



联想 LOE 企业版

V4.3

运维手册

目 录

1. 概要.....	5
2. LOE 云平台术语	5
3. 平台状态检查.....	9
3.1. 硬件状态检查	9
3.1.1. 系统状态检查	9
3.1.2. CPU 状态检查	9
3.1.3. 内存状态检查	10
3.1.4. 网络状态检查	11
3.1.5. 磁盘状态检查	11
3.2. 服务状态检查	14
3.2.1. mysql 集群状态检查	14
3.2.2. rabbitmq 集群状态	15
3.2.3. nova 组件状态检查.....	15
3.2.4. neutron 状态检查.....	16
3.2.5. cinder 状态检查.....	17
3.2.6. ceph 状态检查.....	17
3.2.7. pacemaker 管理的各服务集群状态检查.....	18
3.2.8. l3-agent 的 namespace.....	19
3.2.9. 查看 dhcp-agent 的 namespace.....	19
4. 常见故障处理	20
4.1. 通用故障.....	20
4.1.1. 控制节点宕机	20
4.1.2. 操作系统盘损坏	21
4.1.3. 服务状态异常	21
4.1.3.1. Rabbitmq-server.....	21
4.1.3.2. hwmgmt-api.....	21
4.1.3.3. collectd.....	21
4.1.4. ntp server 不工作.....	22

4.2. 计算	22
4.2.1. 虚拟机状态错误	22
4.2.2. 同一个 vm 运行在多个宿主机上	23
4.2.3. vm (Linux 系统) 用户自己升级失败导致系统无法启动	23
4.3. 存储	24
4.3.1. ceph osd down	24
4.3.2. ceph 'error removing image'	25
4.3.3. mon.node-x store is getting too big!	25
4.3.4. Full osd 或 near full osd	25
4.4. 网络	27
4.4.1. 虚机 ping 不通 , 无法远程登录	27
5. 系统定制 Tips	27
5.1. 修改配置以启/停相关功能	27
5.1.1. 强制计费功能失效	27
5.1.2. 开启包月包年模式	27
5.1.3. 支付宝帐号修改	28
5.1.4. 激活邀请码充值	28
5.1.5. 强制申请与审批功能失效	28
5.1.6. 配置 Host HA 功能	28
5.1.7. 配置 VM HA 功能	33
5.1.8. 启用物理网络拓扑	34
5.1.9. 修改云硬盘 1TB 上限	35
5.1.10. 启用 flat 网络	36
5.1.11. 允许域账户登录	37
5.1.12. 开启 Ceph 存储空间回收通知机制	37
5.2. 系统优化选项	38
5.2.1. 为 10GB 网络修改 MTU 到 9000	38
5.2.2. 调整 Ceph pg_num	39
5.3. 系统运维常用操作	41
5.3.1. 安全开关云平台	41
5.3.2. 通过命令行上传大镜像文件	43

5.3.3. 替换 Ceph OSD 数据盘.....	43
5.4. 更新 LOE 许可证.....	44

1. 概要

联想 LOE 企业版 (简称 LOE,后续将以 LOE 代替) 是用户按需分配计算 / 存储 / 网络等资源的弹性云计算平台,用户只需通过浏览器轻点鼠标,就可以快速完成整个虚拟数据中心的搭建。LOE 云平台上的各类资源可灵活扩展或缩减,云主机秒级启动,云存储稳定可靠,秒级备份,网络隔离及防火墙确保云主机安全,对各类资源秒级计费,可灵活调整计价策略,所有数据采用多副本保证高可靠性,分钟级别的监控告警为运维人员及时全面掌握系统状态提供有力保障,分层的用户管理方便企业级管理员对各级权限的管控。

本手册将将对 LOE 日常运维操作进行详细的介绍,以便您和您的团队更好地管理基于 LOE 的云平台。

2. LOE 云平台术语

为了方便理解文中的内容,本节将介绍 LOE 云平台涉及到的术语及概念。

■ 用户

LOE 云平台提供四个级别的用户,每个级别具有不同的权限,适应企业及用户管理。权限由高到低:云管理员->企业管理员->项目管理员->普通用户。所有用户在整个 LOE 云平台内唯一、统一使用 E-mail 地址登录。

- 1) 云管理员:云管理员具有管理整个云平台的权限,在【权限管理】中,可以管理企业 / 项目 / 用户,并且具有【全局管理】菜单,管理整个平台的各项资源 / 计费定价等。系统内置帐户 admin@example.org 为云管理员帐号。其它云管理员帐户由云管理员创建。
- 2) 企业管理员:云平台针对每个企业计费。企业管理员由云管理员创建。除了具有普通用户的所有权限外,企业管理员在【权限管理】中,可以管理项目 / 用户,并查看费用使用情况。
- 3) 项目管理员:具有管理当前项目配额的功能。项目管理员由企业管理员创建。除了具有

普通用户的所有权限外，企业管理员在【权限管理】中，可以管理用户。

- 4) 普通用户 :具有操作所属项目资源的权限。普通用户可以被云管理员或企业管理员创建。

用户通过登录后，可以操作所属项目的各项资源，如云主机 / 云硬盘等。

■ 云主机

运行在 LOE 云平台上的虚拟机，相当于数据中心的一台物理服务器。用户可以通过选择合适的 CPU / 内存 / 操作系统磁盘空间，网络，安全组等配置创建云主机。系统根据云主机使用的配置和时长计费。

■ 对象存储

用户创建对象存储，用于保存大规模的非结构化数据，例如图片文件、视频文件、虚拟机镜像文件、文档文件等。用户对属于自己权限的数据可以进行检索、下载、复制、删除、更新等操作，同时支持显示对象大小、访问地址、散列值等信息。

■ 云硬盘

为云主机提供块级存储设备，相当于一台物理机的硬盘。云硬盘是独立的资源，它的生命周期独立于云主机，可以被挂载到任何云主机上，也可以从云主机卸载，然后挂接到其他云主机。云平台根据云硬盘的类型 / 容量 / 使用时长计费。

■ 镜像

操作系统的安装模版，用户可以选择合适的操作系统镜像创建所需要的云主机。只有 admin 用户具有上传镜像操作权限，其他权限的用户只能使用和查看。但用户可以通过云主机快照创建新的镜像，并在启动云主机时选择“云主机快照”类型来使用新的镜像。

■ 快照

用户可以对云主机和云硬盘创建快照，保存当时状态下的云主机和云硬盘数据。云主机快照会被保存为镜像，用户可以基于这个镜像创建新的云主机。云硬盘快照保存当时状态下的硬盘数据，并可以基于快照创建新的云硬盘。云平台根据快照的数量和使用时长计费。

■ 安全组

一系列防火墙规则组成安全组，创建云主机时，用户可以选择合适的安全组来保障云主机的安全。安全组对主机上的所有网卡生效，新增网卡也将应用已有的安全组。

■ 公网 IP

独立的 IP 地址资源，用户可以将申请的公网 IP 绑定到自己的云主机上，客户就可以通过这个公网 IP 来访问云主机提供的服务了。公网 IP 也绑定到路由器上，帮助内部网络连接外网或网络之间连接。云平台根据公网 IP 的个数 / 带宽 / 使用时长计费。

■ SSH 密钥对

基于密钥的安全验证登录方法，保证云主机安全。LOE 推荐使用密钥对登录云主机。

■ 网络

网络与现实世界的交换机 / 路由器 / 服务器 / 连线组成的基础设施网络类似，创建网络后，用户可以在网络内创建子网，创建云主机时选择网络，组建服务器集群。

LOE 提供的基础网络包含共享网络 (share_net) 和外部网络 (public_net)，创建在共享网络上的云主机处于同一个网络内，通过安全组保障云主机访问安全。外部网络主要用于公网 IP 地址的分配。

用户可以为项目创建内部网络，并在内部网络中创建子网。如同在物理网络上通过交换机将服务器连接到一起的局域网，服务器通过交换机连接到子网中。不同的内部网络之间是完全隔离的，因此不同的网络中可以配置相同的 IP 地址而不会产生冲突。同一个网络内可以创建多个子网，以适应业务的需求。

■ 路由器

用户创建路由器，为不同的子网提供三层路由，从而让子网内的云主机与其他子网的云主机互联互通。也可以将用户创建的内部网络连接到外部网络，让内部网络的云主机访问 internet。路由器配置公网 IP 后，还可以为内网的云主机做端口转发，以节约公网 IP 地址资源。

■ 负载均衡

用户创建负载均衡，能够将所收到的网络流量分配给若干个提供相同处理功能的虚拟机，并按照特定的算法保证每台虚拟机工作在最优的负载状态，从而达到更高效的使用计算资源的目的。这些虚机构成了一个集群，负载均衡会为集群设置一个对外提供服务的地址 Virtual IP，外部用户通过 Virtual IP 实现对集群的访问。负载均衡必须能够连接外部网络，如果不关联 Floating IP，平台将会随机为负载均衡分配一个公网地址。

■ 防火墙

防火墙提供网络间的访问控制功能，通过防火墙策略中的过滤规则对当前项目中的网络流量进行过滤。防火墙必须与一个防火墙策略相关联，防火墙策略是防火墙规则的集合，防火墙规则支持多种网络协议。每个项目只允许配置一个防火墙。

■ 虚拟专用网

虚拟专用网是在云网络上建立一个临时的、安全、稳定的连接。虚拟专用网是对云网的扩展，可以帮助不同租户间建立可信的安全连接，并保证数据的安全传输。

■ 网络拓扑

展示用户当前所在项目的网络结构图。点击各个设备可以展示详细配置。

■ 告警

用户对资源（云主机 / 云硬盘等）的监控数据设置告警条件，当监控数据达到阈值就会发送告警到通知列表中的邮件。

■ 企业

企业通过云管理员直接创建生成。企业创建成功后，LOE 自动为企业创建企业账户用户计费，创建企业管理员用于企业管理。LOE 对企业账户进行计费，企业管理员管理项目和用户，可以查看其管理范围内项目的消费清单。在其管理范围内的项目名称唯一，但可以与其他企业用户管理的项目名称重合。

■ 项目

项目是定义资源所有权的基本单元，所有资源（如云主机等）都要隶属于某个项目中。项目必须隶属于一个企业。项目名称在单个企业的管理范围内是唯一的，但在整个云平台中可以不唯一。

■ 邀请码

云管理员创建 / 管理邀请码。其它用户可通过邀请码给帐户充值。

3. 平台状态检查

3.1. 硬件状态检查

硬件状态检查主要关注服务器的各硬件组件的可用状态，如磁盘、主板、CPU、内存、网卡、RAID 卡、交换机等等。

检查方式分为现场巡检与硬件监控。现场巡检主要检查磁盘状态等是否变黄，如果变黄表明磁盘损坏，需尽快进行更换。服务器前面板状态等是否变黄，如果变黄，表明主板或 CPU 等组件存在硬件问题。需尽快联系技术支持进行更换。网卡是否正常闪亮，交换机端口是否正常闪亮。等等。

也可登录各硬件厂商提供的远程管理端口来查看各硬件组件的状态。根据厂商不同，管理与访问方式也有所不同。详细内容，请参考厂商提供的硬件管理文档。

3.1.1. 系统状态检查

系统状态检查，主要关注操作系统的主要计算资源。如 CPU，内存，磁盘，网络等等。

3.1.2. CPU 状态检查

通过运行 `mpstat 3` 命令来查看系统所有 CPU 加权平均的资源使用状态。其中 3 表示采样时间间隔为 3 秒。详细内容请参考下图。

```
[root@node-3 ~]# mpstat 3
Linux 2.6.32-431.20.3.el6.x86_64 (node-3.domain.tld) 04/27/2015 _x86_64_ (40 CPU)

05:30:53 PM CPU %usr %nice %sys %iowait %irq %soft %steal %guest %idle
05:30:56 PM all 2.13 0.00 2.07 0.02 0.00 0.11 0.00 0.00 95.67
05:30:59 PM all 2.98 0.00 2.75 0.02 0.00 0.21 0.00 0.00 94.04
05:31:02 PM all 3.67 0.00 2.75 0.02 0.00 0.20 0.00 0.00 93.36
05:31:05 PM all 2.78 0.00 2.50 0.02 0.00 0.13 0.00 0.00 94.57
```

如果想要查看具体的 CPU Core 的使用状态，可以使用参数-P，后跟要查看的 CPU 核编号。如 0，表示 CPU 的一个 Core。详细内容请参考下图。

```
[root@node-3 ~]# mpstat -P 0 3
Linux 2.6.32-431.20.3.el6.x86_64 (node-3.domain.tld) 04/27/2015 _x86_64_ (40 CPU)

05:31:10 PM CPU %usr %nice %sys %iowait %irq %soft %steal %guest %idle
05:31:13 PM 0 25.34 0.00 14.04 0.00 0.00 5.82 0.00 0.00 54.79
05:31:16 PM 0 20.55 0.00 10.96 0.00 0.00 4.45 0.00 0.00 64.04
05:31:19 PM 0 32.07 0.00 13.79 0.00 0.00 7.59 0.00 0.00 46.55
05:31:22 PM 0 35.62 0.00 15.75 0.00 0.00 7.53 0.00 0.00 41.10
```

3.1.3. 内存状态检查

通过运行 free -g 命令可以查看系统的内存与 SWAP 使用率。

```
[root@node-3 ~]# free -g
              total        used         free       shared    buffers       cached
Mem:           504         129         375           0           0           78
-/+ buffers/cache:          50         454
Swap:           3           0           3
```

通过运行 vmstat 3 命令，可以查看系统的内存与 SWAP 状态。3 为采样时间间隔，单位为秒。如果系统中的 swpd 值比较大，而 free 值较小，说明物理内存已经不够，需要添加内存或者采取其他资源释放的操作。

```
[root@node-3 ~]# vmstat 3
procs -----memory----- --swap-- ----io---- --system-- -----cpu-----
r b swpd free buff cache si so bi bo in cs us sy id wa st
7 0 0 393558112 461716 82234368 0 0 2 58 0 0 6 2 92 0 0
0 0 0 393713120 461716 82234800 0 0 0 1724 13727 10651 3 3 95 0 0
3 0 0 393566144 461716 82235328 0 0 0 345 12850 9780 3 3 95 0 0
35 0 0 393712672 461716 82235480 0 0 0 949 15941 11714 4 3 93 0 0
3 0 0 393622112 461716 82235760 0 0 0 499 13568 10224 4 2 94 0 0
1 0 0 393711264 461716 82236312 0 0 0 1251 20358 15090 4 3 93 0 0
4 0 0 393706752 461716 82236592 0 0 0 1125 17509 12029 4 3 93 0 0
1 0 0 393738496 461716 82223904 0 0 0 13208 14087 11596 3 2 95 0 0
```

可以结合 vmstat 命令来判断系统是否繁忙，其中：

项	值	说明
---	---	----

proc	r	等待运行的进程数。一般负载超过了 3 就比较高，超过了 5 就高，超过了 10 就不正常了，服务器的状态很危险。
	b	处在非中断睡眠状态的进程数
memory	swpd	虚拟内存使用情况，单位为 KB。如果大于 0，表示你的机器物理内存不足了
	free	空闲的内存，单位为 KB
	buff	Linux 系统用来存储目录索引/权限等的缓存，单位为 KB
	cache	被用来作为缓存的内存数，如已打开文件的内容，单位为 KB
swap	si	从磁盘交换到内存的交换页数量，单位为 KB。如果大于 0，表示你的机器物理内存不足了
	so	从内存交换到磁盘的交换页数量，单位为 KB。如果大于 0，表示你的机器物理内存不足了
io	bi	块设备每秒接收的块数，单位为 KB。正常情况应该接近 0，否则表明 IO 过于频繁
	bo	块设备每秒发送的块数，单位为 KB。正常情况应该接近 0，否则表明 IO 过于频繁
system	in	每秒的中断数，包括时钟中断
	cs	每秒的环境切换次数，如系统函数调用，线程切换等。这个值要越小越好，太大了，要考虑调低线程或者进程的数目
cpu (百分比)	us	用户空间 cpu 使用时间
	sy	内核空间 cpu 使用时间
	id	空闲 cpu 时间。一般地，id,sy 和 us 之和为 100
	wt	等待 IO cpu 时间

3.1.4. 网络状态检查

通过运行 `netstat -i` 命令，可以查看每个网络接口上的接收与发送的数据包数。并且可以查看 MTU 值与有错误或者丢弃的包数，有助于网络故障的诊断。

```
[root@vpn-server ~]# netstat -i
Kernel Interface table
Iface      MTU Met    RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR Flg
br-eth1    1500  0  5114270      0      0      0  2817690      0      0      0 BMRU
eth0       1500  0  63856040      0      0      0 18233575      0      0      0 BMRU
eth1       1500  0  71612603    493    493      0  73045355      0      0      0 BMRU
lo         65536 0   328772      0      0      0   328772      0      0      0 LRU
virbr0     1500  0      0      0      0      0      0      0      0      0 BMRU
vnet0      1500  0  59148282      0      0      0  63326101      0      0      0 BMRU
```

3.1.5. 磁盘状态检查

通过运行 `df -h` 命令，可以查看系统中的磁盘使用率。如果系统中磁盘的使用率超过 80%，需要采取进一步的操作，如添加新磁盘或者删除不必要的大文件等等。如发现某个分

区空间接近用完，可以进入该分区的挂载点，用以下命令找出占用空间最多的文件或目录，

然后按照从大到小的顺序，找出系统中占用最多空间的前 10 个文件或目录：

```
du -cksh *|sort -rn|head -n 10
```

```
[root@node-3 ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/mapper/os-root 362G  31G  313G   9% /
tmpfs           253G   37M  253G   1% /dev/shm
/dev/md0        194M   25M  159M  14% /boot
/dev/mapper/mongo-mongodb 1.1T   81G  960G   8% /var/lib/mongo
```

```
[root@node-1 storagegmt]# ssh node-3
Warning: Permanently added 'node-3,192.168.2.4' (ECDSA) to the list of known hosts.
Last login: Fri Nov  2 02:13:28 2018 from 192.168.2.2
[root@node-3 ~]# du -cksh *|sort -rn|head -n 10
588K    total
132K    ceph.log
116K    ks-pre.log
88K     anaconda-ks.cfg
84K     original-ks.cfg
84K     cobbler.ks
28K     ks-post.log
16K     post-partition.log
8.0K    2.xml
8.0K    1.xml
```

通过运行 `df -i` 命令，可以查看系统中的 inode 使用率。每个文件都对应一个 inode。系统中的 inode 数是有限的，如果系统中 inode 的使用率超过 80%，需要采取进一步的操作，如添加新磁盘或者删除不需要的文件等等。可以通过命令 `find /var/log -size -5k|wc -l` 看小文件数，删除掉无用的小文件。

```
[root@node-12 ~]# df -i
Filesystem      Inodes    IUsed     IFree IUse% Mounted on
/dev/mapper/os-root 3276800 151636   3125164    5% /
devtmpfs        32946078    826   32945252    1% /dev
tmpfs           32949074     88   32948986    1% /dev/shm
tmpfs           32949074   1428   32947646    1% /run
tmpfs           32949074     16   32949058    1% /sys/fs/cgroup
/dev/md0        64000     338     63662    1% /boot
/dev/mapper/mongo-mongodb 164560896    11 164560885    1% /var/lib/mongo
none            32949074     16   32949058    1% /var/tmp
tmpfs           32949074      1   32949073    1% /run/user/0
```

通过运行 `iostat 3` 命令，可以查看磁盘的 IO 情况。其中 3 为采样时间间隔，单位为秒。


```
[root@node-3 ~]# iostat 3
Linux 2.6.32-431.20.3.el6.x86_64 (node-3.domain.tld) 04/27/2015 _x86_64_ (40 CPU)

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           6.17    0.00    1.57    0.02    0.00   92.24

Device:            tps    Blk_read/s    Blk_wrtn/s    Blk_read    Blk_wrtn
sda                 148.56         43.03        4470.39   155204397   16125798154
sdb                  15.51         39.47        151.35   142360789    545974002
sdc                   6.86         39.43         29.33   142240761   105811426
sdd                   6.02         39.43          0.42   142240465    1526970
sde                   4.86         39.43          0.00   142239673         122
dm-0                 559.53          3.59        4470.39   12957866   16125798032
dm-1                  0.00          0.00          0.00        2416          0
md0                  0.00          0.00          0.00        4772         10
dm-2                 20.93          0.03        181.11    120810    653312032
```

通过运行 `iostat -x 1 5` 命令来查看系统中是否存在 IO 性能瓶颈。iostat 是含在套装 `sysstat` 中的, 可以用 `yum -y install sysstat` 来安装。

常关注的参数：

如果 `%util` 接近 100%, 说明产生的 I/O 请求太多, I/O 系统已经满负荷, 该磁盘可能存在瓶颈。如果 `idle` 小于 70%, I/O 的压力就比较大了, 说明读取进程中有较多的 wait。

```
[root@node-3 ~]# iostat -x 1 5
Linux 2.6.32-431.20.3.el6.x86_64 (node-3.domain.tld) 04/27/2015 _x86_64_ (40 CPU)

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           6.17    0.00    1.58    0.02    0.00   92.24

Device:            rrqm/s    wrqm/s      r/s      w/s    rsec/s    wsec/s  avgrq-sz  avgqu-sz   await  svctm   %util
sda                 5.86    421.89    4.94   143.62    43.02   4469.54    30.38     0.10    0.68   0.10   1.53
sdb                 5.89     7.03    4.86   10.65    39.46    151.39    12.30     0.01    0.33   0.03   0.05
sdc                 5.90     2.76    4.86     2.01    39.43     29.34    10.02     0.00    0.05   0.05   0.03
sdd                 5.89     0.00    4.87     1.15    39.43     0.42     6.63     0.00    0.06   0.06   0.03
sde                 5.90     0.00    4.86     0.00    39.43     0.00     8.12     0.00    0.19   0.18   0.09
dm-0                0.00     0.00     0.04   559.38     3.59   4469.54     8.00     0.48    0.85   0.03   1.51
dm-1                0.00     0.00     0.00     0.00     0.00     0.00     8.00     0.00    0.07   0.07   0.00
md0                 0.00     0.00     0.00     0.00     0.00     0.00     5.50     0.00    0.00   0.00   0.00
dm-2                0.00     0.00     0.00   20.94     0.03   181.16     8.65     0.01    0.46   0.02   0.04

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           3.66    0.00    1.83    0.03    0.00   94.49

Device:            rrqm/s    wrqm/s      r/s      w/s    rsec/s    wsec/s  avgrq-sz  avgqu-sz   await  svctm   %util
sda                 0.00    52.00     0.00   29.00     0.00    608.00    20.97     0.02    0.72   0.38   1.10
sdb                 0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00    0.00   0.00   0.00
sdc                 0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00    0.00   0.00   0.00
sdd                 0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00    0.00   0.00   0.00
sde                 0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00    0.00   0.00   0.00
dm-0                0.00     0.00     0.00   76.00     0.00    608.00     8.00     0.09    1.17   0.14   1.10
dm-1                0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00    0.00   0.00   0.00
md0                 0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00    0.00   0.00   0.00
dm-2                0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00    0.00   0.00   0.00
```

3.2. 服务状态检查

可以通过 `systemctl list-units --all --type=services | grep KEYWORDS` 查看系统关键服务状态,

系统关键服务前缀如下:

```
ceph-osd
ceph-mon
collectd
hwmgmt
httpd
libvirtd
openvswitch
rabbitmq-server
ntpd
neutron
openstack
```

若有服务状态异常, 需要进一步根据其日志检查错误并修复.

3.2.1. mysql 集群状态检查

#确保所有 mysql 集群都在

```
[root@node-1 ~]# mysql -e "show status" | grep wsrep_incoming_addresses
wsrep_incoming_addresses      192.168.0.3:3307,192.168.0.2:3307,192.168.0.4:3307
```

#确保 mysql 是同步的

```
[root@node-1 ~]# mysql -e "show status" | grep wsrep_local_state_comment
wsrep_local_state_comment      Synced
```

#确保 wsrep 同步完成

```
[root@node-1 ~]# mysql -e "show status" | grep wsrep_ready
wsrep_ready                    ON
```

!!! 禁止手动重起 mysql 集群, 比如用命令

```
crm resource restart clone_p_mysql
```

mysql 集群要求必须至少有一个活的节点, 所以不能将 mysql 集群全部关闭, 如果要关闭,

要做相应设置才能恢复。

3.2.2. rabbitmq 集群状态

#确 rabbitmq-server 服务正常

```
[root@node-4 ~]# systemctl status rabbitmq-server
● rabbitmq-server.service - RabbitMQ broker
   Loaded: loaded (/usr/lib/systemd/system/rabbitmq-server.service; enabled; vendor preset: disabled)
   Drop-In: /etc/systemd/system/rabbitmq-server.service.d
            └─limits.conf
   Active: active (running) since Tue 2018-09-04 14:23:21 UTC; 18h ago
   Main PID: 75215 (beam.smp)
   Status: "Initialized"
   CGroup: /system.slice/rabbitmq-server.service
           └─75215 /usr/lib64/erlang/erts-7.3.1.2/bin/beam.smp -W w -A 768 -K true -A30 -P 1048576 -K...
             └─75553 inet_gethost 4
               └─75554 inet_gethost 4

Sep 04 14:23:21 node-4.domain.tld systemd[1]: Started RabbitMQ broker.
Sep 04 14:23:21 node-4.domain.tld rabbitmq-server[75215]: completed with 7 plugins.
Sep 04 14:23:22 node-4.domain.tld rabbitmq-server[75215]: RabbitMQ 3.6.5. Copyright (C) 2007-2016 P...c.
Sep 04 14:23:22 node-4.domain.tld rabbitmq-server[75215]: ## ## Licensed under the MPL. See ...m/
Sep 04 14:23:22 node-4.domain.tld rabbitmq-server[75215]: ## ##
Sep 04 14:23:22 node-4.domain.tld rabbitmq-server[75215]: ##### Logs: /var/log/rabbitmq/rabbi...og
Sep 04 14:23:22 node-4.domain.tld rabbitmq-server[75215]: ##### ## /var/log/rabbitmq/rabbi...og
Sep 04 14:23:22 node-4.domain.tld rabbitmq-server[75215]: ##### ##
Sep 04 14:23:22 node-4.domain.tld rabbitmq-server[75215]: Starting broker...
Sep 04 14:23:23 node-4.domain.tld rabbitmq-server[75215]: completed with 7 plugins.
Hint: Some lines were ellipsized, use -l to show in full.
```

#确保所有的节点都在 running

```
[root@node-1 ~]# rabbitmqctl cluster_status
Cluster status of node 'rabbit@node-1' ...
[{nodes,[{disc,['rabbit@node-1','rabbit@node-2','rabbit@node-3']}]},
 {running_nodes,['rabbit@node-2','rabbit@node-3','rabbit@node-1']},
 {partitions,[]}]
...done.
```

#确保所有的节点能 list queues

```
[root@node-1 ~]# rabbitmqctl list_queues
```

!不建议手动重起 rabbitmq. 若 rabbitmq 状态异常且无其它方法使其恢复, 可在所有控制节

点上尝试以下方法之一:

- 1) systemctl restart rabbitmq-server
- 2) rabbitmqctl stop_app && rabbitmqctl start_app

3.2.3. nova 组件状态检查

```
[root@node-1 ~]# nova service-list
```

```
+-----+-----+-----+-----+-----+-----+-----+
| Id | Binary | Host | Zone | Status | State | Updated_at |
| Disabled Reason |
+-----+-----+-----+-----+-----+-----+-----+
```

1	nova-consoleauth	node-1.domain.tld	internal	enabled	up
2014-12-02T11:23:07.000000 -					
2	nova-scheduler	node-1.domain.tld	internal	enabled	up
2014-12-02T11:23:07.000000 -					
3	nova-conductor	node-1.domain.tld	internal	enabled	up
2014-12-02T11:23:05.000000 -					
4	nova-cert	node-1.domain.tld	internal	enabled	up
2014-12-02T11:23:05.000000 -					
5	nova-conductor	node-3.domain.tld	internal	enabled	up
2014-12-02T11:23:05.000000 -					
8	nova-consoleauth	node-3.domain.tld	internal	enabled	up
2014-12-02T11:23:10.000000 -					
11	nova-scheduler	node-3.domain.tld	internal	enabled	up
2014-12-02T11:23:10.000000 -					
14	nova-cert	node-3.domain.tld	internal	enabled	up
2014-12-02T11:23:08.000000 -					
18	nova-consoleauth	node-2.domain.tld	internal	enabled	up
2014-12-02T11:23:08.000000 -					
21	nova-scheduler	node-2.domain.tld	internal	enabled	up
2014-12-02T11:23:07.000000 -					
24	nova-conductor	node-2.domain.tld	internal	enabled	up
2014-12-02T11:23:07.000000 -					
27	nova-cert	node-2.domain.tld	internal	enabled	up
2014-12-02T11:23:09.000000 -					
30	nova-compute	node-4.domain.tld	nova	enabled	up
2014-12-02T11:23:03.000000 -					
33	nova-compute	node-5.domain.tld	nova	enabled	up
2014-12-02T11:23:03.000000 -					

+-----+-----+-----+-----+-----+-----+-----+

3.2.4. neutron 状态检查

```
[root@node-1 ~]# neutron agent-list
```

id	agent_type	host
01f2b261-4d34-4ff0-87bd-d1ee83d7c4ac	Open vSwitch agent	node-5.domain.tld :-)
15c8774d-190a-4741-8b5c-fef7dc413562	DHCP agent	node-3.domain.tld xxx
212a6adb-394d-409f-9022-73f940086835	Metadata agent	node-2.domain.tld :-)
2bd0cc46-fead-414a-93b7-624bd68f8053	Open vSwitch agent	node-1.domain.tld :-)

alive | admin_state_up |


```

| 309d33a5-211a-431a-8b48-35aecdb11f6b | Open vSwitch agent | node-2.domain.tld | :- ) |
True |
| 5dd63653-2024-4945-a823-dd967262a7f8 | L3 agent | node-3.domain.tld | :- )
| True |
| 62c6e903-7f4e-4178-95fb-23c81977d4c8 | Metadata agent | node-1.domain.tld | :- ) |
True |
| 670cf505-cac0-46a4-a20c-3cc4bb88a914 | DHCP agent | node-1.domain.tld | :- ) |
True |
| 7d98f1fa-042c-494c-ae87-73305ef1be02 | DHCP agent | node-2.domain.tld | xxx
| True |
| aadb6253-5dac-4c62-b7c9-54ffc6f39eee | Metadata agent | node-3.domain.tld | :- ) |
True |
| eceda162-20ee-4dda-b1ae-64bc2b0afbdb | Open vSwitch agent | node-4.domain.tld | :- ) |
True |
| f41489b4-8729-4fa5-b717-f1a223e5749c | L3 agent | node-1.domain.tld | xxx
| True |
| fc286f57-47d3-4c46-9550-579cbd5a0440 | Open vSwitch agent | node-3.domain.tld | :- ) |
True |
+-----+-----+-----+-----+

```

L3 agent 和 DHCP agent 只要各有一个是正常的就行。

3.2.5. cinder 状态检查

```
[root@node-1 ~]# cinder service-list
```

```

+-----+-----+-----+-----+-----+-----+
| Binary | Host | Zone | Status | State | Updated_at |
+-----+-----+-----+-----+-----+-----+
| cinder-scheduler | node-1.domain.tld | nova | enabled | up | 2014-12-02T11:24:27.000000 |
| cinder-scheduler | node-2.domain.tld | nova | enabled | up | 2014-12-02T11:24:28.000000 |
| cinder-scheduler | node-3.domain.tld | nova | enabled | up | 2014-12-02T11:24:30.000000 |
| cinder-volume | node-4.domain.tld | nova | enabled | up | 2014-12-02T11:24:32.000000 |
| cinder-volume | node-5.domain.tld | nova | enabled | up | 2014-12-02T11:24:32.000000 |
+-----+-----+-----+-----+-----+-----+

```

3.2.6. ceph 状态检查

```
[root@node-1 ~]# ceph -s
```

```
cluster 478cd836-609f-4d00-96bb-ac58b813ca99
```

```

health HEALTH_OK
monmap e3: 3 mons at
{node-3=192.168.0.5:6789/0,node-4=192.168.0.6:6789/0,node-5=192.168.0.7:6789/0}
election epoch 12, quorum 0,1,2 node-3,node-4,node-5
osdmap e104: 24 osds: 24 up, 24 in
pgmap v32159: 704 pgs, 6 pools, 1096 MB data, 10009 objects
3380 MB used, 22308 GB / 22311 GB avail
704 active+clean
client io 505 kB/s rd, 317 kB/s wr, 837 op/s

```

若 ceph 状态为 HEALTH_WARN 或 HEALTH_ERROR , 可用 `ceph health detail` 查看详细信息 , 并且需要立刻马上检查 ceph 警告或错误原因 , 并修复。

3.2.7. pacemaker 管理的各服务集群状态检查

```

[root@node-1 ~]# crm status
Last updated: Tue Dec 2 11:40:19 2014
Last change: Tue Dec 2 11:40:13 2014 via crm_attribute on node-1.domain.tld
Stack: classic openais (with plugin)
Current DC: node-1.domain.tld - partition with quorum
Version: 1.1.10-14.el6_5.3-368c726
3 Nodes configured, 3 expected votes
25 Resources configured

```

```
Online: [ node-1.domain.tld node-2.domain.tld node-3.domain.tld ]
```

```

vip__management_old (ocf::es:ns_IPAddr2): Started node-1.domain.tld
vip__public_old (ocf::es:ns_IPAddr2): Started node-3.domain.tld
Clone Set: clone_ping_vip__public_old [ping_vip__public_old]
Started: [ node-1.domain.tld node-2.domain.tld node-3.domain.tld ]
p_openstack-ceilometer-central (ocf::es:ceilometer-agent-central): Started
node-1.domain.tld
p_openstack-ceilometer-alarm-evaluator (ocf::es:ceilometer-alarm-evaluator): Started
node-2.domain.tld
Clone Set: clone_p_mysql [p_mysql]
Started: [ node-1.domain.tld node-2.domain.tld node-3.domain.tld ]
Clone Set: clone_p_haproxy [p_haproxy]
Started: [ node-1.domain.tld node-2.domain.tld node-3.domain.tld ]
p_openstack-heat-engine (ocf::es:openstack-heat-engine): Started node-1.domain.tld
Clone Set: clone_p_neutron-openvswitch-agent [p_neutron-openvswitch-agent]
Started: [ node-1.domain.tld node-2.domain.tld node-3.domain.tld ]
Clone Set: clone_p_neutron-metadata-agent [p_neutron-metadata-agent]
Started: [ node-1.domain.tld node-2.domain.tld node-3.domain.tld ]

```

```
p_neutron-dhcp-agent    (ocf::es:neutron-agent-dhcp):    Started node-1.domain.tld
p_neutron-l3-agent      (ocf::es:neutron-agent-l3):    Started node-2.domain.tld
```

3.2.8. l3-agent 的 namespace

从上面的状态可以看到 l3-agent 在 node-2 , 所以在 node-2 上查看 namespace

```
[root@node-2 ~]# ip netns
haproxy
qrouter-b6aa0473-1fa7-44af-bc67-49efefdeb209
[root@node-2 ~]# ip netns exec qrouter-b6aa0473-1fa7-44af-bc67-49efefdeb209 ip a
18: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
19: qr-38c57255-95: <BROADCAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UNKNOWN
    link/ether fa:16:3e:51:cc:79 brd ff:ff:ff:ff:ff:ff
    inet 192.168.111.1/24 brd 192.168.111.255 scope global qr-38c57255-95
    inet6 fe80::f816:3eff:fe51:cc79/64 scope link
        valid_lft forever preferred_lft forever
20: qg-93b57a01-80: <BROADCAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UNKNOWN
    link/ether fa:16:3e:fc:fc:36 brd ff:ff:ff:ff:ff:ff
    inet 172.16.0.130/24 brd 172.16.0.255 scope global qg-93b57a01-80
    inet6 fe80::f816:3eff:fefc:fc36/64 scope link
        valid_lft forever preferred_lft forever
```

能看到 qr , qg 和对应的 IP

3.2.9. 查看 dhcp-agent 的 namespace

从上面的状态可以看到 dhcp-agent 在 node-1 , 所以在 node-1 上查看 namespace

```
[root@node-1 ~]# ip netns
haproxy
qdhcp-3344c78a-d5a8-419c-bbc6-60424bf0ef14
[root@node-1 ~]# ip netns exec qdhcp-3344c78a-d5a8-419c-bbc6-60424bf0ef14 ip a
32: tap597c28fe-4b: <BROADCAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UNKNOWN
    link/ether fa:16:3e:ec:12:c4 brd ff:ff:ff:ff:ff:ff
    inet 192.168.111.2/24 brd 192.168.111.255 scope global tap597c28fe-4b
    inet6 fe80::f816:3eff:feec:12c4/64 scope link
        valid_lft forever preferred_lft forever
33: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue state UNKNOWN
```

```
link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
inet 127.0.0.1/8 scope host lo
inet6 ::1/128 scope host
    valid_lft forever preferred_lft forever
```

可以看到一个 tap 设备，ip 是 dhcp 的 IP。

如果虚拟机分配不到 IP，请检查 dhcp 服务是否正常。

```
[root@node-1 ~]# ps aux | grep dhcp
neutron  13376  0.0  1.4 169052 34032 ?        S    03:15   0:21 /usr/bin/python
/usr/bin/neutron-dhcp-agent          --config-file=/etc/neutron/neutron.conf
--config-file=/etc/neutron/dhcp_agent.ini
nobody   24012  0.0  0.0 12948   660 ?        S    03:20   0:00 dnsmasq --no-hosts
--no-resolv --strict-order --bind-interfaces --interface=tap597c28fe-4b --except-interface=lo
--pid-file=/var/lib/neutron/dhcp/3344c78a-d5a8-419c-bbc6-60424bf0ef14/pid
--dhcp-hostsfile=/var/lib/neutron/dhcp/3344c78a-d5a8-419c-bbc6-60424bf0ef14/host
--addn-hosts=/var/lib/neutron/dhcp/3344c78a-d5a8-419c-bbc6-60424bf0ef14/addn_hosts
--dhcp-optsfile=/var/lib/neutron/dhcp/3344c78a-d5a8-419c-bbc6-60424bf0ef14/opts
--leasefile-ro --dhcp-range=set:tag0,192.168.111.0,static,120s --dhcp-lease-max=256 --conf-file=
--domain=openstacklocal
root     29846  0.0  0.0 103248   888 pts/1    S+   13:02   0:00 grep dhcp
[root@node-1 ~]# cat /var/lib/neutron/dhcp/3344c78a-d5a8-419c-bbc6-60424bf0ef14/host
fa:16:3e:ad:7e:d2,host-192-168-111-3.openstacklocal,192.168.111.3
fa:16:3e:51:cc:79,host-192-168-111-1.openstacklocal,192.168.111.1
fa:16:3e:ec:12:c4,host-192-168-111-2.openstacklocal,192.168.111.2
fa:16:3e:8d:54:33,host-192-168-111-4.openstacklocal,192.168.111.4
```

如果正常这个 host 文件里会有虚拟机的 mac 地址和分配的 IP 的记录，如果没有这条记录，请及时处理。

4. 常见故障处理

4.1. 通用故障

4.1.1. 控制节点宕机

ThinkCloud 的 OpenStack 集群中拥有 3 台 controller，任何一台 controller 宕机后。其余两台 controller 会自动接管宕机 controller 上的所有服务。运维人员需要做的，只需修复该宕

机 controller , 并重启使其上线运行即可。

4.1.2. 操作系统盘损坏

操作系统盘已经通过使用服务器 RAID 卡 , 配置了 RAID 1 或则 RAID5 对操作系统数据进行保护。并且服务器使用的磁盘 , 大多为支持热插拔的 SATA 或者 SAS 磁盘。直接将坏盘拔下 , 插入新盘 , RAID 卡会自动进行数据恢复。

4.1.3. 服务状态异常

4.1.3.1. Rabbitmq-server

现象 1 : Rabbitmq-server 状态正常 , 但 OpenStack 组件日志中报告 amqp connection timeout , 可能的原因有

- 1 . 网络异常 , 需先修复网络问题
- 2 . 使用消息队列的服务异常 . 可重启相关服务
- 3 . rabbitmq 自身异常 , 如消息阻塞等 , 可按 1.2.2 节所述重启 rabbitmq-server 服务 .

现象 2 : Rabbitmq-server 服务状态异常 , 可按 1.2.2 节所述重启 rabbitmq-server 服务 .

4.1.3.2. hwmgmt-api

hwmgmt-api 是系统关键服务 , 用于收集系统硬件信息及硬件告警 . 若 hwmgmt-api 不能正常工作 , 会导致 OpenStack 页面无法访问 . 可查看日志文件 /var/log/hwmgmt/hwmgmt-api.log 定位问题 .

4.1.3.3. collectd

collectd 是系统监控服务 , 需要内置 OpenStack 访问帐号于其配置文件 /etc/gnocchi/collectd.conf , 当内置帐号密码修改后 , 需要同步到其配置文件 .

4.1.4. ntp server 不工作

ThinkCloud OpenStack 默认将部署节点作为 ntp server, 若部署节点被移除后, 可能导致云平台各节点时间不一致, 进而导致 OpenStack 关键服务异常。在此情况下, 可将某一控制节点设为 ntp server, 具体操作如下:

1. 获取该控制节点 br-fw-admin 网口的 ip: ifconfig br-fw-admin

```
[root@node-1 ~]# ifconfig br-fw-admin
br-fw-admin: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.4 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::4b1:67ff:fef2:e141 prefixlen 64 scopeid 0x20<link>
    ether 06:b1:67:f2:e1:41 txqueuelen 1000 (Ethernet)
    RX packets 3276856 bytes 603029031 (575.0 MiB)
    RX errors 0 dropped 90 overruns 0 frame 0
    TX packets 2976841 bytes 1383031313 (1.2 GiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

2. 编辑该控制节点的/etc/ntp.conf, 在文件最后加上
server 127.127.1.0
fudge 127.127.1.0 stratum 8
3. 重启该控制节点上的 ntp 服务: systemctl restart ntpd
4. 在其它各节点上/etc/ntp.conf 的 server list 中添加该控制节点, 并重启 ntpd 服务。
server 192.168.1.4

4.2. 计算

4.2.1. 虚拟机状态错误

用户对虚拟机进行创建、迁移、重启操作时, 如果操作失败虚拟机状态变为错误, 会不断收到报警邮件。

处理方案:

1. 登录控制节点, 通过 nova show vm_id 查看 vm 所在的节点
2. 恢复虚拟机状态: nova reset-state vm_id --active
3. 关闭虚拟机

4. 登录到虚拟机所在节点，查看日志信息 (/var/log/nova/compute.log)，查看错误信息

找线索，通常原因如下：

- 4.1) 如果是重启引发的多数是存储集群有问题
- 4.2) 迁移引发的，从网络，云硬盘状态、存储集群三方面分析
- 4.3) 创建引发的，查看资源配额，网络，存储

4.2.2. 同一个 vm 运行在多个宿主机上

同一个 vm 运行在两个不同的宿主机上，通常由迁移失败引发的。

解决方法：

1. 通过 nova show vm_uuid 查看虚拟机实际在哪个宿主机
2. 关闭虚拟机
3. 登录到另一台宿主机上，通过 virsh destroy instance_name

4.2.3. vm (Linux 系统) 用户自己升级失败导致系统无法启动

解决方法：

1. 关闭虚拟机，查看该虚拟机的 id
2. 新建一个 vm,并找到该 vm 所在节点
3. 登录新建虚拟机所在的节点
4. 新建 attach.xml，内容如下：

```
<disk type='network' device='disk'>
```

```
<driver name='qemu' type='raw' cache='none' />
```

```
<auth username='compute'>
```

```
<secret type='ceph' uuid='a5d0dd94-57c4-ae55-ffe0-7e3732a24455' />
```

```
</auth>
```

```
<source
```

```
protocol='rbd'
```

```
name='compute/6f0ae0fe-150c-4837-9366-c4e54e2b8767_disk'>
```

```
<host name='192.168.0.3' port='6789'>
```

```
<host name='192.168.0.4' port='6789'>
```

```
<host name='192.168.0.5' port='6789'>
```

```
</source>
```

```
<backingStore/>
```

```
<target dev='vdb' bus='virtio'>
```

```
</disk>
```

- 1) 用出问题的虚拟机的 id 替换标红的 uuid
- 2) 在新建 vm 所在宿主机上执行 `virsh attach-device attach.xml`
- 3) 等进到 vm 系统中 , 挂载云硬盘到/tmp 如 : `mount /dev/vdb /tmp`
- 4) 编辑/tmp/boot/grub/grub.conf 修改启动选项 , 完成退出 , 并卸载该盘 , `umount /tmp`
- 5) 在新建 vm 所在宿主机上执行 `virsh detach-device attach.xml`
- 10) 重启问题虚拟机

4.3. 存储

4.3.1. ceph osd down

解决方法 :

- 1) 通过 `ceph osd tree|grep down` 查找 osd 所在的节点
- 2) 登录到该节点 , 通过 `dmesg` 信息查看是否有关于磁盘方面的错误 , 并通过 `ipmi` 确认是否为硬件问题
- 3) 如果是硬件问题 , 把该 osd 移出集群 , 具体方法如下 :


```
/etc/init.d/ceph stop osd.$osd_num
ceph osd out osd.$osd_num
ceph osd crush rm osd.$osd_num
ceph osd rm osd.$osd_num
ceph auth del osd.$osd_num
umount /var/lib/ceph/osd/ceph-$osd_num
rm -rf /var/lib/ceph/osd/ceph-$osd_num
```

注：确认好是硬件问题，否则不要轻易移除 osd

4) 如果不是硬件问题，启动该 osd，启动命令：`/etc/init.d/ceph start osd.$osd_num`

4.3.2. ceph 'error removing image'

虚拟机删除失败，如查看 log 有 ceph 'error removing image' 信息是可能 vm 运行在两个宿主机上，解决方法参见 [2.2.2](#)

4.3.3. mon.node-x store is getting too big!

解决方法：

1. 登录到对应节点如 node-1 执行 `ceph tell mon.node-1 compact`

注：压缩完成后,mon 状态为 down,等 up 后在进行另一个 mon 压缩

2. 重启 mon，删除 LOG.old 文件，如：重启 node-1 上 mon

```
>> /etc/init.d/ceph stop mon.node-1
>> rm /var/lib/ceph/mon/ceph-node-1/store.db/LOG.old
```

4.3.4. Full osd 或 near full osd

解决方法：

1. Ceph osd df 查看各 osd 使用量，通过手动调整各 osd 权重看是否能恢复 ceph 到健康状态：

```
[root@node-2 ~]# ceph osd df
ID WEIGHT REWEIGHT SIZE USE AVAIL %USE VAR
0 2.17999 1.00000 2233G 1980G 253G 88.65 1.14
3 2.50000 1.00000 2233G 1680G 553G 75.22 0.97
6 2.50000 1.00000 2233G 1681G 552G 75.27 0.97
9 2.20000 1.00000 2233G 1633G 600G 73.11 0.94
12 1.79999 1.00000 2233G 1864G 369G 83.46 1.07
15 2.17999 1.00000 2233G 1578G 655G 70.66 0.91
18 2.17999 1.00000 2233G 1871G 362G 83.79 1.08
21 2.17999 0.80925 2233G 1779G 454G 79.67 1.03
24 1.79999 1.00000 2233G 1832G 400G 82.05 1.06
27 2.29999 1.00000 2233G 1512G 721G 67.71 0.87
30 2.17999 1.00000 2233G 1958G 275G 87.68 1.13
33 2.17999 1.00000 2233G 1786G 447G 79.96 1.03
1 2.17999 1.00000 2233G 1628G 605G 72.88 0.94
4 2.17999 1.00000 2233G 1837G 396G 82.26 1.06
7 2.17999 1.00000 2233G 1758G 475G 78.73 1.01
10 2.17999 1.00000 2233G 1669G 564G 74.74 0.96
13 1.39999 1.00000 2233G 1535G 698G 68.75 0.89
17 1.79999 1.00000 2233G 1659G 574G 74.29 0.96
19 2.17999 1.00000 2233G 1791G 442G 80.21 1.03
23 2.17999 0.72020 2233G 1776G 457G 79.51 1.02
25 2.17999 1.00000 2233G 1546G 687G 69.23 0.89
29 1.79999 1.00000 2233G 1817G 416G 81.37 1.05
31 2.17999 1.00000 2233G 1578G 655G 70.66 0.91
34 2.17999 1.00000 2233G 1701G 532G 76.17 0.98
2 3.00000 1.00000 2233G 1994G 239G 89.29 1.15
5 2.17999 0.80753 2233G 1708G 525G 76.47 0.98
8 2.17999 0.81128 2233G 1820G 413G 81.50 1.05
11 2.17999 1.00000 2233G 1671G 562G 74.83 0.96
14 2.17999 1.00000 2233G 1781G 452G 79.73 1.03
16 2.17999 1.00000 2233G 1847G 386G 82.69 1.06
20 2.17999 1.00000 2233G 1897G 336G 84.94 1.09
22 1.79999 1.00000 2233G 1579G 654G 70.69 0.91
28 1.79999 1.00000 2233G 1711G 522G 76.62 0.99
32 2.17999 1.00000 2233G 1625G 608G 72.78 0.94
35 2.17999 1.00000 2233G 1744G 489G 78.10 1.01
26 1.79999 1.00000 2234G 1585G 649G 70.94 0.91
36 2.17999 1.00000 2233G 1744G 489G 78.10 1.01
37 2.50000 1.00000 2233G 1681G 552G 75.25 0.97
38 2.17999 1.00000 2233G 1748G 485G 78.26 1.01
39 2.29999 1.00000 2233G 1591G 641G 71.26 0.92
40 2.17999 0.82684 2233G 1745G 488G 78.14 1.01
41 2.17999 1.00000 2233G 1672G 561G 74.86 0.96
42 2.17999 1.00000 2233G 1833G 400G 82.06 1.06
43 2.29999 1.00000 2233G 1638G 595G 73.33 0.94
44 2.17999 1.00000 2233G 1798G 434G 80.53 1.04
45 2.17999 1.00000 2233G 1859G 374G 83.25 1.07
46 2.17999 1.00000 2233G 1911G 321G 85.59 1.10
47 2.50000 1.00000 2234G 1638G 595G 73.33 0.94
TOTAL 104T 83293G 23935G 77.68
MIN/MAX VAR: 0.87/1.15 STDDEV: 5.39
```

通过命令 `ceph osd crush reweight osd.x yyy` 将使用率偏高的 weight 下调，偏低的上调，

这里 yyy 值参考 WEIGHT.

若上述方法仍不能让 `ceph -s` 恢复到 HEALTH_OK 或 HEALTH_WARN,则进一步适量增大调

整:

```
ceph tell osd.* injectargs '--mon_osd_full_ratio 0.97' #was 0.95
ceph tell osd.* injectargs '--mon_osd_nearfull_ratio 0.9' #was 0.85
ceph tell osd.* injectargs '--osd_backfill_full_ratio 0.9' #was 0.85
ceph tell osd.* injectargs '--osd_failsafe_nearfull_ratio 0.92' #was 0.9
```

2. 检查 pg_num/pgp_num 是否合理

每个 pool 的 pg num 的经验计算公式: $(100 * \text{OSD 盘数}) / (\text{副本数} * \text{pool 数})$, 向上取最接

近的 2 的倍数. 注: 这里的 pool 数仅计算真实数据的 pool.

4.4. 网络

4.4.1. 虚机 ping 不通，无法远程登录

解决方法：

1. 通过 vnc 登录虚拟机验证系统是否正常
2. 查看安全组，是否开放 ICMP 端口
3. 查看路由设置,有些需要设置默认路由。Linux 通过 `ip route show` ,windows 通过 `route print`
 - 3.1) 只有 10.x.x.x 网段查看默认路由
 - 3.2) 配置了 43.x.x.x 和 10.x.x.x 网段的，43.x.x.x 对应默认路由
10.x.x.x 需要单独添加路由规则

5. 系统定制 Tips

5.1. 修改配置以启/停相关功能

5.1.1. 强制计费功能失效

部署时选装了计费模块，若想让计费功能失效，可在所有控制节点上设置：

```
#vim /etc/openstack-dashboard/local_settings
ENABLE_BILLING = False
#service httpd restart
```

5.1.2. 开启包月包年模式

要需要到控制节点上修改配置文件，具体如下：

在 controller 节点执行以下操作：

```
#vim /etc/openstack-dashboard/local_settings
```

添加下面配置

```
PREBILLING = True #包年包月
```

然后重启服务

```
#service httpd restart
```

5.1.3. 支付宝帐号修改

若需要修改支付宝帐号，可在所有控制节点上设置：

```
#vim /usr/share/openstack-dashboard/easystack_dashboard/settings.py
ALIPAY_KEY = 'xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx'
ALIPAY_PARTNER = 'xxxxxxxxxxxxxxxx' # partner ID
ALIPAY_SELLER_EMAIL = 'xxxxxxx' #alipay account
#service httpd restart
```

5.1.4. 激活邀请码充值

在所有控制节点上：

```
#vim /etc/openstack-dashboard/local_settings
INVCODE_RECHARGE = True
PREBILLING = True
#service httpd restart
```

5.1.5. 强制申请与审批功能失效

部署时选装了申请与审批模块，若想让申请与审批功能失效，可在所有控制节点上设置：

```
#vim /etc/openstack-dashboard/local_settings
APPROVAL_ENABLED = False
#service httpd restart
```

5.1.6. 配置 Host HA 功能

■ 功能说明：

1. 人为关物理机或调整网卡设置前，要使用 `nova service-disable {node-name}`
`nova-compute` 将物理机置成维护模式，明确告诉 Host HA 不需要监管该物理主机。
2. 因为后台存储 / 网络等原因，虚拟机状态不正确，需要管理员介入，恢复虚拟机状态。

- 人工介入时，需要将 Host HA 运行时失效，可创建一个空文件
`/etc/lenovo/enable_ha.conf`，使 Host HA 结束当前监管 等待 `/var/log/nova/hagent.log`
 出现 " skip Host HA by user " 字样才可以人工介入。若需要重新使 Host HA 生效，
 可删除该文件。
- 确认每个计算节点的 `/etc/nova/nova.conf` 配置项 `my_ip={管理网 I P}`
`resize_confirm_window=600`，控制节点 `allow_resize_to_same_host=False`，
`allow_migrate_to_same_host = False`
- 某个物理机断电，很短的时间内重新插电，仅当系统已经重新插电，IPMI 状态显示系统为 ON，由于系统没有完全启动，其他所有状态都是 FAI L，Host HA 会执行 POWER OFF 和 Evacuate。这种情况下，虚机的业务已经受到影响，即使重新加电服务器，等到操作系统及服务起来还需要比较长时间，这期间用 IPMI 关物理机，把虚机迁移走也是必要的。
- 极特殊情况下，在接连几个检测周期里，物理机生产网卡恰好依次出现问题，每次检测周期内都满足有问题的物理机总数小于或等于 `/etc/lenovo/hagent.conf` 中的配置项 `fault_hosts_number_threshold` 时，按照目前的处理逻辑，每检查到一个失败，就会 migrate 一次，或者当物理机上虚机过多，没有物理机可以 migrate 时，需要管理员介入。
- 当一台计算节点发生故障需要迁移其上的虚机时，系统会 disable 其 nova-compute 服务，如下图：

```

[root@node-1 ~]# source /root/openrc
[root@node-1 ~]# nova service-list

```

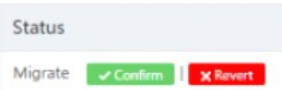
Id	Binary	Host	Zone	Status	State	Updated_at	Disabled Reason
1	nova-cert	node-1.domain.tld	internal	enabled	up	2017-03-30T06:15:09.000000	-
2	nova-consoleauth	node-1.domain.tld	internal	enabled	up	2017-03-30T06:15:08.000000	-
8	nova-scheduler	node-1.domain.tld	internal	enabled	up	2017-03-30T06:15:10.000000	-
9	nova-conductor	node-1.domain.tld	internal	enabled	up	2017-03-30T06:15:14.000000	-
13	nova-cert	node-3.domain.tld	internal	enabled	up	2017-03-30T06:15:17.000000	-
16	nova-consoleauth	node-3.domain.tld	internal	enabled	up	2017-03-30T06:15:09.000000	-
25	nova-scheduler	node-3.domain.tld	internal	enabled	up	2017-03-30T06:15:14.000000	-
28	nova-conductor	node-3.domain.tld	internal	enabled	up	2017-03-30T06:15:17.000000	-
34	nova-cert	node-2.domain.tld	internal	enabled	up	2017-03-30T06:15:10.000000	-
37	nova-consoleauth	node-2.domain.tld	internal	enabled	up	2017-03-30T06:15:08.000000	-
43	nova-scheduler	node-2.domain.tld	internal	enabled	up	2017-03-30T06:15:09.000000	-
46	nova-conductor	node-2.domain.tld	internal	enabled	up	2017-03-30T06:15:11.000000	-
55	nova-compute	node-5.domain.tld	nova	enabled	up	2017-03-30T06:15:13.000000	-
59	nova-compute	node-4.domain.tld	nova	disabled	up	2017-03-30T06:14:13.000000	Disabled by Host HA
61	nova-compute	node-6.domain.tld	nova	enabled	up	2017-03-30T06:15:16.000000	-

用户 recover 此节点时，需要手动 enable 其 nova-compute 服务，如下图：

```
[root@node-1 ~]# nova service-enable node-4.domain.tld nova-compute
```

Host	Binary	Status
node-4.domain.tld	nova-compute	enabled

8. 根据多元组信息采取对应的动作见下表：

Power	虚拟机访问存储网络 (ceph-public)	管理网	生产网	当作存储节点	动作	备注
OFF					Evacuate	一个条件就够
ON	ON	OFF	ON		None	管理网不影响生产
ON	ON	ON	OFF	ON/OFF	Migrate	Migrate 虚机。需要 Dashboard 上云主机列表中确认 Migrate 还是 revert. 
ON	ON	OFF	Unknown		None	管理网 OFF，生产网卡状态未知，算作 ON
ON	ON	OFF	ON		None	这种情况，不能检测到虚机生产网卡 down，不会出现这种状态
ON	ON	OFF	OFF		None	这种情况，不能检测到虚机生产网卡 down，不会出现这种状态。
ON	OFF	ON	ON	ON	None	Ceph-public 断，会导致 nova-compute 为 down 状态，也是存储节点，不采取任何措施
ON	OFF	ON	ON	OFF	Power off Evacuate	Ceph-public 断，会导致 nova-compute 为 down 状态，只能 Evacuate.
ON	OFF	OFF		OFF	Power off Evacuate	不做存储节点，管理网 / 虚机访问后端存储，power off/evacuate
ON	OFF	OFF		ON	None	当作存储节点 不做操作
ON	OFF	ON	OFF	ON	None	Ceph-public 断，会导致 nova-compute 为 down 状态，也是存储节点，不采取任何措施。
ON	OFF	ON	OFF	OFF	Power off	Ceph-public 断，会导致

					Evacuate	nova-compute 为 down 状态，只能 Evacuate.
--	--	--	--	--	----------	--

■ 启动 Host HA 功能:

控制节点 HA: #crm resource start lenovo-hagent

控制节点非 HA: #systemctl restart lenovo-hagent

需要修改以下配置文件使 host ha 生效:

所有 controller 节点上修改/etc/lenovo/下的 hagent.conf、ipmi.conf 及 ping_list.conf 等文件。配置文件参考：



1. /etc/lenovo/hagent.conf

```
[DEFAULT]
use_rpc=False
debug = False
verbose=True
log_file = /var/log/nova/hagent.log
check_interval = 5          #监控的最小间隔
on_shared_storage = True    #目前只支持 True，当为 False 时不会采取任何 action。
evacuate_interval=1
action_in_same_aggregate=False
fault_hosts_number_threshold=2  #指同一时刻有问题的计算节点总数超过这个值，说明发生了整个平台
                                #的故障，不采取任何动作，如设置为 2，有三台出现问题就不处理。
service_down_time=60
dry_run=False               #当为 true 时，只检测不做任何 action，可用于测试
mixed_hosts=null            #定义融合节点（计算 + 存储）集合。当此项设置为 null 时，系统默认融合节
                                #点为计算节点，当某融合节点的存储网络不可用时系统会迁移其上虚拟机
                                #若用户定义了融合节点集合，则集合中某融合节点的存储网络不可用
                                #时系统不执行迁移操作。

[Auth]
username = admin
password = admin
auth_url = http://192.168.2.2:5000/v2.0
project_id = admin
region_name = RegionOne
```

```
[ping]
ping_list_file_path = /etc/lenovo/ping_list.conf
packet_count = 5      # 检测时的发包数目
packet_interval = 1 # 检测时的发包间隔
ip_item_format=MANAGEMENT_IP,CEPH_PUBLIC_IP
```

```
[ipmi]
conf_file_path = /etc/lenovo/ipmi.conf
separator=' '
```

2. /etc/lenovo/ipmi.conf

```
#please use IMM IP/username/password for this
#Host's IPMI access information the format like this
# {HC-node-name}={ipmi_address} {ipmi_username} {ipmi_password}
node-4.domain.tld=10.240.220.7 USERID PASSWORD
node-5.domain.tld=10.240.220.8 USERID PASSWORD
node-6.domain.tld=10.240.220.9 USERID PASSWORD
```

3. /etc/lenovo/ping_list.conf

```
# The list for the ping inspector
# Before the '=' is the node name.
# The IPs are seperated by ','
# ip_item_format = br-mgmt ip,br-storage ip
# The sequence of the IPs are specified by ip_item_format in the hagent.conf
node-4.domain.tld=192.168.2.6,192.168.4.5
node-5.domain.tld=192.168.2.7,192.168.4.6
node-6.domain.tld=192.168.2.8,192.168.4.7
```

注 若客户环境无法配 BMC 网络 ,可在所有控制节点上修改/etc/hwmgmt/hwmgmt.conf ,

置 ha_ipmi_enabled = False

```
[ha]
# for configure Host HA in dashboard

# If this value is changed, you should restart hwmgmt-api.service
# and lenovo-hagent.service on all controller nodes.
ha_ipmi_enabled = False
```

并重启 lenovo-hagent 及 hwmgmt-api 服务 :

控制节点 HA:

```
# crm resource start lenovo-hagent // 任一控制节点
```

```
# systemctl restart hwmgmt-api // 所有控制节点
```


控制节点非 HA: #systemctl restart lenovo-hagent & systemctl restart hwmgmt-api

■ 关闭 Host HA 功能:

```
#crm resource stop lenovo-hagent
```

5.1.7. 配置 VM HA 功能

■ 功能说明 :

1. 在云平台中有些业务并不能通过集群等手段实现高可用 ,需要对这类型单个虚拟机进行监控 ,出现故障如虚拟机内部 crash ,意外关机 ,根据预先设定策略 ,采取措施恢复虚拟机 ,即针对单个虚拟机的高可用方案。
2. VM HA 只运行在 compute 节点。

■ 启动 VM HA 功能:

```
#service lenovo-vagent start
```

需要在 compute 节点上修改配置文件/etc/lenovo/vagent.conf。配置文件参考 :



vagent.conf

```
[DEFAULT]
log_file= /var/log/nova/vagent.log
# log_level: CRITICAL, ERROR, WARNING, INFO, DEBUG
log_level = DEBUG
# enable task current supports ha, maintain
enable_tasks= ha

[Auth]
# Find configurations from /ect/nova/nova.conf on compute node
username = admin
password = admin
auth_url=http://192.168.0.2:5000/v2.0
project_id = admin

[HA]
on_shared_storage = True
check_interval = 20
```

```

# detect strategy , current supports ping,crashed, norunning
available_strategy = norunning

# how to combine the strategy ,supports all, any
strategy_combine= all

# what pre-actions before taking actual actions
# supports pre_notify, email
pre_actions =

# what post-actions before taking actual actions
# supports post_notify, email
post_actions =

# take action when service is up , supports reboot, migrate, rebuild, start
up_action = start

# take action when service is down, only supports evacuate
down_action = evacuate

[MAINTAIN]
check_interval = 30
flag_file_path = /etc/lenovo/agent_flag.conf

```

■ 关闭 Host HA 功能:

```
#service lenovo-vagent stop
```

5.1.8. 启用物理网络拓扑

5. 配置交换机机 ip

为环境中所有交换机配置一个 controller 节点可达的管理 ip。

例如，在 Lenovo Networking Operating System (NOS) G8272 交换机上配置如下: ip address 192.168.10.1/24

6. 设置交换机 lldp：确认所有的交换机已开启 lldp。

例如，在 Lenovo Networking Operating System (NOS) Version 10.4.2.0 G8272 交换机上

已默认开启 lldp，可通过 display lldp interface all 查看各端口 lldp 信息。

其他交换机需根据官方交换机配置文档开启 lldp(一般在 config 模式下使用 feature lldp 开启)

7. 设置交换机 snmp：在所有交换机上配置 snmp。

例如，在 Lenovo Networking Operating System (NOS) Version 10.4.2.0 G8272 交换机

上：

- 1) `snmp-server enable snmp`
- 2) `snmp-server version v1v2v3`
- 3) `snmp-server community public group network-operator`

8. 配置 proton 配置文件

在 controller 节点上，修改/etc/proton/proton.conf.

- 在[topology]中设置 `switch_ip_range`，代表 1 内配置的所有交换机管理 ip 地址

的范围，可以是一个列表，以逗号隔开，每个范围可以是一个 ip 地址，或者

用掩码表示的 ip 地址范围，或者 ip 地址 A 到 B 的范围，例如：

```
switch_ip_range = 192.168.10.1, 192.168.20/24, 192.168.30.1-192.168.30.10
```

- 在[topology]中设置 `snmp_version`

1) 若 `snmp_version` 为 2，则需要配置 `snmp_v2_community = public`

2) 若 `snmp_version` 为 3，则需要配置如下项：

```
snmp_v3_seclevel = authPriv
snmp_v3_authproto = MD5
snmp_v3_authpass = admin
snmp_v3_privproto = DES
snmp_v3_privpass = admin
snmp_v3_secname = admin
```

9. 重启 pronton-topology 服务

- 若 ThinkCloud 为 HA 环境，使用 `crm` 重启服务：

```
crm resource restart proton-topology
```

- 若 ThinkCloud 为 multi-node 环境，使用 `systemctl` 重启服务：

```
systemctl restart proton-topology.service
```

5.1.9. 修改云硬盘 1TB 上限

1. 在 3 个 controller 节点执行以下操作：

```
#vim /etc/openstack-dashboard/local_settings
```

最下面添加下面这项配置

```
MAX_VOLUME_SIZE = 100000000 # Volume 大小, MB
```

然后重启服务

```
#service httpd restart
```

2. 在任一 Controller 节点执行以下操作：

```
#source openrc
```

```
#keystone tenant-list //获取所要修改配额的 tenant id
```

```
#cinder quota-update --gigabytes -1 --backup-gigabytes -1 --volumes -1
```

```
--snapshots -1 --backups -1 xxxx //此 xxxx 为 tenant id
```

5.1.10. 启用 flat 网络

1. 在控制节点和计算节点上，修改 /etc/neutron/plugins/ml2/ml2_conf.ini

- 1) 将 type_drivers = vlan,flat,local 修改为 type_drivers = flat,vlan,local

- 2) 将 tenant_network_types = vlan 修改为 tenant_network_types = flat

- 3) 将 mechanism_drivers=openvswitch,lenovo 修改为 mechanism_drivers=openvswitch

- 4) 在控制节点执行 systemctl restart neutron-server

在计算节点执行 systemctl restart neutron-openvswitch-agent

2. 配置修改完之后，便可以开始创建 flat 网络

进入某个 controller 执行以下命令：

```
# source /root/openrc
```

```
# neutron net-create flat_network_name --provider:network_type flat  
--provider:physical_network physnet2 --shared
```

对于参数 physnet2 需要根据 /etc/neutron/plugins/ml2/openvswitch_agent.ini

中的 bridge_mappings =physnet1:br-ex,physnet2:br-prv

physnet2 指向了 br-prv ,而我们的又将 private 网络配置为 flat 网络 ,因此 这个地方跟

参为 physnet2

3. 修改保存后即可在 UI 上这个网络下创建子网 , 需要注意的是创建子网时 , 不要 enable DHCP 和网关

5.1.11. 允许域账户登录

系统默认用户邮箱登录 , 若需要通过域账户登录 , 可在所有控制节点上设置 :

```
#vim /etc/openstack-dashboard/local_settings
DOMAIN_LOGIN_ENABLE = True
#service httpd restart
```

若域账户登录过程中出现“MemcachedKeyLengthError: Key length is > 250”的错误 , 可编辑

/etc/openstack-dashboard/local_settings , 可替换 memcached 配置项如下:

```
def hash_key(key, key_prefix, version):
    new_key = ':'.join([key_prefix, str(version), key])
    if len(new_key) > 250:
        import hashlib
        m = hashlib.md5()
        m.update(new_key)
        new_key = m.hexdigest()
    return new_key
CACHES = {
    'default': {
        'BACKEND': 'django.core.cache.backends.memcached.MemcachedCache',
        'LOCATION': '127.0.0.1:11211',
        'KEY_FUNCTION': hash_key,
    }
}
```

5.1.12. 开启 Ceph 存储空间回收通知机制

在 OpenStack 云环境的部署中 , Ceph 存储具备 Thin provision 的功能 , 这项功能实现了存储按需分配的能力。以 RBD image 为例 , 它本身是稀疏格式的 , 也就是说它所占用 objects 会随着用户写入数据的增加而增加 (Thin provision)。当用户删除数据以后 , 这些 object

不再使用,但并没有被释放。因为从 Ceph 的角度讲,它并不知道文件系统中发生的事情。

若要回收,需要开启回收通知机制,步骤如下:

1. 给所有镜像加 hw_scsi_model 和 hw_disk_bus 两个属性:

```
#glance image-update --property hw_scsi_model=virtio-scsi --property hw_disk_bus=scsi
xxxxx
```

2. 编辑各计算节点上的 nova.conf,在[libvirt]中添加:

```
[libvirt]
...
hw_disk_discard = unmap
```

并重启 nova-compute 服务

3. 在对接 ceph 的 cinder 节点 (ThinkCloud OpenStack 默认位于控制节点) 的

/etc/cinder/cinder.conf 的 ceph backend section 里添加:

```
...
volume_driver=cinder.volume.drivers.rbd.RBDDriver
report_discard_supported=True
...
```

并重启 cinder-volume 服务。

4. 在虚机中执行 fstrim 通知 ceph 回收空间:

```
#fstrim -v mountpoint_of_ceph_hdd
```

计算节点上改了之后要重启 nova-compute 服务, cinder 要重启 cinder-volume

系统默认用户邮箱登录,若需要通过域账户登录,可在所有控制节点上设置:

```
#vim /etc/openstack-dashboard/local_settings
DOMAIN_LOGIN_ENABLE = True
#service httpd restart
```

5.2. 系统优化选项

5.2.1. 为 10GB 网络修改 MTU 到 9000

1. 在控制节点上 执行 ovs-vsctl show 查看 所有的 management 网络 是通过 bond 网

卡是哪几块,假设是 bond0 绑定了 eth4, eth6, 则 编辑修改

/etc/sysconfig/network-scripts/下的 eth4,eth6 , 以及 br-mgmt 在其后面添加配置项
MTU=9000

2. 修改完后, 执行 `systemctl restart network`。查看 ovs 是否自动识别到新的 mtu 值 ,

执行 `ovs-vsctl get int br-mgmt mtu`

返回结果为 9000,否则执行 `ovs-vsctl set int br-mgmt mtu=9000`

由于控制节点没有接入 storage 网络, 所以 storage 无需配置。

3. 在计算节点和 ceph 节点上 执行 `ovs-vsctl show` 同样查看网卡绑定关系。假设得到

关系为 br-mgmt 为 bond0 绑定了 eth4,eth6 ; br-storage 为 bond1 绑定了 eth5,eth7 ,

因此, 需要修改 MTU=9000 的网卡为 eth4,eth6,eth5,eth7 br-storage,br-mgmt , 修改

完后执行 `systemctl restart network` 同样执行 `ovs-vsctl get int br-mgmt mtu` 和

`ovs-vsctl get int br-storage mtu` 来确认 网桥的 MTU 是否被修改。

4. 执行完以上操作, 下面需要验证是否生效。通过在 host 上执行 ping 命令来查看 :

`ping -I TARGET_IFC -M do -s 8972 TARGET_IP` 其中 TARGET_IFC 代表网卡而

TARGET_IP 表示目标 IP

例如 : `ping -I br-mgmt -M do -s 8972 192.168.2.8`

5.2.2. 调整 Ceph pg_num

当 ceph 集群中仅少量 osd 提示 full 或 near full,可能是 pg_num 设置不合理造成的.ceph 的数据存储结构应该都很容易查到。就是 file->object->pg->OSD->physics disk 。因此, 一旦这里的 pg 数设置过小, pg 到 OSD 的映射不均匀就会造成 OSD 上分配到的数据不均匀。这种解决方法就是重新调整 pg_num 和 pgp_num 。

1. 执行如下命令查看 ceph osd 上 pg 分布情况

```
ceph pg dump | awk '
/^pg_stat/{ col=1; while($col!="up"){col++; col++ }
```

```

/^([0-9a-f]+\.[0-9a-f]+)/ { match($0,/^[0-9a-f]+/); pool=substr($0, RSTART, RLENGTH);
poollist[pool]=0;
up=$col; i=0; RSTART=0; RLENGTH=0; delete osds; while(match(up,/([0-9]+)/)>0)
{ osds[++i]=substr(up,RSTART,RLENGTH); up = substr(up, RSTART+RLENGTH) }
for(i in osds) {array[osds[i],pool]++; osdlist[osds[i]];}
}
END {
printf("\n");
printf("pool : \t"); for (i in poollist) printf("%s\t",i); printf(" | SUM \n");
for (i in poollist) printf("-----"); printf("-----\n");
for (i in osdlist) { printf("osd.%i\t", i); sum=0;
for (j in poollist) { printf("%i\t", array[i,j]); sum+=array[i,j]; poollist[j]+=array[i,j] };
printf(" | %i\n",sum) }
for (i in poollist) printf("-----"); printf("-----\n");
printf("SUM : \t"); for (i in poollist) printf("%s\t",poollist[i]); printf(" |\n");
}'

```

2. 推荐的 ceph pool pg num

参考: <https://ceph.com/pgcalc/>

每个 pool 的 pg num 的经验计算公式: (100*OSD 盘数)/(副本数*pool 数),向上取最接近的 2 的倍数. 注: 这里的 pool 数仅计算真实数据的 pool.

3. 调整数据同步参数, 减少数据同步时对业务的影响

当调整 PG/PGP 的值时, 会引发 ceph 集群的 backfill 操作, 数据会以最快的数据进行平衡, 因此可能导致集群不稳定. 因此首先设置 backfill ratio 到一个比较小的值,

通过下面的命令设置:

```

# ceph tell osd.* injectargs '--osd-max-backfills 1'
# ceph tell osd.* injectargs '--osd-recovery-max-active 1'
# ceph tell osd.* injectargs '--osd_recovery_max_single_start 1'

```

注: ThinkCloud OpenStack 中已预设为以上值

4. 平滑调整 ceph pool 的 pg num

(1) 检查 pool 的 pg num, pgp num, 及复制 size, 执行如下命令:

```

# ceph osd dump | grep size | grep volumes
pool 2 'volumes' replicated size 3 min_size 2 crush_ruleset 0 object_hash rjenkins
pg_num 128 pgp_num 128 last_change 45 flags hashpspool stripe_width 0

```


(2) 使用上述公式 , 根据 OSD 数量、复制 size、pool 的数量 , 计算出新的 PG 数量 , 假设是 1024.

(3) 按 2 的倍数逐渐平滑增大 pool 的 pg_num 和 pgp_num: 256->512->1024 :

```
# ceph osd pool set volumes pg_num 256
```

```
-----等待 pg 创建完成-----
```

```
# ceph osd pool set volumes pgp_num 256
```

```
-----ceph -s 查看待 recover 进度完成-----
```

```
# ceph osd pool set volumes pg_num 512
```

```
-----等待 pg 创建完成-----
```

```
# ceph osd pool set volumes pgp_num 512
```

```
-----ceph -s 查看待 recover 进度完成-----
```

```
# ceph osd pool set volumes pg_num 1024
```

```
-----等待 pg 创建完成-----
```

```
# ceph osd pool set volumes pgp_num 1024
```

```
-----ceph -s 查看待 recover 进度完成-----
```

(4) 如果有其他 pool , 同步调整它们的 pg_num 和 pgp_num , 以使负载更加均衡。

5.3. 系统运维常用操作

5.3.1. 安全开关云平台

1. 通过 Horizon web 管理界面或者命令行关闭虚拟机

亦可在控制节点命令行模式下参考以下命令批量关机 :

```
for ins in `mysql -e "select uuid from nova.instances where vm_state='active' "|awk '{print $1}'`;do if [[ "$ins" =~ "uuid" ]];then continue;fi;nova stop $ins;done
```

注意 : 请确认虚拟机之间是否有特殊的关闭顺序要求。

2. 关闭 Ceph OSD 节点

设置 noout 标记，防止因为 Ceph 节点断电引起重平衡

(1) 登陆到 Controller 节点或者 Ceph OSD 节点，执行：

```
ceph osd set noout
```

(2) 确认集群状态，可以看到 noout 标记

```
# ceph -s
cluster 7eb2b84a-c73e-4050-b735-3ba643ffc603
health HEALTH_WARN
noout flag(s) set
```

(3) 登陆到每个 Ceph OSD 节点，执行：init 0

3. 关闭计算节点

登陆每一个计算节点，执行: init 0

4. 关闭控制节点

对 3 个控制节点，判断哪一台是主控制节点，首先关闭其他 2 台控制节点，最后关闭主控制节点，并且记住关闭的顺序，后面启动的时候需要相反的顺序启动：

(1) 首先判断哪一台是主控制节点 若 web ip 为 172.16.1.2, 则在控制节点上执行 ip -a | grep 172.16.1.3，有输出即为主控节点。

(2) 登陆到其他两台控制节点，分别执行：

```
init 0
```

每个节点关闭完成，多等几分钟，使集群有充足的时间在剩余的控制节点之间重新分配服务。

(3) 最后登陆到 controller01 节点，执行：init 0

至此，集群已经安全关闭。

重启集群过程正好相反，先启动控制节点，再启动 Ceph 节点，最后启动计算节点，启动计算节点后再登录 web 管理界面将相应的虚拟机启动。

需要特别注意的是，启动控制节点的时候，按照关闭相反的顺序进行启动。

5.3.2. 通过命令行上传大镜像文件

当镜像文件超大时，通过 Horizon web 可能上传不成功，可尝试通过下面的命令行上传：

```
# source /root/openrc.v2
# glance image-create --disk-format raw --container-format bare --disk-format qcow2
--name centos7 --is-public True --file CentOS7_all_tools.qcow2 --progress
```

5.3.3. 替换 Ceph OSD 数据盘

1. 所有控制节点上停掉以下服务：

```
#systemctl stop storagemgmt-agent
#systemctl stop openstack-gnocchi-metricd
#systemctl stop openstack-gnocchi-statsd
```

2. 在某一控制节点上操作 Ceph 进入维护模式

```
#ceph osd set noout
```

3. 在对应的 ceph osd 节点上移除待替换的 osd

```
#/etc/init.d/ceph osd -a stop osd.x
#ceph osd down osd.x
#ceph osd out osd.x
#ceph osd crush rm osd.x
#ceph auth del osd.x
#ceph osd rm osd.x
```

注：需要等待 ceph -s 为 HEALTH_OK 再继续剩余步骤。

4. 检查待替换的盘上是否存在软 RAID，若有，删除之

```
#mdadm --detail /dev/md0
#mdadm /dev/md0 --fail /dev/sdX3 --remove /dev/sdX3
```

5. 替换新盘并格式化

```
#ceph-disk zap /dev/sdx
```

或者

```
#parted /dev/sdx
#>mklable gpt
```

可用 ceph-disk list 看状态

6. 在某一控制节点上操作创建 ceph osd

```
#ceph-deploy osd create node-x:/dev/sdx:/dev/sd?
```

冒号前面为 osd 分区, 后面为日志分区(可选), 如 ceph-deploy osd create

```
node-5:/dev/sdc:/dev/sde7
```

7. 检查新的 osd 盘是否自动挂载

登录 osd 节点, df -h 查看新的 ceph-osd.x 分区是否自动挂载, 若无,如下方式手动挂载:

```
#mount /dev/sdx1 /var/lib/ceph/osd/ceph-x (注: 若替换 ceph osd 盘, 确保之前的分区先 umount )
```

8. 检查新的 ceph osd 分区服务是否启动

登录 osd 节点, systemctl list-units | grep osd.x.查看是否有相关服务 running,若无 ,执行

```
#/etc/init.d/ceph -a start osd.x
```

9. 确认以下信息:

- ✓ ceph -s 查看 ceph 状态, 确保集群的状态为 HEALTH OK
- ✓ 所有 PG 都是 active + clean
- ✓ 所有 OSD 都是 up 状态

10. 退出维护模式

```
#ceph osd unset noout
```

11. 恢复停掉的服务

```
#systemctl restart storagemgmt-agent
#systemctl restart openstack-gnocchi-metricd
#systemctl restart openstack-gnocchi-statsd
```

注: 若替换多张盘, 必须 one by one 重复步骤 3~9

5.4. 更新 LOE 许可证

更新许可证方法如下:

1. 在 Carrier 节点上, 利用上传 License 的方式上传。

2. 在所有控制节点上：

- 1) 使用下列命令更新 license。

```
wget http://\${master\_ip}:8080/licenses/tcos.lic -O /etc/lenovo/tcos.lic
```

- 2) 重启 keystone 服务:

```
#service openstack-keystone restart
```

- 3) 重启 httpd 服务:

```
#service httpd restart
```