**最小化安装机器情况下，安装mysql5.6**

1. 初始化机器环境
2. 安装阿里云yum源
3. 备份原来的yum源

cp /etc/yum.repos.d/CentOS-Base.repo /etc/yum.repos.d/CentOS-Base.repo.bak

1. 设置aliyun的yum源：

Wget -O /etc/yum.repos.d/CentOS-Base.repo <http://mirrors.aliyun.com/repo/Centos-7.repo>

1. 添加EPEL源:

wget -P /etc/yum.repos.d/ <http://mirrors.aliyun.com/repo/epel-7.repo>

1. 清理缓存并生成新的缓存:

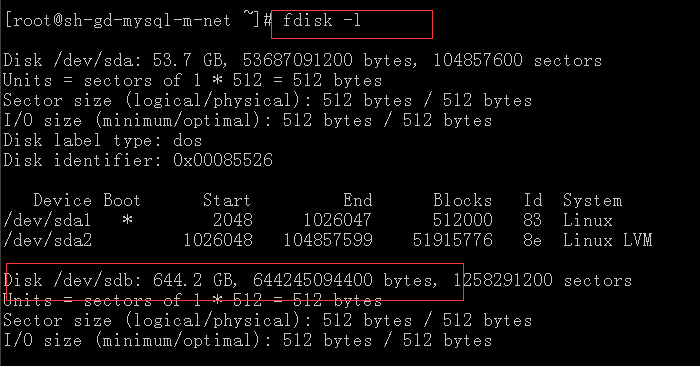
sudo yum clean all

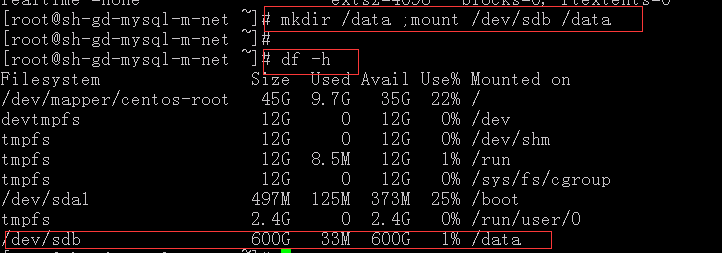
sudo yum makecache

1. 挂载数据盘
   1. 查看当前是否有磁盘可挂载
      1. fdisk –l
      2. 若是没有发现可挂载的硬盘，需要重启机器

reboot

重启后发现，我们需要的数据盘



* 1. 开始格式化硬盘并且挂载
     1. mkfs.xfs /dev/sdb
     2. 
     3. 放到开机启动内，以便机器开机后，可以第一时间进行挂载动作

vim /etc/rc.local

添加：mount /dev/sdb /data

1. 安装vim工具
   1. Yum install vim
2. 卸载centos7自带的firewall服务，安装iptables服务
   1. systemctl stop firewalld.service
   2. yum remove firewall\*
   3. yum install iptables-services
   4. 编辑iptables初始规则

# 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 80 -j ACCEPT

-A INPUT -m state --state NEW -m tcp -p tcp --dport 3306 -j ACCEPT

-A INPUT -j REJECT --reject-with icmp-host-prohibited

-A FORWARD -j REJECT --reject-with icmp-host-prohibited

COMMIT

* 1. 启动并且配置开机启动

systemctl restart iptables.service #最后重启防火墙使配置生效

systemctl enable iptables.service #设置防火墙开机启动

1. 内核优化
   1. [root@sh-gd-mysql-m-net scripts]# cat /etc/sysctl.conf

# System default settings live in /usr/lib/sysctl.d/00-system.conf.

# To override those settings, enter new settings here, or in an /etc/sysctl.d/<name>.conf file

# For more information, see sysctl.conf(5) and sysctl.d(5).

icmp\_echo\_ignore\_broadcasts = 1

net.ipv4.icmp\_ignore\_bogus\_error\_responses = 1

net.ipv4.ip\_forward = 0

net.ipv4.conf.all.send\_redirects = 0

net.ipv4.conf.default.send\_redirects = 0

net.ipv4.conf.default.rp\_filter = 1

net.ipv4.conf.default.accept\_source\_route = 0

kernel.sysrq = 0

kernel.core\_uses\_pid = 1

net.ipv4.tcp\_syncookies = 1

kernel.msgmax = 65536

kernel.shmmax = 68719476736

kernel.shmall = 4294967296

net.ipv4.tcp\_max\_tw\_buckets = 6000

net.ipv4.tcp\_sack = 1

net.ipv4.tcp\_window\_scaling = 1

net.ipv4.tcp\_rmem = 4096 87380 4194304

net.ipv4.tcp\_wmem = 4096 16384 4194304

net.core.wmem\_default = 8388608

net.core.rmem\_default = 8388608

net.core.rmem\_max = 16777216

net.core.wmem\_max = 16777216

net.core.netdev\_max\_backlog = 262144

net.ipv4.tcp\_max\_syn\_backlog = 262144

net.ipv4.tcp\_timestamps = 0

net.ipv4.tcp\_synack\_retries = 1

net.ipv4.tcp\_syn\_retries = 1

net.ipv4.tcp\_tw\_recycle = 1

net.ipv4.tcp\_tw\_reuse = 1

net.ipv4.tcp\_mem = 94500000 915000000 927000000

net.ipv4.tcp\_fin\_timeout = 1

net.ipv4.tcp\_keepalive\_time = 30

net.ipv4.ip\_local\_port\_range = 1024 65000

net.ipv4.conf.all.accept\_redirects = 0

net.ipv4.conf.default.accept\_redirects = 0

net.ipv4.conf.all.secure\_redirects = 0

net.ipv4.conf.default.secure\_redirects = 0

* 1. 修改limit限制

/etc/security/limits.conf

\* soft nofile 65535

\* hard nofile 65535

\* soft npoc 65535

\* hard npoc 65535

1. **安装mysql5.6**
   1. 由于centos7默认的mysql数据库为其分支，mariadb，我们需要添加mysql的yum源
      1. rpm -Uvh <http://dev.mysql.com/get/mysql-community-release-el7-5.noarch.rpm>
      2. 查看可安装的mysql版本
         1. yum repolist enabled | grep "mysql.\*-community.\*"
      3. 安装mysqlserver
         1. yum -y install mysql-community-server
      4. 优化server的配置文件

# For advice on how to change settings please see

# http://dev.mysql.com/doc/refman/5.6/en/server-configuration-defaults.html

[client]

default-character-set=utf8

[mysqld]

#

# Remove leading # and set to the amount of RAM for the most important data

# cache in MySQL. Start at 70% of total RAM for dedicated server, else 10%.

# innodb\_buffer\_pool\_size = 128M

#

# Remove leading # to turn on a very important data integrity option: logging

# changes to the binary log between backups.

# log\_bin

#

# Remove leading # to set options mainly useful for reporting servers.

# The server defaults are faster for transactions and fast SELECTs.

# Adjust sizes as needed, experiment to find the optimal values.

# join\_buffer\_size = 128M

# sort\_buffer\_size = 2M

# read\_rnd\_buffer\_size = 2M

datadir=/data/mysql

socket=/var/lib/mysql/mysql.sock

# Disabling symbolic-links is recommended to prevent assorted security risks

symbolic-links=0

innodb\_buffer\_pool\_size = 128M

###################################

#innodb

user=mysql

innodb\_buffer\_pool\_size=6G

innodb\_log\_file\_size=4G

innodb\_log\_buffer\_size = 8M

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

innodb\_file\_per\_table=1

innodb\_file\_io\_threads=4

innodb\_flush\_method=O\_DIRECT

innodb\_io\_capacity=2000

innodb\_io\_capacity\_max=6000

innodb\_lru\_scan\_depth=2000

innodb\_thread\_concurrency = 0

innodb\_additional\_mem\_pool\_size=16M

innodb\_autoinc\_lock\_mode = 2

##################################################

# Binary log/replication

log-bin

sync\_binlog=1

sync\_relay\_log=1

relay-log-info-repository=TABLE

master-info-repository=TABLE

expire\_logs\_days=7

binlog\_format=ROW

transaction-isolation=READ-COMMITTED

#################################################

#cache

tmp\_table\_size=512M

character-set-server=utf8

collation-server=utf8\_general\_ci

skip-external-locking

back\_log=1024

key\_buffer\_size=1024M

thread\_stack=256k

read\_buffer\_size=8M

thread\_cache\_size=64

query\_cache\_size=128M

max\_heap\_table\_size=256M

query\_cache\_type=1

binlog\_cache\_size = 2M

table\_open\_cache=128

thread\_cache=1024

thread\_concurrency=8

wait\_timeout=30

join\_buffer\_size = 1024M

sort\_buffer\_size = 8M

read\_rnd\_buffer\_size = 8M

#################################################

#connect

max-connect-errors=100000

max-connections=1000

#################################################

explicit\_defaults\_for\_timestamp=true

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

[mysqld\_safe]

log-error=/var/log/mysqld.log

pid-file=/var/run/mysqld/mysqld.pid

* 1. 配置mysql主从
     1. 安装另外一台机器，配置方法和master一样，直到安装完成mysql，并且完成相应优化
     2. Master配置文件内更改，没有就添加,在my.cnf内
        1. log-bin=master-bin.log
        2. server-id=1
     3. slave配置文件内更改，没有就天津，在my.cnf内
        1. log-bin=slave-bin.log
        2. server-id=2
     4. master/slave 进行重启
        1. systemctl restart mysqld
     5. master上面创建用于同步的帐号

mysql> GRANT REPLICATION SLAVE ON \*.\* TO 'slave'@'10.32.30.231' IDENTIFIED BY '123456';

Query OK, 0 rows affected (0.00 sec)

mysql> flush privileges;

Query OK, 0 rows affected (0.00 sec)

* + 1. 由于是环境，所有，不需要停掉master的写，去同步数据
       1. 查看master的状态

mysql> show master status;

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

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

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

| master-bin.000001 | 409 | | | |

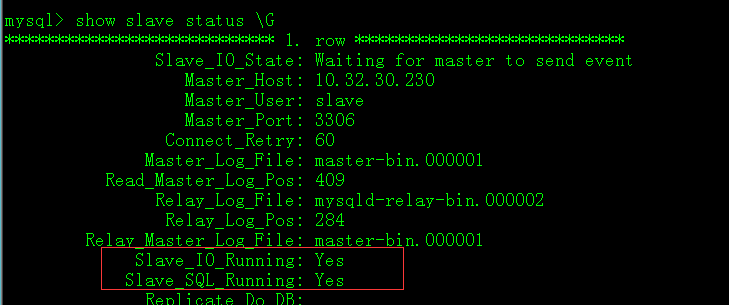
+-------------------+----------+--------------+------------------+-------------------+

1 row in set (0.00 sec)

* + 1. 在slave上面，设置master向slave同步的细节，注意标红部分
       1. mysql> CHANGE MASTER TO MASTER\_HOST='10.32.30.230',MASTER\_USER='slave',MASTER\_PASSWORD='123456',MASTER\_LOG\_FILE='master-bin.000001',MASTER\_LSTER\_LOG\_POS=409;
       2. 启动slave进行复制
          1. mysql> start slave;

Query OK, 0 rows affected (0.00 sec)

* + - * 1. 查看slave的状态，如果标红的均为yes，表示主从成功



* + 1. 数据测试

Master上面创建test库，然后去从库上面查看，若有，则验证成功

