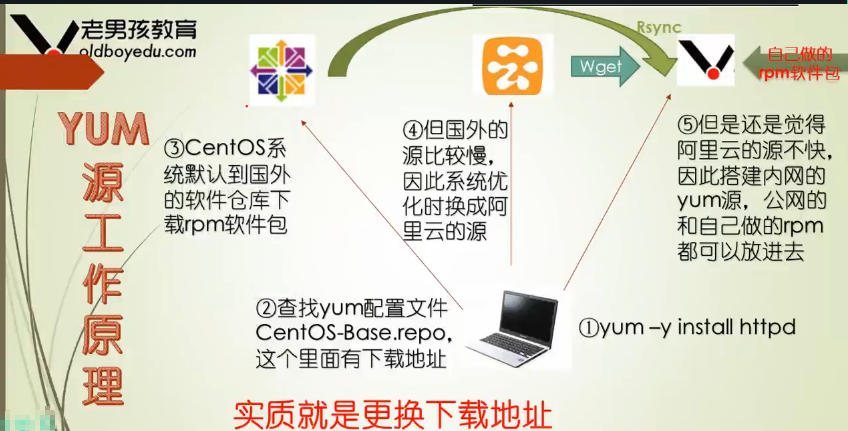
[root@m01 tools]# tar xf nginx\_yum.tar.gz

[root@m01 tools]# rpm -ivh --aid \*rpm



# YUM仓库搭建

Repository(软件仓库)多种rpm软件包的集合



一般企业内部最佳是搭建自己的YUM仓库，配置rpm定制包。

实战

## 创建yum仓库目录

mkdir -p /application/yum/centos6.6/x86\_64/

cd /application/yum/centos6.6/x86\_64/

\cp tools/\* /application/yum/centos6.6/x86\_64/

rz # 上传rpm包到此目录，此目录下面还可以包括文件夹

## 安装createrepo软件

yum install createrepo –y

## 初始化repodata索引文件

createrepo -pdo /application/yum/centos6.6/x86\_64/ /application/yum/centos6.6/x86\_64/

## 添加新的rpm包

# 只下载软件不安装

yumdownloader pcre-devel openssl-devel

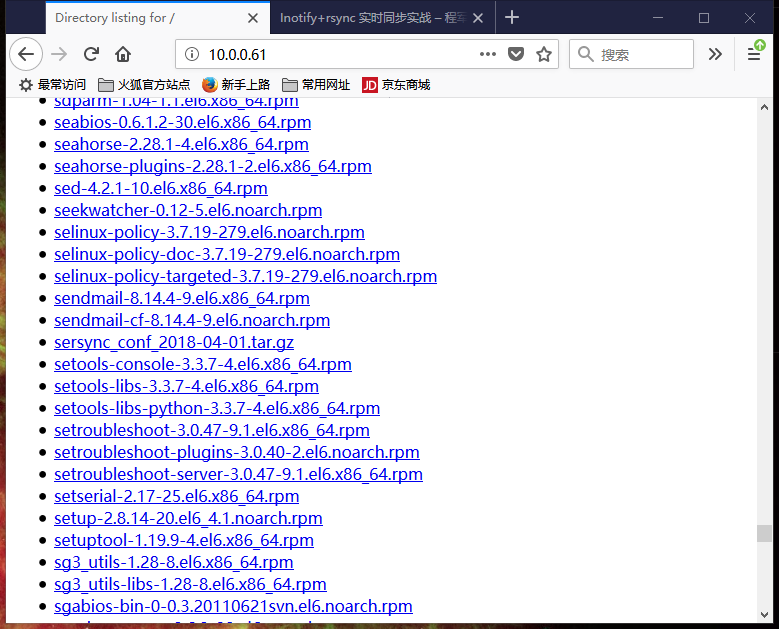
createrepo --update /application/yum/centos6.6/x86\_64/

## 提供yum服务

# 可以用Apache或nginx提供web服务，但用Python的http模块更简单，适用于内网环境

cd /application/yum/centos6.6/x86\_64/

python -m SimpleHTTPServer 80 &>/dev/null &



平时yum安装软件时不删除安装包

# cat /etc/yum.conf

keepcache=1

# 安装包存储目录

cachedir=/var/cache/yum/$basearch/$releasever

# /var/cache/yum/x86\_64/6/base/packages

## 6. 客户端配置

先避免干扰，把yum.repos.d目录下的源移动到新建的tmp目录下

# cd /etc/yum.repos.d

[root@B yum.repos.d]# vi yum.repo

[oldboy]

name=Server

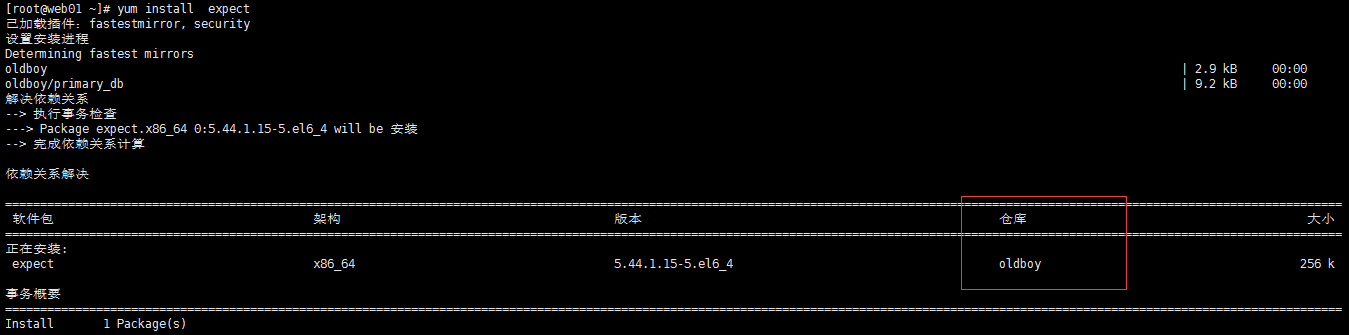
baseurl=http://10.0.0.61

enable=1

gpgcheck=0

[root@YUM ~]# yum --enablerepo=yum --disablerepo=base,extras,updates,epel list

# 指定使用yum库



上面是临时使用内网yum源，想永久并简单使用yum -y install lrzsz命令，就需要修改配置文件将默认的repo文件关闭。

[root@oldboy ~]# cd /etc/yum.repos.d/

[root@oldboy yum.repos.d]# vim CentOS-Base.repo

1. # 在每一个启动的源加上
2. # enabled=0 #改为1就启用，没有此参数也是启用。

[base]

…………

enabled=0

[updates]

…………

enabled=0

[extras]

…………

enabled=0

1. # 还有其他开启的仓库就使用这个办法关闭。

## 镜像yum源

上面只是将自己制作的rpm包，放入yum源。但还有一种企业需求，说的更具体一点，平时学生上课yum安装软件都是从公网下载的，占用带宽，因此在学校里搭建一个内网yum服务器，但又考虑到学生回家也要使用yum安装软件，如果yum软件的数据库文件repodata不一样，就会有问题。因此我想到的解决方法就是直接使用公网yum源的repodata。

1. 镜像同步公网yum源
2. 上游yum源必须要支持rsync协议，否则不能使用rsync进行同步。
3. http://mirrors.ustc.edu.cn/status/
4. CentOS官方标准源：rsync://mirrors.ustc.edu.cn/centos/
5. epel源：rsync://mirrors.ustc.edu.cn/epel/
6. 同步命令：
7. # 使用rsync同步yum源，为了节省带宽、磁盘和下载时间，我只同步了CentOS6的rpm包，这样所有的rpm包只占用了21G，全部同步需要300G左右。
8. # 同步base源，小技巧，我们安装系统的光盘镜像含有部分rpm包，大概3G，这些就不用重新下载。

/usr/bin/rsync -av rsync://mirrors.ustc.edu.cn/centos/6/os/x86\_64/ /data/yum\_data/centos/6/os/x86\_64/

/usr/bin/rsync -av rsync://mirrors.ustc.edu.cn/centos/6/extras/x86\_64/ /data/yum\_data/centos/6/extras/x86\_64/

/usr/bin/rsync -av rsync://mirrors.ustc.edu.cn/centos/6/updates/x86\_64/ /data/yum\_data/centos/6/updates/x86\_64/

#### # epel源

/usr/bin/rsync -av --exclude=debug rsync://mirrors.ustc.edu.cn/epel/6/x86\_64/ /data/yum\_data/epel/6/x86\_64/

1. 学生使用内网yum源方法
2. # 可以自建一个内网dns，如果没有，可使用hosts解析。

echo '192.168.0.200 mirrors.aliyun.com' >>/etc/hosts

1. 结果展示

[root@KVM data]# du -sh yum\_data

21G yum\_data

[root@KVM data]# tree -L 3 yum\_data/

yum\_data/

├── centos

│ ├── 6

│ │ ├── extras

│ │ ├── os

│ │ └── updates

│ └── RPM-GPG-KEY-CentOS-6

├── epel

│ └── 6

│ └── x86\_64

## Yum安装定制的nginx程序

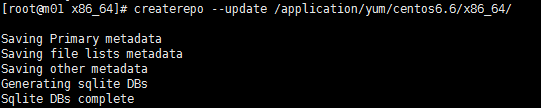
现将fpm阶段弄好的两个包上传到yum仓库中

nginx-1.6.3-1.x86\_64.rpm

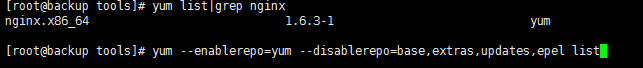
nginx\_yum.tar.gz

再同步一下仓库索引

createrepo --update /application/yum/centos6.6/x86\_64/



**客户端需要同步yum仓库数据**



### Yum安装nginx

**yum install nginx -y**

