**<常用,易忘命令.docx>**

1. xshell 在linux与wins本机间传输文件

①小文件

包:lrzsz

rz -E wins本机上传到linux

sz /root/files从linux下载文件到wins本机

②大文件

sftp:

lcd选择wins本地路径

put/get 以上支持拖拽:)

2.linux间远程登录文件系统存取文件

lftp sftp://192.168.0.250 -u root

1. drbd+NFS+keepalived

drbd:分布式 d 复制replication块block设备device

ceph的块设备:rbd 复制块设备

1）在两台机器上安装keepalived，VIP为192.168.1.200

2）将DRBD的挂载目录/data作为NFS的挂载目录。远程客户机使用vip地址挂载NFS

3）当Primary主机发生宕机或NFS挂了的故障时，Secondary主机提权升级为DRBD的主节点，并且VIP资源也会转移过来。

当Primary主机的故障恢复时，会再次变为DRBD的主节点，并重新夺回VIP资源。从而实现故障转移

WEB集群

(通过nfs,

mount VIP://var/web 网站根目录)

mount

drbd+nfs+keep(主) A

(备)B

nfs: /etc/exports

/var/web \*(rw) 作为可更新的网站根目录发布源

(两台主机操作)

drbd: mount /dev/drbd0 /var/web

drbd作用:提供主备NFS服务器间的数据同步!与rsync相比,可以传输大量数据~

保证数据的一致性。

keep中:最好都设置为BACKUP 并且设置 一台! nopreempt 一台nopreempt

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

项目简析：

项目名称 nfs+drbd高可用共享文件存储

集中进行后端web集群网页文件目录的部署，同时设置高可用。

项目要点：

1.配置nfs服务器，作为可更新的网站根目录发布源

2.nfs服务器上配置keepalived高可用

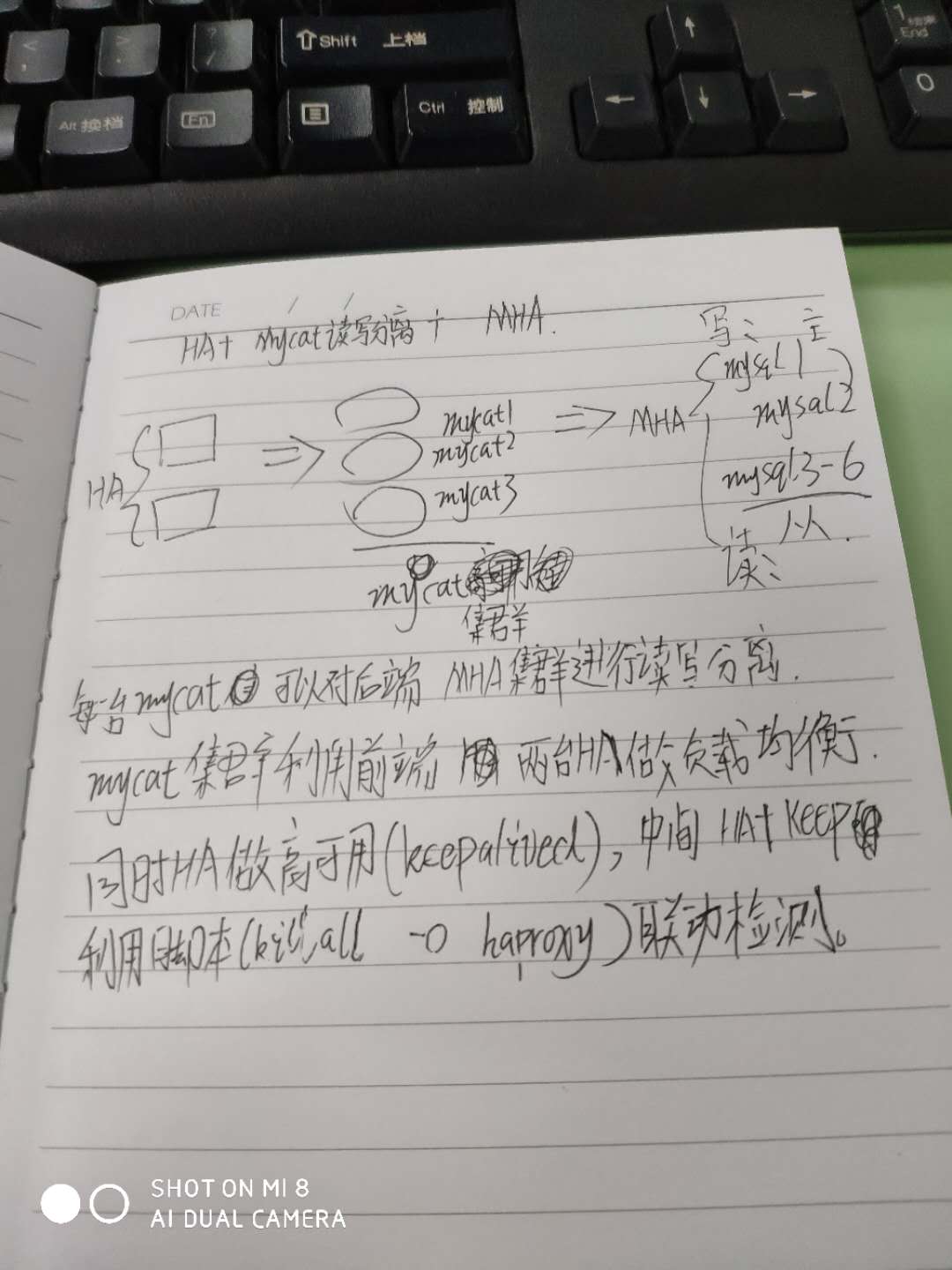
3.web服务器进行挂载VIP的共享目录的配置

4.nfs服务器上配置drbd保证数据的一致性，进行高可用nfs服务器间的数据同步。

本人：负责drbd，nfs服务的配置

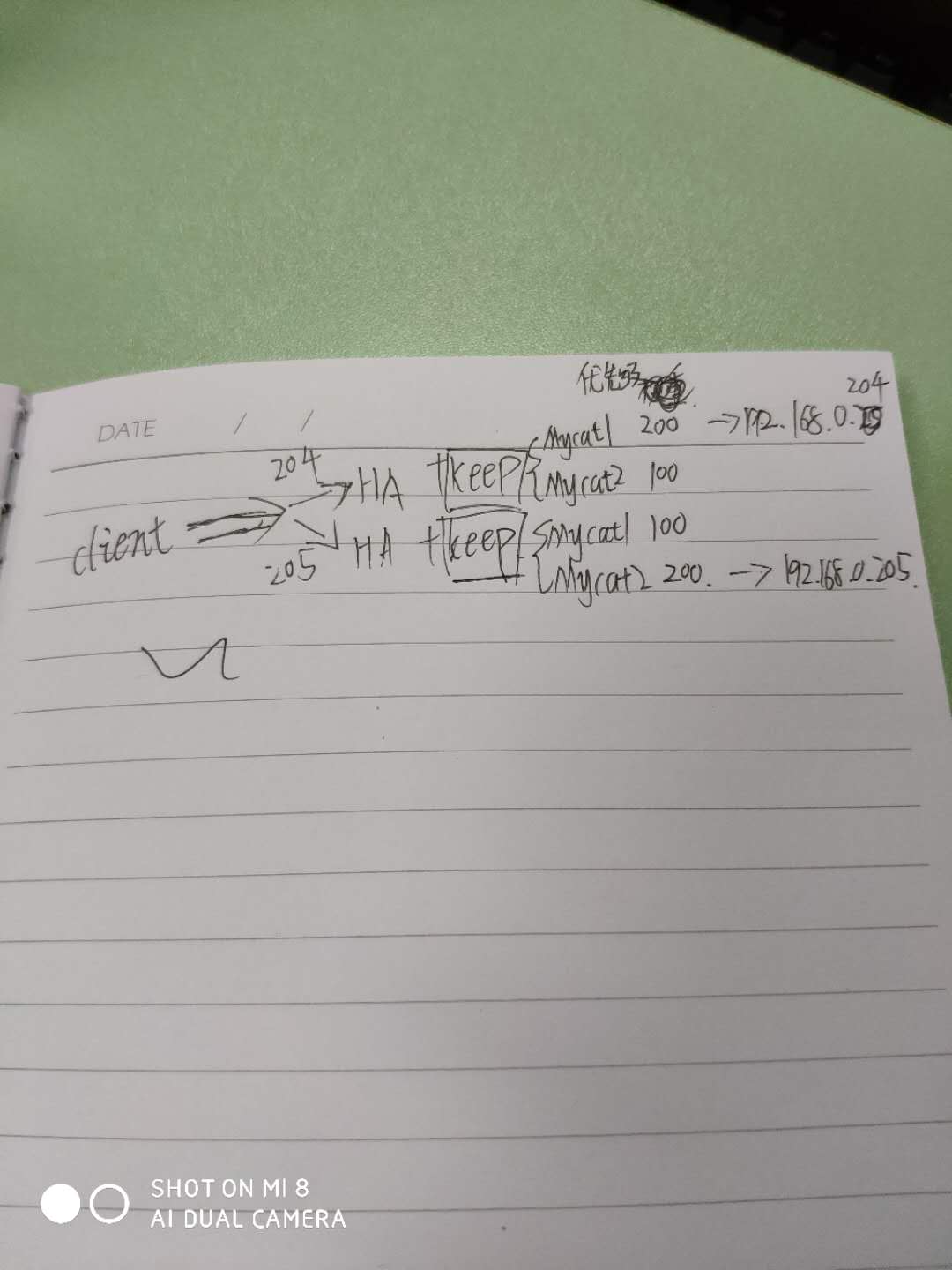
2017/12-2017/12

1. ha(keepalived)+mycat集群,读写分离+MHA



haproxy+keepalived双机双工:通过脚本,设置两个VIP,同时工作:

两台都正常时协同工作,一台死了,另外一台会接管工作(等于实现了负载均衡+高可用)



haproxy.cfg:

listen mycat\_3306 \*:3306

mode tcp # mysql 得使用 tcp 协议

option tcpka # 使用长连接

balance leastconn # 最小连接调度算法

server mycat\_01 192.168.1.13:8066 check inter 3000 rise 1 maxconn 1000 fall 3

server mycat\_02 192.168.1.14:8066 check inter 3000 rise 1 maxconn 1000 fall 3

server mycat\_03 192.168.1.15:8066 check inter 3000 rise 1 maxconn 1000 fall 3

keepalived.conf:(两台机器改不同优先级)

! Configuration File for keepalived

global\_defs {

router\_id mycat

}

vrrp\_script chk\_haproxy {

script "killall -0 haproxy" # cheaper than pidof

interval 2 # check every 2 seconds

}

vrrp\_instance Mycat\_1 {

state BACKUP

interface eth0

track\_interface {

eth0

}

virtual\_router\_id 150

priority 200

! nopreempt

advert\_int 2

authentication {

auth\_type PASS

auth\_pass test\_mycat

}

virtual\_ipaddress {

192.168.1.100/24 brd 192.168.1.255 dev eth0 label eth0:1

}

track\_script {

chk\_haproxy weight=0 # +2 if process is present

}

}

vrrp\_instance Mycat\_2{

state BACKUP

interface eth0

track\_interface {

eth0

}

virtual\_router\_id 151

priority 100

nopreempt

advert\_int 2

authentication {

auth\_type PASS

auth\_pass test\_mycat1

}

virtual\_ipaddress {

192.168.1.101/24 brd 192.168.1.255 dev eth0 label eth0:2

}

track\_script {

chk\_haproxy weight=0 # +2 if process is present

}

}

mycat的两个配置文件,server.xml

可以设置逻辑(虚拟)库名,即用户访问8066的mycat可以看到的库,访问时通过select @@hostname 反复使用,可以查看轮询结果,即轮询后端读数据库服务器.

schema.xml

从下往上配置,下为配置读写分离的具体服务器,writehost和readhost,writehost设置为MHA中使用的主库VIP,

往上为集群组成一个名为datacluster,相当于整合成一个数据库,即datahost

再上面为datanode,虚拟的数据库节点,使用上面定义的datahost,

最上面在scheme库中指明虚拟库使用的数据库节点是定义的datanode.

**<?xml version="1.0"?>**

**<!DOCTYPE mycat:schema SYSTEM "schema.dtd">**

**<mycat:schema xmlns:mycat="http://io.mycat/">**

**<schema name="mydb" checkSQLschema="false" sqlMaxLimit="100"**

**dataNode="dn1">**

**</schema>**

**<dataNode dataHost="db\_cluster" database="mydb" name="dn1"/>**

**<dataHost name="db\_cluster" maxCon="1000" minCon="10" balance="3"**

**writeType="0" dbType="mysql" dbDriver="native"**

**switchType="1" slaveThreshold="100">**

**<heartbeat>select user()</heartbeat>**

**<!-- can have multi write hosts -->**

**<writeHost host="hostMaster" url="192.168.1.10:3306"**

**user="root"**

**password="toor">**

**<!-- can have multi read hosts -->**

**<readHost host="hostS2" url="mysql15:3306" user="read"**

**password="daer" />**

**<readHost host="hostS2" url="mysql16:3306" user="read"**

**password="daer" />**

**<readHost host="hostS2" url="mysql17:3306" user="read"**

**password="daer" />**

**<readHost host="hostS2" url="mysql18:3306" user="read"**

**password="daer" />**

**</writeHost>**

**</dataHost>**

**</mycat:schema>**

以上项目小结：

具体化项目的内容，你做的是什么

第一个：ha(keepalived)+mycat集群,读写分离+MHA

Mha 管理服务器的搭建 maxscale读写分离服务器的配置（前端对多台读写服务器进行负载均衡）

难点在于ha和keep的双机双工:这是一个后续架构优化

结合ha和keep，假如某一台ha服务故障，则无法实现调度，无法正常工作，此时另一台接管调度工作；正常情况下两个VIP同时工作

手段：通过Keep的 vrrp\_script

vrrp\_script chk\_haproxy {

script "killall -0 haproxy" # cheaper than pidof

interval 2 # check every 2 seconds

}

track\_script {

chk\_haproxy weight=0 # +2 if process is present

}

Mha主节点设置故障切换脚本，候选主库和非候选主库等

第二个：

自下而上Nginx的web服务器+ha和keep的调度服务器

本人负责ha+keep的部署

PS：利用nfs+drbd高可用共享文件存储

集中进行后端web集群网页文件目录的部署，同时设置高可用。

两台nfs服务器，通过drbd进行数据同步，keep进行高可用，web服务器挂载nfs共享目录到网页根目录进行统一部署使用。

难点：利用keep的notify\_master与drbd联动，若为主节点，则运行预设脚本将本机drbd设为主，并且挂载设备到本nfs数据盘服务器的nfs共享目录中

5.web主机要配置php的redis模块,以使web服务器能连接redis集群进行缓存