Nginx调度器及健康检查

#环境：centos7.4虚拟机3台，关闭防火墙、SELinux、清空iptables规则，搭建好yum

#IP地址： proxy：192.168.1.1

Backend1:192.168.1.2

Backend2:192.168.1.3

Client：192.168.1.254（真机）

#配置后台

[root@backend1 ~]# yum -y install httpd mariadb-server

[root@backend1 ~]# echo backend1 >> /var/www/html/index.html

[root@backend1 ~]# systemctl start httpd mariadb

[root@backend1 ~]# mysqladmin -uroot password 123456

[root@backend2 ~]# yum -y install httpd mariadb-server

[root@backend2 ~]# echo backend2 >> /var/www/html/index.html

[root@backend2 ~]# systemctl start httpd mariadb

[root@backend2 ~]# mysqladmin -uroot password 123456

#配置代理服务器

[root@proxy ~]# ls

nginx-1.12.2.tar.gz ngx\_healthcheck\_module.tar.gz

[root@proxy ~]# tar -xf nginx-1.12.2.tar.gz

[root@proxy ~]# yum -y install gcc make pcre-devel openssl-devel

[root@proxy ~]# useradd -s /sbin/nologin nginx

[root@proxy ~]# cd nginx-1.12.2/

[root@proxy nginx-1.12.2]# ./configure --prefix=/usr/local/nginx --user=nginx --group=nginx --with-http\_ssl\_module --with-stream --with-http\_stub\_status\_module --without-http\_autoindex\_module --without-http\_ssi\_module

[root@proxy nginx-1.12.2]# make && make install

#配置nginx

[root@proxy nginx-1.12.2]# cd /usr/local/nginx

[root@proxy nginx]# vim conf/nginx.conf

[root@proxy nginx]# cat conf/nginx.conf | grep -v ^$ | grep -v "#"

user nginx;

worker\_processes auto;

pid logs/nginx.pid;

events {

worker\_connections 65535;

}

stream {

upstream backssh {

server 192.168.1.2:22;

server 192.168.1.3:22;

}

upstream backmysql {

server 192.168.1.2:3306;

server 192.168.1.3:3306;

}

server {

listen 22222;

proxy\_pass backssh;

}

server {

listen 33306;

proxy\_pass backmysql;

}

}

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"';

sendfile on;

keepalive\_timeout 65;

gzip on;

upstream webs {

server 192.168.1.2:80 weight=1 max\_fails=2 fail\_timeout=10;

server 192.168.1.3:80 weight=2 max\_fails=2 fail\_timeout=10;

}

server {

listen 80;

server\_name localhost;

charset utf-8;

access\_log logs/host.access.log main;

location / {

proxy\_pass http://webs/;

}

error\_page 404 /404.html;

error\_page 500 502 503 504 /50x.html;

location = /50x.html {

root html;

}

}

}

[root@proxy nginx]# echo "页面丢失了" >> html/404.html

[root@proxy nginx]# ./sbin/nginx -t

nginx: the configuration file /usr/local/nginx/conf/nginx.conf syntax is ok

nginx: configuration file /usr/local/nginx/conf/nginx.conf test is successful

[root@proxy nginx]# ./sbin/nginx

[root@proxy nginx]# netstat -antpu | grep nginx

tcp 0 0 0.0.0.0:22222 0.0.0.0:\* LISTEN 4211/nginx: master

tcp 0 0 0.0.0.0:80 0.0.0.0:\* LISTEN 4211/nginx: master

tcp 0 0 0.0.0.0:33306 0.0.0.0:\* LISTEN 4211/nginx: master

[root@proxy nginx]#

#测试代理

#测试7层web代理

[root@hostos ~]# curl http://192.168.1.1

backend2

[root@hostos ~]# curl http://192.168.1.1

backend1

#测试4层ssh代理

[root@hostos ~]# ssh -p 22222 192.168.1.1 "hostname" 2> /dev/null

backend2.tedu.cn

[root@hostos ~]# ssh -p 22222 192.168.1.1 "hostname" 2> /dev/null

backend1.tedu.cn

#测试4层mysql代理

[root@backend1 ~]# mysql -hlocalhost -uroot -p123456 << EOF

> grant all on \*.\* to root@'%' identified by '123456' with grant option;

> delete from mysql.user where password=’’;

> flush privileges;

> EOF

[root@backend2 ~]# mysql -hlocalhost -uroot -p123456 << EOF

> grant all on \*.\* to root@'%' identified by '123456' with grant option;

> delete from mysql.user where password=’’;

> flush privileges;

> EOF

[root@hostos ~]# mysql -h192.168.1.1 -uroot -p123456 -P33306 -e "select @@hostname" 2> /dev/null

+------------------+

| @@hostname |

+------------------+

| backend1.tedu.cn |

+------------------+

[root@hostos ~]# mysql -h192.168.1.1 -uroot -p123456 -P33306 -e "select @@hostname" 2> /dev/null

+------------------+

| @@hostname |

+------------------+

| backend2.tedu.cn |

+------------------+

#测试状态监测

[root@proxy ~]#

[root@proxy ~]# cd /usr/local/nginx/

[root@proxy nginx]# vim conf/nginx.conf

[root@proxy nginx]# sed -rn "68,74p" conf/nginx.conf

location / {

proxy\_pass http://webs/;

}

location /webstat {

stub\_status on;

}

[root@proxy nginx]# ./sbin/nginx -s reload

[root@proxy nginx]# curl 192.168.1.1/webstat/

Active connections: 1

server accepts handled requests

3 3 3

Reading: 0 Writing: 1 Waiting: 0

[root@proxy nginx]#

#7层健康检查

[root@proxy ~]# wget <https://github.com/yaoweibin/nginx_upstream_check_module/archive/master.zip>

[root@proxy ~]# unzip master.zip

[root@proxy ~]# mv nginx\_upstream\_check\_module-master/ nginx\_upstream\_check\_module

[root@proxy ~]# cd /root/nginx-1.12.2/

#根据nginx版本号进行打补丁

[root@proxy nginx-1.12.2]# patch -p1 < /root/nginx\_upstream\_check\_module/check\_1.12.1+.patch

patching file src/http/modules/ngx\_http\_upstream\_hash\_module.c

patching file src/http/modules/ngx\_http\_upstream\_ip\_hash\_module.c

patching file src/http/modules/ngx\_http\_upstream\_least\_conn\_module.c

patching file src/http/ngx\_http\_upstream\_round\_robin.c

patching file src/http/ngx\_http\_upstream\_round\_robin.h

[root@proxy nginx-1.12.2]# ./configure --prefix=/usr/local/nginx --user=nginx --group=nginx --with-http\_ssl\_module --with-stream --with-http\_stub\_status\_module --without-http\_autoindex\_module --without-http\_ssi\_module --add-module=/root/nginx\_upstream\_check\_module

[root@proxy nignx-1.12.2]# make

#平滑升级nginx

[root@proxy nginx-1.12.2]# /usr/local/nginx/sbin/nginx -V

nginx version: nginx/1.12.2

built by gcc 4.8.5 20150623 (Red Hat 4.8.5-16) (GCC)

built with OpenSSL 1.0.2k-fips 26 Jan 2017

TLS SNI support enabled

configure arguments: --prefix=/usr/local/nginx --user=nginx --group=nginx --with-http\_ssl\_module --with-stream --with-http\_stub\_status\_module --without-http\_autoindex\_module --without-http\_ssi\_module

[root@proxy nginx-1.12.2]# mv /usr/local/nginx/sbin/nginx /usr/local/nginx/sbin/nginx.old

[root@proxy nginx-1.12.2]# cp objs/nginx /usr/local/nginx/sbin/

[root@proxy nginx-1.12.2]# make upgrade

/usr/local/nginx/sbin/nginx -t

nginx: the configuration file /usr/local/nginx/conf/nginx.conf syntax is ok

nginx: configuration file /usr/local/nginx/conf/nginx.conf test is successful

kill -USR2 `cat /usr/local/nginx/logs/nginx.pid`

sleep 1

test -f /usr/local/nginx/logs/nginx.pid.oldbin

kill -QUIT `cat /usr/local/nginx/logs/nginx.pid.oldbin`

[root@proxy nginx-1.12.2]# /usr/local/nginx/sbin/nginx -V

nginx version: nginx/1.12.2

built by gcc 4.8.5 20150623 (Red Hat 4.8.5-16) (GCC)

built with OpenSSL 1.0.2k-fips 26 Jan 2017

TLS SNI support enabled

configure arguments: --prefix=/usr/local/nginx --user=nginx --group=nginx --with-http\_ssl\_module --with-stream --with-http\_stub\_status\_module --without-http\_autoindex\_module --without-http\_ssi\_module --add-module=/root/nginx\_upstream\_check\_module

[root@proxy nginx-1.12.2]# cd /usr/local/nginx/

[root@proxy nginx]# vim conf/nginx.conf

[root@proxy nginx]# sed -rn "56,62p;79,82p" conf/nginx.conf

upstream webs {

server 192.168.1.2:80 weight=1 max\_fails=2 fail\_timeout=10;

server 192.168.1.3:80 weight=2 max\_fails=2 fail\_timeout=10;

check interval=3000 rise=2 fall=3 timeout=2000;

check\_http\_send "HEAD / HTTP/1.1\r\nConnection: keep-alive\r\n\r\n";

check\_http\_expect\_alive http\_2xx http\_3xx;

}

location /heastat {

check\_status;

access\_log off;

}

[root@proxy nginx]# ./sbin/nginx -t

nginx: the configuration file /usr/local/nginx/conf/nginx.conf syntax is ok

nginx: configuration file /usr/local/nginx/conf/nginx.conf test is successful

[root@proxy nginx]# ./sbin/nginx -s reload

#测试结果

[root@proxy nginx]# curl 192.168.1.1/heastat?format=json

{"servers": {

"total": 2,

"generation": 1,

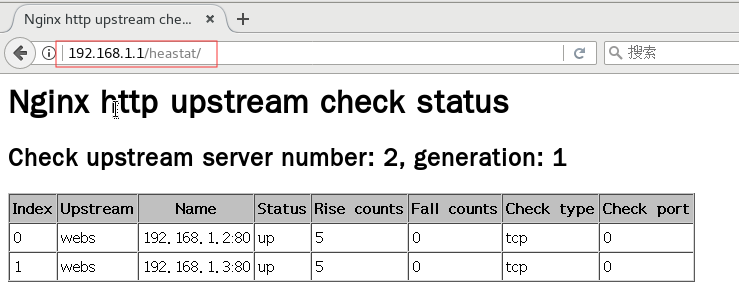
"server": [

{"index": 0, "upstream": "webs", "name": "192.168.1.2:80", "status": "up", "rise": 3, "fall": 0, "type": "tcp", "port": 0},

{"index": 1, "upstream": "webs", "name": "192.168.1.3:80", "status": "up", "rise": 3, "fall": 0, "type": "tcp", "port": 0}

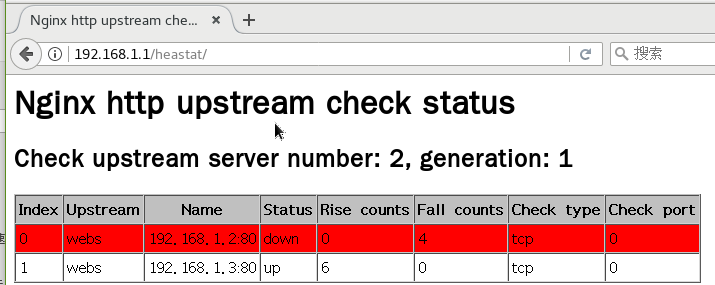
]

}}



#宕机测试

[root@backend1 ~]# systemctl stop httpd



#四层健康检查

[root@proxy ~]# git clone <https://github.com/zhouchangxun/ngx_healthcheck_module.git>

[root@proxy ~]# rm -rf nginx-1.12.2

[root@proxy ~]# tar -xf nginx-1.12.2.tar.gz

[root@proxy ~]# cd nginx-1.12.2/

[root@proxy nginx-1.12.2]# patch -p1 < /root/ngx\_healthcheck\_module/nginx\_healthcheck\_for\_nginx\_1.12+.patch

patching file src/http/modules/ngx\_http\_upstream\_hash\_module.c

patching file src/http/modules/ngx\_http\_upstream\_ip\_hash\_module.c

patching file src/http/modules/ngx\_http\_upstream\_least\_conn\_module.c

patching file src/http/ngx\_http\_upstream\_round\_robin.c

patching file src/http/ngx\_http\_upstream\_round\_robin.h

patching file src/stream/ngx\_stream\_upstream\_hash\_module.c

patching file src/stream/ngx\_stream\_upstream\_least\_conn\_module.c

patching file src/stream/ngx\_stream\_upstream\_round\_robin.c

patching file src/stream/ngx\_stream\_upstream\_round\_robin.h

[root@proxy nginx-1.12.2]# ./configure --prefix=/usr/local/nginx --user=nginx --group=nginx --with-http\_ssl\_module --with-stream --with-http\_stub\_status\_module --without-http\_autoindex\_module --without-http\_ssi\_module --add-module=/root/ngx\_healthcheck\_module

[root@proxy nginx-1.12.2]# make

[root@proxy nginx-1.12.2]# /usr/local/nginx/sbin/nginx -V

nginx version: nginx/1.12.2

built by gcc 4.8.5 20150623 (Red Hat 4.8.5-16) (GCC)

built with OpenSSL 1.0.2k-fips 26 Jan 2017

TLS SNI support enabled

configure arguments: --prefix=/usr/local/nginx --user=nginx --group=nginx --with-http\_ssl\_module --with-stream --with-http\_stub\_status\_module --without-http\_autoindex\_module --without-http\_ssi\_module --add-module=/root/nginx\_upstream\_check\_module

[root@proxy nginx-1.12.2]# mv /usr/local/nginx/sbin/nginx /usr/local/nginx/sbin/nginx.old.1

[root@proxy nginx-1.12.2]# cp objs/nginx /usr/local/nginx/sbin/

[root@proxy nginx-1.12.2]# make upgrade

/usr/local/nginx/sbin/nginx -t

nginx: the configuration file /usr/local/nginx/conf/nginx.conf syntax is ok

nginx: configuration file /usr/local/nginx/conf/nginx.conf test is successful

kill -USR2 `cat /usr/local/nginx/logs/nginx.pid`

sleep 1

test -f /usr/local/nginx/logs/nginx.pid.oldbin

kill -QUIT `cat /usr/local/nginx/logs/nginx.pid.oldbin`

[root@proxy nginx-1.12.2]# /usr/local/nginx/sbin/nginx -V

nginx version: nginx/1.12.2

built by gcc 4.8.5 20150623 (Red Hat 4.8.5-16) (GCC)

built with OpenSSL 1.0.2k-fips 26 Jan 2017

TLS SNI support enabled

configure arguments: --prefix=/usr/local/nginx --user=nginx --group=nginx --with-http\_ssl\_module --with-stream --with-http\_stub\_status\_module --without-http\_autoindex\_module --without-http\_ssi\_module --add-module=/root/ngx\_healthcheck\_module

[root@proxy nginx-1.12.2]#

[root@proxy nginx-1.12.2]# cd /usr/local/nginx/

[root@proxy nginx]# vim conf/nginx.conf

[root@proxy nginx]# sed -rn "16,26p;58,72p" conf/nginx.conf

upstream backssh {

server 192.168.1.2:22;

server 192.168.1.3:22;

check interval=3000 rise=2 fall=3 timeout=2000 default\_down=true type=tcp;

}

upstream backmysql {

server 192.168.1.2:3306;

server 192.168.1.3:3306;

check interval=3000 rise=2 fall=3 timeout=2000 default\_down=true type=tcp;

}

upstream webs {

server 192.168.1.2:80 weight=1 max\_fails=2 fail\_timeout=10;

server 192.168.1.3:80 weight=2 max\_fails=2 fail\_timeout=10;

check interval=3000 rise=2 fall=3 timeout=2000 type=http;

check\_http\_send "GET / HTTP/1.0\r\n\r\n";

check\_http\_expect\_alive http\_2xx http\_3xx;

}

server {

listen 8888;

location /allstat {

healthcheck\_status;

access\_log off;

}

}

[root@proxy nginx]# ./sbin/nginx -t

nginx: the configuration file /usr/local/nginx/conf/nginx.conf syntax is ok

nginx: configuration file /usr/local/nginx/conf/nginx.conf test is successful

[root@proxy nginx]# ./sbin/nginx -s reload

[root@proxy nginx]# netstat -anptu | grep nginx

tcp 0 0 0.0.0.0:22222 0.0.0.0:\* LISTEN 8824/nginx: master

tcp 0 0 0.0.0.0:80 0.0.0.0:\* LISTEN 8824/nginx: master

tcp 0 0 0.0.0.0:8888 0.0.0.0:\* LISTEN 8824/nginx: master

tcp 0 0 0.0.0.0:33306 0.0.0.0:\* LISTEN 8824/nginx: master

#测试检查

[root@proxy nginx]# curl 192.168.1.1:8888/allstat?format=json

{"servers": {

"total": 6,

"generation": 4,

"http": [

{"index": 0, "upstream": "webs", "name": "192.168.1.2:80", "status": "up", "rise": 18, "fall": 0, "type": "http", "port": 0},

{"index": 1, "upstream": "webs", "name": "192.168.1.3:80", "status": "up", "rise": 20, "fall": 0, "type": "http", "port": 0}

],

"stream": [

{"index": 0, "upstream": "backssh", "name": "192.168.1.2:22", "status": "up", "rise": 104, "fall": 0, "type": "tcp", "port": 0},

{"index": 1, "upstream": "backssh", "name": "192.168.1.3:22", "status": "up", "rise": 110, "fall": 0, "type": "tcp", "port": 0},

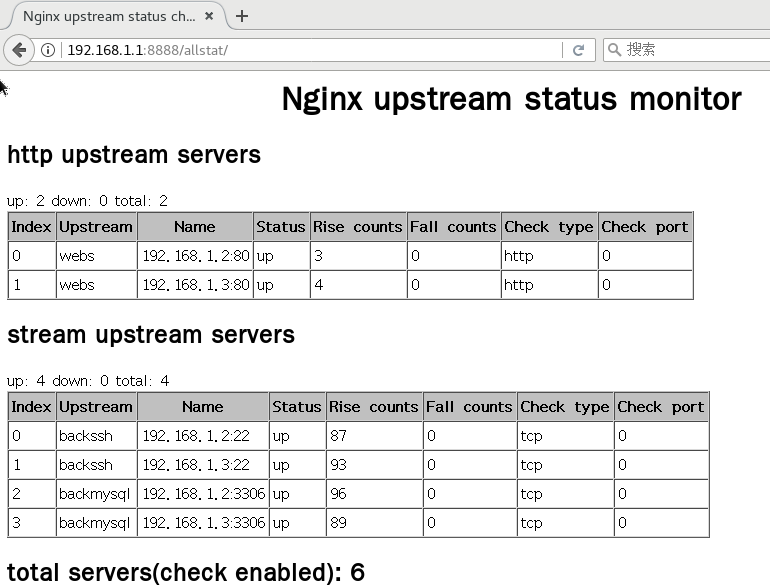
{"index": 2, "upstream": "backmysql", "name": "192.168.1.2:3306", "status": "up", "rise": 112, "fall": 0, "type": "tcp", "port": 0},

{"index": 3, "upstream": "backmysql", "name": "192.168.1.3:3306", "status": "up", "rise": 106, "fall": 0, "type": "tcp", "port": 0}

]

}}

[root@proxy nginx]#



#宕机测试

[root@backend1 ~]# systemctl stop httpd mariadb

[root@proxy nginx]# curl 192.168.1.1:8888/allstat?format=json

{"servers": {

"total": 6,

"generation": 4,

"http": [

{"index": 0, "upstream": "webs", "name": "192.168.1.2:80", "status": "down", "rise": 0, "fall": 4, "type": "http", "port": 0},

{"index": 1, "upstream": "webs", "name": "192.168.1.3:80", "status": "up", "rise": 45, "fall": 0, "type": "http", "port": 0}

],

"stream": [

{"index": 0, "upstream": "backssh", "name": "192.168.1.2:22", "status": "up", "rise": 129, "fall": 0, "type": "tcp", "port": 0},

{"index": 1, "upstream": "backssh", "name": "192.168.1.3:22", "status": "up", "rise": 137, "fall": 0, "type": "tcp", "port": 0},

{"index": 2, "upstream": "backmysql", "name": "192.168.1.2:3306", "status": "down", "rise": 0, "fall": 3, "type": "tcp", "port": 0},

{"index": 3, "upstream": "backmysql", "name": "192.168.1.3:3306", "status": "up", "rise": 132, "fall": 0, "type": "tcp", "port": 0}

]

}}

