主机信息：CentOS release 6.5 (Final)

Nginx版本：nginx-1.8.0

软件包：nginx-1.8.0.tar.gz

jemalloc-3.6.0.tar.gz

1. 安装步骤

安装编译环境

yum -y install gcc gcc+ gcc-c++ openssl openssl-devel pcre-devel zlib-devel

添加用户www

useadd www

2． 安装jemalloc

#cd jemalloc-3.6.0

#./configure

#make && make install

指定类库Lib路径

#echo '/usr/local/lib' > /etc/ld.so.conf.d/local.conf

#ldconfig

1. 编译安装nginx

#cd nginx-1.8.0

#./configure --prefix=/data/nginx --user=www --group=www --with-http\_stub\_status\_module --with-http\_ssl\_module --with-http\_gzip\_static\_module --with-ld-opt="-ljemalloc" --with-cc-opt='-O2'

#make&make install

1. 配置nginx

#vim /data/nginx/conf/nginx.conf

具体配置见svn 范例：

user www www;

worker\_processes 8;

error\_log logs/error.log error;

#error\_log logs/error.log notice;

#error\_log logs/error.log info;

pid logs/nginx.pid;

worker\_rlimit\_nofile 204800;

events {

use epoll;

worker\_connections 204800 ;

}

http {

include mime.types;

default\_type application/octet-stream;

charset utf-8;

log\_format main '$proxy\_add\_x\_forwarded\_for $remote\_user [$time\_local] "$request" '

'$status $body\_bytes\_sent "$http\_referer" '

'"$http\_user\_agent" "$http\_x\_forwarded\_for" '

'upstream: $upstream\_addr';

#access\_log logs/access.log main;

sendfile on;

tcp\_nopush on;

client\_header\_buffer\_size 32k;

large\_client\_header\_buffers 4 32k;

client\_max\_body\_size 8m;

client\_body\_buffer\_size 512k;

server\_names\_hash\_bucket\_size 128;

server\_tokens off;

gzip on;

gzip\_min\_length 1k;

gzip\_buffers 4 16k;

gzip\_http\_version 1.1;

gzip\_comp\_level 2;

gzip\_types text/plain application/x-javascript text/css application/xml text/javascript;

gzip\_vary on;

keepalive\_timeout 60;

upstream fdfs\_group2 {

server 10.10.99.90:8080 weight=5 max\_fails=2 fail\_timeout=30s;

server 10.10.102.157:8080 weight=5 max\_fails=2 fail\_timeout=30s;

}

upstream crowd {

server 10.10.87.70:8081 weight=5 max\_fails=2 fail\_timeout=30s;

server 10.10.86.71:8081 weight=5 max\_fails=2 fail\_timeout=30s;

server 10.10.95.4:8081 weight=5 max\_fails=2 fail\_timeout=30s;

server 10.10.90.134:8081 weight=5 max\_fails=2 fail\_timeout=30s;

}

upstream user {

server 10.10.87.70:8082 weight=5 max\_fails=2 fail\_timeout=30s;

server 10.10.86.71:8082 weight=5 max\_fails=2 fail\_timeout=30s;

server 10.10.95.4:8082 weight=5 max\_fails=2 fail\_timeout=30s;

server 10.10.90.134:8082 weight=5 max\_fails=2 fail\_timeout=30s;

}

upstream mobile {

server 10.10.87.70:8083 weight=5 max\_fails=2 fail\_timeout=30s;

server 10.10.86.71:8083 weight=5 max\_fails=2 fail\_timeout=30s;

server 10.10.95.4:8083 weight=5 max\_fails=2 fail\_timeout=30s;

server 10.10.90.134:8083 weight=5 max\_fails=2 fail\_timeout=30s;

}

upstream store {

server 10.10.87.70:8084 weight=5 max\_fails=2 fail\_timeout=30s;

server 10.10.90.134:8084 weight=5 max\_fails=2 fail\_timeout=30s;

}

upstream manager{

server 10.10.86.71:8084 weight=5 max\_fails=2 fail\_timeout=30s;

server 10.10.95.4:8084 weight=5 max\_fails=2 fail\_timeout=30s;

}

upstream order {

server 10.10.91.72:8084 weight=5 max\_fails=2 fail\_timeout=30s;

server 10.10.92.204:8084 weight=5 max\_fails=2 fail\_timeout=30s;

server 10.10.87.252:8084 weight=5 max\_fails=2 fail\_timeout=30s;

server 10.10.82.19:8084 weight=5 max\_fails=2 fail\_timeout=30s;

}

upstream web {

server 10.10.91.72:8083 weight=5 max\_fails=2 fail\_timeout=30s;

server 10.10.92.204:8083 weight=5 max\_fails=2 fail\_timeout=30s;

server 10.10.87.252:8083 weight=5 max\_fails=2 fail\_timeout=30s;

server 10.10.82.19:8083 weight=5 max\_fails=2 fail\_timeout=30s;

}

upstream partner {

server 10.10.97.212:8085 weight=5 max\_fails=2 fail\_timeout=30s;

server 10.10.106.99:8085 weight=5 max\_fails=2 fail\_timeout=30s;

}

server {

listen 80;

server\_name 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;

location ~\* ^/crowd-web/.\*$ {

proxy\_pass http://crowd;

include proxy.conf;

error\_log logs/crowd\_error.log error;

access\_log logs/crowd\_access.log main;

}

location ~\* ^/user-web/.\*$ {

proxy\_pass http://user;

include proxy.conf;

error\_log logs/user\_error.log error;

access\_log logs/user\_access.log main;

}

location ~ ^/$ {

rewrite\_log on;

rewrite ^/$ /crowd-web/index.html;

}

location ~\* ^/productstore-store/.\*$ {

proxy\_pass http://store;

include proxy.conf;

error\_log logs/store\_error.log error;

access\_log logs/store\_access.log main;

}

location ~\* ^/productstore-manager/.\*$ {

proxy\_pass http://manager;

include proxy.conf;

error\_log logs/manager\_error.log error;

access\_log logs/manager\_access.log main;

}

location ~\* ^/productstore-order/.\*$ {

proxy\_pass http://order;

include proxy.conf;

error\_log logs/order\_error.log error;

access\_log logs/order\_access.log main;

}

location ~\* ^/productstore-web/.\*$ {

proxy\_pass http://web;

include proxy.conf;

error\_log logs/web\_error.log error;

access\_log logs/web\_access.log main;

}

location ~\* ^/mobile-gateway-web/.\*$ {

proxy\_pass http://mobile;

include proxy.conf;

error\_log logs/web\_error.log error;

access\_log logs/web\_access.log main;

}

location ~\* ^/partner-web/.\*$ {

proxy\_pass http://partner;

include proxy.conf;

error\_log logs/web\_error.log error;

access\_log logs/web\_access.log main;

}

location /jjjr-static {

alias /usr/local/jjjr\_static\_resource;

#index index.html index.htm;

}

location /jjjr {

alias /usr/local/jjjr-manager-static;

#index index.html index.htm;

}

location /nginxstatus {

stub\_status on;

allow 10.10.109.47;

allow 10.10.24.197;

access\_log off;

deny all;

}

proxy\_intercept\_errors on;

error\_page 404 = @error404;

location @error404 {

rewrite ^(.\*)$ http://www.17money.com/jjjr-static/errors/error404.html;

}

error\_page 500 502 503 504 = @error50x;

location @error50x {

rewrite ^(.\*)$ http://www.17money.com/jjjr-static/errors/error500.html;

}

location /group2/M00 {

proxy\_pass http://fdfs\_group2;

include proxy.conf;

}

}

server {

listen 443 ssl;

proxy\_set\_header Host $host;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

ssl\_certificate\_key /root/17money.key;

ssl\_certificate /root/17money.crt;

ssl\_session\_cache shared:SSL:1m;

ssl\_session\_timeout 5m;

ssl on;

ssl\_protocols SSLv2 TLSv1 TLSv1.1 TLSv1.2;

ssl\_ciphers HIGH:!aNULL:!MD5;

location = /crowd-web/welcome {

# proxy\_pass http://crowd;

rewrite ^(.\*) http://$host$1 permanent;

}

location ~\* ^/crowd-web/.\*$ {

proxy\_pass http://crowd;

include proxy.conf;

}

location ~\* ^/user-web/.\*$ {

proxy\_pass http://user;

include proxy.conf;

}

location ~\* ^/productstore-order/.\*$ {

proxy\_pass http://order;

include proxy.conf;

}

}

}

1. 设置启动脚本

#vim /etc/init.d/nginx

# config: /etc/sysconfig/nginx

# pidfile: /data/nginx/logs/nginx.pid

# Source function library.

. /etc/rc.d/init.d/functions

# Source networking configuration.

. /etc/sysconfig/network

# Check that networking is up.

[ "$NETWORKING" = "no" ] && exit 0

nginx="/data/nginx/sbin/nginx"

prog=$(basename $nginx)

NGINX\_CONF\_FILE="/data/nginx/conf/nginx.conf"

[ -f /etc/sysconfig/nginx ] && . /etc/sysconfig/nginx

lockfile=/var/lock/subsys/nginx

make\_dirs() {

# make required directories

user=`nginx -V 2>&1 | grep "configure arguments:" | sed 's/[^\*]\*--user=\([^ ]\*\).\*/\1/g' -`

options=`$nginx -V 2>&1 | grep 'configure arguments:'`

for opt in $options; do

if [ `echo $opt | grep '.\*-temp-path'` ]; then

value=`echo $opt | cut -d "=" -f 2`

if [ ! -d "$value" ]; then

# echo "creating" $value

mkdir -p $value && chown -R $user $value

fi

fi

done

}

start() {

[ -x $nginx ] || exit 5

[ -f $NGINX\_CONF\_FILE ] || exit 6

make\_dirs

echo -n $"Starting $prog: "

daemon $nginx -c $NGINX\_CONF\_FILE

retval=$?

echo

[ $retval -eq 0 ] && touch $lockfile

return $retval

}

stop() {

echo -n $"Stopping $prog: "

killproc $prog -QUIT

retval=$?

echo

[ $retval -eq 0 ] && rm -f $lockfile

return $retval

}

restart() {

configtest || return $?

stop

sleep 1

start

}

reload() {

configtest || return $?

echo -n $"Reloading $prog: "

killproc $nginx -HUP

RETVAL=$?

echo

}

force\_reload() {

restart

}

configtest() {

$nginx -t -c $NGINX\_CONF\_FILE

}

rh\_status() {

status $prog

}

rh\_status\_q() {

rh\_status >/dev/null 2>&1

}

case "$1" in

start)

rh\_status\_q && exit 0

$1

;;

stop)

rh\_status\_q || exit 0

$1

;;

restart|configtest)

$1

;;

reload)

rh\_status\_q || exit 7

$1

;;

force-reload)

force\_reload

;;

status)

rh\_status

;;

condrestart|try-restart)

rh\_status\_q || exit 0

;;

\*)

echo $"Usage: $0 {start|stop|status|restart|condrestart|try-restart|reload|force-reload|configtest}"

exit 2

esac

1. 启动nginx

#Service nginx start