mysql/mariadb篇

基础安装

初始化

```
#初始化
mysql_secure_installation
#登录不加载权限
mysqld_safe --skip-grant-tables
```

密码管理

```
#修改密码
mysqladmin -u root -p password
#修改本地密码
alter user 'root'@'localhost' identified by '12345678';
#查看初始化密码
grep -i 'temporary password' /var/log/mysqld.log
#也可以配置密码强度
validate_password.length = 8
validate_password.mixed_case_count = 1
validate_password.number_count = 1
validate_password.special_char_count = 1
#查询   mysql内置密码强度
SHOW VARIABLES LIKE 'validate_password%';
#修改   mysql密码策略参数
SET GLOBAL validate_password.policy = LOW;
```

用户管理

```
#创建网络超级管理员
create user 'root'@'%' identified by '12345678';
#授予网络超级管理员权限
grant all privileges on *.* to 'root'@'%' with grant option;
```

```
#刷新权限
flush privileges;
#删除用户
drop user 'laoda'@'12345678';
#查看用户权限
show grants for [用户名      ];
```

日志探寻

```
#日志清空
sudo truncate -s 0 /var/log/mysql.log
#日志查询
cat /var/log/mysqld.log | grep '2021-09-13'
tail -n500 /var/log/mysqld.log|grep -E 'Warning|ERROR'
cat group|grep mysql
cat passwd|grep mysql
```

表管理

```
#表字段描述
desc [表名    ];
#表索引
show index from [表名    ];
#创建表
create table if not exists '表名    '(
'id' int UNSIGNED AUTO_INCREMENT,
'字段名      ' varchar(100) not null,
primary  key ('id')
)engine=InnoDB default charset=utf8mb4;
#删除
drop table [表名    ];
#清空
truncate table [表名    ];
#查看创建表的            sql
show create table [表名    ];
#插入数据
insert into [表名    ]
([字段名      1],[字段名      2],[时间字段        ])
values
('值 1','值 2',now()),
```

```
('值1','值2',now());
#追加约束
alter table [表名] alter [字段名] set default [默认值];
#删除约束
alter table [表名] alter [字段名] drop default;
```

数据库管理

```
#新建
create database [数据库名] default charset utf8mb4 collate utf8_general_ci;
```

数据库备份

物理备份

```
#查询数据存储目录
show variables like 'datadir';
#归档源包
show variables like 'datadir';
tar -zcvf mysql.tar.gz mysql
#发送源包到目标机
scp mysql.tar.gz root@[目标ip]: [目标文件地址]
#解压到当目标Mysql的目录
tar -zxvf mysql.tar.gz
#临时关闭centos保护机制
```

逻辑备份

```
#备份机导出.sql文件
mysqlsdump -uroot -p12345678 --all-databases > all-first-$(data +%f).sql
#目标机导入.sql
mysql -uroot -p12345678 < all-first.sql
#查找log
show variables LIKE '%log%';
#修改配置文件
#开启二进制日志文件记录并设置二进制日志保存路径!!!
log_bin = mysqlbin
# 在每次提交事务时同步写入binlog!!!
```

```
sync_binlog = 1
# 设置服务器 ID!!!
server-id = 1
# 设置 binlog格式为 ROW!!!
binlog_format = ROW
#日志滚动
flush  logs;
#查询当前 bin-log
show BINARY LOGS;!!所有
show MASTER STATUS;
#查看具体操作
SHOW BINLOG EVENTS in 'binlog.000002';
#删除 bin-log文件
reset master;
#业务是在一直存入时
flush binary logs;
flush logs [binlog名 ] -> [binlog名 +1]
purge binary logs to '1-8';
#日志恢复
mysqlbinlog --no-defaults /var/lib/mysql/[日志文件名 ] --start-position=[开始值 ] --stop-position=[结束值 ] | mysql -uroot -p12345678 [数据库名 ]
#日志回滚
输出增量数据
mysqlbinlog /var/lib/mysql/[binlog名 ] --start-position=[起始值 ] --stop-position=[结束值 ] >export.sql
灌入整备数据
mysql -uroot -p12345678 < all.first.sql
灌入增量数据
source ./export.sql
```

主从复制

主库配置

```
#[必须 ]主服务器唯一 ID
server-id=1
#[必须 ]启用二进制日志 ,指名路径。比如：自己本地的路径 /log/mysqlbin
log-bin=mysql-bin
#[可选 ] 0（默认）表示读写（主机）， 1表示只读（从机）
read-only=0
```

```
# 自动清理 30天前的 log文件
expire_logs_days=30
#控制单个二进制日志大小。此参数的最大和默认值是 1GB
max_binlog_size=500M
#[可选 ]设置不要复制的数据库
binlog-ignore-db=test
#[可选 ]设置需要复制的数据库 ,默认全部记录。比如: binlog-do-db=atguigu_master_slave
binlog-do-db=【需要复制的主数据库名字】
#[可选 ]设置 binlog格式
binlog_format=STATEMENT
```

重启主库

```
#重启
systemctl restart mysqld
# 查看 log_bin日志是否启动成功
show variables like '%log_bin%';
#查看主从复制位置
show master status;
#创建主从复制账户
slave
```

配置从库

```
#[必须 ]从服务器唯一 ID
server-id=2
#[可选 ]启用中继日志
relay-log=mysql-relay
#设置数据库实例是否为只读模式
read_only = 1
#将主服务器的配置信息存储在数据库的表
master_info_repository=TABLE
#将中继日志的相关信息存储在数据库的表
relay_log_info_repository=TABLE
```

重启从库

```
#停止从库
stop slave;
```

```
#修改主从信息
change master to
master_host='192.168.148.130',
master_user='slave',
master_password='12345678',
master_port=3306,
master_log_file='mysql-bin.000003',
master_log_pos=346;
#启动从库
start slave;
#查看主从同步结果
show slave status\G;
```

集群部署

安装附属

```
#配置附属包
yum -y install perl-Time-HiRes
yum -y install perl-DBD-Mysql.x86_64
yum -y install libaio
yum -y install rsync
yum -y install lsof
yum -y install libev.so.4*
yum -y install perl-JSON.noarch
yum -y install perl.x86_64
yum -y install perl-devel.x86_64
yum -y install socat
yum -y install net-tools
yum -y install openssl-devel
#一般按不需要安装
yum -y install libboost_program_options.so*
rpm -ivh percona-xtrabackup*
```

清除自带

```
移除之前的      mariadb或者  mysql相配置
```

安装 Mysql-wsrep

```
自己想办法
```

安装 galera-4

```
rpm -ivh galera-4-26.4.16-1.el7.x86_64.rpm


//安装结果
rpm -qa | grep -E 'galera|mysql|percona'
```

编辑配置

```
#查找
find / -name libgalera_smm.so
#编辑  my.cnf
[mysqld]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
log-error=/var/log/mysqld.log
pid-file=/var/run/mysqld/mysqld.pid

net_read_timeout=3600
net_write_timeout=9000
max_allowed_packet=10000M
interactive_timeout=28800000
wait_timeout=28800000
max_connections=1000

wsrep_provider = /usr/lib64/galera-4/libgalera_smm.so
wsrep_cluster_name="mysql_galera_cluster"
wsrep_cluster_address= "gcomm://192.168.188.129,192.168.188.128"
#wsrep_sst_method=xt rabackup
wsrep_sst_auth=laoda:12345678
wsrep_node_name=node1
wsrep_node_address="192.168.188.129"
```

引导启动

```
#开始引导
/usr/bin/mysqld_bootstrap
```

```
//引导重启
#查找文件
grastate.dat
#修改牛逼程度
safe_to_bootstrap
```

启动结果

```
show status like '%wsrep_cluster%';
show variables like 'wsrep_cluster_address';
show variables like 'wsrep_auto_increment_control';
show variables like '%max_allowed_pack%';
show variables like 'innodb_buffer_pool%';
show global status like 'wsrep_local_state_comment';
```

读写分离　　proxysql中间件

仓库配置

```
cat > /etc/yum.repos.d/proxysql.repo << EOF
[proxysql]
name=ProxySQL YUM repository
baseurl=https://repo.proxysql.com/ProxySQL/proxysql-2.5.x/centos/\$releasever
gpgcheck=1
gpgkey=https://repo.proxysql.com/ProxySQL/proxysql-2.5.x/repo_pub_key
EOF
```

规则配置

```
#创建读写组
insert into mysql_replication_hostgroups ( writer_hostgroup, reader_hostgroup, comment) values (10,20,'proxy');
#创建主机地址
insert into mysql_servers(hostgroup_id,hostname,port,comment) values
(10,'192.168.148.130',3306,masterinstance);
insert into mysql_servers(hostgroup_id,hostname,port,comment) values
(20,'192.168.148.130',3306,slaveinstance);
#配置监控账户
set mysql-monitor_username='monitor';
set mysql-monitor_password='12345678';
```

```
#配置用户
insert into mysql_users (username,password,default_hostgroup) values ('proxysql','12345678',10);
#配置路由规则
insert into mysql_query_rules(rule_id,active,match_pattern,destination_hostgroup,apply) values
(1,1,'^select.*for update$',10,1);
insert into mysql_query_rules(rule_id,active,match_pattern,destination_hostgroup,apply) values
(2,1,'^select',20,1);
#读写配置
set global read_only=1;  只读
set global read_only=0;  读写
#查看只读是否开启
show global variables like 'read_only';
```

日志监控

```
#日志配置
select * from monitor.mysql_server_connect_log;#对 connect指标的监控
select * from mysql_server_ping_log limit 10;  #对心跳信息的监控              (对 ping 指标的监控        )
select * from mysql_server_read_only_log limit 10; #看看   read_only的日志监控
#查看请求日志
select hostgroup,schemaname,username,digest_text,count_star from  stats_mysql_query_digest;
SELECT hostgroup,schemaname,digest,digest_text,count_star,sum_time FROM stats_mysql_query_digest ORDER
BY sum_time DESC;
#后端监控
show tables from monitor;
```

配置加载

```
#常用配置加载
load mysql users to runtime;
save mysql users to disk;
load mysql servers to runtime;
save mysql servers to disk;
load mysql query rules to runtime;
save mysql query rules to disk;
load mysql variables to runtime;
save mysql variables to disk;
load admin variables to runtime;
save admin variables to disk;
```

版本号 #2

由 叶子 创建于 31 十月 2024 20:18:42

由 叶子 更新于 31 十月 2024 20:39:54