**合服说明：**

1. **修改service/merger/defines.lua，依次迭代下去。以及shell/merger\_pulldata.py增加此次合服服务器，合服次数顺延**
2. **合服前一天，合服测试服测试（合服测试服：2核4G 外网测试服相同配置 mergeh7.demigame.com）**
3. **配置lualib/public/serverinfo.lua修改merger集群，配置需要合并的两个服务器serverkey（h7dmerger集群）**
4. **修改shell/master/info/server.list和shell/master/list/h7dmerger.list中gs为被合服的serverkey**
5. **关闭合服测试服务器上游戏进程，（包括gs，cs，bs）**
6. **同步代码到合服测试服，**
   1. **开发服 ./shell/master/sync\_control.sh h7dmerger**
   2. **同步服 ./shell/master/sync\_server.sh h7dmerger**
7. **在合服测试服的代码目录下执行python ./shell/merger\_pulldata.py 1（如果已经拉取了数据可以使用python ./shell/merger\_pulldata.py 1 date，其中date为当天的日期，如2017-12-06）清空主从库，并从正式服拉取**
8. **bs cs gs起服，并在 cs 执行：echo "start\_merger no"|nc localhost 10002，在gs执行：echo "start\_merger no"|nc localhost 7002，其中no为当次测试的合服次数（service/merger/defines.lua下面配置的次数）**
9. **查看cs.log和gs.log等待cs gs合服完成，看合服中途报错，修之**
10. **gs.log输出，收到reboot now输出，需关闭close gs并启动（如果有多次合服需重启后在合）**
11. **通知客户端打包，进入合服测试服测试**
12. **检查合服后的玩家数量，合服补偿等相关游戏内容是否正常，合服补偿现阶段写死在服务器端，在service/world/mergermgr.lua文件中**

**注1：在合服测试进行合服的时候注意内存情况，一般合服测试服机器配置稍差需限制mongo内存大小（可查看mongo配置文件中cacheSizeGB: 0.5）**

**注2：如果在执行python ./shell/merger\_pulldata.py有报错有可能oom(减少mongo的cacheSizeGB)**

**注3：./shell/merger\_pulldata.py 会用到mongoresrore。 mongorestore的时候会启用多连接来加速，而连接所占用的内存是不归存储引擎管的**

1. **合服当天，关服后*mongobak备份数*据。**
2. **修改后台中被合服的‘合服编号’，改为宿主服，删除shell/master/info/server.list、shell/master/list/pro\_gs.list、shell/master/list/production.list中被合服的信息，先不要删lualib/public/serverinfo.lua中的被合服信息**
3. **同步代码到正式服**
4. **启动cs，如果已有cs进程，需要在线更新service/merger/define.lua文件，run启动宿主服，在 cs 执行：echo "start\_merger no"|nc localhost 10002，**

**并在gs执行：echo "start\_merger no"|nc localhost 7002, 其中no为当次测试的合服次数**

1. **查看gs.log，收到reboot now输出，可以close gs并run启动gs**
2. **注释掉lualib/public/serverinfo.lua中的被合服信息，更新到外服**
3. **通知测试测试合服**
4. **测试没问题放玩家**
5. **测试中若发现有部分未合并到，可在merger服务中写单对该部分合并的fix函数，在fixbug写对应的fixbug函数执行。**
6. **若问题严重，还原备份数据，并重新合服**