

6.MySQL升级与降级–inplace(就地升级)√

6.1 升级方式

不管哪种方式升级，都应该先做备份，方便失败回退

6.1.1 inplace(就地升级)

- 1.在一台服务器上（单节点）原版本升级到新版本风险较大(如果升级没有备份，无法回退)。
- 2.适合于主从环境，先升级从库再升级主库。
- 3.就地升级于数据量无关。

冷备的方法：

- 1.关闭数据库
`/etc/init.d mysqld stop`
- 2.复制数据目录到备份路径下
`cp -r /data/3306/data/ /opt/`

6.1.2 merging(逻辑备份迁移升级)

- 1.适合单机或主从环境
- 2.备份数据会花费大量时间，但是不会影响原生产

6.2 升级的注意事项（inplace）

来自于MySQL官网[官网升级前的注意事项 <https://dev.mysql.com/doc/refman/8.0/en/upgrade-before-you-begin.html>](https://dev.mysql.com/doc/refman/8.0/en/upgrade-before-you-begin.html)

Upgrade is only supported between General Availability (GA) releases.
Upgrade from MySQL 5.6 to 5.7 is supported. Upgrading to the latest release is recommended before upgrading to the next version.
For example, upgrade to the latest MySQL 5.6 release before upgrading to MySQL 5.7.
Upgrade that skips versions is not supported. For example, upgrading directly from MySQL 5.5 to 5.7 is not supported.
Upgrade within a release series is supported. For example, upgrading from MySQL 5.7.x to 5.7.y is supported. Skipping a release is also supported. For example, upgrading from MySQL 5.7.x to 5.7.z is supported.

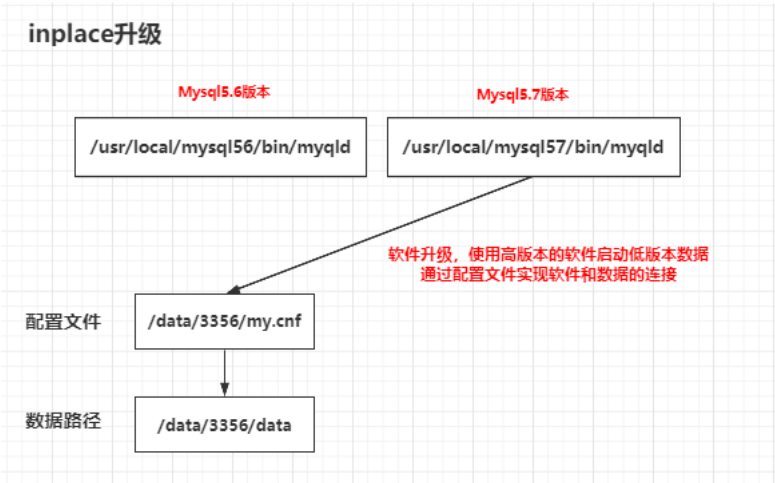
翻译如下：

- a. 支持GA版本之间升级
- b. 5.6--> 5.7 ,先将5.6升级至最新版，再升级到5.7
- c. 5.5 ----> 5.7 ,先将5.5 小版本升级至最新，再5.5----> 5.6最新，再5.6---->5.7 最新
- d. 回退方案要提前考虑好，最好升级前要备份(特别是往8.0版本升级)。
- e. 降低停机时间（停业务的时间），在业务不繁忙时期升级，做好足够的演练

6.3 inplace升级过程原理（生产思路）

升级=软件升级+数据升级

- 0. 准备回退方案，备份原数据库数据。
- 1. 安装新版本软件
- 2. 以优雅的方式关闭原数据库（挂维护页）
- 3. 使用新版本软件“挂”旧版本数据启动(--skip-grant-tables ,--skip-networking)
- 4. 升级 :只是升级系统表。升级时间和数据量无关的。
- 5. 正常重启数据库。
- 6. 验证各项功能是否正常。（大型企业有专业人员测试）
- 7. 业务恢复。



6.4 inplace升级操作

6.4.1 Mysq5.6(5.6.46)版本升级到5.7(5.7.30)版本

0.准备环境

- 0.上传5.6和5.7版本的软件包，并进行解压

```
⚠ Reach fileSize limit mysql-5.6.46-linux-glibc2.12-x86_64.tar.gz ⓘ
⚠ Reach fileSize limit mysql-5.7.30-linux-glibc2.12-x86_64.tar.gz ⓘ
```

tar xf 压缩包名称

- 1.创建mysql用户和数据目录

```
[root@db01 ~]# useradd mysql
[root@db01 ~]# mkdir -p /data/335{6..7}/data
[root@db01 ~]# chown -R mysql. /data
```

- 2.做软连接

```
ln -s /opt/mysql-5.6.46-linux-glibc2.12-x86_64 /usr/local/mysql56
ln -s /opt/mysql-5.7.30-linux-glibc2.12-x86_64 /usr/local/mysql57
3.准备配置文件
cat >/data/3356/my.cnf <<EOF
[mysqld]
user=mysql
basedir=/usr/local/mysql56
datadir=/data/3356/data
socket=/tmp/mysql3356.sock
port=3356
[mysql]
socket=/tmp/mysql3356.sock
EOF
cat >/data/3357/my.cnf <<EOF
[mysqld]
user=mysql
basedir=/usr/local/mysql57
datadir=/data/3357/data
socket=/tmp/mysql3357.sock
port=3357
[mysql]
socket=/tmp/mysql3357.sock
EOF
4.初始化
5.7版本初始化: /usr/local/mysql57/bin/mysqld --defaults-file=/data/3357/my.cnf --initialize-insecure
5.6版本初始化: /usr/local/mysql56/scripts/mysql_install_db --defaults-file=/data/3356/my.cnf --user=mysql --
basedir=/usr/local/mysql56 --datadir=/data/3356/data
5.启动
/usr/local/mysql57/bin/mysqld_safe --defaults-file=/data/3357/my.cnf &
/usr/local/mysql56/bin/mysqld_safe --defaults-file=/data/3356/my.cnf &
6.登陆
mysql -S /tmp/mysql3356.sock
mysql -S /tmp/mysql3357.sock
```

1.优雅的关闭数据库(5.6实例)

默认数据库快速关闭参数是开启的

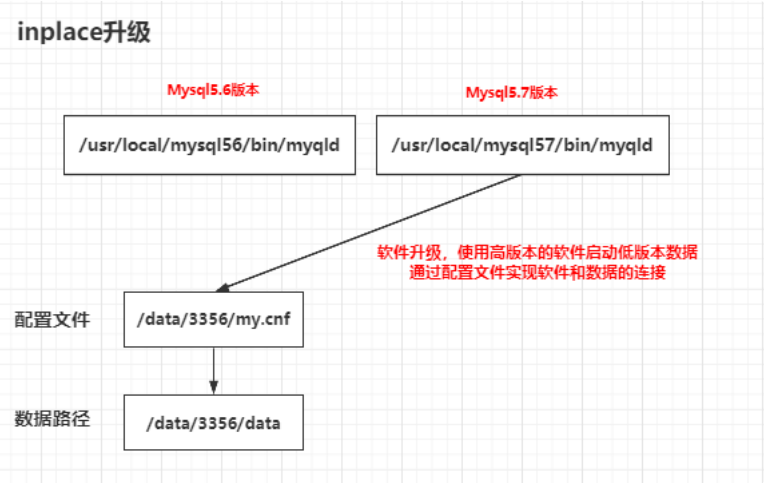
```
mysql> select @@innodb_fast_shutdown;
+-----+
| @@innodb_fast_shutdown |
+-----+
| 1 |
```

+-----+

我们登陆到数据库中将该参数关闭， 或者在配置文件中设置 `innodb_fast_shutdown=0`

```
1.登陆5.6版本的数据库
mysql -S /tmp/mysql3356.sock
2.修改参数
set global innodb_fast_shutdown=0
3.退出数据库， 关闭数据库
/usr/local/mysql56/bin/mysqladmin -S /tmp/3356.sock shutdown
```

2.高版本软件挂低版本软件启动(软件升级)



1.将低版本的配置文件中指定软件的basedir路径改成5.7版本软件的路径

```
[mysqld]
user=mysql
basedir=/usr/local/mysql57
datadir=/data/3356/data
socket=/tmp/mysql3356.sock
port=3356
[mysql]
socket=/tmp/mysql3356.sock
```

2.高版本软件启动低版本配置文件

```
/usr/local/mysql57/bin/mysqld_safe --defaults-file=/data/3356/my.cnf --skip-grant-tables --skip-networking &
高版本软件          指定低版本配置文件          跳过加载授权表          跳过网络连接（不允许别人在升级过程
中连接修改）
```

3.登陆5.6版本的实例发现软件已经升级成为5.7版本，但是数据(系统表)还是5.6未升级

```
mysql -S /tmp/mysql3356.sock
```

```
[root@db01 3356]# mysql -S /tmp/mysql3356.sock
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.7.30 MySQL Community Server (GPL)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

5.6版本系统表数据

test----->sys

5.7版本系统表数据

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| test |
+-----+
4 rows in set (0.03 sec)
```

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
4 rows in set (0.07 sec)
```

3.高版本的软件升级低版本的库（数据升级）

```
/usr/local/mysql57/bin/mysql_upgrade -S /tmp/mysql3356.sock --force
```

高版本软件 mysql_upgrade

低版本的库

强制

```
[root@db01 3356]# /usr/local/mysql57/bin/mysql_upgrade -S /tmp/mysql3356.sock --force
Checking server version.
Running queries to upgrade MySQL server.
Checking system database.
mysql.columns_priv          OK
mysql.db                    OK
mysql.engine_cost           OK
mysql.event                 OK
mysql.func                  OK
mysql.general_log           OK
mysql.gtid_executed         OK
mysql.help_category         OK
mysql.help_keyword          OK
mysql.help_relation         OK
mysql.help_topic            OK
mysql.innodb_index_stats    OK
mysql.innodb_table_stats    OK
mysql.ndb_binlog_index      OK
mysql.plugin                OK
mysql.proc                  OK
mysql.procs_priv            OK
mysql.proxies_priv          OK
mysql.server_cost           OK
mysql.servers               OK
mysql.slave_master_info     OK
mysql.slave_relay_log_info  OK
mysql.slave_worker_info     OK
mysql.slow_log              OK
mysql.tables_priv           OK
mysql.time_zone             OK
mysql.time_zone_leap_second OK
mysql.time_zone_name        OK
mysql.time_zone_transition  OK
mysql.time_zone_transition_type OK
mysql.user                  OK
Upgrading the sys schema.
Checking databases.
sys.sys_config              OK
Upgrade process completed successfully.
Checking if update is needed.
[root@db01 3356]#
```

4.高版本软件关闭升级后的低版本库

```
/usr/local/mysql57/bin/mysqladmin -S /tmp/mysql3356.sock shutdown
```

高版本的软件 mysql_admin 升级后低版本的库 关闭

5.高版本的软件启动升级后的低版本库

```
/usr/local/mysql57/bin/mysqld_safe --defaults-file=/data/3356/my.cnf &
```

6.登陆升级后低版本的库

```
mysql -S /tmp/mysql3356.sock
```

发现数据也已经成功升级拥有5.7版本的sys系统库

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
| test |
+-----+
5 rows in set (0.00 sec)
```

6.4.2 Mysql5.7 (5.7.30) 版本升级到8.0 (8.0.20) 版本

0.首先我们在升级前要对新版本进行一定的了解

8.0版本新特性

- 1.新增支持mysql-shell的工具，可以调用这个命令，做升级之前的预检查
- 2.升级不需要手动去mysql_upgrade
- 3.限制：升级前必须做备份，否则无法回退

1.首先下载mysql8.0.20版本的mysql-shell工具 软件包路径 <<https://downloads.mysql.com/archives/>>

mysql-shell-8.0.20-linux-glibc2.12-x86-64bit.tar.gz (34.5 MB)

- 1.上传解压
- 2.将软件包路径移动重命名到/usr/local/mysqlsh
mv mysql-shell-8.0.20-linux-glibc2.12-x86-64bit /usr/local/mysqlsh
- 3.为这个软件的命令做一个环境变量
vim /etc/profile
export PATH=/usr/local/mysqlsh/bin:\$PATH
source /etc/profile
然后我们就可以直接调用mysqlsh (mysqlsh 这个命令是在/usr/local/mysqlsh/bin下)

```
[root@db01 local]# mysqlsh
MySQL Shell 8.0.20

Copyright (c) 2016, 2020, Oracle and/or its affiliates. All rights reserved.
Oracle is a registered trademark of Oracle Corporation and/or its affiliates.
Other names may be trademarks of their respective owners.

Type '\help' or '? for help; '\quit' to exit.
MySQL JS >
```

4.创建mysql-shell工具的连接用户

```
mysql> create user root@'10.0.0.%' identified with mysql_native_password by '123';
```

```
mysql > grant all on *.* to root@'10.0.0.%';
```

5.使用mysql-shell工具连接需要升级的库进行预检查

```
mysqlsh root:123@10.0.0.51:3357 -e "util.checkForServerUpgrade()" >/tmp/up.log
```

会将检查的结果导入到/tmp/up.log中

3.优雅的关闭数据库(5.7实列)

默认数据库快速关闭参数是开启的

```
mysql> select @@innodb_fast_shutdown;
+-----+
| @@innodb_fast_shutdown |
+-----+
| 1 |
+-----+
```

我们登陆到数据库中将该参数关闭，或者在配置文件中设置 `innodb_fast_shutdown=0`

1.登陆5.6版本的数据库

```
]# mysql -S /tmp/mysql3357.sock
```

2.修改参数

```
mysql> set global innodb_fast_shutdown=0
```

3.mysql5.7版本支持在数据库里面shutdown,关闭数据库

```
mysql> shutdown;
```

4.高版本软件挂低版本数据启动

mysql8.0版本省略了mysql_upgrade，会自动升级。

因为8.0版本我们做了环境变量，所以不用写绝对路径

1.将低版本的配置文件中指定软件的basedir路径改成8.0版本软件的路径

```
[mysqld]
```



```
user=mysql
basedir=/usr/local/mysql
datadir=/data/3357/data
socket=/tmp/mysql3357.sock
port=3357
[mysql]
socket=/tmp/mysql3357.sock
```

2.高版本软件启动低版本配置文件

```
mysqld_safe --defaults-file=/data/3357/my.cnf --skip-grant-tables --skip-networking &
```

高版本软件 指定低版本配置文件 跳过加载授权表 跳过网络连接（不允许别人在升级过程中连接修改）

5.登陆升级后的低版本库查看版本

```
mysql> select version();
+-----+
| version() |
+-----+
| 8.0.20    |
+-----+
1 row in set (0.00 sec)
```

升级成功!!!

6.5 inplace降级操作

6.5.0 降级的介绍

- 1.在MySQL8.0版本之前是可以支持降级操作的，mysql8.0版本不能降级只能使用备份恢复
- 2.如果数据量极大的情况下，恢复数据和回退的时间会很长，所以可以使用降级操作（**停机时间较少**）

6.5.1 Mysql5.7版本降级限制

- 仅在一般可用性（GA）版本之间支持降级。
- 使用**逻辑降级**方法支持从MySQL 5.7降级到5.6。
- 不支持跳过版本的降级。例如，不支持直接从MySQL 5.7降级到5.5。
- 支持在发行系列中进行降级。例如，从MySQL 5.7降级。 **z** 至5.7。 **y** 支持。还支持跳过发行版。例如，从MySQL 5.7降级。 **z** 至5.7。 **x** 支持。

6.5.2 降级前对不同版本特性的了解

随着不同版本的发布，一定会有功能上的差异，会出现差异性。所以我们在做降级的时候要知道各个版本之间的区别。**多删少补**

根据官网说明会有四个地方需要注意 **官方文档** <<https://dev.mysql.com/doc/refman/5.7/en/downgrading-to-previous-series.html>>

1.系统表的变更

我们从低版本到高版本的顺序记录版本之间的变化，如果降级到那个版本需要和这个版本的特性保持一致即可。

系统表中plugin列定义变更

Mysql5.6.23到5.6.的最高版本 : `ALTER TABLE mysql.user MODIFY plugin CHAR(64) COLLATE utf8_bin DEFAULT 'mysql_native_password';`

Mysql5.6.22以及之前的版本: `ALTER TABLE mysql.user MODIFY plugin CHAR(64) COLLATE utf8_bin DEFAULT '';`

`help_*` 和 `time_zone*` 系统表更改 `MyISAM` 到 `InnoDB`

Mysql5.7.5

系统表的 `Password` 列 `mysql.user`删除

Mysql 5.7.6

`sys` 默认情况下会在数据目录安装过程中安装,建议删除sys

Mysql5.7.7

MySQL用户名的最大长度从16个字符增加到32个字符

Mysql5.7.8

存储user @ host字符串值的系统表列的长度增加到77个字符

Mysql5.7.13

2.innodb的变更

重要的是我们降级时,避免版本差异性，使用优雅关机 `innodb_fast_shutdown=0`

3.记录变更

4.SQL变更

6.5.3 MySQL 5.7.30 TO 5.7.10 inplace downgrade演练

0.环境准备，上传解压5.7.30和5.7.10的二进制软件包

1.为软件做软连接

```
ln -s /opt/mysql-5.7.10-linux-glibc2.12-x86_64 /usr/local/mysql5710
ln -s /opt/mysql-5.7.30-linux-glibc2.12-x86_64 /usr/local/mysql57
```

2.因为Mysql5.7.13存储user @ host字符串值的系统表列的长度增加到77个字符，我们降级到13更前的10版本需要修改字符长度

我们连接到原版本5.7.30数据库进行字符长度的修改

```
mysql -S /tmp/mysql3357.sock
官网提供的修改sql语句
set sql_mode='STRICT_TRANS_TABLES,ERROR_FOR_DIVISION_BY_ZERO,NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION' ;
set global
sql_mode='STRICT_TRANS_TABLES,ERROR_FOR_DIVISION_BY_ZERO,NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION' ;
ALTER TABLE mysql.proc MODIFY definer char(77) CHARACTER SET utf8 COLLATE utf8_bin NOT NULL DEFAULT '';

ALTER TABLE mysql.event MODIFY definer char(77) CHARACTER SET utf8 COLLATE utf8_bin NOT NULL DEFAULT '';

ALTER TABLE mysql.tables_priv MODIFY Grantor char(77) COLLATE utf8_bin NOT NULL DEFAULT '';
```

```
ALTER TABLE mysql.procs_priv MODIFY Grantor char(77) COLLATE utf8_bin NOT NULL DEFAULT '';
```

3.实例5.7.30进行优雅的关闭

```
mysql> set global innodb_fast_shutdown=0;
mysql> shutdown;
```

4.删除5.7.30数据目录下的redo日志，让redo日志重新生成

```
rm -rf /data/3357/data/ib_logfile*
```

5.修改被降级的配置文件（5.7.30），指定降级目标版本软件（5.7.10）

```
[mysqld]
user=mysql
basedir=/usr/local/mysql5710
datadir=/data/3357/data
socket=/tmp/mysql3357.sock
port=3357

[mysql]
socket=/tmp/mysql3357.sock
```

6.指定降级目标版本软件（5.7.10）挂起被降级的5.7.30（软件降级）

```
/usr/local/mysql5710/bin/mysqld_safe --defaults-file=/data/3357/my.cnf --skip-grant-tables --skip-networking &
5.7.10版本
```

7.指定降级目标版本软件（5.7.10）降级操作被降级的5.7.30（数据降级）

```
/usr/local/mysql5710/bin/mysql_upgrade -S /tmp/mysql3357.sock
5.7.10版本的mysql_upgrade          指定5.7.30版本的本地连接
```

inplace(%E5%B0%B1%E5%9C%B0%E5%8D%87%E7%BA%A7)%E2%88%9A%20%7C%206.1%20%E5%8D%87%E7%BA%A7%E6%96%B9%E5%BC%8F%E4%B8%8D%E7%AE%A1%E5%93%AA%E7%A7%8C