

组网及说明

不涉及

问题描述

该问题前台的表现可能是：前台界面登录异常、下发或者删除虚拟机任务卡住等。后台看，rabbitmq的存储卷100%占满。

```
tmpfs          6.5G   0   6.5G   0% /run/user/0
/dev/mapper/CoreAPI  99G   61M   94G   1% /var/lib/kubelet/plugins/kubernetes.io/iscsi/8.8.8:3260-iqn.2011-11:dss.coreapi-lun-0
/dev/mapper/webApp  50G   53M   47G   1% /var/lib/kubelet/plugins/kubernetes.io/iscsi/8.8.8:3260-iqn.2011-11:dss.webapp-lun-0
/dev/mapper/PostgreSQL  79G   675M   74G   1% /var/lib/kubelet/plugins/kubernetes.io/iscsi/8.8.8:3260-iqn.2011-11:dss.postgresql-lun-0
/dev/mapper/RabbitMQ  9.8G   9.2G   19M  100% /var/lib/kubelet/plugins/kubernetes.io/iscsi/8.8.8:3260-iqn.2011-11:dss.rabbitmq-lun-0
[root@cloudoscluster1 ~]#
```

过程分析

一般cloudos2.0的rabbitmq的存储卷100%占满有两个原因

- 1、由于cloudos管理的虚拟机数量庞大（一般认为多于1000就算规模比较大的），导致rabbitmq的消息队列产生大量堆积，从而占满。
- 2、由于存储性能，存储链路等故障问题，导致rabbitmq存储卷损坏，持续有报错日志打印，占满该卷。尤其是cloudos2.0早期开局不规范的局点，将cloudos部署在虚拟机上，使用虚拟机环境的共享存储出现该问题的概率较大。

本案例只讨论第二种情况，即由于rabbitmq存储卷损坏导致该卷占满的情况的处理方法。

解决方法

对于rabbitmq存储卷损坏导致该卷占满的情况的处理，一般是将这个卷进行格式化处理的临时解决方案（具体如下）。彻底解决包括两个方面，一是提高存储性能，确保存储链路稳定，规范开局，不要使用虚拟机部署生产环境的cloudos，二是待1139H07版本发布之后，可以考虑升级到此版本，此版本增加了rabbitmq存储卷的检测机制，能够及时自动处理占满的情况。下面来看一下将这个卷进行格式化处理的临时解决方案。

- 1、先进行查看环境中前台其他的pod是否是running的，如不是，请及时处理
- 2、df -h记录一下这个卷的挂载的路径。如下是/dev/mapper/RabbitMQ

```
/dev/mapper/PostgreSQL  /9G  6.5M  /4G  1% /var/lib/kubelet/plugins/kubernetes.io/iscsi/8.8.8:3260-iqn.2011-11:dss.postgresql-lun-0
/dev/mapper/RabbitMQ    9.8G  9.2G  19M  100% /var/lib/kubelet/plugins/kubernetes.io/iscsi/8.8.8:3260-iqn.2011-11:dss.rabbitmq-lun-0
[root@cloudoscluster1 ~]#
```

- 3、使用/opt/bin/kubectl -s 127.0.0.1:8888 scale --replicas=0 rc rabbitmqrc命令，将rabbitmq的 pod置0，暂停这个pod。
- 4、将这个卷格式化，mkfs.ext4 /dev/mapper/RabbitMQ（步骤2中查到的），具体如下：

```
you have new mail in /var/spool/mail/root
[root@CloudosCluster1 ~]# mkfs.ext4 /dev/mapper/RabbitMQ
mke2fs 1.42.9 (28-Dec-2015)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
355360 inodes, 2621440 blocks
131072 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=2151677952
30 block groups
32768 blocks per group, 32768 fragments per group
3192 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632

Allocating group tables: done
writing inode tables: done
creating journal (32768 blocks): done
writing superblocks and filesystem accounting information: done
```

- 5、使用/opt/bin/kubectl -s 127.0.0.1:8888 scale --replicas=1 rc rabbitmqrc命令，将rabbitmq的 pod置1，启动这个pod。
- 6、等大概2分钟，等rabbitmq的pod起来之后，/opt/bin/kubectl --server=127.0.0.1:8888 get pod -o wide命令看一下，所有的pod是否都是running的。
- 7、重新检查cloudos相关功能是否恢复。