

环境说明

按照之前的步骤安装2台服务器:

Redis01	172.16.100.10
Redis02	172.16.100.20

开启防火墙端口和关闭seLinux

打开6379端口

```
[root@Redis01 bin]# cat /etc/sysconfig/iptables
# Firewall configuration written by system-config-firewall
# Manual customization of this file is not recommended.
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 6379 -j ACCEPT
-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A FORWARD -j REJECT --reject-with icmp-host-prohibited
COMMIT
[root@Redis01 bin]#
```



关闭selinux

```
[root@Redis01 bin]# cat /etc/selinux/config
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#   enforcing - SELinux security policy is enforced.
#   permissive - SELinux prints warnings instead of enforcing.
#   disabled - No SELinux policy is loaded.
SELINUX=disabled
# SELINUXTYPE= can take one of these two values:
#   targeted - Targeted processes are protected.
#   mls - Multi Level Security protection.
SELINUXTYPE=targeted
```



然后重启iptables服务。

修改配置文件

修改主服务器配置文件，如果不需要密码验证则无需修改此文件，修改后请重新启动Redis服务

```
***** SECURITY *****
# Require clients to issue AUTH <PASSWORD> before processing any other
# commands. This might be useful in environments in which you do not trust
```

```
# Commands. This might be useful in environments in which you do not trust
# others with access to the host running redis-server.
#
# This should stay commented out for backward compatibility and because most
# people do not need auth (e.g. they run their own servers).
#
# Warning: since Redis is pretty fast an outside user can try up to
# 150k passwords per second against a good box. This means that you should
# use a very strong password otherwise it will be very easy to break.
#
requirepass master123
#
# Command renaming.
#
```

修改从服务器配置文件，

找到slaveof字段，如果主服务器有密码验证，则要配置相同的密码，如下如：

```
# master if the replication link is lost for a relatively small amount of
# time. You may want to configure the replication backlog size (see the next
# sections of this file) with a sensible value depending on your needs.
# 3) Replication is automatic and does not need user intervention. After a
# network partition slaves automatically try to reconnect to masters
# and resynchronize with them.
#
slaveof 172.16.100.10 6379
#
# If the master is password protected (using the "requirepass" configuration
# directive below) it is possible to tell the slave to authenticate before
# starting the replication synchronization process, otherwise the master will
# refuse the slave request.
#
masterauth master123
#
# When a slave loses its connection with the master, or when the replication
# is still in progress, the slave can act in two different ways:
```

验证主从复制

在主上设置一个键值数据

```
[root@Redis01 bin]#
[root@Redis01 bin]# ./redis-cli -a master123
127.0.0.1:6379> set name "Hello world!!!"
OK
127.0.0.1:6379>
```

在从上获取一个键值数据

```
[root@Redis02 bin]#
[root@Redis02 bin]# ./redis-cli
127.0.0.1:6379> get name
"Hello world!!!"
127.0.0.1:6379>
```

通过info查看连接信息，在主上执行下面的命令：

```
127.0.0.1:6379>
127.0.0.1:6379>
127.0.0.1:6379> info
# Replication
role:master
connected_slaves:1
slave0:ip=172.16.100.20,port=6379,state=online,offset=646,lag=1
master_repl_offset:646
repl_backlog_active:1
repl_backlog_size:1048576
```

```
repl_backlog_size:1048576
repl_backlog_first_byte_offset:2
repl_backlog_histlen:645
```

不停机切换

Redis的主从模式中的主与从没有什么区别，唯一的区别就是在这个模式下从服务器是不允许被写入数据的，看下图：

我在从服务器上做一个SET操作，提示如下

```
127.0.0.1:6379> set A3 3
(error) READONLY You can't write against a read only slave.
```

其实主从切换就是把从变成主，主变成从。基本步骤如下：

设置从服务器可写：

修改配置文件，如下图：

```
# Since Redis 2.6 by default slaves are read-only.
#
# Note: read only slaves are not designed to be exposed to untrusted clients
# on the internet. It's just a protection layer against misuse of the i
# Still a read only slave exports by default all the administrative com
# such as CONFIG, DEBUG, and so forth. To a limited extent you can impr
# security of read only slaves using 'rename-command' to shadow all the
# administrative / dangerous commands.
slave-read-only no
```

修改完成后，重启从服务器。

修改和重启从服务器后，从服务器依然可以从主服务器去同步数据，这一点不用担心。

把应用访问的IP更换成从服务器的IP：

这一步忽略，实际上就是业务对数据库访问IP的变更。

把从服务器设置为主服务器：

在从上执行这个命令

```
127.0.0.1:6379> slaveof no one
OK
```

上面这个命令是将从服务器关闭复制功能，将从服务器作为独立的主服务器运行。这个很好理解，就是它不去同步其他的数据库服务器了，自己单独运行，那么自然就是主服务器。

上面的命令只是关闭复制功能，并没有切换主服务器。比如如果我们同时有主服务器和从服务器，

上面的命令还有其他用法就是强制切换主服务器，比如如果我们暂时想让从服务器去其他主服务器去同步数据，可以这样运行，如下图：

注意：假如当前服务器已经是A服务器的从服务器，而你又使用该命令让当前服务器成为B服务器的，么当前服务器将丢弃之前同步的就数据，开始对新的主服务器进行同步。
这个命令只是临时生效，临时覆盖配置文件中的slaveof设置，重启后恢复。

```
1 slaveof MasterIP MasterPort
```

验证一下复制是否停止：

在原来的主服务器上运行info命令，如下显示：

虽然角色是主，但是SLAVE的连接已经没有了

```
# Replication
role:master
connected_slaves:0
master_repl_offset:3135
repl_backlog_active:1
repl_backlog_size:1048576
repl_backlog_first_byte_offset:2
repl_backlog_histlen:3134
```



我们做一个SET操作，然后在原来的从服务器（Redis02）上执行一下GET操作，看看是否还可以获取数值：

```
[root@Redis01 bin]#
[root@Redis01 bin]#
[root@Redis01 bin]# ./redis-cli -a master123
127.0.0.1:6379> set A5 5
OK
127.0.0.1:6379> █
```



```
[root@Redis02 bin]#
[root@Redis02 bin]# ./redis-cli
127.0.0.1:6379> get A5
(nil)
127.0.0.1:6379> █
```



可以看到无法获取。

在Redis02上执行info命令看看结果：

显示它自己已经变成主了。

```
# Replication
role:master
connected_slaves:0
master_repl_offset:3079
repl_backlog_active:0
repl_backlog_size:1048576
```



```
repl_backlog_first_byte_offset:0  
repl_backlog_histlen:0
```

技术博客

通过上述的操作我们就可以替换原来的主服务器或者在主服务器失败的时候可以手动切换