[Top](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/PROJECT2/DAY02/CASE/01/index.html" \l "page_top_case)

# NSD PROJECT2 DAY02

1. [案例1：升级网站运行平台](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/PROJECT2/DAY02/CASE/01/index.html" \l "case1)
2. [案例2：部署缓存服务](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/PROJECT2/DAY02/CASE/01/index.html" \l "case2)
3. [案例3：数据迁移](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/PROJECT2/DAY02/CASE/01/index.html" \l "case3)
4. [案例4：部署LB集群](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/PROJECT2/DAY02/CASE/01/index.html" \l "case4)
5. [案例5：部署HA集群](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/PROJECT2/DAY02/CASE/01/index.html" \l "case5)

## 1 案例1：升级网站运行平台

### 1.1 问题

具体配置如下：

1. 清除当前配置
2. 部署LNMP
3. 测试配置

### 1.2 步骤

实现此案例需要按照如下步骤进行。

步骤一：清除当前配置(web33和web44服务器都要配置)

1）停止网站服务

1. [root@web33 ~]# systemctl stop httpd
2. [root@web33 ~]# systemctl disable httpd
3. [root@web44 ~]# systemctl stop httpd
4. [root@web44 ~]# systemctl disable httpd

2）卸载共享存储

1. [root@web33 ~]# umount /var/www/html //卸载当前挂载
2. [root@web33 ~]# vim /etc/fstab //清除开机挂载
3. #192.168.4.30/sitedir /var/www/html nfs defaults 0 0
4. :wq
5. [root@web44 ~]# umount /var/www/html //卸载当前挂载
6. [root@web44 ~]# vim /etc/fstab //清除开机挂载
7. #192.168.4.30/sitedir /var/www/html nfs defaults 0 0
8. :wq

步骤二：部署LNMP

1）安装软件

1. [root@web33 ~]# yum -y install gcc zlib-devel pcre-devel //安装源码Nginx依赖软件
2. 已安装:
3. gcc.x86\_64 0:4.8.5-28.el7 pcre-devel.x86\_64 0:8.32-17.el7 zlib-devel.x86\_64 0:1.2.7-17.el7
4. 作为依赖被安装:
5. cpp.x86\_64 0:4.8.5-28.el7 glibc-devel.x86\_64 0:2.17-222.el7 glibc-headers.x86\_64 0:2.17-222.el7 kernel-headers.x86\_64 0:3.10.0-862.el7 libmpc.x86\_64 0:1.0.1-3.el7
6. mpfr.x86\_64 0:3.1.1-4.el7
7. 完毕！
8. [root@web33 ~]#
9. [root@web33 ~]# tar -zxvf nginx-1.12.2.tar.gz //解压
10. [root@web33 ~]# cd nginx-1.12.2 //进源码目录
11. [root@web33 nginx-1.12.2]# ./configure //配置
12. ......
13. Configuration summary
14. + using system PCRE library
15. + OpenSSL library is not used
16. + using system zlib library
17. nginx path prefix: "/usr/local/nginx"
18. nginx binary file: "/usr/local/nginx/sbin/nginx"
19. nginx modules path: "/usr/local/nginx/modules"
20. nginx configuration prefix: "/usr/local/nginx/conf"
21. nginx configuration file: "/usr/local/nginx/conf/nginx.conf"
22. nginx pid file: "/usr/local/nginx/logs/nginx.pid"
23. nginx error log file: "/usr/local/nginx/logs/error.log"
24. nginx http access log file: "/usr/local/nginx/logs/access.log"
25. nginx http client request body temporary files: "client\_body\_temp"
26. nginx http proxy temporary files: "proxy\_temp"
27. nginx http fastcgi temporary files: "fastcgi\_temp"
28. nginx http uwsgi temporary files: "uwsgi\_temp"
29. nginx http scgi temporary files: "scgi\_temp"
30. [root@web33 nginx-1.12.2]# make //编译
31. ……
32. ……
33. sed -e "s|%%PREFIX%%|/usr/local/nginx|" \
34. -e "s|%%PID\_PATH%%|/usr/local/nginx/logs/nginx.pid|" \
35. -e "s|%%CONF\_PATH%%|/usr/local/nginx/conf/nginx.conf|" \
36. -e "s|%%ERROR\_LOG\_PATH%%|/usr/local/nginx/logs/error.log|" \
37. < man/nginx.8 > objs/nginx.8
38. make[1]: 离开目录“/root/nginx-1.12.2”
39. [root@web33 nginx-1.12.2]#
40. [root@web33 nginx-1.12.2]# make install //安装
41. ……
42. ……
43. cp conf/nginx.conf '/usr/local/nginx/conf/nginx.conf.default'
44. test -d '/usr/local/nginx/logs' \
45. || mkdir -p '/usr/local/nginx/logs'
46. test -d '/usr/local/nginx/logs' \
47. || mkdir -p '/usr/local/nginx/logs'
48. test -d '/usr/local/nginx/html' \
49. || cp -R html '/usr/local/nginx'
50. test -d '/usr/local/nginx/logs' \
51. || mkdir -p '/usr/local/nginx/logs'
52. make[1]: 离开目录“/root/nginx-1.12.2”
53. [root@web33 nginx-1.12.2]#
54. [root@web33 nginx-1.12.2]# ls /usr/local/nginx //查看安装目录
55. conf html logs sbin
56. [root@web33 nginx-1.12.2]#
57. [root@web33 ~]# yum -y install php-fpm //安装php-fpm 软件
58. ……
59. ……
60. 已安装:
61. php-fpm.x86\_64 0:5.4.16-45.el7
62. 作为依赖被安装:
63. libzip.x86\_64 0:0.10.1-8.el7 php-common.x86\_64 0:5.4.16-45.el7
64. 完毕！
65. [root@web33 ~]# yum -y install php php-mysql //安装php 及 php-mysql 软件
66. ……
67. ……
68. 已安装:
69. php.x86\_64 0:5.4.16-45.el7 php-mysql.x86\_64 0:5.4.16-45.el7
70. 作为依赖被安装:
71. mariadb-libs.x86\_64 1:5.5.56-2.el7 php-cli.x86\_64 0:5.4.16-45.el7 php-pdo.x86\_64 0:5.4.16-45.el7
72. 完毕！
73. [root@web33 ~]#
74. [root@web33 ~]# yum -y install mariadb-server mariadb-devel mariadb //安装mariadb服务软件
75. ……
76. ……
77. 已安装:
78. mariadb.x86\_64 1:5.5.56-2.el7 mariadb-devel.x86\_64 1:5.5.56-2.el7 mariadb-server.x86\_64 1:5.5.56-2.el7
79. 作为依赖被安装:
80. keyutils-libs-devel.x86\_64 0:1.5.8-3.el7 krb5-devel.x86\_64 0:1.15.1-18.el7 libaio.x86\_64 0:0.3.109-13.el7
81. libcom\_err-devel.x86\_64 0:1.42.9-11.el7 libkadm5.x86\_64 0:1.15.1-18.el7 libselinux-devel.x86\_64 0:2.5-12.el7
82. libsepol-devel.x86\_64 0:2.5-8.1.el7 libverto-devel.x86\_64 0:0.2.5-4.el7 openssl-devel.x86\_64 1:1.0.2k-12.el7
83. perl-Compress-Raw-Bzip2.x86\_64 0:2.061-3.el7 perl-Compress-Raw-Zlib.x86\_64 1:2.061-4.el7 perl-DBD-MySQL.x86\_64 0:4.023-6.el7
84. perl-DBI.x86\_64 0:1.627-4.el7 perl-Data-Dumper.x86\_64 0:2.145-3.el7 perl-IO-Compress.noarch 0:2.061-2.el7
85. perl-Net-Daemon.noarch 0:0.48-5.el7 perl-PlRPC.noarch 0:0.2020-14.el7
86. 完毕！
87. [root@web33 ~]#

[root@web44 ~]# yum -y install gcc zlib-devel pcre-devel //安装源码Nginx依赖软件

已安装:

gcc.x86\_64 0:4.8.5-28.el7 pcre-devel.x86\_64 0:8.32-17.el7 zlib-devel.x86\_64 0:1.2.7-17.el7

作为依赖被安装:

cpp.x86\_64 0:4.8.5-28.el7 glibc-devel.x86\_64 0:2.17-222.el7 glibc-headers.x86\_64 0:2.17-222.el7 kernel-headers.x86\_64 0:3.10.0-862.el7 libmpc.x86\_64 0:1.0.1-3.el7

mpfr.x86\_64 0:3.1.1-4.el7

完毕！

[root@web44 ~]#

[root@web44 ~]# tar -zxvf nginx-1.12.2.tar.gz //解压

[root@web44 ~]# cd nginx-1.12.2 //进源码目录

[root@web44 nginx-1.12.2]# ./configure //配置

......

Configuration summary

+ using system PCRE library

+ OpenSSL library is not used

+ using system zlib library

nginx path prefix: "/usr/local/nginx"

nginx binary file: "/usr/local/nginx/sbin/nginx"

nginx modules path: "/usr/local/nginx/modules"

nginx configuration prefix: "/usr/local/nginx/conf"

nginx configuration file: "/usr/local/nginx/conf/nginx.conf"

nginx pid file: "/usr/local/nginx/logs/nginx.pid"

nginx error log file: "/usr/local/nginx/logs/error.log"

nginx http access log file: "/usr/local/nginx/logs/access.log"

nginx http client request body temporary files: "client\_body\_temp"

nginx http proxy temporary files: "proxy\_temp"

nginx http fastcgi temporary files: "fastcgi\_temp"

nginx http uwsgi temporary files: "uwsgi\_temp"

nginx http scgi temporary files: "scgi\_temp"

[root@web44 nginx-1.12.2]# make //编译

……

……

sed -e "s|%%PREFIX%%|/usr/local/nginx|" \

-e "s|%%PID\_PATH%%|/usr/local/nginx/logs/nginx.pid|" \

-e "s|%%CONF\_PATH%%|/usr/local/nginx/conf/nginx.conf|" \

-e "s|%%ERROR\_LOG\_PATH%%|/usr/local/nginx/logs/error.log|" \

< man/nginx.8 > objs/nginx.8

make[1]: 离开目录“/root/nginx-1.12.2”

[root@web44 nginx-1.12.2]#

[root@web44 nginx-1.12.2]# make install //安装

……

……

cp conf/nginx.conf '/usr/local/nginx/conf/nginx.conf.default'

test -d '/usr/local/nginx/logs' \

|| mkdir -p '/usr/local/nginx/logs'

test -d '/usr/local/nginx/logs' \

|| mkdir -p '/usr/local/nginx/logs'

test -d '/usr/local/nginx/html' \

|| cp -R html '/usr/local/nginx'

test -d '/usr/local/nginx/logs' \

|| mkdir -p '/usr/local/nginx/logs'

make[1]: 离开目录“/root/nginx-1.12.2”

[root@web44 nginx-1.12.2]#

[root@web44 nginx-1.12.2]# ls /usr/local/nginx //查看安装目录

conf html logs sbin

[root@web44 nginx-1.12.2]#

[root@web44 ~]# yum -y install php-fpm //安装php-fpm 软件

……

……

已安装:

php-fpm.x86\_64 0:5.4.16-45.el7

作为依赖被安装:

libzip.x86\_64 0:0.10.1-8.el7 php-common.x86\_64 0:5.4.16-45.el7

完毕！

[root@web44 ~]# yum -y install php php-mysql //安装php 及 php-mysql 软件

……

……

已安装:

php.x86\_64 0:5.4.16-45.el7 php-mysql.x86\_64 0:5.4.16-45.el7

作为依赖被安装:

mariadb-libs.x86\_64 1:5.5.56-2.el7 php-cli.x86\_64 0:5.4.16-45.el7 php-pdo.x86\_64 0:5.4.16-45.el7

完毕！

[root@web44 ~]#

[root@web44 ~]# yum -y install mariadb-server mariadb-devel mariadb //安装mariadb服务软件

……

……

已安装:

mariadb.x86\_64 1:5.5.56-2.el7 mariadb-devel.x86\_64 1:5.5.56-2.el7 mariadb-server.x86\_64 1:5.5.56-2.el7

作为依赖被安装:

keyutils-libs-devel.x86\_64 0:1.5.8-3.el7 krb5-devel.x86\_64 0:1.15.1-18.el7 libaio.x86\_64 0:0.3.109-13.el7

libcom\_err-devel.x86\_64 0:1.42.9-11.el7 libkadm5.x86\_64 0:1.15.1-18.el7 libselinux-devel.x86\_64 0:2.5-12.el7

libsepol-devel.x86\_64 0:2.5-8.1.el7 libverto-devel.x86\_64 0:0.2.5-4.el7 openssl-devel.x86\_64 1:1.0.2k-12.el7

perl-Compress-Raw-Bzip2.x86\_64 0:2.061-3.el7 perl-Compress-Raw-Zlib.x86\_64 1:2.061-4.el7 perl-DBD-MySQL.x86\_64 0:4.023-6.el7

perl-DBI.x86\_64 0:1.627-4.el7 perl-Data-Dumper.x86\_64 0:2.145-3.el7 perl-IO-Compress.noarch 0:2.061-2.el7

perl-Net-Daemon.noarch 0:0.48-5.el7 perl-PlRPC.noarch 0:0.2020-14.el7

完毕！

[root@web44 ~]#

2）挂载共享存储

1. [root@web33 ~]# vim /etc/fstab //开机挂载
2. 192.168.4.30/sitedir /usr/local/nginx/html nfs defaults 0 0
3. :wq
4. [root@web33 ~]# mount -a //挂载设备
5. [root@web33 ~]# mount | grep "/usr/local/nginx/html" //查看挂载
6. 192.168.4.30:/sitedir on /usr/local/nginx/html type nfs4 (rw,relatime,vers=4.1,rsize=262144,wsize=262144,namlen=255,hard,proto=tcp,port=0,timeo=600,retrans=2,sec=sys,clientaddr=192.168.4.33,local\_lock=none,addr=192.168.4.30)
7. [root@web33 ~]#
8. [root@web44 ~]# vim /etc/fstab //开机挂载
9. 192.168.4.30/sitedir /usr/local/nginx/html nfs defaults 0 0
10. :wq
11. [root@web44 ~]# mount -a //挂载设备
12. [root@web44 ~]# mount | grep "/usr/local/nginx/html" //查看挂载
13. 192.168.4.30:/sitedir on /usr/local/nginx/html type nfs4 (rw,relatime,vers=4.1,rsize=262144,wsize=262144,namlen=255,hard,proto=tcp,port=0,timeo=600,retrans=2,sec=sys,clientaddr=192.168.4.33,local\_lock=none,addr=192.168.4.30)
14. [root@web44 ~]#

3）启动服务

1. [root@web33 ~]# vim +65 /usr/local/nginx/conf/nginx.conf //修改主配置文件
2. location ~ \.php$ {
3. root html;
4. fastcgi\_pass 127.0.0.1:9000;
5. fastcgi\_index index.php;
6. # fastcgi\_param SCRIPT\_FILENAME /scripts$fastcgi\_script\_name;
7. include fastcgi.conf;
8. }
9. :wq
10. [root@web33 ~]# /usr/local/nginx/sbin/nginx //启动服务
11. [root@web33 ~]#
12. [root@web33 ~]# netstat -utnlp | grep :80 //查看端口
13. tcp 0 0 0.0.0.0:80 0.0.0.0:\* LISTEN 26335/nginx: master
14. [root@web33 ~]#
15. [root@web33 ~]# systemctl start php-fpm
16. [root@web33 ~]#
17. [root@web33 ~]# netstat -utnlp | grep :9000
18. tcp 0 0 127.0.0.1:9000 0.0.0.0:\* LISTEN 26345/php-fpm: mast
19. [root@web33 ~]#
20. [root@web44 ~]# vim +65 /usr/local/nginx/conf/nginx.conf //修改主配置文件
21. location ~ \.php$ {
22. root html;
23. fastcgi\_pass 127.0.0.1:9000;
24. fastcgi\_index index.php;
25. # fastcgi\_param SCRIPT\_FILENAME /scripts$fastcgi\_script\_name;
26. include fastcgi.conf;
27. }
28. :wq
29. [root@web44 ~]# /usr/local/nginx/sbin/nginx //启动服务
30. [root@web44 ~]#
31. [root@web44 ~]# netstat -utnlp | grep :80 //查看端口
32. tcp 0 0 0.0.0.0:80 0.0.0.0:\* LISTEN 26335/nginx: master
33. [root@web44 ~]#
34. [root@web44 ~]# systemctl start php-fpm
35. [root@web44 ~]#
36. [root@web44 ~]# netstat -utnlp | grep :9000
37. tcp 0 0 127.0.0.1:9000 0.0.0.0:\* LISTEN 26345/php-fpm: mast
38. [root@web44 ~]#

4）测试配置

1. [root@nfs30 ~]# vim /sitedir/test2.php //在nfs30共享目录编写php脚本文件
2. <?php
3. $school="tarena" ; //定义变量
4. echo $school ; //输出变量值
5. ?>
6. :wq
7. [root@client50 ~]# curl http://192.168.4.33/test2.php //访问web33服务器
8. tarena
9. [root@client50 ~]# curl http://192.168.4.44/test2.php //访问web44服务器
10. tarena

## 2 案例2：部署缓存服务

### 2.1 问题

具体操作如下：

1. 部署redis服务器
2. 创建redis集群
3. 配置网站服务器
4. 测试配置

### 2.2 方案

克隆7台虚拟机配置要求如图-1所示。



图-1

### 2.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：部署redis服务器（6台都要配置）

1）搭建redis服务器

1. [root@redisA ~]# rpm -q gcc || yum -y install gcc //安装编译工具
2. [root@redisA ~]# tar -zxvf redis-4.0.8.tar.gz //解压
3. [root@redisA ~]# cd redis-4.0.8/ //进源码目录
4. [root@redisA redis-4.0.8]# make install //安装软件
5. ……
6. ……
7. INSTALL install
8. INSTALL install
9. INSTALL install
10. INSTALL install
11. INSTALL install
12. make[1]: 离开目录“/root/redis-4.0.8/src”
13. [root@redisA redis-4.0.8]#
14. [root@redisA redis-4.0.8]# ./utils/install\_server.sh //初始化配置
15. Welcome to the redis service installer
16. This script will help you easily set up a running redis server
17. Please select the redis port for this instance: [6379] //端口号
18. Selecting default: 6379
19. Please select the redis config file name [/etc/redis/6379.conf] //主配置文件
20. Selected default - /etc/redis/6379.conf
21. Please select the redis log file name [/var/log/redis\_6379.log] //日志文件
22. Selected default - /var/log/redis\_6379.log
23. Please select the data directory for this instance [/var/lib/redis/6379] //数据库目录
24. Selected default - /var/lib/redis/6379
25. Please select the redis executable path [/usr/local/bin/redis-server] //服务启动启动程序
26. Selected config: //配置总结
27. Port : 6379
28. Config file : /etc/redis/6379.conf
29. Log file : /var/log/redis\_6379.log
30. Data dir : /var/lib/redis/6379
31. Executable : /usr/local/bin/redis-server
32. Cli Executable : /usr/local/bin/redis-cli
33. Is this ok? Then press ENTER to go on or Ctrl-C to abort.
34. Copied /tmp/6379.conf => /etc/init.d/redis\_6379
35. Installing service...
36. Successfully added to chkconfig!
37. Successfully added to runlevels 345!
38. Starting Redis server... //服务启动提示
39. Installation successful! //安装完成提示
40. [root@redisA redis-4.0.8]#
41. [root@redisA redis-4.0.8]# /etc/init.d/redis\_6379 stop //停止服务
42. Stopping ...
43. Redis stopped
44. [root@redisA redis-4.0.8]#
45. [root@redisA redis-4.0.8]# vim /etc/redis/6379.conf //修改配置文件，启用集群配置
46. 70 bind 192.168.4.51
47. 815 cluster-enabled yes
48. 823 cluster-config-file nodes-6379.conf
49. 829 cluster-node-timeout 5000
50. :wq
51. [root@redisA redis-4.0.8]# /etc/init.d/redis\_6379 start //启动服务
52. Starting Redis server...
53. [root@redisA redis-4.0.8]# netstat -utnlp | grep redis-server //查看端口
54. tcp 0 0 192.168.4.51:6379 0.0.0.0:\* LISTEN 29720/redis-server //redis服务端口
55. tcp 0 0 192.168.4.51:16379 0.0.0.0:\* LISTEN 29720/redis-server //集群端口
56. [root@redisB ~]# rpm -q gcc || yum -y install gcc //安装编译工具
57. [root@redisB ~]# tar -zxvf redis-4.0.8.tar.gz //解压
58. [root@redisB ~]# cd redis-4.0.8/ //进源码目录
59. [root@redisB redis-4.0.8]# make install //安装软件
60. ……
61. ……
62. INSTALL install
63. INSTALL install
64. INSTALL install
65. INSTALL install
66. INSTALL install
67. make[1]: 离开目录“/root/redis-4.0.8/src”
68. [root@redisB redis-4.0.8]#
69. [root@redisB redis-4.0.8]# ./utils/install\_server.sh //初始化配置
70. Welcome to the redis service installer
71. This script will help you easily set up a running redis server
72. Please select the redis port for this instance: [6379] //端口号
73. Selecting default: 6379
74. Please select the redis config file name [/etc/redis/6379.conf] //主配置文件
75. Selected default - /etc/redis/6379.conf
76. Please select the redis log file name [/var/log/redis\_6379.log] //日志文件
77. Selected default - /var/log/redis\_6379.log
78. Please select the data directory for this instance [/var/lib/redis/6379] //数据库目录
79. Selected default - /var/lib/redis/6379
80. Please select the redis executable path [/usr/local/bin/redis-server] //服务启动启动程序
81. Selected config: //配置总结
82. Port : 6379
83. Config file : /etc/redis/6379.conf
84. Log file : /var/log/redis\_6379.log
85. Data dir : /var/lib/redis/6379
86. Executable : /usr/local/bin/redis-server
87. Cli Executable : /usr/local/bin/redis-cli
88. Is this ok? Then press ENTER to go on or Ctrl-C to abort.
89. Copied /tmp/6379.conf => /etc/init.d/redis\_6379
90. Installing service...
91. Successfully added to chkconfig!
92. Successfully added to runlevels 345!
93. Starting Redis server... //服务启动提示
94. Installation successful! //安装完成提示
95. [root@redisB redis-4.0.8]#
96. [root@redisB redis-4.0.8]# /etc/init.d/redis\_6379 stop //停止服务
97. Stopping ...
98. Redis stopped
99. [root@redisB redis-4.0.8]#
100. [root@redisB redis-4.0.8]# vim /etc/redis/6379.conf //修改配置文件，启用集群配置
101. 70 bind 192.168.4.52
102. 815 cluster-enabled yes
103. 823 cluster-config-file nodes-6379.conf
104. 829 cluster-node-timeout 5000
105. :wq
106. [root@redisB redis-4.0.8]# /etc/init.d/redis\_6379 start //启动服务
107. Starting Redis server...
108. [root@redisB redis-4.0.8]# netstat -utnlp | grep redis-server //查看端口
109. tcp 0 0 192.168.4.52:6379 0.0.0.0:\* LISTEN 29720/redis-server //redis服务端口
110. tcp 0 0 192.168.4.52:16379 0.0.0.0:\* LISTEN 29720/redis-server //集群端口
111. [root@redisC ~]# rpm -q gcc || yum -y install gcc //安装编译工具
112. [root@redisC ~]# tar -zxvf redis-4.0.8.tar.gz //解压
113. [root@redisC ~]# cd redis-4.0.8/ //进源码目录
114. [root@redisC redis-4.0.8]# make install //安装软件
115. ……
116. ……
117. INSTALL install
118. INSTALL install
119. INSTALL install
120. INSTALL install
121. INSTALL install
122. make[1]: 离开目录“/root/redis-4.0.8/src”
123. [root@redisC redis-4.0.8]#
124. [root@redisC redis-4.0.8]# ./utils/install\_server.sh //初始化配置
125. Welcome to the redis service installer
126. This script will help you easily set up a running redis server
127. Please select the redis port for this instance: [6379] //端口号
128. Selecting default: 6379
129. Please select the redis config file name [/etc/redis/6379.conf] //主配置文件
130. Selected default - /etc/redis/6379.conf
131. Please select the redis log file name [/var/log/redis\_6379.log] //日志文件
132. Selected default - /var/log/redis\_6379.log
133. Please select the data directory for this instance [/var/lib/redis/6379] //数据库目录
134. Selected default - /var/lib/redis/6379
135. Please select the redis executable path [/usr/local/bin/redis-server] //服务启动启动程序
136. Selected config: //配置总结
137. Port : 6379
138. Config file : /etc/redis/6379.conf
139. Log file : /var/log/redis\_6379.log
140. Data dir : /var/lib/redis/6379
141. Executable : /usr/local/bin/redis-server
142. Cli Executable : /usr/local/bin/redis-cli
143. Is this ok? Then press ENTER to go on or Ctrl-C to abort.
144. Copied /tmp/6379.conf => /etc/init.d/redis\_6379
145. Installing service...
146. Successfully added to chkconfig!
147. Successfully added to runlevels 345!
148. Starting Redis server... //服务启动提示
149. Installation successful! //安装完成提示
150. [root@redisC redis-4.0.8]#
151. [root@redisC redis-4.0.8]# /etc/init.d/redis\_6379 stop //停止服务
152. Stopping ...
153. Redis stopped
154. [root@redisC redis-4.0.8]#
155. [root@redisC redis-4.0.8]# vim /etc/redis/6379.conf //修改配置文件，启用集群配置
156. 70 bind 192.168.4.53
157. 815 cluster-enabled yes
158. 823 cluster-config-file nodes-6379.conf
159. 829 cluster-node-timeout 5000
160. :wq
161. [root@redisC redis-4.0.8]# /etc/init.d/redis\_6379 start //启动服务
162. Starting Redis server...
163. [root@redisC redis-4.0.8]# netstat -utnlp | grep redis-server //查看端口
164. tcp 0 0 192.168.4.53:6379 0.0.0.0:\* LISTEN 29720/redis-server //redis服务端口
165. tcp 0 0 192.168.4.53:16379 0.0.0.0:\* LISTEN 29720/redis-server //集群端口
166. [root@redisD ~]# rpm -q gcc || yum -y install gcc //安装编译工具
167. [root@redisD ~]# tar -zxvf redis-4.0.8.tar.gz //解压
168. [root@redisD ~]# cd redis-4.0.8/ //进源码目录
169. [root@redisD redis-4.0.8]# make install //安装软件
170. ……
171. ……
172. INSTALL install
173. INSTALL install
174. INSTALL install
175. INSTALL install
176. INSTALL install
177. make[1]: 离开目录“/root/redis-4.0.8/src”
178. [root@redisD redis-4.0.8]#
179. [root@redisD redis-4.0.8]# ./utils/install\_server.sh //初始化配置
180. Welcome to the redis service installer
181. This script will help you easily set up a running redis server
182. Please select the redis port for this instance: [6379] //端口号
183. Selecting default: 6379
184. Please select the redis config file name [/etc/redis/6379.conf] //主配置文件
185. Selected default - /etc/redis/6379.conf
186. Please select the redis log file name [/var/log/redis\_6379.log] //日志文件
187. Selected default - /var/log/redis\_6379.log
188. Please select the data directory for this instance [/var/lib/redis/6379] //数据库目录
189. Selected default - /var/lib/redis/6379
190. Please select the redis executable path [/usr/local/bin/redis-server] //服务启动启动程序
191. Selected config: //配置总结
192. Port : 6379
193. Config file : /etc/redis/6379.conf
194. Log file : /var/log/redis\_6379.log
195. Data dir : /var/lib/redis/6379
196. Executable : /usr/local/bin/redis-server
197. Cli Executable : /usr/local/bin/redis-cli
198. Is this ok? Then press ENTER to go on or Ctrl-C to abort.
199. Copied /tmp/6379.conf => /etc/init.d/redis\_6379
200. Installing service...
201. Successfully added to chkconfig!
202. Successfully added to runlevels 345!
203. Starting Redis server... //服务启动提示
204. Installation successful! //安装完成提示
205. [root@redisD redis-4.0.8]#
206. [root@redisD redis-4.0.8]# /etc/init.d/redis\_6379 stop //停止服务
207. Stopping ...
208. Redis stopped
209. [root@redisD redis-4.0.8]#
210. [root@redisD redis-4.0.8]# vim /etc/redis/6379.conf //修改配置文件，启用集群配置
211. 70 bind 192.168.4.54
212. 815 cluster-enabled yes
213. 823 cluster-config-file nodes-6379.conf
214. 829 cluster-node-timeout 5000
215. :wq
216. [root@redisD redis-4.0.8]# /etc/init.d/redis\_6379 start //启动服务
217. Starting Redis server...
218. [root@redisD redis-4.0.8]# netstat -utnlp | grep redis-server //查看端口
219. tcp 0 0 192.168.4.54:6379 0.0.0.0:\* LISTEN 29720/redis-server //redis服务端口
220. tcp 0 0 192.168.4.54:16379 0.0.0.0:\* LISTEN 29720/redis-server //集群端口
221. [root@redisE ~]# rpm -q gcc || yum -y install gcc //安装编译工具
222. [root@redisE ~]# tar -zxvf redis-4.0.8.tar.gz //解压
223. [root@redisE ~]# cd redis-4.0.8/ //进源码目录
224. [root@redisE redis-4.0.8]# make install //安装软件
225. ……
226. ……
227. INSTALL install
228. INSTALL install
229. INSTALL install
230. INSTALL install
231. INSTALL install
232. make[1]: 离开目录“/root/redis-4.0.8/src”
233. [root@redisE redis-4.0.8]#
234. [root@redisE redis-4.0.8]# ./utils/install\_server.sh //初始化配置
235. Welcome to the redis service installer
236. This script will help you easily set up a running redis server
237. Please select the redis port for this instance: [6379] //端口号
238. Selecting default: 6379
239. Please select the redis config file name [/etc/redis/6379.conf] //主配置文件
240. Selected default - /etc/redis/6379.conf
241. Please select the redis log file name [/var/log/redis\_6379.log] //日志文件
242. Selected default - /var/log/redis\_6379.log
243. Please select the data directory for this instance [/var/lib/redis/6379] //数据库目录
244. Selected default - /var/lib/redis/6379
245. Please select the redis executable path [/usr/local/bin/redis-server] //服务启动启动程序
246. Selected config: //配置总结
247. Port : 6379
248. Config file : /etc/redis/6379.conf
249. Log file : /var/log/redis\_6379.log
250. Data dir : /var/lib/redis/6379
251. Executable : /usr/local/bin/redis-server
252. Cli Executable : /usr/local/bin/redis-cli
253. Is this ok? Then press ENTER to go on or Ctrl-C to abort.
254. Copied /tmp/6379.conf => /etc/init.d/redis\_6379
255. Installing service...
256. Successfully added to chkconfig!
257. Successfully added to runlevels 345!
258. Starting Redis server... //服务启动提示
259. Installation successful! //安装完成提示
260. [root@redisE redis-4.0.8]#
261. [root@redisE redis-4.0.8]# /etc/init.d/redis\_6379 stop //停止服务
262. Stopping ...
263. Redis stopped
264. [root@redisE redis-4.0.8]#
265. [root@redisE redis-4.0.8]# vim /etc/redis/6379.conf //修改配置文件，启用集群配置
266. 70 bind 192.168.4.56
267. 815 cluster-enabled yes
268. 823 cluster-config-file nodes-6379.conf
269. 829 cluster-node-timeout 5000
270. :wq
271. [root@redisE redis-4.0.8]# /etc/init.d/redis\_6379 start //启动服务
272. Starting Redis server...
273. [root@redisE redis-4.0.8]# netstat -utnlp | grep redis-server //查看端口
274. tcp 0 0 192.168.4.56:6379 0.0.0.0:\* LISTEN 29720/redis-server //redis服务端口
275. tcp 0 0 192.168.4.56:16379 0.0.0.0:\* LISTEN 29720/redis-server //集群端口
276. [root@redisF ~]# rpm -q gcc || yum -y install gcc //安装编译工具
277. [root@redisF ~]# tar -zxvf redis-4.0.8.tar.gz //解压
278. [root@redisF ~]# cd redis-4.0.8/ //进源码目录
279. [root@redisF redis-4.0.8]# make install //安装软件
280. ……
281. ……
282. INSTALL install
283. INSTALL install
284. INSTALL install
285. INSTALL install
286. INSTALL install
287. make[1]: 离开目录“/root/redis-4.0.8/src”
288. [root@redisF redis-4.0.8]#
289. [root@redisF redis-4.0.8]# ./utils/install\_server.sh //初始化配置
290. Welcome to the redis service installer
291. This script will help you easily set up a running redis server
292. Please select the redis port for this instance: [6379] //端口号
293. Selecting default: 6379
294. Please select the redis config file name [/etc/redis/6379.conf] //主配置文件
295. Selected default - /etc/redis/6379.conf
296. Please select the redis log file name [/var/log/redis\_6379.log] //日志文件
297. Selected default - /var/log/redis\_6379.log
298. Please select the data directory for this instance [/var/lib/redis/6379] //数据库目录
299. Selected default - /var/lib/redis/6379
300. Please select the redis executable path [/usr/local/bin/redis-server] //服务启动启动程序
301. Selected config: //配置总结
302. Port : 6379
303. Config file : /etc/redis/6379.conf
304. Log file : /var/log/redis\_6379.log
305. Data dir : /var/lib/redis/6379
306. Executable : /usr/local/bin/redis-server
307. Cli Executable : /usr/local/bin/redis-cli
308. Is this ok? Then press ENTER to go on or Ctrl-C to abort.
309. Copied /tmp/6379.conf => /etc/init.d/redis\_6379
310. Installing service...
311. Successfully added to chkconfig!
312. Successfully added to runlevels 345!
313. Starting Redis server... //服务启动提示
314. Installation successful! //安装完成提示
315. [root@redisF redis-4.0.8]#
316. [root@redisF redis-4.0.8]# /etc/init.d/redis\_6379 stop //停止服务
317. Stopping ...
318. Redis stopped
319. [root@redisF redis-4.0.8]#
320. [root@redisF redis-4.0.8]# vim /etc/redis/6379.conf //修改配置文件，启用集群配置
321. 70 bind 192.168.4.57
322. 815 cluster-enabled yes
323. 823 cluster-config-file nodes-6379.conf
324. 829 cluster-node-timeout 5000
325. :wq
326. [root@redisF redis-4.0.8]# /etc/init.d/redis\_6379 start //启动服务
327. Starting Redis server...
328. [root@redisF redis-4.0.8]# netstat -utnlp | grep redis-server //查看端口
329. tcp 0 0 192.168.4.57:6379 0.0.0.0:\* LISTEN 29720/redis-server //redis服务端口
330. tcp 0 0 192.168.4.57:16379 0.0.0.0:\* LISTEN 29720/redis-server //集群端口

步骤二：创建redis集群

1）配置管理主机

1. [root@mgm ~]# yum -y install ruby rubygems //安装依赖
2. ……
3. ……
4. 已安装:
5. ruby.x86\_64 0:2.0.0.648-33.el7\_4 rubygems.noarch 0:2.0.14.1-33.el7\_4
6. 作为依赖被安装:
7. libyaml.x86\_64 0:0.1.4-11.el7\_0 ruby-irb.noarch 0:2.0.0.648-33.el7\_4 ruby-libs.x86\_64 0:2.0.0.648-33.el7\_4 rubygem-bigdecimal.x86\_64 0:1.2.0-33.el7\_4
8. rubygem-io-console.x86\_64 0:0.4.2-33.el7\_4 rubygem-json.x86\_64 0:1.7.7-33.el7\_4 rubygem-psych.x86\_64 0:2.0.0-33.el7\_4 rubygem-rdoc.noarch 0:4.0.0-33.el7\_4
9. 完毕！
10. [root@mgm ~]#
11. [root@mgm ~]# gem install redis-3.2.1.gem //安装依赖软件gem程序
12. Successfully installed redis-3.2.1
13. Parsing documentation for redis-3.2.1
14. Installing ri documentation for redis-3.2.1
15. 1 gem installed
16. [root@mgm ~]#
17. [root@mgm ~]# tar -zxvf redis-4.0.8.tar.gz
18. [root@mgm ~]# cp redis-4.0.8/src/redis-trib.rb /root/bin/ //拷贝脚本
19. [root@mgm ~]#
20. [root@mgm ~]# chmod +x /root/bin/redis-trib.rb //确保脚本有执行权限
21. [root@mgm ~]#
22. [root@mgm ~]# redis-trib.rb help //查看帮助
23. Usage: redis-trib <command> <options> <arguments ...>
24. create host1:port1 ... hostN:portN
25. --replicas <arg>
26. check host:port
27. info host:port
28. fix host:port
29. --timeout <arg>
30. reshard host:port
31. --from <arg>
32. --to <arg>
33. --slots <arg>
34. --yes
35. --timeout <arg>
36. --pipeline <arg>
37. rebalance host:port
38. --weight <arg>
39. --auto-weights
40. --use-empty-masters
41. --timeout <arg>
42. --simulate
43. --pipeline <arg>
44. --threshold <arg>
45. add-node new\_host:new\_port existing\_host:existing\_port
46. --slave
47. --master-id <arg>
48. del-node host:port node\_id
49. set-timeout host:port milliseconds
50. call host:port command arg arg .. arg
51. import host:port
52. --from <arg>
53. --copy
54. --replace
55. help (show this help)
56. For check, fix, reshard, del-node, set-timeout you can specify the host and port of any working node in the cluster.
57. [root@mgm ~]#

2）创建集群

1. ]# redis-trib.rb create --replicas 1 \
2. 192.168.4.51:6379 192.168.4.52:6379 192.168.4.53:6379 \ 192.168.4.54:6379 192.168.4.56:6379 192.168.4.57:6379
3. >>> Performing hash slots allocation on 6 nodes...
4. Using 3 masters:
5. 192.168.4.51:6379
6. 192.168.4.52:6379
7. 192.168.4.53:6379
8. Adding replica 192.168.4.57:6379 to 192.168.4.51:6379
9. Adding replica 192.168.4.56:6379 to 192.168.4.52:6379
10. Adding replica 192.168.4.54:6379 to 192.168.4.53:6379
11. M: d9f8fe6d6d9dd391be8e7904501db1535e4d17cb 192.168.4.51:6379
12. slots:0-5460 (5461 slots) master
13. M: 324e05df3f143ef97e50d09be0328a695e655986 192.168.4.52:6379
14. slots:5461-10922 (5462 slots) master
15. M: 9e44139cffb8ebd7ed746aabbf4bcea9bf207645 192.168.4.53:6379
16. slots:10923-16383 (5461 slots) master
17. S: d9634ba0aa5c1a07193da4a013da6051c1515922 192.168.4.54:6379
18. replicates 9e44139cffb8ebd7ed746aabbf4bcea9bf207645
19. S: 2d343a9df48f6f6e207949e980ef498466a44dad 192.168.4.57:6379
20. replicates d9f8fe6d6d9dd391be8e7904501db1535e4d17cb
21. S: 894dd0008053f6fb65e9e4a36b755d9351607500 192.168.4.56:6379
22. replicates 324e05df3f143ef97e50d09be0328a695e655986
23. Can I set the above configuration? (type 'yes' to accept): yes //同意以上配置
24. >>> Nodes configuration updated
25. >>> Assign a different config epoch to each node
26. >>> Sending CLUSTER MEET messages to join the cluster
27. Waiting for the cluster to join...
28. >>> Performing Cluster Check (using node 192.168.4.51:6379)
29. M: d9f8fe6d6d9dd391be8e7904501db1535e4d17cb 192.168.4.51:6379
30. slots:0-5460 (5461 slots) master
31. 1 additional replica(s)
32. S: d9634ba0aa5c1a07193da4a013da6051c1515922 192.168.4.54:6379
33. slots: (0 slots) slave
34. replicates 9e44139cffb8ebd7ed746aabbf4bcea9bf207645
35. S: 894dd0008053f6fb65e9e4a36b755d9351607500 192.168.4.56:6379
36. slots: (0 slots) slave
37. replicates 324e05df3f143ef97e50d09be0328a695e655986
38. M: 324e05df3f143ef97e50d09be0328a695e655986 192.168.4.52:6379
39. slots:5461-10922 (5462 slots) master
40. 1 additional replica(s)
41. M: 9e44139cffb8ebd7ed746aabbf4bcea9bf207645 192.168.4.53:6379
42. slots:10923-16383 (5461 slots) master
43. 1 additional replica(s)
44. S: 2d343a9df48f6f6e207949e980ef498466a44dad 192.168.4.57:6379
45. slots: (0 slots) slave
46. replicates d9f8fe6d6d9dd391be8e7904501db1535e4d17cb
47. [OK] All nodes agree about slots configuration.
48. >>> Check for open slots...
49. >>> Check slots coverage...
50. [OK] All 16384 slots covered. //提示16384个槽分配完毕
51. [root@mgm ~]#

3）查看集群信息

1. [root@mgm ~]# redis-trib.rb info 192.168.4.51:6379 //查看集群信息
2. 192.168.4.51:6379 (d9f8fe6d...) -> 0 keys | 5461 slots | 1 slaves.
3. 192.168.4.52:6379 (324e05df...) -> 0 keys | 5462 slots | 1 slaves.
4. 192.168.4.53:6379 (9e44139c...) -> 0 keys | 5461 slots | 1 slaves.
5. [OK] 0 keys in 3 masters.
6. keys per slot on average

[root@mgm ~]# redis-trib.rb check 192.168.4.51:6379 //检测集群

>>> Performing Cluster Check (using node 192.168.4.51:6379)

M: d9f8fe6d6d9dd391be8e7904501db1535e4d17cb 192.168.4.51:6379

slots:0-5460 (5461 slots) master

1 additional replica(s)

S: d9634ba0aa5c1a07193da4a013da6051c1515922 192.168.4.54:6379

slots: (0 slots) slave

replicates 9e44139cffb8ebd7ed746aabbf4bcea9bf207645

S: 894dd0008053f6fb65e9e4a36b755d9351607500 192.168.4.56:6379

slots: (0 slots) slave

replicates 324e05df3f143ef97e50d09be0328a695e655986

M: 324e05df3f143ef97e50d09be0328a695e655986 192.168.4.52:6379

slots:5461-10922 (5462 slots) master

1 additional replica(s)

M: 9e44139cffb8ebd7ed746aabbf4bcea9bf207645 192.168.4.53:6379

slots:10923-16383 (5461 slots) master

1 additional replica(s)

S: 2d343a9df48f6f6e207949e980ef498466a44dad 192.168.4.57:6379

slots: (0 slots) slave

replicates d9f8fe6d6d9dd391be8e7904501db1535e4d17cb

[OK] All nodes agree about slots configuration.

>>> Check for open slots...

>>> Check slots coverage...

[OK] All 16384 slots covered.

4）测试配置（在客户端连接集群中的任意一台服务器存取数据）

1. [root@client50 ~]# redis-cli -c -h 192.168.4.51 -p 6379 //连接服务器51
2. 192.168.4.51:6379>
3. 192.168.4.51:6379> set x 100 //存储
4. -> Redirected to slot [16287] located at 192.168.4.53:6379 //提示存储在53主机
5. OK
6. 192.168.4.53:6379> keys \*
7. 1) "x"
8. 192.168.4.53:6379>
9. 192.168.4.53:6379> set y 200
10. OK
11. 192.168.4.53:6379> keys \*
12. 1) "y"
13. 2) "x"
14. 192.168.4.53:6379> set z 300 //存储
15. -> Redirected to slot [8157] located at 192.168.4.52:6379 //提示存储在52主机
16. OK
17. 192.168.4.52:6379> keys \* //在52主机查看数据 只有变量z
18. 1) "z"
19. 192.168.4.52:6379> get x
20. -> Redirected to slot [16287] located at 192.168.4.53:6379 //连接53主机获取数据
21. "100"
22. 192.168.4.53:6379> keys \*
23. 1) "y"
24. 2) "x"
25. 192.168.4.53:6379> get z
26. -> Redirected to slot [8157] located at 192.168.4.52:6379
27. "300"
28. 192.168.4.52:6379> set i 400
29. -> Redirected to slot [15759] located at 192.168.4.53:6379
30. OK
31. 192.168.4.53:6379> set j 500
32. -> Redirected to slot [3564] located at 192.168.4.51:6379
33. OK
34. 192.168.4.51:6379>

步骤三：配置网站服务器 （2台网站服务器都要配置）

1. [root@web33 ~ ]# yum -y install php-devel //安装依赖
2. ……
3. ……
4. 已安装:
5. php-devel.x86\_64 0:5.4.16-45.el7
6. 作为依赖被安装:
7. autoconf.noarch 0:2.69-11.el7 automake.noarch 0:1.13.4-3.el7 m4.x86\_64 0:1.4.16-10.el7 perl-Test-Harness.noarch 0:3.28-3.el7 perl-Thread-Queue.noarch 0:3.02-2.el7
8. 完毕！
9. [root@web33 ~]#
10. [root@web33 ~]# tar -zxvf redis-cluster-4.3.0.tgz //解压
11. [root@web33 ~]# cd redis-4.3.0/ //进入源码目录
12. [root@web33 redis-4.3.0]# phpize //创建configure命令及配置信息文件/usr/bin/php-config
13. Configuring for:
14. PHP Api Version: 20100412
15. Zend Module Api No: 20100525
16. Zend Extension Api No: 220100525
17. [root@web33 redis-4.3.0]#
18. [root@web33 redis-4.3.0]# ./configure --with-php-config=/usr/bin/php-config
19. ……
20. ……
21. configure: creating ./config.status
22. config.status: creating config.h
23. config.status: config.h is unchanged
24. config.status: executing libtool commands
25. [root@web33 redis-4.3.0]#
26. [root@web33 redis-4.3.0]# make //编译
27. ……
28. ……
29. Build complete.
30. Don't forget to run 'make test'.
31. [root@web33 redis-4.3.0]#
32. [root@web33 redis-4.3.0]# make install //安装
33. Installing shared extensions: /usr/lib64/php/modules/ //提示模块安装目录
34. [root@web33 redis-4.3.0]#
35. [root@web33 redis-4.3.0]# ls /usr/lib64/php/modules/ //查看目录列表
36. curl.so fileinfo.so json.so mysqli.so mysql.so pdo\_mysql.so pdo.so pdo\_sqlite.so phar.so redis.so sqlite3.so zip.so
37. [root@web33 redis-4.3.0]#

[root@web44 ~ ]# yum -y install php-devel //安装依赖

……

……

已安装:

php-devel.x86\_64 0:5.4.16-45.el7

作为依赖被安装:

autoconf.noarch 0:2.69-11.el7 automake.noarch 0:1.13.4-3.el7 m4.x86\_64 0:1.4.16-10.el7 perl-Test-Harness.noarch 0:3.28-3.el7 perl-Thread-Queue.noarch 0:3.02-2.el7

完毕！

[root@web44 ~]#

[root@web44 ~]# tar -zxvf redis-cluster-4.3.0.tgz //解压

[root@web44 ~]# cd redis-4.3.0/ //进入源码目录

[root@web44 redis-4.3.0]# phpize //创建configure命令及配置信息文件/usr/bin/php-config

Configuring for:

PHP Api Version: 20100412

Zend Module Api No: 20100525

Zend Extension Api No: 220100525

[root@web44 redis-4.3.0]#

[root@web44 redis-4.3.0]# ./configure --with-php-config=/usr/bin/php-config

……

……

configure: creating ./config.status

config.status: creating config.h

config.status: config.h is unchanged

config.status: executing libtool commands

[root@web44 redis-4.3.0]#

[root@web44 redis-4.3.0]# make //编译

……

……

Build complete.

Don't forget to run 'make test'.

[root@web44 redis-4.3.0]#

[root@web44 redis-4.3.0]# make install //安装

Installing shared extensions: /usr/lib64/php/modules/ //提示模块安装目录

[root@web44 redis-4.3.0]#

[root@web44 redis-4.3.0]# ls /usr/lib64/php/modules/ //查看目录列表

curl.so fileinfo.so json.so mysqli.so mysql.so pdo\_mysql.so pdo.so pdo\_sqlite.so phar.so redis.so sqlite3.so zip.so

[root@web44 redis-4.3.0]#

修改配置文件

1. [root@web33 redis-4.3.0]# vim /etc/php.ini
2. 728 extension\_dir = "/usr/lib64/php/modules/" //模块目录
3. 730 extension = "redis.so" //模块名
4. :wq
5. [root@web33 redis-4.3.0]# systemctl restart php-fpm //重启php-fpm服务
6. [root@web33 redis-4.3.0]# php -m | grep -i redis //查看模块
7. redis
8. [root@web33 redis-4.3.0]#
9. [root@web44 redis-4.3.0]# vim /etc/php.ini
10. 728 extension\_dir = "/usr/lib64/php/modules/" //模块目录
11. 730 extension = "redis.so" //模块名
12. :wq
13. [root@web44 redis-4.3.0]# systemctl restart php-fpm //重启php-fpm服务
14. [root@web44 redis-4.3.0]# php -m | grep -i redis //查看模块
15. redis
16. [root@web44 redis-4.3.0]#

步骤四：测试配置

1）在存储服务器共享目录下，创建连接集群PHP脚本

1. nfs30~ ]# vim /sitedir/set\_data.php //存储数据脚本
2. <?php
3. $redis\_list = ['192.168.4.51:6379','192.168.4.52:6379','192.168.4.53:6379','192.168.4.54:6379','192.168.4.56:6379','192.168.4.57:6379']; //定义redis服务器列表
4. $client = new RedisCluster(NUll,$redis\_list); //定义连接redis服务器变量
5. $client->set("i","tarenaA "); //存储数据 变量名 i
6. $client->set("j","tarenaB "); //存储数据 变量名 j
7. $client->set("k","tarenaC "); //存储数据 变量名 k
8. ?>
9. :wq
10. nfs30~ ]# vim /sitedir/get\_data.php //获取数据脚本
11. <?php
12. $redis\_list = ['192.168.4.51:6379','192.168.4.52:6379','192.168.4.53:6379','192.168.4.54:6379','192.168.4.56:6379','192.168.4.57:6379']; //定义redis服务器列表
13. $client = new RedisCluster(NUll,$redis\_list); //定义连接redis服务器变量
14. echo $client->get("i"); //获取变量i 的数据
15. echo $client->get("j"); //获取变量j 的数据
16. echo $client->get("k"); //获取变量k 的数据
17. ?>
18. :wq
19. nfs30~ ]# vim /sitedir/test3.php //存/取数据脚本
20. <?php
21. $redis\_list = ['192.168.4.51:6379','192.168.4.52:6379','192.168.4.53:6379','192.168.4.54:6379','192.168.4.56:6379','192.168.4.57:6379'];
22. $client = new RedisCluster(NUll,$redis\_list);
23. $client->set(“name“,”panglijing”); //存数据
24. echo $client->get(“name”); //取数据
25. ?>
26. :wq

2）访问网站执行脚本(在任意主机访问网站服务器都可以)

1. ]# curl http://192.168.4.33/set\_data.php
2. ]# curl http://192.168.4.33/get\_data.php
3. ]# curl http://192.168.4.33/test3.php

3）命令行连接任意一台redis服务器查看数据(在任意主机连接redis服务器都可以)

1. ]# redis-cli -c -h 192.168.4.51 -p 6379
2. 192.168.4.51:6379> keys \*
3. 1) i
4. 192.168.4.51:6379> exit
5. ]# redis-cli -c -h 192.168.4.52 -p 6379
6. 192.168.4.52:6379> keys \*
7. 1) j
8. 192.168.4.52:6379> exit

## 3 案例3：数据迁移

### 3.1 问题

要求如下：

1. 配置从服务器
2. 配置第1台PXC服务器
3. 配置第2台PXC服务器
4. 配置第3台PXC服务器
5. 公共配置
6. 测试配置

### 3.2 方案

创建3台新的虚拟机，具体配置要求如图-2所示。



图-2

### 3.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：配置从服务器(把主机192.168.4.66 配置为192.168.4.11的从服务器)

1）在192.168.4.66主机安装数据库服务软件并启动mysqld服务

1. [root@pxcnode66 ~]# tar -xvf mysql-5.7.17.tar //解包
2. ./mysql-community-client-5.7.17-1.el7.x86\_64.rpm
3. ./mysql-community-common-5.7.17-1.el7.x86\_64.rpm
4. ./mysql-community-devel-5.7.17-1.el7.x86\_64.rpm
5. ./mysql-community-embedded-5.7.17-1.el7.x86\_64.rpm
6. ./mysql-community-embedded-compat-5.7.17-1.el7.x86\_64.rpm
7. ./mysql-community-embedded-devel-5.7.17-1.el7.x86\_64.rpm
8. ./mysql-community-libs-5.7.17-1.el7.x86\_64.rpm
9. ./mysql-community-libs-compat-5.7.17-1.el7.x86\_64.rpm
10. ./mysql-community-minimal-debuginfo-5.7.17-1.el7.x86\_64.rpm
11. ./mysql-community-server-5.7.17-1.el7.x86\_64.rpm
12. ./mysql-community-test-5.7.17-1.el7.x86\_64.rpm
13. [root@pxcnode66 ~]#
14. [root@pxcnode66 ~]# ls \*.rpm //查看软件列表
15. mysql-community-client-5.7.17-1.el7.x86\_64.rpm
16. mysql-community-common-5.7.17-1.el7.x86\_64.rpm
17. mysql-community-devel-5.7.17-1.el7.x86\_64.rpm
18. mysql-community-embedded-5.7.17-1.el7.x86\_64.rpm
19. mysql-community-embedded-compat-5.7.17-1.el7.x86\_64.rpm
20. mysql-community-embedded-devel-5.7.17-1.el7.x86\_64.rpm
21. mysql-community-libs-5.7.17-1.el7.x86\_64.rpm
22. mysql-community-libs-compat-5.7.17-1.el7.x86\_64.rpm
23. mysql-community-minimal-debuginfo-5.7.17-1.el7.x86\_64.rpm
24. mysql-community-server-5.7.17-1.el7.x86\_64.rpm
25. mysql-community-test-5.7.17-1.el7.x86\_64.rpm
26. [root@pxcnode66 ~]#
27. [root@pxcnode66 ~]# yum -y install mysql-community-\*.rpm //安装软件
28. 已加载插件：fastestmirror
29. 正在检查 mysql-community-client-5.7.17-1.el7.x86\_64.rpm: mysql-community-client-5.7.17-1.el7.x86\_64
30. mysql-community-client-5.7.17-1.el7.x86\_64.rpm 将被安装
31. ……
32. ……
33. 已安装:
34. mysql-community-client.x86\_64 0:5.7.17-1.el7 mysql-community-common.x86\_64 0:5.7.17-1.el7
35. mysql-community-devel.x86\_64 0:5.7.17-1.el7 mysql-community-embedded.x86\_64 0:5.7.17-1.el7
36. mysql-community-embedded-compat.x86\_64 0:5.7.17-1.el7 mysql-community-embedded-devel.x86\_64 0:5.7.17-1.el7
37. mysql-community-libs.x86\_64 0:5.7.17-1.el7 mysql-community-libs-compat.x86\_64 0:5.7.17-1.el7
38. mysql-community-minimal-debuginfo.x86\_64 0:5.7.17-1.el7 mysql-community-server.x86\_64 0:5.7.17-1.el7
39. mysql-community-test.x86\_64 0:5.7.17-1.el7
40. 作为依赖被安装:
41. perl-Data-Dumper.x86\_64 0:2.145-3.el7 perl-JSON.noarch 0:2.59-2.el7
42. 完毕！
43. [root@pxcnode66 ~]#
44. [root@pxcnode66 ~]# systemctl start mysqld //启动服务
45. [root@pxcnode66 ~]# ls /var/lib/mysql //查看数据库文件列表
46. auto.cnf client-cert.pem ibdata1 ibtmp1 mysql.sock.lock public\_key.pem sys
47. ca-key.pem client-key.pem ib\_logfile0 mysql performance\_schema server-cert.pem
48. ca.pem ib\_buffer\_pool ib\_logfile1 mysql.sock private\_key.pem server-key.pem
49. [root@pxcnode66 ~]# systemctl enable mysqld //设置服务开机运行
50. [root@pxcnode66 ~]# netstat -utnlp | grep :3306 //查看端口
51. tcp6 0 0 :::3306 :::\* LISTEN 1531/mysqld
52. [root@pxcnode66 ~]#
53. [root@pxcnode66 ~]# grep password /var/log/mysqld.log //查看初始密码
54. 2019-07-05T01:56:51.895852Z 1 [Note] A temporary password is generated for root@localhost: bB0\*uCmu:.Kj
55. [root@pxcnode66 ~]# mysql -uroot -p'bB0\*uCmu:.Kj' //初始密码登录
56. mysql: [Warning] Using a password on the command line interface can be insecure.
57. Welcome to the MySQL monitor. Commands end with ; or \g.
58. Your MySQL connection id is 3
59. Server version: 5.7.17
60. Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.
61. Oracle is a registered trademark of Oracle Corporation and/or its
62. affiliates. Other names may be trademarks of their respective
63. owners.
64. Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
65. mysql>
66. mysql> alter user root@"localhost" identified by "123qqq...A";//修改登录密码
67. Query OK, 0 rows affected (0.01 sec)
68. mysql>
69. mysql> exit //断开连接
70. Bye
71. [root@pxcnode66 ~]# mysql -uroot -p123qqq...A //新密码登录
72. mysql: [Warning] Using a password on the command line interface can be insecure.
73. Welcome to the MySQL monitor. Commands end with ; or \g.
74. Your MySQL connection id is 4
75. Server version: 5.7.17 MySQL Community Server (GPL)
76. Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.
77. Oracle is a registered trademark of Oracle Corporation and/or its
78. affiliates. Other names may be trademarks of their respective
79. owners.
80. Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
81. mysql> show databases; //查看数据库
82. +--------------------+
83. | Database |
84. +--------------------+
85. | information\_schema |
86. | mysql |
87. | performance\_schema |
88. | sys |
89. +--------------------+
90. 4 rows in set (0.00 sec)
91. Mysql>

2）修改服务主配置文件

1. [root@pxcnode66 ~]# vim /etc/my.cnf
2. [mysqld]
3. server\_id=66 //指定server\_id
4. :wq
5. [root@pxcnode66 ~]# systemctl restart mysqld //重启服务
6. [root@pxcnode66 ~]#

3）确保数据一致 （pxcnode66主机 使用mysql11主机的完全备份恢复数据确保数据一致 ）

1. [root@mysql11 ~]# rpm -ivh libev-4.15-1.el6.rf.x86\_64.rpm //安装依赖软件
2. [root@mysql11 ~]# yum -y install percona-xtrabackup-24-2.4.7-1.el7.x86\_64.rpm //安装在线热备软件
3. [root@mysql11 ~]# innobackupex --user root --password 123qqq...A --slave-info /allbak --no-timestamp //备份所有数据，并记录备份数据对应的binlog日志名
4. [root@mysql11 ~]# scp -r /allbak root@192.168.4.66:/root/ //把备份文件发送给pxcnode66主机
5. [root@pxcnode66 ~]# rpm -ivh libev-4.15-1.el6.rf.x86\_64.rpm //安装依赖软件
6. [root@pxcnode66 ~]# yum -y install percona-xtrabackup-24-2.4.13-1.el7.x86\_64.rpm //安装在线热备软件
7. [root@pxcnode66 ~]# systemctl stop mysqld //停止服务
8. [root@pxcnode66 ~]# rm -rf /var/lib/mysql/\* //清空数据库目录
9. [[root@pxcnode66](mailto:root@pxcnode66) ~]# innobackupex --apply-log /root/allbak/ //准备恢复数据
10. [[root@pxcnode66](mailto:root@pxcnode66) ~]# innobackupex --copy-back /root/allbak/ //恢复数据
11. [[root@pxcnode66](mailto:root@pxcnode66) ~]# chown -R mysql:mysql /var/lib/mysql //修改所有者
12. [[root@pxcnode66](mailto:root@pxcnode66) ~]# systemctl start mysqld //启动服务

4）指定主服务器

1. [root@pxcnode66 ~]# cat /root/allbak/xtrabackup\_info | grep master11 //查binlog日志
2. binlog\_pos = filename 'master11.000001', position '7700'
3. [root@pxcnode66 ~]# mysql -uroot -p123qqq...A //管理员登录指定主服务器信息
4. mysql> change master to
5. master\_host="192.168.4.11", //主服务器ip地址
6. master\_user="repluser", //主服务器授权用户
7. master\_password="123qqq...A", //授权密码
8. master\_log\_file="master11.000001", //binlog日志名
9. master\_log\_pos=7700; //日志偏移量
10. Query OK, 0 rows affected, 2 warnings (0.31 sec)
11. mysql> start slave ; //启动slave 程序
12. Query OK, 0 rows affected (0.09 sec)
13. mysql> exit //断开连接
14. Bye
15. [root@pxcnode66 ~]#
16. [root@pxcnode66 ~]# mysql -uroot -p123qqq...A -e "show slave status\G" | grep -i 192.168.4.11 //查看主服务器地址
17. mysql: [Warning] Using a password on the command line interface can be insecure.
18. Master\_Host: 192.168.4.11 //主服务器ip地址
19. [root@pxcnode66 ~]# mysql -uroot -p123qqq...A -e "show slave status\G" | grep -i "yes" //查看状态信息
20. mysql: [Warning] Using a password on the command line interface can be insecure.
21. Slave\_IO\_Running: Yes //IO线程正常
22. Slave\_SQL\_Running: Yes //SQL线程正常
23. [root@pxcnode66 ~]#

步骤二：配置第1台PXC服务器(192.168.4.66)

1）停止mysqld服务、卸载mysqld服务软件

1. [root@pxcnode66 ~]# systemctl stop mysqld //停止服务
2. [root@pxnode66 ~]# rpm -qa | grep -i mysql //查看安装的MySQL服务软件
3. mysql-community-server-5.7.17-1.el7.x86\_64
4. mysql-community-embedded-compat-5.7.17-1.el7.x86\_64
5. mysql-community-common-5.7.17-1.el7.x86\_64
6. mysql-community-client-5.7.17-1.el7.x86\_64
7. mysql-community-devel-5.7.17-1.el7.x86\_64
8. mysql-community-test-5.7.17-1.el7.x86\_64
9. mysql-community-libs-compat-5.7.17-1.el7.x86\_64
10. mysql-community-minimal-debuginfo-5.7.17-1.el7.x86\_64
11. perl-DBD-MySQL-4.023-6.el7.x86\_64
12. mysql-community-libs-5.7.17-1.el7.x86\_64
13. mysql-community-embedded-5.7.17-1.el7.x86\_64
14. mysql-community-embedded-devel-5.7.17-1.el7.x86\_64
15. [root@pxcnode66 ~]#
16. [root@pxcnode66 ~]# rpm -e --nodeps mysql-community-server mysql-community-embedded-compat mysql-community-common mysql-community-client mysql-community-devel \
17. > mysql-community-test mysql-community-libs-compat mysql-community-minimal-debuginfo mysql-community-libs mysql-community-embedded mysql-community-embedded-devel //卸载所有的MySQL服务软件
18. 警告：/etc/my.cnf 已另存为 /etc/my.cnf.rpmsave
19. [root@pxcnode66 ~]#

2）安装PXC软件、修改配置文件、启动mysql服务

1. [root@pxcnode66 ~]# cd PXC //进软件目录
2. [root@pxcnode66 PXC]# rpm -ivh qpress-1.1-14.11.x86\_64.rpm //安装依赖
3. 警告：qpress-1.1-14.11.x86\_64.rpm: 头V3 DSA/SHA1 Signature, 密钥 ID 6cb7b81f: NOKEY
4. 准备中... ################################# [100%]
5. 正在升级/安装...
6. 1:qpress-1.1-14.11 ################################# [100%]
7. [root@pxcnode66 PXC]#
8. [root@pxcnode66 PXC]# tar -xvf Percona-XtraDB-Cluster-5.7.25-31.35-r463-el7-x86\_64-bundle.tar //解压PXC软件包
9. Percona-XtraDB-Cluster-57-5.7.25-31.35.1.el7.x86\_64.rpm
10. Percona-XtraDB-Cluster-57-debuginfo-5.7.25-31.35.1.el7.x86\_64.rpm
11. Percona-XtraDB-Cluster-client-57-5.7.25-31.35.1.el7.x86\_64.rpm
12. Percona-XtraDB-Cluster-devel-57-5.7.25-31.35.1.el7.x86\_64.rpm
13. Percona-XtraDB-Cluster-full-57-5.7.25-31.35.1.el7.x86\_64.rpm
14. Percona-XtraDB-Cluster-garbd-57-5.7.25-31.35.1.el7.x86\_64.rpm
15. Percona-XtraDB-Cluster-server-57-5.7.25-31.35.1.el7.x86\_64.rpm
16. Percona-XtraDB-Cluster-shared-57-5.7.25-31.35.1.el7.x86\_64.rpm
17. Percona-XtraDB-Cluster-shared-compat-57-5.7.25-31.35.1.el7.x86\_64.rpm
18. Percona-XtraDB-Cluster-test-57-5.7.25-31.35.1.el7.x86\_64.rpm
19. [root@pxcnode66 PXC]#
20. [root@pxcnode66 PXC]# yum -y install Percona-XtraDB-Cluster-\*.rpm //安装软件
21. 已安装:
22. Percona-XtraDB-Cluster-57.x86\_64 0:5.7.25-31.35.1.el7 Percona-XtraDB-Cluster-57-debuginfo.x86\_64 0:5.7.25-31.35.1.el7
23. Percona-XtraDB-Cluster-client-57.x86\_64 0:5.7.25-31.35.1.el7 Percona-XtraDB-Cluster-devel-57.x86\_64 0:5.7.25-31.35.1.el7
24. Percona-XtraDB-Cluster-full-57.x86\_64 0:5.7.25-31.35.1.el7 Percona-XtraDB-Cluster-garbd-57.x86\_64 0:5.7.25-31.35.1.el7
25. Percona-XtraDB-Cluster-server-57.x86\_64 0:5.7.25-31.35.1.el7 Percona-XtraDB-Cluster-shared-57.x86\_64 0:5.7.25-31.35.1.el7
26. Percona-XtraDB-Cluster-shared-compat-57.x86\_64 0:5.7.25-31.35.1.el7 Percona-XtraDB-Cluster-test-57.x86\_64 0:5.7.25-31.35.1.el7
27. 作为依赖被安装:
28. keyutils-libs-devel.x86\_64 0:1.5.8-3.el7 krb5-devel.x86\_64 0:1.15.1-18.el7 libcom\_err-devel.x86\_64 0:1.42.9-11.el7 libkadm5.x86\_64 0:1.15.1-18.el7
29. libselinux-devel.x86\_64 0:2.5-12.el7 libsepol-devel.x86\_64 0:2.5-8.1.el7 libverto-devel.x86\_64 0:0.2.5-4.el7 openssl-devel.x86\_64 1:1.0.2k-12.el7
30. pcre-devel.x86\_64 0:8.32-17.el7 perl-Env.noarch 0:1.04-2.el7 perl-Test-Harness.noarch 0:3.28-3.el7 perl-Test-Simple.noarch 0:0.98-243.el7
31. zlib-devel.x86\_64 0:1.2.7-17.el7
32. 完毕！
33. [root@pxcnode66 PXC]#
34. [root@pxcnode66 PXC]# vim /etc/percona-xtradb-cluster.conf.d/mysqld.cnf //修改数据库服务配置文件
35. [mysqld]
36. server-id=66 //指定server\_id
37. :wq
38. [root@pxcnode66 PXC]#
39. [root@pxcnode66 PXC]# vim /etc/percona-xtradb-cluster.conf.d/wsrep.cnf //修改集群服务配置文件
40. wsrep\_cluster\_address=gcomm://     不需要写ip地址
41. wsrep\_node\_address=192.168.4.66 //指定本机Ip地址
42. wsrep\_cluster\_name=pxc-cluster //指定集群名称（另外2台的集群名称要于此相同）
43. wsrep\_node\_name=pxcnode66 //指定本机主机名
44. wsrep\_sst\_auth="sstuser:123qqq...A" //数据全量同步授权用户及密码
45. :wq
46. [root@pxcnode66 PXC]#
47. [root@pxcnode66 PXC]# systemctl start mysql //启动服务
48. [root@pxcnode66 PXC]# netstat -utnlp | grep :3306 //查看MySQL服务端口
49. tcp6 0 0 :::3306 :::\* LISTEN 24482/mysqld
50. [root@pxcnode66 PXC]# netstat -utnlp | grep :4567 //查看集群通信端口
51. tcp 0 0 0.0.0.0:4567 0.0.0.0:\* LISTEN 24472/mysqld
52. [root@pxcnode66 PXC]# systemctl enable mysql //设置服务开机运行
53. [root@pxcnode66 PXC]#

3）数据库管理员登录、用户授权、查看状态信息

1. [root@pxcnode66 PXC]# mysql -uroot -p123qqq...A //管理员登录
2. mysql> grant all on \*.\* to sstuser@"localhost" identified by "123qqq...A"; //用户授权
3. Query OK, 0 rows affected, 1 warning (0.10 sec)
4. mysql> show status like "%wsrep%"; //查看集群状态信息
5. | wsrep\_incoming\_addresses | 192.168.4.66:3306 |
6. | wsrep\_cluster\_weight | 1 |
7. | wsrep\_desync\_count | 0 |
8. | wsrep\_evs\_delayed | |
9. | wsrep\_evs\_evict\_list | |
10. | wsrep\_evs\_repl\_latency | 0/0/0/0/0 |
11. | wsrep\_evs\_state | OPERATIONAL |
12. | wsrep\_gcomm\_uuid | 73809cc5-cf00-11e9-aac3-b223959fecdf |
13. | wsrep\_cluster\_conf\_id | 1 |
14. | wsrep\_cluster\_size | 1 |
15. | wsrep\_cluster\_state\_uuid | 73848b1a-cf00-11e9-9058-36c1ac1e1359 |
16. | wsrep\_cluster\_status | Primary |
17. | wsrep\_connected | ON |
18. | wsrep\_local\_bf\_aborts | 0 |
19. | wsrep\_local\_index | 0 |
20. | wsrep\_provider\_name | Galera |
21. | wsrep\_provider\_vendor | Codership Oy <info@codership.com> |
22. | wsrep\_provider\_version | 3.35(rddf9876) |
23. | wsrep\_ready | ON |
24. +----------------------------------+--------------------------------------+
25. 71 rows in set (0.00 sec)
26. mysql> exit ;
27. [root@pxcnode66 ~]#
28. [root@pxcnode66 ~]# mysql -uroot -p123qqq...A -e "show slave status\G" | grep -i "yes" //查看状态信息依然是192.168.4.11的从服务器
29. mysql: [Warning] Using a password on the command line interface can be insecure.
30. Slave\_IO\_Running: Yes //IO线程正常
31. Slave\_SQL\_Running: Yes //SQL线程正常
32. [root@pxcnode66 ~]#

步骤三：配置第2台PXC服务器(192.168.4.10)

1）安装PXC软件

1. [root@pxcnode10 ~]# cd PXC //进软件目录
2. [root@pxcnode10 PXC]# rpm -ivh qpress-1.1-14.11.x86\_64.rpm //安装依赖
3. 警告：qpress-1.1-14.11.x86\_64.rpm: 头V3 DSA/SHA1 Signature, 密钥 ID 6cb7b81f: NOKEY
4. 准备中... ################################# [100%]
5. 正在升级/安装...
6. 1:qpress-1.1-14.11 ################################# [100%]
7. [root@pxcnode10 PXC]#
8. [root@pxcnode10 PXC]# tar -xvf Percona-XtraDB-Cluster-5.7.25-31.35-r463-el7-x86\_64-bundle.tar //解压PXC软件包
9. Percona-XtraDB-Cluster-57-5.7.25-31.35.1.el7.x86\_64.rpm
10. Percona-XtraDB-Cluster-57-debuginfo-5.7.25-31.35.1.el7.x86\_64.rpm
11. Percona-XtraDB-Cluster-client-57-5.7.25-31.35.1.el7.x86\_64.rpm
12. Percona-XtraDB-Cluster-devel-57-5.7.25-31.35.1.el7.x86\_64.rpm
13. Percona-XtraDB-Cluster-full-57-5.7.25-31.35.1.el7.x86\_64.rpm
14. Percona-XtraDB-Cluster-garbd-57-5.7.25-31.35.1.el7.x86\_64.rpm
15. Percona-XtraDB-Cluster-server-57-5.7.25-31.35.1.el7.x86\_64.rpm
16. Percona-XtraDB-Cluster-shared-57-5.7.25-31.35.1.el7.x86\_64.rpm
17. Percona-XtraDB-Cluster-shared-compat-57-5.7.25-31.35.1.el7.x86\_64.rpm
18. Percona-XtraDB-Cluster-test-57-5.7.25-31.35.1.el7.x86\_64.rpm
19. [root@pxcnode10 PXC]#
20. [root@pxcnode10 PXC]# yum -y install Percona-XtraDB-Cluster-\*.rpm //安装软件
21. 已安装:
22. Percona-XtraDB-Cluster-57.x86\_64 0:5.7.25-31.35.1.el7 Percona-XtraDB-Cluster-57-debuginfo.x86\_64 0:5.7.25-31.35.1.el7
23. Percona-XtraDB-Cluster-client-57.x86\_64 0:5.7.25-31.35.1.el7 Percona-XtraDB-Cluster-devel-57.x86\_64 0:5.7.25-31.35.1.el7
24. Percona-XtraDB-Cluster-full-57.x86\_64 0:5.7.25-31.35.1.el7 Percona-XtraDB-Cluster-garbd-57.x86\_64 0:5.7.25-31.35.1.el7
25. Percona-XtraDB-Cluster-server-57.x86\_64 0:5.7.25-31.35.1.el7 Percona-XtraDB-Cluster-shared-57.x86\_64 0:5.7.25-31.35.1.el7
26. Percona-XtraDB-Cluster-shared-compat-57.x86\_64 0:5.7.25-31.35.1.el7 Percona-XtraDB-Cluster-test-57.x86\_64 0:5.7.25-31.35.1.el7
27. 作为依赖被安装:
28. keyutils-libs-devel.x86\_64 0:1.5.8-3.el7 krb5-devel.x86\_64 0:1.15.1-18.el7 libcom\_err-devel.x86\_64 0:1.42.9-11.el7 libkadm5.x86\_64 0:1.15.1-18.el7
29. libselinux-devel.x86\_64 0:2.5-12.el7 libsepol-devel.x86\_64 0:2.5-8.1.el7 libverto-devel.x86\_64 0:0.2.5-4.el7 openssl-devel.x86\_64 1:1.0.2k-12.el7
30. pcre-devel.x86\_64 0:8.32-17.el7 perl-Env.noarch 0:1.04-2.el7 perl-Test-Harness.noarch 0:3.28-3.el7 perl-Test-Simple.noarch 0:0.98-243.el7
31. zlib-devel.x86\_64 0:1.2.7-17.el7
32. 完毕！
33. [root@pxcnode10 PXC]#

2）修改配置文件

1. [root@pxcnode10 PXC]# vim /etc/percona-xtradb-cluster.conf.d/mysqld.cnf //修改数据库服务配置
2. [mysqld]
3. server-id=10 //指定server\_id
4. :wq
5. [root@pxcnode10 PXC]#
6. [root@pxcnode10 PXC]# vim /etc/percona-xtradb-cluster.conf.d/wsrep.cnf //修改集群服务配置文件
7. wsrep\_cluster\_address=gcomm://192.168.4.66,192.168.4.10     //集群成员列表
8. wsrep\_node\_address=192.168.4.10 //指定本机Ip地址
9. wsrep\_cluster\_name=pxc-cluster //指定集群名称（另外2台的集群名称要于此相同）
10. wsrep\_node\_name=pxcnode10 //指定本机主机名
11. wsrep\_sst\_auth="sstuser:123qqq...A" //数据全量同步授权用户及密码
12. :wq
13. [root@pxcnode10 PXC]#

3）启动mysql服务

1. [root@pxcnode10 PXC]# systemctl start mysql //启动服务
2. [root@pxcnode10 PXC]# systemctl enable mysql //服务开机运行
3. [root@pxcnode10 PXC]# netstat -utnlp | grep :3306 //查看MySQL服务端口
4. tcp6 0 0 :::3306 :::\* LISTEN 24482/mysqld
5. [root@pxcnode10 PXC]# netstat -utnlp | grep :4567 //查看集群端口
6. tcp6 0 0 :::4567 :::\* LISTEN 24489/mysqld
7. [root@pxcnode10 PXC]#
8. mysql> show status like "%wsrep%"; //查看集群状态信息
9. | wsrep\_incoming\_addresses | 192.168.4.66:3306，192.168.4.10:3306|
10. | wsrep\_cluster\_weight | 1 |
11. | wsrep\_desync\_count | 0 |
12. | wsrep\_evs\_delayed | |
13. | wsrep\_evs\_evict\_list | |
14. | wsrep\_evs\_repl\_latency | 0/0/0/0/0 |
15. | wsrep\_evs\_state | OPERATIONAL |
16. | wsrep\_gcomm\_uuid | 73809cc5-cf00-11e9-aac3-b223959fecdf |
17. | wsrep\_cluster\_conf\_id | 1 |
18. | wsrep\_cluster\_size | 1 |
19. | wsrep\_cluster\_state\_uuid | 73848b1a-cf00-11e9-9058-36c1ac1e1359 |
20. | wsrep\_cluster\_status | Primary |
21. | wsrep\_connected | ON |
22. | wsrep\_local\_bf\_aborts | 0 |
23. | wsrep\_local\_index | 0 |
24. | wsrep\_provider\_name | Galera |
25. | wsrep\_provider\_vendor | Codership Oy <info@codership.com> |
26. | wsrep\_provider\_version | 3.35(rddf9876) |
27. | wsrep\_ready | ON |
28. +----------------------------------+--------------------------------------+
29. 71 rows in set (0.00 sec)
30. mysql> exit ;
31. [root@pxcnode10 ~]#

步骤四：配置第3台PXC服务器(192.168.4.88)

1）安装PXC软件

1. [root@pxcnode88 ~]# cd PXC //进软件目录
2. [root@pxcnode88 PXC]# rpm -ivh qpress-1.1-14.11.x86\_64.rpm //安装依赖
3. 警告：qpress-1.1-14.11.x86\_64.rpm: 头V3 DSA/SHA1 Signature, 密钥 ID 6cb7b81f: NOKEY
4. 准备中... ################################# [100%]
5. 正在升级/安装...
6. 1:qpress-1.1-14.11 ################################# [100%]
7. [root@pxcnode88 PXC]#
8. [root@pxcnode88 PXC]# tar -xvf Percona-XtraDB-Cluster-5.7.25-31.35-r463-el7-x86\_64-bundle.tar //解压PXC软件包
9. Percona-XtraDB-Cluster-57-5.7.25-31.35.1.el7.x86\_64.rpm
10. Percona-XtraDB-Cluster-57-debuginfo-5.7.25-31.35.1.el7.x86\_64.rpm
11. Percona-XtraDB-Cluster-client-57-5.7.25-31.35.1.el7.x86\_64.rpm
12. Percona-XtraDB-Cluster-devel-57-5.7.25-31.35.1.el7.x86\_64.rpm
13. Percona-XtraDB-Cluster-full-57-5.7.25-31.35.1.el7.x86\_64.rpm
14. Percona-XtraDB-Cluster-garbd-57-5.7.25-31.35.1.el7.x86\_64.rpm
15. Percona-XtraDB-Cluster-server-57-5.7.25-31.35.1.el7.x86\_64.rpm
16. Percona-XtraDB-Cluster-shared-57-5.7.25-31.35.1.el7.x86\_64.rpm
17. Percona-XtraDB-Cluster-shared-compat-57-5.7.25-31.35.1.el7.x86\_64.rpm
18. Percona-XtraDB-Cluster-test-57-5.7.25-31.35.1.el7.x86\_64.rpm
19. [root@pxcnode88 PXC]#
20. [root@pxcnode88 PXC]# yum -y install Percona-XtraDB-Cluster-\*.rpm //安装软件
21. 已安装:
22. Percona-XtraDB-Cluster-57.x86\_64 0:5.7.25-31.35.1.el7 Percona-XtraDB-Cluster-57-debuginfo.x86\_64 0:5.7.25-31.35.1.el7
23. Percona-XtraDB-Cluster-client-57.x86\_64 0:5.7.25-31.35.1.el7 Percona-XtraDB-Cluster-devel-57.x86\_64 0:5.7.25-31.35.1.el7
24. Percona-XtraDB-Cluster-full-57.x86\_64 0:5.7.25-31.35.1.el7 Percona-XtraDB-Cluster-garbd-57.x86\_64 0:5.7.25-31.35.1.el7
25. Percona-XtraDB-Cluster-server-57.x86\_64 0:5.7.25-31.35.1.el7 Percona-XtraDB-Cluster-shared-57.x86\_64 0:5.7.25-31.35.1.el7
26. Percona-XtraDB-Cluster-shared-compat-57.x86\_64 0:5.7.25-31.35.1.el7 Percona-XtraDB-Cluster-test-57.x86\_64 0:5.7.25-31.35.1.el7
27. 作为依赖被安装:
28. keyutils-libs-devel.x86\_64 0:1.5.8-3.el7 krb5-devel.x86\_64 0:1.15.1-18.el7 libcom\_err-devel.x86\_64 0:1.42.9-11.el7 libkadm5.x86\_64 0:1.15.1-18.el7
29. libselinux-devel.x86\_64 0:2.5-12.el7 libsepol-devel.x86\_64 0:2.5-8.1.el7 libverto-devel.x86\_64 0:0.2.5-4.el7 openssl-devel.x86\_64 1:1.0.2k-12.el7
30. pcre-devel.x86\_64 0:8.32-17.el7 perl-Env.noarch 0:1.04-2.el7 perl-Test-Harness.noarch 0:3.28-3.el7 perl-Test-Simple.noarch 0:0.98-243.el7
31. zlib-devel.x86\_64 0:1.2.7-17.el7
32. 完毕！
33. [root@pxcnode88 PXC]#

2）修改配置文件

1. [root@pxcnode88 PXC]# vim /etc/percona-xtradb-cluster.conf.d/mysqld.cnf //修改数据库服务配置
2. [mysqld]
3. server-id=88 //指定server\_id
4. :wq
5. [root@pxcnode88 PXC]#
6. [root@pxcnode88 PXC]# vim /etc/percona-xtradb-cluster.conf.d/wsrep.cnf //修改集群服务配置文件
7. wsrep\_cluster\_address=gcomm://192.168.4.66     //集群成员ip地址
8. wsrep\_node\_address=192.168.4.88 //指定本机Ip地址
9. wsrep\_cluster\_name=pxc-cluster //指定集群名称（另外2台的集群名称要于此相同）
10. wsrep\_node\_name=pxcnode88 //指定本机主机名
11. wsrep\_sst\_auth="sstuser:123qqq...A" //数据全量同步授权用户及密码
12. :wq
13. [root@pxcnode88 PXC]#

3）启动mysql服务

1. [root@pxcnode88 PXC]# systemctl start mysql //启动服务
2. [root@pxcnode88 PXC]# systemctl enable mysql //服务开机运行
3. [root@pxcnode88 PXC]# netstat -utnlp | grep :3306 //查看MySQL服务端口
4. tcp6 0 0 :::3306 :::\* LISTEN 24472/mysqld
5. [root@pxcnode88 PXC]# netstat -utnlp | grep :4567 //查看集群端口
6. tcp6 0 0 :::4567 :::\* LISTEN 24486/mysqld
7. [root@pxcnode88 PXC]#
8. #

步骤五：公共配置(192.168.4.88、192.168.4.10、192.168.4.66)

1）修改192.168.4.88主机的集群配置文件

1. [root@pxcnode88 ~]# vim /etc/percona-xtradb-cluster.conf.d/wsrep.cnf
2. wsrep\_cluster\_address=gcomm://192.168.4.66,192.168.4.10,192.168.4.88 //指定集群成员列表
3. :wq
4. [root@pxcnode88 ~]#

2）修改192.168.4.10主机的集群配置文件

1. [root@pxcnode10 ~]# vim /etc/percona-xtradb-cluster.conf.d/wsrep.cnf
2. wsrep\_cluster\_address=gcomm://192.168.4.66,192.168.4.88,192.168.4.10 //指定集群成员列表
3. :wq
4. [root@pxcnode10 ~]#

3）修改192.168.4.66主机的集群配置文件

1. [root@pxcnode66 ~]# vim /etc/percona-xtradb-cluster.conf.d/wsrep.cnf
2. wsrep\_cluster\_address=gcomm://192.168.4.66,192.168.4.88,192.168.4.10 //指定集群成员列表
3. :wq
4. [root@pxcnode66 ~]#

步骤6：测试配置:在网站服务器连接PXC集群主机存取数据：

1）存储数据：在网站服务器连接PXC集群主机存储数据

1. [root@web33 ~]# mysql -h192.168.4.66 -uyaya99 -p123qqq…A gamedb
2. Mysql> insert into gamedb.user values (“pljA”);
3. Mysql> exit ;
4. [root@web33 ~]# mysql -h192.168.4.10 -uyaya99 -p123qqq…A gamedb
5. Mysql> insert into gamedb.user values (“pljB”);
6. Mysql> exit ;
7. [root@web33 ~]# mysql -h192.168.4.88 -uyaya99 -p123qqq…A gamedb
8. Mysql> insert into gamedb.user values (“pljC”);
9. Mysql> exit ;

2）查询数据：在网站服务器连接PXC集群主机查询数据

1. [root@web44 ~]# mysql -h192.168.4.66 -uyaya99 -p123qqq…A gamedb
2. Mysql> select \* from gamedb.user;
3. +------+
4. | name|
5. +------+
6. | pljA|
7. | pljB|
8. | pljC|
9. +------+
10. Mysql> exit ;
11. [root@web44 ~]# mysql -h192.168.4.10 -uyaya99 -p123qqq…A gamedb
12. Mysql> select \* from gamedb.user;
13. +------+
14. | name|
15. +------+
16. | pljA|
17. | pljB|
18. | pljC|
19. +------+
20. Mysql> exit ;Mysql> exit ;
21. [root@web44 ~]# mysql -h192.168.4.88 -uyaya99 -p123qqq…A gamedb
22. Mysql> select \* from gamedb.user;
23. +------+
24. | name|
25. +------+
26. | pljA|
27. | pljB|
28. | pljC|
29. +------+
30. Mysql> exit ;Mysql> exit ;

## 4 案例4：部署LB集群

### 4.1 问题

配置步骤如下：

1. 安装软件
2. 修改配置文件
3. 启动服务
4. 测试配置

### 4.2 方案

拓扑结构如图-3所示。创建1台新的虚拟机，配置ip地址 eth0 192.168.4.99 主机名 haproxy99 ；运行haproxy服务 接受客户端访问数据库的连接请求，把请求平均分发给3台PXC集群主机。



图-3

### 4.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：安装软件: 在haproxy99主机上安装haproxy软件

1. [root@haproxy99 ~]# yum -y install haproxy
2. ……
3. Running transaction
4. 正在安装 : haproxy-1.5.18-7.el7.x86\_64 1/1
5. 验证中 : haproxy-1.5.18-7.el7.x86\_64 1/1
6. 已安装:
7. haproxy.x86\_64 0:1.5.18-7.el7
8. 完毕！
9. [root@haproxy99 ~]#

步骤二：修改配置文件

1. [root@haproxy99 ~]# vim /etc/haproxy/haproxy.cfg
2. Global //全局配置默认即可
3. log 127.0.0.1 local2
4. chroot /var/lib/haproxy
5. pidfile /var/run/haproxy.pid
6. maxconn 4000
7. user haproxy
8. group haproxy
9. daemon
10. stats socket /var/lib/haproxy/stats
11. defaults //默认配置(不需要修改)
12. mode http
13. log global
14. option httplog
15. option dontlognull
16. option http-server-close
17. option forwardfor except 127.0.0.0/8
18. option redispatch
19. retries 3
20. timeout http-request 10s
21. timeout queue 1m
22. timeout connect 10s
23. timeout client 1m
24. timeout server 1m
25. timeout http-keep-alive 10s
26. timeout check 10s
27. maxconn 3000
28. listen status //定义监控页面
29. mode http //模式为http
30. bind \*:80 //端口80
31. stats enable //启用配置
32. stats uri /admin //访问目录名
33. stats auth admin:admin //登录用户与密码
34. listen mysql\_3306 \*:3306 //定义haproxy服务名称与端口号
35. mode tcp //mysql服务 得使用 tcp 协议
36. option tcpka //使用长连接
37. balance roundrobin //调度算法
38. server mysql\_01 192.168.4.66:3306 check //第1台数据库服务器
39. server mysql\_02 192.168.4.10:3306 check //第2台数据库服务器
40. server mysql\_03 192.168.4.88:3306 check //第3台数据库服务器
41. :wq
42. [root@haproxy99 haproxy]#

步骤三：启动服务

1. [root@haproxy99 ~]# systemctl start haproxy //启动服务
2. [root@haproxy99 ~]# systemctl enable haproxy //开机运行
3. Created symlink from /etc/systemd/system/multi-user.target.wants/haproxy.service to /usr/lib/systemd/system/haproxy.service.
4. [root@haproxy99 ~]# netstat -utnlp | grep :3306 //查看端口
5. tcp6 0 0 :::3306 :::\* LISTEN 29768/haproxy
6. [root@haproxy99 ~]#

步骤四：测试配置：在网站服务器连接haproxy99主机访问数据

1. [root@web33 ~]# mysql –h192.168.4.99 –uyaya99 –p123qqq…A -e 'select @@hostname'
2. mysql: [Warning] Using a password on the command line interface can be insecure.
3. +------------+
4. | @@hostname |
5. +------------+
6. | pxcnode66 | //第1次连接
7. +------------+
8. [root@web33 ~]#
9. [root@web33 ~]# mysql –h192.168.4.99 –uyaya99 –p123qqq…A -e 'select @@hostname'
10. mysql: [Warning] Using a password on the command line interface can be insecure.
11. +------------+
12. | @@hostname |
13. +------------+
14. | pxcnode10 | //第2次连接
15. +------------+
16. [root@web33 ~]#
17. [root@web33 ~]# mysql –h192.168.4.99 –uyaya99 –p123qqq…A -e 'select @@hostname'
18. mysql: [Warning] Using a password on the command line interface can be insecure.
19. +------------+
20. | @@hostname |
21. +------------+
22. | pxcnode88 | //第3次连接
23. +------------+
24. [root@web33 ~]#

## 5 案例5：部署HA集群

### 5.1 问题

具体配置如下：

1. 准备备用调度器主机
2. 安装软件
3. 修改配置文件
4. 启动服务
5. 测试配置

### 5.2 方案

拓扑结构如图-4所示。创建1台新的虚拟机，在eth0 接口配置ip地址为192.168.4.98做备用调度器。

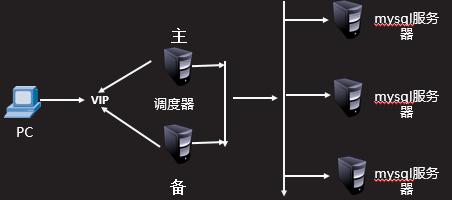


图-4

### 5.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：准备备用调度器主机

1）克隆好虚拟机后配置ip地址、设置主机名

1. [root@localhost ~]# setip
2. Network name(eth0/eth1/eth2/eth3):eth0
3. Set IP(IP/24):192.168.4.98/24
4. Set Gateway(default none):
5. 192.168.4.98/24
6. 连接已成功激活（D-Bus 活动路径：/org/freedesktop/NetworkManager/ActiveConnection/7）
7. [root@localhost ~]# hostname haproxy98
8. [root@localhost ~]# ifconfig eth0 | head -2
9. Eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
10. inet 192.168.4.98 netmask 255.255.255.0 broadcast 192.168.2.255 [root@localhost ~]#
11. [root@haproxy98 ~]# yum -y install haproxy
12. 正在安装 : haproxy-1.5.18-7.el7.x86\_64 1/1
13. 验证中 : haproxy-1.5.18-7.el7.x86\_64 1/1
14. 已安装:
15. haproxy.x86\_64 0:1.5.18-7.el7
16. 完毕！
17. [root@haproxy98 ~]#

3）修改haproxy98主机haproxy.conf文件（直接拷贝haproxy99主机的配置文件也可以）

1. [root@haproxy98 ~]# scp root@192.168.4.99:/etc/haproxy/haproxy.cfg /etc/haproxy/
2. Warning: Permanently added '192.168.4.99' (ECDSA) to the list of known hosts.
3. root@192.168.4.99's password: //输入haproxy99主机的密码
4. haproxy.cfg 100% 3142 6.0MB/s 00:00
5. [root@haproxy98 ~]#

4）启动haproxy服务

1. [root@haproxy98 ~]# systemctl start haproxy //启动服务
2. [root@haproxy98 ~]# systemctl enable haproxy //服务开机运行
3. Created symlink from /etc/systemd/system/multi-user.target.wants/haproxy.service to /usr/lib/systemd/system/haproxy.service.
4. [root@haproxy98 ~]# netstat -utnlp | grep :3306 //查看端口
5. tcp6 0 0 :::3306 :::\* LISTEN 29768/haproxy

步骤二：安装软件

1）在haproxy99主机安装keepalived软件

1. [root@haproxy99 ~]# yum -y install keepalived.x86\_64
2. 已安装:
3. keepalived.x86\_64 0:1.3.5-6.el7
4. 作为依赖被安装:
5. lm\_sensors-libs.x86\_64 0:3.4.0-4.20160601gitf9185e5.el7 net-snmp-agent-libs.x86\_64 1:5.7.2-32.el7
6. net-snmp-libs.x86\_64 1:5.7.2-32.el7
7. [root@haproxy99 ~]#

2）在haproxy98主机安装keepalived软件

1. [root@haproxy98 ~]# yum -y install keepalived.x86\_64
2. 已安装:
3. keepalived.x86\_64 0:1.3.5-6.el7
4. 作为依赖被安装:
5. lm\_sensors-libs.x86\_64 0:3.4.0-4.20160601gitf9185e5.el7 net-snmp-agent-libs.x86\_64 1:5.7.2-32.el7
6. net-snmp-libs.x86\_64 1:5.7.2-32.el7
7. [root@haproxy98 ~]#
8. 完毕！
9. [root@haproxy98 ~]#

步骤三：修改配置文件

1）修改haproxy99主机的配置文件

1. [root@haproxy99 ~]# sed -i '36,$d' /etc/keepalived/keepalived.conf //删除无关的配置行
2. [root@haproxy99 ~]#vim /etc/keepalived/keepalived.conf
3. global\_defs {
4. ……
5. ……
6. vrrp\_iptables //禁止iptables
7. }
8. vrrp\_instance VI\_1 {
9. state MASTER //主服务器标识
10. interface eth0
11. virtual\_router\_id 51
12. priority 150 //haproxy99 主机做主服务器，优先级要比 haproxy88主机高
13. advert\_int 1
14. authentication {
15. auth\_type PASS //主备服务器连接方式
16. auth\_pass 1111 //连接密码
17. }
18. virtual\_ipaddress {
19. 192.168.4.100 //定义vip地址
20. }
21. }
22. [root@haproxy99 ~]# scp /etc/keepalived/keepalived.conf root@192.168.4.98: /etc/keepalived/
23. root@192.168.4.98's password: //输入haproxy98主机的密码

2）修改haproxy98主机的配置文件

1. [root@haproxy98 ~]#vim /etc/keepalived/keepalived.conf
2. global\_defs {
3. ……
4. ……
5. vrrp\_iptables //禁止iptables
6. }
7. vrrp\_instance VI\_1 {
8. state BACKUP //备用服务器标识
9. interface eth0
10. virtual\_router\_id 51
11. priority 100 //优先级要比 haproxy99低
12. advert\_int 1
13. authentication {
14. auth\_type PASS
15. auth\_pass 1111
16. }
17. virtual\_ipaddress {
18. 192.168.4.100 //定义vip地址
19. }
20. }
21. [root@haproxy98 ~]#

步骤四：启动服务

1）在haproxy99主机启动keepalived服务

1. [root@haproxy99 ~]# systemctl start keepalived.service //启动服务
2. [root@haproxy99 ~]#
3. [root@haproxy99 ~]# ip addr show | grep 192.168.4.100 //查看vip地址
4. inet 192.168.4.100/32 scope global eth0
5. [root@haproxy99 ~]#

2）在haproxy98主机启动keepalived服务

1. [root@haproxy98 ~]# systemctl start keepalived.service //启动服务
2. [root@haproxy98 ~]#
3. [root@haproxy98 ~]# ip addr show | grep 192.168.4.100 //查看不到vip
4. [root@haproxy98 ~]#

步骤五：测试配置

1）客户端连接vip地址，访问数据库服务

1. [root@web33 ~]# mysql -h192.168.4.100 –uyaya99 –p123qqq…A -e 'select @@hostname'
2. mysql: [Warning] Using a password on the command line interface can be insecure.
3. +------------+
4. | @@hostname |
5. +------------+
6. | pxcnode66 |
7. +------------+
8. [root@web33 ~]#
9. [root@web33 ~]# mysql -h192.168.4.100 –uyaya99 –p123qqq…A -e 'select @@hostname'
10. mysql: [Warning] Using a password on the command line interface can be insecure.
11. +------------+
12. | @@hostname |
13. +------------+
14. | pxcnode10 |
15. +------------+
16. [root@web33 ~]# mysql -h192.168.4.100 –uyaya99 –p123qqq…A -e 'select @@hostname'
17. mysql: [Warning] Using a password on the command line interface can be insecure.
18. +------------+
19. | @@hostname |
20. +------------+
21. | pxcnode88 |
22. +------------+
23. [root@web33 ~]#

2）测试高可用

1. [root@haproxy99 ~]# ip addr show | grep 192.168.4.100 //在haproxy99 主机查看VIP地址
2. inet 192.168.4.100/32 scope global eth0
3. [root@haproxy99 ~]#
4. [root@haproxy99 ~]# systemctl stop keepalived.service //停止keepalived服务
5. [root@haproxy99 ~]#
6. [root@haproxy99 ~]#
7. [root@haproxy99 ~]# ip addr show | grep 192.168.4.100 //查看不到vip地址
8. [root@haproxy99 ~]#
9. [root@haproxy98 ~]# ip addr show | grep 192.168.4.100 //在备用的haproxy98主机查看地址
10. inet 192.168.4.100/32 scope global eth0
11. [root@haproxy98 ~]#
12. //客户端连接vip地址访问数据库服务
13. [root@web33 ~]# mysql -h192.168.4.100 –uyaya99 –p123qqq…A -e 'select @@hostname'
14. mysql: [Warning] Using a password on the command line interface can be insecure.
15. +------------+
16. | @@hostname |
17. +------------+
18. | pxcnode66 |
19. +------------+
20. [root@web33 ~]#
21. [root@web33 ~]# mysql -h192.168.4.100 –uyaya99 –p123qqq…A -e 'select @@hostname'
22. mysql: [Warning] Using a password on the command line interface can be insecure.
23. +------------+
24. | @@hostname |
25. +------------+
26. | pxcnode10 |
27. +------------+
28. [root@web33 ~]# mysql -h192.168.4.100 –uyaya99 –p123qqq…A -e 'select @@hostname'
29. mysql: [Warning] Using a password on the command line interface can be insecure.
30. +------------+
31. | @@hostname |
32. +------------+
33. | pxcnode88 |
34. +------------+
35. [root@web33 ~]#