凡客诚品

# THE NEW ESB

Installation manual

程诚

**2012/11/2**

# 目录

# 1.安装 linux 开发工具

#yum groupinstall "Development Tools"
#yum -y install pcre* zlib* openssl*
（版本：openssl-1.0.1,pcre-8.31, zlib-1.2.7）

# 2.根目录下新建一个目录（根据自己的喜好选择路径）用于放置下载的程序

#mkdir/nginx

# 3.下载并解压 Nginx-1.2.4

#wget -c http://nginx.org/download/nginx-1.2.4.tar.gz
#tar zxvfnginx-1.2.4.tar.gz

# 4.下载并解压 substitutions4nginx（该模块用于 nginx 的 location 中的响应内容多处替换）

#svn checkout http://substitutions4nginx.googlecode.com/svn/trunk/ substitutions4nginx
#curdir=$(pwd)

# 5.新建用户

# groupadd www 创建 www 用户组
# useradd -g www -s /sbin/nologin -M www 创建 www 用户并将其添加到 www 用户组
# mkdir /www 创建/www 网站目录
# chmod +w /www 给/www 目录写权限
# chown -R www:www /www 将网站根目录/www 所有者和所属组设置为 www 用户和组

# 6.makenginx

#cd Nginx-1.2.4
#./configure --with-cc=c99 --builddir=objs.msvc8 --with-http_stub_status_module --with-http_sub_module --with-http_ssl_module --with-http_gzip_static_module --with-ipv6 --with-pcre --user=www --group=www --add-module=../substitutions4nginx
#make
#make install

# 7.limits 设置

#ulimit -n 655350 (同时修改/usr/local/nginx/conf/nginx.conf，添加 worker_rlimit_nofile 655350)
如下图：

```
#user  nobody;
worker_processes  32;
worker_rlimit_nofile 655350;

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

#pid        logs/nginx.pid;
```

#vi/etc/security/limits.conf

查找修改下面内容之后保存
* softnofile 655360
* hardnofile 655360
（*表示全局变量，soft 为软件，hard 为硬件，nofile 为文件打开数）
配置完毕之后输入 ulimit –n 则会显示 655360

# 8.nginx 的配置

## nginx.conf

```
#user    nobody;
worker_processes    32; #配置为处理核心的倍数，可以提高性能
worker_rlimit_nofile 655350;

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

#pid              logs/nginx.pid;


events {
      useepoll;
      worker_connections    10240;
}
```

```
http {
    #设置服务名称长度为 64
    server_names_hash_bucket_size 64;
    includemime.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;


    #加入 server 的配置
    includeservers.conf;

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

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

## servers.conf

```
server {
```

```
listen          80;
server_name    e.vancloa.cn;

        #charset koi8-r;
        access_log off;
access_log    html/logs.html access;

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


        #wcf
        location ^~/D4745CB4C374C834B4109E1F20E0D400{
            proxy_pass http://pointservice.vancldb.com/productpointservice.svc;
            subs_filter_types text/xml;
            subs_filter http://pointservice.vancldb.com/ http://$host/;
            subs_filter (\w+([-+.']\w+)*).svc D4745CB4C374C834B4109E1F20E0D400 r;
        }

        #webservice
        location ^~/16D81B3E8116C5C408401E2913015E27{
            proxy_pass http://app-cust-www.vancl.com/BasicInfoService.asmx;
            subs_filter_types text/xml;
            #subs_filter                                    \"/(\w+([-+.']\w+)*).asmx\?disco\"
\"/16D81B3E8116C5C408401E2913015E27?disco\" r;
            subs_filter http://app-cust-www.vancl.com/ http://$host/;
            subs_filter (\w+([-+.']\w+)*).asmx 16D81B3E8116C5C408401E2913015E27 r;
        }

        location ^~/zlntqhy{
            proxy_pass http://testforesb.com/Service1.svc;
            subs_filter_types text/xml;
            subs_filter http://testforesb.com/ http://$host/;
            subs_filter (\w+([-+.']\w+)*).svc zlntqhy r;
        }

    #状态配置
    includestatus.conf;
}
```

## status.conf

```
location /nginx_status {
        stub_status on;
        access_log off;
         #allow 127.0.0.1;#设置为可访问该状态信息的 ip
         #deny all;
}
```

# 9.启动 nginx

#/usr/local/nginx/sbin/nginx　　#启动
(#/usr/local/nginx/sbin/nginx　-s　stop #停止)

# 10.安装 PHP 和 spawn-fcgi

下载 php-5.4.8.tar.gz
#tar zxvfphp-5.4.8.tar.gz
#cd php-5.4.8
#./configure--prefix=/usr/local/php--enable-fastcgi --enable-debug　--enable-force-cgi-redirect
#make
#make install

下载 spawn-fcgi-1.6.0.tar.gz
#tar spawn-fcgi-1.60.tar.gz
#cd spawn-fcgi-1.6.0/
#./configure　--prefix=/usr/local/spawn-fcgi
#make && make install

然后仅仅是把 spawn-fcgi 这个文件 copy 去 php 的 bin 目录下，为什么仅仅只要一个文件呢？好吧，我也不知道。
#cp spawn-fcgi /usr/local/php/bin
#chmod +x /usr/local/php/bin/spawn-fcgi
#启动 phpcgi 进程
#/usr/local/php/bin/spawn-fcgi -a 127.0.0.1 -p 9000 -C 250 -u www -f /usr/local/php/bin/php-cgi
这样一来环境就搭建的差不多了，这中间容易出问题的就是 php 安装的时候可能会报缺某些工具包

# 11.安装 FCGI 模块

# wget http://search.cpan.org/CPAN/authors/id/B/BO/BOBTFISH/FCGI-0.67.tar.gz
# tarzxvf FCGI-0.67.tar.gz
# cd FCGI-0.67

```
# perl Makefile.PL
# make
# make install
```

## 12. 安装 IO 和 IO::ALL 模块

```
# wget http://search.cpan.org/CPAN/authors/id/G/GB/GBARR/IO-1.25.tar.gz
# tarzxvf IO-1.25.tar.gz
# cd IO-1.25
# perl Makefile.PL
# make
# make install
# wget http://search.cpan.org/CPAN/authors/id/I/IN/INGY/IO-All-0.39.tar.gz
# tarzxvf IO-All-0.39.tar.gz
# cd IO-All
# perl Makefile.PL
# make
# make install
```

## 13.下载 Perl 脚本

我把这个脚本放在 /usr/local/nginx/perl-fcgi.pl

```
# chmod 755 /usr/local/nginx/perl-fcgi.pl
```

## 14.cgi 启动/停止脚本 (nobody 为 nginx 的运行用户)

```
# vi /usr/local/nginx/start_perl_cgi.sh

#!/bin/bash
#set -x
dir=/usr/local/nginx

stop ()
{
#pkill    -f    $dir/perl-fcgi.pl
kill $(cat $dir/logs/perl-fcgi.pid)
rm $dir/logs/perl-fcgi.pid 2>/dev/null
rm $dir/logs/perl-fcgi.sock 2>/dev/null
echo "stop perl-fcgi done"
```

```
}

start ()
{
rm $dir/now_start_perl_fcgi.sh 2>/dev/null

chownnobody.root $dir/logs
echo "$dir/perl-fcgi.pl -l $dir/logs/perl-fcgi.log -pid $dir/logs/perl-fcgi.pid -S $dir/logs/perl-fcgi.sock" >>$dir/now_start_perl_fcgi.sh

chownnobody.nobody $dir/now_start_perl_fcgi.sh
chmodu+x $dir/now_start_perl_fcgi.sh

sudo -u nobody $dir/now_start_perl_fcgi.sh
echo "start perl-fcgi done"
}

case $1 in
stop)
stop
;;
start)
start
;;
restart)
stop
start
;;
esac
```

# chmod 755 /usr/local/nginx/start_perl_cgi.sh

启动脚本

# /usr/local/nginx/start_perl_cgi.sh start

正常情况下在/usr/local/nginx/logs 下生成 perl-fcgi.sock 这个文件,如果没有生成,那就要检查下上面的步聚了.

**#chmod 777 /usr/local/nginx/logs/nginx-fcgi.sock**

# 15.添加 www 用户

# groupadd www 创建 www 用户组
# useradd -g www -s /sbin/nologin -M www 创建 www 用户并将其添加到 www 用户组
# mkdir /www 创建/www 网站目录
# chmod +w /www 给/www 目录写权限
# chown -R www:www /www 将网站根目录/www 所有者和所属组设置为 www 用户和组

# 16.安装 nagios

# wget http://prdownloads.sourceforge.net/sourceforge/nagios/nagios-3.4.1.tar.gz
# tar zxvf nagios-3.4.1.tar.gz
# cd nagios-3.4.1
# useradd -m -s /bin/bash nagios
# groupaddnagios
# usermod -G nagiosnagios
# groupaddnagcmd
# usermod -a -G nagcmdnagios
# usermod -a -G nagcmd www
# ./configure --prefix=/data/nagios --with-command-group=nagcmd
# make
# make all
# make install
# make install-init # 生成 init 启动脚本
# make install-config # 安装示例配置文件
# make install-commandmode # 设置相应的目录权限

# 17. nagiox 插件安装

# wget http://prdownloads.sourceforge.net/sourceforge/nagiosplug/nagios-plugins-1.4.15.tar.gz
# tar zxvf nagios-plugins-1.4.15.tar.gz
# cd nagios-plugins-1.4.15
# ./configure --with-nagios-user=nagios --with-nagios-group=nagios --prefix=/data/nagios
# make
# make install

# 18. nrpe 安装 # tar zxvf nrpe-2.12.tar.gz

下载 nrpe-2.13.tar.gz
# cd nrpe-2.13
# ./configure

# make all
# cpsrc/check_nrpe /data/nagios/libexec/

## 19. nagios 配置

加入系统服务并设为开机自动
# chkconfig --add nagios
# chkconfignagios on # mkdir /data/nagios/var/rw
# chownnagios.nagios /data/nagios/var/rw # 测试配置文件可用
# /data/nagios/bin/nagios -v /data/nagios/etc/nagios.cfg # 取消用户认证(方便调试)
# vi /data/nagios/etc/cgi.cfg
找到 use_authentication=1 并把值改为 0 # 修改联系人邮箱
# vi /data/nagios/etc/objects/contacts.cfg # 定义 check_nrpe 命令
# vi /data/nagios/etc/objects/commands.cfg

```
define command{
command_namecheck_nrpe
command_line /data/nagios/libexec/check_nrpe -H $HOSTADDRESS$ -c $ARG1$
}
```

# 启动服务
# servicenagios start

## 20.nginx 配置

### Nagios.conf

```
server {
listen          80;
server_name    e.nagios.cn;

        #charset koi8-r;
        #access_log off;
access_log    html/logs.html access;

            location / {
                root     /data/nagios/share;
                index    index.html index.htm index.php;
            }
```

```
        location ~ .*\.(php|php5)?$
        {
        root /data/nagios/share;
        fastcgi_pass    127.0.0.1:9000;
        fastcgi_indexindex.php;
        includefcgi.conf;
        }

        location /nagios {
            alias /data/nagios/share;
        }

        location /cgi-bin/images {
            alias /data/nagios/share/images;
        }

        location /cgi-bin/stylesheets {
            alias /data/nagios/share/stylesheets;
        }

        location /pnp4nagios {
            root /data/pnp4nagios/share;
            index    index.html index.htm index.php;
        }

        location ~ .*\.(cgi|pl)?$
        {
        gzip off;
        root      /data/nagios/sbin;
        rewrite ^/nagios/cgi-bin/(.*)\.cgi /$1.cgi break;
        fastcgi_passunix:/usr/local/nginx/logs/perl-fcgi.sock;
        fastcgi_indexindex.cgi;
        includefcgi.conf;
        fastcgi_read_timeout      60;
        }
}
```

## fcgi.conf

```
fastcgi_param    GATEWAY_INTERFACE    CGI/1.1;
fastcgi_param    SERVER_SOFTWARE       nginx;
fastcgi_param    QUERY_STRING          $query_string;
```

```
fastcgi_param   REQUEST_METHOD      $request_method;
fastcgi_param   CONTENT_TYPE        $content_type;
fastcgi_param   CONTENT_LENGTH      $content_length;
fastcgi_param   SCRIPT_FILENAME     $document_root$fastcgi_script_name;
fastcgi_param   SCRIPT_NAME         $fastcgi_script_name;
fastcgi_param   REQUEST_URI         $request_uri;
fastcgi_param   DOCUMENT_URI        $document_uri;
fastcgi_param   DOCUMENT_ROOT       $document_root;
fastcgi_param   SERVER_PROTOCOL     $server_protocol;
fastcgi_param   REMOTE_ADDR         $remote_addr;
fastcgi_param   REMOTE_PORT         $remote_port;
fastcgi_param   SERVER_ADDR         $server_addr;
fastcgi_param   SERVER_PORT         $server_port;
fastcgi_param   SERVER_NAME         $server_name;
```

# 21.mongodb 驱动安装

下载 mongodb 驱动 mongodb-mongo-c-driver.tar.gz

```
#tar mongodb-mongo-c-driver.tar.gz
#cd mongodb-mongo-c-driver
#make
#make install
#make docs
#make STD=c89
```

编译测试

```
#gcc --std=c99 -static -I/usr/local/include -L/usr/local/lib -o example docs/examples/example.c
–lmongoc
#./example
```

# 22．将 mongodb 驱动以扩展功能插入到 nginx 中

```
#mkdir /nignx /nginx-1.2.4/auto/lib/mongo
#de    /nignx /nginx-1.2.4/auto/lib/mongo
#viconf
```

**输入下面的代码**

```
# Copyright (C) Igor Sysoev
# Copyright (C) Nginx, Inc.
```

```
ngx_feature="Mongo library"
ngx_feature_name=
ngx_feature_run=no
ngx_feature_incs=
ngx_feature_path=
ngx_feature_libs="-lmongoc"
ngx_feature_test=
        . auto/feature


if [ $ngx_found = no ]; then

        # FreeBSD port

ngx_feature="Mongo library in /usr/local/"

ngx_feature_libs="-I/usr/local/include -L/usr/local/lib -lmongoc"

        . auto/feature
fi


if [ $ngx_found = yes ]; then
        CORE_LIBS="$CORE_LIBS $ngx_feature_libs"

else

cat<< END

$0: error: the Mongo module requires the Mongo library.
You can either do not enable the module or install the library.

END

exit 1
fi
```

**添加编译选项**

#vi/nginx/nginx-1.2.4/auto/options

在 78 行添加

```
HTTP_MONGO=NO
```

在 210 行添加

--with-http_mongo_module)　　HTTP_MONGO=YES　　;;

#vi　/nginx/nginx-1.2.4/auto/lib/conf
在 77 行添加

```
if [ $HTTP_MONGO = YES ]; then
    . auto/lib/mongo/conf
Fi
```

这样就配置完成了
在编译是加上　--with-http_mongo_module
下面是完整的编译参数：
Configure --with-cc=c99 --builddir=objs.msvc8 --with-http_stub_status_module --with-http_sub_module --with-http_ssl_module --with-http_gzip_static_module --with-ipv6 --with-pcre --user=www --group=www --add-module=../substitutions4nginx --with-http_mongo_module

**使用 mongodb 驱动的时候一定要确保驱动已经安装成功。**

加入 mongodb 驱动成功编译之后
如果启动 nginx 出现 error while loading shared libraries: libmongoc.so.0.6: cannot open shared object file: No such file or directory 解决办法。
#vi/etc/ld.so.conf
在最后加入
/usr/local/lib

#/sbin/ldconfig（重启配置）
这样就可以成功启动 nginx 了。