

MySQL5.7配置主主同步集群

v1.0.20220601.1

一，双机高可用方案。

1，主机环境。

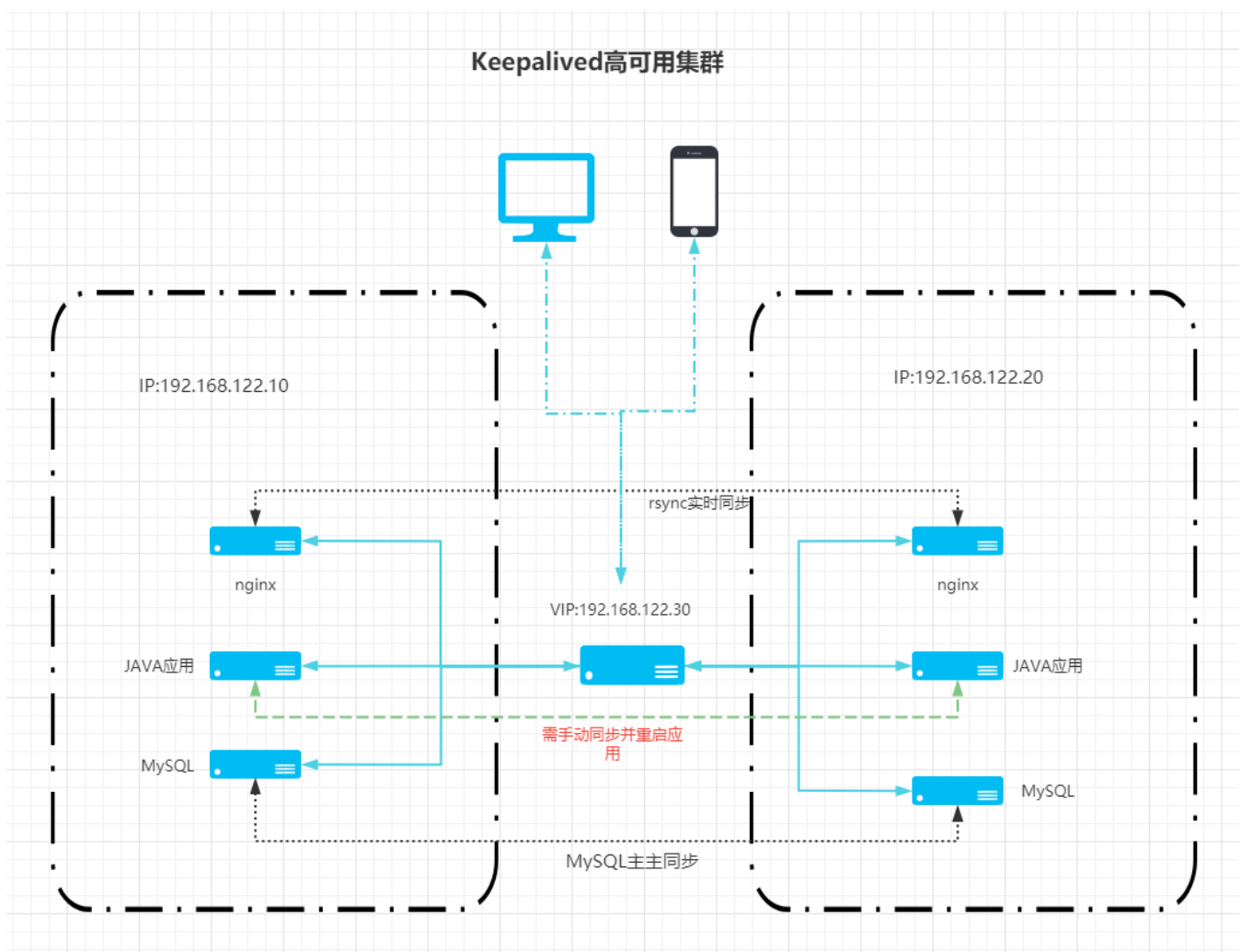
主机：IP: 192.168.122.10, Centos7, MySQL5.7, nginx, keepalived, rsync

备机：IP: 192.168.122.20, Centos7, MySQL5.7, nginx, keepalived, rsync

VIP: 192.168.122.30

实现效果：两台主机配置mysql双主同步，nginx+rsync实现web资源实时双向同步，vip提供高可用，JAVA应用需手动同步。

2，业务系统高可用拓扑图



二, MySQL安装

1, 主机与备机安装方法一样。

```
sed -i 's|SELINUX=enforcing|SELINUX=disabled|' /etc/selinux/config
# 关闭selinux,并重启系统

[root@vm10 ~]# cat /etc/hosts
127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.122.10 vm10
192.168.122.20 vm20

#添加主机名称与ip信息
#添加ssh互信, 这里不再说明操作方法

yum -y install bash-completion wget
wget https://repo.huaweicloud.com/mysql/Downloads/MySQL-5.7/mysql-5.7.33-1.el7.x86_64.rpm-
bundle.tar

for i in vm10 vm20;do ssh $i 'tar xvf mysql-5.7.33-1.el7.x86_64.rpm-bundle.tar && yum
localinstall *.rpm -y';done
#解压并安装mysql包。

#注意: my.cnf配置文件。
server-id = 1                                #backup这台设置2 这个选项有的注意1和2
log-bin = mysql-bin                          #这个选项有的
binlog-ignore-db = mysql,information_schema  #忽略写入binlog日志的库
auto-increment-increment = 2                #字段变化增量值
auto-increment-offset = 1                   #初始字段ID为1,另外一台主机配置为2
slave-skip-errors = all                      #忽略所有复制产生的错误

# mysql中有自增长字段,在做数据库的主主同步时需要设置自增长的两个相关配置: auto_increment_offset和
auto_increment_increment。
# auto-increment-increment表示自增长字段每次递增的量,其默认值是1。它的值应设为整个结构中服务器的总数,
我这里用到两台服务器,所以值设为2。
# auto-increment-offset是用来设定数据库中自动增长的起点(即初始值),因为这两台服务器都设定了一次自动增长值
2,所以它们的起点必须得不同,这样才能避免两台服务器数据同步时出现主键冲突。
#注: 可以在my.cnf文件中添加“binlog_do_db=数据库名”配置项(可以添加多个)来指定要同步的数据库

for i in vm10 vm20;do ssh $i 'systemctl enable mysqld && systemctl start mysqld';done
#启用mysql服务

cat /var/log/mysqld.log |grep password |awk '{ print $NF }'
#查看mysql账号root的密码

#使用set global修改策略的参数值
mysql> set global validate_password_length=6;
mysql> set global validate_password_mixed_case_count=0;
mysql> set global validate_password_special_char_count=0;
mysql> set global validate_password_policy=LOW;
mysql> ALTER USER 'root'@'localhost' IDENTIFIED BY 'abc132451@';
mysql> flush privileges;

#查看策略
```

```
mysql> show variables like 'validate_password%';
```

```
[root@vm10 ~]# firewall-cmd --permanent --add-service=mysql
```

```
[root@vm10 ~]# firewall-cmd --reload
```

2, 添加 slave账号, 数据库未有数据。

以下是在IP: 192.168.122.10主机的数据库操作。

```
mysql> grant replication slave on *.* to 'repl'@'192.168.122.%' identified by 'repl@pass';
mysql> flush privileges;
```

#锁表, 不让数据写入

```
mysql> show master status;
```

```
+-----+-----+-----+-----+-----+
-+
| File           | Position | Binlog_Do_DB | Binlog_Ignore_DB | Executed_Gtid_Set |
|
+-----+-----+-----+-----+-----+
-+
| mysql-bin.000001 |      1174 |              | mysql,information_schema |
|
+-----+-----+-----+-----+-----+
-+
```

3, 配置备机数据库。

以下是在IP: 192.168.122.20主机的数据库操作。

```
mysql> change master to
master_host='192.168.122.10',master_user='repl',master_password='repl@pass',master_log_file=
'mysql-bin.000001',master_log_pos=1174;
```

```
mysql> start slave;
```

```
mysql> show slave status\G
```

#如果只是配置主从数据库同步, 这里就可以结束了。

#以下操作是给主数据库添加从机数据库同步功能。

```
mysql> grant replication slave on *.* to 'repl'@'192.168.122.%' identified by 'repl@pass';
mysql> flush privileges;
mysql> show master status;
```

```
+-----+-----+-----+-----+-----+
-+
| File           | Position | Binlog_Do_DB | Binlog_Ignore_DB | Executed_Gtid_Set |
|
+-----+-----+-----+-----+-----+
-+
| mysql-bin.000002 |       611 |              | mysql,information_schema |
|
```

```
+-----+-----+-----+-----+-----+
-+
```

4, 配置主机的同步。

以下是在IP: 192.168.122.10主机的数据库操作。

```
mysql> change master to
master_host='192.168.122.20',master_user='repl',master_password='replpass',master_log_file=
'mysql-bin.000002',master_log_pos=611;
mysql> start slave;
mysql> show slave status\G
```

5, 测试双主同步

#在一台主机上面执行命令

```
create database blog;
CREATE USER `blog`@`%` IDENTIFIED WITH mysql_native_password BY '12345678' PASSWORD EXPIRE
NEVER;
grant all privileges on blog.* to blog@'%' identified by "12345678";
flush privileges;
```

#在另外一台主机上面执行命令

```
create database bbs;
CREATE USER `bbs`@`%` IDENTIFIED WITH mysql_native_password BY '12345678' PASSWORD EXPIRE
NEVER;
grant all privileges on bbs.* to bbs@'%' identified by "12345678";
flush privileges;
```

#查看双台主机的数据是否同步。

```
mysql -u blog -h 192.168.122.20 -p
mysql -u bbs -h 192.168.122.10 -p
```

三, nginx安装与实时同步

1, 安装nginx或是bt面板+nginx。

```
for i in vm10 vm20;do ssh $i 'rpm -ivh
http://nginx.org/packages/centos/7/noarch/RPMS/nginx-release-centos-7-0.el7ngx.noarch.rpm
&& yum -y install nginx && systemctl start nginx && systemctl enable nginx';done
```

nginx的配置文件根据情况修改, 这里不再操作

2, 安装rsync+lsyncd同步软件。

```
for i in vm10 vm20;do ssh $i 'yum -y install epel-release && yum -y install lsyncd ';done
```

此处我就是采用root用户, 故而不进行创建新用户, 具体操作过程根据实际情况进行配置。

```
[root@vm10 ~]# cat /etc/rsyncd.conf
uid = root
gid = root
# 锁死目录
use chroot = no
# 最大连接数
max connections = 10000
# 端口号
port = 873
log file = /var/log/rsyncd.log
pid file = /var/run/rsyncd.pid

# 可读可写
read only = no
# 模块名
[nginx]
# 指定目录
path = /usr/share/nginx/html
# 可读可写
read only = no
# 授权用户
auth users = root
# 授权IP 指向另外一台主机ip
hosts allow = 192.168.122.20/32
# 授权用户的密码
secrets file = /root/.rsync_server.passwd

# 以上是/etc/rsyncd.conf配置文件内容，这需同步的目录为nginx的目录根据情况添加或修改
# 两台主机都要配置，根据情况修改配置文件内容。

[root@vm10 ~]# cat /root/.rsync_server.passwd
root:12345678
[root@vm10 ~]# cat /root/.rsync_client.passwd
12345678
# 创建密码文件，并授权
[root@vm10 ~]# chmod 600 /root/.rsync_*

systemctl start rsyncd && systemctl enable rsyncd

#启用rsyncd服务
```

3, 配置lsyncd实时监控软件。

```
[root@vm10 ~]# cat /etc/lsyncd.conf
settings {
    logfile = "/var/log/lsyncd/lsyncd.log",
    statusFile = "/var/log/lsyncd/lsyncd.status",
    inotifyMode = "closewrite",
    maxProcesses = 7,
}
sync {
    default.rsync,
    source = "/usr/share/nginx/html",
    target = "root@192.168.122.20::nginx",
```

```

delete="running",
exclude = { ".*" },
delay = 1,
rsync = {
    binary = "/usr/bin/rsync",
    archive = true,
    compress = true,
    verbose = true,
    password_file = "/root/.rsync_client.passwd",
    _extra = {"--bwlimit=200"}
}
}

```

修改lsyncd.conf配置文件，采用rsync方式同步数据，两台主机都要配置，只是target配置成对方主机信息。

```
systemctl start lsyncd && systemctl enable lsyncd
```

#启用服务

4, 配置firewalld防火墙规则并测试同步效果。

```
firewall-cmd --permanent --add-service=rsyncd
```

```
firewall-cmd --reload
```

配置防火墙规则

```
[root@vm10 ~]# echo 222255555 > /usr/share/nginx/html/1.html
```

#添加测试的html文件

```
[root@vm10 ~]# curl http://192.168.122.10/1.html
```

```
222255555
```

```
[root@vm10 ~]# curl http://192.168.122.20/1.html
```

```
222255555
```

经测试两台主机文件已经同步成功。

注意：两台主机都要重启系统后测试是否正常同步。

5, lsyncd服务守护进程。

```
[root@vm10 ~]# cat /usr/local/bin/lzdps.sh
```

```
#!/bin/sh
```

lsync进程检测脚本

```
lsync_NAME=lsyncd
```

```
while true; do
```

```
    lsync_NUM=$(ps aux | grep lsyncd | grep -v grep | wc -l)
```

少于1, 重启进程

```
    if [ "${lsync_NUM}" -lt "1" ]; then
```

```
        echo "${lsync_NAME} was killed"
```

```
        date "+%Y-%m-%d %H:%M:%S"
```

```
        systemctl restart ${lsync_NAME}
```

```
    fi
```

```
done
```

```
exit 0
```

6, rsyncd服务守护进程。

```
[root@vm10 ~]# cat /usr/local/bin/rsdps.sh
#!/bin/sh
#syncd进程检测脚本
rsync_NAME=rsync
lsync_NAME=lsyncd

while true; do
    rsync_NUM=$(ps aux | grep rsync | grep -v grep | wc -l)
    # 少于1, 重启进程
    if [ "${rsync_NUM}" -lt "1" ]; then
        echo "${rsync_NAME} was killed"
        date "+%Y-%m-%d %H:%M:%S"
        systemctl restart ${rsync_NAME}
        sleep 3
        systemctl restart ${lsync_NAME}
    fi
done
exit 0
```

7, 添加任务计划。

```
# 添加执行权限
[root@vm10 ~]# chmod u+x /usr/local/bin/*.sh

#添加计划任务
crontab -l

*/2 * * * * /usr/local/bin/lsdps.sh >> /var/log/lsdps.log 2>&1
*/3 * * * * /usr/local/bin/rsdps.sh >> /var/log/rsdps.log 2>&1
```