

CN_Redundant_MySQL_如何配置冗余 runtime 系统的 MySQL replication

Exported from Confluence on 2024 January 26

Table of Contents

需求 3

MySQL replication 相关详细配置步骤 3

需求

需求

客户项目配置是冗余的 **runtime** 系统，项目中有对相关数据的记录要求，涉及到 **MySQL** 数据库（原为 **MySQL** 数据库，在被 Oracle 公司收购后变为付费软件；**MariaDB** 为和 **MySQL** 兼容的免费软件，为了和之前的 **APROL** 版本统一，现在还是统称 **MySQL** 数据库），这就需要对冗余系统的 **MySQL** 数据库进行相应的复制，这部分配置工作需要在 **AprolConfig** 里面进行。因为 **MySQL replication** 的配置涉及到相关证书生成、传送等比较繁琐的内容，下面就具体的整个流程做一个详细的说明。

注意，非冗余 **runtime** 系统项目是无需配置 **MySQL replication** 的。

MySQL replication 相关详细配置步骤

具体步骤

有关冗余 **runtime** 系统的 **MySQL replication** 机制这里不做描述，下面只说明具体的配置过程，以 **APROL R4.2-05** 为例。（不同版本配置稍微有些区别）

1. 冗余 runtime 的主、从机的 ssh 配置

在进行 **AprolConfig** 里面的 **MySQL replication** 配置之前，需要先对冗余 **runtime** 的主、从机之间的 **ssh** 相互访问进行必要的密钥生成和分发工作。

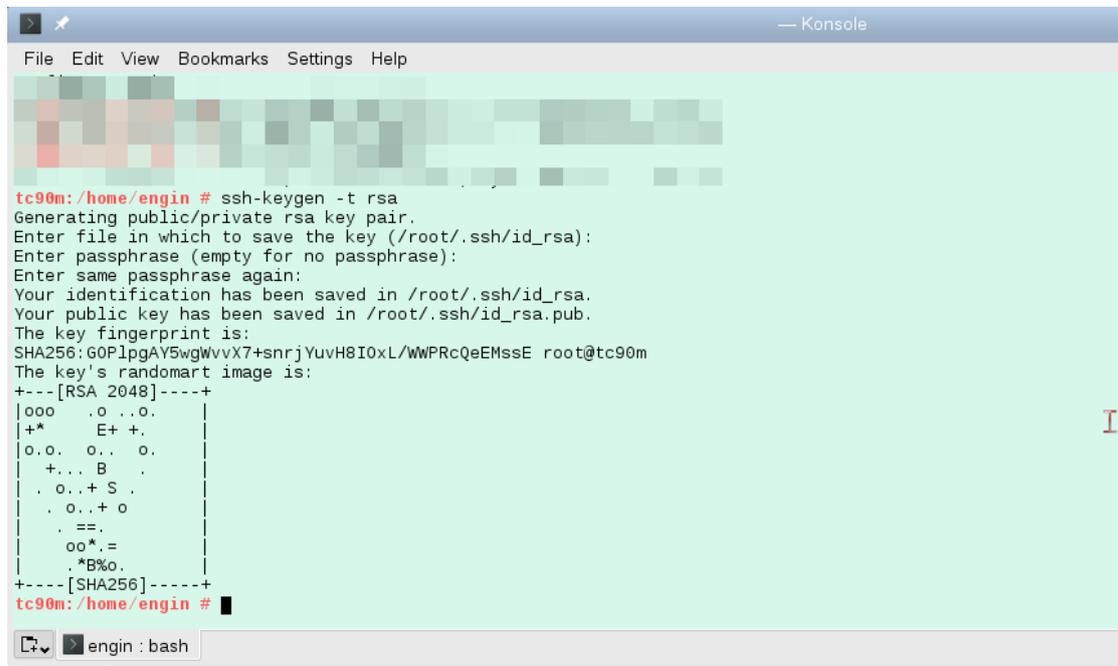
冗余 runtime 系统的主机的密钥生成

首先需要在冗余 **runtime** 的主机上进行 **ssh** 访问的密钥的生成。

具体步骤如下：

- 1) 打开 **konsole**，输入命令 `su - root` 回车，按提示输入 **root** 的密码并回车。
- 2) 继续在当前 **konsole** 里面输入命令 `ssh-keygen -t rsa` 并回车。
- 3) 在接下来的询问存储密钥的文件时，直接回车即可（默认保存在 `/root/.ssh/id_rsa`）
- 4) 在询问设置密码时，同样直接按回车即可；再继续输入确认密码时也是继续按回车。
- 5) 这样，一对密钥就创建完成，并存储在 `/root/.ssh/id_rsa.pub` 里面。

下面为执行 **ssh-keygen** 的一个示例图。



```
tc90m:/home/engin # ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:G0PlpgAY5wgWvvX7+snrjYuvH8I0xL/WWPRcQeEMssE root@tc90m
The key's randomart image is:
+---[RSA 2048]-----+
|ooo .o..o.          |
|+*   E+ +          |
|o.o. o.. o.        |
|+... B .           |
|.o..+ S .          |
|.o..+ o            |
|.==.               |
|.oo*.,=            |
|. *B%o.            |
+---[SHA256]-----+
tc90m:/home/engin #
```

冗余 runtime 的主、从机的密钥分发

接下来的下面执行的步骤，在冗余 runtime 的主机和从机上都需要做同样的操作！

具体步骤如下：

- 1) 打开 **konsole**，输入命令 **su - root** 回车，按提示输入 **root** 的密码并回车。
- 2) 检查是否存在 **/root/.ssh** 这个文件夹，如不存在，使用 **mkdir** 手动创建 **/root/.ssh** 这个文件夹。
- 3) 把 **/root/.ssh** 这个文件夹的权限修改为 **0700**。
- 4) 使用 **cp** 或 **cat** 命令把 **id_rsa.pub** 的内容追加到 **/root/.ssh/authorized_keys** 文件里。
- 5) 修改 **authorized_keys** 文件的权限为 **0600**。

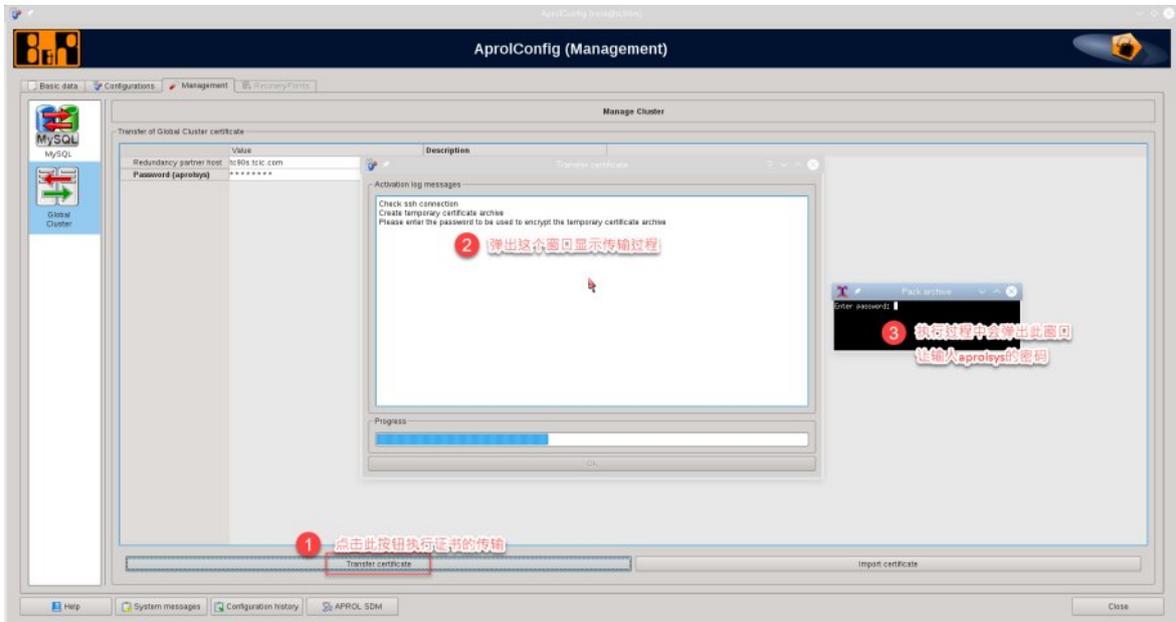
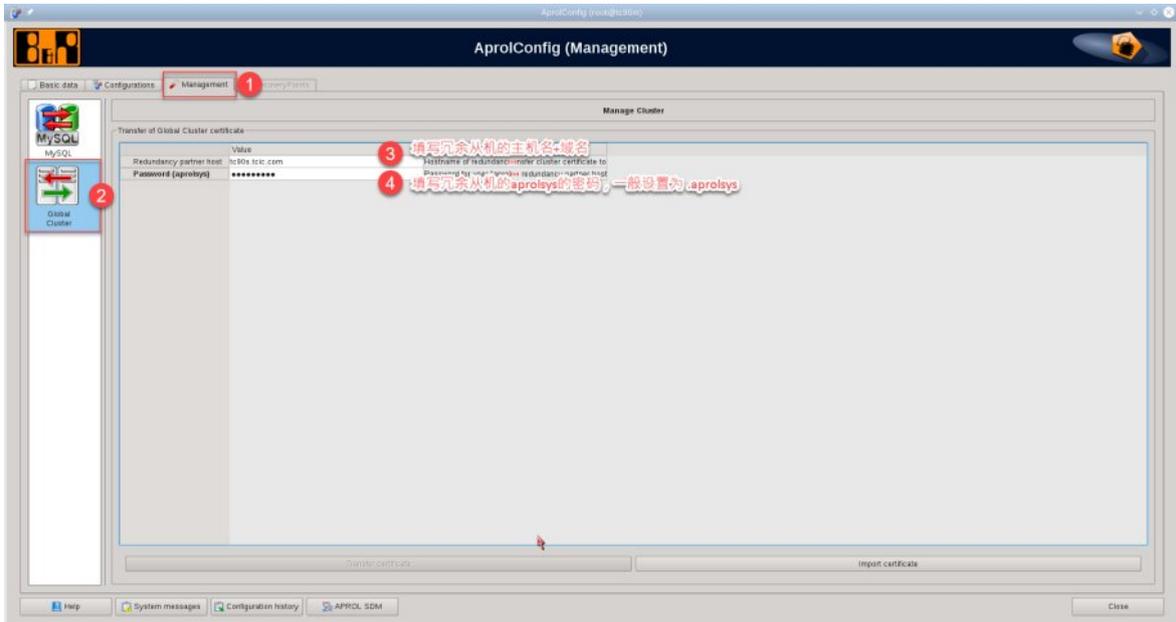
下图为在冗余 runtime 的主机上执行步骤的过程截图。

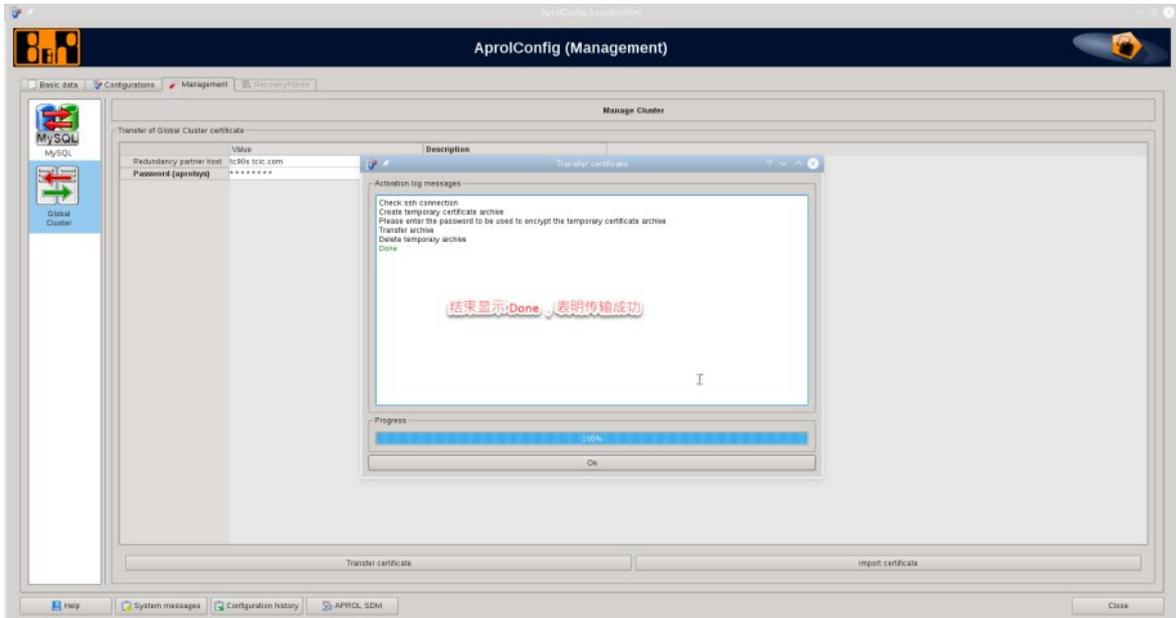
```
Apr0lConfig (Management)
-- K0ns0le
File Edit View Bookmarks Settings Help
-rw-r--r-- 1 root root 26 Nov 27 09:09 dmrc
drwxr-xr-x 2 root root 6 Nov 26 16:52 Documents
drwxr-xr-x 2 root root 6 Nov 26 16:52 Downloads
drwx----- 3 root root 181 Nov 26 16:52 .gnupg
drwxr-xr-x 2 root root 6 Nov 27 09:09 .gtk-3.0
-rw-r--r-- 1 root root 495 Nov 27 09:09 .gtkrc-2.0
drwxr-xr-x 4 root root 48 Nov 27 09:07 inst-sys
drwxr-xr-x 2 root root 24 Nov 26 16:48 kbd
drwx----- 2 root root 25 Nov 27 15:01 kde4
drwxr-xr-x 3 root root 19 Nov 27 09:09 .local
drwxr-xr-x 3 root root 21 Nov 27 09:07 .mozilla
drwxr-xr-x 2 root root 6 Nov 26 16:52 Music
-rw-r--r-- 1 root root 47 Nov 27 09:27 .my.cnf
drwxr-xr-x 3 root root 17 Nov 27 09:12 (null)
drwxr-xr-x 2 root root 6 Nov 26 16:52 Pictures
-rw-r--r-- 1 root root 1102 Nov 27 09:47 profile
-rwxr-xr-x 1 root root 998 Nov 27 09:47 .profile
drwxr-xr-x 2 root root 6 Nov 26 16:52 Public
drwx----- 2 root root 43 Nov 26 17:00 .putty
-rw----- 1 root root 1024 Nov 27 09:30 rnd
drwx----- 2 root root 38 Nov 27 15:17 .ssh
drwxr-xr-x 3 root root 6 Nov 27 09:09 .sftp
drwxr-xr-x 2 root root 6 Nov 26 16:52 Templates
drwxr-xr-x 2 root root 31 Nov 27 08:34 tmp
drwxr-xr-x 2 root root 6 Nov 26 16:52 Videos
-rw----- 1 root root 50 Nov 27 15:01 .xauthwmemb
-rw----- 1 root root 207 Nov 27 09:12 .xauthority
-rw----- 1 root root 50 Nov 27 10:05 .xauthouY6v
drwxr-xr-x 2 root root 34 Nov 26 16:53 .xdg.menu_cache
-rwxr-xr-x 1 root root 1116 Nov 27 09:09 .xserverrc
-rw----- 1 root root 0 Nov 26 16:52 .xsession-errors
-rw----- 1 root root 1606 Nov 27 09:26 .xsession-errors-0
tc90n:~ # chmod 0700 .ssh
tc90n:~ # cd .ssh/
tc90n:~/.ssh # ll
total 8
-rw-r----- 1 root root 1679 Nov 27 15:17 id_rsa
-rw-r----- 1 root root 392 Nov 27 15:17 id_rsa.pub
tc90n:~/.ssh # cp id_rsa.pub authorized_keys
tc90n:~/.ssh # ll
total 12
-rw-r----- 1 root root 392 Nov 27 15:20 authorized_keys
-rw-r----- 1 root root 1679 Nov 27 15:17 id_rsa
-rw-r----- 1 root root 392 Nov 27 15:17 id_rsa.pub
tc90n:~/.ssh # chmod 0600 authorized_keys
tc90n:~/.ssh # ll
total 12
-rw-r----- 1 root root 392 Nov 27 15:20 authorized_keys
-rw-r----- 1 root root 1679 Nov 27 15:17 id_rsa
-rw-r----- 1 root root 392 Nov 27 15:17 id_rsa.pub
tc90n:~/.ssh #
```

在冗余 runtime 的从机上执行步骤也是一样，只是需要事先把冗余主机上的 id_rsa.pub 文件通过U盘拷贝或者 FTP 方式拷贝到冗余从机的某文件夹先存放，之后再拷贝到 /root/.ssh 这个文件夹里。另外，冗余 runtime 的从机上 /root/.ssh 这个文件夹应该是手动创建的。同样参考上图，这里就不上图了。

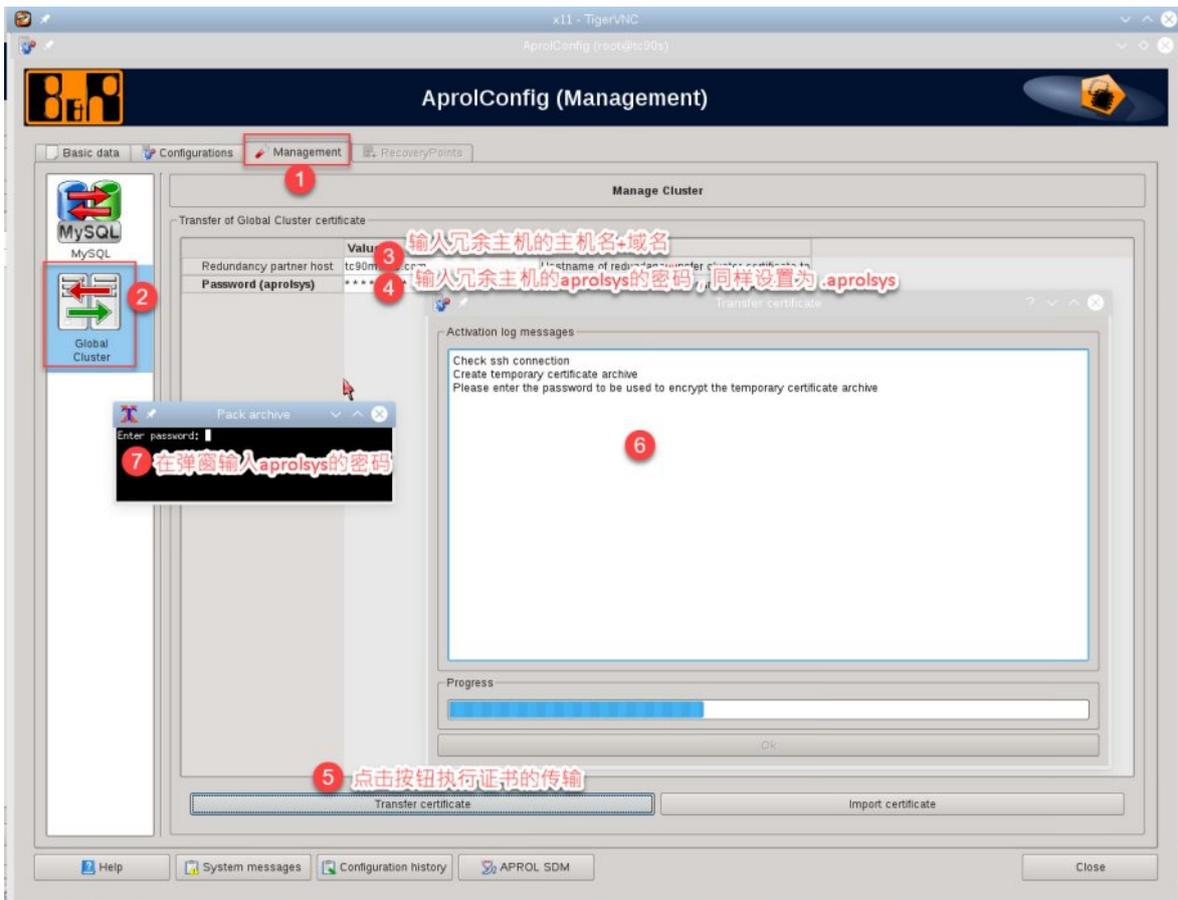
2. 冗余 runtime 系统主、从机执行 Global Cluster 的配置

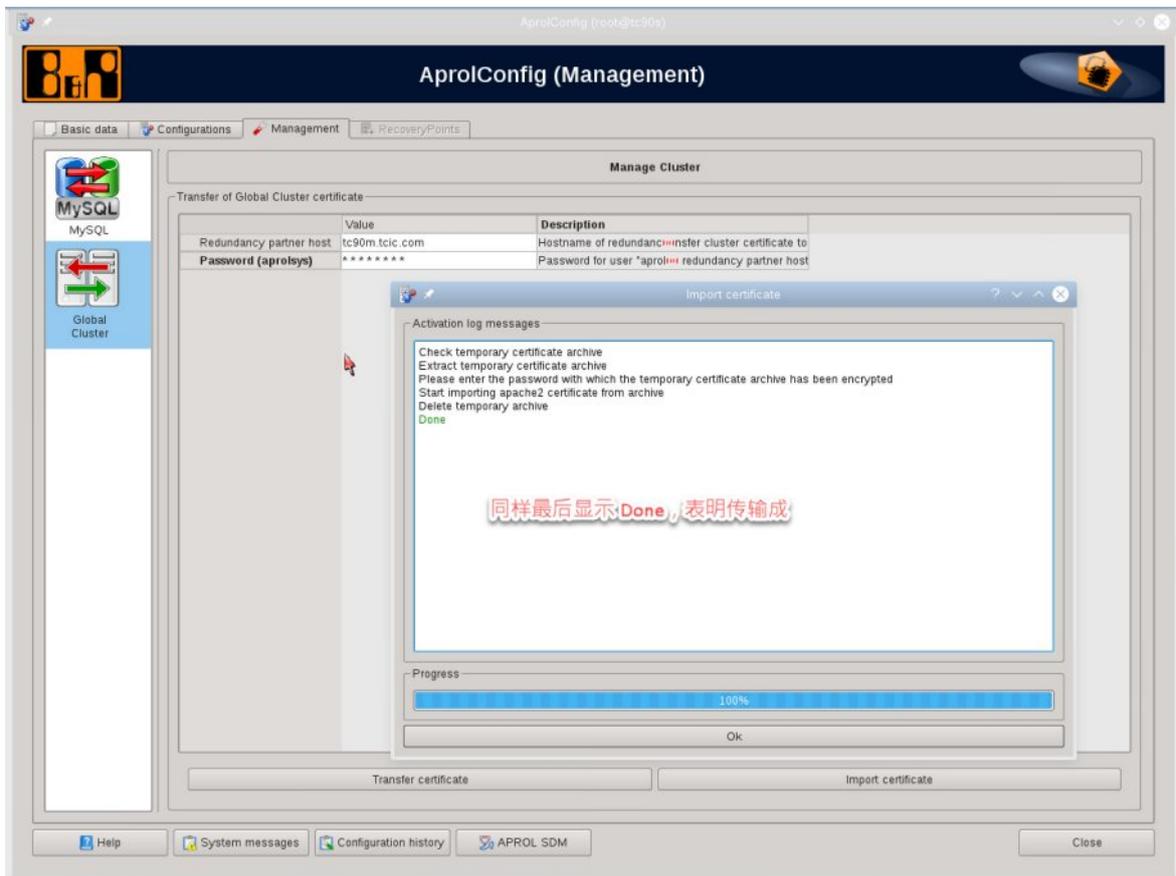
先在冗余 runtime 的主机上打开 Apr0lConfig 先开始 Global Cluster 的配置，具体如下面的三张图。





接下来，在冗余 runtime 的从机上打开 AprolConfig 同样进行 Global Cluster 的配置，具体如下面的二张图。



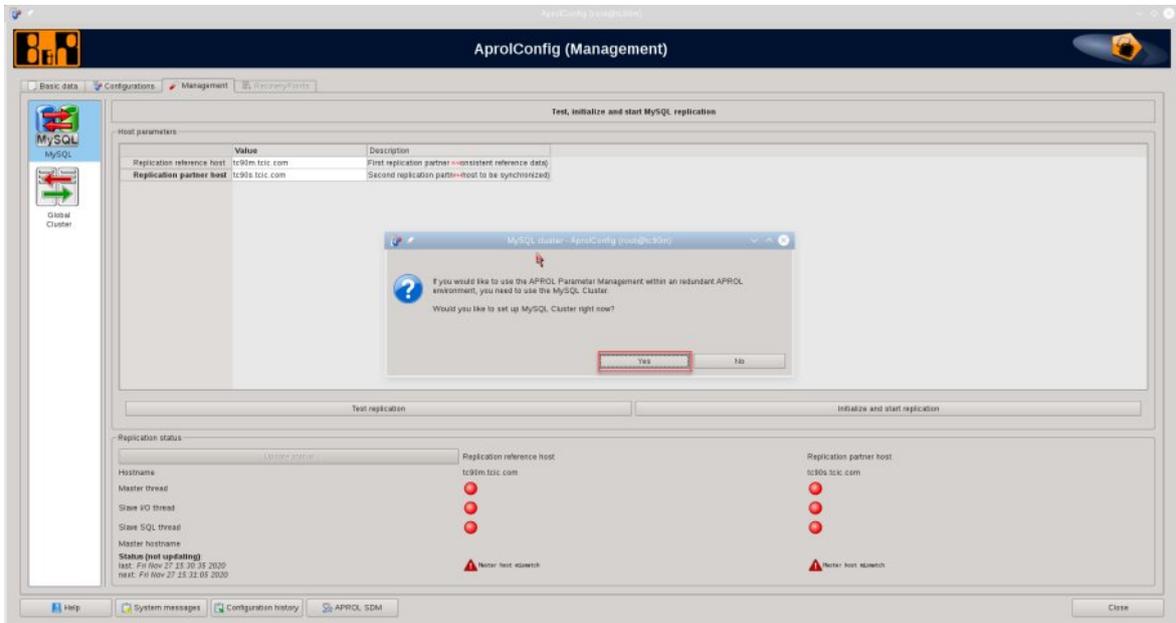
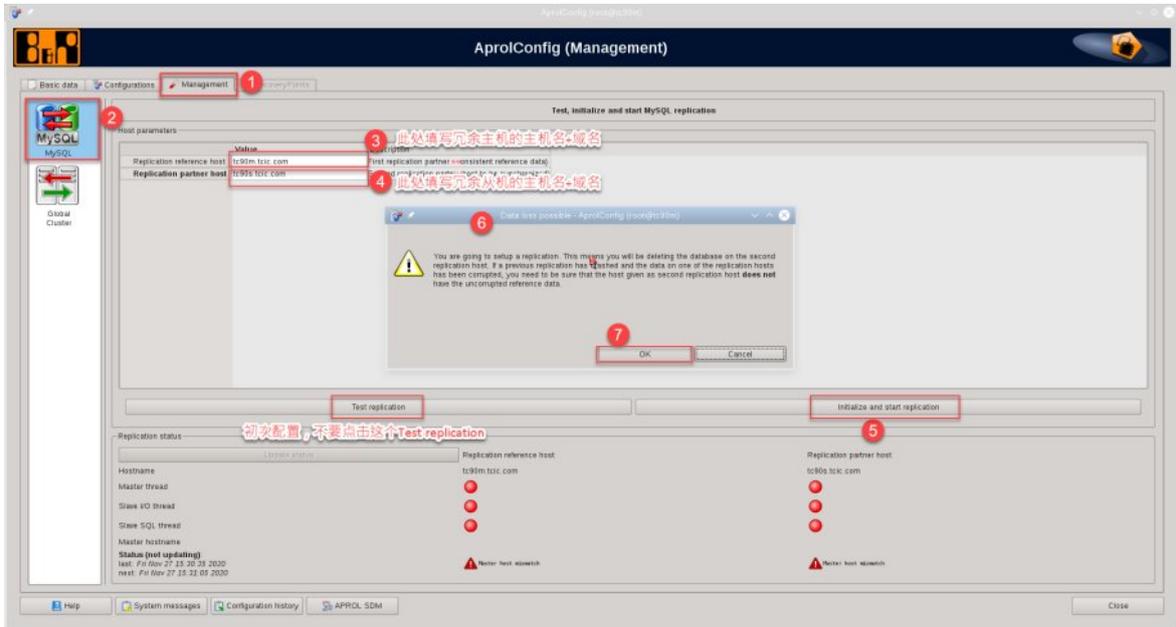


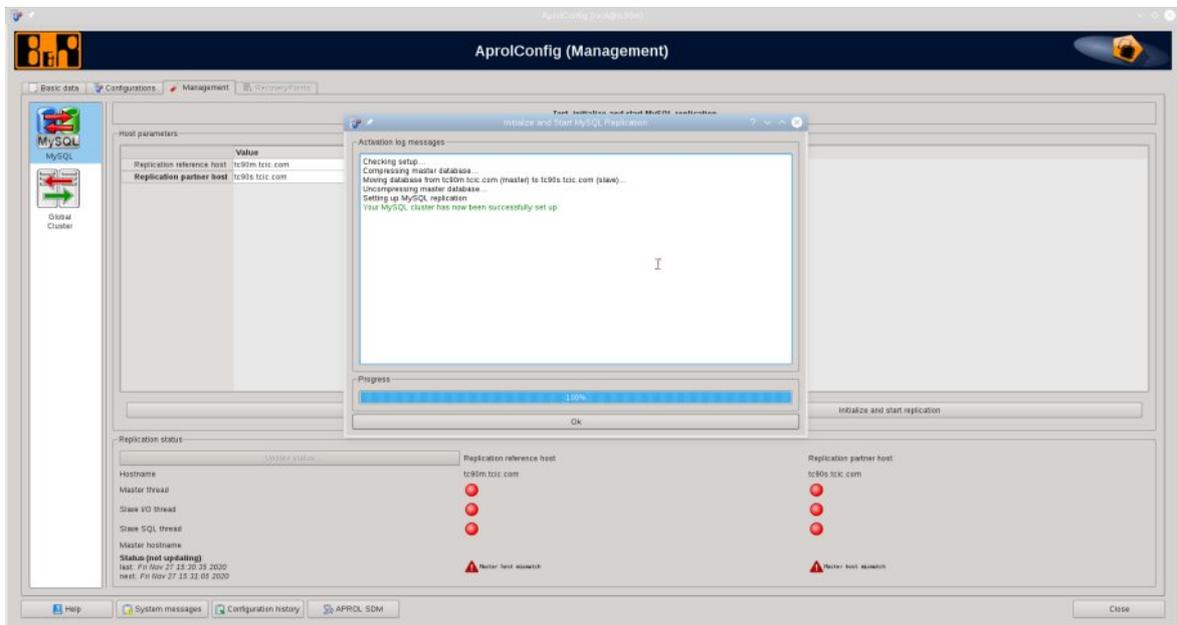
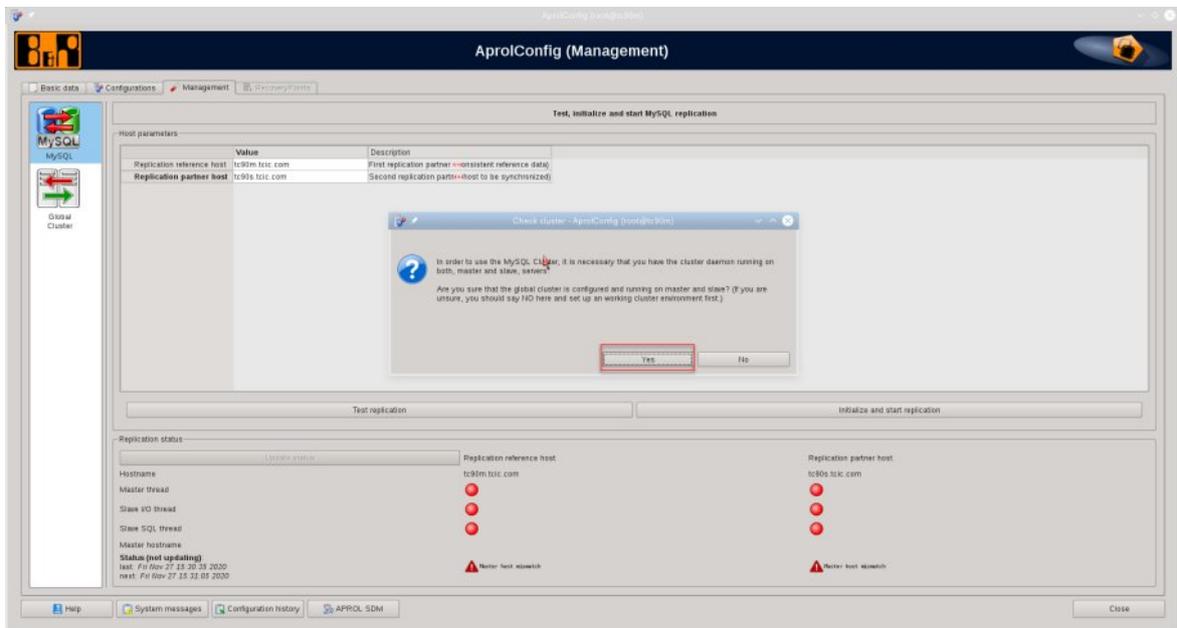
⚠ 注意事项

在 APROL R4.2-06P10 版本，Global Cluster 的配置已经被简化了，无需执行这步配置工作。

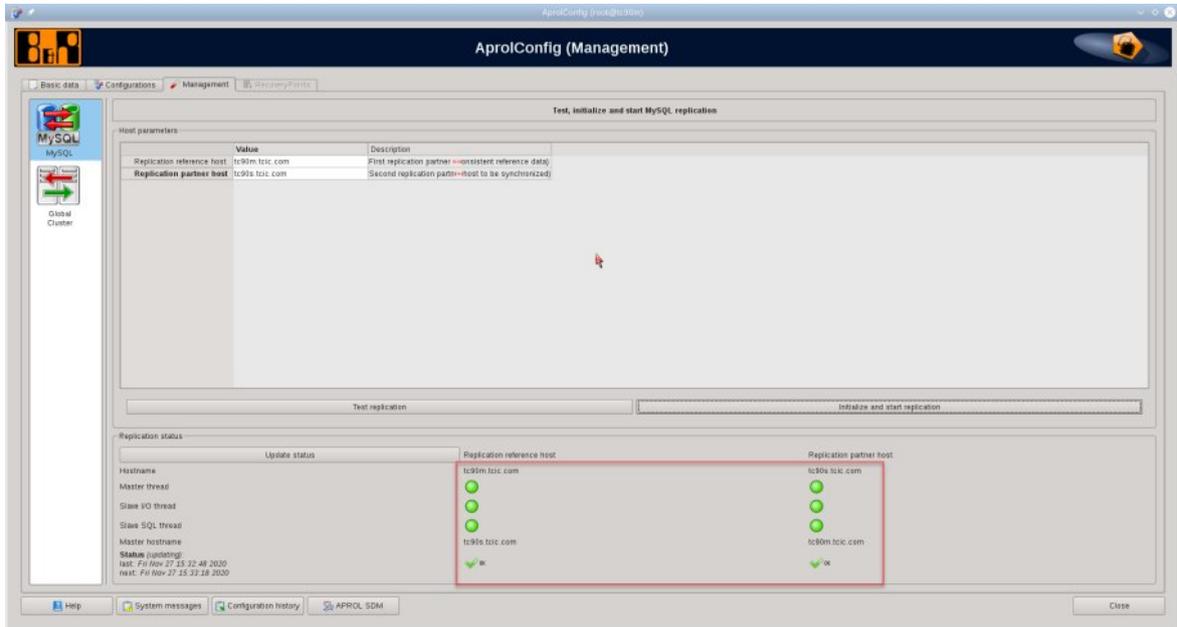
3. 冗余 runtime 系统主机上执行 MySQL replication 配置

最后，还是在冗余 runtime 的主机上，打开 AproxConfig，如下图所示步骤进行相应的设置，然后点击“Initialize and start replication”进行初始化和数据库复制，对后续弹出的询问窗口，均选择“Yes”即可。





最后，MySQL replication 成功的标志就是如下图圈选的这些状态都是绿色的圈圈。



⚠ 注意事项

注意，在冗余 runtime 的主机上进行了 MySQL replication 的配置后，不需要在冗余 runtime 的从机上执行同样的配置了！

4. MySQL 复制状态的监测方法

对于已经配置好的冗余 runtime 系统的主、从机的 MySQL replication 的状态，可以有两种不同的方法来查看，一种是通过命令 **BuMySQLToolbox** 命令来查看，另一种是调用 **SysMon** 库的 **MonAprSysRunRed01** 超级宏来实现。推荐调用 **SysMon** 库的 **MonAprSysRunRed01** 超级宏的方法。

方法一：使用 BuMySQLToolbox 命令

通过命令 **BuMySQLToolbox** 命令来查看冗余 runtime 系统的主、从机的 MySQL replication 的状态，如下图所示为查看冗余 runtime 系统的主、从机的状态的结果。

```
[...] show_log : Start system report
[-]-help, [-]-h, [-]-? : Show this help, then exit
cjhwm:~/.ssh # BuMySQLToolbox -show_master_status -hostname cjhwm
BuMySQLToolbox (MySQL Toolbox, Version 3.1.24)
APROL R 4.2-05 (22.01.2019), (64Bit),
© 1995-2020 B&R Industrial Automation GmbH
#++LOGFILE: mysql-bin.000002
#++LOGPOS: 312
success!
cjhwm:~/.ssh # BuMySQLToolbox -show_slave_status -hostname cjhws
BuMySQLToolbox (MySQL Toolbox, Version 3.1.24)
APROL R 4.2-05 (22.01.2019), (64Bit),
© 1995-2020 B&R Industrial Automation GmbH
#++SLAVEIOSTATE: Waiting for master to send event
#++MASTERHOST: 192.168.5.31
#++MASTERUSER: replicant
#++MASTERPORT: 3306
#++MASTERLOGFILE: mysql-bin.000002
#++MASTERLOGPOS: 312
#++RELAYLOGFILE: mysqld-relay-bin.000003
#++RELAYLOGPOS: 599
#++SLAVEIORUNNING: Yes
#++SLAVESQLRUNNING: Yes
#++LASTERRNO: 0
#++LASTERROR:
#++LASTIOERRNO: 0
#++LASTIOERROR:
#++LASTSQLERRNO: 0
#++LASTSQLERROR:
success!
cjhwm:~/.ssh # █
```

其中，有关冗余 runtime 主机的状态的说明如下图所示。

BuMySQLToolbox -show_master_status -hostname <Hostname>

Output	Explanation
#++LOGFILE: mysql-bin.000002	Current bin log file in which the data to be replicated is entered.
#++LOGPOS:230	Position of the last written entry in the bin log file
Success!	Reading of the master status successful

有关冗余 runtime 从机的状态的说明如下图所示。

BuRMySQLToolbox -show_slave_status -hostname <Hostname>

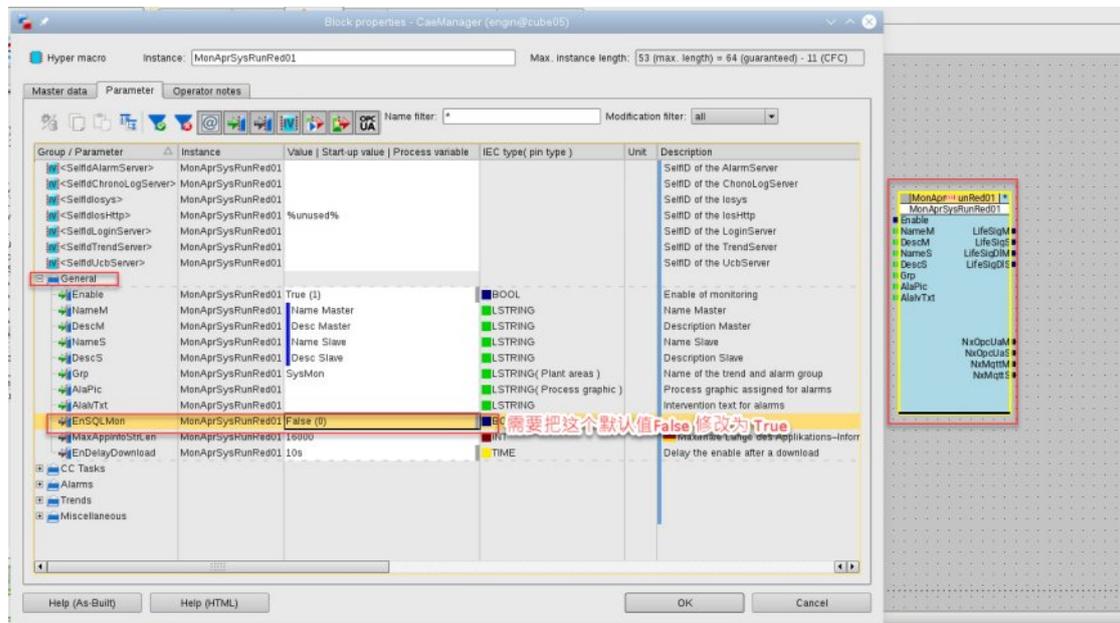
Output	Explanation
###MASTERHOST: redutest1	Name of the replication partner
###MASTERUSER: replicant	Name of the replication user in the MySQL database (assigned by the system)
###MASTERLOGFILE: mysql-bin.000002	Name of the bin log file from which the data is currently being read.
###MASTERLOGPOS: 230	Position of the last entry read from the bin log file
###SLAVEIORUNNING: YES	Status of the slave I/O thread, YES=started and successfully connected to the master, NO=not started or not connected to the master, CONNECTING=Connection setup with the master
###SLAVESQLRUNNING: YES	Status of the slave SQL thread, YES=started, NO=not started
###LASTERRNO: 0	Last error, 0=no error
###LASTERROR:	Last error text, <Empty string>=no error
Success!	Reading slave status successful

使用命令 **BuRMySQLToolbox** 命令来查看冗余 runtime 系统的主、从机的 MySQL replication 的状态的方法是简单，随时可以查看，但对最终用户并不友好，最终用户并不了解该命令以及如何使用。

方法二：CFC 调用 SysMon 库 MonAprSysRunRed01 超级宏

另一种查看冗余 runtime 系统的主、从机的 MySQL replication 的状态，是调用 SysMon 库的 **MonAprSysRunRed01** 超级宏来实现。

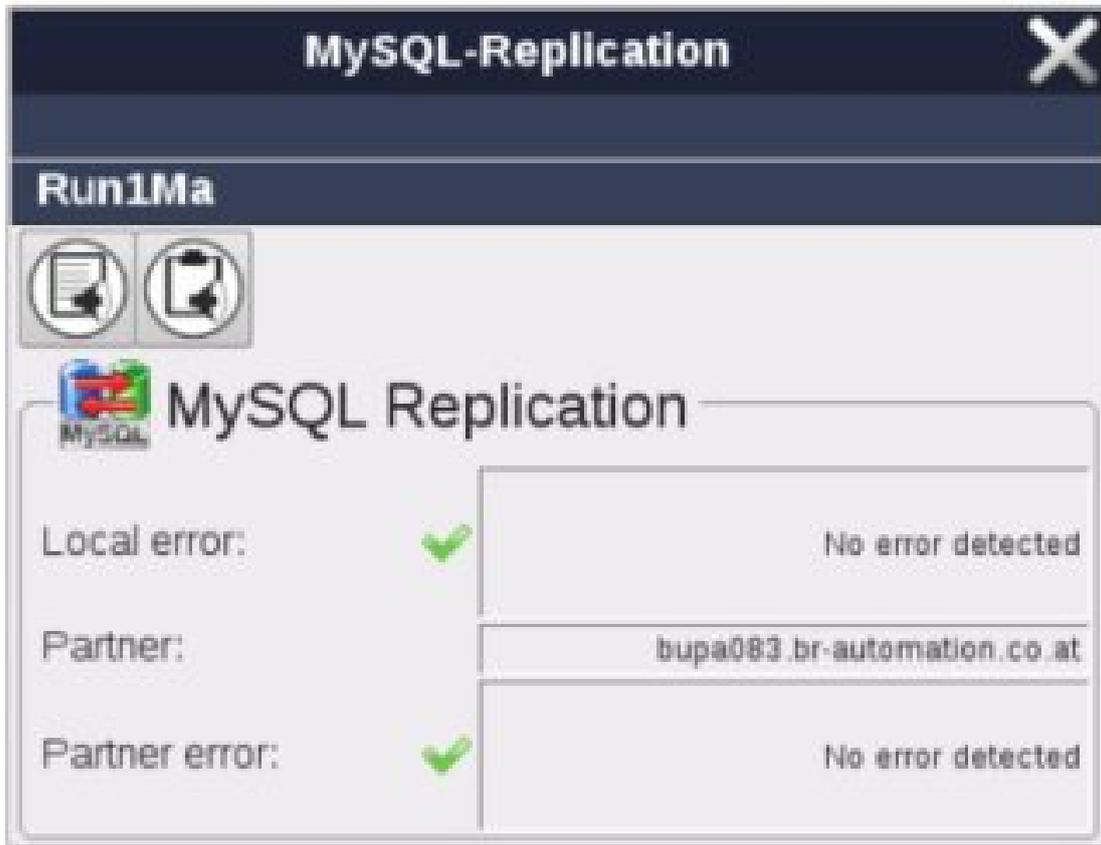
即在一个只运行在 runtime 上的 CFC 里面，直接调用 SysMon 库的 **MonAprSysRunRed01** 的超级宏，注意在超级宏的属性里面要如图对 General 下的 EnSQLMon 这个参数的默认值修改为 True 才行，如下图。



这样，在运行画面里面就可以得到如下图所示的操作面板。



点击上图的 MySQL-Replication 后会得到如下图类似的操作面板显示 **MySQL replication** 的状态。



使用 **MonAprSysRunRed01** 超级宏来查看冗余 **runtime** 系统的主、从机的 **MySQL replication** 的状态的方法是需要调用这个超级宏并配置，但对最终用户非常友好，需要查看状态时只要点击开来就行，并且相关的报警等都记录在历史报警里面。