Managing CPS Disks

Adding a New Disk

This section describes the procedures needed to add a new disk to a VM.

Prerequisites

- All the VMs were created using the deployment process.
- This procedure assumes the datastore that will be used to have the virtual disk has sufficient space to add the virtual disk.
- This procedure assumes the datastore has been mounted to the VMware ESX server, regardless of the backend NAS device (SAN or iSCSI, etc).

ESX Server Configuration

Step 1 Login to the ESX server shell, and make sure the datastore has enough space:

```
vmkfstools -c 4g /vmfs/volumes/datastore_name/VMNAME/xxxx.vmdk -d thin
```

Step 2 Execute `vim-cmd vmsvc/getallvms` to get the vmid of the VM where the disk needs to be added.

```
Vmid Name                      File                                Guest OS    Version Annotation
173  vminstaller-AIO [datastore5] vminstaller-AIO/vminstaller-AIO.vmx centos64Guest vmx-08
```

Step 3 Assign the disk to the VM.

The *xxxx* is the disk name, and 0 and 1 indicate the SCSI device number.

In this example, this is the second disk:

```
vim-cmd vmsvc/device.diskaddexisting vmid /vmfs/volumes/path to xxxx.vmdk 0 1
```

Target VM Configuration

Step 1 Log in as root user on your Linux virtual machine.

Step 2 Open a terminal session.

Step 3 Execute the `df` command to examine the current disks that are mounted and accessible.

Step 4 Create an ext4 file system on the new disk:

```
mkfs -t ext4 /dev/sdb
```

Note `b` in `/dev/sdb` is the second SCSI disk. It warns that you are performing this operation on an entire device, not a partition. That is correct, since you created a single virtual disk of the intended size. This is assuming you have specified the correct device. Make sure you have selected the right device; there is no undo.

Step 5 Execute the following command to verify the existence of the disk you created:

```
# fdisk -l
```

Step 6 Execute the following command to create a mount point for the new disk:

```
# mkdir /<NewDirectoryName>
```

Step 7 Execute the following command to display the current `/etc/fstab`:

```
# cat /etc/fstab
```

Step 8 Execute the following command to add the disk to `/etc/fstab` so that it is available across reboots:

```
/dev/sdb /<NewDirectoryName> ext4 defaults 1 3
```

Step 9 Reboot the VM.

```
shutdown -r now
```

Step 10 Execute the `df` command to check the file system is mounted and the new directory is available.

Update the collectd process to use the new file system to store KPIs

After the disk is added successfully, **collectd** can use the new disk to store the KPIs.

Step 1 SSH into pcrfclient01/pcrfclient02.

Step 2 Execute the following command to open the logback.xml file for editing:

```
vi /etc/collectd.d/logback.xml
```

Step 3 Update the file element <*file*> with the new directory that was added in the `/etc/fstab`.

Step 4 Execute the following command to restart `collectd`:

```
monit restart collectd
```

Note The content of logback.xml will be overwritten to the default path after a new upgrade. Make sure to update it after an upgrade.

Mounting the Replication Set from Disk to tmpfs After Deployment

You can mount all of the members of the Replication set to tmpfs, or you can mount specific members to tmpfs. These scenarios are described in the following sections.

Scenario 1 – Mounting All Members of the Replication Set to tmpsf

Step 1 Modify `mongoConfig.cfg` file using the vi editor on cluster manager. Change the DBPATH directory for the SPR Replication set that needs to be put on tmpfs.

Note Make sure you change the path to `/var/data/sessions.1`, which is the tmpfs filesystem. Also, make sure to run diagnostics.sh before and after the activity.

The following example shows the contents of `mongoConfig.cfg` file before modification:

```
[SPR-SET1]
SETNAME=set06
OPLOG_SIZE=5120
ARBITER1=pcrfclient01a:27720
ARBITER_DATA_PATH=/var/data/sessions.6
MEMBER1=sessionmgr04a:27720
MEMBER2=sessionmgr03a:27720
MEMBER3=sessionmgr04b:27720
MEMBER4=sessionmgr03b:27720
DATA_PATH=/var/data/sessions.4
[SPR-SET1-END]
```

The following example shows the contents of `mongoConfig.cfg` file after modification:

```
[SPR-SET1]
SETNAME=set06
OPLOG_SIZE=5120
ARBITER1=pcrfclient01a:27720
ARBITER_DATA_PATH=/var/data/sessions.6
MEMBER1=sessionmgr04a:27720
MEMBER2=sessionmgr03a:27720
MEMBER3=sessionmgr04b:27720
MEMBER4=sessionmgr03b:27720
DATA_PATH=/var/data/sessions.1/set06
[SPR-SET1-END]
```

Step 2 Run `build_set` to generate new MongoDB startup scripts. It generates new mongod startup scripts for all the SPR Replication sets:

```
build_set.sh --spr --create-scripts
```

In this example, we are generating new MongoDB startup scripts for the SPR database. Use balance/session depending on your activity.

Step 3 In you need to generate new MongoDB scripts for specific setname, run the following command:

```
build_set.sh --spr --create-scripts --setname set06
```

Step 4 Verify that the new mongo script is generated. SSH to one of the session manager servers and run the following command. The DBPATH should match what you modified in Step 1, on page 3. For example:

```
grep /var/data sessionmgr-27720
```

You should see the following output:

```
DBPATH=/var/data/sessions.1/set06
```

Step 5 Copy the `mongoConfig.cfg` file to all nodes using the following command:

```
copytoall /etc/broadhop/mongoConfig.cfg /etc/broadhop/mongoConfig.cfg
```

Step 6 Run `build_etc.sh` to update puppet files, which retains the updated `mongoConfig.cfg` file after reboot.

Step 7 Stop and start the mongo databases one by one.

Step 8 Run `diagnostics.sh`.

Step 9 If this is an Active/Active GEOHA setup, scp the `mongoConfig.cfg` file to Site-B cluster manager, and do the following:

a) Copy the `mongoConfig.cfg` file from Cluster Manager to all nodes using the following command:

```
copytoall /etc/broadhop/mongoConfig.cfg /etc/broadhop/mongoConfig.cfg
```

b) Run `build_etc.sh` to update puppet files, which retains the updated `mongoConfig.cfg` file after reboot.

Scenario 2 – Mounting Specific Members of the Replication Set to tmpfs

Step 1 Ssh to the respective session manager.

Step 2 Edit the mongoDB startup file using the vi editor. In this example we are modifying the SPR member.

```
[root@sessionmgr01 init.d]# vi /etc/init.d/sessionmgr-27720
```

Step 3 Change the DBPATH directory from `DBPATH=/var/data/sessions.4` to `DBPATH=/var/data/sessions.1/set06`.

Step 4 Save and exit the file (using !wq).

Step 5 Enter the following commands to stop and start the SPR DB member:

```
/usr/bin/systemctl stop sessionmgr-27720
/usr/bin/systemctl start sessionmgr-27720
```

Step 6 Wait for the recovery to finish.

Manage Disks to Accommodate Increased Subscriber Load

If you need to prepare CPS for an increased number of subscribers (> 10 million), you can clone and repartition the sessionmgr disks as per your requirement.

Clone Sessionmgr01 VM

Downtime: No downtime

Before you begin

- Before disk repartition, clone sessionmgr01. This step is optional but to reduce the risk of losing the data during disk repartitioning, the customer can take the backup of sessionmgr01 VM. If the customer does not have enough space to take the backup this step can be ignored.
- Blade with enough space to hold cloned image of sessionmgr01.

Step 1 Login to vSphere Client on sessionmgr01 blade with administrator credentials.

Step 2 Right-click sessionmgr01 and select **Clone** > Choose appropriate inventory in which blade resides > Choose the blade with enough space to hold sessionmgr01 image > **Next** > **Next** > **Finish**.

Step 3 Cloning starts. Wait for it to finish the process.

Disk Repartitioning of Sessionmgr01 VM

Downtime: During this procedure Sessionmgr01 is shut down 2 times. Estimate approximately 30 minutes of downtime for sessionmgr01.

CPS continues to operate using the other sessionmgr02 while sessionmgr01 is stopped as part of procedure.

Before you begin

None

Step 1 Login to sessionmgr01 as a root user.

Step 2 The following commands may be executed to help identify which partition requires additional space.

```
 synph# df -h/synph
synphFilesystem                       Size  Used Avail Use% Mounted on/synph
synph/dev/mapper/vg_shiprock-lv_root  7.9G  1.5G  6.0G  20% //synph
synphtmpfs                            1.9G     0  1.9G   0% /dev/shm/synph
synph/dev/sda1                        485M   32M  428M   7% /boot/synph
synph/dev/mapper/vg_shiprock-lv_home  2.0G   68M  1.9G   4% /home/synph
synph/dev/mapper/vg_shiprock-lv_var    85G   16G   65G  20% /var/synph
synphtmpfs                            2.3G  2.1G  172M  93% /var/data/sessions.1/synph
synph/synph
synph# pvdisplay/synph
synph  --- Physical volume ---/synph
synph  PV Name               /dev/sda2/synph
synph  VG Name               vg_shiprock/synph
synph  PV Size               99.51 GiB / not usable 3.00 MiB/synph
synph  Allocatable           yes (but full)/synph
synph  PE Size               4.00 MiB/synph
synph  Total PE              25474/synph
synph  Free PE               0/synph
synph  Allocated PE          25474/synph
synph  PV UUID               l3Mjox-tLfK-jj4X-98dJ-K3c1-EOel-SlOBq1/synph
```

```
synph/synph
synph# vgdisplay/synph
synph--- Volume group ---/synph
synph  VG Name               vg_shiprock/synph
synph  System ID             /synph
synph  Format                lvm2/synph
synph  Metadata Areas        1/synph
synph  Metadata Sequence No  5/synph
synph  VG Access             read/write/synph
synph  VG Status             resizable/synph
synph  MAX LV                0/synph
synph  Cur LV                4/synph
synph  Open LV               4/synph
synph  Max PV                0/synph
synph  Cur PV                1/synph
synph  Act PV                1/synph
synph  VG Size               99.51 GiB/synph
synph  PE Size               4.00 MiB/synph
synph  Total PE              25474/synph
synph  Alloc PE / Size       25474 / 99.51 GiB/synph
synph  Free  PE / Size       0 / 0   /synph
synph  VG UUID               P1ET44-jiEI-DIbd-baYt-fVom-bhUn-zgs5Fz/synph
```

- (df -h): /var is /dev/mapper/vg_shiprock-lv_var. This is equivalent to device /dev/vg_shiprock/lv_var.
- (pvdisplay): vg_shiprock (used by lv_var which is /var) is on /dev/sda2.

Step 3 Execute the fdisk command to check the disk size.

```
# fdisk -l /dev/sda

Disk /dev/sda: 107.4 GB, 107374182400 bytes
255 heads, 63 sectors/track, 13054 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x0008dcae

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *           1          64      512000   83  Linux
Partition 1 does not end on cylinder boundary.
/dev/sda2              64       13055   104344576   8e  Linux LVM
```

Step 4 Power down the Virtual Machine.

```
# shutdown -h now
```

Note If cloning is not possible because of space limitation on Blade, backup of sessionmgr01 VM can be taken by saving OVF of sessionmgr01 VM to local storage like Laptop, Desktop. (Both cloning and OVF backup are optional steps, but either one of them is highly recommended.)

Step 5 Log in using the VMware vSphere Client as an administrator (e.g. root) to the ESXi host which has your Linux Virtual Machine on it.

Step 6 Right-click on the Virtual Machine and select Edit Settings > Click Hard Disk 1 > Increase the Provisioned Size of the Hard Disk.

Step 7 Power ON the Virtual Machine.

Step 8 Login (ssh) to the Virtual Machine as root user.

Step 9 Confirm that disk space has been added to the /dev/sda partition.

```
# fdisk -l /dev/sda

Disk /dev/sda: 70.5 GB, 79529246720 bytes
255 heads, 63 sectors/track, 9668 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
```

Step 10 Execute the following commands (Bold Characters indicates actual inputs from user (all of them are in lower case).

```
# fdisk /dev/sda
The number of cylinders for this disk is set to 7832.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
1) software that runs at boot time (e.g., old versions of LILO)
2) booting and partitioning software from other OSs
   (e.g., DOS FDISK, OS/2 FDISK)
Command (m for help): p
Disk /dev/sda: 64.4 GB, 64424509440 bytes
255 heads, 63 sectors/track, 7832 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *           1          13      104391   83  Linux
/dev/sda2              14        7179    57560895   8e  Linux LVM
Command (m for help): d
Partition number (1-4): 2
Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 2
First cylinder (14-7832, default 14):  [press enter]
Using default value 14
Last cylinder +sizeM/+sizeK (14-7832,default 7832): [press enter]
Using default value 7832
Command (m for help): t
Partition number (1-4): 2
Hex code (type L to list codes): 8e
Changed system type of partition 2 to 8e (Linux LVM)
Command (m for help): w
The partition table has been altered!
Calling ioctl() to re-read partition table.
WARNING: Re-reading the partition table failed with error 16: Device or resource busy.
The kernel still uses the old table.
The new table will be used at the next reboot.
Syncing disks.
```

Step 11 Reboot the sessionmgr01 VM by executing the following command:

```
# reboot
```

This ensures that the new setting match up with the kernel.

Step 12 After reboot, execute following command:

```
# pvresize /dev/sda2
Physical volume "/dev/sda2" changed
1 physical volume(s) resized / 0 physical volume(s) not resized
```

Step 13 Confirm that the additional free space is added in sessionmgr VM.

```
# vgdisplay
  --- Volume group ---
  VG Name               vg_shiprock
  System ID
  Format                lvm2
```

```
  Metadata Areas        1
  Metadata Sequence No  5
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                4
  Open LV               4
  Max PV                0
  Cur PV                1
  Act PV                1
VG Size             129.51 GiB
PE Size              4.00 MiB
Total PE             32974
Alloc PE / Size     25474 / 99.51 GiB
Free PE / Size     7500 / 30.00 GB
VG UUID              pPSNBU-FRWO-z3aC-iAxS-ewaw-jOFT-dTcBKd
```

Step 14 Verify that the /var partition is mounted on /dev/mapper/vg_shiprock-lv_var.

```
#df -h
Filesystem           Size Used Avail Use% Mounted on
/dev/mapper/vg_shiprock-lv_root
      18G 2.5G    15G 15% /
/dev/mapper/vg_shiprock-lv_home
      5.7G 140M 5.3G   3% /home
/dev/mapper/vg_shiprock-lv_var
      85G   16G   65G   20% /var
/dev/sda1              99M   40M   55M  43% /boot
tmpfs                  16G     0   16G   0% /dev/shm
tmpfs                 8.0G  1.1G  7.0G  14% /data/sessions.1
```

Step 15 Extend /var partition to take up additional free space.

```
#lvextend -l +100%FREE /dev/mapper/vg_shiprock-lv_var
 Extending logical volume lv_var to 120.00 GB
 Logical volume lv_var successfully resized
```

Step 16 Check the newly added space in /dev/mapper/vg_shiprock-lv_var.

```
# lvdisplay
```

Step 17 Add space to VM file system.

```
# resize2fs /dev/mapper/vg_shiprock-lv_var
resize2fs 1.39 (29-May-2006)
Filesystem at /dev/mapper/vg_shiprock-lv_var is mounted on /var; on-line resizing required
Performing an on-line resize of /dev/mapper/vg_shiprock-lv_var to 6553600 (4k) blocks.
The filesystem on /dev/mapper/vg_shiprock-lv_var is now 6553600 blocks long.
```

Step 18 Check the increased size of /var partition.

```
# df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/mapper/vg_shiprock-lv_root
                       23G  2.1G   20G  10% /
/dev/mapper/vg_shiprock-lv_home
                      5.7G  140M  5.3G   3% /home
/dev/mapper/vg_shiprock-lv_var
                      130G   16G   95G  12% /var
/dev/sda1              99M   40M   55M  43% /boot
tmpfs                 2.0G     0  2.0G   0% /dev/shm
```

Cloning and Disk Repartitioning of Sessionmgr02 VM

Repeat Clone Sessionmgr01 VM, on page 5 and Disk Repartitioning of Sessionmgr01 VM, on page 5 on sessionmgr02 for cloning and disk repartitioning of sessionmgr02 VM.