

今天我们继续讲解数据库高可用架构的搭建，在做好之前的准备工作之后，咱们今天就可以继续来做了，首先，要在MySQL主库所在的机器上去添加VIP，所谓VIP就是虚拟VIP地址，这个大家可以关注一下，不懂的自行搜索，是一个重要的网络概念。

`ip addr add xx.xx.xx.xx dev eth0`，这里的xx.xx.xx.xx，就是你自定义的一个VIP地址

接着就可以启动MHA manager节点了，在MHA manager所在机器上执行下述命令：`nohup masterha_manager --conf=/etc/mha/mha.conf > /tmp/mha_manager.log < /dev/null 2>&1 &`，这样就可以启动MHA的manager节点了

接着验证一下启动是否成功：`masterha_check_status --conf=/etc/mha/mha.conf`，此时只要看到MHA manager正常工作就行了，接着就可以测试一下数据库高可用了，比如你可以先把主库停了：`mysqladmin -uroot -proot shutdown`

然后从库会自动获取到主库机器上的VIP的，同时从库会被转换为新的主库，其他从库也会指向新的主库，这些都是MHA自动给你完成的，然后你可以把宕机的主库重新启动，然后把他配置为从库，指向新的主库就可以了

整体来说，数据库的高可用架构就是这么个意思，其实搭建虽然很繁琐，但是只要搭建好了，基本就是比较自动化的了，相信大家结合之前的一些内容，应该都能理解，只不过在搭建的过程中可能会遇到一些小问题，可以自己尝试去解决一下。

今天的内容就讲到这里了，这也是本周的内容，多给大家留一些时间去自己尝试一下做这个实验，下周和下周我们会讲解最后两周内容，是有关分库分表这块的一些实践案例的，大家有时间可以看一下猿技术窝的《互联网Java工程师面试突击第一季》里的分库分表的部分，里面有一些前置的分库分表的基础知识，大家可以先预习一下，后续我们会直接讲解分库分表这块的案例了。

**End**