Azure共享磁盘在“SLES for SAP / SLE HA 15 SP2”上的实例场景

Original Zhiqiang Zhou SUSE订阅 Today

> 2020 年 7 月 [1] 微软公有云Azure共享磁盘 正式支持 SUSE Linux Enterprise Server for SAP 和 SUSE Linux Enterprise High Availability Extension 15 SP1 及以上版本。借助这一新功能，它为云环境中的关键任务应用程序（例如，SAP 工作负载）提供了更大的灵活性。Azure 共享磁盘为微软公有云虚拟机里面运行的SUSE Linux 企业服务器操作系统提供高性能存储，在此基础上SUSE Linux 企业服务器高可用性产品进一步增加容错能力。

在概念上，微软 Azure 共享磁盘与其他本地的传统共享磁盘技术没有什么不同。本文主要参照最新的 SLE HA 15 SP2 管理指南[2]，针对 微软公有云Azure 环境优化调整，实例演练下面两个场景：

- Active-Passive NFS server
- Active-Active OCFS2 集群文件系统

注:本文使用了如下一些缩略词：
“SLES”为“SUSE Linux Enterprise Server”
“SLES for SAP”为“SUSE Linux Enterprise Server for SAP”
“SLE HA”为“SUSE Linux Enterprise High Availability Extension”
“SBD” 为STONITH Block Device

Azure云环境准备

检查 Azure Cloud Shell环境

本地命令行工具`azure-cli`的版本必须是2.3.1 及以上版本

```
suse@tumbleweed:~> az --version
```

也可以直接使用云端环境 https://shell.azure.com ， 云端环境基本能用，只是对Linux管理员来说太不灵活。

创建两个虚拟机和一个共享磁盘

1. 从微软Azure云市场获取 SUSE 镜像的URN

```
URN=`az vm image list --publisher SUSE -f sles-sap-15-sp2-byos \
--sku gen2 --all --query "[-1].urn"|tr -d '"'`; echo $URN
```

返回值是最新SLES for SAP 15 SP2的URN， 在后面的命令行中使用到它:
SUSE:sles-sap-15-sp2-byos:gen2:2020.09.21

2. 从无到有，创建两个虚拟机

注：目前Azure云上，所有提供 managed disks 的区域都支持 Azure 共享磁盘[3]，

```
LC=westus2
RG=asd_${LC}_001

az group create --name $RG --location $LC
az ppg create -n ppg_$RG -g $RG -l $LC -t standard

az vm create --resource-group $RG --image $URN --ppg ppg_$RG \
  --admin-username $USER --ssh-key-values ~/.ssh/id_rsa.pub \
  --size Standard_D2s_v3 --name asd-sles15sp2-n1

az vm create --resource-group $RG --image $URN --ppg ppg_$RG \
  --admin-username $USER --ssh-key-values ~/.ssh/id_rsa.pub \
  --size Standard_D2s_v3 --name asd-sles15sp2-n2
```

3. 创建Azure共享磁盘并添加到虚拟机里面

这里创建两个共享磁盘，一个用做数据盘，一个用做SBD盘。对于IO负载很重的集群，使用有独立IO路径的一个或多个SBD盘是必须的。SBD盘尺寸很小，在Azure计费规则下，独立的SBD盘费用低廉。

```
DN=asd_shared_disk_152_sbd
az disk create -g $RG -n $DN -z 256 --sku Premium_LRS --max-shares 2
diskId=$(az disk show -g $RG -n $DN --query 'id' -o tsv); echo $diskId
az vm disk attach -g $RG --name $diskId --cachin None --vm-name asd-sles15sp2-n1
az vm disk attach -g $RG --name $diskId --cachin None --vm-name asd-sles15sp2-n2

DN=asd_shared_disk_152_data
az disk create -g $RG -n $DN -z 256 --sku Premium_LRS --max-shares 2
diskId=$(az disk show -g $RG -n $DN --query 'id' -o tsv); echo $diskId
az vm disk attach -g $RG --name $diskId --cachin None --vm-name asd-sles15sp2-n1
az vm disk attach -g $RG --name $diskId --cachin None --vm-name asd-sles15sp2-n2
```

Azure里面创建HA集群

1. 更新SUSE Linux 企业服务器操作系统补丁

初次进入虚拟机操作系统，第一件事就是更新SUSE Linux 企业服务器操作系统补丁:

```
sudo SUSEConnect -r $REGCODE
sudo zypper up -y
sudo reboot
```

注：订阅用户注册码可以从官方网站获得 https://scc.suse.com

2. SBD 必须使用看门狗

需要在所有节点上配置，并加载软狗：

```
sudo modprobe softdog; echo "softdog"|sudo tee /etc/modules-load.d/softdog.conf
```

注：共有云上不提供硬狗，只有使用软狗

验证软狗是否加载成功:

```
sudo sbd query-watchdog
```

3. 在任一个节点上准备SBD 分区

SBD磁盘只需要一个很小的4兆空间[REF：SLE HA指南– SBD分区][4]。用 `lsblk` 命令查看确认使用正确的磁盘设备名 “/dev/sdX”, 注意OS重启后磁盘设备名可能因配置而变化，SUSE Linux尽最大努力但不保证磁盘设备名不变:

```
sudo lsblk
```

因此，磁盘分区时特意采用Label标签：

```
sudo parted /dev/sdc mklabel GPT
sudo parted /dev/sdc mkpart sbd-sles152 1MiB 5MiB

sudo parted /dev/sdd mklabel GPT
sudo parted /dev/sdd mkpart asd-data1 10GiB 20GiB
sudo parted /dev/sdd mkpart asd-data2 20GiB 30GiB
```

在其他节点上验证前面的分区设置：

```
sudo partprobe; sleep 5; sudo ls -l /dev/disk/by-partlabel/
```

引导创建一个基本的HA集群

1. 手动配置各个节点上的root超级用户，并确认ssh可以免密码访问

在所有节点上，运行：

```
sudo ssh-keygen -q -t rsa -N '' -f /root/.ssh/id_rsa <<&1 >/dev/null
```

然后，把所有节点上的 /root/.ssh/id_rsa.pub 追加入 /root/.ssh/authorized_keys，
进一步确证**ssh**可以免密码访问：

```
asd-sles15sp2-n1:~> sudo ssh asd-sles15sp2-n2
asd-sles15sp2-n2:~> sudo ssh asd-sles15sp2-n1
```

2. 引导创建一个最基本的集群

```
asd-sles15sp2-n1:~> sudo crm cluster init -y -u -s /dev/disk/by-partlabel/sbd-sles152 -A 10.0.0.9
```

注：本文专注于Azure共享磁盘，囿于篇幅，不展开介绍如何设置Azure负载均衡器来配合VIP (10.0.0.9) 生效。

上一步在节点1上完成后，让节点2加入集群：

```
asd-sles15sp2-n2:~> sudo crm cluster join -y -c asd-sles15sp2-n1
```

开始监控集群的状态：

```
asd-sles15sp2-n1:~> sudo crm_mon -rR
```

3. 优化Azure云里面的集群
步骤：修改SBD配置

在虚拟机环境中的OS重启通常非常快，为了减少集群业务的抖动，修改SBD_DELAY_START为“yes”是必要的，并且这里需要相应修改sbd.service里面的”TimeoutSec=”值。它带来的副作用会延迟集群启动，例如本文例子会有2分多钟的延迟，集群RTO也随之加大。

```
sudo augtool -s set /files/etc/sysconfig/sbd/SBD_DELAY_START yes
```

可以手动在所有节点上修改，也可用csync2工具把改动部署到所有节点上：

```
sudo csync2 -xv
```

在所有节点上，手动加入systemd sbd 配置文件来调整“TimeoutSec=”：

```
echo -e "[Service]\nTimeoutSec=144" | sudo tee
/etc/systemd/system/sbd.service.d/sbd_delay_
start.conf
sudo systemctl daemon-reload
```

```
[Service]
TimeoutSec=144
```

步骤：优化SBD在磁盘上的元数据

磁盘上SBD watchdog timeout元数据会被 SBD 后台进程用来初始化看门狗驱动，缺省值是5秒。为了集群更健壮地应对微软公有云上有计划的维护活动，比如给虚拟机和网络带来的抖动，可以考虑适当增加watchdog timeout元数据为60秒，相应的msgwait 元数据为120秒。 [REF: SLE HA Guide – Setting Up SBD with Devices][5]

修改SBD磁盘的元数据，需要在一个节点上重新创建SBD磁盘：

```
SBD_DEVICE=/dev/disk/by-partlabel/sbd-sles152
sudo sbd -d ${SBD_DEVICE} -1 60 -4 120 create
```

验证SBD磁盘的元数据：

```
sudo sbd -d ${SBD_DEVICE} dump
```

为了重新初始化看门狗驱动，必须重新启动 sbd 后台进程：

```
sudo crm cluster run "crm cluster restart"
```

步骤：测试SBD stonith

集群运行起来， 可以继续测试一下SBD的功能，比如重启机器：[REF: SLE HA Guide – Testing SBD and Fencing][6]:

```
SBD_DEVICE=/dev/disk/by-partlabel/sbd-sles152
sudo sbd -d ${SBD_DEVICE} message asd-sles15sp2-n2 reset
```

步骤：优化corosync 超时参数

和SBD类似，为了集群更健壮地应对微软公有云上有计划的维护活动，比如给虚拟机和网络带来的抖动，可以考虑适当增加corosync token 超时参数。但是要注意，碰上真的永久性故障，这会牺牲服务器自恢复的时间（RTO），因为理论上corosync 需要更长的时间判定错误。为此，修改corosync token 超时参数为30秒可能是合适的，相应的 consensus 超时参数修改为36秒。d

在所有节点上修改corosync.conf ，可参考`man corosync.conf`：

```
sudo vi /etc/corosync/corosync.conf
```

```
token: 30000
consensus: 36000
```

可以手动在所有节点上修改，也可用csync2工具把改动部署到所有节点上：

```
sudo csync2 -xv
```

在一个节点上运行下面的命令，使得所有节点重新加载corosync的改动：

```
sudo corosync-cfgtool -R
```

验证参数的改动：

```
sudo /usr/sbin/corosync-cmapctl |grep -w -e totem.token -e totem.consensus
```

Active-Passive NFS server

警告：这里的**Active-Passive NFS server**是一种弱实现。在商业环境里面，如果要保证强语义的锁机制，则需要更为高级的实现。

1. 创建逻辑卷和文件系统

在一个节点上，执行下面的步骤。

步骤：修改lvm2的配置
编辑：

```
sudo vi /etc/lvm/lvm.conf
```

设置 `system_id_source = “uname”`。缺省值是“none”，这意味着 lvm 会屏蔽所有带有 systemid 的逻辑卷。

设置 `auto_activation_volume_list = []`，以回避操作系统启动时自动激活逻辑卷。事实上，HA LVM所用到的逻辑卷必须得由Pacemaker集群管理来激活。对于 `shared` 逻辑卷（也叫 `lvmlockd` 逻辑卷)，不用关心这个选项，因为没有集群栈不能激活。

验证一下：

```
sudo lvmconfig global/system_id_source
sudo lvmconfig activation/auto_activation_volume_list
```

```
system_id_source=”uname”
auto_activation_volume_list=[]
```

可以手动在所有节点上修改，也可用csync2工具把改动部署到所有节点上：

```
sudo csync2 -xv
```

步骤：创建逻辑卷

```
sudo pvcreate /dev/disk/by-partlabel/asd-data1
sudo vgcreate vg1 /dev/disk/by-partlabel/asd-data1
sudo lvcreate -l 50%VG -n lv1 vg1
```

步骤：初始化文件系统

```
sudo mkfs.xfs /dev/vg1/lv1
```

2. 优化所有节点上的NFS server配置

调整NFSV4LEASETIME 参数，合理减少故障切换时间。在所有节点上更改 /etc/sysconfig/nfs：

```
sudo augtool -s set /files/etc/sysconfig/nfs/NFSV4LEASETIME 60
```

3. 初始引导NFS server

在一个节点上运行如下命令：

```
sudo crm configure \
primitive p_nfsserver systemd:nfs-server \
        op monitor interval=30s

sudo crm configure \
primitive p_vg1 LVM-activate \
        params vgname=vg1 vg_access_mode=system_id \
        op start timeout=90s interval=0 \
        op stop timeout=90s interval=0 \
        op monitor interval=30s timeout=90s

sudo crm configure \
primitive p_fs Filesystem \
        op monitor interval=30s \
        op_params OCF_CHECK_LEVEL=20 \
        params device="/dev/vg1/lv1" directory="/srv/nfs" fstype=xfs
```

注：如果‘directory=’ 指定的目录不存在，`Filesystem`资源会主动创建

```
sudo crm configure \
primitive p_exportfs exportfs \
        op monitor interval=30s \
        params clientspec="*" directory="/srv/nfs" fsid=1 \
        options="rw,mp" wait_for_leasetime_on_stop=true
```

注：设计上，Pacemaker会试图把业务资源分散到多个节点，比如 `p_vg1 p_fs p_nfsserver p_exportfs admin-ip`。在没有把他们最后配置到资源组之前，如下面的步骤，集群可能会报告一些假的错误消息。

```
sudo crm configure group g_nfs p_vg1 p_fs p_nfsserver p_exportfs admin-ip
```

清除集群里面那些假的错误消息：

```
sudo crm_resource -C
```

验证一下NFS server：

```
suse@asd-sles15sp2-n2:~> sudo showmount -e asd-sles15sp2-n1
suse@asd-sles15sp2-n2:~> sudo showmount -e asd-sles15sp2-n2

 Export list for asd-sles15sp2-n1:
 /srv/nfs *
```

Active-Active OCFS2 集群文件系统

1. 启动`dlm` and `lvmlockd` 后台进程

这个步骤必须在创建`shared` 逻辑卷之前：[REF: SLE HA Guide] [7]。在一个节点上运行：

```
sudo crm configure \
primitive dlm ocf:pacemaker:controld \
        op monitor interval=60 timeout=60

sudo crm configure \
primitive lvmlockd lvmlockd \
        op start timeout=90 interval=0 \
        op stop timeout=90 interval=0 \
        op monitor interval=30 timeout=90

sudo crm configure group g_ocfs2 dlm lvmlockd
sudo crm configure clone c_ocfs2 g_ocfs2 meta interleave=true
```

2. 在一个节点上创建`shared`逻辑卷磁盘

```
sudo ls -l /dev/disk/by-partlabel/
sudo pvcreate /dev/disk/by-partlabel/asd-data2
sudo vgcreate --shared vg2-shared /dev/disk/by-partlabel/asd-data2
sudo lvcreate -an -l 50%VG -n lv1 vg2-shared

sudo crm configure \
primitive p_vg_shared LVM-activate \
                        params   vgname=vg2-shared   vg_access_mode=lvmlockd
activation_mode=shared \
        op start timeout=90s interval=0 \
        op stop timeout=90s interval=0 \
        op monitor interval=30s timeout=90s

sudo crm configure modgroup g_ocfs2 add p_vg_shared
```

3. 在一个节点上创建ocfs2

[REF: SLE HA Guide] [8]

```
sudo mkfs.ocfs2 /dev/vg2-shared/lv1
```

4. 最后把ocfs2运行在所有节点上

[REF: SLE HA Guide] [9]

```
sudo crm configure \
primitive p_ocfs2 Filesystem \
        params device="/dev/vg2-shared/lv1" directory="/srv/ocfs2" fstype=ocfs2 \
        op monitor interval=20 timeout=40 \
        op_params OCF_CHECK_LEVEL=20 \
        op start timeout=60 interval=0 \
        op stop timeout=60 interval=0

sudo crm configure modgroup g_ocfs2 add p_ocfs2
```

在一个节点上写入文本文件：

```
asd-sles15sp2-n1:~> echo "'Hello' from `hostname`" | sudo tee /srv/ocfs2/hello_world
```

在另外的节点上验证：

```
asd-sles15sp2-n2:~> cat /srv/ocfs2/hello_world
```

自此，轻松完成两个实例！

[1] https://azure.microsoft.com/en-us/blog/announcing-the-general-availability-of-azure-shared-disks-and-new-azure-disk-storage-enhancements/
[2] https://documentation.suse.com/sle-ha/15-SP2/html/SLE-HA-all/book-sleha-guide.html
[3] https://docs.microsoft.com/en-us/azure/virtual-machines/disks-shared
[4] https://documentation.suse.com/sle-ha/15-SP2/html/SLE-HA-all/cha-ha-storage-protect.html?&_ga=2.148667377.773351976.1606302998-1142908039.1605698094#sec-ha-storage-protect-overview
[5] https://documentation.suse.com/sle-ha/15-SP2/html/SLE-HA-all/cha-ha-storage-protect.html?&_ga=2.79958225.773351976.1606302998-1142908039.1605698094#sec-ha-storage-protect-fencing-setup
[6] https://documentation.suse.com/sle-ha/15-SP2/html/SLE-HA-all/cha-ha-storage-protect.html?&_ga=2.189912173.773351976.1606302998-1142908039.1605698094#sec-ha-storage-protect-test
[7] https://documentation.suse.com/sle-ha/15-SP2/html/SLE-HA-all/cha-ha-clvm.html?&_ga=2.86126814.773351976.1606302998-1142908039.1605698094#sec-ha-clvm-config
[8] https://documentation.suse.com/sle-ha/15-SP2/html/SLE-HA-all/cha-ha-ocfs2.html?&_ga=2.89838941.773351976.1606302998-1142908039.1605698094#sec-ha-ocfs2-create
[9] https://documentation.suse.com/sle-ha/15-SP2/html/SLE-HA-all/cha-ha-ocfs2.html?_ga=2.257023053.773351976.1606302998-1142908039.1605698094#sec-ha-ocfs2-mount

往期回顾

- SUSECON Digital—SLES for SAP and HA
- SUSE在SAP蓝宝石大会：成功的 S/4HANA项目的首选平台
- SQL Server的Linux之旅—第3部分：Azure选项

SUSE开源先锋已上线

扫码关注我们

您将收获最新的开源技术资讯、海量的技术资料和视频！

我们是开放型
开源软件公司

关注“SUSE 订阅”

长按二维码关注

点击“在看”，会及时收到心仪内容哦

Read more

喜欢此内容的人还喜欢

高端存储过时了 ？

云头条