

DBA2 DAY01



DBA进阶

NSD DBA2

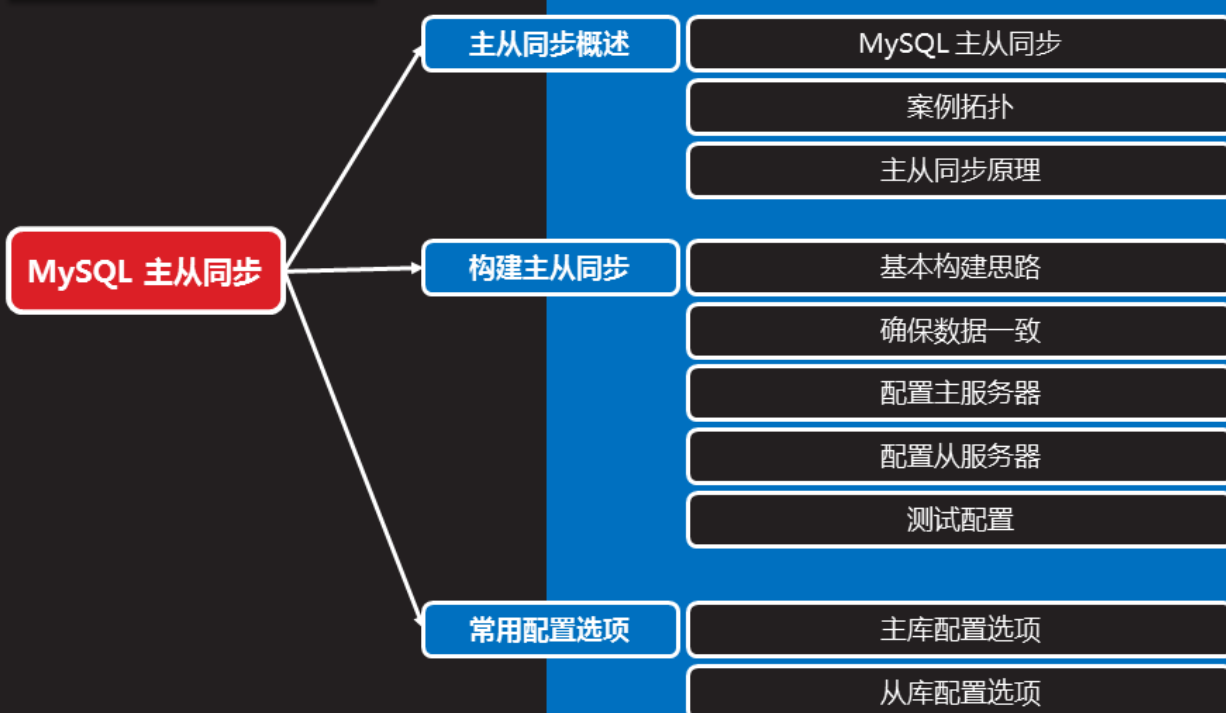
DAY01

内容

上午	09:00 ~ 09:30	MySQL 主从同步
	09:30 ~ 10:20	
	10:30 ~ 11:20	
	11:30 ~ 12:00	
下午	14:00 ~ 14:50	MySQL主从同步模式
	15:00 ~ 15:50	
	16:10 ~ 17:00	
	17:10 ~ 18:00	总结和答疑



MySQL 主从同步



主从同步概述



MySQL 主从同步

- 对指定库的异地同步？
- MySQL主-->从复制架构的实现？
- MySQL服务器的只读控制？

知识讲解



主从同步原理

知识讲解

- Master , 记录数据更改操作
 - 启用 binlog 日志
 - 设置binlog日志格式
 - 设置server_id
- Slave 运行2个线程
 - Slave_IO : 复制master主机 binlog日志文件里的SQL到本机的relay-log文件里。
 - Slave_SQL : 执行本机relay-log文件里的SQL语句 , 重现Master的数据操作。



主从同步原理（续1）



构建主从同步

基本构建思路

知识讲解

- 确保数据相同
 - 从库必须要有主库上的数据。
- 配置主服务器
 - 启用binlog日志、授权用户、查看当前正使用的日志
- 配置从服务器
 - 设置server_id，指定主库信息
- 测试配置
 - 客户端连接主库写入数据，在从库上也能查询到。



确保数据一致（续1）

- Slave 服务器
 - 离线导入由Master提供的备份
 - 清空同名库（若有的话）

```
[root@dbsvr2 ~]# scp dbsvr1:/root/mytest.sql ./
.. ..
```

//直接scp远程拷贝

```
[root@dbsvr2 ~]# mysql -u root -p < mytest.sql
Enter password:
```

//验证口令

配置主服务器

知识讲解

- 调整运行参数
 - 启用binlog及允许同步

```
[root@dbsvr1 mysql]# vim /etc/my.cnf
[mysqld]
log_bin=日志名           //启用binlog日志
server_id = id值         //指定服务器ID号
binlog_format= "mixed"   //指定日志格式
.. ..
[root@dbsvr1 mysql]# systemctl restart mysqld //启服务
```



配置主服务器（续1）

知识讲解

- 授权用户
 - 允许replicater从192.168.4.0/24网段访问
 - 对所有库（默认不允许对单个库）有同步权限

```
[root@dbsvr1 ~]# mysql -u root -p
Enter password:
.. ..
mysql> GRANT REPLICATION SLAVE ON *.* TO
> 用户名@'从库ip地址' IDENTIFIED BY '密码';
```



配置从服务器

- 调整运行参数
 - 指定server_id 不允许与主库server_id值相同

知识讲解

```
[root@dbsvr2 ~]# vim /etc/my.cnf
[mysqld]
server_id = id值                                //指定服务器ID
...
[root@dbsvr2 ~]# systemctl restart mysqld      //启动服务
```



配置从服务器（续1）

- 指定主库信息

知识讲解

```
mysql> CHANGE MASTER TO
-> MASTER_HOST= '192.168.4.10' ,      //主库ip地址
-> MASTER_USER= 'replicater' ,        //主库授权用户名
-> MASTER_PASSWORD= 'pwd123' ,       //授权用户密码
-> MASTER_LOG_FILE='dbsvr1-bin.000004', //日志文件
-> MASTER_LOG_POS=334;                //偏移位置
...
mysql> START SLAVE;                    //启动slave进程
...
```

1. Master信息会自动保存到 /var/lib/mysql/master.info 文件
2. 以后要更改Master信息时，应先 STOP SLAVE;



配置从服务器（续2）

- 查看Slave状态
 - 确认IO线程、SQL线程都已运行

```
mysql> SHOW SLAVE STATUS\G
```

```
***** 1. row *****
```

```
Slave_IO_State: Waiting for master to send event
```

```
Master_Host: 192.168.4.10
```

```
Master_User: replicater
```

```
.. ..
```

```
Slave_IO_Running : Yes
```

```
//IO线程已运行
```

```
Slave_SQL_Running : Yes
```

```
//SQL线程已运行
```

知识讲解



配置从服务器（续3）

- 相关文件

文件名	说明
master.info	主库信息
relay-log.info	中继日志信息
主机名-relay-bin.xxxxxxx	中继日志
主机名-relay-bin.index	索引文件

知识讲解



测试配置

知识讲解

- 在Master上操纵数据
 - 新建newdb库、newtbl表
 - 任意插入几条表记录
- 在Slave上查看数据更改情况
 - 确认新建的newdb库、newtbl表
 - 列出newtbl表的所有记录



案例1：MySQL一主一从

课堂练习

1. 构建 主-->从 复制结构
2. 其中主机192.168.4.10作为主库
3. 主机192.168.4.10作为从库



常用配置选项

主库配置选项

- 适用于Master服务器

选 项	用 途
binlog_do_db=name	设置Master对哪些库记日志
binlog_ignore_db=name	设置Master对哪些库不记日志



从库配置选项

- 适用于Slave服务器

知识讲解

选项	用途
log_slave_updates	记录从库更新，允许链式复制（A-B-C）
relay_log=dbsvr2-relay-bin	指定中继日志文件名
replicate_do_db=mysql	仅复制指定库，其他库将被忽略，此选项可设置多条（省略时复制所有库）
replicate_ignore_db=test	不复制哪些库，其他库将被忽略，ignore-db与do-db只需选用其中一种



主从同步模式

主从同步模式

结构类型

主从同步结构

配置主从从结构

案例拓扑

配置主库

配置第1台从库

配置第2台从库

测试配置

启动 服务

复制模式

复制模式介绍

模式配置

结构类型

主从同步结构

知识讲解

- 基本应用
 - 单向复制：主 --> 从
 - 扩展应用
 - 链式复制：主 --> 从 --> 从
 - 互为主从：主 <--> 主
 - 一主多从：从 <-- 主 --> 从
- ↓
从



配置主从从结构

拓扑结构

- 主从从

Master 服务器
192.168.4.51/24



第2台从库
Slave 服务器
192.168.4.53/24

第1台从库

Slave 服务器
192.168.4.52/24

知识讲解



配置主库

知识讲解

- 1) 用户授权
- 2) 启用binlog日志
- 3) 重启服务

```
mysql> grant replication slave on *.* to 用户名@'从库IP地址'  
identified by '密码';
```

```
# vim /etc/my.cnf  
[mysqld]  
log-bin=日志名  
Server_id=id 号  
binlog_format='mixed'
```

```
# systemctl restart mysqld
```



配置第2台从库

知识讲解

- 1) 修改配置文件 `# vim /etc/my.cnf`
- 2) 指定主库信息 `[mysqld]`
`server_id=id 号`
- 3) 启动slave进程 `# systemctl restart mysqld`
- 4) 查看状态信息

```
mysql> CHANGE MASTER TO MASTER_HOST='第1台从库IP地址',  
-> MASTER_USER='用户名',  
-> MASTER_PASSWORD='密码',  
-> MASTER_LOG_FILE='binlog日志文件名',  
-> MASTER_LOG_POS=偏移量;
```

```
mysql> start slave; //启动slave进程  
mysql> show slave status\G; //检查状态
```



案例2：配置主从同步结构

具体要求如下：

- 配置主机192.168.4.51为主数据库服务器
- 配置主机192.168.4.52为51主机的从库服务器
- 配置主机192.168.4.53为52主机的从库服务器
- 客户端连接主数据库服务器51主机创建的数据，连接52和53主机时，也可以访问到库、表、记录。

课堂练习



复制模式

复制模式介绍

知识讲解

- 异步复制 (Asynchronous replication)
 - 主库执行完一次事务后，立即将结果返给客户端，并不关心从库是否已经接收并处理
- 全同步复制 (Fully synchronous replication)
 - 当主库执行完一次事务，且所有从库都执行了该事务后才返回给客户端
- 半同步复制 (Semisynchronous replication)
 - 介于异步复制和全同步复制之间
 - 主库在执行完一次事务后，等待至少一个从库接收到并写到relay log中才返回给客户端



模式配置

知识讲解

- 查看是否允许动态加载模块
 - 默认允许

```
mysql> show variables like 'have_dynamic_loading';
```

```
mysql> show variables like "have_dynamic_loading";
+-----+-----+
| Variable_name | Value |
+-----+-----+
| have_dynamic_loading | YES |
+-----+-----+
1 row in set (0.00 sec)
```



模式配置 (续1)

- 命令行加载插件
 - 用户需有SUPER权限

知识讲解

```
mysql> INSTALL PLUGIN rpl_semi_sync_master
-> SONAME 'semisync_master.so' ;           //主库上执行
```

```
mysql> INSTALL PLUGIN rpl_semi_sync_slave
-> SONAME 'semisync_slave.so' ;           //从库上执行
```

```
mysql> SELECT PLUGIN_NAME, PLUGIN_STATUS FROM
INFORMATION_SCHEMA.PLUGINS WHERE PLUGIN_NAME LIKE
'%semi%';                                   //查看
```

PLUGIN_NAME	PLUGIN_STATUS
rpl_semi_sync_master	ACTIVE
rpl_semi_sync_slave	ACTIVE



模式配置 (续2)

- 启用半同步复制
 - 在安装完插件后，半同步复制默认是关闭的

知识讲解

```
//主库上执行
```

```
mysql> SET GLOBAL rpl_semi_sync_master_enabled = 1;
```

```
//在从库上执行
```

```
mysql> SET GLOBAL rpl_semi_sync_slave_enabled = 1;
```

```
//查看
```

```
mysql> show variables like 'rpl_semi_sync_ %_enabled';
```

```
mysql> show variables like "rpl_semi_sync_%_enabled";
```

Variable_name	Value
rpl_semi_sync_master_enabled	ON
rpl_semi_sync_slave_enabled	ON



模式配置（续3）

- 永久启用半同步复制
 - 需要修改到主配置文件 /etc/my.cnf
 - 添加相关设置到 [mysqld] 部分

//主库的配置

```
plugin-load=rpl_semi_sync_master=semisync_master.so  
rpl_semi_sync_master_enabled=1
```

//从库的配置

```
plugin-load=rpl_semi_sync_slave=semisync_slave.so  
rpl_semi_sync_slave_enabled=1
```

知识讲解



