http://mirrors.cn99.com/centos/7/isos/x86\_64/CentOS-7-x86\_64-DVD-1708.iso CentOS镜像

wget https://cdn.mysql.com//Downloads/MySQL-5.7/mysql-5.7.22-1.el7.x86\_64.rpm-bundle.tar (mysql安装路径)

wget -O /etc/yum.repos.d/CentOS-Base.repo http://mirrors.aliyun.com/repo/Centos-7.repo yum源

rpm -e --nodeps 包名

yum -y localinstall \*.rpm 解决本地rpm的依赖包

>/var/log/wtmp 清除历史用户登录信息

>/var/run/utmp 清除当前登录用户信息

>/var/log/btmp 清除登录失败用户信息

/var/log/secure：记录登入系统存取数据的文件，例如 pop3, ssh, telnet, ftp 等都会被记录；

/var/log/wtmp：记录登入者的讯息数据，由于本文件已经被编码过，所以必须使用 last指令来取出文件的内容；

/var/log/messages：尤为重要，几乎发生的错误讯息（或是重要信息）都会被记录在此；

/var/log/boot.log：记录开机或者是一些服务启动的时候，所显示的启动或关闭讯息；

/var/log/maillog 或 /var/log/mail/\*：纪录邮件存取或往来( sendmail 与 pop3 )的使用者记录；

/var/log/cron：记录 crontab 这个例行性服务的内容的。

主主环境（这里只介绍2台主的配置方案）:

1.CentOS 6.8 64位 2台：masterA（192.168.10.11），masterB（192.168.10.12）

2.官方Mysql5.6版本

搭建过程：

1.安装MySQL服务（建议源码安装）

1.1 yum安装依赖包

yum -y install make gcc gcc-c++ ncurses-devel bison openssl-devel

1.2 添加MySQL所需要的用户和组

groupadd -g 27 mysql

adduser -u 27 -g mysql -s /sbin/nologin mysql

1.3 下载MySQL源码包

mkdir -p /data/packages/src

cd /data/packages/

wget http://distfiles.macports.org/cmake/cmake-3.2.3.tar.gz

wget http://dev.mysql.com/get/Downloads/MySQL-5.6/mysql-5.6.34.tar.gz

1.4 创建mysql数据目录

mkdir -p /usr/local/mysql/data

1.5 解压编译安装cmake、MySQL

复制代码

cd /data/packages/src

tar -zxvf ../cmake-3.2.3.tar.gz

cd cmake-3.2.3/

./bootstrap

gmake

make install

cd ../

tar xf mysql-5.6.34.tar.gz

cd mysql-5.6.34

cmake . -DCMAKE\_INSTALL\_PREFIX=/usr/local/mysql -DSYSCONFDIR=/etc \

-DWITH\_SSL=bundled -DDEFAULT\_CHARSET=utf8 -DDEFAULT\_COLLATION=utf8\_general\_ci \

-DWITH\_INNOBASE\_STORAGE\_ENGINE=1 -DWITH\_MYISAM\_STORAGE\_ENGINE=1 \

-DMYSQL\_TCP\_PORT=3306 -DMYSQL\_UNIX\_ADDR=/tmp/mysql.sock \

-DMYSQL\_DATADIR=/usr/local/mysql/data

make && make install

复制代码

1.6 添加开机启动脚本

cp support-files/mysql.server /etc/rc.d/init.d/mysqld

1.7 添加masterA配置文件/etc/my.cnf

复制代码

[client]

port = 3306

socket = /tmp/mysql.sock

[mysqld]

basedir = /usr/local/mysql

port = 3306

socket = /tmp/mysql.sock

datadir = /usr/local/mysql/data

pid-file = /usr/local/mysql/data/mysql.pid

log-error = /usr/local/mysql/data/mysql.err

server-id = 1

auto\_increment\_offset = 1

auto\_increment\_increment = 2 #奇数ID

log-bin = mysql-bin #打开二进制功能,MASTER主服务器必须打开此项

binlog-format=ROW

binlog-row-p\_w\_picpath=minimal

log-slave-updates=true

gtid-mode=on

enforce-gtid-consistency=true

master-info-repository=TABLE

relay-log-info-repository=TABLE

sync-master-info=1

slave-parallel-workers=0

sync\_binlog=0

binlog-checksum=CRC32

master-verify-checksum=1

slave-sql-verify-checksum=1

binlog-rows-query-log\_events=1

#expire\_logs\_days=5

max\_binlog\_size=1024M #binlog单文件最大值

replicate-ignore-db = mysql #忽略不同步主从的数据库

replicate-ignore-db = information\_schema

replicate-ignore-db = performance\_schema

replicate-ignore-db = test

replicate-ignore-db = zabbix

max\_connections = 3000

max\_connect\_errors = 30

skip-character-set-client-handshake #忽略应用程序想要设置的其他字符集

init-connect='SET NAMES utf8' #连接时执行的SQL

character-set-server=utf8 #服务端默认字符集

wait\_timeout=1800 #请求的最大连接时间

interactive\_timeout=1800 #和上一参数同时修改才会生效

sql\_mode=NO\_ENGINE\_SUBSTITUTION,STRICT\_TRANS\_TABLES #sql模式

max\_allowed\_packet = 10M

bulk\_insert\_buffer\_size = 8M

query\_cache\_type = 1

query\_cache\_size = 128M

query\_cache\_limit = 4M

key\_buffer\_size = 256M

read\_buffer\_size = 16K

skip-name-resolve

slow\_query\_log=1

long\_query\_time = 6

slow\_query\_log\_file=slow-query.log

innodb\_flush\_log\_at\_trx\_commit = 2

innodb\_log\_buffer\_size = 16M

[mysql]

no-auto-rehash

[myisamchk]

key\_buffer\_size = 20M

sort\_buffer\_size = 20M

read\_buffer = 2M

write\_buffer = 2M

[mysqlhotcopy]

interactive-timeout

[mysqldump]

quick

max\_allowed\_packet = 16M

[mysqld\_safe]

复制代码

1.8 特别参数说明

log-slave-updates = true #将复制事件写入binlog,一台服务器既做主库又做从库此选项必须要开启

#masterA自增长ID

auto\_increment\_offset = 1

auto\_increment\_increment = 2 #奇数ID

#masterB自增加ID

auto\_increment\_offset = 2

auto\_increment\_increment = 2 #偶数ID

1.9 添加masterB配置文件/etc/my.cnf

复制代码

[client]

port = 3306

socket = /tmp/mysql.sock

[mysqld]

basedir = /usr/local/mysql

port = 3306

socket = /tmp/mysql.sock

datadir = /usr/local/mysql/data

pid-file = /usr/local/mysql/data/mysql.pid

log-error = /usr/local/mysql/data/mysql.err

server-id = 2

auto\_increment\_offset = 2

auto\_increment\_increment = 2 #偶数ID

log-bin = mysql-bin #打开二进制功能,MASTER主服务器必须打开此项

binlog-format=ROW

binlog-row-p\_w\_picpath=minimal

log-slave-updates=true

gtid-mode=on

enforce-gtid-consistency=true

master-info-repository=TABLE

relay-log-info-repository=TABLE

sync-master-info=1

slave-parallel-workers=0

sync\_binlog=0

binlog-checksum=CRC32

master-verify-checksum=1

slave-sql-verify-checksum=1

binlog-rows-query-log\_events=1

#expire\_logs\_days=5

max\_binlog\_size=1024M #binlog单文件最大值

replicate-ignore-db = mysql #忽略不同步主从的数据库

replicate-ignore-db = information\_schema

replicate-ignore-db = performance\_schema

replicate-ignore-db = test

replicate-ignore-db = zabbix

max\_connections = 3000

max\_connect\_errors = 30

skip-character-set-client-handshake #忽略应用程序想要设置的其他字符集

init-connect='SET NAMES utf8' #连接时执行的SQL

character-set-server=utf8 #服务端默认字符集

wait\_timeout=1800 #请求的最大连接时间

interactive\_timeout=1800 #和上一参数同时修改才会生效

sql\_mode=NO\_ENGINE\_SUBSTITUTION,STRICT\_TRANS\_TABLES #sql模式

max\_allowed\_packet = 10M

bulk\_insert\_buffer\_size = 8M

query\_cache\_type = 1

query\_cache\_size = 128M

query\_cache\_limit = 4M

key\_buffer\_size = 256M

read\_buffer\_size = 16K

skip-name-resolve

slow\_query\_log=1

long\_query\_time = 6

slow\_query\_log\_file=slow-query.log

innodb\_flush\_log\_at\_trx\_commit = 2

innodb\_log\_buffer\_size = 16M

[mysql]

no-auto-rehash

[myisamchk]

key\_buffer\_size = 20M

sort\_buffer\_size = 20M

read\_buffer = 2M

write\_buffer = 2M

[mysqlhotcopy]

interactive-timeout

[mysqldump]

quick

max\_allowed\_packet = 16M

[mysqld\_safe]

复制代码

1.10 初始化MySQL

cd /usr/local/mysql

scripts/mysql\_install\_db --user=mysql

1.11 为启动脚本赋予可执行权限并启动MySQL

chmod +x /etc/rc.d/init.d/mysqld

/etc/init.d/mysqld start

2. 配置主从同步

2.1 添加主从同步账户

masterA上：

mysql> grant replication slave on \*.\* to 'repl'@'192.168.10.12' identified by '123456';

mysql> flush privileges;

masterB上：

mysql> grant replication slave on \*.\* to 'repl'@'192.168.10.11' identified by '123456';

mysql> flush privileges;

2.2 查看主库的状态

masterA上：

复制代码

mysql> show master status;

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

| File | Position | Binlog\_Do\_DB | Binlog\_Ignore\_DB | Executed\_Gtid\_Set |

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

| mysql-bin.000003 | 120 | | | |

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

1 row in set (0.00 sec)

复制代码

masterB上

复制代码

mysql> show master status;

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

| File | Position | Binlog\_Do\_DB | Binlog\_Ignore\_DB | Executed\_Gtid\_Set |

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

| mysql-bin.000003 | 437 | | | |

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

1 row in set (0.00 sec)

复制代码

2.3 配置同步信息：

masterA上：

mysql> change master to master\_host='192.168.10.12',master\_port=3306,master\_user='repl',master\_password='123456',master\_log\_file='mysql-bin.000003',master\_log\_pos=437;

mysql> start slave;

mysql> show slave status\G;

显示有如下状态则正常：

Slave\_IO\_Running: Yes

Slave\_SQL\_Running: Yes

masterB上：

#本人是测试环境，可以保证没数据写入，否则需要的步骤是：先masterA锁表-->masterA备份数据-->masterA解锁表 -->masterB导入数据-->masterB设置主从-->查看主从

mysql> change master to master\_host='192.168.10.11',master\_port=3306,master\_user='repl',master\_password='123456',master\_log\_file='mysql-bin.000003',master\_log\_pos=120;

start slave;

mysql> show slave status\G;

显示有如下状态则正常：

Slave\_IO\_Running: Yes

Slave\_SQL\_Running: Yes

3.测试主从同步

3.1 在masterA上创建一个数据库测试同步效果

复制代码

mysql> show databases;

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

| Database |

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

| information\_schema |

| mysql |

| performance\_schema |

| test |

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

4 rows in set (0.00 sec)

mysql> create database test01;

Query OK, 1 row affected (0.00 sec)

mysql> show databases;

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

| Database |

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

| information\_schema |

| mysql |

| performance\_schema |

| test |

| test01 |

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

5 rows in set (0.00 sec)

mysql> quit

Bye

[root@masterA data]#

复制代码

3.2 到masterB查看是否已经同步创建数据库

复制代码

mysql> show databases;

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

| Database |

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

| information\_schema |

| mysql |

| performance\_schema |

| test |

| test01 |

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

5 rows in set (0.00 sec)

mysql> quit

Bye

[root@masterB data]#

复制代码

4. 开启MySQL5.6的GTID功能

masterA和masterB分别执行如下命令：

复制代码

mysql> stop slave;

Query OK, 0 rows affected (0.00 sec)

mysql> change master to MASTER\_AUTO\_POSITION=1;

Query OK, 0 rows affected (0.01 sec)

mysql> start slave;

Query OK, 0 rows affected (0.00 sec)

按照系统更新yum源，我的是centos7，所以我运行下面的代码！

CentOS/RHEL 7.x:yum -y install epel-release(安装epel源）

1、rpm -Uvh https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm

2、rpm -Uvh https://mirror.webtatic.com/yum/el7/webtatic-release.rpm

3、yum -y install php70w

如果是centos6，那么执行以下代码：

CentOS/RHEL 6.x:

1、rpm -Uvh https://dl.fedoraproject.org/pub/epel/epel-release-latest-6.noarch.rpm

2、rpm -Uvh https://mirror.webtatic.com/yum/el6/latest.rpm

然后就可以直接yum安装php7.0了，可以安装的拓展如下：

yum install php70w-common php70w-fpm php70w-opcache php70w-gd php70w-mysqlnd php70w-mbstring php70w-pecl-redis php70w-pecl-memcached php70w-devel

就这一条命令，就可以安装了下面的拓展了！

php-api, php-bz2, php-calendar, php-ctype, php-curl, php-date, php-exif, php-fileinfo, php-filter, php-ftp, php-gettext, php-gmp, php-hash, php-iconv, php-json, php-libxml, php-openssl, php-pcre, php-pecl-Fileinfo, php-pecl-phar, php-pecl-zip, php-reflection, php-session, php-shmop, php-simplexml, php-sockets, php-spl, php-tokenizer, php-zend-abi, php-zip, php-zlib

CentOS防火墙在虚拟机的CENTOS装好APACHE不能用,郁闷,解决方法如下

/sbin/iptables -I INPUT -p tcp --dport 80 -j ACCEPT

/sbin/iptables -I INPUT -p tcp --dport 22 -j ACCEPT

/sbin/iptables -F

/sbin/iptables -P INPUT DROP

/sbin/iptables -P OUTPUT ACCEPT

/sbin/iptables -P FORWARD ACCEPT

/sbin/iptables -A INPUT -m state --state INVALID -j LOG --log-level 4 --log-prefix 'InvalidDrop '

/sbin/iptables -A INPUT -m state --state INVALID -j DROP

/sbin/iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

/sbin/iptables -A INPUT -j LOG --log-level 4 --log-prefix 'In'

/sbin/iptables -A OUTPUT -m limit -j LOG --log-level 4 --log-prefix 'Out'

/sbin/iptables -A INPUT -i lo -j ACCEPT

/sbin/iptables -A INPUT -s 127.0.0.1 -j ACCEPT

/sbin/iptables -A INPUT -p tcp --dport 21 -j ACCEPT

/sbin/iptables -A INPUT -p tcp --dport 22 -j ACCEPT

/sbin/iptables -A INPUT -p tcp --dport 25 -j ACCEPT

/sbin/iptables -A INPUT -p tcp --dport 53 -j ACCEPT

/sbin/iptables -A INPUT -p tcp --dport 110:65534 -j ACCEPT

/sbin/iptables -A INPUT -p udp --dport 21:65534 -j ACCEPT

/sbin/iptables -A INPUT -i eth1 -p tcp -s 58.83.238.20 --dport 80 -j ACCEPT

/sbin/iptables -A INPUT -i eth1 -p tcp -s 58.83.238.21 --dport 80 -j ACCEPT

/sbin/iptables -A INPUT -i eth1 -p tcp -s 219.142.118.0/24 --dport 80 -j ACCEPT

/sbin/iptables -A INPUT -i eth1 -p tcp -s 10.0.0.0/8 --dport 80 -j ACCEPT

还原为iptables

systemctl stop firewalld

systemctl mask firewalld

yum install iptables-services

systemctl enable iptables

systemctl stop iptables

systemctl start iptables

systemctl restart iptables

systemctl reload iptables

mysql查询报错1055解决办法

sql\_mode='STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION'

APACHE下多个域名绑定到一个目录下 如何设置

例：比如域名分别为www.site1.com;site1.com;www.site1.cn

<VirtualHost 202.100.1.1>

ServerName www.site1.cn

ServerAlias site1.com

ServerAlias www.site1.com

DocumentRoot "D:/PC\_webserver/phproot/site"

</VirtualHost>

http强制跳转https

RewriteEngine on

RewriteCond %{HTTPS} !=on

RewriteRule ^(.\*) https://%{SERVER\_NAME}$1 [L,R]