跨级建云 – Mysql灾备

1. 概述
2. Mysql相关知识
   1. Mysql复制支持一从多主，在从上通过增加复制通道来实现，前提是需要分库来避免数据多写冲突。
   2. Mysql用户可以对不同数据库分别控制权限(对旧连接不生效，需要清理连接到指定库上的旧连接)。
   3. Mysql可以根据需要设置复制时候的过滤规则(规律规则是全局的，不能对通道单独设置，因此建议meeting起一个单独的Mysql实例，避免波及到其他库)。
3. 实际产生灾备场景
   1. 断电
   2. 路由器故障、网络中断
   3. 拔板卡(管理板卡、资源板卡)
4. 方案设计
5. 模型

从

业务

主

从

主平台域A(正常使用)

灾备平台域B(正常使用+未接管)

主

meeting-B

meeting-A

meeting-A

meeting-A

meeting-B

meeting-A

业务

**主平台域A正常工作**

复制

可读写

禁止读写

从

业务

主

从

主平台域A(故障)

灾备平台域B(正常使用+接管)

主

meeting-B

meeting-A

meeting-A

meeting-A

meeting-B

meeting-A

业务

**灾备平台域B接管后**

复制

可读写

禁止读写

* 1. 灾备平台域为每个需要灾备的目标平台域创建对应的数据库(meeting-moid)。
  2. 部署时按照灾备关系和平台域主备关系搭建复制架构，这个复制关系搭建后不再变化，除非是部署关系发生改变。
  3. 主平台域复制灾备平台域数据时，应该增加过滤功能，只复制自己平台域的数据库数据。
  4. 根据平台域内虚IP状态来决定读写权限，没有虚IP的主机不可写，有虚IP的主机在保证同步完对方数据后可读写。---这个对keepalived依赖比较大
  5. PMS根据灾备平台域是否启用和数据同步是否完成来决定开启和关闭指定平台域数据的读写权限，具体控制下面有说明。

说明：上面的a,b,c部分应由部署完成，e由PMS来完成，d部分之前是部署脚本保证的，建议也交给PMS处理吧，毕竟逻辑是极度相似的。

1. PMS控制策略(平台域内主备状态切换)
   1. 主--->备：将mysql设为不可读写。
   2. 备--->主：判断和对方的数据同步是否已经结束，如果结束，则开启mysql访问权限。否则重新判断，直至超时。
2. PMS控制策略(主平台域是否可用)
   1. 可用--->不可用：将mysql设为不可读写。
   2. 不可用--->可用：判断和对方的数据同步是否已经结束，如果结束，则开启mysql访问权限。否则重新判断，直至超时。---这个判断有难度，可能很难精确判断，因为灾备平台域本身也可能在写入数据。
3. PMS控制策略(灾备平台域是否接管)
   1. 灾备未接管--->灾备接管：判断和对方的数据同步是否结束，如果结束，则开启mysql访问权限。否则重新判断，直至超时。
   2. 灾备接管--->灾备接管结束：将mysql设为不可读写。
4. 模块波及

|  |  |
| --- | --- |
| **模块** | **波及** |
| CMC | 根据用户域所属平台域使用Mysql |
| PMS | 控制平台域内主备切换流程  控制灾备处理流程  控制各个Mysql服务器的读写权限 |
| 部署 | 根据灾备情况启动备Mysql服务器  根据部署情况创建相关的数据库，并配置初始用户权限  根据部署情况配置Mysql复制参数，并搭建复制的初始环境 |

1. 关于方案的补充说明：
2. 数据同步可能出错的几种场景：
   1. 主平台域本身的主备切换可能导致复制出错：主机由于某种原因不能访问，备机等待同步状态超时后强制切为主。----可能影响主平台域，其灾备平台域的数据同步也可能受影响从而出错。
   2. 灾备发生后，灾备平台域和主平台域失去联系，数据同步未完成，等待超时后灾备平台域接管了主平台域数据。
   3. 灾备平台域接管后发生主从强制切换也可能导致复制失败。
   4. 灾备结束时，由于主平台域和灾备平台域失联，或者难精确判断主平台域和灾备平台域同步是否完成(难点在于灾备平台域仍然能写入其本身平台域数据)等原因，也可能出现同步出错。
3. 数据同步异常后处理方式建议:
   1. 错误发生在备机(当前mysql未提供读写服务):

对某些可控的错误进行忽略跳过，对于严重的错误需要做一次全同步解决。

* 1. 错误发生在主机(当前正在正常使用)：

建议的处理方案是保持当前的主从关系不动，在主上跳过失败的同步操作继续执行直至同步完对方操作(这么做是为了最大程度保证数据少丢失)，然后把对应从机和当前的主机再做一次全同步，防止后续切换导致错误蔓延。

* 1. 如果a，b同时发生，应该先处理b，再处理a，避免从机数据丢失。