# 基础架构服务器的HA方案说明及部署方法

# 背景

本方案只针对已经安装好了一套不具备HA能力的CloudOS，也只解决基础架构服务器的HA问题。

更新过程中整个CloudOS是不可用的。

本方案采用主备模式，使用Lsync在主备机之间同步数据，采用Pacemaker实现Failover，数据永远只会从主机同步到备机。Failover时，VIP会飘移至备机，备机的Docker会启动并启动所有容器，备机的Lsync服务也会随之启动。

# 准备工作

准备RHEL7.5的DVD ISO文件, 若没有可用的Yum源，则需要配置ISO Yum源。

本方案在Failover的情况下需要同时启动全部的服务，对VM的压力会比较大，因此可以酌情提高VM的规格。

记录原【基础架构-管理功能】服务器（下称“主机”）的IP地址（下称“原IP”），它将被作为基础架构功能的VIP使用。

1. 停止“主机”上的服务 docker stop `docker ps -q`
2. 停止“主机” Docker服务 systemctl stop docker
3. 禁止“主机” docker自动启动 systemctl disable docker
4. 备份“主机”的 /var/lib/cloudtogo/ 目录

#cp -r /var/lib/cloudtogo/ /data/cloudtogo/cloudtogo\_backupByHA\_20191014

1. 为“主机”分配新的IP地址。

# ifconfig -a //查看所有网卡的状态

#vim /etc/sysconfig/network-scripts/ifcfg-eth0

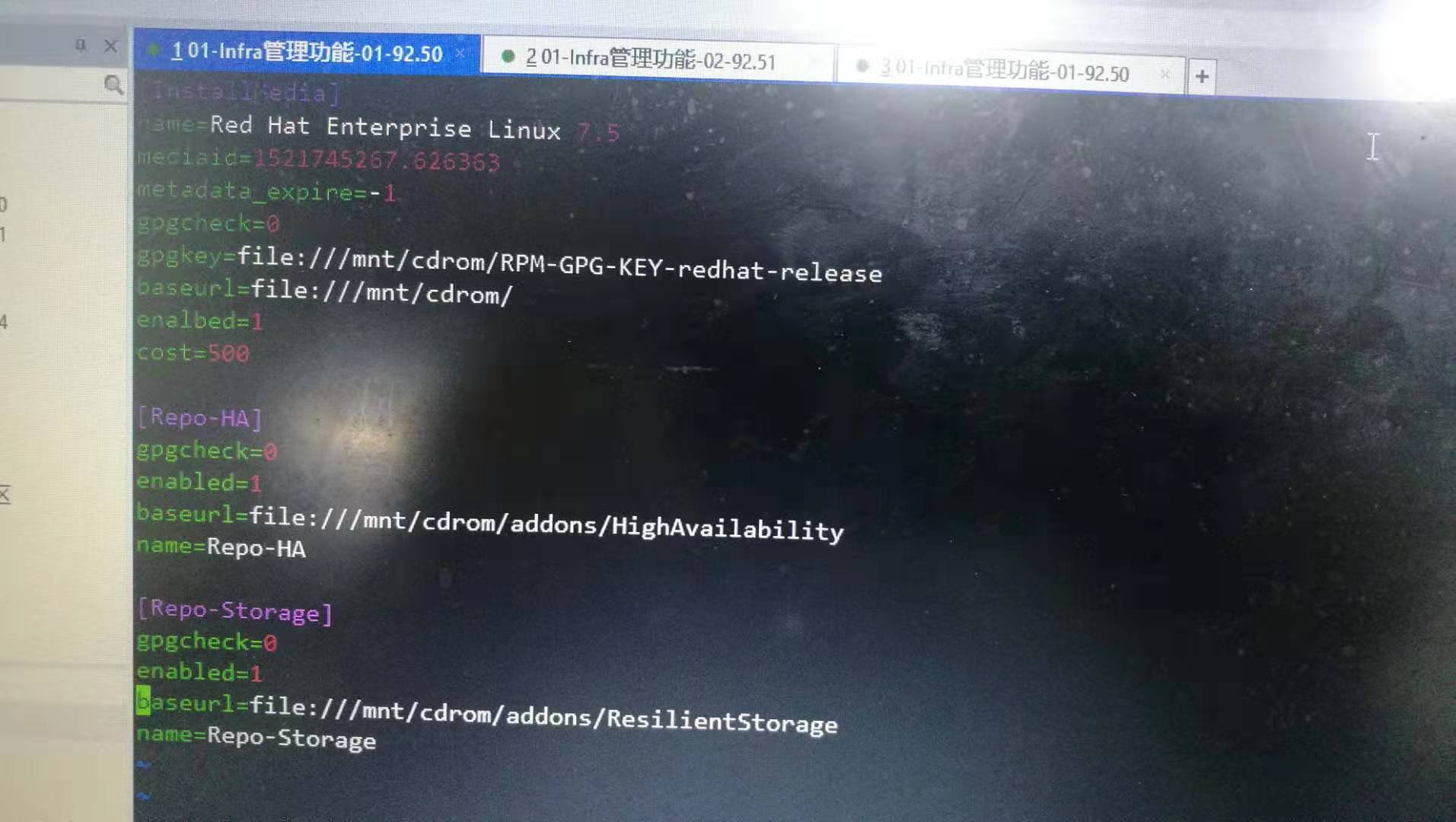
# service network restart //重启网络服务：

准备【基础架构-管理功能】的备机（下称“备机”），

1. 创建一台和“主机”规格相同的VM，并将“原IP”分配给它，
2. 将“主机”的 /root/.ctg 目录所有内容复制到“备机”，
3. 在“备机”上先后进入 /root/.ctg/bizdb.pack.res/ 和 /root/.ctg/infra.pack.res/ 目录，执行 manual.sh ，
4. 停止“备机”上的服务 docker stop `docker ps -q`
5. 停止“备机” Docker服务 systemctl stop docker
6. 禁止“备机” docker自动启动 systemctl disable docker
7. 清理“备机”上的数据，rm -rf /var/lib/cloudtogo/\*
8. 为“备机”分配新的IP地址。

# 安装HA组件

本方案的资源包位于网盘的【HA/Infrastructure】目录，下载后分别上传至“主机”和“备机”，并解压（tar xf ha-infra.tpxz）。资源包中提供了Lsyncd的RPM包，但它还依赖于Rsync，本程序还需要安装 corosync， pacemaker和pcs，这4个包都可以在RHEL的DVD中找到，因此，开始安装前需要挂载DVD并配置Yum源。（注意：addones中的HighAvailability及ResilientStorage两个HA的附加源也加加入.repo文件中） --- 这里一定要是RHEL7.5的DVD ISO文件



开始安装前，可以尝试 yum install -y rsync corosync pacemaker pcs 以确保Yum源可用，减少安装中断次数。

配置步骤如下，其中提到的VIP即前文提到的“原IP”，MASTER\_IP即“主机”的IP，STANDBY\_IP为“备机”的IP，

1. 一定要先确保“主机”和“备机”有安装lvm2

# yum list installed｜grep lvm2

# yum install -y lvm2

1. 配置“主机”和“备机”的SSH免密互信，使双方root用户可以登录对方

# ssh-keygen -t rsa

三个回车

主机上执行: ssh-copy-id infra-standby ，按提示输入备机root密码

备机上执行: ssh-copy-id infra-master，按提示输入主机root密码

这样在~/.ssh目录下就会有known\_hosts文件。

#ssh-copy-id -i ~/.ssh/id\_rsa.pub 192.14.4.15

输入对方密码

再在对方机器也执行一次相同过程，使用双方完成了免密钥登录

验证：

#ssh [root@192.16.4.15](mailto:root@192.16.4.15)

#ssh root@192.16.4.51

**!!!注意：**

测试免密登录对方机器后，一定要exit，避免操作机器错误。

测试免密登录对方机器后，一定要exit，避免操作机器错误。

测试免密登录对方机器后，一定要exit，避免操作机器错误。

1. 在“备机”上执行资源包中的 standby.sh ${VIP} ${MASTER\_IP} ${STANDBY\_IP},

[root@infra-standby] #cd /home/HA

[root@infra-standby] #ls

[root@infra-standby] #./standby.sh 10.11.0.20 10.11.0.21 10.11.0.22

1. 在“主机”上执行资源包中的 master.sh ${VIP} ${MASTER\_IP} ${STANDBY\_IP}。

[root@infra-master] #./master.sh 10.11.0.20 10.11.0.21 10.11.0.22

1. 安装过程中对主机使用权重进行了配置



用于防止安装完成后VIP立即漂移到备机，导致数据同步不完整。但是该配置会在主机重启后VIP立即从备机漂移到主机。建议将高可用扩展完成后备份主机/var/lib/cloudtogo目录文件，并删除配置。

pcs constraint location --full 查看constraint id

pcs constraint location remove <constraint id>

主机上执行: ssh-copy-id infra-standby ，按提示输入备机root密码

备机上执行: ssh-copy-id infra-master，按提示输入主机root密码

安装完成后，主机的pcs会启动docker和lsyncd，注意观察启动过程即可，由于需要一次启动多个服务，因此过程会久一些。可使用 pcs cluster status 和 pcs resource 命令查看集群状态，如需调试可以使用其他pcs命令。

在启动或切换后也可以在当前主机上通过top命令来大致判断服务是否可用，服务启动完成后，使用CPU较多的只有mysql和一个java进程，若观察到大量java进程在使用CPU，则说明还在启动中。

**HA重建：**

1. 主机执行pcs cluster destroy --all清理集群，这时会删除集群，也会停止Docker服务。备机也可尝试清理一下。
2. 清理“备机”上的数据，rm -rf /var/lib/cloudtogo/\*
3. 备机上重新执行./standby.sh 10.11.0.20 10.11.0.21 10.11.0.22
4. 主机上得新执行./master.sh 10.11.0.20 10.11.0.21 10.11.0.22

**infra主备手动切换操作**

1. 在当前工作的节点上，执行下面的命令调整monitor的间隔和超时时间，即通过延长monitor间隔防止自动切换，如果有多次切换，也只需执行一次

    pcs resource update virtual\_ip op monitor interval=3600s timeout=3600s

2. 在当前工作的节点上，执行下面的命令把当前节点设置成备节点，触发切换多次切换时需执行多次

安装好HA后，先把备机设置为standby，再把主机设置为standby，最后选择主机unstandby后正常运行

pcs node standby `hostname`

pcs node unstandby `hostname`

关于这个手动切换的操作，我认为可以再保守些。在默认情况下，主备均设置为standby状态，确定使用哪台服务后，只将那台设置为unstandby。运行过程中，若master故障，若master还存活在pcs集群中，则先standby这台机，然后再在备机上执行unstandby。这样可以绝对杜绝自动切换。

**修复主备数据不同步问题**

pcs status

vim /etc/lsyncd.conf

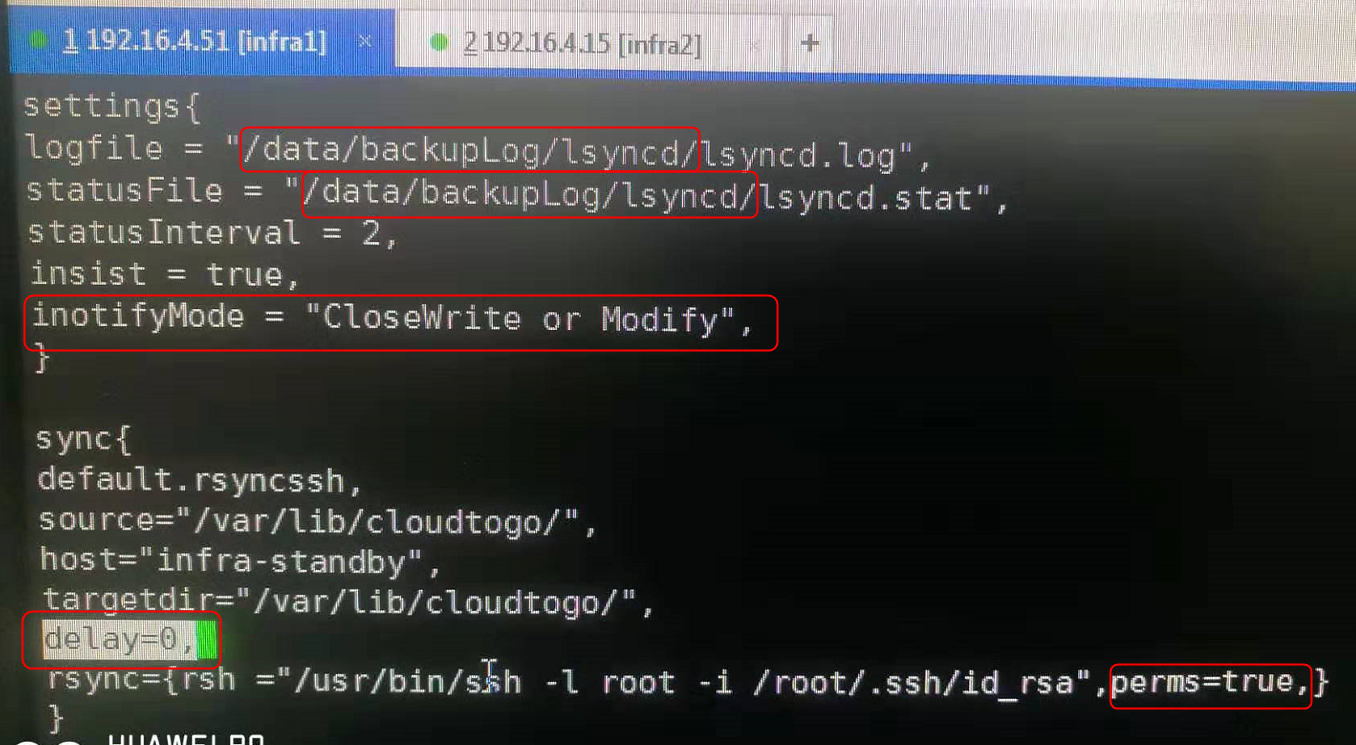
insist = true后加多一条设置：inotifyMode = “CloseWrite or Modify”,

sync={}中加多delay=0,

rsync={}中加多perms=true,whole\_file=true, //第一次

rsync={}中加多perms=true, //第二次去掉whole\_file=true,了。

设置完成后，要systemctl restart lsyncd



**保证lsyncd在docker停下来以后再停，在docker起来之前就启动**

pcs constraint order stop docker then stop lsyncd

pcs constraint order start lsyncd then start docker

**Mongo无法启动的问题：**

文件损坏是不可避免的，稍微推理下就可以知道，至于概率如何只能看运气了

只能找个之前的文件进行恢复了。/var/lib/cloudtogo