Netkiller Web 手札

陈景峯 著

Apache HTTP Server

NGINX

Netkiller Linux Web 手札

目录

范例清单

Netkiller Linux Web 手札

Apache, Lighttpd, Nginx, Resin, Tomcat, Jboss, Zope...

ISBN#

Mr. Neo Chan, 陈景峯(BG7NYT)

中国广东省深圳市望海路半岛城邦三期
518067
+86 13113668890

<netkiller@msn.com>

$Date: 2013-04-10 15:03:49 +0800 (Wed, 10 Apr 2013) $, $Id: book.xml 559 2013-04-10 07:03:49Z netkiller $

电子书最近一次更新于 2021-09-23 20:10:29

Netkiller 系列电子书 http://www.netkiller.cn
Netkiller Web 手札
陈景峯 著
Apache HTTP Server
NGINX
知乎: netkiller | Bilibili: netkiller | QQ: 13721218 | 微信: 13113668890

http://www.netkiller.cn
http://netkiller.github.io
http://netkiller.sourceforge.net
微信订阅号 netkiller-ebook
微信：13113668890 请注明“读者”
QQ：13721218 请注明“读者”
QQ群：128659835 请注明“读者”
知乎专栏 | 多维度架构

2017-02-13

致读者

Netkiller 系列手札 已经被 Github 收录，并备份保存在北极地下250米深的代码库中，备份会保留1000年。

Preserving open source software for future generations

The world is powered by open source software. It is a hidden cornerstone of modern civilization, and the shared heritage of all humanity.

The GitHub Arctic Code Vault is a data repository preserved in the Arctic World Archive (AWA), a very-long-term archival facility 250 meters deep in the permafrost of an Arctic mountain.

We are collaborating with the Bodleian Library in Oxford, the Bibliotheca Alexandrina in Egypt, and Stanford Libraries in California to store copies of 17,000 of GitHub’s most popular and most-depended-upon projects—open source’s “greatest hits”—in their archives, in museum-quality cases, to preserve them for future generations.

https://archiveprogram.github.com/arctic-vault/

自述

Netkiller 系列电子书 http://www.netkiller.cn
Netkiller Web 手札
陈景峯 著
Apache HTTP Server
NGINX
知乎: netkiller | Bilibili: netkiller | QQ: 13721218 | 微信: 13113668890

《Netkiller 系列 手札》是一套免费系列电子书，netkiller 是 nickname 从1999 开使用至今，“手札” 是札记，手册的含义。

2003年之前我还是以文章形式在BBS上发表各类技术文章，后来发现文章不够系统，便尝试写长篇技术文章加上章节目录等等。随着内容增加，不断修订，开始发布第一版，第二版......

```
IT知识变化非常快，而且具有时效性，这样发布非常混乱，经常有读者
发现第一版例子已经过时，但他不知道我已经发布第二版。

我便有一种想法，始终维护一个文档，不断更新，使他保持较新的版本不
过时。

第一部电子书是《PostgreSQL 实用实例参考》开始我使用
Microsoft Office Word 慢慢随着文档尺寸增加 Word 开始表现出
力不从心。

我看到PostgreSQL 中文手册使用SGML编写文档，便开始学习
Docbook SGML。使用Docbook写的第一部电子书是《Netkiller
Postfix Integrated Solution》这是Netkiller 系列手札的原
型。

至于“手札”一词的来历，是因为我爱好摄影，经常去一个台湾摄影网
站，名字就叫“摄影家手札”。

由于硬盘损坏数据丢失 《Netkiller Postfix Integrated
Solution》 的 SGML文件已经不存在；Docbook SGML存在很多缺陷
UTF-8支持不好，转而使用Docbook XML.
```

目前技术书籍的价格一路飙升，动则￥80，￥100，少则￥50，￥60. 技术书籍有时效性，随着技术的革新或淘汰，大批书记成为废纸垃圾。并且这些书技术内容雷同，相互抄袭，质量越来越差，甚至里面给出的例子错误百出，只能购买影印版，或者翻译的版本。

在这种背景下我便萌生了自己写书的想法，资料主要来源是我的笔记与例子。我并不想出版，只为分享，所有我制作了基于CC License 发行的系列电子书。

本书注重例子，少理论（捞干货），只要你对着例子一步一步操作，就会成功，会让你有成就感并能坚持学下去，因为很多人遇到障碍就会放弃，其实我就是这种人，只要让他看到希望，就能坚持下去。

1. 写给读者

为什么写这篇文章

有很多想法,工作中也用不到所以未能实现，所以想写出来,和大家分享.有一点写一点,写得也不好,只要能看懂就行,就当学习笔记了.

开始零零碎碎写过一些文档，也向维基百科供过稿，但维基经常被ZF封锁，后来发现sf.net可以提供主机存放文档，便做了迁移。并开始了我的写作生涯。

这篇文档是作者20年来对工作的总结,是作者一点一滴的积累起来的，有些笔记已经丢失，所以并不完整。

因为工作太忙整理比较缓慢。目前的工作涉及面比较窄所以新文档比较少。

我现在花在技术上的时间越来越少，兴趣转向摄影，无线电。也想写写摄影方面的心得体会。

写作动力:

曾经在网上看到外国开源界对中国的评价，中国人对开源索取无度，但贡献却微乎其微.这句话一直记在我心中，发誓要为中国开源事业做我仅有的一点微薄贡献

另外写文档也是知识积累，还可以增加在圈内的影响力.

人跟动物的不同,就是人类可以把自己学习的经验教给下一代人.下一代在上一代的基础上再创新,不断积累才有今天.

所以我把自己的经验写出来,可以让经验传承

没有内容的章节:

目前我自己一人维护所有文档，写作时间有限，当我发现一个好主题就会加入到文档中，待我有时间再完善章节，所以你会发现很多章节是空无内容的.

文档目前几乎是流水帐试的写作，维护量很大，先将就着看吧.

我想到哪写到哪,你会发现文章没一个中心,今天这里写点,明天跳过本章写其它的.

文中例子绝对多,对喜欢复制然后粘贴朋友很有用,不用动手写,也省时间.

理论的东西,网上大把,我这里就不写了,需要可以去网上查.

我爱写错别字,还有一些是打错的,如果发现请指正.

文中大部分试验是在Debian/Ubuntu/Redhat AS上完成.

写给读者

至读者:

我不知道什么时候，我不再更新文档或者退出IT行业去从事其他工作，我必须给这些文档找一个归宿，让他能持续更新下去。

我想捐赠给某些基金会继续运转，或者建立一个团队维护它。

我用了20年时间坚持不停地写作，持续更新，才有今天你看到的《Netkiller 手扎》系列文档，在中国能坚持20年，同时没有任何收益的技术类文档，是非常不容易的。

有很多时候想放弃，看到外国读者的支持与国内社区的影响，我坚持了下来。

中国开源事业需要各位参与，不要成为局外人，不要让外国人说：中国对开源索取无度，贡献却微乎其微。

我们参与内核的开发还比较遥远，但是进个人能力，写一些文档还是可能的。

系列文档

下面是我多年积累下来的经验总结，整理成文档供大家参考:

Netkiller Architect 手札
Netkiller Developer 手札
Netkiller PHP 手札
Netkiller Python 手札
Netkiller Testing 手札

2. 作者简介

陈景峯 (ㄔㄣ ㄐㄧㄥ ㄈㄥ)

Nickname： netkiller | English name: Neo chen | Nippon name: ちんけいほう (音訳) | Korean name: 천징봉 | Thailand name: ภูมิภาพภูเขา | Vietnam: Trần Cảnh Phong

Callsign: BG7NYT | QTH: ZONE CQ24 ITU44 ShenZhen, China

程序猿，攻城狮，挨踢民工, Full Stack Developer, UNIX like Evangelist, 业余无线电爱好者（呼号：BG7NYT）,户外运动，山地骑行以及摄影爱好者。

《Netkiller 系列 手札》 的作者

成长阶段

1981年1月19日(庚申年腊月十四)出生于黑龙江省青冈县建设乡双富大队第一小队

1989年9岁随父母迁居至黑龙江省伊春市，悲剧的天朝教育，不知道那门子归定，转学必须降一级，我本应该上一年级，但体制让我上学前班，那年多都10岁了

1995年小学毕业，体制规定借读要交3000两银子(我曾想过不升初中)，亲戚单位分楼告别平房，楼里没有地方放东西，把2麻袋书送给我，无意中发现一本电脑书BASIC语言，我竟然看懂了，对于电脑知识追求一发而不可收，后面顶零花钱，压岁钱主要用来买电脑书《MSDOS 6.22》《新编Unix实用大全》《跟我学Foxbase》。。。。。。

1996年第一次接触UNIX操作系统，BSD UNIX, Microsoft Xinux(盖茨亲自写的微软Unix，知道的人不多)

1997年自学Turbo C语言，苦于没有电脑，后来学校建了微机室才第一次使用QBASIC(DOS 6.22 自带命令)，那个年代只能通过软盘拷贝转播，Trubo C编译器始终没有搞到，

1997年第一次上Internet网速只有9600Bps, 当时全国兴起各种信息港域名格式是www.xxxx.info.net, 访问的第一个网站是NASA下载了很多火星探路者拍回的照片，还有“淞沪”sohu的前身

1998~2000年在哈尔滨学习计算机，充足的上机时间，但老师让我们练打字（明伦五笔/WT）打字不超过80个/每分钟还要强化训练，不过这个给我的键盘功夫打了好底。

1999年学校的电脑终于安装了光驱，在一张工具盘上终于找到了Turbo C, Borland C++与Quick Basic编译器，当时对VGA图形编程非常感兴趣，通过INT33中断控制鼠标，使用绘图函数模仿windows界面。还有操作 UCDOS 中文字库，绘制矢量与点阵字体。

2000年沉迷于Windows NT与Back Office各种技术，神马主域控制器，DHCP，WINS，IIS，域名服务器，Exchange邮件服务器，MS Proxy, NetMeeting...以及ASP+MS SQL开发；用56K猫下载了一张LINUX。ISO镜像，安装后我兴奋的24小时没有睡觉。

职业生涯

2001 年来深圳进城打工,成为一名外来务工者. 在一个4人公司做PHP开发，当时PHP的版本是2.0, 开始使用Linux Redhat 6.2.当时很多门户网站都是用FreeBSD,但很难搞到安装盘，在网易社区认识了一个网友,从广州给我寄了一张光盘，FreeBSD 3.2

2002 年我发现不能埋头苦干,还要学会"做人".后辗转广州工作了半年，考了一个Cisco CCNA认证。回到深圳重新开始，在车公庙找到一家工作做Java开发

2003 年这年最惨,公司拖欠工资16000元,打过两次官司2005才付清.

2004 年开始加入分布式计算团队,目前成绩，工作仍然是Java开发并且开始使用PostgreSQL数据库。

2004-10月开始玩户外和摄影

2005-6月成为中国无线电运动协会会员,呼号BG7NYT,进了一部Yaesu FT-60R手台。公司的需要转回PHP与MySQL，相隔几年发现PHP进步很大。在前台展现方面无人能敌，于是便前台使用PHP，后台采用Java开发。

2006 年单身生活了这么多年,终于找到归宿. 工作更多是研究PHP各种框架原理

2007 物价上涨,金融危机，休息了4个月（其实是找不到工作），关外很难上439.460中继，搞了一台Yaesu FT-7800.

2008 终于找到英文学习方法，《Netkiller Developer 手札》，《Netkiller Document 手札》

2008-8-8 08:08:08 结婚,后全家迁居湖南省常德市

2009《Netkiller Database 手札》,2009-6-13学车，年底拿到C1驾照

2010 对电子打击乐产生兴趣，计划学习爵士鼓。由于我对Linux热爱，我轻松的接管了公司的运维部，然后开发运维两把抓。我印象最深刻的是公司一次上架10个机柜，我们用买服务器纸箱的钱改善伙食。我将40多台服务器安装BOINC做压力测试，获得了中国第二的名次。

2011 平凡的一年，户外运动停止，电台很少开，中继很少上，摄影主要是拍女儿与家人，年末买了一辆山地车

2012 对油笔画产生了兴趣，活动基本是骑行银湖山绿道，

2013 开始学习民谣吉他，同时对电吉他也极有兴趣；最终都放弃了。这一年深圳开始推数字中继2013-7-6日入手Motorola

MOTOTRBO XIR P8668，Netkiller 系列手札从Sourceforge向Github迁移；年底对MYSQL UDF，Engine与PHP扩展开发产生很浓的兴趣，拾起遗忘10+年的C，写了几个mysql扩展（图片处理，fifo管道与ZeroMQ），10月份入Toyota Rezi 2.5V并写了一篇《攻城狮的苦逼选车经历》

2014-9-8 在淘宝上买了一架电钢琴 Casio Privia PX-5S pro 开始陪女儿学习钢琴，由于这家钢琴是合成器电钢，里面有打击乐，我有对键盘鼓产生了兴趣。

2014-10-2号罗浮山两日游，对中国道教文化与音乐产生了兴趣，10月5号用了半天时间学会了简谱。10月8号入Canon 5D Mark III + Canon Speedlite 600EX-RT香港过关被查。

2014-12-20号对乐谱制作产生兴趣（https://github.com/SheetMusic/Piano），给女儿做了几首钢琴伴奏曲，MuseScore制谱然后生成MIDI与WAV文件。

2015-09-01 晚饭后拿起爵士鼓基础教程尝试在Casio Privia PX-5S pro演练，经过反复琢磨加上之前学钢琴的乐理知识，终于在02号晚上，打出了简单的基本节奏，迈出了第一步。

2016 对弓箭（复合弓）产生兴趣，无奈天朝法律法规不让玩。每周游泳轻松1500米无压力，年底入 xbox one s 和 Yaesu FT-2DR, 同时开始关注功放音响这块

2017 7月9号入 Yamaha RX-V581 功放一台，连接Xbox打游戏爽翻了，入Kindle电子书，计划学习蝶泳，果断放弃运维和开发知识体系转攻区块链。

2018 从溪山美地搬到半岛城邦，丢弃了多年攒下的家底。11 月开始玩 MMDVM，使用 Yaesu FT-7800 发射，连接MMDVM中继板，树莓派，覆盖深圳湾，散步骑车通联两不误。

2019 卖了常德的房子，住了5次院，哮喘反复发作，决定停止电子书更新，兴趣转到知乎，B站

2020 准备找工作

职业生涯路上继续打怪升级

3. 如何获得文档

下载 Netkiller 手札 (epub,kindle,chm,pdf)

EPUB https://github.com/netkiller/netkiller.github.io/tree/master/download/epub

MOBI https://github.com/netkiller/netkiller.github.io/tree/master/download/mobi

PDF https://github.com/netkiller/netkiller.github.io/tree/master/download/pdf

CHM https://github.com/netkiller/netkiller.github.io/tree/master/download/chm

通过 GIT 镜像整个网站

https://github.com/netkiller/netkiller.github.com.git

$ git clone https://github.com/netkiller/netkiller.github.com.git

镜像下载

整站下载

```
wget -m http://www.netkiller.cn/index.html
```

指定下载

```
wget -m wget -m http://www.netkiller.cn/linux/index.html
```

Yum 下载文档

获得光盘介质，RPM包，DEB包，如有特别需要，请联系我

YUM 在线安装电子书

http://netkiller.sourceforge.net/pub/repo/

```
# cat >> /etc/yum.repos.d/netkiller.repo <<EOF
[netkiller]
name=Netkiller Free Books
baseurl=http://netkiller.sourceforge.net/pub/repo/
enabled=1
gpgcheck=0
gpgkey=
EOF
```

查找包

```
# yum search netkiller

netkiller-centos.x86_64 : Netkiller centos Cookbook
netkiller-cryptography.x86_64 : Netkiller cryptography
Cookbook
netkiller-docbook.x86_64 : Netkiller docbook Cookbook
netkiller-linux.x86_64 : Netkiller linux Cookbook
netkiller-mysql.x86_64 : Netkiller mysql Cookbook
netkiller-php.x86_64 : Netkiller php Cookbook
netkiller-postgresql.x86_64 : Netkiller postgresql Cookbook
netkiller-python.x86_64 : Netkiller python Cookbook
netkiller-version.x86_64 : Netkiller version Cookbook
```

安装包

```
yum install netkiller-docbook
```

4. 打赏（Donations）

If you like this documents, please make a donation to support the authors' efforts. Thank you!

您可以通过微信，支付宝，贝宝给作者打赏。

银行(Bank)

招商银行(China Merchants Bank)

开户名：陈景峰

账号：9555500000007459

微信（Wechat）

支付宝（Alipay）

PayPal Donations

https://www.paypal.me/netkiller

5. 联系方式

主站 http://www.netkiller.cn/

备用 http://netkiller.github.io/

繁体网站 http://netkiller.sourceforge.net/

联系作者

Mobile: +86 13113668890

Email: netkiller@msn.com

QQ群: 128659835 请注明“读者”

QQ: 13721218

ICQ: 101888222

注：请不要问我安装问题！

博客 Blogger

知乎专栏 https://zhuanlan.zhihu.com/netkiller

LinkedIn: http://cn.linkedin.com/in/netkiller

OSChina: http://my.oschina.net/neochen/

Facebook: https://www.facebook.com/bg7nyt

Flickr: http://www.flickr.com/photos/bg7nyt/

Disqus: http://disqus.com/netkiller/

solidot: http://solidot.org/~netkiller/

SegmentFault: https://segmentfault.com/u/netkiller

Reddit: https://www.reddit.com/user/netkiller/

Digg: http://www.digg.com/netkiller

Twitter: http://twitter.com/bg7nyt

weibo: http://weibo.com/bg7nyt

Xbox club

我的 xbox 上的ID是 netkiller xbox，我创建了一个俱乐部 netkiller 欢迎加入。

Radio

CQ CQ CQ DE BG7NYT:

如果这篇文章对你有所帮助,请寄给我一张QSL卡片, qrz.cn or qrz.com or hamcall.net

Personal Amateur Radiostations of P.R.China

ZONE CQ24 ITU44 ShenZhen, China

Best Regards, VY 73! OP. BG7NYT

守听频率 DMR 438.460 -8 Color 12 Slot 2 Group 46001

守听频率 C4FM 439.360 -5 DN/VW

MMDVM Hotspot:

Callsign: BG7NYT QTH: Shenzhen, China

YSF: YSF80337 - CN China 1 - W24166/TG46001

DMR: BM_China_46001 - DMR Radio ID: 4600441

第 1 章 Nginx

1. Installing

1.1. Netkiller OSCM 一键安装 （CentOS 7）

```
# curl -s
https://raw.githubusercontent.com/oscm/shell/master/web/nginx/s
table/nginx.sh | bash
```

1.2. Installing by apt-get under the debain/ubuntu

```
$ sudo apt-get install nginx
```

```
sudo /etc/init.d/nginx start
```

1.3. CentOS

http://nginx.org/packages/centos/$releasever/$basearch/

$releasever 是版本号

$basearch 处理器架构

http://nginx.org/packages/centos/6/x86_64/

```
cat > /etc/yum.repos.d/nginx.repo <<EOF
[nginx]
name=nginx repo
baseurl=http://nginx.org/packages/centos/6/x86_64/
gpgcheck=0
enabled=1
EOF
```

i386

```
cat > /etc/yum.repos.d/nginx.repo <<EOF
[nginx]
name=nginx repo
baseurl=http://nginx.org/packages/centos/5/i386/
gpgcheck=0
enabled=1
EOF
```

```
yum search nginx
============================================= Matched: nginx
=============================================
nginx.x86_64 : high performance web server

yum install -y nginx
chkconfig nginx on
service nginx start
```

spawn-fcgi script

```
yum -y install spawn-fcgi
```

/etc/sysconfig/spawn-fcgi

移除SOCKET与OPTIONS注释, apache改为nginx

```
# cat /etc/sysconfig/spawn-fcgi
# You must set some working options before the "spawn-fcgi"
service will work.
# If SOCKET points to a file, then this file is cleaned up by
the init script.
#
# See spawn-fcgi(1) for all possible options.
#
# Example :
SOCKET=/var/run/php-fcgi.sock
OPTIONS="-u apache -g apache -s $SOCKET -S -M 0600 -C 32 -F 1 -
P /var/run/spawn-fcgi.pid -- /usr/bin/php-cgi"
```

```
chkconfig spawn-fcgi on
```

starting spawn-fcgi

```
/etc/init.d/spawn-fcgi start
```

check port

```
# netstat -nl
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address               Foreign Address
State
tcp        0      0 0.0.0.0:22                  0.0.0.0:*
LISTEN
tcp        0      0 :::22                       :::*
LISTEN
Active UNIX domain sockets (only servers)
```

```
Proto RefCnt Flags       Type       State         I-Node Path
unix  2      [ ACC ]     STREAM     LISTENING     25282
/var/run/php-fcgi.sock
unix  2      [ ACC ]     STREAM     LISTENING     8227
@/com/ubuntu/upstart
```

```
                                <para>Unix domain socket</para>
                                <![CDATA[

        location ~ \.php$ {
            fastcgi_pass   unix:/var/run/php-fcgi.sock;
            fastcgi_index  index.php;
            fastcgi_param  SCRIPT_FILENAME  /var/www/nginx-
default$fastcgi_script_name;
            include        fastcgi_params;
        }
```

TCP/IP

```
/usr/bin/spawn-fcgi -a 127.0.0.1 -p 9000 -u nginx -g nginx -d
/www -C 32 -F 1 -P /var/run/spawn-fcgi.pid -f /usr/bin/php-cgi
```

```
        location ~ \.php$ {
            fastcgi_pass   127.0.0.1:9000;
            fastcgi_index  index.php;
            fastcgi_param  SCRIPT_FILENAME  /var/www/nginx-
default$fastcgi_script_name;
            include        fastcgi_params;
        }
```

```
# netstat -tulpn | grep :9000
tcp        0      0 127.0.0.1:9000              0.0.0.0:*
```

```
LISTEN      26877/php-cgi
```

```
chkconfig nginx on
```

check config

```
nginx -t
```

php-fpm

```
rpm -Uvh
http://download.fedora.redhat.com/pub/epel/6/x86_64/epel-
release-6-5.noarch.rpm
yum install nginx -y
```

```
chkconfig nginx on
```

check config

```
nginx -t
```

```
yum -y install mysql mysql-server
yum -y install php php-cgi php-mysql php-mbstring php-gd php-
fastcgi
yum -y install perl-DBI perl-DBD-MySQL
```

其他 php-fpm YUM源

```
rpm --import http://rpms.famillecollet.com/RPM-GPG-KEY-remi
rpm -ivh http://rpms.famillecollet.com/enterprise/remi-release-
6.rpm
```

```
# rpm -Uvh
http://centos.alt.ru/repository/centos/6/i386/centalt-release-
6-1.noarch.rpm
# yum update
```

fastcgi backend

```
upstream backend  {
  server   localhost:1234;
}

fastcgi_pass   backend;
```

1.4. installing by source

```
cd /usr/local/src/
wget http://www.nginx.org/download/nginx-1.0.6.tar.gz

./configure --prefix=/usr/local/server/nginx \
--with-openssl=/usr/include \
--with-pcre=/usr/include/pcre/ \
--with-http_stub_status_module \
--without-http_memcached_module \
--without-http_fastcgi_module \
--without-http_rewrite_module \
--without-http_map_module \
--without-http_geo_module \
--without-http_autoindex_module
```

rpm 所使用的编译参数

```
nginx -V
nginx: nginx version: nginx/1.0.6
nginx: built by gcc 4.4.4 20100726 (Red Hat 4.4.4-13) (GCC)
nginx: TLS SNI support enabled
nginx: configure arguments: --prefix=/etc/nginx/ --sbin-
path=/usr/sbin/nginx --conf-path=/etc/nginx/nginx.conf --error-
log-path=/var/log/nginx/error.log --http-log-
path=/var/log/nginx/access.log --pid-path=/var/run/nginx.pid --
lock-path=/var/run/nginx.lock --http-client-body-temp-
path=/var/cache/nginx/client_temp --http-proxy-temp-
path=/var/cache/nginx/proxy_temp --http-fastcgi-temp-
path=/var/cache/nginx/fastcgi_temp --http-uwsgi-temp-
path=/var/cache/nginx/uwcgi_temp --http-scgi-temp-
path=/var/cache/nginx/scgi_temp --user=nginx --group=nginx --
with-http_ssl_module --with-http_realip_module --with-
http_addition_module --with-http_sub_module --with-
http_dav_module --with-http_flv_module --with-
http_gzip_static_module --with-http_random_index_module --with-
http_secure_link_module --with-http_stub_status_module --with-
mail --with-mail_ssl_module --with-file-aio --with-ipv6
```

```
# nginx -V
nginx version: nginx/1.2.3
built by gcc 4.4.4 20100726 (Red Hat 4.4.4-13) (GCC)
TLS SNI support enabled
configure arguments: --prefix=/etc/nginx/ --sbin-
path=/usr/sbin/nginx --conf-path=/etc/nginx/nginx.conf --error-
log-path=/var/log/nginx/error.log --http-log-
path=/var/log/nginx/access.log --pid-path=/var/run/nginx.pid --
lock-path=/var/run/nginx.lock --http-client-body-temp-
path=/var/cache/nginx/client_temp --http-proxy-temp-
path=/var/cache/nginx/proxy_temp --http-fastcgi-temp-
path=/var/cache/nginx/fastcgi_temp --http-uwsgi-temp-
path=/var/cache/nginx/uwsgi_temp --http-scgi-temp-
path=/var/cache/nginx/scgi_temp --user=nginx --group=nginx --
with-http_ssl_module --with-http_realip_module --with-
```

```
http_addition_module --with-http_sub_module --with-
http_dav_module --with-http_flv_module --with-http_mp4_module -
-with-http_gzip_static_module --with-http_random_index_module -
-with-http_secure_link_module --with-http_stub_status_module --
with-mail --with-mail_ssl_module --with-file-aio --with-ipv6 --
with-cc-opt='-O2 -g'
```

1.5. CentOS 7

```
#!/bin/bash
rpm -ivh http://nginx.org/packages/centos/7/noarch/RPMS/nginx-
release-centos-7-0.el7.ngx.noarch.rpm
yum install -y nginx

cp /etc/nginx/nginx.conf{,.original}

vim /etc/nginx/nginx.conf <<VIM > /dev/null 2>&1
:%s/worker_processes  1;/worker_processes  8;/
:%s/worker_connections  1024;/worker_connections  4096;/
:%s/#gzip/server_tokens off;\r    gzip/
:%s/#gzip/gzip/
:wq
VIM

sed -i '4iworker_rlimit_nofile 65530;' /etc/nginx/nginx.conf

systemctl enable nginx
systemctl start nginx
```

测试配置文件是否正确

```
# nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is
ok
nginx: configuration file /etc/nginx/nginx.conf test is
successful
```

1.6. Mac

安装

```
neo@MacBook-Pro ~ % brew install nginx
```

启动

```
neo@MacBook-Pro ~ % brew services start nginx
==> Successfully started `nginx` (label: homebrew.mxcl.nginx)
```

重启

```
neo@MacBook-Pro /usr/local/etc/nginx % brew services restart
nginx
Stopping `nginx`... (might take a while)
==> Successfully stopped `nginx` (label: homebrew.mxcl.nginx)
==> Successfully started `nginx` (label: homebrew.mxcl.nginx)
```

配置文件在 /usr/local/etc/nginx 下，默认使用 8080端口

nginx.conf 文件如下

```


#user  nobody;
worker_processes  1;

#error_log  logs/error.log;
#error_log  logs/error.log  notice;
#error_log  logs/error.log  info;

#pid        logs/nginx.pid;


events {
    worker_connections  1024;
}


http {
    include       mime.types;
    default_type  application/octet-stream;

    #log_format  main  '$remote_addr - $remote_user
[$time_local] "$request" '
    #                  '$status $body_bytes_sent
"$http_referer" '
    #                  '"$http_user_agent"
"$http_x_forwarded_for"';

    #access_log  logs/access.log  main;

    sendfile        on;
    #tcp_nopush     on;

    #keepalive_timeout  0;
    keepalive_timeout  65;

    #gzip  on;

    server {
        listen       8080;
        server_name  localhost;

        #charset koi8-r;

        #access_log  logs/host.access.log  main;

        location / {
            root   html;
            index  index.html index.htm;
        }

        #error_page  404              /404.html;

        # redirect server error pages to the static page
/50x.html
        #
        error_page   500 502 503 504  /50x.html;
        location = /50x.html {
            root   html;
        }

        # proxy the PHP scripts to Apache listening on
127.0.0.1:80
        #
        #location ~ \.php$ {
        #    proxy_pass   http://127.0.0.1;
        #}

        # pass the PHP scripts to FastCGI server listening on
127.0.0.1:9000
        #
        #location ~ \.php$ {
        #    root           html;
        #    fastcgi_pass   127.0.0.1:9000;
        #    fastcgi_index  index.php;
        #    fastcgi_param  SCRIPT_FILENAME
/scripts$fastcgi_script_name;
        #    include        fastcgi_params;
        #}

        # deny access to .htaccess files, if Apache's document
root
        # concurs with nginx's one
        #
        #location ~ /\.ht {
        #    deny  all;
        #}
    }


    # another virtual host using mix of IP-, name-, and port-
based configuration
    #
    #server {
    #    listen       8000;
    #    listen       somename:8080;
    #    server_name  somename  alias  another.alias;

    #    location / {
    #        root   html;
    #        index  index.html index.htm;
    #    }
    #}
    # HTTPS server
    #
    #server {
    #    listen       443 ssl;
    #    server_name  localhost;

    #    ssl_certificate      cert.pem;
    #    ssl_certificate_key  cert.key;

    #    ssl_session_cache    shared:SSL:1m;
    #    ssl_session_timeout  5m;

    #    ssl_ciphers  HIGH:!aNULL:!MD5;
    #    ssl_prefer_server_ciphers  on;

    #    location / {
    #        root   html;
    #        index  index.html index.htm;
    #    }
    #}
    include servers/*;
}
```

php-fpm

mac下自带的软件

```
neo@MacBook-Pro ~ % php -v
PHP 5.6.30 (cli) (built: Feb  7 2017 16:18:37)
Copyright (c) 1997-2016 The PHP Group
Zend Engine v2.6.0, Copyright (c) 1998-2016 Zend Technologies
```

启动php-fpm方法如下

```
cd /private/etc
```

```
sudo cp php-fpm.conf.default php-fpm.conf
```

修改error_log项, 改为error_log = /usr/local/var/log/php-fpm.log

启动 php-fpm

```
php-fpm
```

1.7. rotate log

log shell

一些特别的情况下需要切割日志，请参考下面的例子

```
# cat /srv/bin/rotatelog.sh

#!/bin/bash
# run this script at 0:00

#Nginx Log Path
log_dir="/var/log/nginx"
date_dir=`date +%Y/%m/%d/%H`

mkdir -p ${log_dir}/${date_dir} > /dev/null 2>&1
mv ${log_dir}/access.log ${log_dir}/${date_dir}/access.log
mv ${log_dir}/error.log ${log_dir}/${date_dir}/error.log

kill -USR1 `cat /var/run/nginx.pid`

gzip ${log_dir}/${date_dir}/access.log &
gzip ${log_dir}/${date_dir}/error.log &
```

/etc/logrotate.d/nginx

如果是非源码安装，一般情况nginx都会自带日志切割处理配置文件。

```
# cat /etc/logrotate.d/nginx
/var/log/nginx/*.log {
        daily
        missingok
        rotate 52
        compress
        delaycompress
        notifempty
        create 640 root adm
        sharedscripts
        postrotate
                [ -f /var/run/nginx.pid ] && kill -USR1 `cat
/var/run/nginx.pid`
        endscript
}
```

2. Nginx 命令

```
root@netkiller ~ % nginx -h
nginx version: nginx/1.12.1
Usage: nginx [-?hvVtTq] [-s signal] [-c filename] [-p prefix]
[-g directives]

Options:
  -?,-h         : this help
  -v            : show version and exit
  -V            : show version and configure options then exit
  -t            : test configuration and exit
  -T            : test configuration, dump it and exit
  -q            : suppress non-error messages during
configuration testing
  -s signal     : send signal to a master process: stop, quit,
reopen, reload
  -p prefix     : set prefix path (default: /etc/nginx/)
  -c filename   : set configuration file (default:
/etc/nginx/nginx.conf)
  -g directives : set global directives out of configuration
file
```

2.1. -V show version and configure options then exit

```
[root@netkiller tmp]# nginx -v
nginx version: nginx/1.10.1

[root@netkiller tmp]# nginx -V
nginx version: nginx/1.10.1
built by gcc 4.8.5 20150623 (Red Hat 4.8.5-4) (GCC)
built with OpenSSL 1.0.1e-fips 11 Feb 2013
TLS SNI support enabled
configure arguments: --prefix=/etc/nginx --sbin-
path=/usr/sbin/nginx --modules-path=/usr/lib64/nginx/modules --
```

```
conf-path=/etc/nginx/nginx.conf --error-log-
path=/var/log/nginx/error.log --http-log-
path=/var/log/nginx/access.log --pid-path=/var/run/nginx.pid --
lock-path=/var/run/nginx.lock --http-client-body-temp-
path=/var/cache/nginx/client_temp --http-proxy-temp-
path=/var/cache/nginx/proxy_temp --http-fastcgi-temp-
path=/var/cache/nginx/fastcgi_temp --http-uwsgi-temp-
path=/var/cache/nginx/uwsgi_temp --http-scgi-temp-
path=/var/cache/nginx/scgi_temp --user=nginx --group=nginx --
with-http_ssl_module --with-http_realip_module --with-
http_addition_module --with-http_sub_module --with-
http_dav_module --with-http_flv_module --with-http_mp4_module -
-with-http_gunzip_module --with-http_gzip_static_module --with-
http_random_index_module --with-http_secure_link_module --with-
http_stub_status_module --with-http_auth_request_module --with-
http_xslt_module=dynamic --with-
http_image_filter_module=dynamic --with-
http_geoip_module=dynamic --with-http_perl_module=dynamic --
add-dynamic-module=njs-1c50334fbea6/nginx --with-threads --
with-stream --with-stream_ssl_module --with-http_slice_module -
-with-mail --with-mail_ssl_module --with-file-aio --with-ipv6 -
-with-http_v2_module --with-cc-opt='-O2 -g -pipe -Wall -Wp,-
D_FORTIFY_SOURCE=2 -fexceptions -fstack-protector-strong --
param=ssp-buffer-size=4 -grecord-gcc-switches -m64 -
mtune=generic'
```

2.2. -t : test configuration and exit

CentOS 6

```
$ sudo service nginx configtest
Testing nginx configuration: nginx.
```

通用方法

```
root@netkiller ~ % nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is
ok
```

```
nginx: configuration file /etc/nginx/nginx.conf test is
successful
```

2.3. test configuration, dump it and exit

```
root@netkiller ~ % nginx -T
nginx: the configuration file /etc/nginx/nginx.conf syntax is
ok
nginx: configuration file /etc/nginx/nginx.conf test is
successful
# configuration file /etc/nginx/nginx.conf:

user  nginx;
worker_processes  auto;
worker_rlimit_nofile 65530;

error_log  /var/log/nginx/error.log warn;
pid        /var/run/nginx.pid;


events {
    worker_connections  4096;
}


http {
    include       /etc/nginx/mime.types;
    default_type  application/octet-stream;

    log_format  main  '$remote_addr - $remote_user
[$time_local] "$request" '
                      '$status $body_bytes_sent "$http_referer"
'
                      '"$http_user_agent"
"$http_x_forwarded_for"';

    access_log  /var/log/nginx/access.log  main;

    sendfile        on;
    #tcp_nopush     on;

    keepalive_timeout  65;
    server_tokens off;
    gzip  on;
    gzip_types text/plain text/css application/json
application/x-javascript application/xml;

    include /etc/nginx/conf.d/*.conf;
}

# configuration file /etc/nginx/mime.types:

types {
    text/html                                      html htm shtml;
    text/css                                       css;
    text/xml                                       xml;
    image/gif                                      gif;
    image/jpeg                                     jpeg jpg;
    application/javascript                         js;
    application/atom+xml                           atom;
    application/rss+xml                            rss;

    text/mathml                                    mml;
    text/plain                                     txt;
    text/vnd.sun.j2me.app-descriptor               jad;
    text/vnd.wap.wml                               wml;
    text/x-component                               htc;

    image/png                                      png;
    image/tiff                                     tif tiff;
    image/vnd.wap.wbmp                             wbmp;
    image/x-icon                                   ico;
    image/x-jng                                    jng;
    image/x-ms-bmp                                 bmp;
    image/svg+xml                                  svg svgz;
    image/webp                                     webp;

    application/font-woff                          woff;
    application/java-archive                       jar war ear;
    application/json                               json;
    application/mac-binhex40                       hqx;
    application/msword                             doc;
    application/pdf                                pdf;
    application/postscript                         ps eps ai;
    application/rtf                                rtf;
    application/vnd.apple.mpegurl                  m3u8;
    application/vnd.ms-excel              xls;
    application/vnd.ms-fontobject         eot;
    application/vnd.ms-powerpoint         ppt;
    application/vnd.wap.wmlc              wmlc;
    application/vnd.google-earth.kml+xml  kml;
    application/vnd.google-earth.kmz      kmz;
    application/x-7z-compressed           7z;
    application/x-cocoa                   cco;
    application/x-java-archive-diff       jardiff;
    application/x-java-jnlp-file          jnlp;
    application/x-makeself                run;
    application/x-perl                    pl pm;
    application/x-pilot                   prc pdb;
    application/x-rar-compressed          rar;
    application/x-redhat-package-manager  rpm;
    application/x-sea                     sea;
    application/x-shockwave-flash         swf;
    application/x-stuffit                 sit;
    application/x-tcl                     tcl tk;
    application/x-x509-ca-cert            der pem crt;
    application/x-xpinstall               xpi;
    application/xhtml+xml                 xhtml;
    application/xspf+xml                  xspf;
    application/zip                       zip;

    application/octet-stream              bin exe dll;
    application/octet-stream              deb;
    application/octet-stream              dmg;
    application/octet-stream              iso img;
    application/octet-stream              msi msp msm;

    application/vnd.openxmlformats-
officedocument.wordprocessingml.document    docx;
    application/vnd.openxmlformats-
officedocument.spreadsheetml.sheet          xlsx;
    application/vnd.openxmlformats-
officedocument.presentationml.presentation  pptx;

    audio/midi                            mid midi kar;
    audio/mpeg                            mp3;
    audio/ogg                             ogg;
    audio/x-m4a                           m4a;
    audio/x-realaudio                     ra;

    video/3gpp                            3gpp 3gp;
    video/mp2t                                    ts;
    video/mp4                                     mp4;
    video/mpeg                                    mpeg mpg;
    video/quicktime                               mov;
    video/webm                                    webm;
    video/x-flv                                   flv;
    video/x-m4v                                   m4v;
    video/x-mng                                   mng;
    video/x-ms-asf                                asx asf;
    video/x-ms-wmv                                wmv;
    video/x-msvideo                               avi;
}

# configuration file /etc/nginx/conf.d/default.conf:
server {
    listen       80;
    server_name  localhost;

    #charset koi8-r;
    #access_log  /var/log/nginx/host.access.log  main;

    location / {
        root   /usr/share/nginx/html;
        index  index.html index.htm;
    }

    #error_page  404              /404.html;

    # redirect server error pages to the static page /50x.html
    #
    error_page   500 502 503 504  /50x.html;
    location = /50x.html {
        root   /usr/share/nginx/html;
    }

    # proxy the PHP scripts to Apache listening on 127.0.0.1:80
    #
    #location ~ \.php$ {
    #    proxy_pass   http://127.0.0.1;
    #}

    # pass the PHP scripts to FastCGI server listening on
127.0.0.1:9000
    #
    #location ~ \.php$ {
```

```
    #    root           html;
    #    fastcgi_pass   127.0.0.1:9000;
    #    fastcgi_index  index.php;
    #    fastcgi_param  SCRIPT_FILENAME
/scripts$fastcgi_script_name;
    #    include        fastcgi_params;
    #}

    # deny access to .htaccess files, if Apache's document root
    # concurs with nginx's one
    #
    #location ~ /\.ht {
    #    deny  all;
    #}
}
```

3. nginx.conf 配置文件

3.1. 处理器配置

worker_processes = CPU 数量

```
user www;
worker_processes 1;

error_log /var/log/nginx/error.log warn;
pid /var/run/nginx.pid;
```

3.2. events 配置

连接数配置

```
events {
    worker_connections  4096;
}
```

3.3. http 配置

缓冲区相关设置

自定义缓冲区相关设置

```
client_body_buffer_size 1K;
client_header_buffer_size 1k;
```

```
client_max_body_size 1k;
large_client_header_buffers 2 1k;
```

上传文件提示 client intended to send too large body，配置下面参数可以解决。

```
server {
  ...
  client_max_body_size 200M;
}
```

超时设置

超时相关设置

```
        client_body_timeout 10;
        client_header_timeout 10;
        keepalive_timeout 65;
        send_timeout 10;
```

gzip

```
        gzip on;
        gzip_min_length 1000;
        gzip_buffers 4 8k;
        gzip_types text/plain text/css application/json
application/x-javascript application/xml;
```

```
        gzip on;
        gzip_http_version 1.0;
        gzip_disable "MSIE [1-6].";
        gzip_types text/plain application/x-javascript text/css
text/javascript;
```

gzip_types 压缩类型

```
        gzip_types text/plain text/css application/javascript
text/javascript application/x-javascript text/xml
application/xml application/xml+rss application/json;
```

text/html 是 gzip_types 默认值，所以不要将text/html加入到 gzip_types

测试，验证 gzip 正常工作

```
neo@netkiller:~/workspace$ curl -s -I -H 'Accept-Encoding:
gzip,deflate' http://img.netkiller.cn/js/react.js | grep gzip
Content-Encoding: gzip
```

如果提示 Content-Encoding: gzip 便是配置正确

不仅仅只能压缩html,js,css还能压缩json

```
neo@netkiller:~$ curl -s -I -H 'Accept-Encoding: gzip,deflate'
http://inf.netkiller.cn/list/json/2.json
HTTP/1.1 200 OK
Server: nginx
Date: Thu, 15 Dec 2016 03:36:31 GMT
```

```
Content-Type: application/json; charset=utf-8
Connection: keep-alive
Cache-Control: max-age=60
Access-Control-Allow-Origin: *
Access-Control-Allow-Headers: Content-Type,Origin
Access-Control-Allow-Methods: GET,OPTIONS
Content-Encoding: gzip
```

CDN支持

配置 gzip_proxied any; 后CDN才能识别 gzip

```
server_tokens off;
gzip on;
gzip_types text/plain text/css application/javascript
text/javascript application/x-javascript text/xml
application/xml application/xml+rss application/json;
gzip_proxied any;
```

使用包含配置文件配置 **gzip**

```
cat <<-EOF> /etc/nginx/conf.d/gzip.conf
gzip on;
gzip_vary on;
gzip_proxied any;
gzip_min_length 1000;
gzip_types text/plain text/css application/javascript
application/json application/xml application/octet-stream;
EOF

# text/html 类型无需配置，否则会提示
# nginx: [warn] duplicate MIME type "text/html" in
/etc/nginx/conf.d/default.conf
```

server_tokens

隐藏nginx版本号

```
http {
...
server_tokens off;
...
}
```

ssi

```
http {
        ssi on;
}

location / {
        ssi on;
        ssi_silent_errors on;
        ssi_types text/shtml;
}
```

```
        ssi on;
        ssi_silent_errors on;
        ssi_types text/shtml;
        ssi_value_length 256;

        server_names_hash_bucket_size 128;
        client_header_buffer_size 32k;
        large_client_header_buffers 4 32k;
        client_max_body_size 8m;
```

ssi_silent_errors 默认值是off，开启后在处理SSI文件出错时不输出错误提示:"[an error occurred while processing the directive] "

ssi_types 默认是ssi_types text/html，如果需要shtml支持，则需要设置：ssi_types text/shtml

ssi_value_length 默认值是 256，用于定义SSI参数的长度。

3.4. Nginx 变量

可用的全局变量

```
$args
$content_length
$content_type
$document_root
$document_uri
$host
$http_user_agent
$http_cookie
$http_referer
$limit_rate
$request_body_file
$request_method
$remote_addr
$remote_port
$remote_user
$request_filename
$request_uri
$query_string
$scheme
$server_protocol
$server_addr
$server_name
$server_port
$uri
```

$host

抽取域名中的域，例如www.netkiller.cn 返回netkiller.cn

```
if ($host ~* ^www\.(.*)) {
    set $domain $1;
    rewrite ^(.*) http://user.$domain permanent;
}
```

提取主机

```
if ($host ~* ^(.+)\.example\.com$) {
    set $subdomain $1;
    rewrite ^(.*) http://www.example.com/$subdomain permanent;
}
```

提取 domain 例如 www.netkiller.cn 提取后 netkiller.cn

只处理二级域名 example.com 不处理三级域名

```
        if ($host ~* ^([^\.]+)\.([^\.]+)$) {
           set $domain $1.$2;
        }
```

处理三级域名

```
        set $domain $host;
        if ($host ~* ^([^\.]+)\.([^\.]+)\.([^\.]+)$) {
            set $domain $2.$3;
        }
```

http_user_agent

```
## Block http user agent - wget ##
if ($http_user_agent ~* (Wget|Curl) ) {
   return 403;
}

## Block Software download user agents ##
if ($http_user_agent ~* LWP::Simple|BBBike|wget) {
       return 403;
}

if ($http_user_agent ~ (msnbot|scrapbot) ) {
    return 403;
}


if ($http_user_agent ~ (Spider|Robot) ) {
    return 403;
}

if ($http_user_agent ~ MSIE) {
    rewrite ^(.*)$ /msie/$1 break;
}
```

禁止非浏览器访问

禁止非浏览器访问

```
if ($http_user_agent ~ ^$) {
        return 412;
}
```

测试是否生效

```
tail -f /var/log/nginx/www.mydomain.com.access.log
```

```
telnet 192.168.2.10 80
GET /index.html HTTP/1.0
Host: www.mydomain.com
```

http_user_agent 没有设置不允许访问

```
        if ($http_user_agent = "") { return 403; }
```

验证测试，首先使用curl -A 指定一个 空的User Agent，应该返回403.

```
curl -A ""  http://www.example.com/xml/data.json

<html>
<head><title>403 Forbidden</title></head>
<body bgcolor="white">
<center><h1>403 Forbidden</h1></center>
<hr><center>nginx</center>
</body>
</html>
```

http_referer

```
if ($http_referer ~* "PHP/5.2.14"){return 403;}
```

valid_referers/invalid_referer

```
valid_referers none blocked *.example.com example.com;
if ($invalid_referer) {
        #rewrite ^(.*)$  http://www.example.com/cn/$1;
        return 403;
}
```

request_filename

```
    location / {
        root   /www/mydomain.com/info.mydomain.com;
        index  index.html;

                rewrite ^/$  http://www.mydomain.com/;

                valid_referers none blocked *.mydomain.com;
                if ($invalid_referer) {
                        return 403;
                }

        proxy_intercept_errors  on;
            proxy_set_header  X-Real-IP  $remote_addr;
        proxy_set_header  X-Forwarded-For
$proxy_add_x_forwarded_for;
        proxy_set_header  Host            $host;


        if (!-f $request_filename) {
          proxy_pass http://old.mydomain.com;
          break;
        }
    }
```

request_uri

```
server {
    listen       80;
    server_name  quote.mydomain.com;

    charset utf-8;
    access_log  /var/log/nginx/quote.mydomain.com.access.log
main;
```

```
    location / {
        root   /www/mydomain.com/info.mydomain.com;
        index  index.html ;

                rewrite ^/$  http://www.mydomain.com/;

                valid_referers none blocked *.mydomain.com;
                if ($invalid_referer) {
                        #rewrite ^(.*)$
http://www.mydomain.com/cn/$1;
                        return 403;
                }

        proxy_intercept_errors  on;
            proxy_set_header  X-Real-IP  $remote_addr;
        proxy_set_header  X-Forwarded-For
$proxy_add_x_forwarded_for;
        proxy_set_header  Host            $host;

                if ( $request_uri ~
"^/xml/(sge|cgse|futures|stock|bonds)\.xml$") {
              proxy_pass http://21.16.22.12/$request_uri;
                break;
        }

        if (!-f $request_filename) {
                  proxy_pass http://cms.mydomain.com;
                  break;
        }

    }

    location ~ \.xml$ {
        proxy_pass http://21.16.22.12/public/datas$request_uri;
        break;
    }

    location ~* ^/public/datas/\w+\.xml$ {
        proxy_pass http://21.16.22.12/$request_uri;
        break;
    }
}
```

```
#add for yiiframework
        if (!-e $request_filename){
                   rewrite (.*) /index.php break;
        }

        location ~ .*\.php?$
        {
                  #fastcgi_pass  unix:/tmp/php-cgi.sock;
                  include fcgi.conf;
                  fastcgi_pass  127.0.0.1:10080;
                  fastcgi_index index.php;

                  set $path_info $request_uri;

                  if ($request_uri ~ "^(.*)(\?.*)$") {
                        set $path_info $1;
                  }
                  fastcgi_param PATH_INFO $path_info;
        }
#end for yiiframework
```

remote_addr

```
location /name/(match) {
    if ($remote_addr !~ ^10.10.20) {
        limit_rate 10k;
    }

    proxy_buffering off;
    proxy_pass http://10.10.20.1/${1}.html;
}

if ($remote_addr ~* "192.168.0.50|192.168.0.51|192.168.0.56") {
        proxy_pass http://www.netkiller.cn/error;
}
```

```
location ~ /(\d+) {
    if ($remote_addr ~ (\d+)\.\d+\.) {
    }

    echo $1;
}
```

```
$ curl 127.0.0.1/134
127

$ curl 192.168.0.1/134
192
```

http_cookie

```
if ($http_cookie ~* "id=([^;]+)(?:;|$)") {
    set $id $1;
}
```

request_method

```
location ~* /restful {
        if ($request_method = PUT ) {
        return 403;
        }
        if ($request_method = DELETE ) {
        return 403;
        }
        if ($request_method = POST ) {
        return 403;
        }
        proxy_method GET;
        proxy_pass http://backend;
}
```

```
if ($request_method = POST) {
    return 405;
}
```

```
if ($request_method !~ ^(GET|HEAD|POST)$) {
        return 403;
}
```

limit_except

```
limit_except GET {
        allow 192.168.1.1;
        deny all;
}
```

invalid_referer

```
if ($invalid_referer) {
    return 403;
}
```

$request_body - HTTP POST 数据

用户日志

将 POST 数据记录到日志中

```
    log_format  main  '$remote_addr - $remote_user
[$time_local] "$request" '
                      '$status $body_bytes_sent "$http_referer"
'
                      '"$http_user_agent"
```

```
"$http_x_forwarded_for" - "$request_body"';
```

注意：用户登录通常使用POST方式，所以记录POST数据到日志会带来安全问题，例如用户密码泄露。

$request_body 用于缓存

因为nginx 使用 url 作为缓存的key (Nginx 将url地址 md5后作为缓存的 key)，所以默认情况下 Nginx 只能处理 HTTP GET 缓存。

对于 HTTP POST 请求，提交数据放在HTTP Head 头部提交到服务器的， 提交前后URL始终不变，Nginx 无法区分相同网址两次请求的内容有变化。

但是我们可以自定义 缓存 key 例如： "$request_uri|$request_body" 我们将请求地址加上post内容作为缓存的key，这样nginx 便可以区分每次提交后的页面变化。

```
 proxy_cache_path /tmp/cache levels=1:2
keys_zone=netkiller:128m inactive=1m;

 server {
  listen 8080;
  server_name localhost;

  location / {
   try_files $uri @backend;
  }

  location @backend {
   proxy_pass http://node1.netkiller.cn:8080;
   proxy_cache netkiller;
   proxy_cache_methods POST;
   proxy_cache_key "$request_uri|$request_body";
   proxy_buffers 8 32k;
   proxy_buffer_size 64k;
   proxy_cache_valid 5s;
   proxy_cache_use_stale updating;
   add_header X-Cached $upstream_cache_status;
  }
 }
```

自定义变量

```
if ( $host ~* (.*)\.(.*)\.(.*)) {
        set $subdomain $1;
}
location / {
    root  /www/$subdomain;
    index index.html index.php;
}
```

```
if ( $host ~* (\b(?!www\b)\w+)\.\w+\.\w+ ) {
    set $subdomain /$1;
}

location / {
    root /www/public_html$subdomain;
    index index.html index.php;
}
```

if 条件判断

判断相等

```
if ($query_string = "") {
        set $args "";
}
```

正则匹配

```
if ( $host ~* (.*)\.(.*)\.(.*)) {
        set $subdomain $1;
}
location / {
    root /var/www/$subdomain;
    index index.html index.php;
}
```

```
if ($remote_addr ~ "^(172.16|192.168)" && $http_user_agent ~*
"spider") {
    return 403;
}

set $flag 0;
if ($remote_addr ~ "^(172.16|192.168)") {
    set $flag "1";
}
if ($http_user_agent ~* "spider") {
    set $flag "1";
}
if ($flag = "1") {
    return 403;
}
```

```
if ($request_method = POST ) {
        return 405;
}
if ($args ~ post=140){
        rewrite ^ http://example.com/ permanent;
}
```

```
location /only-one-if {
    set $true 1;

    if ($true) {
        add_header X-First 1;
    }

    if ($true) {
        add_header X-Second 2;
    }

    return 204;
}
```

3.5. server

listen

绑定IP地址

```
                listen 80; 相当于0.0.0.0:80监听所有接口上的IP地址
                listen 192.168.0.1 80;
                listen 192.168.0.1:80;
```

配置默认主机 default_server

```
        server {
                listen 80;
                server_name acc.example.net;
                ...
        }

        server {
                listen 80 default_server;
                server_name www.example.org;
                ...
        }
```

单域名虚拟主机

```
# cat /etc/nginx/conf.d/images.conf
server {
        listen 80;
        server_name images.example.com;

        #charset koi8-r;
        access_log /var/log/nginx/images.access.log main;

        location / {
                root /www/images;
                index index.html index.htm;
        }

        #error_page 404 /404.html;

        # redirect server error pages to the static page
/50x.html
        #
        error_page 500 502 503 504 /50x.html;
                location = /50x.html {
                root /usr/share/nginx/html;
        }

        # proxy the PHP scripts to Apache listening on
127.0.0.1:80
        #
        #location ~ \.php$ {
        # proxy_pass http://127.0.0.1;
        #}

        # pass the
        PHP scripts to FastCGI server listening on
127.0.0.1:9000
        #
        #location ~ \.php$ {
        # root html;
        # fastcgi_pass 127.0.0.1:9000;
        # fastcgi_index index.php;
        # fastcgi_param SCRIPT_FILENAME
/scripts$fastcgi_script_name;
        # include fastcgi_params;
        #}

        # deny access to .htaccess files, if Apache's document
root
        # concurs with nginx's one
        #
        #location ~ /\.ht {
        # deny all;
        #}
}
```

绑定多个域名

```
        server_name images.example.com img1.example.com
img2.example.com;
```

使用通配符匹配

```
        server_name *.example.com
        server_name www.*;
```

正则匹配

```
        server_name ~^(.+)\.example\.com$;
        server_name ~^(www\.)?(.+)$;
```

ssl 虚拟主机

```
mkdir /etc/nginx/ssl
```

cp your_ssl_certificate to /etc/nginx/ssl

```
# HTTPS server
#
server {
        listen 443;
        server_name localhost;

        root html;
        index index.html index.htm;

        ssl on;
        #ssl_certificate cert.pem;
        ssl_certificate ssl/example.com.pem;
        ssl_certificate_key ssl/example.com.key;

        ssl_session_timeout 5m;

        ssl_protocols SSLv3 TLSv1;
        ssl_ciphers
ALL:!ADH:!EXPORT56:RC4+RSA:+HIGH:+MEDIUM:+LOW:+SSLv3:+EXP;
        ssl_prefer_server_ciphers on;

        location / {
                try_files $uri $uri/ /index.html;
        }
}
```

configtest

```
$ sudo service nginx configtest
Testing nginx configuration: nginx.
```

443 port test

```
$ openssl s_client -connect www.example.com:443
```

HTTP2 配置 SSL证书

自颁发证书

创建自颁发证书，SSL有两种证书模式，单向认证和双向认证，下面是单向认证模式。

```
mkdir -p /etc/pki/nginx/private/
cd /etc/pki/nginx/
openssl req -x509 -nodes -days 365 -newkey rsa:4096 -keyout
/etc/pki/nginx/private/server.key -out
/etc/pki/nginx/server.crt
```

建议使用域名命名证书

```
openssl req -x509 -nodes -days 365 -newkey rsa:4096 -keyout
/etc/nginx/ssl/api.netkiller.cn.key -out
/etc/nginx/ssl/api.netkiller.cn.crt

Generating a 4096 bit RSA private key
..........++
..............................................++
writing new private key to
'/etc/nginx/ssl/api.netkiller.cn.key'
-----
You are about to be asked to enter information that will be
incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished
Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XX]:CN
State or Province Name (full name) []:Guangdong
Locality Name (eg, city) [Default City]:Shenzhen
Organization Name (eg, company) [Default Company Ltd]:CF
Organizational Unit Name (eg, section) []:CF
Common Name (eg, your name or your server's hostname)
[]:api.netkiller.cn
Email Address []:netkiller@msn.com
```

注意: Common Name (eg, your name or your server's hostname) []:api.netkiller.cn 要跟你的 nginx server_name api.netkiller.cn 一样。

spdy

Nginx 配置 spdy

```
upstream api.netkiller.cn {
        #server api1.netkiller.cn:7000;
        #server api2.netkiller.cn backup;
}

server {
        listen 443 ssl spdy;
        server_name api.netkiller.cn;

        ssl_certificate /etc/nginx/ssl/api.netkiller.cn.crt;
        ssl_certificate_key
/etc/nginx/ssl/api.netkiller.cn.key;
        ssl_session_cache shared:SSL:20m;
        ssl_session_timeout 60m;
        ssl_protocols TLSv1 TLSv1.1 TLSv1.2;

        charset utf-8;
        access_log /var/log/nginx/api.netkiller.cn.access.log;
        error_log /var/log/nginx/api.netkiller.cn.error.log;

        location / {
                proxy_pass
                http://api.netkiller.cn;
                proxy_http_version 1.1;
                proxy_set_header Host $host;
                proxy_set_header X-Real-IP $remote_addr;
                proxy_set_header X-Forwarded-For
$proxy_add_x_forwarded_for;
                proxy_ignore_client_abort on;
        }

        #location / {
        # proxy_pass http://127.0.0.1:7000;
        #}

}
```

spdy 是google提出的标准，现在已经归入 http2 标准，Nginx 1.10 之后建议使用 http2 替代 spdy.

HTTP2

```
        server {
                listen 443 ssl http2;

                ssl_certificate server.crt;
                ssl_certificate_key server.key;
        }
```

用户访问 HTTP时强制跳转到 HTTPS

497 - normal request was sent to HTTPS

```
                                #让http请求重定向到https请求

        server {
                listen 80;
                error_page 497 https://$host$uri?$args;
                rewrite ^(.*)$ https://$host$1 permanent;
        }
```

```
        server {
                listen 80
                listen 443 ssl http2;

                ssl_certificate server.crt;
                ssl_certificate_key server.key;

                error_page 497 https://$host$uri?$args;

                if ($scheme = http) {
                        return 301
https://$server_name$request_uri;
```

```
                    }
        }
```

SSL 双向认证

生成证书

CA

```
touch /etc/pki/CA/index.txt
echo 00 > /etc/pki/CA/serial

制作 CA 私钥
openssl genrsa -out ca.key 2048

制作 CA 根证书（公钥）
openssl req -new -x509 -days 3650 -key ca.key -out ca.crt
```

服务器端

服务器端证书

```
制作服务端私钥
openssl genrsa -out server.pem 2048
openssl rsa -in server.pem -out server.key

生成签发请求
openssl req -new -key server.pem -out server.csr

用 CA 签发
openssl x509 -req -sha256 -in server.csr -CA ca.crt -CAkey
ca.key -CAcreateserial -days 3650 -out server.crt
```

客户端

生成客户端证书

```
openssl genrsa -des3 -out client.key 2048
openssl req -new -key client.key -out client.csr

生成签发请求
openssl req -new -key server.pem -out server.csr

用 CA 签发
openssl ca -in client.csr -cert ca.crt -keyfile ca.key -out
client.crt -days 3650
```

浏览器证书

生成浏览器证书

```
                                                openssl pkcs12
-export -inkey client.key -in client.crt -out client.pfx
```

SOAP 证书

```
                                                cat client.crt
client.key > soap.pem
```

```
        $header = array(
                'local_cert' => "soap.pem", //client.pem文件路径
                'passphrase' => "passw0rd" //client证书密码
                );
        $client = new SoapClient(FILE_WSDL, $header);
```

```
```

过程演示

例 1.1. Nginx SSL 双向认证，证书生成过程

```
root@VM_7_221_centos /etc/nginx/ssl % openssl genrsa -out
ca.key 2048
Generating RSA private key, 2048 bit long modulus
......................................................+++
.....................................+++
e is 65537 (0x10001)

root@VM_7_221_centos /etc/nginx/ssl % openssl req -new -x509 -
days 3650 -key ca.key -out ca.crt
You are about to be asked to enter information that will be
incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished
Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XX]:CN
State or Province Name (full name) []:GD
Locality Name (eg, city) [Default City]:Shenzhen
Organization Name (eg, company) [Default Company Ltd]:GW
Organizational Unit Name (eg, section) []:DEV
Common Name (eg, your name or your server's hostname)
[]:api.netkiller.cn
Email Address []:netkiller@msn.com
```

```
root@VM_7_221_centos /etc/nginx/ssl % openssl genrsa -out
server.pem 2048
Generating RSA private key, 2048 bit long modulus
.............+++
.......................................................+++
e is 65537 (0x10001)

root@VM_7_221_centos /etc/nginx/ssl % openssl rsa -in
server.pem -out server.key
writing RSA key

root@VM_7_221_centos /etc/nginx/ssl % openssl req -new -key
server.pem -out server.csr
You are about to be asked to enter information that will be
incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished
Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XX]:CN
State or Province Name (full name) []:GD
Locality Name (eg, city) [Default City]:Shenzhen
Organization Name (eg, company) [Default Company Ltd]:GW
Organizational Unit Name (eg, section) []:DEV
Common Name (eg, your name or your server's hostname)
[]:api.netkiller.cn
Email Address []:netkiller@msn.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:

root@VM_7_221_centos /etc/nginx/ssl % openssl x509 -req -sha256
-in server.csr -CA ca.crt -CAkey ca.key -CAcreateserial -days
3650 -out server.crt
Signature ok
subject=/C=CN/ST=GD/L=Shenzhen/O=GW/OU=DEV/CN=api.netkiller.cn/
emailAddress=netkiller@msn.com
Getting CA Private Key
```

Nginx 配置

```
mkdir /etc/nginx/ssl
cp server.crt server.key ca.crt /etc/nginx/ssl
cd /etc/nginx/ssl
```

/etc/nginx/conf.d/api.netkiller.cn.conf

```
server {
    listen       443 ssl;
    server_name  api.netkiller.cn;

    access_log off;

    ssl on;
    ssl_certificate /etc/nginx/ssl/server.crt;
    ssl_certificate_key /etc/nginx/ssl/server.key;
    ssl_client_certificate /etc/nginx/ssl/ca.crt;
    ssl_verify_client on;

    location / {
        proxy_pass http://localhost:8443;
    }
}
```

重启 nginx 服务器

```
                                        root@VM_7_221_centos
/etc/nginx % systemctl restart nginx
```

测试双向认证

首先直接请求

```
root@VM_7_221_centos /etc/nginx % curl -k
https://api.netkiller.cn/
<html>
<head><title>400 No required SSL certificate was sent</title>
</head>
<body bgcolor="white">
<center><h1>400 Bad Request</h1></center>
<center>No required SSL certificate was sent</center>
<hr><center>nginx</center>
</body>
</html>
```

使用证书请求

```
curl --insecure --key client.key --cert ./client.crt:123456
https://api.netkiller.cn
```

注意： --cert 参数需要写入路径和密码

server_name 配置

匹配所有域名

```
        server_name _;
```

泛解析主机

```
server {
    listen       80;
    server_name  example.org  www.example.org;
    ...
}

server {
    listen       80;
    server_name  *.example.org;
    ...
}

server {
    listen       80;
    server_name  mail.*;
    ...
}

server {
    listen       80;
    server_name  ~^(?<user>.+)\.example\.net$;
    ...
}
```

location

```
                                location / {
                                root /www;
                                index index.html index.htm;
                                }
```

禁止访问特定目录

location 匹配到特定的 path 将拒绝用户访问。

```
    location ~ /\.ht {
        deny  all;
    }

        location ~ ^/(config|include)/ {
                deny all;
                break;
        }

```

引用document_root之外的资源

引用document_root之外的资源需要 root 绝对路径指向目标文件夹

```

        location / {
                root /www/example.com/m.example.com;
                try_files $uri $uri/ @proxy;
        }
        location ^~ /module/ {
                root /www/example.com/www.example.com;
        }

        # 下面的写法是错误的，通过error_log 我们可以看到被定为
到/www/example.com/m.example.com/module
        location /module/ {
                root /www/example.com/www.example.com;
        }

```

处理扩展名

```

    location ~ \.php$ {
        root           /opt/netkiller.cn/cms.netkiller.cn;
        fastcgi_pass   127.0.0.1:9000;
        fastcgi_index  index.php;
        fastcgi_param  SCRIPT_FILENAME
/opt/netkiller.cn/cms.netkiller.cn$fastcgi_script_name;
        include        fastcgi_params;
    }
```

```
```

location 中关闭日志

```
	location = /favicon.ico {
		log_not_found off;
		access_log off;
	}

	location = /robots.txt {
		allow all;
		log_not_found off;
		access_log off;
	}
```

匹配多个目录

```
    location ~ /(dev|stage|prod) {
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header REMOTE-HOST $remote_addr;
        proxy_set_header X-Forwarded-For
$proxy_add_x_forwarded_for;
        proxy_pass http://api.netkiller.cn:8080;
    }
```

root 通过$host智能匹配目录

为每个host创建一个目录太麻烦，

```
	server {
```

```
                listen 80;
                server_name www.netkiller.cn news.netkiller.cn
bbs.netkiller.cn;

                charset utf-8;
                access_log /var/log/nginx/test.access.log main;

                location / {
                        root /www/netkiller.cn/$host;
                        index index.html index.htm;
                }
        }
```

处理主机名中的域

```
        server {
                listen 80;
                server_name *.example.com example.com;
                if ($host = 'example.com' ) {
                        rewrite ^/(.*)$
http://www.example.com/$1 permanent;
                }

                if ( $host ~* (.*)\.(.*)\.(.*)) {
                        set $subdomain $1;
                        set $domain $2.$3;
                }

                root /www/$domain/$subdomain;
                index index.html index.php;

                location ~ .*\.(php|shtml)?$ {
                        fastcgi_pass 127.0.0.1:9000;
                        fastcgi_index index.php;
                        include fcgi.conf;
                }
        }
```

或者采用这种格式 /www/example.com/www.example.com

```
                                root /www/$domain/$host;
```

更简洁的方法，只需在 /www/下面创建 域名目录即可例如/www/www.example.com

```
server {
        listen 80;
        server_name *.example.com example.com;
        if ($host = 'example.com' ) {
                rewrite ^/(.*)$ http://www.example.com/$1
permanent;
        }

        root /www/$host;
        index index.html index.php;

        location ~ .*\.(php|shtml)?$ {
                fastcgi_pass 127.0.0.1:9000;
                fastcgi_index index.php;
                include fcgi.conf;
        }
}
```

expires

expires 格式

例 1.2. Expires Examples

```
                expires 1 January, 1970, 00:00:01 GMT;
                expires 60s;
                expires 30m;
```

```
        expires 24h;
        expires 1d;
        expires max;
        expires off;

        expires 24h;
        expires modified +24h;
        expires @15h30m;
        expires 0;
        expires -1;
        expires epoch;
        add_header Cache-Control private;

```

注意：expires仅仅适用于200, 204, 301, 302,304

单个文件匹配

```
location ~* \.css$ {
        expires 30d;
}
```

扩展名匹配

```
#图片类资源缓存5天，并且不记录请求日志
location ~ .*\.(ico|gif|jpg|jpeg|png|bmp|swf)$
{
        expires 5d;
        access_log off;
}

#css/js 缓存一天，不记录请求日志
location ~ .*\.(js|css)$
{
        access_log off;
        expires 1d;
```

```
                add_header Pragma public;
                add_header Cache-Control "public";
        }
```

```
        location ~ .*\.
(htm|html|gif|jpg|jpeg|png|bmp|swf|ioc|rar|zip|txt|flv|mid|doc|
ppt|pdf|xls|mp3|wma)$
        {
                expires 30d;
        }
        location ~ .*\.(js|css)$
        {
                expires 1h;
        }
```

```
                location ~* \.(js|css|jpg|jpeg|gif|png|swf)$ {
                        if (-f $request_filename) {
                                expires 1h;
                                break;
                        }
                }

                location ~* \.(jpg|jpeg|gif|css|png|js|ico)$ {
                        expires max;
                }

                #cache control: all statics are cacheable for
24 hours
                location / {
                        if ($request_uri ~* \.
(ico|css|js|gif|jpe?g|png)$) {
                                expires 72h;
                                break;
                        }
                }
```

例 1.3. nginx expires

```
	location ~ .*\.(gif|jpg|jpeg|png|bmp|swf|ico)$ {
		expires 1d;
		access_log off;
	}

	location ~ .*\.(js|css)$ {
		expires 1d;
		access_log off;
	}
	location ~ .*\.(html|htm)$
	{
		expires 1d;
		access_log off;
	}
```

通过 add_header / more_set_headers 设置缓存

add_header 实例

```
	location ~* \.(?:ico|css|js|gif|jpe?g|png)$ {
		expires 30d;
		add_header Pragma public;
		add_header Cache-Control "public";
	}
```

more_set_headers 实例

```
	location ~ \.(ico|pdf|flv|jp?g|png|gif|js|css|webp|swf)
(\.gz)?(\?.*)?$ {
		more_set_headers 'Cache-Control: max-
age=86400';
```

```
                ...
                proxy_cache_valid 200 2592000;
                ...
        }
```

s-maxage 作用于 Proxy

```
        location ~ \.(ico|pdf|flv|jp?g|png|gif|js|css|webp|swf)
(\.gz)?(\?.*)?$ {
                more_set_headers 'Cache-Control: s-
maxage=86400';
        }
```

$request_uri

```
                if ($request_uri ~* "\.(ico|css|js|gif|jpe?
g|png)\?[0-9]+$") {
                        expires max;
                        break;
                }
```

下面例子是缓存 /detail/html/5/4/321035.html， 但排除 /detail/html/5/4/0.html

```
        if ($request_uri ~ ^/detail/html/[0-9]+/[0-9]/[^0][0-
9]+\.html ) {
                expires 1d;
        }
```

$request_filename

```
	if (-f $request_filename) {
		expires 1d;
	}
```

access

```
	#防止access文件被下载
	location ~ /\.ht {
		deny all;
	}
```

```
	location ~ ^/upload/.*\.php$
	{
		deny all;
	}

	location ~ ^/static/images/.*\.php$
	{
		deny all;
	}
```

```
	location ~ /\.ht {
		deny all;
	}

	location ~ .*\.(sqlite|sq3)$ {
		deny all;
```

```
        }
```

IP 地址

```
                location / {
                        deny 192.168.0.1;
                        allow 192.168.1.0/24;
                        allow 10.1.1.0/16;
                        allow 2001:0db8::/32;
                        deny all;
                }
```

限制IP访问*.php文件

```
        location ~ ^/private/.*\.php$
        {
                allow 222.222.22.35;
                allow 192.168.1.0/249;
                deny all;
        }
```

autoindex

开启目录浏览

```
# vim /etc/nginx/sites-enabled/default

location / {
        autoindex on;
}
```

```
# /etc/init.d/nginx reload
Reloading nginx configuration: nginx.
```

另外Nginx的目录流量有两个比较有用的参数，可以根据自己的需求添加：

```
autoindex_exact_size off;
默认为on，显示出文件的确切大小，单位是bytes。
改为off后，显示出文件的大概大小，单位是kB或者MB或者GB

autoindex_localtime on;
默认为off，显示的文件时间为GMT时间。
改为on后，显示的文件时间为文件的服务器时间
```

try_files

```
        server {
                listen 80;
                server_name www.example.com example.com;

                location / {
                        try_files $uri $uri/
/index.php?/$request_uri;
                }

                location /example {
                        alias /www/example/;
                        index index.php index.html;
                }
                error_page 500 502 503 504 /50x.html;
                        location = /50x.html {
                        root /usr/share/nginx/html;
                }

                location ~ \.php$ {
                        root html;
                        fastcgi_pass 127.0.0.1:9000;
                        fastcgi_index index.php;
                        fastcgi_param SCRIPT_FILENAME
/www/example$fastcgi_script_name;
                        include fastcgi_params;
                }

                location ~ /\.ht {
                        deny all;
                }
        }
```

add_header

Cache

# 相关页面设置Cache-Control头信息

```
        if ($request_uri ~* "^/$|^/news/.+/|^/info/.+/") {
                add_header Cache-Control max-age=3600;
        }

        if ($request_uri ~* "^/suggest/|^/categories/") {
                add_header Cache-Control max-age=86400;
        }
```

```
add_header     Nginx-Cache     "HIT  from  www.example.com";
or
```

```
add_header     Nginx-Cache     "$upstream_cache_status  from
www.example.com";
```

Access-Control-Allow

```
                location ~* \.(eot|ttf|woff)$ {
                        add_header Access-Control-Allow-Origin
*;
                }

                location /js/ {
                        add_header Access-Control-Allow-Origin
https://www.mydomain.com/;
                        add_header Access-Control-Allow-Methods
GET,OPTIONS;
                        add_header Access-Control-Allow-Headers
*;
                }
```

```
        location / {
                if ($request_method = OPTIONS ) {
                        add_header Access-Control-Allow-Origin
"http://example.com";
                        add_header Access-Control-Allow-Methods
"GET, OPTIONS";
                        add_header Access-Control-Allow-Headers
"Authorization";
                        add_header Access-Control-Allow-
Credentials "true";
                        add_header Content-Length 0;
                        add_header Content-Type text/plain;
                        return 200;
                }
        }
```

允许 所有头部 所有域 所有方法

```
server {
        ...
        location / {
                add_header 'Access-Control-Allow-Origin' '*';
                add_header 'Access-Control-Allow-Headers' '*';
                add_header 'Access-Control-Allow-Methods' '*';
                # OPTIONS 直接返回204
                if ($request_method = 'OPTIONS') {
                        return 204;
                }
        }
        ...
}
```

使用 $http_origin 变量

```
add_header 'Access-Control-Allow-Origin' $http_origin;
add_header 'Access-Control-Allow-Credentials' 'true';
add_header 'Access-Control-Allow-Methods' 'GET, POST, OPTIONS';
add_header 'Access-Control-Allow-Headers' 'DNT,web-token,app-
token,Authorization,Accept,Origin,Keep-Alive,User-Agent,X-Mx-
ReqToken,X-Data-Type,X-Auth-Token,X-Requested-With,If-Modified-
Since,Cache-Control,Content-Type,Range';
add_header 'Access-Control-Expose-Headers' 'Content-
Length,Content-Range';
if ($request_method = 'OPTIONS') {
        add_header 'Access-Control-Max-Age' 1728000;
        add_header 'Content-Type' 'text/plain; charset=utf-8';
        add_header 'Content-Length' 0;
        return 204;
}
```

client_max_body_size 上传文件尺寸限制

```
client_max_body_size 2M;
```

return

301 跳转

```
        server {
                listen 80;
                server_name m.example.com;

                location / {
                        return 301
$scheme://www.example.com$request_uri;
                }
        }

        server {
                listen 80;
                listen 443 ssl;
                server_name www.old-name.com;
                return 301 $scheme://www.new-
name.com$request_uri;
        }
```

3.6. rewrite

```
Rewrite Flags
last - 基本上都用这个Flag。
break - 中止Rewirte，不在继续匹配
redirect - 返回临时重定向的HTTP状态302
permanent - 返回永久重定向的HTTP状态301
```

```
文件及目录匹配，其中:
-f和!-f用来判断是否存在文件
-d和!-d用来判断是否存在目录
-e和!-e用来判断是否存在文件或目录
-x和!-x用来判断文件是否可执行

正则表达式全部符号解释
~ 为区分大小写匹配
~* 为不区分大小写匹配
!~和!~* 分别为区分大小写不匹配及不区分大小写不匹配
(pattern) 匹配 pattern 并获取这一匹配。所获取的匹配可以从产生的
Matches 集合得到，在VBScript 中使用 SubMatches 集合，在JScript 中则
使用 $0…$9 属性。要匹配圆括号字符，请使用 ‘\(’ 或 ‘\)’。
^ 匹配输入字符串的开始位置。
$ 匹配输入字符串的结束位置。
```

```
        server {
                listen 80;
                server_name www.example.com example.com ;
                if ($host = "example.com" )
                {
                        rewrite ^/(.*)$
http://www.example.com/$1 permanent;
                }
                if ($host != "www.example.com" )
                {
                        rewrite ^/(.*)$
http://www.example.com/$1 permanent;
                }
        }
```

处理泛解析

```
        if ($host ~ '(.*)\.example\.com' ) {
                set $subdomain $1;
                rewrite "^/(.*)$" /$subdomain/$1;
        }
```

处理扩展名

```
        location ~* \.(js|css|jpg|jpeg|gif|png|swf)$ {
                if (!-f $request_filename){
                        rewrite /(.*)
http://images.example.com/$1;
                }
        }
```

http get 参数处理

需求如下

```
原理地址:
http://www.netkiller.cn/redirect/index.html?skuid=133

目的地址:
http://www.netkiller.cn/to/133.html
```

注意：nginx rewrite 并不支持http get 参数处理，也就是说“?”之后的内容rewrite根部获取不到。

下面的例子是行不通的

```
rewrite ^/redirect/index\.html\?skuid=(\d+)$ /to/$1.html
permanent ;
```

我们需要通过正在查出参数，然后赋值一个变量，再将变量传递给rewrite。具体做法是：

```
server {
    listen       80;
    server_name www.netkiller.cn;

    #charset koi8-r;
    access_log  /var/log/nginx/test.access.log  main;

    location / {
        root   /www/test;
        index  index.html;

                if ($request_uri ~* "^/redirect/index\.html\?
skuid=([0-9]+)$") {
                set $argv1 $1;
                rewrite .* /to/$argv1.html? permanent;
        }
    }
}
```

测试结果

```
[neo@netkiller conf.d]$ curl -I
http://www.netkiller.cn/redirect/index.html?skuid=133
HTTP/1.1 301 Moved Permanently
Server: nginx
Date: Tue, 12 Apr 2016 06:59:33 GMT
Content-Type: text/html
```

```
Content-Length: 178
Location: http://www.netkiller.cn/to/133.html
Connection: keep-alive
```

正则取非

需求如下，除了2015年保留，其他所有页面重定向到新页面

```
				rewrite ^/promotion/(?!2015\/)(.*) https://www.netkiller.cn/promotion.html permanent;
```

去掉扩展名

需求

```
http://www.example.com/article/10       => http://www.example.com/article/10.html
```

```
location / {
    if (!-e $request_filename){
        rewrite ^(.*)$ /$1.html last;
        break;
    }
}
```

添加扩展名

```
```

```
原地址
http://ipfs.netkiller.cn/ipfs/QmcA1Fsrt6jGTVqAUNZBqaprMEdFaFkmk
zA5s2M6mF85UC
目标地址:
http://ipfs.netkiller.cn/ipfs/QmcA1Fsrt6jGTVqAUNZBqaprMEdFaFkmk
zA5s2M6mF85UC.mp4
```

```
    location / {
        rewrite ^/(.*)\.mp4$ /$1 last;
        proxy_pass      http://127.0.0.1:8080;
    }
```

3.7. upstream 负载均衡

```
http {
        upstream myapp1 {
                server srv1.example.com;
                server srv2.example.com;
                server srv3.example.com;
        }

        server {
                listen 80;
                location / {
                        proxy_pass http://myapp1;
                }
        }
}
```

weight 权重配置

```
        upstream myapp1 {
                server srv1.example.com weight=3;
                server srv2.example.com;
                server srv3.example.com;
        }
```

backup 实现热备

```
        upstream backend {
                server backend1.example.com weight=5;
                server backend2.example.com:8080;
                server unix:/tmp/backend3;

                server backup1.example.com:8080 backup;
                server backup2.example.com:8080 backup;
        }

        server {
                location / {
                        proxy_pass http://backend;
                }
        }
```

3.8. Proxy

ngx_http_proxy_module

```
# cat /etc/nginx/nginx.conf

#user  nobody;
worker_processes  4;

#error_log  logs/error.log;
#error_log  logs/error.log  notice;
#error_log  logs/error.log  info;
```

```
#pid        logs/nginx.pid;


events {
    worker_connections  40960;
        use epoll;
}


http {
    include       mime.types;
    default_type  application/octet-stream;

    #log_format  main  '$remote_addr - $remote_user
[$time_local] "$request" '
    #                  '$status $body_bytes_sent
"$http_referer" '
    #                  '"$http_user_agent"
"$http_x_forwarded_for"';

    #access_log  logs/access.log  main;

    access_log  /dev/null;

    sendfile        on;
    #tcp_nopush     on;

    #keepalive_timeout  0;
    keepalive_timeout  65;

    #gzip  on;

upstream backend{
#        server 172.16.0.6:80;
        server 10.0.0.68:80;
        server 10.0.0.69:80;
}


    server {
        listen       80;
        server_name  localhost;

        #charset koi8-r;
        #access_log  logs/host.access.log  main;

#         location / {
#             root   html;
#             index  index.html index.htm;
#         }

    access_log  /dev/null;
    error_log   /dev/null;


    location / {
#        proxy_pass $scheme://$host$request_uri;
#        proxy_set_header Host $http_host;

#        proxy_buffers 256 4k;
#        proxy_max_temp_file_size 0;

#        proxy_connect_timeout 30;

#        proxy_cache_valid 200 302 10m;
#        proxy_cache_valid 301 1h;
#        proxy_cache_valid any 1m;



         proxy_pass      http://backend;

         proxy_redirect          off;
         proxy_set_header        Host $host;
#         proxy_set_header        X-Real-IP $remote_addr;
#         proxy_set_header        X-Forwarded-For
$proxy_add_x_forwarded_for;
         client_max_body_size    10m;
         client_body_buffer_size 128k;
         proxy_connect_timeout   30;
         proxy_send_timeout      30;
         proxy_read_timeout      30;
         proxy_buffer_size       4k;
         proxy_buffers           256 4k;
         proxy_busy_buffers_size 64k;
         proxy_temp_file_write_size 64k;
        tcp_nodelay on;
    }
```

```
        #error_page  404              /404.html;

        # redirect server error pages to the static page
/50x.html
        #
        error_page   500 502 503 504  /50x.html;
        location = /50x.html {
            root   html;
        }
    }

}
```

proxy_cache

/etc/nginx/conf.d/

```
proxy_cache_path /www/cache keys_zone=www:128m;
server {

        location / {
      proxy_pass http://example.net;
      proxy_cache www;
      proxy_cache_key $uri;
      proxy_cache_valid  200 302  60m;
      proxy_cache_valid  404      1m;
    }
}
```

proxy_cache_valid 配置HTTP状态码与缓存时间

```
proxy_cache_valid any 1m; 任何内容缓存一分钟

proxy_cache_valid  200 302  60m; 状态200，302页面缓存 60分钟

proxy_cache_valid  404      1m;  状态404页面缓存1分钟
```

```
http {
  proxy_cache_path  /var/www/cache levels=1:2 keys_zone=my-
cache:8m max_size=1000m inactive=600m;
  proxy_temp_path /var/www/cache/tmp;


  server {
    location / {
      proxy_pass http://example.net;
      proxy_cache mycache;
      proxy_cache_valid  200 302  60m;
      proxy_cache_valid  404      1m;
    }
  }
}
```

```
location / {
  proxy_pass http://localhost;
  proxy_set_header   Host             $host;
  proxy_set_header   X-Real-IP        $remote_addr;
  proxy_set_header   X-Forwarded-For
$proxy_add_x_forwarded_for;
  proxy_ignore_headers Set-Cookie;
  proxy_ignore_headers Cache-Control;
  proxy_cache_bypass        $http_secret_header;
  add_header X-Cache-Status $upstream_cache_status;
}
```

```
server {
          listen       80;
          server_name  example.org;
          root   /var/www;
          index  index.html index.php;

        location ~* .+.(ico|jpg|gif|jpeg|css|js|flv|png|swf)$ {
                expires max;
        }

        location / {
                proxy_pass       http://backend;
                proxy_set_header  X-Real-IP  $remote_addr;
                proxy_set_header Host $http_host;
                proxy_cache cache;
                proxy_cache_key $host$request_uri;
                proxy_cache_valid 200 304 12h;
                proxy_cache_valid 302 301 12h;
                proxy_cache_valid any 1m;
                proxy_ignore_headers Cache-Control Expires;
                proxy_pass_header Set-Cookie;
        }

}
```

```
proxy_cache_valid 200 302 10m;
proxy_cache_valid 301      1h;
proxy_cache_valid any      1m;
```

rewrite + proxy_pass

需求如下

```
http://www.example.com/images/logo.jpg  =>
http://images.example.com/logo.jpg
```

如果直接 proxy_pass http://images.example.com; 的后果是 http://images.example.com/images/logo.jpg，我们需要去掉images目录，这里使用rewrite /images/(.+)$ /$1 break;实现

```
    location ^~ /images/ {
                rewrite /images/(.+)$ /$1 break;
                proxy_pass http://images.example.com;
                break;
    }
```

request_filename + proxy_pass

如果文件不存在，那么去指定的节点上寻找

```
   location / {
        root  /www;
        proxy_intercept_errors  on;
        if (!-f $request_filename) {
          proxy_pass http://172.16.1.1;
          break;
        }
    }
        location / {
        root  /www/images;
        proxy_intercept_errors  on;
        if (!-f $request_filename) {
          proxy_pass http://172.16.1.2;
          break;
        }
    }
```

$request_uri 与 proxy_pass 联合使用

```
server {
    listen       80;
    server_name  info.example.com;

    #charset koi8-r;
    access_log  /var/log/nginx/info.example.com.access.log
main;

    location / {
        root   /www/example.com/info.example.com;
        index  index.html index.htm;

        rewrite ^/$  http://www.example.com/;

        valid_referers none blocked *.example.com;
        if ($invalid_referer) {
                #rewrite ^(.*)$  http://www.example.com/cn/$1;
                return 403;
        }

        proxy_intercept_errors  on;
#           proxy_set_header  X-Real-IP  $remote_addr;
#            proxy_set_header  X-Forwarded-For
$proxy_add_x_forwarded_for;
#            proxy_set_header  Host            $host;
#
#            proxy_cache one;
#            proxy_cache_valid  200 302 304 10m;
#            proxy_cache_valid  301 1h;
#            proxy_cache_valid  any 1m;

        if ( $request_uri ~
"^/public/datas/(sge|cgse|futures|fx_price|gold_price|stock|bon
ds)\.xml$") {
                proxy_pass http://211.176.212.212$request_uri;
                break;
        }

        if (!-f $request_filename) {

          proxy_pass http://infoadmin.example.com;
          #proxy_pass http://backend;
          break;
        }
    }

    location ~ ^/index\.php$ {
        return 403;
    }
    location ~ ^/(config|include|crontab|/systemmanage)/ {
        deny all;
        break;
    }
    #error_page  404              /404.html;

    # redirect server error pages to the static page /50x.html
    #
    error_page   500 502 503 504  /50x.html;
    location = /50x.html {
        root   /usr/share/nginx/html;
    }
```

```
}
```

try_files 与 proxy_pass 共用

需求，在web目录下索引静态，如果不存在便进入proxy处理，通常proxy后面是tomcat等应用服务器。

我们可以使用 try_files 与 proxy_pass 实现我们的需求

```
server {
    listen       80;
    server_name  m.netkiller.cn;

    charset utf-8;
    access_log  /var/log/nginx/m.netkiller.cn.access.log;

    location / {
                root /www/example.com/m.example.com;
                try_files $uri $uri/ @proxy;
    }

    location @proxy {
                proxy_pass http://127.0.0.1:8000;
        proxy_set_header    Host    $host;
        proxy_set_header    X-Real-IP   $remote_addr;
        proxy_set_header    X-Forwarded-For
$proxy_add_x_forwarded_for;
    }

    #error_page  404              /404.html;

    # redirect server error pages to the static page /50x.html
    #
    error_page   500 502 503 504  /50x.html;
    location = /50x.html {
        root   /usr/share/nginx/html;
    }

    # deny access to .htaccess files, if Apache's document root
    # concurs with nginx's one
    #
    #location ~ /\.ht {
    #    deny  all;
    #}

    location ~ ^/WEB-INF/ {
        deny  all;
    }

    location ~ \.(html|js|css|jpg|png|gif|swf)$ {
        root /www/example.com/m.example.com;
        expires 1d;
    }
    location ~ \.(ico|fla|flv|mp3|mp4|wma|wmv|exe)$ {
        root /www/example.com/m.example.com;
        expires 7d;
    }
    location ~ \.flv {
        flv;
    }

    location ~ \.mp4$ {
        mp4;
    }

    location /module {
        root /www/example.com/m.example.com;
    }

}
```

Proxy 与 SSI

背景：nginx + tomcat 模式，nginx 开启 SSI , Tomcat 动态页面中输出 SSI 标签

```
# cat  /etc/nginx/conf.d/www.netkiller.cn.conf
server {
    listen       80;
    server_name  www.netkiller.cn;

    charset utf-8;
    access_log  /var/log/nginx/www.netkiller.cn.access.log;

    location / {
        #index  index.html index.htm;
                proxy_pass http://127.0.0.1:8080;
        proxy_set_header    Host    $host;
        proxy_set_header    X-Real-IP   $remote_addr;
        proxy_set_header    X-Forwarded-For
$proxy_add_x_forwarded_for;
    }

    #error_page  404              /404.html;

    # redirect server error pages to the static page /50x.html
    #
    error_page   500 502 503 504  /50x.html;
    location = /50x.html {
        root   /usr/share/nginx/html;
    }
}
```

test.jsp 文件

```
<%@ page language="java"
import="java.util.*,java.text.SimpleDateFormat"
pageEncoding="UTF-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
        <head>
        <title>show time</title>
</head>
<body>
<%

        Date date=new Date();
    SimpleDateFormat ss=new SimpleDateFormat("yyyy-MM-dd
HH:mm:ss");
    String lgtime=ss.format(date);
%>
        <center>
        <h1><%=lgtime%></h1>
        </center>

        <!--# set var="test" value="Hello netkiller!" -->
        <!--# echo var="test" -->

</body>
</html>
```

测试并查看源码，你会看到SSI标签

```
        <!--# set var="test" value="Hello netkiller!" -->
        <!--# echo var="test" -->
```

解决方案

```
    location / {
        ssi on;
        proxy_set_header Accept-Encoding "";
        proxy_pass http://127.0.0.1:8080;
        proxy_set_header    Host    $host;
        proxy_set_header    X-Real-IP   $remote_addr;
        proxy_set_header    X-Forwarded-For
$proxy_add_x_forwarded_for;
    }
```

再次测试，你将看不到SSI标签，只能看到文本输出Hello netkiller!

Host

Proxy 通过IP地址访问目的主机，如果目的主机是虚拟主机，你就需要告诉目的主机是那个域名。

proxy_set_header Host www.example.com;

proxy_set_header Host $server_name;

```
server {
    listen       80;
    server_name  www.example.com;

    charset utf-8;
    access_log  /var/log/nginx/www.example.com.access.log
main;

    proxy_set_header   Host $server_name;

    location / {
        proxy_pass http://154.21.16.57;
    }

    #error_page  404              /404.html;

    # redirect server error pages to the static page /50x.html
    #
    error_page   500 502 503 504  /50x.html;
    location = /50x.html {
        root   /usr/share/nginx/html;
    }
}
```

expires

```
location / {
    root /var/www;
    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_redirect false;

    if ($request_uri ~* "\.(ico|css|js|gif|jpe?g|png)\?[0-
9]+$") {
        expires max;
        break;
    }
    if (-f $request_filename) {
        break;
    }
    if (-f $request_filename/index.html) {
        rewrite (.*) $1/index.html break;
    }
    if (-f $request_filename.html) {
        rewrite (.*) $1.html break;
    }

    proxy_pass http://backend;
}
```

X-Forwarded-For

```
proxy_set_header    X-Real-IP   $remote_addr;
proxy_set_header    X-Forwarded-For $proxy_add_x_forwarded_for;
```

X-Sendfile

http://wiki.nginx.org/NginxXSendfile

proxy_http_version

proxy_http_version 1.0 | 1.1;

proxy_set_header

```
proxy_set_header    Host    $host;
proxy_set_header    X-Real-IP   $remote_addr;
proxy_set_header    X-Forwarded-For $proxy_add_x_forwarded_for;
proxy_set_header User-Agent "Mozilla/5.0 (compatible; MSIE
10.6; Windows NT 6.1; Trident/5.0; InfoPath.2; SLCC1; .NET CLR
3.0.4506.2152; .NET CLR 3.5.30729; .NET CLR 2.0.50727) 3gpp-gba
UNTRUSTED/1.0";
proxy_set_header X-Forwarded-URI $request_uri;
```

隐藏头部信息

例如响应头:

```
Connection: keep-alive
Content-Encoding: gzip
Content-Type: text/html; charset=UTF-8
Date: Thu, 04 Feb 2018 02:20:36 GMT
Transfer-Encoding: chunked
Vary: Accept-Encoding
X-Powered-By: PHP/7.2.0
```

隐藏 PHP 版本信息

```
proxy_hide_header X-Powered-By;
```

忽略头

```
location /images/ {
    proxy_cache my_cache;
    proxy_ignore_headers Cache-Control;
    proxy_cache_valid any 30m;
    ...
}
```

proxy_pass_request_headers 透传 Header

有时用户会设置自定义的 HTTP 头信息，这些不符合 HTTP 的头信息如果需要会被 proxy_pass 过滤并丢弃。

```
proxy_pass_request_headers off;
```

默认系统是开启的

```
proxy_pass_request_headers on;
```

timeout 超时时间

proxy_connect_timeout: 链接超时设置，后端服务器连接的超时时间，发起握手等候响应超时时间。

proxy_read_timeout: 连接成功后，等候后端服务器响应时间，其实已经进入后端的排队之中等候处理，也可以说是后端服务器处理请求的时间。

proxy_send_timeout: 后端服务器数据回传时间，就是在规定时间之内后端服务器必须传完所有的数据。

```
    location / {
        proxy_pass http://127.0.0.1:8000;
        proxy_set_header    Host    $host;
        proxy_set_header    X-Real-IP   $remote_addr;
        proxy_set_header    X-Forwarded-For
$proxy_add_x_forwarded_for;
        proxy_connect_timeout 60s;
        proxy_read_timeout 300s;
        proxy_send_timeout 300s;
    }
```

sub_filter 文本替换

```
        proxy_set_header Accept-Encoding ""; # 防止网站gzip
        sub_filter '景峰' '景峯'; # 有效
        sub_filter 'neo' 'netkiller';
        sub_filter_once off;    # 全部替换
```

example

/api/ 走代理，其他页面走 Nginx

代理特定目录

```
server {
    listen 443 ssl http2;
    server_name  www.netkiller.cn netkiller.cn;

    ssl_certificate ssl/netkiller.cn.crt;
    ssl_certificate_key ssl/netkiller.cn.key;
    ssl_session_cache shared:SSL:20m;
    ssl_session_timeout 60m;
    charset utf-8;
    #access_log  /var/log/nginx/host.access.log  main;

    location / {
        root   /opt/netkiller.cn/www.netkiller.cn;
        index  index.html;
    }

    location ^~ /api/ {
        proxy_pass http://127.0.0.1:8080;
        proxy_http_version 1.1;
                proxy_set_header    Host    $host;
                break;
    }

    #error_page  404              /404.html;

    # redirect server error pages to the static page /50x.html
    #
    error_page   500 502 503 504  /50x.html;
    location = /50x.html {
        root   /usr/share/nginx/html;
    }

    # deny access to .htaccess files, if Apache's document root
    # concurs with nginx's one
    #
    location ~ /\.ht {
        deny  all;
    }

}
```

upstream 实例

```
127.0.0.1               api.example.com
172.16.0.10     api1.example.com
172.16.0.11     api2.example.com
```

```
upstream api.example.com {
    least_conn;
    server api1.example.com;
    server api2.example.com;
}

server {
    listen       80;
    server_name  api.example.com;

    charset utf-8;
    access_log  /var/log/nginx/api.example.com.access.log;

    location / {
                proxy_pass        http://api.example.com;
                proxy_set_header X-Real-IP  $remote_addr;
                #proxy_set_header Host $host;
                proxy_set_header Host api.example.com;
    }

    #error_page  404              /404.html;

    # redirect server error pages to the static page /50x.html
    #
    error_page   500 502 503 504  /50x.html;
    location = /50x.html {
        root   /usr/share/nginx/html;
    }

    # proxy the PHP scripts to Apache listening on 127.0.0.1:80
    #
    #location ~ \.php$ {
    #    proxy_pass   http://127.0.0.1;
    #}
}
```

Tomcat 实例

```
server {
    listen       80;
    server_name  m.netkiller.cn;
    charset utf-8;
    access_log  /var/log/nginx/m.netkiller.cn.access.log;

    location / {
                root /www/example.com/m.example.com;
                rewrite ^(.*)\;jsessionid=(.*)$ $1 break;
                try_files $uri $uri/ @proxy;
    }

    location @proxy {
                proxy_pass http://127.0.0.1:8000;
        proxy_set_header    Host    $host;
        proxy_set_header    X-Real-IP   $remote_addr;
        proxy_set_header    X-Forwarded-For
$proxy_add_x_forwarded_for;
    }

    #error_page  404              /404.html;

    # redirect server error pages to the static page /50x.html
    #
    error_page   500 502 503 504  /50x.html;
    location = /50x.html {
        root   /usr/share/nginx/html;
    }

    # deny access to .htaccess files, if Apache's document root
    # concurs with nginx's one
    #
    #location ~ /\.ht {
    #    deny  all;
    #}

    location ~ ^/WEB-INF/ {
        deny  all;
    }

    location ~ \.(html|js|css|jpg|png|gif|swf)$ {
        root /www/example.com/m.example.com;
        expires 1d;
    }
    location ~ \.(ico|fla|flv|mp3|mp4|wma|wmv|exe)$ {
        root /www/example.com/m.example.com;
        expires 7d;
    }
    location ~ \.flv {
        flv;
    }

    location ~ \.mp4$ {
        mp4;
    }

    location /module {
        root /www/example.com/m.example.com;
    }

}
```

上面的jsessionid处理方式

Nginx -> Nginx -> Tomcat

背景各种原因需要再Nginx前面再增加一层Nginx虽然需求很变态，本着学习的目的试了试。

这里还使用了 http2 加速 nginx ssl http2 -> nginx ssl http2 -> Tomcat 8080

```
    server {
        listen       443 ssl http2;
        server_name  www.netkiller.cn;

        ssl_certificate      ssl/netkiller.cn.crt;
        ssl_certificate_key  ssl/netkiller.cn.key;

    #    ssl_session_cache    shared:SSL:1m;
    #    ssl_session_timeout  5m;
    #    ssl_ciphers  HIGH:!aNULL:!MD5;
    #    ssl_prefer_server_ciphers  on;

        location / {

                proxy_buffers 16 4k;
                proxy_buffer_size 2k;

                proxy_pass https://www.netkiller.cn;
                proxy_pass_header   Set-Cookie;
                add_header From     www.netkiller.cn;
                proxy_set_header    Cookie $http_cookie;
                proxy_set_header    Host    $host;
                proxy_set_header    X-Real-IP   $remote_addr;
                proxy_set_header    X-Forwarded-For
$proxy_add_x_forwarded_for;
                proxy_cookie_domain www.netkiller.cn
netkiller.cn;
                #proxy_cookie_path / "/; secure; HttpOnly";
                proxy_set_header Accept-Encoding "";
                proxy_ignore_client_abort  on;

        }
    }
```

有几点需要注意：

如果是443你需要挂在证书，需要透传cookie给目的主机，否则你将无法支持Session，应用程序需要从 X-Forwarded-For 获取IP地址。

Proxy 处理 Cookie

下面是一个通过 proxy_pass 代理live800的案例，我们需要处理几个地方：

Host头处理，Cookie传递，替换原因页面中的域名，替换文件有html,css,xml,css,js

```


    location ~ ^/k800 {
        #rewrite              ^/live800/(.*)  /$1 break;

        proxy_pass            http://118.23.24.15;
        proxy_pass_header    Set-Cookie;
        proxy_set_header     Accept-Encoding "";
        proxy_set_header     X-Forwarded-For
```

```
$proxy_add_x_forwarded_for;
        proxy_set_header    Host www.example.com;
        proxy_set_header    Cookie $http_cookie;

        sub_filter_types text/html text/css text/xml text/css
text/javascript;
        sub_filter 'www.example.com'  '$host';
        sub_filter_once off;
    }
```

Proxy 添加 CORS 头

```
server {
    listen       80;
    listen 443 ssl http2;

    server_name api.netkiller.cn;
    ssl_certificate ssl/netkiller.cn.crt;
    ssl_certificate_key ssl/netkiller.cn.key;
    ssl_session_cache shared:SSL:20m;
    ssl_session_timeout 60m;
    ssl_protocols TLSv1 TLSv1.1 TLSv1.2;

    charset utf-8;
    access_log  /var/log/nginx/api.netkiller.cn.access.log
main;

    location / {
        proxy_pass      http://127.0.0.1:7000;
    }

    location ^~ /api {
                add_header Access-Control-Allow-Origin *;
                add_header Access-Control-Allow-Headers
Content-Type,Origin;
                add_header Access-Control-Allow-Methods
GET,OPTIONS;
        proxy_pass http://other.example.com/api/;
    }
}
```

```

```

通过 Proxy 汉化 restful 接口

通过 proxy 汉化 restful 接口返回的 json 字符串。

背景，有这样一个需求，前端HTML5通过ajax与restful交互，ajax会显示接口返回json数据，由于js做了混淆无法修改与restful交互的逻辑，但是json反馈结果需要汉化。

汉化前接口如下，返回message为 "message":"Full authentication is required to access this resource"

```
neo@netkiller ~/workspace/Developer/Python % curl
http://api.netkiller.cn/restful/member/get/1.json

{"timestamp":1505206067543,"status":401,"error":"Unauthorized",
"message":"Full authentication is required to access this
resource","path":"/restful/member/get/1.json"}
```

建立一个代理服务器，代理介于用户和接口之间，ajax 访问接口需要经过这个代理服务器中转。

增加 /etc/nginx/conf.d/api.netkiller.cn.conf 配置文件

```
server {
        listen 80;
        server_name api.netkiller.cn;

        charset utf-8;

        location / {
                proxy_pass http://localhost:8443;
                proxy_http_version 1.1;
                proxy_set_header    Host    $host;

                sub_filter_types application/json;
        sub_filter 'Full authentication is required to access
this resource'  '用户验证错误';
        sub_filter_once off;
        }

}
```

所谓汉化就是字符串替换，使用nginx sub_filter 模块。

重新启动 nginx 然后测试汉化效果

```
neo@netkiller ~/workspace/Developer/Python % curl
http://api.netkiller.cn/restful/member/get/1.json

{"timestamp":1505208931927,"status":401,"error":"Unauthorized",
"message":"用户验证错误","path":"/restful/member/get/1.json"}
```

现在我们看到效果是 "message":"用户验证错误"

HTTP2 proxy_pass http://

```
server {
    listen       80;
    listen 443 ssl http2;
    server_name  www.netkiller.cn netkiller.cn;

    ssl_certificate ssl/netkiller.cn.crt;
    ssl_certificate_key ssl/netkiller.cn.key;
    ssl_session_cache shared:SSL:20m;
    ssl_session_timeout 60m;

    charset utf-8;
    #access_log  /var/log/nginx/host.access.log  main;

    error_page  497              https://$host$uri?$args;

    if ($scheme = http) {
        return 301 https://$server_name$request_uri;
    }

    location ^~ /member/ {
        proxy_pass https://47.75.176.32:443;
                proxy_set_header    Host www.netkiller.cn;
        break;
    }

    location / {
        root   /opt/www.netkiller.cn;
        index  index.html index.php;
    }

    #error_page  404              /404.html;

    # redirect server error pages to the static page /50x.html
    #
    error_page   500 502 503 504  /50x.html;
    location = /50x.html {
        root   /usr/share/nginx/html;
    }

    # proxy the PHP scripts to Apache listening on 127.0.0.1:80
    #
    #location ~ \.php$ {
    #    proxy_pass   http://127.0.0.1;
    #}

    # pass the PHP scripts to FastCGI server listening on
127.0.0.1:9000
    #
    location ~ \.php$ {
        root           html;
        fastcgi_pass   127.0.0.1:9000;
        fastcgi_index  index.php;
        fastcgi_param  SCRIPT_FILENAME
/opt/www.netkiller.cn$fastcgi_script_name;
        include        fastcgi_params;
    }
    # deny access to .htaccess files, if Apache's document root
    # concurs with nginx's one
    #
    location ~ /\.ht {
        deny  all;
    }

}
```

IPFS

```
mkdir -p /var/cache/nginx/ipfs
chown nginx:root /var/cache/nginx/ipfs
```

```
proxy_cache_path /var/cache/nginx/ipfs keys_zone=ipfs:4096m;
server {
    listen       80;
    server_name  localhost;

    #charset koi8-r;
    access_log  /var/log/nginx/ipfs.access.log;

    location / {
        proxy_pass      http://127.0.0.1:8080;
        proxy_cache ipfs;
        proxy_cache_valid  200 30d;
            expires max;
    }

    location ~* .+.(mp4)$ {
        rewrite ^/(.*)\.mp4$ /$1 last;
        proxy_pass      http://127.0.0.1:8080;
        proxy_cache ipfs;
        proxy_cache_valid  200 30d;
        expires max;
        mp4;
    }

    #error_page  404              /404.html;

    # redirect server error pages to the static page /50x.html
    #
    error_page   500 502 503 504  /50x.html;
    location = /50x.html {
        root   /usr/share/nginx/html;
    }

}
```

查看缓存情况

```
[root@netkiller ~]# find /var/cache/nginx/ipfs/
/var/cache/nginx/ipfs/
/var/cache/nginx/ipfs/47c3015c7a497f26f650a817f5a179ab
```

HTTP Auth 认证冲突

nginx 代理 springboot，Springboot 使用了 JWT 认证，HTTP头为 Authorization: Bearer {BASE64}

admin.netkiller.cn 后台需要限制登陆，公司没有固定IP地址，尝试了 VPN 方案，被封。最终决定使用 HTTP Auth，HTTP Auth 使用 HTTP Authorization: Basic {BASE64}。

问题来了，由于HTTP的 key 都是 Authorization，Authorization: Basic 会覆盖掉 Authorization: Bearer 导致 Springboot 无法认证返回 401.

使用下面👇配置解决，注意⚠️调试的时候需要每次关闭浏览器，否则会保留状态，不生效。

```
auth_basic off;
proxy_pass_request_headers on;
```

完成的例子

```
server {
    listen       80;
    listen       443 ssl http2;
    server_name  admin.netkiller.cn;

    include /etc/nginx/default.d/*.conf;

    access_log /var/log/nginx/admin.netkiller.cn.access.log;
    error_log /var/log/nginx/admin.netkiller.cn.error.log;

    error_page 497 https://$host$uri?$args;

    if ($scheme = http) {
            return 301 https://$server_name$request_uri;
    }

    location / {
        auth_basic "Administrator's Area";
        auth_basic_user_file  htpasswd;
        root   /opt/netkiller.cn/admin.netkiller.cn;
            try_files $uri $uri/ /index.html;
            index  index.html;
    }

    location /api/ {
                auth_basic off;
                proxy_pass_request_headers on;
        proxy_set_header Host $http_host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header REMOTE-HOST $remote_addr;
        proxy_set_header X-Forwarded-For
$proxy_add_x_forwarded_for;
        proxy_pass http://api.netkiller.cn:8080/;
    }

    error_page 404 /404.html;
        location = /40x.html {
    }

    error_page 500 502 503 504 /50x.html;
        location = /50x.html {
    }
}
```

3.9. fastcgi

spawn-fcgi

config php fastcgi

```
sudo vim /etc/nginx/sites-available/default

        location ~ \.php$ {
                fastcgi_pass   127.0.0.1:9000;
                fastcgi_index  index.php;
                fastcgi_param  SCRIPT_FILENAME
/scripts$fastcgi_script_name;
                include fastcgi_params;
        }
```

Spawn-fcgi

We still need a script to start our fast cgi processes. We will extract one from Lighttpd. and then disable start script of lighttpd

```
$ sudo apt-get install lighttpd
$ sudo chmod -x /etc/init.d/lighttpd
```

```
$ sudo touch /usr/bin/php-fastcgi
$ sudo vim /usr/bin/php-fastcgi

#!/bin/sh
/usr/bin/spawn-fcgi -a 127.0.0.1 -p 9000 -u www-data -f
/usr/bin/php5-cgi
```

fastcgi daemon

```
$ sudo touch /etc/init.d/nginx-fastcgi
$ sudo chmod +x /usr/bin/php-fastcgi
$ sudo vim /etc/init.d/nginx-fastcgi

This is also a new empty file, add the following and save:

#!/bin/bash
PHP_SCRIPT=/usr/bin/php-fastcgi
RETVAL=0
case "$1" in
start)
$PHP_SCRIPT
RETVAL=$?
;;
stop)
killall -9 php
RETVAL=$?
;;
restart)
killall -9 php
$PHP_SCRIPT
RETVAL=$?
;;
*)
echo "Usage: nginx-fastcgi {start|stop|restart}"
exit 1
```

```
;;
esac
exit $RETVAL

We need to change some permissions to make this all work.
$ sudo chmod +x /etc/init.d/nginx-fastcgi
```

create a test file

```
sudo vim /var/www/nginx-default/index.php
<?php echo phpinfo(); ?>
```

php-fpm

php5-fpm

```
sudo apt-get install php5-cli php5-cgi php5-fpm
```

```
/etc/init.d/php5-fpm start
```

编译 php-fpm

```
./configure --prefix=/srv/php-5.3.8 \
--with-config-file-path=/srv/php-5.3.8/etc \
--with-config-file-scan-dir=/srv/php-5.3.8/etc/conf.d \
--enable-fpm \
--with-fpm-user=www \
--with-fpm-group=www \
--with-pear \
--with-curl \
```

如果出现 fpm 编译错误，取消--with-mcrypt 可以编译成功。

```
--with-gd \
--with-jpeg-dir \
--with-png-dir \
--with-freetype-dir \
--with-xpm-dir \
--with-iconv \
--with-mcrypt \
--with-mhash \
--with-zlib \
--with-xmlrpc \
--with-xsl \
--with-openssl \
--with-mysql=/srv/mysql-5.5.16-linux2.6-i686 \
--with-mysqli=/srv/mysql-5.5.16-linux2.6-i686/bin/mysql_config
\
--with-pdo-mysql=/srv/mysql-5.5.16-linux2.6-i686 \
--with-sqlite=shared \
--with-pdo-sqlite=shared \
--disable-debug \
--enable-zip \
--enable-sockets \
--enable-soap \
--enable-mbstring \
--enable-magic-quotes \
--enable-inline-optimization \
--enable-gd-native-ttf \
--enable-xml \
--enable-ftp \
--enable-exif \
--enable-wddx \
--enable-bcmath \
--enable-calendar \
--enable-sqlite-utf8 \
--enable-shmop \
--enable-dba \
--enable-sysvsem \
--enable-sysvshm \
--enable-sysvmsg

make && make install
```

如果出现 fpm 编译错误，取消--with-mcrypt 可以编译成功。

```
# cp sapi/fpm/init.d.php-fpm /etc/init.d/php-fpm
# chmod 755 /etc/init.d/php-fpm
# ln -s /srv/php-5.3.5 /srv/php
# cp /srv/php/etc/php-fpm.conf.default /srv/php/etc/php-
fpm.conf
# cp php.ini-production /srv/php/etc/php.ini
```

```
groupadd -g 80 www
adduser -o --home /www --uid 80 --gid 80 -c "Web User" www
```

php-fpm.conf

```
# grep -v ';' /srv/php-5.3.5/etc/php-fpm.conf | grep -v "^$"
[global]
pid = run/php-fpm.pid
error_log = log/php-fpm.log
[www]
listen = 127.0.0.1:9000

user = www
group = www
pm = dynamic
pm.max_children = 2048
pm.start_servers = 20
pm.min_spare_servers = 5
pm.max_spare_servers = 35

pm.max_requests = 500
```

```
chkconfig --add php-fpm
```

php-fpm 状态

```
    location /nginx_status {
        stub_status on;
        access_log   off;
        allow 202.82.21.12;
        deny all;
    }
    location ~ ^/(status|ping)$ {
        access_log off;
        allow 202.82.21.12;
        deny all;
        fastcgi_pass 127.0.0.1:9000;
                fastcgi_param SCRIPT_FILENAME
$fastcgi_script_name;
        include fastcgi_params;
    }
```

fastcgi_pass

```
   location ~ ^(.+\.php)(.*)$
   {
        fastcgi_pass 127.0.0.1:9000;
        fastcgi_index   index.php;
        fastcgi_split_path_info ^(.+\.php)(.*)$;
        fastcgi_param   SCRIPT_FILENAME
$document_root$fastcgi_script_name;
        fastcgi_param   PATH_INFO       $fastcgi_path_info;
        fastcgi_param   PATH_TRANSLATED
$document_root$fastcgi_path_info;
        include fastcgi_params;
    }
```

Unix Socket

```
location ~ .*\.(php|php5)?$  {
        #fastcgi_pass  127.0.0.1:9000;
        fastcgi_pass   unix:/dev/shm/php-fpm.sock;
        fastcgi_index index.php;
        include fastcgi.conf;
}
```

nginx example

```
server {
    listen       80;
    listen 443 ssl http2;
    server_name  cms.netkiller.cn;

    ssl_certificate ssl/netkiller.cn.crt;
    ssl_certificate_key ssl/netkiller.cn.key;
    ssl_session_cache shared:SSL:20m;
    ssl_session_timeout 60m;

    charset utf-8;
    access_log  /var/log/nginx/cms.netkiller.cn.access.log
main;

    error_page  497              https://$host$uri?$args;

    if ($scheme = http) {
        return 301 https://$server_name$request_uri;
    }

    location ~ ^/wp-content/uploads/.*\.php$ {
        deny all;
    }

    location / {
        root   /opt/netkiller.cn/cms.netkiller.cn;
        index  index.html index.php;
    }

    #error_page  404              /404.html;

    # redirect server error pages to the static page /50x.html
    #
    error_page   500 502 503 504  /50x.html;
    location = /50x.html {
        root   /usr/share/nginx/html;
    }

    # proxy the PHP scripts to Apache listening on 127.0.0.1:80
    #
    #location ~ \.php$ {
    #    proxy_pass   http://127.0.0.1;
    #}

    # pass the PHP scripts to FastCGI server listening on
127.0.0.1:9000
    #
    location ~ \.php$ {
        root           /opt/netkiller.cn/cms.netkiller.cn;
        fastcgi_pass   127.0.0.1:9000;
        fastcgi_index  index.php;
        fastcgi_param  SCRIPT_FILENAME
/opt/netkiller.cn/cms.netkiller.cn$fastcgi_script_name;
        include        fastcgi_params;
    }

    # deny access to .htaccess files, if Apache's document root
    # concurs with nginx's one
    #
    #location ~ /\.ht {
    #    deny  all;
    #}

}
```

4. Nginx module

4.1. stub_status 服务器状态采集模块

```
location /nginx_status {
        stub_status on;
        access_log  off;
        allow 127.0.0.1;
        deny all;
}
```

php-fpm 状态

```
    location ~ ^/(status|ping)$ {
        access_log off;
        allow 202.82.21.12;
        deny all;
        fastcgi_pass 127.0.0.1:9000;
                fastcgi_param SCRIPT_FILENAME
$fastcgi_script_name;
        include fastcgi_params;
    }
```

4.2. sub_filter 页面中查找和替换

```
location / {
    sub_filter '<a href="http://127.0.0.1:8080/'  '<a
href="https://$host/';
    sub_filter '<img src="http://127.0.0.1:8080/' '<img
src="https://$host/';
```

```
    sub_filter_once on;
}
```

替换掉proxy_pass页面中的内容

```
    location ~ ^/live800 {
        proxy_pass           http://218.23.24.53;
        rewrite              ^/live800/(.*)  /$1 break;
        proxy_set_header    Accept-Encoding "";
        proxy_set_header    X-Forwarded-For
$proxy_add_x_forwarded_for;
        proxy_set_header    Host www.abc.com;

        sub_filter_types text/html text/css text/xml text/css
text/javascript;
        sub_filter 'www.abc.com'  '$host';
        sub_filter_once off;
    }
```

4.3. auth_basic HTTP 认证模块

```
cd /usr/local/nginx/conf
server {
        listen 80;
        server_name www.example.com;
        root /var/www/htdocs;
        index index.html;

        location / {
                try_files $uri $uri/ /index.html;
                auth_basic            "Administrator’s Area";
                auth_basic_user_file  conf/htpasswd;
        }
}
```

使用 htpasswd 生几个密码文件

生成密码文件

```
$ sudo apt-get install apache2-utils

htpasswd -c -d htpasswd user_name
```

提示

必须使用 -d Force CRYPT encryption of the password. 选项,

使用 openssl 生成密码

```
[root@netkiller ~]# openssl passwd
Password:
Verifying - Password:
9/cEBEuF8T/xQ

[root@stage nginx]# openssl passwd 123456
DuWluVqmkZG8A

[root@stage nginx]# echo "admin:DuWluVqmkZG8A" > htpasswd
```

4.4. valid_referers

例 1.4. Example: valid_referers

```
location /photos/ {
  valid_referers none blocked www.mydomain.com mydomain.com;

  if ($invalid_referer) {
    return   403;
  }
}
```

```
location ~* \.(gif|jpg|jpeg|png|bmp|txt|zip|jar|swf)$ {
        valid_referers none blocked *.mydomain.com;
        if ($invalid_referer) {
                rewrite ^/
http://www.mydomain.com/default.gif;
                #return 403;
        }

}

location /images/ {
        alias /www/images/;
        valid_referers none blocked *.mydomain.com;
        if ($invalid_referer) {
                rewrite ^/
http://www.mydomain.com/default.gif;
        }
}
```

4.5. ngx_http_flv_module

```
location ~ \.flv$ {
    flv;
}
```

4.6. ngx_http_mp4_module

```
location /video/ {
    mp4;
    mp4_buffer_size       1m;
    mp4_max_buffer_size   5m;
    mp4_limit_rate        on;
    mp4_limit_rate_after  30s;
}
```

4.7. limit_zone

```
limit_zone   one  $binary_remote_addr  10m;

server {
        location /download/ {
        limit_conn   one  1;
}
```

4.8. image_filter

```
image_filter 配置项:
image_filter off;       在所在location关闭模块处理。
image_filter test;    确保应答是JPEG，GIF或PNG格式的图像。否则错误
415 (Unsupported Media Type) 将被返回。
image_filter size;    以JSON格式返回图像信息。
image_filter rotate 90 | 180 | 270;    将图像逆时针旋转指定角度。 参
数的值可以包含变量。 可以单独使用，或与 resize 和 crop 变换同时使用.
image_filter resize width height;    按比例缩小图像至指定大小。 如果
想只指定其中一维，另一维可以指定为: “-”。 如果有错误发生，服务器会返回
415 (Unsupported Media Type). 参数的值可以包含变量。 当与 rotate 参
数同时使用时，旋转发生在缩放 之后。
```

image_filter crop width height;   按比例以图像的最短边为准对图像大小进行缩小，然后裁剪另一边多出来的部分。 如果想只指定其中一维，另一维可以指定为： “-”。 如果有错误发生，服务器会返回 415 (Unsupported Media Type). 参数的值可以包含变量。 当与 rotate 参数同时使用时，旋转发生在裁剪 之前。

image_filter_buffer 配置项：
image_filter_buffer size; 例如 image_filter_buffer 1M; 设置用来读图像的缓冲区的最大值。 若图像超过这个大小，服务器会返回 415 (Unsupported Media Type).
image_filter_jpeg_quality quality; 例如
image_filter_jpeg_quality 75;设置变换后的JPEG图像的 质量 。 可配置值： 1 ~ 100 。 更小的值意味着更差的图像质量以及更少需要传输的数据。 推荐的最大值是95. 参数的值可以包含变量。
image_filter_sharpen percent; image_filter_sharpen 0;   增加最终图像的锐度。 锐度百分比可以超过100. 0为关闭锐化。 参数的值可以包含变量。
image_filter_transparency on|off; image_filter_transparency on; 定义当对PNG，或者GIF图像进行颜色变换时是否需要保留透明度。 损失透明度有可能可以获得更高的图像质量。 PNG图像中的alpha通道的透明度默认会一直被保留。

比如所有的图片并修改尺寸为 800x600

```
       location ~* \.(jpg|gif|png)$ {
               image_filter resize 800 600;
       }
```

匹配images目录所有图片并修改尺寸为1920x1080

```
       location ~* /images/.*\.(jpg|gif|png)$ {
               image_filter resize 1920 1080;
       }
```

再比如用url来指定

```
location ~* (.*\.(jpg|gif|png))!(.*)x(.*)$ {
    set $width      $3;
    set $height     $4;
        rewrite "(.*\.(jpg|gif|png))(.*)$" $1;
}
```

```
location ~* .*\.(jpg|gif|png)$ {
        image_filter resize $width $height;
}

location ~* /images/(.+)_(d+)x(d+).(jpg|gif|png)$ {
    set $height $2;
    set $width $3;
    if ($height = "0") {
        rewrite /images/(.+)_(d+)x(d+).(jpg|gif|png)$
/images/$1.$4 last;
    }
    if ($width = "0") {
        rewrite /images/(.+)_(d+)x(d+).(jpg|gif|png)$
/images/$1.$4 last;
    }

    #根据给定的长宽生成缩略图
    image_filter resize $height $width;

    #原图最大2M，要裁剪的图片超过2M返回415错误，根据你的需求调节参数
image_filter_buffer
    image_filter_buffer 2M;

    #error_page  415                    /images/404.jpg;
    try_files /images/$1.$4     /images/404.jpg;
}

location ~* /images {

}

location ~* ^/images/resize/([\d\-]+)_([\d\-]+)/(.+) {
    alias /www/example.com/img.example.com/$3;
    image_filter test;
    image_filter resize $1 $2;
    image_filter_buffer 2M;
    image_filter_jpeg_quality 95;
    image_filter_sharpen 90;
    expires 60d;
}
```

4.9. ngx_stream_proxy_module

ngx_stream_proxy_module 用法与 ngx_http_proxy_module 及其相似, 前者用于tcp代理或负载均衡。后者只能用于 http 的代理

注意模块的proxy_pass指令只能在server段使用, 提供域名或ip地址和端口转发，协议可以是tcp，也可以是udp。

```
server {
    listen 127.0.0.1:80;
    proxy_pass 127.0.0.1:8080;
}

server {
    listen 25;
    proxy_connect_timeout 1s;
    proxy_timeout 1m;
    proxy_pass mail.example.com:25;
}

server {
    listen 53 udp;
    proxy_responses 1;
    proxy_timeout 20s;
    proxy_pass dns.example.com:53;
}

server {
    listen [::1]:8000;
    proxy_pass unix:/tmp/stream.socket;
}
```

4.10. ngx_http_mirror_module

```
location / {
    mirror /mirror;
    proxy_pass http://backend;
}
```

```
location /mirror {
    internal;
    proxy_pass http://test_backend$request_uri;
}
```

4.11. limit_except

```
    location /api/ {

        limit_except PUT DELETE {
            proxy_pass http://127.0.0.1:9080;
        }
    }
```

4.12. geoip_country_code

```
location /google {
    if ( $geoip_country_code ~ (RU|CN) ) {
        proxy_pass http://www.google.hk;
    }
}
```

5. Example

5.1. Nginx + Tomcat

例 1.5. Nginx + Tomcat

```
server {
    listen       80;
    server_name  www.example.com;

    charset utf-8;
    access_log  /var/log/nginx/www.example.com.access.log;

    location / {
            proxy_pass http://127.0.0.1:8080;
        proxy_set_header    Host    $host;
        proxy_set_header    X-Real-IP   $remote_addr;
        proxy_set_header    X-Forwarded-For
$proxy_add_x_forwarded_for;
    }

    #error_page  404              /404.html;

    # redirect server error pages to the static page /50x.html
    #
    error_page   500 502 503 504  /50x.html;
    location = /50x.html {
        root   /usr/share/nginx/html;
    }

    location ~ ^/WEB-INF/ {
        deny  all;
    }

    location ~ \.(html|js|css|jpg|png|gif|swf)$ {
        root /www/example.com/www.example.com;
        expires 1d;
    }
    location ~ \.(ico|fla|flv|mp3|mp4|wma|wmv|exe)$ {
        root /www/example.com/www.example.com;
        expires 7d;
    }
    location ~ \.flv {
        flv;
    }

    location ~ \.mp4$ {
        mp4;
    }

}
```

5.2. 拦截index.html

背景：网站推广审核需要隐藏或不现实首页，其他页面正常

需求：要求访问首页事显示指定页面

```

server {
    listen       80;
    server_name any.netkiller.cn;

    charset utf-8;
    access_log  /var/log/nginx/any.netkiller.cn.access.log;
    error_log  /var/log/nginx/any.netkiller.cn.error.log;

    location /index.html {
                ssi on;
                proxy_set_header Accept-Encoding "";
        proxy_pass http://172.16.0.1/www/temp.html;
        proxy_set_header Host www.netkiller.cn;

    }

    location / {
                ssi on;
                rewrite ^/$ /zt/your.html;

                proxy_set_header Accept-Encoding "";
                proxy_pass http://127.0.0.1:8080;
        proxy_set_header    Host    $host;
        proxy_set_header    X-Real-IP   $remote_addr;
        proxy_set_header    X-Forwarded-For
$proxy_add_x_forwarded_for;
    }

    error_page  404              /error/404.html;
    error_page  403              /error/403.html;
    error_page  502              /error/502.html;
    error_page  500 502 503 504  /error/500.html;

    location ~ ^/WEB-INF/ {
        deny  all;
    }

    location ~ \.(html|js|css|jpg|png|gif|swf)$ {
        root /www/netkiller.cn/www.netkiller.cn;
        expires 1d;
    }
    location ~ \.(ico|fla|flv|mp3|mp4|wma|wmv|exe)$ {
        root /www/netkiller.cn/www.netkiller.cn;
        flv;
        mp4;
        expires 7d;
    }
    location /zt {
                root /www/netkiller.cn/www.netkiller.cn;
                rewrite ^(.*)\;jsessionid=(.*)$ $1 break;
                expires 1d;
    }
    location ^~ /zt/other/ {
                ssi on;
        proxy_set_header Accept-Encoding "";
        proxy_pass http://172.16.0.1/www/;
        proxy_set_header Host www.netkiller.cn;
                proxy_cache www;
                proxy_cache_valid  200 302  1m;
    }

    location /module {
        root /www/netkiller.cn/www.netkiller.cn;
    }
}
```

5.3. Session 的 Cookie 域处理

环境

```
User -> Http2 CDN -> Http2 Nginx -> proxy_pass 1.1 -> Tomcat
```

背景，默认情况下 tomcat 不会主动推送 Cookie 域，例如下面的HTTP头

```
Set-Cookie:
JSESSIONID=8542E9F58C71937B3ABC97F002CE039F;path=/;HttpOnly
```

这样带来一个问题，在浏览器中默认Cookie域等于 HTTP_HOST 头（www.example.com），如果网站只有一个域名没有问题，如果想共享Cookie给子域名下所有域名 *.example.com 无法显示。

通过配置Tomcat sessionCookieDomain="example.com" 可以实现推送 Cookie 域

```
<Context path="" docBase="/www/netkiller.cn/www.netkiller.cn"
reloadable="false" sessionCookieName="PHPSESSID"
sessionCookieDomain="netkiller.cn" sessionCookiePath="/" />
```

这样的配置一般用户的需求都可以满足。我的需求中还有一项，在服务器绑定多个域名（二级域名）。问题来了 Tomcat 将始终推送 netkiller.cn 这个域。其他域名无法正确设置Cookie

```
$ curl -s -I -H https://www.netkiller.cn/index.jsp | grep Set-
Cookie
```

```
Set-Cookie:
PHPSESSID=4DBAF36AA7B79CE1ACBA8DD67702B945;domain=netkiller.cn;
path=/;HttpOnly

$ curl -s -I -H 'Host: www.test.com'
https://www.test.com/index.jsp | grep Set-Cookie
Set-Cookie:
PHPSESSID=4DBAF36AA7B79CE1ACBA8DD67702B945;domain=netkiller.cn;
path=/;HttpOnly

$ curl -s -I -H 'Host: www.example.com'
https://www.example.com/index.jsp | grep Set-Cookie
Set-Cookie:
PHPSESSID=4DBAF36AA7B79CE1ACBA8DD67702B945;domain=netkiller.cn;
path=/;HttpOnly
```

怎样处理需求呢，我想了两个方案，一个方案是在Nginx中配置，另一个方案是在代码中解决。其中Nginx处理起来比较灵活无需开发测试介入，最终选择nginx方案

```
server {
        listen       443 ssl http2 default_server;
        server_name _;
    location ~ \.(do|jsp|action)$ {

        ssi on;
            proxy_set_header Accept-Encoding "";
            proxy_pass http://127.0.0.1:8080;
        proxy_set_header    Host    $host;
        proxy_set_header    X-Real-IP   $remote_addr;
        proxy_set_header    X-Forwarded-For
$proxy_add_x_forwarded_for;

        set $domain $host;
            if ($host ~* ^([^\.]+)\.([^\.]+)\.([^\.]+)$) {
                set $domain $2.$3;
            }
            proxy_cookie_domain netkiller.cn $domain;
    }
}
```

server_name _; 接受任何域名绑定，default_server 将vhost 设置为默认主机。最终测试结果：

```
$ curl -s -I -H https://www.netkiller.cn/index.jsp | grep Set-
Cookie
Set-Cookie:
PHPSESSID=4DBAF36AA7B79CE1ACBA8DD67702B945;domain=netkiller.cn;
path=/;HttpOnly

$ curl -s -I -H https://www.example.com/index.jsp | grep Set-
Cookie
Set-Cookie:
PHPSESSID=4DBAF36AA7B79CE1ACBA8DD67702B945;domain=example.com;p
ath=/;HttpOnly

$ curl -s -I -H https://www.domain.com/index.jsp | grep Set-
Cookie
Set-Cookie:
PHPSESSID=4DBAF36AA7B79CE1ACBA8DD67702B945;domain=domain.com;pa
th=/;HttpOnly
```

6. FAQ

6.1. 405 Not Allowed?

6.1.1. 405 Not Allowed?

静态页面POST会提示405 Not Allowed错误.

```
# curl -d name=neo http://www.mydoamin.com/index.html
<html>
<head><title>405 Not Allowed</title></head>
<body bgcolor="white">
<center><h1>405 Not Allowed</h1></center>
<hr><center>nginx</center>
</body>
</html>
```

```
server {
    listen       80 default;
    server_name  myid.mydomain.com;

    charset utf-8;
    access_log
/var/log/nginx/myid.mydomain.com.access.log  main;

    if ($http_user_agent ~* ^$){
      return 412;
    }
    ###########################

    location / {
        root   /www/mydomain.com/myid.mydomain.com;
        index  index.html index.php;
        #error_page 405 =200 $request_filename;
    }

    #error_page  404              /404.html;
    #
    error_page 405 =200 @405;
    location @405 {
        #proxy_set_header  Host            $host;
        proxy_method GET;
        proxy_pass http://myid.mydomain.com;

    }


    # redirect server error pages to the static page
/50x.html
    #
    error_page   500 502 503 504  /50x.html;
    location = /50x.html {
        root   /usr/share/nginx/html;
    }
}
```

6.2. 413 Request Entity Too Large

上传文件大小限制

6.2.1. 413 Request Entity Too Large

error.log 提示:

client intended to send too large body

client_max_body_size 8m;

修改 /etc/nginx/nginx.conf 文件。

```
http {
    include       /etc/nginx/mime.types;
    default_type  application/octet-stream;

    log_format  main  '$remote_addr - $remote_user
[$time_local] "$request" '
                      '$status $body_bytes_sent
"$http_referer" '
                      '"$http_user_agent"
"$http_x_forwarded_for"';

    access_log  /var/log/nginx/access.log  main;

    sendfile        on;
    #tcp_nopush     on;

    keepalive_timeout  65;

    server_tokens off;
    gzip  on;
    gzip_min_length 1k;
    gzip_types text/plain text/html text/css
application/javascript text/javascript application/x-
javascript text/xml application/xml
application/xml+rss application/json;
    gzip_vary on;

    client_max_body_size 8m;

    include /etc/nginx/conf.d/*.conf;
}
```

6.3. 499 Client Closed Request

6.3.1. Nginx access.log 日志显示

111.85.11.15 - - [25/Jun/2016:19:20:35 +0800] "GET /xxx/xxx/xxx.jsp HTTP/1.1" 499 88 "-" "Mozilla/5.0

(Macintosh; Intel Mac OS X 10_10_3) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/43.0.2357.130 Safari/537.36 JianKongBao Monitor 1.1"

配置 proxy_ignore_client_abort on;

```
    location / {

                ssi on;
                proxy_set_header Accept-Encoding "";
                proxy_pass http://127.0.0.1:8080;
        proxy_set_header    Host    $host;
        proxy_set_header    X-Real-IP   $remote_addr;
        proxy_set_header    X-Forwarded-For
$proxy_add_x_forwarded_for;
                proxy_ignore_client_abort  on;
    }
```

6.4. 502 Bad Gateway?

6.4.1. 502 Bad Gateway

error.log 提示:

upstream sent too big header while reading response header from upstream?

修改fastcgi配置

```
location ~ \.php$ {

        fastcgi_buffers 8 16k;
        fastcgi_buffer_size 32k;
        。。。
        。。。
}
```

6.5. 504 Gateway Time-out

6.5.1. 504 Gateway Time-out

问题一般出现在 nginx 通过 proxy_pass 代理其他服务的场景

由于代理的后端服务处理时间较长，超过了timeout阀值，就会报出 504 错误

修改fastcgi配置

```
client_max_body_size      50m; //文件大小限制，默认1m
client_header_timeout     1m;
client_body_timeout       1m;
proxy_connect_timeout      60s;
proxy_read_timeout       1m;
proxy_send_timeout       1m;

每个参数的意思：

client_max_body_size     限制上传数据的的大小，若超过所设定的大小，返回413错误。
client_header_timeout    读取请求头的超时时间，若超过所设定的大小，返回408错误。
client_body_timeout              读取请求实体的超时时间，若超过所设定的大小，返回413错误。
proxy_connect_timeout    此参数为等待的最长时间，默认为60秒，官方推荐最长不要超过75秒。
proxy_read_timeout               http请求代理后，nginx会等待处理结果。此参数即为服务器响应时间，默认60秒。
proxy_send_timeout               http请求被服务器处理完
```

```
后，把数据传返回给Nginx的用时，默认60秒。
```

6.6. proxy_pass

```
nginx: [emerg] "proxy_pass" cannot have URI part in location
given by regular expression, or inside named location, or
inside "if" statement, or inside "limit_except" block in
/etc/nginx/conf.d/www.mydomain.com.conf:25
nginx: configuration file /etc/nginx/nginx.conf test failed
```

在location,if中使用证则匹配proxy_pass末尾不能写/

```
        if ($request_uri ~*
"^/info/{cn|tw}/{news|info}/\d\.html") {
                proxy_pass http://info.example.com/;
                break;
        }

    location ~ ^/info/ {
                proxy_pass http://info.example.com/;
                break;
    }
```

proxy_pass http://info.example.com/; 改为 proxy_pass http://info.example.com; 可以解决

6.7. proxy_pass SESSION 丢失问题

如果用户Cookie信息没有经过 proxy_pass 传递给最终服务器，SESSION信息将丢失，解决方案

```
proxy_set_header   Cookie $http_cookie;
```

6.8. [alert] 55785#0: *11449 socket() failed (24: Too many open files) while connecting to upstream

配置 worker_rlimit_nofile 参数即可

```
user  nginx;
worker_processes  8;
worker_rlimit_nofile 65530;
```

配置 ulimit 也能达到同样效果，但我更喜欢 worker_rlimit_nofile 因为它仅仅作用于nginx,而不是全局配置。

6.9. server_name 与 SSI 注意事项

```
server_name www.example.com www.example.net www.example.org;
```

下来SSI标签无论你使用那个域名访问，输出永远是server_name 的第一域名www.example.com

```
<!--#echo var="SERVER_NAME"-->
```

需要通过SERVER_NAME判定展示不同结果时需要注意。

6.10. location 跨 document_root 引用，引用 document_root 之外的资源

下面的例子是 Document root 是 /www/netkiller.com/m.netkiller.com， 我们需要 /www/netkiller.com/www.netkiller.com 中的资源。

```
server {
    listen       80;
    server_name  m.netkiller.com;

    charset utf-8;
    access_log  /var/log/nginx/m.netkiller.com.access.log;
    error_log  /var/log/nginx/m.netkiller.com.error.log;

    location / {
                root /www/netkiller.com/m.netkiller.com;
                index.html
    }

    location /module {
        root /www/netkiller.com/www.netkiller.com;
    }

}
```

```
server {
    listen       80;
    server_name  m.netkiller.com;

    charset utf-8;
    access_log  /var/log/nginx/m.netkiller.com.access.log;
    error_log  /var/log/nginx/m.netkiller.com.error.log;

    location / {
                root /www/netkiller.com/m.netkiller.com;
                index.html
    }

    location ^~ /module/ {
        root /www/netkiller.com/www.netkiller.com;
    }

}
```

```
```

上面的例子location /module 是指 /www/netkiller.com/www.netkiller.com + /module，如果 /www/netkiller.com/www.netkiller.com 目录下面没有 module 目录是出现404， error.log显示 "/www/netkiller.cn/www.netkiller.cn/module/index.html" failed (2: No such file or directory)

6.11. nginx: [warn] duplicate MIME type "text/html" in /etc/nginx/nginx.conf

text/html 是 gzip_types 默认值，所以不要将text/html加入到 gzip_types列表内

6.12. 127.0.0.1:8080 failed

链接本地端口失败，已经关闭防火墙，同时使用 curl http://127.0.0.1:8080 一切正常

日志片段

```
2018/09/07 12:31:27 [crit] 10202#10202: *4 connect() to
[::1]:8080 failed (13: Permission denied) while connecting to
upstream, client: 47.90.97.183, server: www.api.netkiller.cn,
request: "GET /api/ HTTP/2.0", upstream:
"http://[::1]:8080/api/", host: "api.netkiller.cn"
2018/09/07 12:31:27 [warn] 10202#10202: *4 upstream server
temporarily disabled while connecting to upstream, client:
47.90.97.183, server: www.api.netkiller.cn, request: "GET /api/
HTTP/2.0", upstream: "http://[::1]:8080/api/", host:
"api.netkiller.cn"
2018/09/07 12:31:27 [crit] 10202#10202: *4 connect() to
127.0.0.1:8080 failed (13: Permission denied) while connecting
to upstream, client: 47.90.97.183, server:
www.api.netkiller.cn, request: "GET /api/ HTTP/2.0", upstream:
"http://127.0.0.1:8080/api/", host: "api.netkiller.cn"
```

```
2018/09/07 12:31:27 [warn] 10202#10202: *4 upstream server
temporarily disabled while connecting to upstream, client:
47.90.97.183, server: www.api.netkiller.cn, request: "GET /api/
HTTP/2.0", upstream: "http://127.0.0.1:8080/api/", host:
"api.netkiller.cn"
```

问题出现在 AWS 亚马逊云主机。经过筛查发现是 SELINUX 问题

```
[root@netkiller ~]# cat /var/log/audit/audit.log | grep nginx |
grep denied | more

type=AVC msg=audit(1536320093.274:345): avc:  denied  {
sys_resource } for  pid=9544 comm="nginx" capability=24
scontext=system_u:system_r:httpd_t:s0
tcontext=system_u:system_r:httpd_t:s0 tclass=capabi
lity
type=AVC msg=audit(1536320093.274:346): avc:  denied  {
sys_resource } for  pid=9545 comm="nginx" capability=24
scontext=system_u:system_r:httpd_t:s0
tcontext=system_u:system_r:httpd_t:s0 tclass=capabi
lity
type=AVC msg=audit(1536320093.275:347): avc:  denied  {
sys_resource } for  pid=9546 comm="nginx" capability=24
scontext=system_u:system_r:httpd_t:s0
tcontext=system_u:system_r:httpd_t:s0 tclass=capabi
lity
type=AVC msg=audit(1536321850.706:459): avc:  denied  {
sys_resource } for  pid=9798 comm="nginx" capability=24
scontext=system_u:system_r:httpd_t:s0
tcontext=system_u:system_r:httpd_t:s0 tclass=capabi
lity
type=AVC msg=audit(1536321850.707:460): avc:  denied  {
sys_resource } for  pid=9799 comm="nginx" capability=24
scontext=system_u:system_r:httpd_t:s0
tcontext=system_u:system_r:httpd_t:s0 tclass=capabi
lity
type=AVC msg=audit(1536321920.108:461): avc:  denied  {
name_connect } for  pid=9796 comm="nginx" dest=8080
scontext=system_u:system_r:httpd_t:s0
tcontext=system_u:object_r:http_cache_port_t:s0 tclass=t
```

```
cp_socket
type=AVC msg=audit(1536321920.109:462): avc:  denied  {
name_connect } for  pid=9796 comm="nginx" dest=8080
scontext=system_u:system_r:httpd_t:s0
tcontext=system_u:object_r:http_cache_port_t:s0 tclass=t
cp_socket
```

6.13. failed (13: Permission denied) while connecting to upstream

问题分析，此问题出在 SELINUX

```
2021/07/13 02:18:52 [crit] 6671#0: *3 connect() to
192.168.60.7:8000 failed (13: Permission denied) while
connecting to upstream, client: 192.168.90.137, server:
www.netkiller.cn, request: "GET / HTTP/2.0", upstream:
"http://192.168.60.7:8000/", host: "www.netkiller.cn"
```

查看 SELINUX 设置

```
[root@localhost ~]# getsebool -a | grep httpd
httpd_anon_write --> off
httpd_builtin_scripting --> on
httpd_can_check_spam --> off
httpd_can_connect_ftp --> off
httpd_can_connect_ldap --> off
httpd_can_connect_mythtv --> off
httpd_can_connect_zabbix --> off
httpd_can_network_connect --> on
httpd_can_network_connect_cobbler --> off
httpd_can_network_connect_db --> off
httpd_can_network_memcache --> off
httpd_can_network_relay --> off
httpd_can_sendmail --> off
httpd_dbus_avahi --> off
```

```
httpd_dbus_sssd --> off
httpd_dontaudit_search_dirs --> off
httpd_enable_cgi --> on
httpd_enable_ftp_server --> off
httpd_enable_homedirs --> off
httpd_execmem --> off
httpd_graceful_shutdown --> off
httpd_manage_ipa --> off
httpd_mod_auth_ntlm_winbind --> off
httpd_mod_auth_pam --> off
httpd_read_user_content --> off
httpd_run_ipa --> off
httpd_run_preupgrade --> off
httpd_run_stickshift --> off
httpd_serve_cobbler_files --> off
httpd_setrlimit --> off
httpd_ssi_exec --> off
httpd_sys_script_anon_write --> off
httpd_tmp_exec --> off
httpd_tty_comm --> off
httpd_unified --> off
httpd_use_cifs --> off
httpd_use_fusefs --> off
httpd_use_gpg --> off
httpd_use_nfs --> off
httpd_use_opencryptoki --> off
httpd_use_openstack --> off
httpd_use_sasl --> off
httpd_verify_dns --> off
```

设置此选项可以解决

```
[root@localhost ~]# setsebool -P httpd_can_network_connect true
```

第 2 章 Caddy

Caddy is a powerful, enterprise-ready, open source web server with automatic HTTPS written in Go.

1. 安装 Caddy

2. 启动 Caddy

前台运行

```
caddy run --config /etc/caddy/Caddyfile --adapter caddyfile
```

2.1. 开启 QUIC

```
caddy run --config /etc/caddy/Caddyfile --adapter caddyfile --
quic
```

3. 命令行

3.1. 文件服务器

将当前目录作为文件服务器的根目录

```
$ caddy file-server
```

指定端口

```
$ caddy file-server --listen :8080
```

不打开 index.html，显示文件目录

```
$ caddy file-server --browse
```

指定文件服务器根目录

```
$ caddy file-server --root ~/public_html
```

4. /etc/caddy/Caddyfile

https://caddyserver.com/docs/caddyfile

4.1. 监听地址

```
localhost
example.com
:443
http://example.com
localhost:8080
127.0.0.1
[::1]:2015
example.com/foo/*
*.example.com
http://
```

```
localhost:8080, example.com, www.example.com
```

泛解析

```
*.example.com
```

4.2. 反向代理

```
http://api.netkiller.cn {
    reverse_proxy /* http://192.168.30.10:8080
    tls netkiller@msn.com
}
```

推送 X-Forwarded-For 头

```
http://www.netkiller.cn {

    root * /opt/netkiller.cn/www.netkiller.cn
    file_server

    reverse_proxy /api/* 192.168.30.10:8080 {
            header_up X-Real-IP {http.request.remote.host}
            header_up X-Forwarded-For
{http.request.remote.host}
        }

}
```

反向代理自签名证书，添加 tls_insecure_skip_verify 配置项

```
netkiller.cn {
    reverse_proxy * {
        to https://192.168.0.10
        transport http {
            tls_insecure_skip_verify
        }
    }
}

api.netkiller.cn {
    reverse_proxy * {
        to 192.168.10.10:443
        transport http {
            tls
            tls_insecure_skip_verify
```

```
		}
	}
}

```

反向代理URL前缀问题

```

举例:
www.netkiller.cn {
	reverse_proxy /api/* http://api.netkiller.cn:8080
}
访问URL:
http://www.netkiller.cn/api/adduser

实际访问的URL是:
http://api.netkiller.cn:8080/api/adduser

我们需要的URL是:
http://api.netkiller.cn:8080/adduser

```

解决方案

```

www.netkiller.cn {
	route /api* {
	uri strip_prefix /api
		reverse_proxy api.netkiller.cn:8088
	}
}

```

第 3 章 Apache Tomcat

1. Tomcat 安装与配置

1.1. Tomcat 6

解压安装

```
chmod +x jdk-6u1-linux-i586.bin
./jdk-6u1-linux-i586.bin
输入"yes"回车

mv jdk1.6.0_01 /usr/local/
ln -s /usr/local/jdk1.6.0_01/ /usr/local/java
```

/etc/profile.d/java.sh

例 3.1. /etc/profile.d/java.sh

```
################################################
### Java environment
################################################
export JAVA_HOME=/usr/local/java
export JRE_HOME=/usr/local/java/jre
export PATH=$PATH:/usr/local/java/bin:/usr/local/java/jre/bin
export
CLASSPATH="./:/usr/local/java/lib:/usr/local/java/jre/lib:/usr/
local/memcached/api/java"
export JAVA_OPTS="-Xms512m -Xmx1024m"
```

下载binary解压到/usr/local/

下载软件包

```
wget http://archive.apache.org/dist/tomcat/tomcat-
6/v6.0.13/bin/apache-tomcat-6.0.13.tar.gz
wget http://archive.apache.org/dist/tomcat/tomcat-
connectors/native/tomcat-native-1.1.10-src.tar.gz
wget http://archive.apache.org/dist/tomcat/tomcat-
connectors/jk/source/jk-1.2.23/tomcat-connectors-1.2.23-
src.tar.gz
```

```
tar zxvf apache-tomcat-6.0.13.tar.gz
mv apache-tomcat-6.0.13 /usr/local/
ln -s /usr/local/apache-tomcat-6.0.13/ /usr/local/tomcat
```

tomcat-native

```
tar zxvf tomcat-native-1.1.10-src.tar.gz
cd tomcat-native-1.1.10-src/jni/native
./configure --with-apr=/usr/local/apache/bin/apr-1-config --
with-java-home=/usr/local/java/
make
make install
```

catalina.sh

```
CATALINA_OPTS="-Djava.library.path=/usr/local/apr/lib"
JAVA_OPTS="-Xss128k -Xms128m -Xmx1024m -XX:PermSize=128M -
XX:MaxPermSize=256m -XX:MaxNewSize=256m"
```

启动

```
startup.sh
```

tomcat-native

```
cd /usr/local/tomcat-6.0.18/bin
tar zxvf tomcat-native.tar.gz
cd tomcat-native-1.1.14-src/jni/native
./configure --with-apr=/usr/local/apr --with-java-
home=/usr/java/jdk1.6.0_11
make && make install
```

启动脚本

例 3.2. /etc/init.d/tomcat

```
# cat /etc/init.d/tomcat
#!/bin/bash
# description: Tomcat Start Stop Restart
# processname: tomcat
# chkconfig: 234 20 80

JAVA_HOME=/srv/java
CATALINA_HOME=/srv/apache-tomcat

# Source function library.
. /etc/init.d/functions

# Source networking configuration.
. /etc/sysconfig/network

if [ -f /etc/sysconfig/tomcat ]; then
        . /etc/sysconfig/tomcat
fi

prog=tomcat
lockfile=/var/lock/subsys/$prog
pidfile=${PIDFILE-/var/run/$prog.pid}
lockfile=${LOCKFILE-/var/lock/subsys/$prog}
RETVAL=0
OPTIONS="--pidfile=${pidfile}"
```

```
start(){
        # Start daemons.
        echo -n $"Starting $prog: "
        #daemon $prog $OPTIONS
        $CATALINA_HOME/bin/startup.sh
        RETVAL=$?
        echo
        [ $RETVAL -eq 0 ] && touch $lockfile
        return $RETVAL

}

stop() {
        echo -n $"Stopping $prog: "
#       killproc -p ${pidfile} -d 10 $httpd
        $CATALINA_HOME/bin/shutdown.sh
        RETVAL=$?
        echo
        [ $RETVAL = 0 ] && rm -f ${lockfile} ${pidfile}
}

case $1 in
    start)
        start
        ;;
    stop)
        stop
        ;;
    restart)
        start
        stop
    ;;
esac
exit 0
```

创建 /etc/init.d/tomcat 文件，复制并粘贴上面的启动脚本

```
vim /etc/init.d/tomcat
chmod +x /etc/init.d/tomcat
chkconfig --add tomcat
chkconfig --level 234 tomcat on
chkconfig --list tomcat
```

1.2. Tomcat 7

Server JRE

安装 Server JRE

```
cd /usr/local/src/

tar zxvf server-jre-7u21-linux-x64.gz
mv jdk1.7.0_21 /srv/
ln -s /srv/jdk1.7.0_21 /srv/java
```

或者

```
curl -sS
https://raw.github.com/netkiller/shell/master/java/server-
jre.sh | bash
```

Tomcat

安装下面步骤安装Tomcat，注意不要使用root启动tomcat。这里使用www用户启动tomcat,这样的目的是让tomcat进程继承www用户权限。

```


cd /usr/local/src/
wget
http://ftp.cuhk.edu.hk/pub/packages/apache.org/tomcat/tomcat-
7/v7.0.40/bin/apache-tomcat-7.0.40.tar.gz
tar zxvf apache-tomcat-7.0.40.tar.gz

mv apache-tomcat-7.0.40 /srv/
```

```
ln -s /srv/apache-tomcat-7.0.40 /srv/apache-tomcat
rm -rf /srv/apache-tomcat/webapps/*

cat > /srv/apache-tomcat/bin/setenv.sh <<'EOF'
export JAVA_HOME=/srv/java
export JAVA_OPTS="-server -Xms512m -Xmx8192m  -XX:PermSize=64M
-XX:MaxPermSize=512m"
export CATALINA_HOME=/srv/apache-tomcat
export
CLASSPATH=$JAVA_HOME/lib:$JAVA_HOME/jre/lib:$CATALINA_HOME/lib:
export
PATH=$PATH:$JAVA_HOME/bin:$JAVA_HOME/jre/bin:$CATALINA_HOME/bin
:
EOF

cp /srv/apache-tomcat/conf/server.xml{,.original}

groupadd -g 80 www
adduser -o --home /srv --uid 80 --gid 80 -c "Web Application"
www

chown www:www -R /srv/*

su - www -c "/srv/apache-tomcat/bin/startup.sh"
```

或者运行下面脚本快速安装

```
curl -sS
https://raw.github.com/netkiller/shell/master/apache/tomcat/ins
tall.sh | bash
```

1.3. Java 8 + Tomcat 8

安装Java 8

```
cd /usr/local/src/
```

```
tar zxf server-jre-8u20-linux-x64.gz
mv jdk1.8.0_20 /srv/
ln -s /srv/jdk1.8.0_20 /srv/java

cat >> /etc/profile.d/java.sh <<'EOF'
export JAVA_HOME=/srv/java
export JAVA_OPTS="-server -Xms512m -Xmx8192m"
export
CLASSPATH=$JAVA_HOME/lib:$JAVA_HOME/jre/lib:$CATALINA_HOME/lib:
export
PATH=$PATH:$JAVA_HOME/bin:$JAVA_HOME/jre/bin:$CATALINA_HOME/bin
:
EOF
```

注意

Java 8 取消了 PermSize 与 MaxPermSize 配置项"

```
cd /usr/local/src/
wget
http://ftp.cuhk.edu.hk/pub/packages/apache.org/tomcat/tomcat-
8/v8.0.12/bin/apache-tomcat-8.0.12.tar.gz
tar zxf apache-tomcat-8.0.12.tar.gz

mv apache-tomcat-8.0.12 /srv/
ln -s /srv/apache-tomcat-8.0.12 /srv/apache-tomcat
rm -rf /srv/apache-tomcat/webapps/*
cp /srv/apache-tomcat/conf/server.xml{,.original}

cat > /srv/apache-tomcat/bin/setenv.sh <<'EOF'
export JAVA_HOME=/srv/java
export JAVA_OPTS="-server -Xms512m -Xmx8192m"
export CATALINA_HOME=/srv/apache-tomcat
export
CLASSPATH=$JAVA_HOME/lib:$JAVA_HOME/jre/lib:$CATALINA_HOME/lib:
/srv/IngrianJCE/lib/ext/IngrianNAE-5.1.1.jar
export
PATH=$PATH:$JAVA_HOME/bin:$JAVA_HOME/jre/bin:$CATALINA_HOME/bin
:
EOF
```

启动 Tomcat

```
groupadd -g 80 www
adduser -o --home /www --uid 80 --gid 80 -c "Web Application" 
www

chown www:www -R /srv/apache-tomcat-*

su - www -c "/srv/apache-tomcat/bin/startup.sh"
```

systemctl 启动脚本

```
curl -s 
https://raw.githubusercontent.com/oscm/shell/master/web/tomcat/
systemctl.sh | bash
```

Session 共享

```
$ git clone https://github.com/chexagon/redis-session-
manager.git
$ cd redis-session-manager/
$ mvn package
$ ls target/redis-session-manager-with-dependencies-2.1.1-
SNAPSHOT.jar
redis-session-manager-with-dependencies-2.1.1-SNAPSHOT.jar

$ cp target/redis-session-manager-with-dependencies-2.1.1-
SNAPSHOT.jar /srv/apache-tomcat/apache-tomcat-8.5.11/lib/
```

如果Redis是 127.0.0.1 配置 conf/context.xml 加入下面一行，

```
<Manager
className="com.crimsonhexagon.rsm.redisson.SingleServerSessionM
anager" />
```

完整的配置

```
    <Manager
className="com.crimsonhexagon.rsm.redisson.SingleServerSessionM
anager"
            endpoint="localhost:6379"
            sessionKeyPrefix="JSESSIONID::"
            saveOnChange="false"
            forceSaveAfterRequest="false"
            dirtyOnMutation="false"
            ignorePattern=".*\\.
(ico|png|gif|jpg|jpeg|swf|css|js)$"
            connectionPoolSize="100"
            database="16"
            password="yourpassword"
            timeout="60000"
            pingTimeout="1000"
            retryAttempts="20"
            retryInterval="1000"
    />
```

例 3.3. Example /srv/apache-tomcat/conf

```
cat context.xml
<?xml version="1.0" encoding="UTF-8"?>
<!--
  Licensed to the Apache Software Foundation (ASF) under one or
more
  contributor license agreements.  See the NOTICE file
distributed with
```

```
  this work for additional information regarding copyright
ownership.
  The ASF licenses this file to You under the Apache License,
Version 2.0
  (the "License"); you may not use this file except in
compliance with
  the License.  You may obtain a copy of the License at

      http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing,
software
  distributed under the License is distributed on an "AS IS"
BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
or implied.
  See the License for the specific language governing
permissions and
  limitations under the License.
-->
<!-- The contents of this file will be loaded for each web
application -->
<Context>

    <!-- Default set of monitored resources. If one of these
changes, the    -->
    <!-- web application will be reloaded.
-->
    <WatchedResource>WEB-INF/web.xml</WatchedResource>

<WatchedResource>${catalina.base}/conf/web.xml</WatchedResource
>

    <!-- Uncomment this to disable session persistence across
Tomcat restarts -->
    <!--
    <Manager pathname="" />
    -->
    <Manager
className="com.crimsonhexagon.rsm.redisson.SingleServerSessionM
anager"
            endpoint="localhost:6379"
            sessionKeyPrefix="JSESSIONID"
            saveOnChange="false"
            forceSaveAfterRequest="false"
            dirtyOnMutation="false"
            ignorePattern=".*\\.
(ico|png|gif|jpg|jpeg|swf|css|js)$"
            connectionPoolSize="100"
            database="0"
            password=""
            timeout="60000"
            pingTimeout="1000"
            retryAttempts="20"
            retryInterval="1000"
    />
</Context>
```

test session

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<title>set session</title>
</head>
<body>
  <%= session.getId() %>
  <%
    session.setAttribute("neo", "netkiller");
  %>
</body>
</html>
```

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
```

```
<title>get session</title>
</head>
<body>
  <%= session.getId() %>
  <br/>
  <br/>
  <%=(String)session.getAttribute("neo")%>

</body>
</html>
```

SSL 证书上

```
neo@MacBook-Pro-Neo ~ % keytool -genkey -v -alias tomcat -
keyalg RSA -keystore conf/tomcat.keystore -validity 36500
输入密钥库口令:
再次输入新口令:
您的名字与姓氏是什么?
  [Unknown]:  Neo
您的组织单位名称是什么?
  [Unknown]:  SF
您的组织名称是什么?
  [Unknown]:  IT
您所在的城市或区域名称是什么?
  [Unknown]:  SZ
您所在的省/市/自治区名称是什么?
  [Unknown]:  GD
该单位的双字母国家/地区代码是什么?
  [Unknown]:  CN
CN=Neo, OU=SF, O=IT, L=SZ, ST=GD, C=CN是否正确?
  [否]:  Y

正在为以下对象生成 2,048 位RSA密钥对和自签名证书 (SHA256withRSA) (有效
期为 36,500 天):
         CN=Neo, OU=SF, O=IT, L=SZ, ST=GD, C=CN
[正在存储conf/tomcat.keystore]

neo@MacBook-Pro-Neo ~ % ll conf/tomcat.keystore
-rw-r--r--  1 neo  staff   2.6K  7 15 17:06
```

```
conf/tomcat.keystore
```

```
 <Connector port="8443"
protocol="org.apache.coyote.http11.Http11NioProtocol"
        maxThreads="150" SSLEnabled="true">
     <SSLHostConfig>
             <Certificate

certificateKeystoreFile="conf/tomcat.keystore"
                  certificateKeystorePassword="12345678"
                  certificateKeystoreType="PKCS12"
             />
     </SSLHostConfig>
</Connector>
```

1.4. Tomcat 9/10

自签名证书生成命令:

```
keytool -genkey -alias tomcat -keyalg RSA -keystore
/srv/apache-tomcat/conf/localhost-rsa.jks

keytool -genkeypair -alias "tomcat" -keyalg "RSA" -keystore
"https.keystore"
加入有效期: keytool -genkeypair -alias "tomcat" -keyalg "RSA" -
keystore "e:\https.keystore" -validity 36000

# CN为域名
keytool -genkeypair -alias "tomcat" -keyalg "RSA" -keystore
tomcat.keystore -keypass "123456" -storepass "123456" -validity
365000 -dname
"CN=www.netkiller.cn.cn,OU=tomcat,O=tomcat,L=SZ,ST=GD,C="
```

```
        <Connector port="443"
protocol="org.apache.coyote.http11.Http11NioProtocol"
maxThreads="150" SSLEnabled="true" scheme="https"
secure="true">
        <SSLHostConfig>
             <Certificate
certificateKeystoreFile="cert/www.netkiller.cn.pfx"
certificateKeystoreType="PKCS12"
certificateKeystorePassword="12345678" />
        </SSLHostConfig>
    </Connector>
```

```
<Connector port="443"
    protocol="org.apache.coyote.http11.Http11NioProtocol"
    maxThreads="150"
    SSLEnabled="true"
    defaultSSLHostConfigName="www.netkiller.cn" >
        <SSLHostConfig hostName="www.netkiller.cn">
                <Certificate
               certificateKeystoreFile="cert/netkiller.pfx"
               certificateKeystorePassword="****"
               type="RSA"/>
        </SSLHostConfig>
</Connector>
```

1.5. 防火墙配置

```
iptables -A INPUT -m state --state NEW -m tcp -p tcp --dport
8080 -j ACCEPT
```

80 跳转 8080 方案

```
iptables -t nat -A PREROUTING -p tcp --dport 80 -j REDIRECT --
to-port 8080
```

取消跳转

```
iptables -t nat -D PREROUTING -p tcp --dport 80 -j REDIRECT --
to-port 8080
```

查看规则

```
iptables -t nat -L
```

例 3.4. tomcat firewall

下面是完整的例子，仅供参考，复制到 /etc/sysconfig/iptables 文件中，重启iptables即可生效。

```
# cat /etc/sysconfig/iptables
# Generated by iptables-save v1.4.7 on Mon Jul 22 15:58:35 2013
*nat
:PREROUTING ACCEPT [7:847]
:POSTROUTING ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A PREROUTING -p tcp -m tcp --dport 80 -j REDIRECT --to-port
8080
COMMIT
# Completed on Mon Jul 22 15:58:35 2013
# Generated by iptables-save v1.4.7 on Mon Jul 22 15:58:35 2013
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [42303:3464247]
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 22 -j
```

```
ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 8080 -j
ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 80 -j
ACCEPT
-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A FORWARD -j REJECT --reject-with icmp-host-prohibited
COMMIT
# Completed on Mon Jul 22 15:58:35 2013
```

1.6. 同时运行多实例

创建工作目录

```
mkdir /srv/apache-tomcat
```

每个端口一个目录

```
tar zxvf apache-tomcat-7.0.x.tar.gz
mv  apache-tomcat-7.0.x /srv/apache-tomcat/8080

tar zxvf apache-tomcat-7.0.x.tar.gz
mv  apache-tomcat-7.0.x /srv/apache-tomcat/9090
```

修改 Server port="8006" 与 Connector port="9090"端口，不要出现重复。

```
<Server port="8006" shutdown="SHUTDOWN">


 <Connector port="9090" protocol="HTTP/1.1"
               connectionTimeout="20000"
               redirectPort="8443" />
```

```
<!--
    <Connector port="8009" protocol="AJP/1.3"
redirectPort="8443" />
-->
```

启动tomcat然后观察catalina.log日志文件，确认每个进程都正确启动。

1.7. Testing file

创建测试文件

vim webapps/ROOT/index.jsp

```
<%@ page contentType="text/html;charset=utf-8"%>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=utf-8">
<title>helloworld!</title>
</head>

<body>
<h1>
<%="It works!"%>
</h1>
<%
out.println("<h3>Hello World!</h3>");
%>
<hr />
<%=new java.util.Date()%>
</body>
</html>
```

使用curl命令测试，测试结果类似下面结果。

```
$ curl http://192.168.6.9/index.jsp

<html>
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=utf-8">
<title>helloworld!</title>
</head>

<body>
<h1>
It works!
</h1>
<h3>Hello World!</h3>

<hr />
Mon Jul 22 16:41:46 HKT 2013
</body>
</html>
```

1.8. mod_jk

mod_jk 安装

```
tar zxvf tomcat-connectors-1.2.23-src.tar.gz
cd tomcat-connectors-1.2.23-src/native/
./configure --with-apxs=/usr/local/apache/bin/apxs
make
make install
chmod 755 /usr/local/apache/modules/mod_jk.so
```

httpd.conf 尾部加入

```
Include conf/mod_jk.conf
```

配置workers.properties

apache/conf/workers.properties

```
# Define 1 real worker using ajp13
worker.list=worker1
# Set properties for worker1 (ajp13)
worker.worker1.type=ajp13
worker.worker1.host=127.0.0.1
worker.worker1.port=8009
worker.worker1.lbfactor=1
worker.worker1.cachesize=128
worker.worker1.cache_timeout=600
worker.worker1.socket_keepalive=1
worker.worker1.reclycle_timeout=300
```

mod_jk.conf

apache/conf/mod_jk.conf

```
[chenjingfeng@d3010 Includes]$ cat mod_jk.conf
<IfModule mod_jk.c>
# Load mod_jk module
LoadModule jk_module            modules/mod_jk.so
# Where to find workers.properties
JkWorkersFile
/usr/local/apache/conf/workers.properties
# Where to put jk logs
JkLogFile              /usr/local/apache/logs/mod_jk.log
# Set the jk log level [debug/error/info]
JkLogLevel             error
# Select the log format
JkLogStampFormat "[%a %b %d %H:%M:%S %Y] "
# JkOptions indicate to send SSL KEY SIZE,
JkOptions     +ForwardKeySize +ForwardURICompat -
ForwardDirectories
# JkRequestLogFormat set the request format
JkRequestLogFormat     "%w %V %T"
JkShmFile     /usr/local/apache2/logs/mod_jk.shm
# Send jsp,servlet for context * to worker named worker1
```

```
JkMount  /status/* worker1
JkMount  /*.jsp worker1
JkMount  /*.jsps worker1
JkMount  /*.do worker1
JkMount  /*Servlet worker1
JkMount  /jk/* worker1
</IfModule>
```

分别测试apache,tomcat

1.9. mod_proxy_ajp

包含虚拟主机配置文件

# vi conf/httpd.conf

```
# Virtual hosts
Include conf/extra/httpd-vhosts.conf
```

虚拟主机中配置ProxyPass,ProxyPassReverse

# vi conf/extra/httpd-vhosts.conf

```
<VirtualHost *:80>
    ServerName netkiller.8800.org
    ProxyPass /images !
        ProxyPass /css !
        ProxyPass /js !
    ProxyPass /ajp ajp://localhost:8009/ajp
    ProxyPassReverse /ajp ajp://localhost:8009/ajp
</VirtualHost>
```

反向代理和均衡负载模块

```
LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_ajp_module modules/mod_proxy_ajp.so
LoadModule proxy_balancer_module modules/mod_proxy_balancer.so

ProxyPass /admin balancer://tomcatcluster/admin
lbmethod=byrequests stickysession=JSESSIONID nofailover=Off
timeout=5 maxattempts=3
ProxyPassReverse /admin balancer://tomcatcluster/admin

<Proxy balancer://tomcatcluster>
        BalancerMember ajp://localhost:8009 route=web1
        BalancerMember ajp://localhost:10009 smax=10 route=web2
        BalancerMember ajp://localhost:11009 route=web3
        BalancerMember ajp://localhost:12009 smax=10 route=web4
</Proxy>
```

1.10. RewriteEngine 连接 Tomcat

```
RewriteEngine On

RewriteRule ^/(.*) ajp://localhost:8009/ajp/$1 [P]
RewriteRule ^/(.*\.(jsp|do|sevlet)) ajp://localhost:8009/ajp/$1
[P]
```

1.11. SSL 双向认证

首先我并不建议使用 tomcat 实现SSL双向验证，这个工作可以交给 Web 服务器完成。但有些场景可能需要，可以参考下面例子。

服务器端证书

```
keytool -genkey -v -alias serverKey -dname "CN=localhost" -
keyalg RSA -keypass xxxxxx -keystore server.ks -storepass
```

```
xxxxxx
```

客户端证书

```
keytool -genkey -v -alias clientKey -dname "CN=SomeOne" -keyalg
RSA -keypass xxxxxx -keystore client.p12 -storepass xxxxxx -
storetype PKCS12
keytool -export -alias clientKey -file clientKey.cer -keystore
client.p12 -storepass xxxxxx -storetype PKCS12
```

导入客户端证书

```
keytool -import -v -alias clientKey -file clientKey.cer -
keystore server.ks -storepass xxxxxx
```

如果希望在 Windows 浏览器中访问，下导入证书方式，双击 client.p12 文件，安装提示导入

配置 Tomcat ，编辑 server.xml 文件

```
<Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true"
maxThreads="1024" scheme="https" secure="true"
clientAuth="true" sslProtocol="TLS"
keystoreFile="server.ks" keystorePass="xxxxxx"
truststoreFile="server.ks " truststorePass="xxxxxx" />
```

2. 配置 Tomcat 服务器

2.1. server.xml

Connector

tomcat 端口默认为8080, 可以通过修改下面port项改为80端口，但不建议你这样使用80端口,tomcat 会继承root权限，这是非常危险的做法。

```
<Connector port="80" protocol="HTTP/1.1"
           connectionTimeout="20000"
           redirectPort="8443" />
```

性能调整

```
<Connector port="8080" protocol="HTTP/1.1"
           connectionTimeout="20000"
           redirectPort="8443"
           maxThreads="2048" />

    <Connector port="8080" protocol="HTTP/1.1"
                            maxThreads="2048"
                            minSpareThreads="64"
                            maxSpareThreads="256"
                            acceptCount="128"
                            enableLookups="false"
                            redirectPort="8443"
                            debug="0"
                            connectionTimeout="20000"
                            disableUploadTimeout="true"
                            URIEncoding="UTF-8" />
```

```
maxThreads="4096"               最大连接数
minSpareThreads="50"    最小空闲线程
maxSpareThreads="100"   最大空闲线程
enableLookups="false"   禁止域名解析
acceptCount="15000"
connectionTimeout="30000"       超时时间
redirectPort="8443"
disableUploadTimeout="true"
URIEncoding="UTF-8"             UTF-8编码
protocol="AJP/1.3"              AJP协议版本
```

HTTPS

```
   <Connector port="443" maxHttpHeaderSize="8192"
               maxThreads="150" minSpareThreads="25"
maxSpareThreads="75"
               enableLookups="false"
disableUploadTimeout="true"
               acceptCount="100" scheme="https" secure="true"
               SSLEngine="on"

SSLCertificateFile="${catalina.base}/conf/localhost.crt"

SSLCertificateKeyFile="${catalina.base}/conf/localhost.key" />
```

compression

压缩传送数据

```
compression="on"
compressionMinSize="2048"
noCompressionUserAgents="gozilla, traviata"
compressableMimeType="text/html,text/xml,text/plain,text/javasc
```

```
ript,text/css"
```

useBodyEncodingForURI

如果你的站点编码非UTF-8,去掉URIEncoding="UTF-8"使用下面选项.

useBodyEncodingForURI="true"

隐藏Tomcat版本信息

在Connector中加入server="Neo App Srv 1.0"

```
vim $CATALINA_HOME/conf/server.xml

    <Connector port="80" protocol="HTTP/1.1"
               connectionTimeout="20000"
               redirectPort="8443"
                                maxThreads="8192"
                                minSpareThreads="64"
                                maxSpareThreads="128"
                                acceptCount="128"
                                enableLookups="false"
                server="Neo App Srv 1.0"/>
```

```
# curl -I http://localhost:8080/
HTTP/1.1 400 Bad Request
Transfer-Encoding: chunked
Date: Thu, 20 Oct 2011 09:51:55 GMT
Connection: close
Server: Neo App Srv 1.0
```

Context

配置虚拟目录

例如我们需要这样的配置

```
http://www.netkiller.cn/news
http://www.netkiller.cn/member
http://www.netkiller.cn/product
```

实现方法

```


<Host name="localhost"  appBase="/www/example.com"
unpackWARs="true" autoDeploy="true">
        <Alias>www.example.com</Alias>

        <Context path="news" docBase="www.example.com/news"
reloadable="false"></Context>
        <Context path="member" docBase="www.example.com/member"
reloadable="false"></Context>
        <Context path="product"
docBase="www.example.com/product" reloadable="false"></Context>

</Host>
```

应用程序安全

关闭war自动部署 unpackWARs="false" autoDeploy="false"。防止被植入木马等恶意程序

关闭 reloadable="false" 也用于防止被植入木马

JSESSIONID

修改 Cookie 变量 JSESSIONID， 这个cookie 是用于维持Session关系。建议你改为PHPSESSID。

```
<Context path="" docBase="path/to/your" reloadable="false"
sessionCookieDomain=".example.com" sessionCookiePath="/"
sessionCookieName="PHPSESSID" />
```

2.2. tomcat-users.xml

```
<?xml version='1.0' encoding='utf-8'?>
<tomcat-users>

<role rolename="manager"/>
<user username="tomcat" password="QI0Ajp7" roles="manager"/>

</tomcat-users>
```

状态监控 http://localhost/manager/status

服务管理 http://localhost/manager/html/list

```
<tomcat-users>
<!--
  NOTE:  By default, no user is included in the "manager-gui"
role required
  to operate the "/manager/html" web application.  If you wish
to use this app,
  you must define such a user - the username and password are
arbitrary.
-->
<!--
  NOTE:  The sample user and role entries below are wrapped in
a comment
  and thus are ignored when reading this file. Do not forget to
remove
```

```
  <!.. ..> that surrounds them.
-->
<!--
  <role rolename="tomcat"/>
  <role rolename="role1"/>
  <user username="tomcat" password="tomcat" roles="tomcat"/>
  <user username="both" password="tomcat" 
roles="tomcat,role1"/>
  <user username="role1" password="tomcat" roles="role1"/>
-->
  <role rolename="manager-gui"/>
  <role rolename="manager-script"/>
  <role rolename="manager-jmx"/>
  <role rolename="manager-status"/>

  <user username="tomcat" password="tomcat" roles="manager-
gui,manager-script,manager-jmx,manager-status"/>
  <role rolename="admin-gui"/>
  <role rolename="admin-script"/>
  <user username="admin" password="admin" roles="admin-
gui,admin-script"/>

</tomcat-users>
```

2.3. context.xml

context.xml 主要用于配置 数据库连接池

开启热部署，生产环境不建议使用

```
<Context reloadable="true">
```

Resources

org.apache.catalina.webresources.Cache.getResource Unable to add the resource at [/WEB-INF/lib/netkiller.jar] to the cache because there was insufficient free space available after evicting expired cache entries - consider increasing the maximum size of the cache

```
<Resources cachingAllowed="true" cacheMaxSize="100000" />
```

session cookie

```
<Context sessionCookieName="PHPSESSID"
sessionCookieDomain=".example.com" sessionCookiePath="/">
        <!-- ... -->
</Context>
```

2.4. logging.properties

修改日志目录

```
1catalina.org.apache.juli.FileHandler.level = FINE
#1catalina.org.apache.juli.FileHandler.directory =
${catalina.base}/logs
1catalina.org.apache.juli.FileHandler.directory =
/www/logs/tomcat
1catalina.org.apache.juli.FileHandler.prefix = catalina.
```

2.5. catalina.properties

配置跳过扫描*.jar

```
tomcat.util.scan.StandardJarScanFilter.jarsToSkip=\*.jar
```

context.xml

```
<JarScanner scanClassPath="false"/>
```

3. 虚拟主机配置

注意

```
   Tomcat 8 取消了 xmlValidation="false"
xmlNamespaceAware="false" 两个配置项。
```

appBase 是防止 war 文件的扫描目录。

3.1. 方案一

将配置写入server.xml文件

```
      <Host name="www.example.com"  appBase="webapps"
            unpackWARs="true" autoDeploy="true"
            xmlValidation="false" xmlNamespaceAware="false">
            <Context path=""
docBase="/www/example.com/www.example.comm" debug="0"
reloadable="false"/>
      </Host>
      <Host name="news.example.com"  appBase="webapps"
            unpackWARs="true" autoDeploy="true"
            xmlValidation="false" xmlNamespaceAware="false">
            <Context path=""
docBase="/www/example/news.example.com" debug="0"
reloadable="false"/>
      </Host>
```

3.2. 方案二

在 $CATALINA_HOME/conf/Catalina/ 下创建配置文件

```
vim server.xml
```

```
<Engine name="Catalina" defaultHost="neo">
    <Host name="neo"    appBase="webapps"/>
    <Host name="other" appBase="webapps"/>
</Engine>
```

Webapps Directory

```
mkdir $CATALINA_HOME/conf/Catalina/neo
```

Configuring Your Contexts

```
mkdir $CATALINA_HOME/conf/Catalina/neo

cp $CATALINA_HOME/conf/Catalina/localhost/manager.xml
$CATALINA_HOME/conf/Catalina/neo/ROOT.xml

or

cp $CATALINA_HOME/conf/Catalina/localhost/manager.xml
$CATALINA_HOME/conf/Catalina/neo
```

3.3. Alias 别名

别名的功能是为虚拟主机绑定多个域名

```
<Host name="www.example.com" debug="9" appBase="webapps"
                              unpackWARs="false"
autoDeploy="false"
                               xmlValidation="false"
xmlNamespaceAware="false">
        <Alias>www.example.net</Alias>
        <Alias>exmaple.com</Alias>
        <Alias>224.25.22.70</Alias>
</Host>
```

3.4. access_log

```
<Host name="localhost" ...>
  ...
  <Valve className="org.apache.catalina.valves.AccessLogValve"
         prefix="localhost_access_log." suffix=".txt"
         pattern="common"/>
  ...
</Host>
```

```
        <Valve
className"org.apache.catalina.valves.AccessLogValve"
                directory="logs/access"
                prefix="www.netkiller.cn.access"
                suffix=".log"
                pattern="%{X-Forwarded-FOR}i %a %v %U %t %m %s
%{User-Agent}i" resolveHosts="false"/>
```

3.5. Context 配置

```
        <Host appBase="webapps" autoDeploy="true"
name="localhost" unpackWARs="true">
                <Valve
className="org.apache.catalina.valves.AccessLogValve"
directory="logs" pattern="%h %l %u %t &quot;%r&quot; %s %b"
prefix="localhost_access_log" suffix=".txt"/>
```

```
                <Context docBase="Struts" path="/Struts"
reloadable="true" source="org.eclipse.jst.jee.server:Struts"/>
        </Host>
```

docBase如果是绝对路径就会忽略appBase="webapps"的设置。

```
<Context path=""
docBase="/www/example.com/www.example.com/WebContent"
reloadable="false">
```

appBase + docBase 方案

```
<Host name="localhost"  appBase="/www/example.com"
unpackWARs="true" autoDeploy="true">
        <Alias>www.example.com</Alias>

        <Context path="" docBase="www.example.com"
reloadable="false"></Context>

</Host>
```

3.6. 主机绑定IP地址

```
    <Host name="223.225.22.72"  appBase="/www/netkiller.cn"
unpackWARs="true" autoDeploy="true">
                <Alias>www.netkiller.cn</Alias>
        <Valve
className="org.apache.catalina.valves.AccessLogValve"
directory="logs"
               prefix="www.netkiller.cn_access_log."
suffix=".log"
```

```
               pattern="%h %l %u %t &quot;%r&quot; %s %b" />
                <Logger
className="org.apache.catalina.logger.FileLogger"
                 directory="logs"  prefix="web_log."
suffix=".txt"  timestamp="true"/>
        <Context path="" docBase="www.netkiller.cn"
reloadable="true"></Context>

        </Host>
        <Host name="223.225.22.73"  appBase="/www/netkiller.cn"
unpackWARs="true" autoDeploy="true">
                <Alias>admin.netkiller.cn</Alias>
                <Valve
className="org.apache.catalina.valves.AccessLogValve"
directory="logs"
                       prefix="admin.netkiller.cn_access_log."
suffix=".log"
                       pattern="%h %l %u %t &quot;%r&quot; %s
%b" />
                <Context path="" docBase="admin.netkiller.cn"
reloadable="true" />
        </Host>
```

4. SSI

编辑 context.xml 文件，增加 privileged="true 属性

```
# vim /srv/apache-tomcat/conf/context.xml

<Context privileged="true">

    <!-- Default set of monitored resources -->
    <WatchedResource>WEB-INF/web.xml</WatchedResource>

    <!-- Uncomment this to disable session persistence across
Tomcat restarts -->
    <!--
    <Manager pathname="" />
    -->

    <!-- Uncomment this to enable Comet connection tacking
(provides events
         on session expiration as well as webapp lifecycle) -->
    <!--
    <Valve
className="org.apache.catalina.valves.CometConnectionManagerVal
ve" />
    -->

</Context>
```

编辑 web.xml 文件，取消下面的注释

```
# vim /srv/apache-tomcat/conf/web.xml

    <servlet>
        <servlet-name>ssi</servlet-name>
        <servlet-class>
          org.apache.catalina.ssi.SSIServlet
        </servlet-class>
        <init-param>
          <param-name>buffered</param-name>
          <param-value>1</param-value>
        </init-param>
        <init-param>
          <param-name>debug</param-name>
          <param-value>0</param-value>
        </init-param>
        <init-param>
          <param-name>expires</param-name>
          <param-value>666</param-value>
        </init-param>
        <init-param>
          <param-name>isVirtualWebappRelative</param-name>
          <param-value>false</param-value>
        </init-param>
        <load-on-startup>4</load-on-startup>
    </servlet>
```

配置需要SSI处理的文件

```
# vim  /srv/apache-tomcat/webapps/ROOT/WEB-INF/web.xml

<web-app xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
                      http://java.sun.com/xml/ns/javaee/web-
app_3_0.xsd"
  version="3.0"
  metadata-complete="true">

  <display-name>Welcome to Tomcat</display-name>
  <description>
     Welcome to Tomcat
  </description>


        <servlet-mapping>
        <servlet-name>ssi</servlet-name>
        <url-pattern>*.shtml</url-pattern>
        <url-pattern>*.html</url-pattern>
        </servlet-mapping>


</web-app>
```

重新启动Tomcat

创建测试文件

```
# vim webapps/ROOT/index.html

<!--#echo var="DATE_LOCAL" -->
```

验证测试结果

```
# curl http://224.25.22.70:8080/
Tuesday, 03-Nov-2015 09:32:30 HKT
```

5. Logging 日志

5.1. 开启 debug 模式

又是我们需要开启debug来排查故障，只需在项目目录下创建文件 WEB-INF/classes/log4j.properties 内容如下

```
log4j.rootLogger=debug,console,file
```

重新启动tomcat将进入Debug模式，你将看到大量的调试信息。

5.2. 切割 catalina.out 日志

```
1) log4j.properties: Add the console to the root logger
log4j.rootLogger = INFO, CATALINA, CONSOLE

2) log4j.properties: Change the DailyRollingFileAppender to:
log4j.appender.CATALINA=org.apache.log4j.rolling.RollingFileApp
ender
log4j.appender.CATALINA.RollingPolicy=org.apache.log4j.rolling.
TimeBasedRollingPolicy
log4j.appender.CATALINA.RollingPolicy.FileNamePattern=${catalin
a.base}/logs/catalina.%d{yyyy-MM-dd}.log
```

6. Init.d Script

6.1. Script 1

```
#!/bin/bash
##############################################################
# Script for Apache and Tomcat
# File:/etc/rc.d/init.d/www
##############################################################
# Setup environment for script execution
#

# chkconfig: - 91 35
# description: Starts and stops the apache and tomcat daemons \
#              used to provide Neo Chen
#
# pidfile: /var/run/www/apache.pid
# pidfile: /var/run/www/tomcat.pid
# config:  /etc/apache2/apache2.conf


#APACHE_HOME=/usr/local/apache
#TOMCAT_HOME=/usr/local/tomcat
#APACHE_USER=apache
#TOMCAT_USER=tomcat

APACHE_HOME=/usr/local/apache-evaluation
TOMCAT_HOME=/usr/local/apache-tomcat-evaluation
APACHE_USER=root
TOMCAT_USER=root

OPEN_FILES=20480

# Source function library.
if [ -f /etc/init.d/functions ] ; then
  . /etc/init.d/functions
elif [ -f /etc/rc.d/init.d/functions ] ; then
  . /etc/rc.d/init.d/functions
else
  exit 0
```

```
fi

if [ ! -d /var/run/www ] ; then
  mkdir /var/run/www
fi

if [ -f /var/lock/subsys/tomcat ] ; then
        echo " "
fi

start() {
        if [ `ulimit -n` != ${OPEN_FILES} ]; then
                ulimit -n ${OPEN_FILES}
        fi
        echo -en "\\033[1;32;1m"
        echo "Starting Tomcat $TOMCAT_HOME ..."
        echo -en "\\033[0;39;1m"
        if [ -s /var/run/www/tomcat.pid ]; then
                echo "tomcat (pid `cat
/var/run/www/tomcat.pid`) already running"
        else
                su - ${TOMCAT_USER} -c
"$TOMCAT_HOME/bin/catalina.sh start > /dev/null"
                echo `pgrep java` > /var/run/www/tomcat.pid
                touch /var/lock/subsys/tomcat
        fi
        sleep 2
        echo -en "\\033[1;32;1m"
        echo "Starting Apache $APACHE_HOME ..."
        echo -en "\\033[0;39;1m"
        su - ${APACHE_USER} -c "$APACHE_HOME/bin/apachectl
start"
        touch /var/lock/subsys/apache
}

stop() {
        echo -en "\\033[1;32;1m"
        echo "Shutting down Apache $APACHE_HOME ..."
        echo -en "\\033[0;39;1m"
        su - ${APACHE_USER} -c "$APACHE_HOME/bin/apachectl
stop"
        sleep 2
        echo -en "\\033[1;32;1m"
        echo "Shutting down Tomcat $TOMCAT_HOME ..."
        echo -en "\\033[0;39;1m"
        su - ${TOMCAT_USER} -c "$TOMCAT_HOME/bin/catalina.sh
stop > /dev/null"
        rm -rf /var/run/www/tomcat.pid
        rm -f /var/lock/subsys/tomcat
        rm -f /var/lock/subsys/apache
}

restart() {
    stop
        if [ "`pgrep java`" = "" ]&& [ "`pgrep httpd`" = "" ];
then
                start
                exit 0
    else
                echo "Usage: $0 killall (^C)"
                echo -n "Waiting: "
    fi
    while true;
        do
                sleep 1
                if [ "`pgrep java`" = "" ] && [ "`pgrep httpd`"
= "" ]; then
                        break
                else
                        echo -n "."
                        #echo -n "Enter your [y/n]: "; read
ISKILL;
                fi
        done
        echo
    start
}

status() {
                ps -aux | grep -e tomcat -e apache

                echo -en "\\033[1;32;1m"
                echo ulimit open files: `ulimit -n`
                echo -en "\\033[0;39;1m"

                echo -en "\\033[1;32;1m"
                echo -en "httpd count:"
                ps axf|grep httpd|wc -l
                echo -en "\\033[0;39;1m"
}
```

```
killall() {
        if [ "`pgrep httpd`" != "" ]; then
                echo -en "\\033[1;32;1m"
                echo "kill Apache pid(`pgrep httpd`) ..."
                kill -9 `pgrep httpd`
                echo -en "\\033[0;39;1m"
        fi
        if [ "`pgrep java`" != "" ]; then
                echo -en "\\033[1;32;1m"
                echo "kill Tomcat pid(`pgrep java`) ..."
                kill -9 `pgrep java`
                echo -en "\\033[0;39;1m"
        fi
        rm -rf /var/run/www/tomcat.pid
        rm -f /var/lock/subsys/tomcat
        rm -f /var/lock/subsys/apache
}

# Determine and execute action based on command line parameter
case "$1" in
        start)
                start
                ;;
        stop)
                stop
                ;;
        restart)
                restart
                ;;
        status)
                status
                ;;
        killall)
                killall
                ;;
        *)
                echo -en "\\033[1;32;1m"
                echo "Usage: $1
{start|stop|restart|status|killall}"
                echo -en "\\033[0;39;1m"
                ;;
esac
echo -en "\\033[0;39;m"
exit 0
```

6.2. Shell Script 2

Apache,Tomcat 运行脚本

例 3.5. /etc/rc.d/init.d/www

```
#!/bin/bash
##############################################################
# Script for Apache and Tomcat
# File:/etc/rc.d/init.d/www
##############################################################
# Setup environment for script execution
#

# chkconfig: - 91 35
# description: Starts and stops the apache and tomcat daemons \
#              used to provide Neo Chen<openunix@163.com>
#
# pidfile: /var/run/www/apache.pid
# pidfile: /var/run/www/tomcat.pid
# config:  /etc/apache2/apache2.conf


#APACHE_HOME=/usr/local/apache
#TOMCAT_HOME=/usr/local/tomcat
#APACHE_USER=apache
#TOMCAT_USER=tomcat

APACHE_HOME=/usr/local/apache
TOMCAT_HOME=/usr/local/tomcat
APACHE_USER=root
TOMCAT_USER=root
WAIT_TIME=10
get_apache_pid(){
    APACHE_PID=`pgrep -o httpd`
    echo $APACHE_PID
}
get_tomcat_pid(){
    TOMCAT_PID=`ps axww | grep catalina.home | grep -v 'grep' |
sed q | awk '{print $1}'`
    echo $TOMCAT_PID
}

#OPEN_FILS=40960

# Source function library.
#if [ -f /etc/init.d/functions ] ; then
#  . /etc/init.d/functions
#elif [ -f /etc/rc.d/init.d/functions ] ; then
#  . /etc/rc.d/init.d/functions
#else
#  exit 0
#fi

if [ ! -d /var/run/www ] ; then
  mkdir /var/run/www
fi

#if [ -f /var/lock/subsys/tomcat ] ; then
#fi

start() {
        #if [ `ulimit -n` -le  ${OPEN_FILES} ]; then
        #       ulimit -n ${OPEN_FILES}
        #fi
        echo -en "\\033[1;32;1m"
        echo "Starting Tomcat $TOMCAT_HOME ..."
        echo -en "\\033[0;39;1m"
        if [ -s /var/run/www/tomcat.pid ]; then
                echo "tomcat (pid `cat
/var/run/www/tomcat.pid`) already running"
        else
                su - ${TOMCAT_USER} -c
"$TOMCAT_HOME/bin/catalina.sh start > /dev/null"
                echo `get_tomcat_pid` > /var/run/www/tomcat.pid
                touch /var/lock/subsys/tomcat
        fi
        sleep 2
        echo -en "\\033[1;32;1m"
        echo "Starting Apache $APACHE_HOME ..."
        echo -en "\\033[0;39;1m"
        su - ${APACHE_USER} -c "$APACHE_HOME/bin/apachectl
start"
```

```
        touch /var/lock/subsys/apache
}

stop() {
        echo -en "\\033[1;32;1m"
        echo "Shutting down Apache $APACHE_HOME ..."
        echo -en "\\033[0;39;1m"
        su - ${APACHE_USER} -c "$APACHE_HOME/bin/apachectl
stop"
        sleep 2
        echo -en "\\033[1;32;1m"
        echo "Shutting down Tomcat $TOMCAT_HOME ..."
        echo -en "\\033[0;39;1m"
        su - ${TOMCAT_USER} -c "$TOMCAT_HOME/bin/catalina.sh
stop > /dev/null"
        rm -rf /var/run/www/tomcat.pid
        rm -f /var/lock/subsys/tomcat
        rm -f /var/lock/subsys/apache
}

restart() {
    stop
    sleep 2
    if [ -z `get_tomcat_pid` ]&& [ -z `get_apache_pid` ]; then
                start
                exit 0
    else
                echo "Usage: $0 killall (^C)"
                echo -n "Waiting: "
    fi
    while true;
        do
                sleep 1
                if [ -z `get_tomcat_pid` ] && [ -z
`get_apache_pid` ]; then
                        break
                else
                        echo -n "."
                fi
        done
        echo
    start
}

k9restart() {
    ISEXIT='false'
    stop
    for i in `seq 1 ${WAIT_TIME}`;
    do
                if [ -z `get_tomcat_pid` ] && [ -z
`get_apache_pid` ]; then
                ISEXIT='true'
                break
                else
                        sleep 1
                fi
        done

        if [ $ISEXIT == 'false' ]; then
            while true;
                do
                        if [ -z `get_tomcat_pid` ] && [ -z
`get_apache_pid` ]; then
                                ISEXIT='true'
                        break
                        fi

                        if [ -n `get_apache_pid` ]; then
                                kill -9 `pgrep httpd`
                        fi
                        if [ -n `get_tomcat_pid` ]; then
                                kill -9 `get_tomcat_pid`
                        fi
                done
                rm -rf /var/run/www/tomcat.pid
                rm -f /var/lock/subsys/tomcat
                rm -f /var/lock/subsys/apache
        fi

        echo

        if [ $ISEXIT == 'true' ]; then
                start
        fi
}

status() {
                #ps -aux | grep -e tomcat -e apache

                echo -en "\\033[1;32;1m"
                echo ulimit open files: `ulimit -n`
                echo -en "\\033[0;39;1m"

                echo -en "\\033[1;32;1m"
                echo -en "httpd count:"
                let hc=`ps axf|grep httpd|wc -l`-1
                echo $hc
                echo -en "apache count:"
                netstat -alp | grep '*:http' | wc -l
                echo -en "tomcat count:"
                netstat -alp | grep '*:webcache' | wc -l
                echo -en "dbconn count:"
                netstat -a | grep ':3433' | wc -l
                echo -en "\\033[0;39;1m"
}

kall() {
        if [ `get_apache_pid` ]; then
                echo -en "\\033[1;32;1m"
                echo "kill Apache pid(`pgrep httpd`) ..."
                kill `pgrep httpd`
                echo -en "\\033[0;39;1m"
        fi
        if [ `get_tomcat_pid` ]; then
                echo -en "\\033[1;32;1m"
                echo "kill Tomcat pid(`pgrep java`) ..."
                kill `pgrep java`
                echo -en "\\033[0;39;1m"
        fi
        rm -rf /var/run/www/tomcat.pid
        rm -f /var/lock/subsys/tomcat
        rm -f /var/lock/subsys/apache
}

reload() {
        killall -HUP httpd
}

tomcat_restart() {
    su - ${TOMCAT_USER} -c "$TOMCAT_HOME/bin/catalina.sh stop >
/dev/null"
    rm -rf /var/run/www/tomcat.pid
    rm -f /var/lock/subsys/tomcat
    sleep 2
    if [ -z `get_tomcat_pid` ]; then
```

```
        su - ${TOMCAT_USER} -c "$TOMCAT_HOME/bin/catalina.sh
start > /dev/null"
        exit 0
    else
        echo "Usage: $0 killall (^C)"
        echo -n "Waiting: "
    fi
    while true;
    do
                sleep 1
                if [ -z `get_tomcat_pid` ]; then
                        echo
                        break
                else
                        echo -n "."
                        #echo -n "Enter your [y/n]: "; read
ISKILL;
                fi
    done
    su - ${TOMCAT_USER} -c "$TOMCAT_HOME/bin/catalina.sh start
> /dev/null"
    echo `get_tomcat_pid` > /var/run/www/tomcat.pid
    touch /var/lock/subsys/tomcat
}


# Determine and execute action based on command line parameter
case $1 in
    apache)
        case "$2" in
            reload)
                reload
                ;;
            *)
                su - ${APACHE_USER} -c
"${APACHE_HOME}/bin/apachectl $2"
                ;;
        esac
        ;;
    tomcat)
        case "$2" in
            restart)
                tomcat_restart
                ;;
            *)
```

```
                su - ${TOMCAT_USER} -c
"${TOMCAT_HOME}/bin/catalina.sh $2"
                ;;
        esac
        ;;
    start)
        start
        ;;
    stop)
        stop
        ;;
    restart)
        restart
        ;;
    status)
        status
        ;;
    killall)
        kall
        ;;
    k9restart)
        k9restart >/dev/null
        ;;
    *)
        echo -en "\\033[1;32;1m"
        echo "Usage: $0
{start|stop|restart|status|killall|k9restart}"
        echo "Usage: $0 apache
{start|restart|graceful|graceful-stop|stop|reload}"
        echo "Usage: $0 tomcat
{debug|run|start|restart|stop|version}"
        echo -en "\\033[0;39;1m"
        ;;
esac
echo -en "\\033[0;39;m"
exit 0
```

```
chmod 700 /etc/init.d/www
```

第 4 章 Apache httpd

LAMP

1. Install

1.1. Quick install apache with aptitude

$ sudo apt-get install apache2$ sudo apt-get install apache2-mpm-worker

```
netkiller@Linux-server:~$ sudo apt-get install apache2
```

command

enable module: a2enmod

enable site: a2ensite

rewrite module

```
$ sudo a2enmod rewrite
```

PHP module

```
$ sudo a2enmod php5
```

deflate module

```
root@neo:/etc/apache2# a2enmod deflate
Module deflate installed; run /etc/init.d/apache2 force-reload
to enable.
root@neo:/etc/apache2# /etc/init.d/apache2 force-reload
 * Forcing reload of apache 2.0 web server...
[ ok ]
root@neo:/etc/apache2#
```

ssl module

a2enmod ssl

a2ensite ssl

/etc/apache2/httpd.conf 加入

```
ServerName 220.201.35.11
```

安全模块

```
netkiller@Linux-server:~$ sudo apt-get install libapache2-mod-
security

netkiller@Linux-server:/etc/apache2$ sudo vi ports.conf
netkiller@Linux-server:/etc/apache2$ cat ports.conf
Listen 80
Listen 443

NameVirtualHost *
NameVirtualHost *:443


netkiller@Linux-server:/etc/apache2$ sudo apache2-ssl-
certificate
or
netkiller@Linux-server:~$ apache2-ssl-certificate -days 365
```

```
netkiller@Linux-server:~$ a2enmod ssl
or
netkiller@Linux-server:/etc/apache2/mods-enabled$ sudo ln -s
../mods-available/ssl.conf
netkiller@Linux-server:/etc/apache2/mods-enabled$ sudo ln -s
../mods-available/ssl.load

netkiller@Linux-server:/etc/apache2/sites-enabled$ sudo mkdir
ssl/
netkiller@Linux-server:/etc/apache2/sites-enabled$ sudo cp
netkiller woodart ssl/


netkiller@Linux-server:/etc/apache2/mods-enabled$ sudo
/etc/init.d/apache2 reload
 * Reloading apache 2.0 configuration...
[ ok ]
netkiller@Linux-server:/etc/apache2/mods-enabled$
```

VirtualHost

VirtualHost 虚拟主机

```
netkiller@Linux-server:/etc/apache2/sites-available$ sudo vi
woodart

#NameVirtualHost neo.6600.org
<VirtualHost 220.201.35.11>
        ServerAdmin openx@163.com

        DocumentRoot /home/netkiller/www
        ServerName neo.6600.org
        ServerAlias www.neo.6600.org
        <Directory /home/netkiller/www>
                Options Indexes FollowSymLinks MultiViews
                AllowOverride All
                Order allow,deny
                allow from all
                # Uncomment this directive is you want to see
```

```
apache2's
                # default start page (in /apache2-default) when
you go to /
                #RedirectMatch ^/$ /apache2-default/
        </Directory>

#       ScriptAlias /cgi-bin/ /home/netkiller/www/
#       <Directory "/home/netkiller/www">
#               AllowOverride None
#               Options +ExecCGI -MultiViews
+SymLinksIfOwnerMatch
#               Order allow,deny
#               Allow from all
#       </Directory>

        ErrorLog /var/log/apache2/neo.error.log

        # Possible values include: debug, info, notice, warn,
error, crit,
        # alert, emerg.
#       LogLevel warn

        CustomLog /var/log/apache2/neo.access.log combined
#       ServerSignature On

</VirtualHost>

netkiller@Linux-server:/etc/apache2/sites-available$ sudo
apache2 -k restart
```

~userdir module - /public_html

~web环境

```
netkiller@Linux-server:~$ mkdir public_html
netkiller@Linux-server:~$ cd public_html/
netkiller@Linux-server:~/public_html$
netkiller@Linux-server:~/public_html$ echo
helloworld>index.html
netkiller@Linux-server:~/public_html$ ls
```

```
index.html
```

http://xxx.xxx.xxx.xxx/~netkiller/

PHP 5

$ sudo apt-get install php5

```
netkiller@Linux-server:~$ sudo apt-get install php5
```

pgsql模块

```
netkiller@Linux-server:~$ sudo apt-get install php5-pgsql

netkiller@Linux-server:~$sudo cp
/usr/lib/php5/20051025/pgsql.so /etc/php5/apache2/
```

php5-gd - GD module for php5

$ sudo apt-get install php5-gd

```
netkiller@Linux-server:~$ apt-cache search gd
libgdbm3 - GNU dbm database routines (runtime version)
libgd2-xpm - GD Graphics Library version 2
php5-gd - GD module for php5
pnm2ppa - PPM to PPA converter
postgresql-doc-8.1 - documentation for the PostgreSQL database
management system
libruby1.8 - Libraries necessary to run Ruby 1.8
ruby1.8 - Interpreter of object-oriented scripting language
Ruby 1.8
klogd - Kernel Logging Daemon
sysklogd - System Logging Daemon
upstart-logd - boot logging daemon
netkiller@Linux-server:~$ sudo apt-get install php5-gd
```

```
netkiller@Linux-server:~$
```

1.2. CentOS 6

Install

Apache

```
[root@development ~]# yum -y install httpd
```

PHP

```
[root@development ~]# yum -y install php
[root@development ~]# yum -y install php-mysql php-gd php-
mbstring php-bcmath
[neo@development ~]$ sudo yum -y install php-pecl-memcache
```

mysql

```
[root@development ~]# yum -y install mysql-server
```

Uninstall

```
# yum remove httpd
```

Configure

Apache

VirtualHost

```
[root@development ~]# vim /etc/httpd/conf.d/vhost.conf
#
# Use name-based virtual hosting.
#
NameVirtualHost *:80
#
# NOTE: NameVirtualHost cannot be used without a port specifier
# (e.g. :80) if mod_ssl is being used, due to the nature of the
# SSL protocol.
#

#
# VirtualHost example:
# Almost any Apache directive may go into a VirtualHost
container.
# The first VirtualHost section is used for requests without a
known
# server name.
#
<VirtualHost *:80>
    ServerAdmin webmaster@dummy-host.example.com
    DocumentRoot /www/docs/dummy-host.example.com
    ServerName dummy-host.example.com
    ErrorLog logs/dummy-host.example.com-error_log
    CustomLog logs/dummy-host.example.com-access_log common
</VirtualHost>
```

MySQL

reset mysql's password

```
[root@development ~]# /usr/bin/mysqladmin -u root password
'new-password'
[root@development ~]# /usr/bin/mysqladmin -u root -h
development.domain.org password 'new-password'
```

Alternatively you can run:

```
[root@development ~]# /usr/bin/mysql_secure_installation
```

Starting

levels

```
[root@development ~]# chkconfig --list mysqld
mysqld          0:off   1:off   2:off   3:off   4:off   5:off
6:off

[root@development ~]# chkconfig --list httpd
httpd           0:off   1:off   2:off   3:off   4:off   5:off
6:off


[root@development ~]# chkconfig httpd on
[root@development ~]# chkconfig --list httpd
httpd           0:off   1:off   2:on    3:on    4:on    5:on
6:off

[root@development ~]# chkconfig mysqld on
```

```
[root@development ~]# chkconfig --list mysqld
mysqld          0:off   1:off   2:on    3:on    4:on    5:on
6:off
```

Apache

```
[root@development ~]# service httpd start
```

MySQL

```
[root@development ~]# service mysqld start
```

```
[root@development ~]# netstat -nat | grep 80
tcp        0      0 :::80                       :::*
LISTEN

[root@development ~]# netstat -nat | grep 3306
tcp        0      0 0.0.0.0:3306                0.0.0.0:*
LISTEN
```

FAQ

compile php

```
[root@development php-5.3.0]# yum install libxml2-devel
[root@development php-5.3.0]# yum install curl-devel
[root@development php-5.3.0]# yum install gd-devel
```

```
[root@development php-5.3.0]# yum install libjpeg-devel
[root@development php-5.3.0]# yum install libpng-devel
[root@development php-5.3.0]# yum install openldap-devel
[root@development php-5.3.0]# yum install mysql-devel
[root@development php-5.3.0]# yum install net-snmp-devel
```

1.3. Compile and then install Apache

Apache 安装与配置

configure

--with-mpm=worker 进程,线程混合方式效率提高不少

--enable-modules='dir mime' 没有它就找不到index.*文件

--enable-rewrite=shared Rewrite用于表态化

--enable-expires=shared 禁止页面被 cache

--enable-authz_host=shared Order权限

--enable-setenvif=shared

--enable-log_config=shared 日志格式

--enable-speling=shared 允许自动修正拼错的URL

--enable-deflate=shared 压缩传送

--enable-mods-shared='cache file-cache disk-cache mem-cache proxy proxy-ajp proxy-balancer' 代理和缓存

用于Java

```
tar zxvf httpd-2.2.4.tar.gz
cd httpd-2.2.4
```

```
./configure --prefix=/usr/local/httpd-2.2.4 \
--with-mpm=worker \
--enable-modules='dir mime' \
--enable-rewrite=shared \
--enable-authz_host=shared \
--enable-alias=shared \
--enable-setenvif=shared \
--enable-log_config=shared \
--enable-speling=shared \
--enable-filter=shared \
--enable-deflate=shared \
--enable-headers=shared \
--enable-expires=shared \
--enable-mods-shared='cache file-cache disk-cache mem-cache
proxy proxy-ajp proxy-balancer' \
--disable-include \
--disable-actions \
--disable-alias \
--disable-asis \
--disable-autoindex \
--disable-auth_basic \
--disable-authn_file \
--disable-authn_default \
--disable-authz_groupfile \
--disable-authz_user \
--disable-authz_default \
--disable-cgi \
--disable-cgid \
--disable-env \
--disable-negotiation \
--disable-status \
--disable-userdir
```

用于PHP

```
[root@development httpd-2.2.14]# yum install zlib-devel.x86_64

./configure --prefix=/usr/local/httpd-2.2.14 \
--with-mpm=worker \
--enable-so \
--enable-mods-shared=all \
--enable-static-support \
--enable-static-htpasswd \
--enable-static-htdigest \
--enable-static-ab \
--disable-include \
--disable-actions \
--disable-alias \
--disable-asis \
--disable-autoindex \
--disable-auth_basic \
--disable-authn_file \
--disable-authn_default \
--disable-authz_groupfile \
--disable-authz_user \
--disable-authz_default \
--disable-cgi \
--disable-cgid \
--disable-env \
--disable-negotiation \
--disable-status \
--disable-userdir
```

make; make install

启动

```
ln -s /usr/local/httpd-2.2.4/ /usr/local/apache

/usr/local/httpd/bin/apachectl start
```

优化编译条件

```
# vim server/mpm/worker/worker.c

# define DEFAULT_SERVER_LIMIT 256
# define MAX_SERVER_LIMIT 20000
# define DEFAULT_THREAD_LIMIT 512
# define MAX_THREAD_LIMIT 20000
```

PHP

过程 4.1. 安装PHP

1. 第一步

```
cd /usr/local/src
wget http://cn2.php.net/get/php-
5.3.0.tar.bz2/from/cn.php.net/mirror
tar jxvf php-5.3.0.tar.bz2
cd php-5.3.0
```

2. 第二步

```
./configure --prefix=/usr/local/php-5.3.0 \
--with-config-file-path=/usr/local/php-5.3.0/etc \
--with-apxs2=/usr/local/apache/bin/apxs \
--with-curl \
--with-gd \
--with-ldap \
--with-snmp \
--enable-zip \
--enable-exif \
--with-libxml-dir \
--with-mysql \
--with-mysqli \
--with-pdo-mysql \
--with-pdo-pgsql

make
make test
make install
```

   a. 建立符号连接

   ```
   ln -s /usr/local/php-5.3.0 /usr/local/php
   ```

b. php.ini

```
cp php.ini-dist /usr/local/php/etc/php.ini
```

c. conf/httpd.conf

```
AddType application/x-httpd-php .php .phtml
AddType application/x-httpd-php-source .phps
```

reload apache

3. 最后一步

phpinfo() 测试文件复杂到apache目录

例 4.1. index.php

```
<?php phpinfo(); ?>
```

`--with-snmp`

`redhat as4 启用 --with-snmp 需要安装下面包`

```
rpm -i elfutils-libelf-devel-0.97.1-3.i386.rpm
rpm -i elfutils-devel-0.97.1-3.i386.rpm
rpm -i beecrypt-devel-3.1.0-6.i386.rpm
rpm -i net-snmp-devel-5.1.2-11.EL4.7.i386.rpm
```

Automation Installing

例 4.2. autolamp.sh

```
#!/bin/bash
HTTPD_SRC=httpd-2.2.15.tar.gz
PHP_SRC=php-5.2.13.tar.gz
MYSQL_SRC='mysql-5.1.45.tar.gz'
MYSQL_LIBS_SRC='mysql-5.1.45-linux-x86_64-glibc23.tar.gz'

SRC_DIR=$(pwd)
HTTPD_DIR=${HTTPD_SRC%%.tar.gz}
PHP_DIR=${PHP_SRC%%.tar.*}
MYSQL_DIR=${MYSQL_SRC%%.tar.*}
MYSQL_LIBS_DIR=${MYSQL_LIBS_SRC%%.tar.*}

function clean(){
        rm -rf $HTTPD_DIR
        rm -rf $PHP_DIR
        rm -rf $MYSQL_DIR
        rm -rf $MYSQL_LIBS_DIR
}
function mysql(){
rm -rf $MYSQL_DIR
tar zxf $MYSQL_SRC
cd $MYSQL_DIR
./configure \
--prefix=/usr/local/$MYSQL_DIR \
--with-mysqld-user=mysql \
--with-unix-socket-path=/tmp/mysql.sock \
--with-charset=utf8 \
--with-collation=utf8_general_ci \
--with-pthread \
--with-mysqld-ldflags \
--with-client-ldflags \
--with-openssl \
--without-docs \
--without-debug \
--without-ndb-debug \
--without-bench
#-—without-isam
#--without-innodb \
#--without-ndbcluster \
#--without-blackhole \
#--without-ibmdb2i \
#--without-federated \
#--without-example \
#--without-comment \
#--with-extra-charsets=gbk,gb2312,utf8 \

#--localstatedir=/usr/local/mysql/data
#--with-extra-charsets=all
make clean
make && make install
cd ..
/usr/local/$MYSQL_DIR/bin/mysql_install_db
}
function httpd(){
rm -rf $HTTPD_DIR
tar zxf $HTTPD_SRC
cd $HTTPD_DIR
./configure --prefix=/usr/local/$HTTPD_DIR \
--with-mpm=worker \
--enable-so \
--enable-mods-shared=all \
--disable-authn_file \
--disable-authn_default \
--disable-authz_groupfile \
--disable-authz_user \
--disable-authz_default \
--disable-auth_basic \
--disable-include \
--disable-env \
--disable-status \
--disable-autoindex \
--disable-asis \
--disable-cgi \
--disable-cgid \
--disable-negotiation \
--disable-actions \
--disable-userdir \
--disable-alias

make clean
make && make install
cd ..
}
function php(){
rm -rf $MYSQL_LIBS_DIR
tar zxf $MYSQL_LIBS_SRC
rm -rf $PHP_DIR
```

```
tar zxf $PHP_SRC
cd $PHP_DIR

./configure --prefix=/usr/local/$PHP_DIR \
--with-config-file-path=/usr/local/$PHP_DIR/etc \
--with-apxs2=/usr/local/$HTTPD_DIR/bin/apxs \
--with-curl \
--with-gd \
--with-jpeg-dir=/usr/lib64 \
--with-iconv \
--with-zlib-dir \
--with-pear \
--with-libxml \
--with-dom \
--with-xmlrpc \
--with-openssl \
--with-mysql=/usr/local/mysql-5.1.45-linux-x86_64-glibc23 \
--with-mysqli \
--with-pdo-mysql \
--enable-memcache \
--enable-zip \
--enable-sockets \
--enable-soap \
--enable-mbstring \
--enable-magic-quotes \
--enable-inline-optimization \
--enable-xml

#make && make test && make install
make &&  make install
cp /usr/local/src/$PHP_DIR/php.ini-dist
/usr/local/$PHP_DIR/php.ini
}
function depend(){
        yum install gcc gcc-c++ -y
        yum install -y libxml2-devel libxslt-devel
        yum install curl-devel -y
        yum install gd-devel libjpeg-devel libpng-devel -y
        yum install ncurses-devel -y
        yum install mysql-devel -y
        yum install libevent-devel -y
}
function java(){
        #yum install java-1.6.0-openjdk -y
        chmod +x jdk-6u20-linux-x64.bin
        ./jdk-6u20-linux-x64.bin
        mv jdk1.6.0_20 ..
        ln -s /usr/local/jdk1.6.0_20 /usr/local/java
}
function memcached(){
        MEMCACHED_PKG=memcached-1.4.5.tar.gz
        MEMCACHED_SRC=memcached-1.4.5
        rm -rf $MEMCACHED_SRC
        tar zxf $MEMCACHED_PKG
        cd $MEMCACHED_SRC
        ./configure --prefix=/usr/local/memcached-1.4.5
        make && make install
}
# See how we were called.
case "$1" in
  clean)
        clean
        ;;
  httpd)
        httpd
        ;;
  php)
        php
        ;;
  mysql)
        if [ -f $0 ] ; then
                mysql
        fi
        ;;
  depend)
        depend
        ;;
  java)
        java
        ;;
  memcached)
        memcached
        ;;
  all)
        clean

        echo ###################################################
        echo # $MYSQL_DIR Installing...
        echo ###################################################
        mysql
        echo ##################################################
        echo # $HTTPD_DIR Installing...
        echo ##################################################
        httpd

        echo ##################################################
        echo # $PHP_DIR Installing...
        echo ##################################################
        php

        ln -s /usr/local/$HTTPD_DIR /usr/local/apache
        ln -s /usr/local/$MYSQL_DIR /usr/local/mysql
        ln -s /usr/local/$PHP_DIR /usr/local/php

        clean
        ;;
  *)
        echo $"Usage: $0 {httpd|php|mysql|all|clean}"
        RETVAL=2
        ;;
esac

exit $RETVAL
```

1.4. XAMPP

XAMPP for Linux

http://www.apachefriends.org/en/xampp-linux.html

install

```
tar xvfz xampp-linux-1.7.3a.tar.gz -C /opt
```

start

```
/opt/lampp/lampp start
```

stop

```
/opt/lampp/lampp stop
```

remove

```
rm -rf /opt/lampp
```

php5

```
./lampp php5
XAMPP: PHP 5.3.8 already active.

./lampp startapache
XAMPP: Starting Apache with SSL (and PHP5)...

./lampp startmysql
XAMPP: Starting MySQL...
```

2. Module

```
模块的做用如下：
mod_access       提供基于主机的访问控制命令
mod_actions 能够运行基于MIME类型的CGI脚本或HTTP请求方法
mod_alias        能执行URL重定向服务
mod_asis         使文档能在没有HTTP头标的情况下被发送到客户端
mod_auth         支持使用存储在文本文件中的用户名、口令实现认证
mod_auth_dbm 支持使用DBM文件存储基本HTTP认证
mod_auth_mysql 支持使用MySQL数据库实现基本HTTP认证
mod_auth_anon 允许以匿名方式访问需要认证的区域
mod_auth_external支持使用第三方认证
mod_autoindex 当缺少索引文件时，自动生成动态目录列表
mod_cern_meta 提供对元信息的支持
mod_cgi          支持CGI
mod_dir          能够重定向任何对不包括尾部斜杠字符命令的请求
mod_env          使你能够将环境变量传递给CGI或SSI脚本
mod_expires 让你确定Apache在服务器响应请求时如何处理Expires
mod_headers 能够操作HTTP应答头标
mod_imap         提供图形映射支持
mod_include 使支持SSI
mod_info         对服务器配置提供了全面的描述
mod_log_agent 允许在单独的日志文件中存储用户代理的信息
mod_log_config 支持记录日志
mod_log_referer 提供了将请求中的Referer头标写入日志的功能
mod_mime 用来向客户端提供有关文档的元信息
mod_negotiation 提供了对内容协商的支持
mod_setenvif 使你能够创建定制环境变量
mod_speling 使你能够处理含有拼写错误或大小写错误的URL请求
mod_status       允许管理员通过WEB管理Apache
mod_unique_id 为每个请求提供在非常特殊的条件下保证是唯一的标识
```

常用模块

```
LoadModule dir_module          modules/mod_dir.so
LoadModule mime_module         modules/mod_mime.so
LoadModule expires_module      modules/mod_expires.so
LoadModule config_log_module   modules/mod_log_config.so
LoadModule alias_module        modules/mod_alias.so
LoadModule rewrite_module     modules/mod_rewrite.so
LoadModule access_module      modules/mod_access.so
LoadModule auth_module        modules/mod_auth.so
```

2.1. Output a list of modules compiled into the server.

This will not list dynamically loaded modules included using the LoadModule directive.

```
[root@development bin]# httpd -l
Compiled in modules:
  core.c
  worker.c
  http_core.c
  mod_so.c
```

2.2. Core

Listen

绑定多个IP

```
#Listen 80
Listen 192.168.3.40:80
Listen 192.168.4.40:80
Listen 192.168.5.40:80
```

Filesystem and Webspace

ref: http://httpd.apache.org/docs/2.2/en/sections.html

Filesystem Containers

```
<Directory /var/web/dir1>
        Options +Indexes
</Directory>

<Files private.html>
        Order allow,deny
        Deny from all
</Files>

<Directory /var/web/dir1>
        <Files private.html>
                Order allow,deny
                Deny from all
        </Files>
</Directory>
```

Webspace Containers

```
<LocationMatch ^/private>
        Order Allow,Deny
        Deny from all
</LocationMatch>
```

Wildcards and Regular Expressions

```
A non-regex wildcard section that changes the configuration of
all user directories could look as follows:

<Directory /home/*/public_html>
Options Indexes
</Directory>
Using regex sections, we can deny access to many types of image
files at once:
```

```
<FilesMatch \.(?i:gif|jpe?g|png)$>
Order allow,deny
Deny from all
</FilesMatch>
```

Options

```
<DirectoryMatch (/var/www/logs|/var/www/logs/*)>
        Options FollowSymLinks MultiViews Indexes

        DirectoryIndex index.html

        AllowOverride AuthConfig
        Order Allow,Deny
        Allow From All

        AuthName "Logs Access"
        AuthType Basic
        AuthUserFile /etc/nagios3/htpasswd.users
        require valid-user
</DirectoryMatch>
```

1. None是禁止所有
2. Indexes 当没有index.html 的时候列出目录
3. FollowSymLinks 允许符号连接，可以通过符号连接跨越DocumentRoot
4. AllowOverride 定义是否允许各个目录用目录中的.htaccess覆盖这里设定的Options
5.

Etag

```
<Directory /www>
        <Files ~ "\.(gif|jpe?g|png|html|css|js)$">
                FileETag INode MTime Size
        </Files>
</Directory>
```

隐藏 Apache 版本信息

```
ServerTokens ProductOnly
ServerSignature Off
```

2.3. mpm

event

ThreadLimit 需要自行添加

ServerLimit 需要自行添加

```
<IfModule mpm_event_module>
    ThreadLimit                 256
    ServerLimit                 4096
    StartServers             4
    MinSpareThreads         75
    MaxSpareThreads        250
    ThreadsPerChild        128
    MaxRequestWorkers      4096
    MaxConnectionsPerChild   0
</IfModule>
```

worker

worker

```
# Server-pool management (MPM specific)
Include conf/extra/httpd-mpm.conf
```

conf/extra/httpd-mpm.conf

mpm_worker_module

```


<IfModule mpm_worker_module>
    ServerLimit         16
    ThreadLimit         128
    StartServers        8
    MaxClients          2048
    MinSpareThreads     64
    MaxSpareThreads     128
    ThreadsPerChild     128
    MaxRequestsPerChild 10000
</IfModule>

<IfModule mpm_worker_module>
    ServerLimit         24
    ThreadLimit         128
    StartServers        8
    MaxClients          3072
    MinSpareThreads     64
    MaxSpareThreads     128
    ThreadsPerChild     128
    MaxRequestsPerChild 10000
</IfModule>

<IfModule mpm_worker_module>
    ServerLimit         16
    ThreadLimit         256
    StartServers        8
    MaxClients          4096
    MinSpareThreads     64
    MaxSpareThreads      256
    ThreadsPerChild      256
    MaxRequestsPerChild 10000
</IfModule>
```

ServerLimit 默认是16，它决定系统最多启动几个httpd进程。
ThreadLimit 默认是64，
ThreadsPerChild* ServerLimit=系统支持的最大并发。
MaxClients<ThreadsPerChild* ServerLimit，MaxClients如果大于400将被限制在400.
400只是理论最大并发，实际并发就是MaxClients的值。
理论并发有什么用我不知道。

指令说明：

StartServers：设置服务器启动时建立的子进程数量。因为子进程数量动态的取决于负载的轻重,所有一般没有必要调整这个参数。

ServerLimit：服务器允许配置的进程数上限。只有在你需要将MaxClients和ThreadsPerChild设置成需要超过默认值16个子进程的时候才需要使用这个指令。不要将该指令的值设置的比MaxClients 和ThreadsPerChild需要的子进程数量高。修改此指令的值必须完全停止服务后再启动才能生效，以restart方式重启动将不会生效。

ThreadLimit：设置每个子进程可配置的线程数ThreadsPerChild上限,该指令的值应当和ThreadsPerChild可能达到的最大值保持一致。修改此指令的值必须完全停止服务后再启动才能生效，以restart方式重启动将不会生效。

MaxClients：用于伺服客户端请求的最大接入请求数量（最大线程数）。任何超过MaxClients限制的请求都将进入等候队列。默认值是"400"，16(ServerLimit)乘以25(ThreadsPerChild)的结果。因此要增加MaxClients的时候，你必须同时增加 ServerLimit的值。笔者建议将初始值设为(以Mb为单位的最大物理内存/2),然后根据负载情况进行动态调整。比如一台4G内存的机器，那么初始值就是4000/2=2000。

MinSpareThreads：最小空闲线程数,默认值是"75"。这个MPM将基于整个服务器监视空闲线程数。如果服务器中总的空闲线程数太少，子进程将产生新的空闲线程。

MaxSpareThreads：设置最大空闲线程数。默认值是"250"。这个MPM将基于整个服务器监视空闲线程数。如果服务器中总的空闲线程数太多，子进程将杀死多余的空闲线程。MaxSpareThreads的取值范围是有限制的。Apache将按照如下限制自动修正你设置的值：worker要求其大于等于 MinSpareThreads加上ThreadsPerChild的和。

ThreadsPerChild：每个子进程建立的线程数。默认值是25。子进程在启

```
动时建立这些线程后就不再建立新的线程了。每个子进程所拥有的所有线程的总数要足
够大，以便可以处理可能的请求高峰。
        MaxRequestsPerChild：设置每个子进程在其生存期内允许伺服的最大
请求数量。到达MaxRequestsPerChild的限制后，子进程将会结束。如果
MaxRequestsPerChild为"0"，子进程将永远不会结束。将
MaxRequestsPerChild设置成非零值有两个好处：可以防止(偶然的)内存泄漏无限
进行而耗尽内存；
给进程一个有限寿命，从而有助于当服务器负载减轻的时候减少活动进程的数量。
如果设置为非零值，笔者建议设为10000-30000之间的一个值。
         公式：
         ThreadLimit >= ThreadsPerChild
         MaxClients <= ServerLimit * ThreadsPerChild 必须是
ThreadsPerChild的倍数
         MaxSpareThreads >= MinSpareThreads+ThreadsPerChild
```

2.4. Apache Log

LogLevel

日志级别

语法：LogLevel level

```
可以选择下列level，依照重要性降序排列：
emerg   紧急(系统无法使用)
alert   必须立即采取措施
crit     致命情况
error   错误情况
warn     警告情况
notice   一般重要情况
info    普通信息
debug   调试信息
```

```
LogLevel crit
```

LogFormat

分割log日志文件

```
<IfModule log_config_module>
    #
    # The following directives define some format nicknames for
use with
    # a CustomLog directive (see below).
    #
    #LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%
{User-Agent}i\"" combined
    LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%
{User-Agent}i\" %{email}C %{nickname}C" combined
    LogFormat "%h %l %u %t \"%r\" %>s %b" common

    <IfModule logio_module>
      # You need to enable mod_logio.c to use %I and %O
      LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%
{User-Agent}i\" %I %O" combinedio
    </IfModule>

    #
    # The location and format of the access logfile (Common
Logfile Format).
    # If you do not define any access logfiles within a
<VirtualHost>
    # container, they will be logged here.  Contrariwise, if
you *do*
    # define per-<VirtualHost> access logfiles, transactions
will be
    # logged therein and *not* in this file.
    #
    #CustomLog logs/access_log common

    #
    # If you prefer a logfile with access, agent, and referer
information
    # (Combined Logfile Format) you can use the following
directive.
    #
    CustomLog logs/access_log combined

    #CookieLog logs/cookie_log
```

```
</IfModule>
```

Compressed

```
# compressed logs
$ CustomLog "|/usr/bin/gzip -c >> /var/log/access_log.gz"
common
```

rotatelogs - Piped logging program to rotate Apache logs

rotatelogs是一个配合Apache管道日志功能使用的简单程序。举例:

```
rotatelogs logfile [ rotationtime [ offset ]] | [ filesizeM ]

选项
logfile
它加上基准名就是日志文件名。如果logfile中包含'%'，则它会被视为用于的
strftime(3)的格式字串；否则，它会被自动加上以秒为单位的.nnnnnnnnnn后
缀。这两种格式都表示新的日志开始使用的时间。
rotationtime
日志文件回卷的以秒为单位的间隔时间
offset
相对于UTC的时差的分钟数。如果省略，则假定为0，并使用UTC时间。比如，要指定
UTC时差为-5小时的地区的当地时间，则此参数应为-300。
filesizeM
指定回卷时以兆字节为单位的后缀字母M的文件大小，而不是指定回卷时间或时差。

下列日志文件格式字串可以为所有的strftime(3)实现所支持，见各种扩展库对应的
strftime(3)的手册。
%A 星期名全称(本地的)
%a 3个字符的星期名(本地的)
%B 月份名的全称(本地的)
%b 3个字符的月份名(本地的)
%c 日期和时间(本地的)
```

```
%d 2位数的一个月中的日期数
%H 2位数的小时数(24小时制)
%I 2位数的小时数(12小时制)
%j 3位数的一年中的日期数
%M 2位数的分钟数
%m 2位数的月份数
%p am/pm 12小时制的上下午(本地的)
%S 2位数的秒数
%U 2位数的一年中的星期数(星期天为一周的第一天)
%W 2位数的一年中的星期数(星期一为一周的第一天)
%w 1位数的星期几(星期天为一周的第一天)
%X 时间 (本地的)
%x 日期 (本地的)
%Y 4位数的年份

CustomLog "|bin/rotatelogs /var/logs/logfile 86400" common
此配置会建立文件"/var/logs/logfile.nnnn"，其中的nnnn是名义上的日志启动
时的系统时间(此时间总是滚动时间的倍数，可以用于cron脚本的同步)。在滚动时间
到达时(在此例中是24小时以后)，会产生一个新的日志。

CustomLog "|bin/rotatelogs /var/logs/logfile 5M" common
此配置会在日志文件大小增长到5兆字节时滚动该日志。

ErrorLog "|bin/rotatelogs /var/logs/errorlog.%Y-%m-%d-%H_%M_%S
5M"
此配置会在错误日志大小增长到5兆字节时滚动该日志，日志文件名后缀会按照如下格
式创建：errorlog.YYYY-mm-dd-HH_MM_SS

ErrorLog "| /usr/local/apache/bin/rotatelogs
/www/logs/www.example.com/error_%Y_%m_%d_log 86400 480"
CustomLog "| /usr/local/apache/bin/rotatelogs
/www/logs/www.example.com/access_%Y_%m_%d_log 86400 480" common

CustomLog "|/usr/local/httpd/bin/rotatelogs
/www/logs/www.example.com/access.%Y-%m-%d.log 86400 480"
combined
```

86400：表示 24小时 60*60*24

480: 表示时区偏移 8 时区等于 60*8

cronolog

cronolog

```
cd /usr/local/src/
wget http://cronolog.org/download/cronolog-1.6.2.tar.gz
tar zxvf cronolog-1.6.2.tar.gz
cd cronolog-1.6.2
./configure --prefix=/usr/local/cronolog
make
make install
```

CustomLog "|/usr/local/cronolog/sbin/cronolog /opt/apache/logs/access_log.%Y%m%d" combined

日志合并

合并多个服务器的日志文件（如log1、log2、log3），并输出到log_all中的方法是:

```
 $ sort -m -t " " -k 4 -o log_all log1 log2 log3
```

日志归档

```
30 4 * * * /usr/bin/gzip -f /www/logs/access.`date -d yesterday
+%Y-%m-%d`.log
```

logger

https://www.sit.auckland.ac.nz/Logging_to_syslog_with_Apache

```
Logging to syslog with Apache
```

First you will need to install syslog-ng. This is the logging server that will send the log data to the syslog box.

```
apt-get update && apt-get install syslog-ng
```

syslog-ng uses a socket device to accept data from apache or whatever program is creating the logs.

Use the configuration here: Syslog-ng default config.

The first part indicates what the socket will be called and where it will live. The second part tells syslog-ng where to send the collected data. The restart syslog-ng (/etc/init.d/syslog-ng restart)l.

Configure apache's logging

Add these directives to send apache's logs via a socket to syslog

```
CustomLog "|/usr/bin/logger -s -t 'monitor.cs.auckland.ac.nz' -p info -u /var/log/apache_log.socket" Combined
ErrorLog "|/usr/bin/logger -s -t 'monitor.cs.auckland.ac.nz' -p err -u /var/log/apache_log.socket"
```

Apache will then use the logger program to send data to syslog. /var/log/apache_log.socket refers to the device that syslog-ng has created. Data sent to this device is sent over the network to the main syslog box.

Troubleshooting

It seems that apache 2.0.54-5 does not like logging to a file and to a process at the same time. In this case log entries will become re-ordered or missed out. You can use the test scripts below to check if this is happening.

Testing

Here are some useful scripts that can help with testing to make sure the logging is working as expected.

You can simulate http accesses using lynx with this command:

```
watch lynx -source http://monitor.cs.auckland.ac.nz/
```

Which will make a http request every two seconds. Or, for a better test:

```
for i in `seq 1 100`; do lynx -source
http://monitor.cs.auckland.ac.nz/$i;sleep 3;done
The result of this test is a sequence of log entires from 1 to
100. If entries are missing or in the wrong order, you know
there is a problem.
```

other

```
CustomLog "|/usr/bin/your_script" Combined
ErrorLog "|/usr/bin/your_script"
```

2.5. mod_access

```
<Directory /www>
  Order Allow,Deny
</Directory>

<Directory /www>
        Order Deny,Allow
        Deny from all
        Allow from apache.org
</Directory>


<Directory /www>
        Order Allow,Deny
        Allow from apache.org
        Deny from foo.apache.org
</Directory>
```

```
A (partial) domain-name
Example: Allow from apache.org
```

```
A full IP address
Example: Allow from 10.1.2.3

A partial IP address
Example: Allow from 10.1

A network/netmask pair
Example: Allow from 10.1.0.0/255.255.0.0

A network/nnn CIDR specification
Example: Allow from 10.1.0.0/16
```

```
<DirectoryMatch (/usr/share/nagios3/htdocs|/usr/lib/cgi-
bin/nagios3|/etc/nagios3/stylesheets)>
        Options FollowSymLinks

        DirectoryIndex index.html

        AllowOverride AuthConfig
        Order Allow,Deny
        Allow From All

        AuthName "Nagios Access"
        AuthType Basic
        AuthUserFile /etc/nagios3/htpasswd.users
        # nagios 1.x:
        #AuthUserFile /etc/nagios/htpasswd.users
        require valid-user
</DirectoryMatch>
```

Apache httpd 2.4.x

```
    <Directory "/www/www.example.com">
                Options Indexes FollowSymLinks
                AllowOverride None
                Require all granted
    </Directory>
```

2.6. VirtualHost

conf/extra/httpd-vhosts.conf

or

/etc/httpd/conf.d/vhost.conf

```
NameVirtualHost *:80

<VirtualHost *:80>
    ServerAdmin webmaster@dummy-host.example.com
    DocumentRoot "/usr/local/httpd-2.2.14/docs/dummy-
host.example.com"
    ServerName dummy-host.example.com
    ServerAlias www.dummy-host.example.com
    ErrorLog "logs/dummy-host.example.com-error_log"
    CustomLog "logs/dummy-host.example.com-access_log" common
</VirtualHost>
```

ServerName/ServerAlias

```
ServerName dummy-host.example.com
ServerAlias www.dummy-host.example.com
```

rotatelogs

```
CustomLog "|/usr/local/httpd/bin/rotatelogs
/www/logs/www.example.com/access.%Y-%m-%d.log 86400 480"
combined
ErrorLog "|/usr/local/httpd/bin/rotatelogs
```

```
/www/logs/www.example.com/error.%Y-%m-%d.log 86400 480"
```

2.7. Alias / AliasMatch

```
Alias /image /ftp/pub/image
AliasMatch ^/icons(.*) /usr/local/apache/icons$1
```

```
cat /etc/httpd/conf.d/logs.conf

Alias /logs "/www/logs"

<Directory "/www/logs">
   Options FollowSymLinks MultiViews Indexes
   AllowOverride None
   Order allow,deny
   Allow from all
#  Order deny,allow
#  Deny from all
#  Allow from 127.0.0.1
#   AuthName "Logs Access"
#   AuthType Basic
#   AuthUserFile /etc/httpd/htpasswd.users
#   Require valid-user
</Directory>
```

2.8. Redirect / RedirectMatch

Redirect

```
Redirect /service http://foo2.example.com/service
Redirect permanent /one http://example.com/two
Redirect 303 /three http://example.com/other
```

```
```

RedirectMatch

```
RedirectMatch (.*)\.gif$ http://www.domain.com$1.jpg
```

```
<VirtualHost *:80>
     ServerName www.old.com
     DocumentRoot /path/to/htdocs
     ......
     <Directory "/path/to/htdocs">
         RedirectMatch ^/(.*)$ http://www.new.com/$1
     </Directory>
</VirtualHost>
```

2.9. Rewrite

Rewrite 需要 AllowOverride All

```
<Directory "/www">
    #
    # Possible values for the Options directive are "None",
"All",
    # or any combination of:
    #   Indexes Includes FollowSymLinks SymLinksifOwnerMatch
ExecCGI MultiViews
    #
    # Note that "MultiViews" must be named *explicitly* ---
"Options All"
    # doesn't give it to you.
    #
    # The Options directive is both complicated and important.
Please see
    # http://httpd.apache.org/docs/2.2/mod/core.html#options
    # for more information.
    #
    Options Indexes FollowSymLinks

    #
    # AllowOverride controls what directives may be placed in
.htaccess files.
    # It can be "All", "None", or any combination of the
keywords:
    #   Options FileInfo AuthConfig Limit
    #
    #AllowOverride None
    AllowOverride All

    #
    # Controls who can get stuff from this server.
    #
    Order allow,deny
    Allow from all

</Directory>
```

R=301

```
RewriteEngine on
RewriteCond %{HTTP_HOST} ^x.x.x.x [NC]
RewriteRule ^/(.*)$ http://www.example.com/$1 [L,R=301]
```

例 4.3. R=301

```
<VirtualHost *:80>
        ServerAdmin webmaster@example.com
        ServerName www.example.com
        ServerAlias www.second.com

    RewriteEngine On
    RewriteCond %{HTTP_HOST} ^www.example.com [NC]
    RewriteRule ^/(.*)$ http://www.other.com/$1 [L,R=301]
    RewriteCond %{HTTP_HOST} ^www.second.com [NC]
    RewriteRule ^/(.*)$ http://www.other.com/$1 [L,R=301]
</VirtualHost>
```

Rewrite + JkMount

JkMount 与 Rewrite 同时使用时

RewriteRule ^/communtiy/top/(.*)$ /community.do?method=activeContent&id=$1 [PT]

后面用[PT]

Apache redirect domain.com to www.domain.com

```
$ vi .htaccess
RewriteEngine on
RewriteCond %{HTTP_HOST} ^domain\.com
RewriteRule ^(.*)$ http://www.domain.com/$1 [R=permanent,L]
```

正则匹配扩展名

```
<VirtualHost *:80>
    ServerAdmin webmaster@example.com
    DocumentRoot "/www/www.example.com/images"
    ServerName images.example.com
    RewriteEngine On
    RewriteRule ^(.+)(jpg|gif|bmp|jpeg|ico|png|css)$
http://images.other.com/$1$2 [R]
    ErrorLog "logs/images.example.com-error.log"
```

```
</VirtualHost>
```

```
<VirtualHost *:80>
        ServerAdmin webmaster@example.com
        ServerName images.example.com
        RewriteEngine On
        RewriteCond %{HTTP_HOST} ^images.example.com [NC]
        RewriteRule ^/(.*) http://images.other.com/$1 [L]
        CustomLog "|/usr/local/httpd/bin/rotatelogs
/www/logs/images/access.%Y-%m-%d.log 100M" common
</VirtualHost>
```

2.10. Proxy

```
ProxyRequests Off

<Proxy *>
        Order deny,allow
        Allow from all
</Proxy>
ProxyPass / http://your.domain.com:8080/
ProxyPassReverse / http://your.domain.com:8080/
```

Reverse proxy

/etc/httpd/conf.d/rails.conf

```
Listen 8080
ProxyRequests Off
<Proxy balancer://cluster>
```

```
        BalancerMember http://127.0.0.1:3001
        BalancerMember http://127.0.0.1:3002
        BalancerMember http://127.0.0.1:3003
        BalancerMember http://127.0.0.1:3004
        BalancerMember http://127.0.0.1:3005
</Proxy>

<VirtualHost *:8080>
        ServerName www.example.com:8080
        DocumentRoot /var/www/project/public
        ProxyPass /images !
        ProxyPass /stylesheets !
        ProxyPass /javascripts !
        ProxyPass / balancer://cluster/
        ProxyPassReverse / balancer://cluster/
        ProxyPreserveHost on
</VirtualHost>
```

2.11. Deflate

mod_deflate

httpd.conf中中加入下列语句：

```
<IfModule mod_deflate.c>
        SetOutputFilter DEFLATE
        DeflateCompressionLevel 9
        AddOutputFilterByType DEFLATE text/html text/plain
text/xml application/x-httpd-php
        AddOutputFilter DEFLATE txt css js
        SetEnvIfNoCase Request_URI \.(?:gif|jpe?g|png)$ no-gzip
dont-vary
        SetEnvIfNoCase Request_URI \.(?:exe|t?
gz|zip|bz2|sit|rar)$ no-gzip dont-vary
        SetEnvIfNoCase Request_URI \.pdf$ no-gzip dont-vary
        DeflateFilterNote Input input_info
        DeflateFilterNote Output output_info
        DeflateFilterNote Ratio ratio_info
        LogFormat '"%r" %{output_info}n/%{input_info}n (%
{ratio_info}n%%)' deflate
        CustomLog logs/deflate_log.log deflate
</IfModule>
```

对目录/usr/local/apache/htdocs有效

```
<Directory "/usr/local/apache/htdocs">
    AllowOverride None
    Options None
    Order allow,deny
    Allow from all
        SetOutputFilter DEFLATE
        DeflateCompressionLevel 9
        AddOutputFilterByType DEFLATE text/html text/plain
text/xml application/x-httpd-php
        AddOutputFilter DEFLATE txt css js
        SetEnvIfNoCase Request_URI \
        \.(?:gif|jpe?g|png)$ no-gzip dont-vary
</Directory>
```

```
<Location />
        AddOutputFilterByType DEFLATE text/html text/plain
text/xml text/css text/javascript
        AddOutputFilterByType DEFLATE application/javascript
application/x-javascript application/x-httpd-php
        AddOutputFilter DEFLATE txt css js
        SetOutputFilter DEFLATE
</Location>
```

Log定义

```
DeflateFilterNote Input instream   # 未压缩前
DeflateFilterNote Output outstream # 压缩后
DeflateFilterNote Ratio ratio   # 百分比
LogFormat '"%r" %{outstream}n/%{instream}n (%{ratio}n%%)'
deflate # 格式定义

CustomLog logs/deflate_log.log deflate # 日志位置
CustomLog "|/usr/local/httpd/bin/rotatelogs
/www/logs/deflate.%Y-%m-%d.log 86400 480" deflate # 分割日志位置
```

测试 gzip,deflate 模块

telnet www.bg7nyt.cn 80

```
GET /index.html HTTP/1.0
Host: www.bg7nyt.cn
Accept-Encoding: gzip,deflate
```

你看到的是乱码,而不是HTML.

```
curl -H Accept-Encoding:gzip,defalte
http://www.example.com/index.html | gunzip
```

gunzip 可以解压压缩内容

2.12. Expires

```
ExpiresActive On
ExpiresByType image/gif "access plus 1 month"
ExpiresByType image/jpeg "access plus 1 month"
ExpiresByType image/x-icon "access plus 1 month"
ExpiresByType image/png "access plus 1 month"
ExpiresByType text/html "access plus 30 minutes"
ExpiresByType text/css  "access plus 30 minutes"
ExpiresByType text/js   "access plus 30 minutes"
ExpiresByType application/x-javascript   "access plus 30
minutes"
```

```
ExpiresByType application/x-shockwave-flash     "access plus 30
minutes"
```

FilesMatch

```
<FilesMatch “\.
(ico|jpg|jpeg|png|gif|js|css|swf|html|htm|gzip)$”>
        ExpiresActive on
        ExpiresDefault “access plus 2 hours”
</FilesMatch>
```

Cache-Control

```
<FilesMatch "\.(gif|jpe?g|png|ico|css|js|swf)$">
        Header set Cache-Control “max-age=1800, public”
        Header set Cache-Control "s-maxage=600"
</FilesMatch>
```

max-age 针对浏览器推送缓存时间

s-maxage 针对代理服务器推送缓存时间

ETag

```
<FilesMatch "\.(gif|jpe?g|png|ico|css|js|swf)$">
        FileETag none
</FilesMatch>

<FilesMatch "\.(gif|jpe?g|png|ico|css|js|swf)$">
```

```
        FileETag MTime
</FilesMatch>
```

禁用ETag， FileETag none

INode 使用文件i-node 做为 etag

MTime 使用修改时间做为etag

Size 使用文件尺寸做为etag

All 相当于 FileETag INode MTime Size

2.13. Cache

htcacheclean -- program for cleaning the disk cache.

mod_disk_cache

```
<IfModule mod_cache.c>
    CacheDefaultExpire 86400
    <ifModule mod_disk_cache.c>
        CacheEnable disk /
        CacheRoot /tmp/apacheCache
        CacheDirLevels 5
        CacheDirLength 5
        CacheMaxFileSize 1048576
        CacheMinFileSize 10
    </ifModule mod_disk_cache.c>
</IfModule mod_cache.c>
```

mod_mem_cache

```
<IfModule mod_cache.c>
    <ifModule mod_mem_cache.c>
        CacheEnable mem /
        MCacheMaxObjectCount 20000
        MCacheMaxObjectSize 1048576
        MCacheMaxStreamingBuffer 65536
        MCacheMinObjectSize 10
        MCacheRemovalAlgorithm GDSF
        MCacheSize 131072
    </ifModule mod_disk_cache.c>
</IfModule mod_cache.c>
```

2.14. usertrack

跟踪用户信息

跟踪用户的cookie,使用log日志文件记录用户的cookie

```
LoadModule usertrack_module modules/mod_usertrack.so

CookieTracking on
CookieDomain .example.com
CookieExpires "10 years"
CookieStyle Cookie

LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-
Agent}i\" %{cookie}n" combined
```

2.15. Charset

Default charset

```
AddCharset UTF-8 .html

AddType 'text/html; charset=UTF-8' html
```

```
AddDefaultCharset UTF-8
```

Files match

```
<FilesMatch "\.(htm|html|css|js)$">
        ForceType 'text/html; charset=UTF-8'
</FilesMatch>

<FilesMatch "\.(htm|html|css|js)$">
        AddDefaultCharset UTF-8
</FilesMatch>
```

Changing the occasional file

```
<Files "example.html">
        AddCharset UTF-8 .html
</Files>

<Files "example.html">
        ForceType 'text/html; charset=UTF-8'
</Files>
```

2.16. Dir

```
<IfModule dir_module>
    DirectoryIndex index.html index.php
</IfModule>
```

2.17. Includes

```
<Directory "/www">
        Options Indexes FollowSymLinks +Includes
</Directory>
```

```
<IfModule mime_module>
        AddType text/html .shtml
    AddOutputFilter INCLUDES .shtml
</IfModule>
```

2.18. Apache Status

开启Apache的status模块，需要修改httpd.conf，增加以下配置段：

```
ExtendedStatus On
<Location /server-status>
  SetHandler server-status
  Order deny,allow
  Deny from all
  Allow from 125.76.229.113
</Location>
```

http://www.domain.com/server-status

Automatic Updates

```
http://your.server.name/server-status?refresh=N
```

```
http://localhost/server-status?auto
```

扩展状态，提供更详细的信息

```
ExtendedStatus On
```

2.19. Mod Perl

ref: http://search.cpan.org/~agrundma/Catalyst-Engine-Apache-1.07/lib/Catalyst/Engine/Apache2/MP20.pm

$ sudo apt-get install libapache2-mod-perl2 $ sudo apt-get install libcatalyst-engine-apache-perl

```
$ sudo vi /etc/apache2/sites-available/catalyst.conf
```

例 4.4. mod_perl.conf

```


PerlSwitches -I/var/www/MyApp/lib
# Preload your entire application
PerlModule MyApp

<VirtualHost 192.168.245.129:80>
        ServerName 192.168.245.129
        DocumentRoot /var/www/MyApp/root

        <Directory /var/www/MyApp/root>
                Options Indexes FollowSymLinks
                AllowOverride None
                Order allow,deny
                Allow from all
        </Directory>

        # If the server is started as:
```

```
        #       httpd -X -D PERLDB
        # then debugging will be turned on
#       <IfDefine PERLDB>
#               PerlRequire conf/db.pl
#               <Location />
#                       PerlFixupHandler Apache::DB
#               </Location>
#       </IfDefine>

        <Location />
                SetHandler modperl
                PerlResponseHandler MyApp
        </Location>

        Alias /static /var/www/MyApp/root/static
        <Location /static>
                SetHandler default-handler
        </Location>
</VirtualHost>
```

db.pl

```
use APR::Pool ();
use Apache::DB ();
Apache::DB->init();
```

enable site

```
$ sudo a2ensite mod_perl.conf
$ sudo /etc/init.d/apache2 restart
```

2.20. mod_pagespeed -

https://developers.google.com/speed/pagespeed/mod

2.21. Module FAQ

```
[root@srv-2 modules]# /etc/init.d/httpd start
Starting httpd: Syntax error on line 358 of
/etc/httpd/conf/httpd.conf:
Invalid command 'Order', perhaps mis-spelled or defined by a
module not included
in the server configuration
[FAILED]
LoadModule access_module /etc/httpd/modules/mod_access.so
LoadModule auth_module /etc/httpd/modules/mod_auth.so
[root@srv-2 modules]# /etc/init.d/httpd start
Starting httpd: Syntax error on line 368 of
/etc/httpd/conf/httpd.conf:
Invalid command 'UserDir', perhaps mis-spelled or defined by a
module not includ
ed in the server configuration
[FAILED]
LoadModule userdir_module /etc/httpd/modules/mod_userdir.so
[root@srv-2 modules]# /etc/init.d/httpd start
Starting httpd: Syntax error on line 396 of
/etc/httpd/conf/httpd.conf:
Invalid command 'DirectoryIndex', perhaps mis-spelled or
defined by a module not
included in the server configuration
[FAILED]
LoadModule dir_module /etc/httpd/modules/mod_dir.so
[root@srv-2 modules]# /etc/init.d/httpd start
Starting httpd: Syntax error on line 419 of
/etc/httpd/conf/httpd.conf:
Invalid command 'TypesConfig', perhaps mis-spelled or defined
by a module not in
cluded in the server configuration
[FAILED]
LoadModule mime_module /etc/httpd/modules/mod_mime.so
[root@srv-2 modules]# /etc/init.d/httpd start
Starting httpd: Syntax error on line 491 of
/etc/httpd/conf/httpd.conf:
Invalid command 'LogFormat', perhaps mis-spelled or defined by
a module not incl
uded in the server configuration
[FAILED]
```

```
LoadModule log_config_module
/etc/httpd/modules/mod_log_config.so
[root@srv-2 modules]# /etc/init.d/httpd start
Starting httpd: Syntax error on line 555 of
/etc/httpd/conf/httpd.conf:
Invalid command 'Alias', perhaps mis-spelled or defined by a
module not included
in the server configuration
[FAILED]
LoadModule alias_module /etc/httpd/modules/mod_alias.so
[root@srv-2 modules]# /etc/init.d/httpd start
Starting httpd: Syntax error on line 582 of
/etc/httpd/conf/httpd.conf:
Invalid command 'SetEnvIf', perhaps mis-spelled or defined by a
module not inclu
ded in the server configuration
[FAILED]
LoadModule setenvif_module /etc/httpd/modules/mod_setenvif.so
[root@srv-2 modules]# /etc/init.d/httpd start
Starting httpd: Syntax error on line 636 of
/etc/httpd/conf/httpd.conf:
Invalid command 'IndexOptions', perhaps mis-spelled or defined
by a module not i
ncluded in the server configuration
[FAILED]
LoadModule autoindex_module /etc/httpd/modules/mod_autoindex.so
[root@srv-2 modules]# /etc/init.d/httpd start
Starting httpd: Syntax error on line 784 of
/etc/httpd/conf/httpd.conf:
Invalid command 'LanguagePriority', perhaps mis-spelled or
defined by a module n
ot included in the server configuration
[FAILED]
LoadModule negotiation_module
/etc/httpd/modules/mod_negotiation.so
[root@srv-2 modules]# /etc/init.d/httpd start
Starting httpd:                                            [
OK  ]
[root@srv-2 modules]#
```

2.22. mod_setenvif

屏蔽爬虫

```
<directory "/www/example.com">
    Order allow,deny
    Allow from all
    BrowserMatchNoCase "iaskspider" badguy
    BrowserMatchNoCase "QihooBot" badguy
    BrowserMatchNoCase "larbin" badguy
    BrowserMatchNoCase "iearthworm" badguy
    BrowserMatchNoCase "Outfoxbot" badguy
    BrowserMatchNoCase "lanshanbot" badguy
    BrowserMatchNoCase "Arthur" badguy
    BrowserMatchNoCase "InfoPath" badguy
    BrowserMatchNoCase "DigExt" badguy
    BrowserMatchNoCase "Embedded" badguy
    BrowserMatchNoCase "EmbeddedWB" badguy
    BrowserMatchNoCase "Wget" badguy
    BrowserMatchNoCase "CNCDialer" badguy
    BrowserMatchNoCase "LWP::Simple" badguy
    BrowserMatchNoCase "WPS" badguy
    deny from env=badguy
</directory>
```

屏蔽下载

```
BrowserMatch "NetAnt" badguy
BrowserMatch "GetRight" badguy
BrowserMatch "JetCar" badguy
BrowserMatch "Mass Downloader" badguy
BrowserMatch "ReGet" badguy
BrowserMatch "DLExpert" badguy
BrowserMatch "FlashGet" badguy
BrowserMatch "Offline Explorer" badguy
BrowserMatch "Teleport" badguy
...........

order deny,allow
deny from env=badguy
```

```
allow from all
```

2.23. PHP 程序安全问题 php_admin_value

php 安全

```
php_admin_value open_basedir /var/www/htdocs/
```

```
<IfModule mod_php5.c>
  php_value include_path ".:/usr/local/lib/php"
  php_admin_flag engine on
</IfModule>
<IfModule mod_php4.c>
  php_value include_path ".:/usr/local/lib/php"
  php_admin_flag engine on
</IfModule>
```

2.24. mod_spdy

mod_spdy 是用于 Apache HTTP 服务器的 Google SPDY 协议实现模块,

SPDY并不是一种用于替代HTTP的协议，而是对HTTP协议的增强。新协议的功能包括数据流的多路复用、请求优先级，以及HTTP包头压缩。谷歌已经开发一个网络服务器原型机，以及支持SPDY协议的Chrome浏览器版本。

https://code.google.com/p/mod-spdy/

3. 设置Apache实现防盗连

```
SetEnvIf Referer "http://news.netkiller.com/" local_referal
SetEnvIf Referer "$" local_referral

Order Deny,Allow
Deny from all
Allow from env=local_referal
```

配置httpd.conf文件

#LoadModule rewrite_module modules/mod_rewrite.so

去掉前面的"#"注释

AllowOverride None

改为

AllowOverride All

配置.htaccess文件

```
RewriteEngine on
RewriteCond % !^http://xxx.cn/.*$       [NC]
RewriteCond % !^http://xxx.cn$                  [NC]
RewriteCond % !^http://www.xxx.cn/.*$   [NC]
RewriteCond % !^http://www.xxx.cn$      [NC]
RewriteRule .*\.(jpg|jpeg|gif|png|bmp|rar|zip|exe)$
http://download.example.com/err.html [R,NC]
```

4. .htaccess

AllowOverride None 改为 AllowOverride All

```
<VirtualHost *:80>
    ServerAdmin neo.chen@live.com
    DocumentRoot "/www/example.com/www.example.com"
    ServerName example.com
    ServerAlias www.example.com
    ErrorLog "logs/www.example.com-error_log"
    CustomLog "logs/www.example.com-access_log" common

    <Directory "/www/example.com/www.example.com">
        Options Indexes FollowSymLinks
        #AllowOverride None
        AllowOverride All
        Require all granted
    </Directory>
    <IfModule dir_module>
        DirectoryIndex index.html index.php
    </IfModule>

    # The following lines prevent .htaccess and .htpasswd files
from being
    # viewed by Web clients.
    #
    <Files ".ht*">
        Require all granted
    </Files>

</VirtualHost>
```

5. Error Prompt

5.1. Invalid command 'Order', perhaps misspelled or defined by a module not included in the server configuration

没有加载 mod_authz_host 模块

```
LoadModule authz_host_module modules/mod_authz_host.so
```

5.2. Invalid command 'AuthUserFile', perhaps misspelled or defined by a module not included in the server configuration

```
LoadModule auth_basic_module
/usr/lib/apache2/modules/mod_auth_basic.so
LoadModule authz_owner_module
/usr/lib/apache2/modules/mod_authz_owner.so
LoadModule authn_file_module
/usr/lib/apache2/modules/mod_authn_file.so
```

第 5 章 Lighttpd

1. 安装Lighttpd

1.1. quick install with aptitude

if you OS is Ubuntu/Debian

apt-get install lighttpd

```
netkiller@shenzhen:~$ sudo apt-get install lighttpd
```

the config file in /etc/lighttpd

```
netkiller@shenzhen:~/document/Docbook/Linux$ find
/etc/lighttpd/
/etc/lighttpd/
/etc/lighttpd/lighttpd.conf
/etc/lighttpd/conf-enabled
/etc/lighttpd/conf-available
/etc/lighttpd/conf-available/10-userdir.conf
/etc/lighttpd/conf-available/10-fastcgi.conf
/etc/lighttpd/conf-available/10-cgi.conf
/etc/lighttpd/conf-available/README
/etc/lighttpd/conf-available/10-ssl.conf
/etc/lighttpd/conf-available/10-proxy.conf
/etc/lighttpd/conf-available/10-auth.conf
/etc/lighttpd/conf-available/10-simple-vhost.conf
/etc/lighttpd/conf-available/10-ssi.conf
```

Enabling and disabling modules could be done by provided e.g.

```
/usr/sbin/lighty-enable-mod fastcgi
```

```
/usr/sbin/lighty-disable-mod fastcgi
```

when you enabled a mod please force-reload it

```
netkiller@shenzhen:/etc/lighttpd$ sudo lighty-enable-mod
fastcgi
Available modules: auth cgi fastcgi proxy simple-vhost ssi ssl
userdir
Already enabled modules: userdir
Enabling fastcgi: ok
Run /etc/init.d/lighttpd force-reload to enable changes
netkiller@shenzhen:/etc/lighttpd$ sudo /etc/init.d/lighttpd
force-reload
 * Stopping web server lighttpd
[ OK ]
 * Starting web server lighttpd
```

1.2. yum install

```
# yum install lighttpd lighttpd-fastcgi -y
# chkconfig lighttpd on
```

创建缓存目录

```
# mkdir -p /var/cache/lighttpd/compress
# chown lighttpd:lighttpd -R /var/cache/lighttpd
```

禁用ipv6

```
# vim /etc/lighttpd/lighttpd.conf

#server.use-ipv6 = "enable"
```

1.3. to compile and then install lighttpd

1. 下载相关软件

   立即下载

```
$ sudo apt-get install libpcre3*

cd /usr/local/src/
wget http://www.lighttpd.net/download/lighttpd-
1.4.15.tar.gz
tar zxvf lighttpd-1.4.15.tar.gz
cd lighttpd-1.4.15
```

2. 编译安装

```
./configure --prefix=/usr/local/lighttpd-1.4.15 \
--with-bzip2 \
--with-memcache
make
make install
```

3. 创建目录与配置文件

```
ln -s /usr/local/lighttpd-1.4.15/ /usr/local/lighttpd
mkdir -p /www/pages
mkdir /www/logs
mkdir /usr/local/lighttpd/htdocs
mkdir /usr/local/lighttpd/logs
mkdir /usr/local/lighttpd/etc
cp ./doc/lighttpd.conf /usr/local/lighttpd/etc/
cd /usr/local/lighttpd/
```

4. 配置lighttpd.conf

vi etc/lighttpd.conf

找到 server.modules

删除 mod_fastcgi 前的注释

跟据你的需求修改下面定义

server.document-root = "/usr/local/lighttpd/htdocs/"

server.errorlog = "/usr/local/lighttpd/logs/lighttpd.error.log"

accesslog.filename = "/usr/local/lighttpd/logs/access.log"

注释 $HTTP["url"]

```
#$HTTP["url"] =~ "\.pdf$" {
#  server.range-requests = "disable"
#}
```

5. 运行lighttpd

```
/usr/local/lighttpd/sbin/lighttpd -f
/usr/local/lighttpd/etc/lighttpd.conf
```

测试

curl http://ip/ 因为/www/pages/下没有HTML页面所以返回:

404 - Not Found

shell script

lighttpd script

例 5.1. /etc/init.d/lighttpd

```
#!/bin/bash
# lighttpd init file for web server
#
# chkconfig: - 100 100
# description: Security, speed, compliance, and flexibility--
all of these describe LightTPD which is rapidly redefining
efficiency of a webserver;
#                                as it is designed and optimized
for high performance environments.
# author: Neo Chen<openunix@163.com>
#
# processname: $PROG
# config:
# pidfile: /var/run/lighttpd

# source function library
. /etc/init.d/functions

PREFIX=/usr/local/lighttpd
PROG=$PREFIX/sbin/lighttpd
OPTIONS="-f /usr/local/lighttpd/etc/lighttpd.conf"
USER=daemon
RETVAL=0
prog="lighttpd"

start() {
        echo -n $"Starting $prog: "
        if [ $UID -ne 0 ]; then
                RETVAL=1
                failure
        else
                daemon --user=$USER $PROG $OPTIONS
                RETVAL=$?
                [ $RETVAL -eq 0 ] && touch
/var/lock/subsys/lighttpd
        fi;
        echo
        return $RETVAL
}

stop() {
        echo -n $"Stopping $prog: "
        if [ $UID -ne 0 ]; then
                RETVAL=1
                failure
        else
                killproc $PROG
                RETVAL=$?
                [ $RETVAL -eq 0 ] && rm -f
/var/lock/subsys/lighttpd
        fi;
        echo
        return $RETVAL
}

reload(){
        echo -n $"Reloading $prog: "
        killproc $PROG -HUP
        RETVAL=$?
        echo
        return $RETVAL
}

restart(){
        stop
        start
}

condrestart(){
    [ -e /var/lock/subsys/lighttpd ] && restart
    return 0
}

case "$1" in
  start)
        start
        ;;
  stop)
        stop
        ;;
  restart)
        restart
        ;;
  reload)
        reload
        ;;
  condrestart)
```

```
        condrestart
        ;;
  status)
        status lighttpd
        RETVAL=$?
        ;;
  *)
        echo $"Usage: $0
{start|stop|status|restart|condrestart|reload}"
        RETVAL=1
esac

exit $RETVAL
```

2. /etc/lighttpd/lighttpd.conf

2.1. max-worker / max-fds

max-worker 我一般设置为与处理器数目相同。

max-fds 最大连接数

```
server.max-worker = 24
server.max-fds = 4096
```

2.2. accesslog.filename

通过cronolog切割日志

```
#### accesslog module
#accesslog.filename          = "/www/logs/lighttpd.access.log"
accesslog.filename = "| /usr/local/sbin/cronolog
/www/logs/%Y/%m/%d/access.log"
```

2.3. ETags

disable etags

```
static-file.exclude-extensions = ( ".php", ".pl", ".fcgi" )
static-file.etags = "disable"
```

2.4. server.tag

隐藏服务器信息

```
server.tag = "Apache"
```

测试结果Server: Apache

```
curl -I http://172.16.0.7/
HTTP/1.1 200 OK
Content-Type: text/html
Content-Length: 4692
Date: Fri, 04 Nov 2011 12:33:19 GMT
Server: Apache
```

3. Module

```
server.modules              = (
#                               "mod_rewrite",
#                               "mod_redirect",
#                               "mod_alias",
                                "mod_access",
#                               "mod_trigger_b4_dl",
#                               "mod_auth",
#                               "mod_status",
#                               "mod_setenv",
#                               "mod_fastcgi",
#                               "mod_proxy",
#                               "mod_simple_vhost",
#                               "mod_evhost",
#                               "mod_userdir",
#                               "mod_cgi",
#                               "mod_compress",
#                               "mod_ssi",
#                               "mod_usertrack",
#                               "mod_expire",
#                               "mod_secdownload",
#                               "mod_rrdtool",
                                "mod_accesslog" )
```

3.1. simple_vhost

```
$ sudo lighty-enable-mod simple-vhost
```

simple-vhost.default-host = "www.example.com"

create your virtual host directory

```
$ mkdir -p /var/www/www.example.com/html
```

create a test file

```
$ echo helloworld!!!> /var/www/www.example.com/html/index.html
```

3.2. ssl

启用 ssl 模块

```
$ sudo lighttpd-enable-mod ssl
[sudo] password for neo:
Available modules: auth cgi fastcgi proxy rrdtool simple-vhost
ssi ssl status userdir
Already enabled modules: cgi fastcgi simple-vhost
Enabling ssl: ok
Run /etc/init.d/lighttpd force-reload to enable changes
```

创建 ssl 证书

```
$ sudo openssl req -new -x509 -keyout server.pem -out
server.pem -days 365 -nodes
$ sudo chmod 400 server.pem
```

3.3. redirect

```


url.redirect                = ( "^/music/(.+)" =>
"http://www.example.org/$1" )


```

301重定向

```
RewriteCond %{HTTP_HOST} ^example\.org$ [NC]
```

```
RewriteRule ^(.*)$ http://www.example.org/$1 [R=301,L]
```

lighttpd 实现上面 apache功能

```
$HTTP["host"] =~ "^example\.org" {
    url.redirect = (
        "^/(.*)$" => "http://www.example.org/$1"
    )
}

$HTTP["host"] =~ "^example\.com$" {
  url.redirect = ( "^/(.*)" => "http://www.example.com/$1" )
}
```

3.4. rewrite

example 1

```
url.rewrite-once = ( "^/wiki/(.*)$" => "/wiki/awki.cgi/$1" )
$HTTP["url"] =~ "^/wiki" {
  $HTTP["url"] !~ "^/wiki/awki.cgi/" {
    url.access-deny = ("")
  }
}
```

example 2

```
$HTTP["host"] =~ "^.*\.(example.org)$" {
  url.rewrite-once = ( "^/(.*)" => "/index.php/$1" )
}
```

example 3

```
$HTTP["host"] =~ "^.*\.(example.org)$" {
  url.rewrite = (
        "^/(images|stylesheet).*" => "/$0",
        "^/(.*)" => "/index.php/$1"
  )
}
```

Lighttpd Rewrite QSA

```
# Apache
RewriteRule ^/index\.html$ /index.php [QSA]
RewriteRule ^/team_(.*)\.html$ /team.php?id=$1 [QSA]

#lighttpd
"^/index\.html(.*)"                     => "/index.php$1",
"^/team_(\w+)\.html\?(.*)"              => "/team.php?
id=$1&$2",
```

ref: http://redmine.lighttpd.net/wiki/lighttpd/MigratingFromApache

```
url.rewrite = (
        "^/index\.html(.*)"                     =>
"/index.php$1",
        "^/index\.html"                         =>
"/index.php",
        "^/team_(.*)\.html"                     => "/team.php?
id=$1",
        "^/team_(\w+)\.html\?(.*)"              => "/team.php?
id=$1&$2"
)
```

3.5. alias

```
$HTTP["host"] =~ "^.*\.(example.org)$" {
    alias.url = (
        "/images" =>
"/home/neo/workspace/Development/photography/application/photog
raphy/images",
        "/stylesheet" =>
"/home/neo/workspace/Development/photography/application/photog
raphy/stylesheet"
    )
}
```

3.6. auth

enable auth

```
$ sudo lighttpd-enable-mod auth
```

/etc/lighttpd/conf-enabled/05-auth.conf

```
$ sudo vim  conf-enabled/05-auth.conf

auth.backend = "plain"
auth.backend.plain.userfile = "/etc/lighttpd/.secret"

auth.require = ( "/tmp/" =>
        (
        "method" => "basic",
        "realm" => "Password protected area",
        "require" => "user=neo"
        )
)
```

create a passwd file

```
$ sudo vim .secret
neo:chen

$ sudo chmod 400 .secret
$ sudo chown www-data /etc/lighttpd/.secret
```

$ sudo /etc/init.d/lighttpd reload

3.7. compress

创建cache目录

```
mkdir -p /var/cache/lighttpd/compress
```

配置lighttpd.conf文件

找到server.modules列表,去掉"mod_compress"注释,再打开compress module的注释

```
#### compress module
compress.cache-dir          = "/var/lighttpd/cache/compress/"
compress.filetype           = ("text/plain", "text/html")
```

Compressing Dynamic Content

php.ini

```
zlib.output_compression = On
zlib.output_handler = On
```

最后使用telnet测试

telnet www.bg7nyt.cn 80

```
GET /index.html HTTP/1.0
Host: 10.10.100.183
Accept-Encoding: gzip,deflate
```

看到乱码输出,而非HTML,表示配置成功.

例 5.2. lighttpd compress

```
$HTTP["host"] =~ "www\.example\.com$" {

        compress.cache-dir = "/www/compress/"
        compress.filetype = ("text/plain", "text/html",
"application/x-javascript", "text/css",
"application/javascript", "text/javascript")

        $HTTP["url"] =~ "(\.png|\.css|\.js|\.jpg|\.gif)$" {
                expire.url = (""=>"access 30 seconds")
        }
}
```

3.8. expire

<access|modification> <number> <years|months|days|hours|minutes|seconds>

```
expire.url = ( "/images/" => "access 1 hours" )
```

Example to include all sub-directories:

```
$HTTP["url"] =~ "^/images/" {
        expire.url = ( "" => "access 1 hours" )
}
```

例 5.3. lighttpd expire

```
$HTTP["host"] =~ "www\.example\.com$" {
        $HTTP["url"] =~ "(\.png|\.css|\.js|\.jpg|\.gif)$" {
                expire.url = (""=>"access 30 seconds")
        }
}
```

3.9. status

```
$ sudo lighty-enable-mod status
$ sudo /etc/init.d/lighttpd force-reload
```

3.10. setenv

```
$HTTP["url"] =~ "^/(.*)" {
        setenv.add-response-header = ( "Cache-Control" => "no-
store, no-cache, must-revalidate, post-check=0, pre-check=0,
max-age=-1" )
}

$HTTP["url"] =~ ".swf" {
       setenv.add-response-header  = ("Pragma" =>"no-
cache","Expires" => "-1")
}

$HTTP["url"] =~ ".swf" {
       setenv.add-response-header  = ("Cache-Control" =>"max-
age=0")
}

$HTTP["url"] =~ ".html" {
       setenv.add-response-header  = ("Cache-Control" =>"s-
maxage=3600")
}

$HTTP["url"] =~ ".css" {
        setenv.add-response-header = (
      "Content-Encoding" => "gzip"
    )
}
```

Automatic Decompression

```
  $HTTP["url"] =~ "(README|ChangeLog|\.txt)\.gz$" {
    setenv.add-response-header  = ( "Content-Encoding" =>
"gzip")
    mimetype.assign = ("" => "text/plain" )
  }
```

3.11. fastcgi

enable fastcgi

enable fastcgi

```
$ sudo lighty-enable-mod fastcgi
```

spawn-fcgi

```


#### fastcgi module
## read fastcgi.txt for more info
## for PHP don't forget to set cgi.fix_pathinfo = 1 in the
php.ini
fastcgi.server             = ( ".php" =>
                               ( "localhost" =>
                                 (
                                   "socket" => "/tmp/php-
fastcgi.socket",
                                   "bin-path" =>
"/usr/local/bin/php-cgi",
                                   "max-procs" => 16,
                                   "bin-environment" => (
                                      "PHP_FCGI_CHILDREN" =>
"128",
                                       "PHP_FCGI_MAX_REQUESTS"
=> "1000"
                                    ),
                                    "broken-scriptfilename" =>
"enable"
                                  )
                                )
                             )

fastcgi.server    = ( ".php" =>
        ((
                "bin-path" => "/usr/bin/php-cgi",
                "socket" => "/tmp/php.socket",
                "max-procs" => 2,
                "idle-timeout" => 200,
                "bin-environment" => (
                        "PHP_FCGI_CHILDREN" => "10",
                        "PHP_FCGI_MAX_REQUESTS" => "10000"
                ),
                "bin-copy-environment" => (
                        "PATH", "SHELL", "USER"
                ),
                "broken-scriptfilename" => "enable"
        ))
)
```

php-fpm

```
fastcgi.server = ( ".php" =>
  ( "localhost" =>
    (
      "host" => "127.0.0.1",
      "port" => "9000"
    )
  )
)
```

PHP

编译安装PHP

1. 下载PHP

```
cd /usr/local/src/
wget http://cn2.php.net/get/php-
5.2.3.tar.bz2/from/cn.php.net/mirror
tar jxvf php-5.2.3.tar.bz2
cd php-5.2.3
```

2. configure

```
./configure --prefix=/usr/local/php-5.2.3 \
--with-config-file-path=/usr/local/php-5.2.3/etc \
--enable-fastcgi \
--enable-force-cgi-redirect \
--with-curl \
--with-gd \
--with-ldap \
--with-snmp \
--enable-zip \
--enable-exif \
--with-pdo-mysql \
--with-pdo-pgsql \

make
make test
make install
```

其它有用的模块

```
--enable-pcntl
```

```
```

3. 符号连接

```
ln -s /usr/local/php-5.2.3 /usr/local/php
ln -s /usr/local/php/bin/php /usr/local/bin/php
```

4. php.ini

```
cp php.ini-dist /usr/local/php/etc/php.ini
```

5. env

```
PHP_FCGI_CHILDREN=384
```

6. 使用 php -v FastCGI 安装情况

   php -v

   显示(cgi-fcgi)表示正确

```
# cd /usr/local/php/
# bin/php -v
PHP 5.2.2 (cgi-fcgi) (built: May 25 2007 15:50:28)
Copyright (c) 1997-2007 The PHP Group
Zend Engine v2.2.0, Copyright (c) 1998-2007 Zend
Technologies
```

   (cgi-fcgi)不能正常工作

```
PHP 5.2.2 (cli) (built: May 25 2007 15:50:28)
Copyright (c) 1997-2007 The PHP Group
Zend Engine v2.2.0, Copyright (c) 1998-2007 Zend
```

```
Technologies
```

使用 php -m 查看PHP Modules

```
# bin/php -m
[PHP Modules]
cgi-fcgi
ctype
date
dom
filter
gd
hash
iconv
json
ldap
libxml
mssql
pcre
PDO
pdo_mysql
pdo_sqlite
posix
Reflection
session
SimpleXML
snmp
SPL
SQLite
standard
tokenizer
xml
xmlreader
xmlwriter
zip

[Zend Modules]
```

apt-get install

```
$ sudo apt-get install php5 php5-cli php5-cgi
```

参考php安装

找到 fastcgi.server 去掉注释

bin-path 改为PHP程序安装目录

```
fastcgi.server             = ( ".php" =>
                               ( "localhost" =>
                                 (
                                  "socket" => "/tmp/php-
fastcgi.socket",
                                   "bin-path" =>
"/usr/local/php/bin/php"
                                 )
                               )
                            )
```

下面例子更复杂一些

1. /usr/local/lighttpd/etc/lighttpd.conf

```
include /usr/local/lighttpd/etc/php-fastcgi.conf
```

2. /usr/local/lighttpd/etc/php-fastcgi.conf

```
fastcgi.server = ( ".php" =>
        ( "localhost" =>
        ( "socket" => "/tmp/php-fastcgi.socket",
           "bin-path" => "/usr/local/php/bin/php",
           "min-procs" => 1,
           "max-procs" => 5,
           "max-load-per-proc" => 4,
                "idle-timeout" => 20
            )
            )
)
```

3. PHP FastCGI环境测试

echo "<?php phpinfo(); ?>" > /www/pages/index.php

curl http://127.0.0.1/index.php

Python

```
sudo apt-get install python
sudo apt-get install python-setuptools
```

Django

```
wget http://www.djangoproject.com/download/0.96/tarball/
tar zxvf Django-0.96.tar.gz
cd Django-0.96
python setup.py install
```

生成项目

```
django-admin.py startproject newtest
```

web server

```
cd newtest/
./manage.py runserver
```

helloworld.py

```
from django.http import HttpResponse

def index(request):
    return HttpResponse("Hello, Django.")
```

urls.py

```
from django.conf.urls.defaults import *

urlpatterns = patterns('',
    # Example:
    # (r'^newtest/', include('newtest.foo.urls')),
    (r'^$', 'newtest.helloworld.index'),

    # Uncomment this for admin:
#     (r'^admin/', include('django.contrib.admin.urls')),
)
```

启动Web Server

```
# ./manage.py runserver
Validating models...
0 errors found.

Django version 0.96, using settings 'newtest.settings'
Development server is running at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

curl http://127.0.0.1:8000/

Python Imaging Library

Debian/Ubuntu

```
sudo apt-get install libjpeg62-dev
sudo apt-get install python-imaging
```

采用源码安装

```
tar zxvf Imaging-1.1.6.tar.gz
cd Imaging-1.1.6/
```

sudo python setup.py install

decoder jpeg not available

首先确认jpeg库是否安装

```
find / -name jpeglib.h
```

然后修改头文件

```
Imaging-1.1.6/libImaging
```

修改Jpeg.h, #include "jpeglib.h" 改为

```
#include "/usr/include/jpeglib.h"
```

Perl

install fastcgi module

```
$ sudo apt-get install libfcgi-perl     libfcgi-procmanager-
perl
```

Installing lighttpd and FastCGI for Catalyst

The examples also use a virtual host regexp that matches either www.myapp.com or myapp.com

```
$HTTP["host"] =~ "^(www.)?mysite.com"
```

Starting the FastCGI server

```
MyApp/script/myapp_fastcgi.pl -l /tmp/myapp.socket -n 5 -d
```

lighttpd.conf

```
server.document-root = "/var/www/MyApp/root"
```

$ sudo vim /etc/lighttpd/conf-available/10-fastcgi.conf

```
fastcgi.server = (
    "" => (
        "MyApp" => (
            "socket" => "/tmp/myapp.socket",
            "check-local" => "disable"
        )
    )
)
```

restart lighttpd

```
neo@master:~$ sudo /etc/init.d/lighttpd restart
 * Stopping web server lighttpd                   [ OK ]
 * Starting web server lighttpd                   [ OK ]
```

Testing

http://127.0.0.1/

More advanced configuration

例 5.4. fastcgi.conf

```
fastcgi.server = (
    "" => (
        "MyApp" => (
            "socket"       => "/tmp/myapp.socket",
            "check-local"  => "disable",
            "bin-path"     =>
"/var/www/MyApp/script/myapp_fastcgi.pl",
            "min-procs"    => 2,
            "max-procs"    => 5,
            "idle-timeout" => 20
        )
    )
)
```

Ruby

UNIX domain sockets

php-fpm.conf

```
listen = /var/run/fastcgi.socket
```

nginx 配置

```
    location ~ /index.php/ {
        root           /www/example.com/api.example.com/htdocs;
        fastcgi_pass   unix:/var/run/fastcgi.socket;
        fastcgi_index  index.php;
        fastcgi_param  SCRIPT_FILENAME
/www/example.com/api.example.com/htdocs$fastcgi_script_name;
        include        fastcgi_params;
    }
```

3.12. user-agent

```
$HTTP["user-agent"] =~
"Googlebot|Sosospider+|eMule|Wget|^Java|^PHP|Ruby|Python" {
  url.rewrite = ( "^/(.*)" => "/crawler.html" )
}
```

```
$HTTP["user-agent"] =~ "Baiduspider+" {
    connection.delay-seconds = 10
}
```

3.13. spdy

```
server {
    listen 443 ssl spdy;

    ssl_certificate server.crt;
    ssl_certificate_key server.key;
    ...
}
```

4. 其他模块

4.1. mod_secdownload 防盗链

5. Example

5.1. s-maxage

s-maxage 头作用于反向代理服务器

例 5.5. Cache

```
$HTTP["url"] =~ "^/images/2010" {
        expire.url = ( "" => "access 15 minutes" )
}

$HTTP["host"] =~ "(img1|img2|img3)\.example\.com" {
    expire.url = ( "" => "access 15 minutes" )
    setenv.add-response-header  = ("Cache-Control" =>"s-
maxage=3600")
}
```

第 6 章 Resin

http://www.caucho.com

1. 安装Resin

JRE

```
$ sudo apt-get install sun-java6-jre
```

下载Resin

注意: Resin Pro 与 Resin 前者要Licence

1.1. 直接使用

简易安装，直接解压缩后即可使用

```
$ wget http://www.caucho.com/download/resin-4.0.1.tar.gz
$ tar zxvf resin-4.0.1.tar.gz
$ sudo mv resin-4.0.1 ..
$ cd ..
$ sudo ln -s resin-4.0.1 resin
```

1.2. Debian/Ubuntu

```
$ wget http://www.caucho.com/download/resin_4.0.1-i386.deb
```

安装 Resin

```
$ sudo dpkg -i resin_4.0.1-i386.deb
```

1.3. 源码安装Resin

源码安装

```
$ cd /usr/local/src/
$ wget http://www.caucho.com/download/resin-4.0.1.tar.gz
$ tar zxvf resin-4.0.1.tar.gz
$ ./configure --prefix=/usr/local/resin-4.0.1 \
--with-apxs=/usr/local/httpd/bin/apxs \
--with-java-home=/usr/local/java \
--enable-64bit \
--enable-lfs \
--enable-ssl \
--enable-debug
$ make && make install
$ cd ..
$ sudo ln -s resin-4.0.1 resin
```

设置 resin 以服务的形式开机自启动

```
$ sudo cp /usr/local/resin/contrib/init.resin /etc/init.d/resin
$ sudo chmod 755 /etc/init.d/resin
$ sudo update-rc.d resin defaults 99
```

2. Compiling mod_caucho.so

```
unix> ./configure --with-apxs=/usr/local/apache/bin/apxs
unix> make && make install
```

```
#
# mod_caucho Resin Configuration
#
LoadModule caucho_module
/usr/local/apache/modules/mod_caucho.so
ResinConfigServer localhost 6802
CauchoConfigCacheDirectory /tmp
CauchoStatus yes
<Location /caucho-status>
  SetHandler caucho-status
</Location>
```

```
<IfModule mod_caucho.c>
ResinConfigServer localhost 6802
<Location /caucho-status>
SetHandler caucho-status
</Location>
</IfModule>

AddHandler caucho-request jsp
<Location /servlet/*>
SetHandler caucho-request
</Location>


<IfModule mod_caucho.c>
        <LocationMatch (.*?)\.action>
```

```
                SetHandler caucho-request
        </LocationMatch>
        <LocationMatch (.*?)\.jsp>
                SetHandler caucho-request
        </LocationMatch>
        <LocationMatch (.*?)\.do>
                SetHandler caucho-request
        </LocationMatch>
</IfModule>
```

3. resin.conf

3.1. Maximum number of threads

Maximum number of threads.

```
<thread-max>4096</thread-max>
```

thread-max数值需要使用ab命令做压力测试，逐步调整。

3.2. Configures the keepalive

```
      <!-- Configures the keepalive -->
      <keepalive-max>128</keepalive-max>
      <keepalive-timeout>15s</keepalive-timeout>
```

3.3. ssl

```
<http address="*" port="443">
  <openssl>
  <certificate-
file>/srv/keys/example.com/star.example.com.crt</certificate-
file>
  <certificate-key-
file>/srv/keys/example.com/star.example.com.key</certificate-
key-file>
  <password>4fff74da-aea4-a9fc-4b5f-e6d497588726</password>
  </openssl>
</http>
```

自颁发证书，首先是使用keytool工具安装证书

```
生成证书:
keytool –genkeypair –keyalg RSA –keysize 2048 SHA1withRSA –
validity 3650  -alias neo –keystore server.keystore –storepass
password –dname "CN=www.example.com, OU=test, O=example.com,
L=SZ, ST=GD, C=CN"

导出证书
-keytool –exportcert –alias neo –keystore server.keystore –
storepass password  –file server.cer –rfc

打印证书
Keytool -printcert –file server.cer

导出证书签发申请
Keytool –certreg –aias neo –keystore server.keystore –storepass
password –file ins.csr –v

导入证书
Keytool –importcert –trustcacerts –alias neo –file server.cer –
keystore server.keystore –storepass password

查看数字证书
Keytool -list

当成功的导入了证书以后就要容器中进行配置才可以使用
首先是要把证书中的那个 server.keystore 和 server.cer这两个文件放入到
Resin服务器的keys这个文件夹中 如果没有的话 就手动的建立这个文件夹
然后去 config 文件夹下配置你的配置文件
我在resin 这个容器中的配置如下

<http address="*" port="443">
    <jsse-ssl>
       <key-store-file>keys/server.keystore</key-store-file>
       <password>password</password>
    </jsse-ssl>
</http>
```

4. virtual hosts

4.1. explicit host

例 6.1. explicit host in resin.conf

```
<resin xmlns="http://caucho.com/ns/resin">
<cluster id="">

<host host-name="www.foo.com">
  <host-alias>foo.com</host-alias>
  <host-alias>web.foo.com</host-alias>

  <root-directory>/opt/www/www.foo.com</root-directory>

  <web-app id="/" document-directory="webapps/ROOT">

  </web-app>
  ...
</host>

</cluster>
</resin>
```

4.2. regexp host

例 6.2. regexp host in resin.conf

```
<resin xmlns="http://caucho.com/ns/resin">
<cluster id="">

<host regexp="([^.]+)\.foo\.com">
  <host-name>${host.regexp[1]}.foo.com</host-name>

  <root-
```

```
directory>/var/www/hosts/www.${host.regexp[1]}.com</root-
directory>

  ...
</host>

</cluster>
</resin>
```

4.3. host-alias

例 6.3. host-alias in the resin.conf

```
<resin xmlns="http://caucho.com">
<cluster id="">

  <host id="www.foo.com" root-directory="/var/www/foo.com">
    <host-alias>foo.com</host-alias>

    <web-app id=""/>
  </host>

</cluster>
</resin>
```

例 6.4. host-alias in a /var/www/hosts/foo/host.xml

```
<host xmlns="http://caucho.com">

  <host-name>www.foo.com</host-name>
  <host-alias>foo.com</host-alias>

  <web-app id="" root-directory="htdocs"/>

</host>
```

例 6.5. host-alias-regexp in the resin.conf

```
<resin xmlns="http://caucho.com">
<cluster id="">

  <host id="www.foo.com" root-directory="/var/www/foo.com">
    <host-alias-regexp>.*foo.com</host-alias-regexp>

    <web-app id=""/>
  </host>

</cluster>
</resin>
```

4.4. configures a deployment directory for virtual hosts

```
<resin xmlns="http://caucho.com/ns/resin">
  <cluster id="app-tier">
    <root-directory>/var/www</root-directory>

    <host-deploy path="hosts">
      <host-default>
        <resin:import path="host.xml" optional="true"/>

        <web-app-deploy path="webapps"/>
      </host-default>
    </host-deploy>
  </cluster>
</resin>
```

$RESIN_HOME/hosts其下的任何目录将对应一个虚拟主机。在$RESIN_HOME/hosts下也可以放置jar文件，其会被展开变成一个虚拟主机。

```
$RESIN_HOME/hosts/www.example.com
$RESIN_HOME/hosts/www.example.net
$RESIN_HOME/hosts/www.example.org
```

4.5. Resources

例 6.6. shared database in host

```
<resin xmlns="http://caucho.com/ns/resin">
  <cluster id="app-tier">
     <server id="a" .../>

     <host id="www.foo.com">
        <database jndi-name="jdbc/test">
            <driver type="org.postgresql.Driver">
                <url>jdbc:postgresql://localhost/test</url>
                <user>caucho</user>
            </driver>
        </database>

        <web-app-default path="webapps"/>
    </host>
  </cluster>
</resin>
```

Oracle JDBC

```
<database>
        <jndi-name>jdbc/test</jndi-name>
        <driver
type="oracle.jdbc.pool.OracleConnectionPoolDataSource">
        <url>jdbc:oracle:thin:@172.16.0.1:1521:database</url>
        <user>user</user>
        <password>password</password>
        </driver>
        <prepared-statement-cache-size>8</prepared-statement-
cache-size>
        <max-connections>1024</max-connections>
        <max-idle-time>20s</max-idle-time>
</database>
```

例 6.7. rewrite-dispatch

```
<resin xmlns="http://caucho.com/ns/resin">
  <cluster id="app-tier">

    <host host-name="www.foo.com">
      <rewrite-dispatch>
        <redirect regexp="^/foo" target="/index.php?foo="/>
      </rewrite-dispatch>
    </host>

  </cluster>
</resin>
```

5. FAQ

5.1. java.lang.OutOfMemoryError: PermGen space

```
vim /usr/local/resin/conf/resin.conf

<jvm-arg>-XX:PermSize=128M</jvm-arg>
<jvm-arg>-XX:MaxPermSize=512m</jvm-arg>
```

第 7 章 Application Server

1. Zope

参考Python安装

1. 下载 Zope-3

```
wget http://www.zope.org/Products/Zope3/3.3.1/Zope-
3.3.1.tgz
tar zxvf Zope-3.3.1.tgz
cd cd Zope-3.3.1
```

2. configure

```
./configure --prefix=/usr/local/Zope --with-
python=/usr/local/python2.4/bin/python

make
make check
make install
```

3. 创建一个Zope实例

```
cd /usr/local/Zope
./bin/mkzopeinstance -u neo:chen -d /usr/local/Zope/webapps
cd webapps
./bin/runzope
```

4. 测试

```
http://netkiller.8800.org:8080/
```

2. JBoss - JBoss Enterprise Middleware

参考Java安装

1. 下载安装 JBoss

```
cd /usr/local/src/
wget
http://nchc.dl.sourceforge.net/sourceforge/jboss/jboss-
5.0.0.Beta2.zip
unzip jboss-5.0.0.Beta2.zip
mv jboss-5.0.0.Beta2 ..
cd ..
ln -s jboss-5.0.0.Beta2 jboss
```

2. 运行 Jboss

```
cd jboss/bin
chmod +x *.sh
./run.sh
```

第 8 章 Web Server Optimization

系统配置

1. Intel(R) Xeon(TM) CPU 3.00GHz
2. Memory 4G
3. Ethernet adapter 1000M

1. ulimit

查看 ulimit

```
ulimit -a
core file size          (blocks, -c) 0
data seg size           (kbytes, -d) unlimited
file size               (blocks, -f) unlimited
pending signals                 (-i) 1024
max locked memory       (kbytes, -l) 32
max memory size         (kbytes, -m) unlimited
open files                      (-n) 1024
pipe size            (512 bytes, -p) 8
POSIX message queues     (bytes, -q) 819200
stack size              (kbytes, -s) 2048
cpu time               (seconds, -t) unlimited
max user processes              (-u) 77824
virtual memory          (kbytes, -v) unlimited
file locks                      (-x) unlimited
```

1.1. open files

对于linux系统，所有设备都以映射为设备文件的方式存在，包括硬件（键盘，鼠标，打印机，显示器，串口，并口，USB，硬盘，内存，网卡，声卡，显卡，等等....），还有软件(管道，socket)，访问这些资源，就相当与打开一个文件，

所以"open files"文件数限制很重要，默认值根本不能满足我们。

查看文件打开数

```
$ cat /proc/sys/fs/file-nr

3200    0       197957
已分配文件句柄的数目       已使用文件句柄的数目       文件句柄的最大数目

查看所有进程的文件打开数
    lsof |wc -l
查看某个进程打开的文件数
        lsof -p pid |wc -l
```

临时更改

```
# ulimit -n 65536
or
# ulimit -SHn 65536
or
# echo "65535" > /proc/sys/fs/file-max
```

永久更改

/etc/security/limits.conf

```
nobody          soft    nofile          40960
root            soft    nofile          40960
nobody          hard    nofile          40960
root            hard    nofile          40960
daemon          soft    nofile          40960
daemon          hard    nofile          40960
```

更省事的方法

```
*           soft    nofile          40960
*           hard    nofile          40960
```

最大线程数限制 threads-max

查看当前值

```
# cat /proc/sys/kernel/threads-max
32624
```

设置

有多种方法加大Linux的threads数，下买是临时更改

```
sysctl -w kernel.threads-max=65536
echo 65536 > /proc/sys/kernel/threads-max
```

永久修改

```
编辑/etc/sysctl.conf
增加
kernel.threads-max = 65536
#sysctl -p 马上生效
```

以上数值仅供参考，随着计算机发展，上面的值已经不太适合，当前流行的服务器。

2. khttpd

homepage: http://www.fenrus.demon.nl

3. php.ini

3.1. Resource Limits

Resource Limits

```
;;;;;;;;;;;;;;;;;;;
; Resource Limits ;
;;;;;;;;;;;;;;;;;;;

max_execution_time = 30     ; Maximum execution time of each
script, in seconds
max_input_time = 60 ; Maximum amount of time each script may
spend parsing request data
;max_input_nesting_level = 64 ; Maximum input variable nesting
level
memory_limit = 512M      ; Maximum amount of memory a script
may consume (16MB)
```

3.2. File Uploads

```
;;;;;;;;;;;;;;;;
; File Uploads ;
;;;;;;;;;;;;;;;;

; Whether to allow HTTP file uploads.
file_uploads = On

; Temporary directory for HTTP uploaded files (will use system
default if not
; specified).
;upload_tmp_dir =

; Maximum allowed size for uploaded files.
upload_max_filesize = 5M
```

3.3. Session Shared

编辑 php.ini 在 [Session]位置添加。

```
extension=memcache.so
memcache.allow_failover = 1
memcache.max_failover_attempts = 20
memcache.chunk_size = 8192
memcache.default_port = 11211

session.save_handler = memcache
session.save_path =
"udp://172.16.0.10:11211,tcp://172.16.0.11:11211"
```

3.4. PATHINFO

```
cgi.fix_pathinfo=1
```

4. APC Cache (php-apc - APC (Alternative PHP Cache) module for PHP 5)

```
$ apt-cache search php-apc
php-apc - APC (Alternative PHP Cache) module for PHP 5

$ sudo apt-get install php-apc
```

apc cache 状态监控

http://pecl.php.net/package/APC

下载解包找到apc.php,放到web服务器上

5. Zend Optimizer

http://www.zend.com/

```
tar zxvf ZendOptimizer-3.2.8-linux-glibc21-i386.tar.gz
cd ZendOptimizer-3.2.8-linux-glibc21-i386
./install
```

过程 8.1. 安装 Zend Optimizer

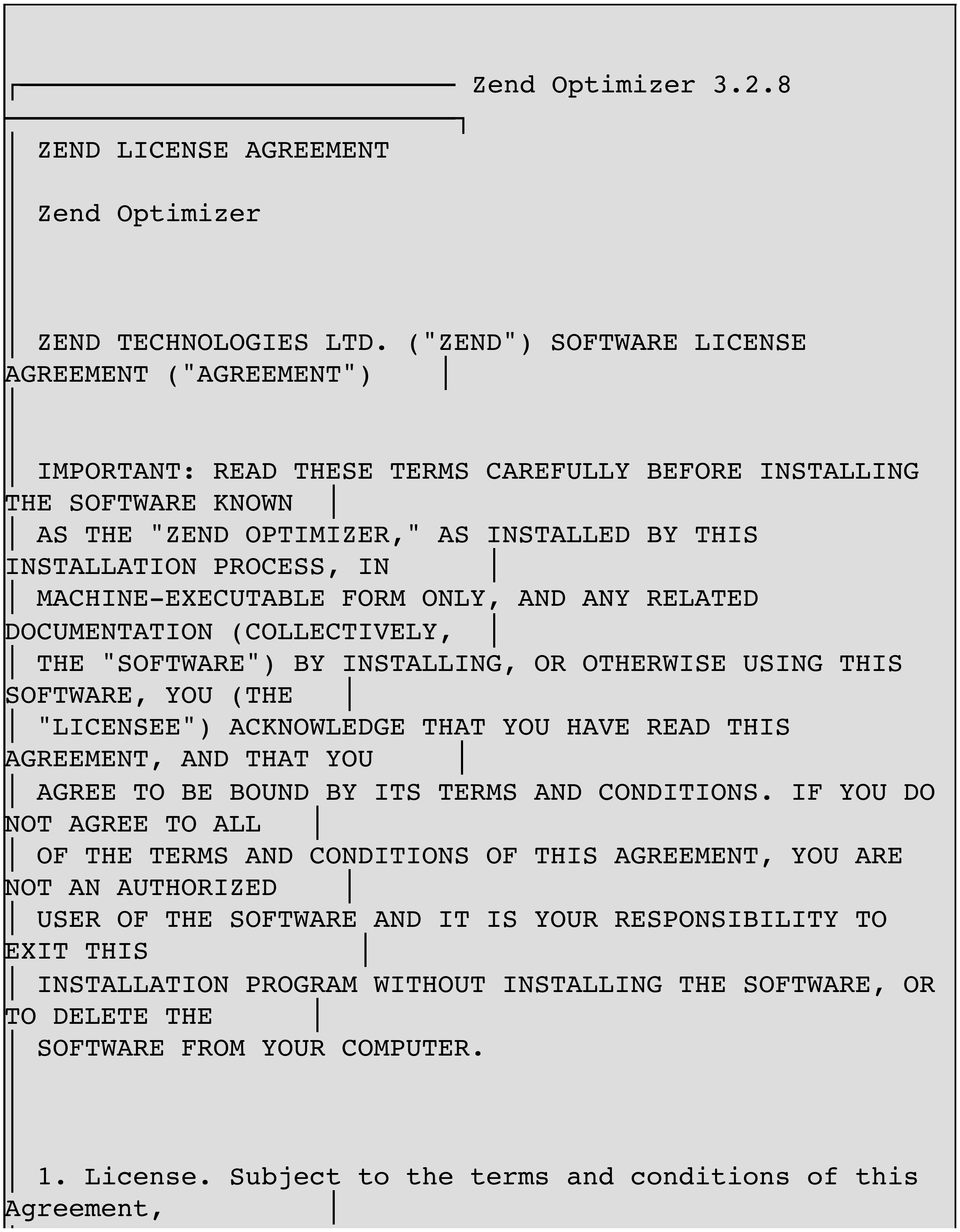
1. 欢迎界面

```
┌──────────────────── Zend Optimizer 3.2.8
─────────────────────┐
│

│  Welcome to the Zend Optimizer 3.2.8 Installation!


│  For more information regarding this procedure, please see
the │
│  Zend Optimizer Installation Guide.



│
├────────────────────────────────────────────────────────────
─────┤
│                           <  OK  >
│
└────────────────────────────────────────────────────────────
─────┘
```

单击 < OK > 按钮

2. LICENSE

Page Down / Page Up 阅读



```
┌──────────────────────────── Zend Optimizer 3.2.8
─────────────────────────────┐
│ ZEND LICENSE AGREEMENT

│ Zend Optimizer
│
│
│
│ ZEND TECHNOLOGIES LTD. ("ZEND") SOFTWARE LICENSE
AGREEMENT ("AGREEMENT")     │
│
│
│ IMPORTANT: READ THESE TERMS CAREFULLY BEFORE INSTALLING
THE SOFTWARE KNOWN   │
│ AS THE "ZEND OPTIMIZER," AS INSTALLED BY THIS
INSTALLATION PROCESS, IN         │
│ MACHINE-EXECUTABLE FORM ONLY, AND ANY RELATED
DOCUMENTATION (COLLECTIVELY,   │
│ THE "SOFTWARE") BY INSTALLING, OR OTHERWISE USING THIS
SOFTWARE, YOU (THE    │
│ "LICENSEE") ACKNOWLEDGE THAT YOU HAVE READ THIS
AGREEMENT, AND THAT YOU        │
│ AGREE TO BE BOUND BY ITS TERMS AND CONDITIONS. IF YOU DO
NOT AGREE TO ALL    │
│ OF THE TERMS AND CONDITIONS OF THIS AGREEMENT, YOU ARE
NOT AN AUTHORIZED     │
│ USER OF THE SOFTWARE AND IT IS YOUR RESPONSIBILITY TO
EXIT THIS              │
│ INSTALLATION PROGRAM WITHOUT INSTALLING THE SOFTWARE, OR
TO DELETE THE       │
│ SOFTWARE FROM YOUR COMPUTER.
│
│
│
│ 1. License. Subject to the terms and conditions of this
Agreement,           │
```

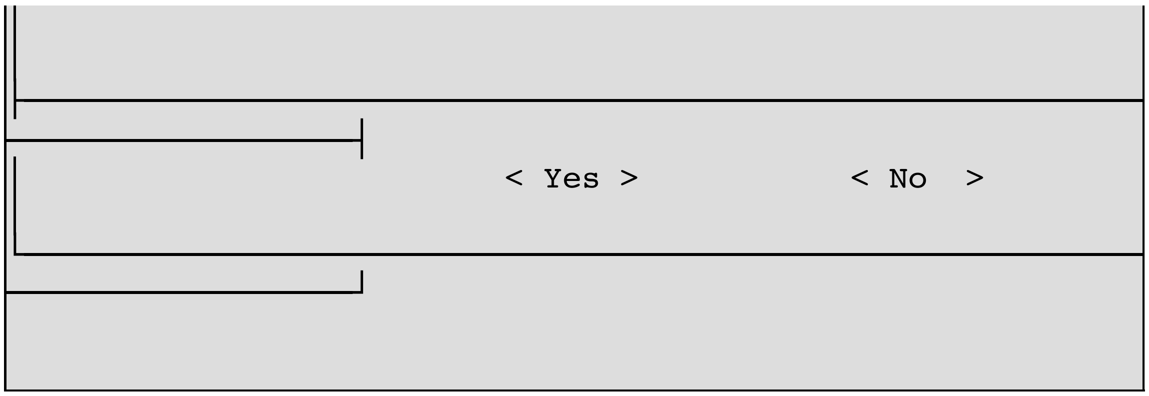
```
│ including, without limitation, Section 2 hereof, Zend
hereby grants to      │
│ Licensee, during the Term (as defined below), a limited,
a non-exclusive    │
│ license (the "License") to: (i) install and operate the
Software on a       │
│ computer or a computer network owned or operated by
Licensee; (ii) make     │
│ copies of the Software; and (iii) sublicense and
distribute a limited,      │
│ non-exclusive sublicense to install, use and sublicense
such copies of the  │
│ Software, provided that any sub-license granted hereunder
shall be subject  │
│ to the limitations and restrictions set forth in this
Agreement.            │
│
│
│ 2. Restrictions. Except as otherwise expressly set forth
herein, Licensee   │
│ or any of its sub-licensees shall not: (a) translate or
decompile, or       │
│ create or attempt to create, by reverse engineering or
otherwise, the       │
│ source code form from the object code supplied hereunder;
(b) modify,       │
│ adapt, translate or create a derivative work from the
Software; (c) remove  │
│ any proprietary notices, labels, or marks on the
Software.                  │
│
│
│ 3. Termination. This Agreement and the License hereunder
shall be in        │
│ effect from and after the date Licensee installs the
Software on a          │
│ computer in accordance with the terms and conditions
hereof and shall       │
│ continue perpetually unless terminated in accordance with
this Section 3.   │
│ This Agreement shall be automatically terminated upon any
breach by         │
│ Licensee of any term or condition of this Agreement. Such
period shall be   │
│
```

```
──────────( 21%)──┤
│                                         < EXIT >
│
└───────────────────────────────────────────────────────────
──────────────────┘

```

单击 < EXIT > 按钮

3. 是否接受LICENSE?

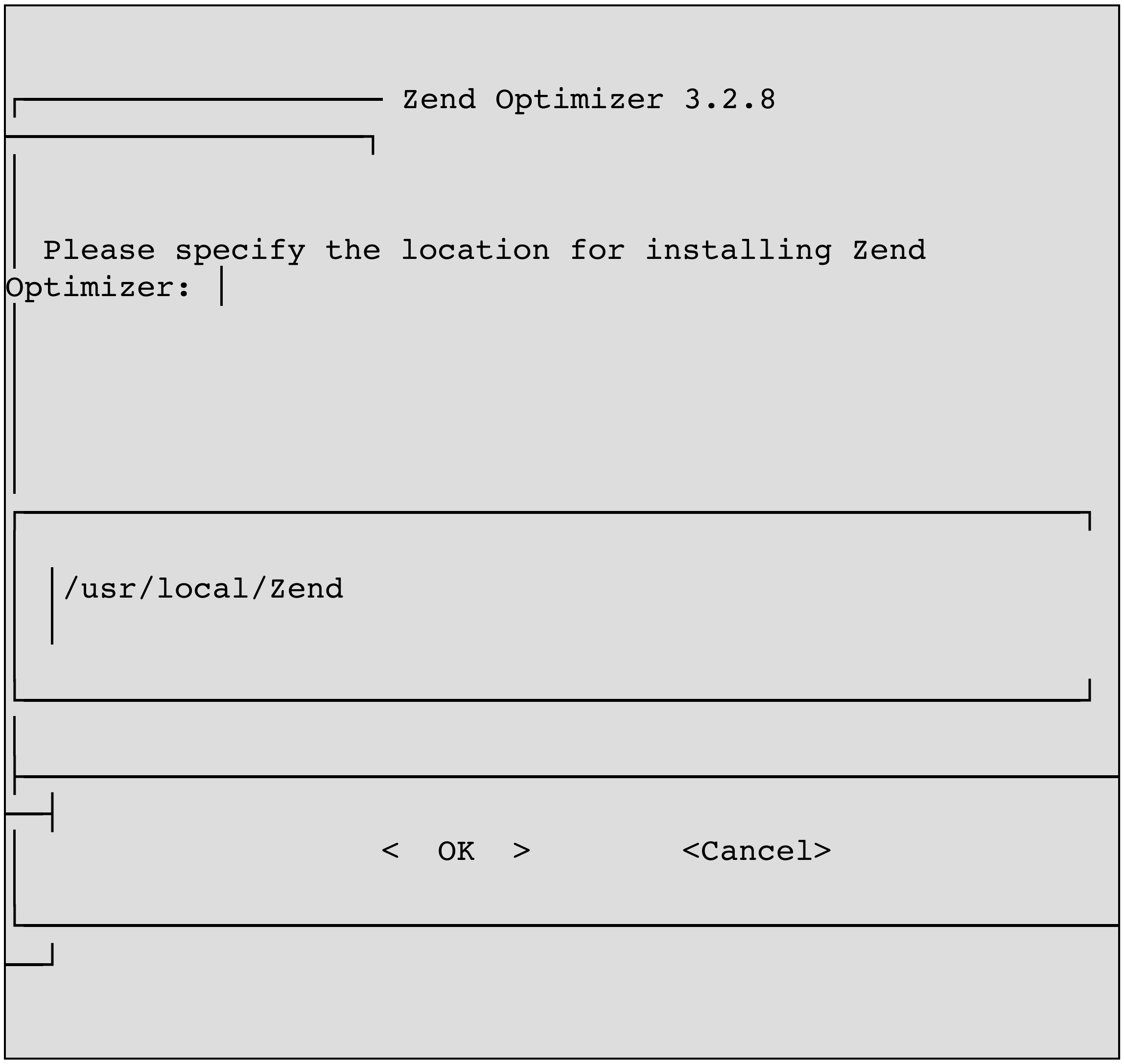
```


┌──────────────────────────── Zend Optimizer 3.2.8
──────────────────────────────┐
│
│
│ IMPORTANT:
│
│ BY SELECTING THE 'YES' OPTION BELOW, DOWNLOADING,
INSTALLING, OR                │
│ OTHERWISE USING THIS SOFTWARE, YOU ACKNOWLEDGE THAT YOU
HAVE READ THE           │
│ LICENSE AGREEMENT, AND THAT YOU AGREE TO BE BOUND BY ITS
TERMS AND              │
│ CONDITIONS.
│
│ IF YOU DO NOT AGREE TO ALL OF THE TERMS AND CONDITIONS OF
SUCH AGREEMENT,   │
│ YOU ARE NOT AN AUTHORIZED USER OF THE SOFTWARE AND IT IS
YOUR                   │
│ RESPONSIBILITY TO EXIT THIS DOWNLOADING/INSTALLATION
PROCESS WITHOUT         │
│ DOWNLOADING OR INSTALLING THE SOFTWARE BY SELECTING THE
'NO' OPTION BELOW, │
│ AND TO DELETE THE SOFTWARE FROM YOUR COMPUTER.
│
│
│
│
│
│ Do you accept the terms of this license?
│
```



```
< Yes >            < No  >
```

单击 < Yes > 按钮

4. Zend Optimizer 安装路径

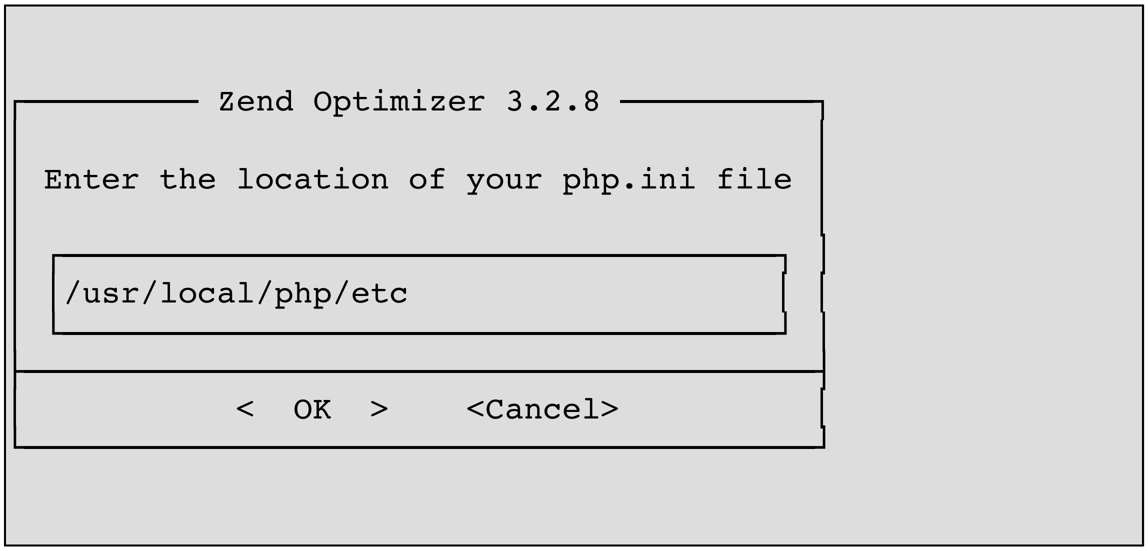


```
Zend Optimizer 3.2.8

 Please specify the location for installing Zend
Optimizer:

  /usr/local/Zend

        <  OK  >         <Cancel>
```

单击 < OK > 按钮

建议安装在/usr/local/Zend_3.2.8

5. php.ini 安装路径

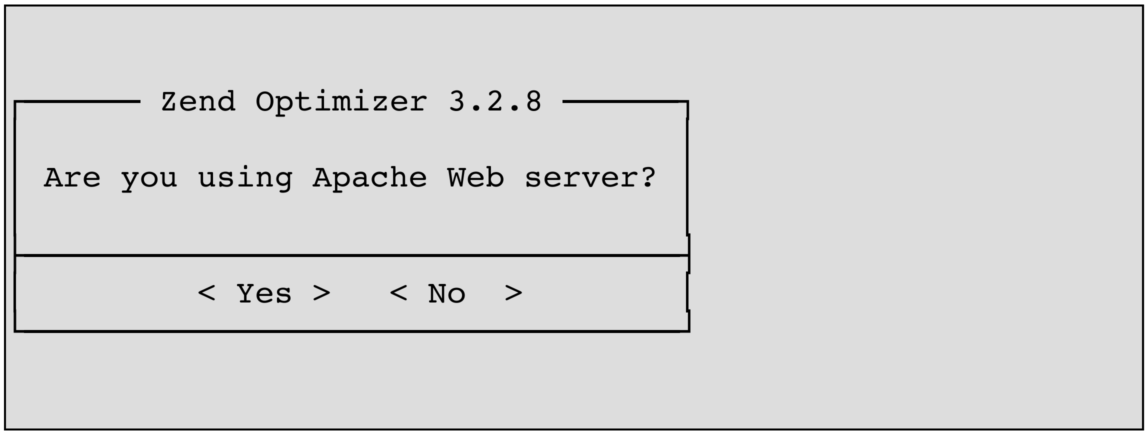


```
 Zend Optimizer 3.2.8
Enter the location of your php.ini file
/usr/local/php/etc
<  OK  >    <Cancel>
```

输入php.ini安装路径

单击 < OK > 按钮

6. 是否使用了Apache?



```
 Zend Optimizer 3.2.8
Are you using Apache Web server?
< Yes >   <  No  >
```

我的环境是 lighttpd 所以选择 No

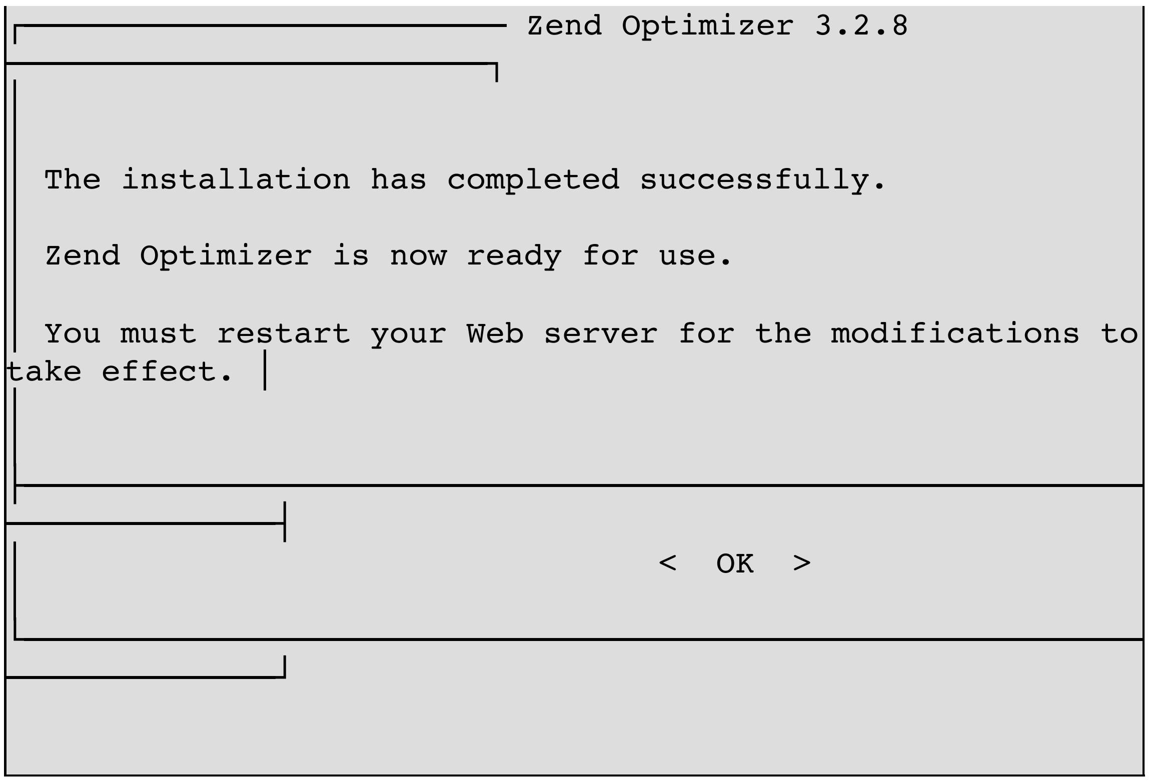
单击 < Yes > 按钮

7. 提示信息

```
┌──────────────────────────────────── Zend Optimizer 3.2.8
───────────────────────────────────┐
│
│
│ The following configuration changes have been made:
│
│
│
│ - The php.ini file has been relocated from
/usr/local/php/etc to /usr/local/Zend_3.2.8/etc │
│
│
│ - A symbolic link for the php.ini file has been created
in /usr/local/php/etc.                          │
│
│
│ - The original php.ini was backed up to
│
│  /usr/local/php/etc/php.ini-zend_optimizer.bak
│
│
│
│
├──────────────────────────────────────────────────────────
──────────────────────────────────┤
│                                         <  OK  >
│
└──────────────────────────────────────────────────────────
──────────────────────────────────┘
```

单击 < OK > 按钮

8. 安装完成



```
Zend Optimizer 3.2.8

The installation has completed successfully.

Zend Optimizer is now ready for use.

You must restart your Web server for the modifications to
take effect.

                    <  OK  >
```

单击＜OK＞按钮

6. eaccelerator

```
tar jxvf eaccelerator-0.9.5.3.tar.bz2
cd eaccelerator-0.9.5.3/
/opt/php/bin/phpize
./configure  --enable-eaccelerator=shared --with-php-
config=/opt/php/bin/php-config
make
make install
```

第 9 章 varnish - a state-of-the-art, high-performance HTTP accelerator

1. Varnish Install

http://varnish.projects.linpro.no/

1. install

```
$ sudo apt-get install varnish
```

2. /etc/default/varnish

```
$ sudo vim /etc/default/varnish
DAEMON_OPTS="-a :80 \
             -T localhost:6082 \
             -f /etc/varnish/default.vcl \
             -s
file,/var/lib/varnish/$INSTANCE/varnish_storage.bin,1G"
```

3. /etc/varnish/default.vcl

```
$ sudo vim /etc/varnish/default.vcl

backend default {
        .host = "127.0.0.1";
        .port = "8080";
}
```

4. reload

```
$ sudo /etc/init.d/varnish force-reload
 * Stopping HTTP accelerator                            [
OK ]
 * Starting HTTP accelerator
```

2. varnish utility

2.1. status

```
$ varnishstat
or
$ varnishstat -n /var/lib/varnish/atom-netkiller/
```

HTTP Head

```
$ curl -I http://bg7nyt.mooo.com/
HTTP/1.1 404 Not Found
X-Powered-By: PHP/5.2.6-3ubuntu4.2
Content-type: text/html
Server: lighttpd/1.4.19
Content-Length: 539
Date: Wed, 23 Sep 2009 00:05:11 GMT
X-Varnish: 938430316
Age: 0
Via: 1.1 varnish
Connection: keep-alive
```

test gzip,defalte

```
$ curl -H Accept-Encoding:gzip,defalte -I
http://bg7nyt.mooo.com/
HTTP/1.1 200 OK
X-Powered-By: PHP/5.2.6-3ubuntu4.2
Content-Encoding: gzip
Vary: Accept-Encoding
Content-type: text/html
Server: lighttpd/1.4.19
Date: Wed, 23 Sep 2009 00:08:51 GMT
X-Varnish: 938430335
Age: 0
Via: 1.1 varnish
```

```
Connection: keep-alive
```

2.2. varnishadm

help messages

```
$ varnishadm -T 127.0.0.1:6082 help
help [command]
ping [timestamp]
status
start
stop
stats
vcl.load <configname> <filename>
vcl.inline <configname> <quoted_VCLstring>
vcl.use <configname>
vcl.discard <configname>
vcl.list
vcl.show <configname>
param.show [-l] [<param>]
param.set <param> <value>
quit
purge.url <regexp>
purge.hash <regexp>
purge <field> <operator> <arg> [&& <field> <oper> <arg>]...
purge.list
```

清除缓存

通过Varnish管理端口，使用正则表达式批量清除缓存：

清除所有缓存

```
/usr/local/varnish/bin/varnishadm -T 127.0.0.1:6082 url.purge
```

```
*$
```

http://bg7nyt.mooo.com/zh-cn/technology/news.html 清除类/zh-cn/下所有缓存

```
/usr/local/varnish/bin/varnishadm -T 127.0.0.1:6082 url.purge
/zh-cn/
```

```
/usr/local/varnish/bin/varnishadm -T 127.0.0.1:3500 url.purge
w*$
```

2.3. varnishtop

```
varnishtop -i rxurl

varnishtop -i txurl

varnishtop -i RxHeader -I Accept-Encoding
```

2.4. varnishhist

2.5. varnishsizes

3. log file

log file

```
$ sudo vim  /etc/default/varnishlog
VARNISHLOG_ENABLED=1
$ sudo /etc/init.d/varnishlog start
 * Starting HTTP accelerator log deamon    [ OK ]

$ sudo vim  /etc/default/varnishncsa
VARNISHNCSA_ENABLED=1
$ sudo /etc/init.d/varnishncsa start
 * Starting HTTP accelerator log deamon    [ OK ]
```

4. Varnish Configuration Language - VCL

Varnish配置文件VCL中的函数详解

```

```

内置的例程

```
vcl_recv
有请求到达后成功接收并分析时被调用，一般以以下几个关键字结束。
error code [reason] 返回code给客户端，并放弃处理该请求
pass 进入pass模式，把控制权交给vcl_pass
pipe 进入pipe模式，把控制权交给vcl_pipe
lookup 在缓存里查找被请求的对象，根据查找结果把控制权交给vcl_hit或
vcl_miss

vcl_pipe
进入pipe模式时被调用。请求被直接发送到backend，后端和客户端之间的后继数据
不进行处理，只是简单传递，直到一方关闭连接。一般以以下几个关键字结束。
error code [reason]
pipe

vcl_pass
进入pass模式时被调用。请求被送到后端，后端应答数据送给客户端，但不进入缓
存。同一连接的后继请求正常处理。一般以以下几个关键字结束。
error code [reason]
pass

vcl_hash
目前不使用

vcl_hit
在lookup以后如果在cache中找到请求的内容事调用。一般以以下几个关键字结束。
error code [reason]
pass
deliver 将找到的内容发送给客户端，把控制权交给vcl_deliver.

vcl_miss
lookup后但没有找到缓存内容时调用，可以用于判断是否需要从后端服务器取内容。
```

```
一般以以下几个关键字结束。
error code [reason]
pass
fetch 从后端取得请求的内容，把控制权交给vcl_fetch.


vcl_fetch
从后端取得内容后调用。一般以以下几个关键字结束。
error code [reason]
pass
insert 将取到的内容插入缓存，然后发送给客户端，把控制权交给vcl_deliver


vcl_deliver
缓存内容发动给客户端前调用。一般以以下几个关键字结束。
error code [reason]
deliver 内容发送给客户端

vcl_timeout
在缓存内容到期前调用。一般以以下几个关键字结束。
fetch 从后端取得该内容
discard 丢弃该内容



vcl_discard
由于到期或者空间不足而丢弃缓存内容时调用。一般以以下几个关键字结束。
discard 丢弃
keep 继续保留在缓存里

如果这些内置例程没有被定义，则执行缺省动作


一些内置的变量
now 当前时间，标准时间点（1970？）到现在的秒数

backend.host 后端的IP或主机名
backend.port 后端的服务名或端口

请求到达后有效的变量
client.ip 客户端IP
server.ip 服务端IP
req.request 请求类型，比如GET或者HEAD或者POST
req.url 请求的URL
req.proto 请求的HTTP版本号
req.backend 请求对应的后端
req.http.header 对应的HTTP头

往后段的请求时有效的变量
bereq.request 比如GET或HEAD
bereq.url URL
bereq.proto 协议版本
bereq.http.header HTTP头

从cache或后端取到内容后有效的变量
obj.proto HTTP协议版本
obj.status HTTP状态代码
obj.response HTTP状态信息
obj.valid 是否有效的HTTP应答
obj.cacheable 是否可以缓存的内容，也就是说如果HTTP返回是200、203、
300、301、302、404、410并且有非0的生存期，则为可缓存
obj.ttl 生存期，秒
obj.lastuse 上一次请求到现在间隔秒数

对客户端应答时有效的变量
resp.proto response的HTTP版本
resp.status 回给客户端的HTTP状态代码
resp.response 回给客户端的HTTP状态信息
resp.http.header HTTP头
```

4.1. unset / set

```
 sub vcl_deliver {
##### Remove some headers
    unset resp.http.Server;
    unset resp.http.X-Powered-By;
    unset resp.http.X-Varnish;
    unset resp.http.Via;
###
        if (obj.hits > 0){
                set resp.http.X-Cache = "cdn cache server
v2.0";
        }else{
                set resp.http.X-Cache = "MISS ";
        }
        return (deliver);
}
```

5. example

例 9.1. default.vcl

```
neo@netkiller:/etc/varnish$ cat default.vcl
# This is a basic VCL configuration file for varnish.  See the
vcl(7)
# man page for details on VCL syntax and semantics.
#
# Default backend definition.  Set this to point to your
content
# server.
#
backend default {
    .host = "127.0.0.1";
    .port = "8080";
}
#
# Below is a commented-out copy of the default VCL logic.  If
you
# redefine any of these subroutines, the built-in logic will be
# appended to your code.
#
sub vcl_recv {
    if (req.http.x-forwarded-for) {
        set req.http.X-Forwarded-For =
            req.http.X-Forwarded-For ", " client.ip;
    } else {
        set req.http.X-Forwarded-For = client.ip;
    }
    if (req.request != "GET" &&
      req.request != "HEAD" &&
      req.request != "PUT" &&
      req.request != "POST" &&
      req.request != "TRACE" &&
      req.request != "OPTIONS" &&
      req.request != "DELETE") {
        /* Non-RFC2616 or CONNECT which is weird. */
        return (pipe);
    }
    if (req.request != "GET" && req.request != "HEAD") {
        /* We only deal with GET and HEAD by default */
        return (pass);
    }
    if (req.http.Authorization || req.http.Cookie) {
        /* Not cacheable by default */
/*        return (pass);*/
        return (lookup);
    }
    return (lookup);
}

sub vcl_pipe {
    # Note that only the first request to the backend will have
    # X-Forwarded-For set.  If you use X-Forwarded-For and want
to
    # have it set for all requests, make sure to have:
    # set req.http.connection = "close";
    # here.  It is not set by default as it might break some
broken web
    # applications, like IIS with NTLM authentication.
    return (pipe);
}

sub vcl_pass {
    return (pass);
}

sub vcl_hash {
    set req.hash += req.url;
    if (req.http.host) {
        set req.hash += req.http.host;
    } else {
        set req.hash += server.ip;
    }
    return (hash);
}

sub vcl_hit {
    if (!obj.cacheable) {
        return (pass);
    }
    return (deliver);
}
```

```
sub vcl_miss {
    return (fetch);
}

sub vcl_fetch {
    if (!beresp.cacheable) {
        return (pass);
    }
    if (beresp.http.Set-Cookie) {
#         return (pass);
        return (deliver);
    }
    return (deliver);
}

sub vcl_deliver {
    return (deliver);
}
#
# sub vcl_error {
#     set obj.http.Content-Type = "text/html; charset=utf-8";
#     synthetic {"
# <?xml version="1.0" encoding="utf-8"?>
# <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
#  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
# <html>
#   <head>
#     <title>"} obj.status " " obj.response {"</title>
#   </head>
#   <body>
#     <h1>Error "} obj.status " " obj.response {"</h1>
#     <p>"} obj.response {"</p>
#     <h3>Guru Meditation:</h3>
#     <p>XID: "} req.xid {"</p>
#     <hr>
#     <p>Varnish cache server</p>
#   </body>
# </html>
# "};
#     return (deliver);
# }
```

第 10 章 Apache Traffic Server

1. Install

```
yum install gcc gcc-c++ make autoconf -y
yum -y install  tcl lzma tcl-devel expat expat-devel pcre-devel
perl perl-devel
```

```
cd /usr/local/src/
wget
http://mirror.bjtu.edu.cn/apache//trafficserver/trafficserver-
3.0.1.tar.bz2
tar -xvjf trafficserver-3.0.1.tar.bz2
```

```
cd trafficserver-3.0.1
./configure --prefix=/srv/trafficserver-3.0.1 && make && make
install
```

2. Configure

```
修改配置
vi records.config
  CONFIG proxy.config.proxy_name STRING cache1
### 修改成cache的server name即可
  CONFIG proxy.config.cluster.ethernet_interface STRING eth0
### 修改成需要侦听的interface名称，默认是 null
  CONFIG proxy.config.admin.user_id STRING nobody
### 用来运行 traffic server 的用户,默认是nobody
  CONFIG proxy.config.http.server_port INT 80
### traffic server 侦听的端口，默认是8080

vi cache.config
dest_domain=www.example.com scheme=http     revalidate=2h

vi remap.conf
map http://www.example.com      http://10.0.0.51  #前一个是用户访
问的地址，后一个是源站点的IP，或者域名

配置变更应用生效
/srv/ts/bin/traffic_line -x

启动服务

/srv/ts/bin/trafficserver start


./traffic_shell
show
show:cache
show:cache-stats
show:proxy-stats

./logstats -i www.example.com

如果服务器down掉，默认会生成core文件，在/ts
使用

ts/bin/traffic_server -c core.1234
```

第 11 章 Cherokee

1. Installing Cherokee

```
apt-get install cherokee
```

Cherokee can be configured through a web-based control panel which we can start as follows:

```
cherokee-admin -b
```

cherokee script

```
/etc/init.d/cherokee restart
```

第 12 章 Jetty

第 13 章 Other Web Server

1. Python SimpleHTTPServer

```
python -m SimpleHTTPServer &
```

```
curl http://localhost:8000/
```

第 14 章 web 服务器排名

http://news.netcraft.com/

1. HTTP状态码

http://zh.wikipedia.org/wiki/HTTP%E7%8A%B6%E6%80%81%E7%A0%81

第 15 章 HTTP2

1. Chrome

检查你的浏览器是否支持 HTTP2 chrome://net-internals/#http2

```
HTTP/2 Enabled: true
```

表示正常