Restoring a large enterprise MDM environment in VMware from MDS backup

Objective:
Replicate a customer's large enterprise MDM environment in VMware without wasting disk space for snapshots and backup partitions.

Expectations:
You should already be familiar with creating a virtual machine in VMware with enough memory and available disk space to replicate your customer's MDM environment.

Limitations:
When you create a virtual machine to replicate a large MDM environment the Gaia installer will reserve a large amount of space for the snapshot and backups partition which is not necessary when replicating a customer environment.

Solution:
This document will show you how to create a virtual machine with a small initial virtual disk and later add a second larger virtual disk that will be used for replicating the customer MDM environment. This greatly reduces the wasted space of the large backup and snapshot partition.

Before you start:
What can you do to reduce the size before replicating the MDM environment?
- sk92664: Domain Management Server database is filled with large sized directories in $FWDIR/conf/db_snapshot/ directory structure
- Delete old database revisions. If possible have the customer perform this clean up before they run their MDS backup that is provided for replication.

Building the system:
Create a new virtual machine with a single disk of 48GB of space and run through the Gaia installation process by mounting the appropriate ISO for your customer MDM environment. I have used R80.20 for creating this documentation. Once the install is complete go through the first time wizard and be sure to select Multi-Domain Management and follow the steps to set up an MDS environment.

NOTE: Make sure that you run 'mdsconfig' and disable the 'auto start' option.

Check the existing disk partitions after the installation process completes

```
# cat /proc/partitions
major minor  #blocks  name

    3     0   33554432 hda
    3     1     305203 hda1
    3     2    3140707 hda2
    3     3   30105810 hda3
  253     0   10485760 dm-0
  253     1    6291456 dm-1

# fdisk -l

Disk /dev/hda: 34.3 GB, 34359738368 bytes
255 heads, 63 sectors/track, 4177 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/hda1   *           1          38      305203+  83  Linux
/dev/hda2              39         429     3140707+  82  Linux swap / Solaris
/dev/hda3             430        4177    30105810   8e  Linux LVM

# df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/mapper/vg_splat-lv_current
                      9.7G  4.4G  4.9G  48% /
/dev/hda1             289M   17M  258M   7% /boot
tmpfs                 980M     0  980M   0% /dev/shm
/dev/mapper/vg_splat-lv_log
                      5.9G  230M  5.3G   5% /var/log
```

<u>Now shut down the virtual machine and add a second larger virtual disk to the virtual machine.</u>

In my experience you want to create this partition to be at least 8-10 times the size of the MDS backup file provided by the customer. This means for a 20GB MDS backup file you will want between a 160-200GB large partition for the restore. You can experiment with the sizes if you have time to spare but large disks are cheap so I prefer to go larger and save time rebuilding if I size too small the first time.

Check the list of partitions to be sure the newly added virtual disk is recognized

```
# cat /proc/partitions
major minor  #blocks  name

    3     0   33554432 hda
    3     1     305203 hda1
    3     2    3140707 hda2
    3     3   30105810 hda3
    3    64   16777216 hdb <----------- This is the new disk I added.
  253     0   10485760 dm-0
  253     1    6291456 dm-1
```

```
# fdisk -l

Disk /dev/hda: 34.3 GB, 34359738368 bytes
255 heads, 63 sectors/track, 4177 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/hda1   *           1          38      305203+  83  Linux
/dev/hda2              39         429     3140707+  82  Linux swap / Solaris
/dev/hda3             430        4177    30105810   8e  Linux LVM

Disk /dev/hdb: 17.1 GB, 17179869184 bytes
15 heads, 63 sectors/track, 35507 cylinders
Units = cylinders of 945 * 512 = 483840 bytes

Disk /dev/hdb doesn't contain a valid partition table
```

Mount and partition the new virtual disk using the 'parted' utility

```
# parted
GNU Parted 1.8.1
Using /dev/hda
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) select /dev/hdb <-  The name of the device from /proc/partitions (or fdisk -l)
Using /dev/hdb
(parted) mklabel gpt
(parted) print

Model: VMware Virtual IDE Hard Drive (ide)
Disk /dev/hdb: 17.2GB
Sector size (logical/physical): 512B/512B
Partition Table: gpt

Number  Start  End  Size  File system  Name  Flags

(parted) mkpart primary ext3 0 16384 <-  Size in GB x 1024.
(parted) print

Model: VMware Virtual IDE Hard Drive (ide)
Disk /dev/hdb: 17.2GB
Sector size (logical/physical): 512B/512B
Partition Table: gpt

Number  Start    End      Size     File system  Name       Flags
 1      17.4kB   16.4GB   16.4GB                 primar/
```

Exit the 'parted' utility

```
(parted) quit
```

Create the new file system for the new large virtual disk

```
# fdisk -l

Disk /dev/hda: 34.3 GB, 34359738368 bytes
255 heads, 63 sectors/track, 4177 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/hda1   *           1          38      305203+  83  Linux
/dev/hda2              39         429     3140707+  82  Linux swap / Solaris
/dev/hda3             430        4177    30105810   8e  Linux LVM

WARNING: GPT (GUID Partition Table) detected on '/dev/hdb'! The util
fdisk doesn't support GPT. Use GNU Parted.


Disk /dev/hdb: 17.1 GB, 17179869184 bytes
255 heads, 63 sectors/track, 2088 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/hdb1              1        2089    16777215+  ee  EFI
                                                       GPT

# mkfs -t ext3 /dev/hdb1 <- Name obtained from the command output above.
mke2fs 1.39 (21-Jun-
2016) Filesystem
label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096
(log=2)
2003424 inodes, 3999995 blocks
199999 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=4097835008
123 block groups
32768 blocks per group, 32768 fragments per group
16288 inodes per group
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632,
        2654208

Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 27 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
```

Create a temporary mount point for the new partition

```
# mkdir /mnt/tmp
# mount -t ext3 /dev/hdb1 /mnt/tmp
```

Verify that the new partition is mounted

```
# mount
/dev/mapper/vg_splat-lv_current on / type ext3 (rw)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
/dev/hda1 on /boot type ext3 (rw)
tmpfs on /dev/shm type tmpfs (rw)
/dev/mapper/vg_splat-lv_log on /var/log type ext3 (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
/dev/hdb1 on /mnt/tmp type ext3 (rw) <- This is the new partition.
```

Copy contents of /var/opt to /mnt/tmp after stopping the MDS processes

```
# mdsstop
# cp -a /var/opt/* /mnt/tmp
# ls -al /mnt/tmp
total 92
drwxr-xr-x 20 admin root     4096 Dec 16 23:50 .
drwxr-xr-x  3 admin root     4096 Dec 16 23:46 ..
drwxrwx---  3 admin config   4096 Dec 16 22:36 CPCON66CMP-R80
drwxrwx---  3 admin config   4096 Dec 16 22:36 CPEdgecmp-R80
drwxrwx---  3 admin config   4096 Dec 16 22:36 CPNGXCMP-R80
drwxrwx---  3 admin config   4096 Dec 16 22:36 CPR71CMP-R80
drwxrwx---  3 admin config   4096 Dec 16 22:36 CPR7520CMP-R80
drwxrwx---  3 admin config   4096 Dec 16 22:36 CPR7540CMP-R80
drwxrwx---  3 admin config   4096 Dec 16 22:36 CPR7540VSCMP-R80
drwxrwx---  3 admin config   4096 Dec 16 22:36 CPR75CMP-R80
drwxrwx---  3 admin config   4096 Dec 16 22:36 CPR76CMP-R80
drwxrwx---  3 admin config   4096 Dec 16 22:36 CPSG80CMP-R80
drwxrwx---  3 admin config   4096 Dec 16 22:36 CPSG80R75CMP-R80
drwxrwx---  3 admin config   4096 Dec 16 22:36 CPV40Cmp-R80
drwxrwx---  9 admin config   4096 Dec 16 22:37 CPmds-R80
drwxrwx---  5 admin root     4096 Dec 16 17:25 CPshrd-R80
drwxrwx--x  3 admin root     4096 Dec 16 17:26 CPsuite-R80
drwxrwx---  3 admin root     4096 Dec 16 22:40 fw.boot
drwx------  2 admin root    16384 Dec 16 23:45 lost+found
drwxrwx---  3 admin root     4096 Dec 16 17:27 uf
```

Delete the no longer needed contents of /var/opt

```
# rm -fR /var/opt/*
```

Unmount the new large partition

```
# umount /mnt/tmp
```

Verify that the new partition is no longer mounted

```
# mount
/dev/mapper/vg_splat-lv_current on / type ext3 (rw)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
/dev/hda1 on /boot type ext3 (rw)
tmpfs on /dev/shm type tmpfs (rw)
/dev/mapper/vg_splat-lv_log on /var/log type ext3 (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
```

Create a new permanent mount point for the new large partition

```
# more /etc/fstab <- Unmodified /etc/fstab should look like.
/dev/mapper/vg_splat-lv_current /                    ext3    defaults
1 1
LABEL=/boot             /boot                 ext3    defaults        1 2
devpts                  /dev/pts              devpts  gid=5,mode=620  0 0
tmpfs                   /dev/shm              tmpfs   defaults        0 0
proc                    /proc                 proc    defaults        0 0
sysfs                   /sys                  sysfs   defaults        0 0
/dev/mapper/vg_splat-lv_log /var/log              ext3    defaults        1 2
LABEL=SWAP-hda2         swap                  swap    defaults        0 0

# cp /etc/fstab /etc/fstab.original <- Backup and then edit the file.
[Expert@R8020ActiveMDM:0]# vi /etc/fstab
```

Add the following line:

```
/dev/hdb1               /var/opt              ext3    defaults        0 0
```

The device name will vary depending on your virtual machine environment. Adjust the naming accordingly however all the other lines will remain the same.

Once modified you should have the following output:

```
# more /etc/fstab
/dev/mapper/vg_splat-lv_current /                      ext3    defaults
1 1
LABEL=/boot             /boot                ext3    defaults      1 2
devpts                  /dev/pts             devpts  gid=5,mode=620  0 0
tmpfs                   /dev/shm             tmpfs   defaults      0 0
proc                    /proc                proc    defaults      0 0

sysfs                   /sys                 sysfs   defaults      0 0
/dev/mapper/vg_splat-lv_log /var/log              ext3    defaults        1 2
LABEL=SWAP-hda2         swap                 swap    defaults      0 0
/dev/hdb1               /var/opt             ext3    defaults      0 0
```

Reboot the virtual machine to add the new large virtual disk to the MDS and verify the new disk is listed

```
# df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/mapper/vg_splat-lv_current
                      9.7G  4.0G  5.3G  43% /
/dev/hda1             289M   24M  251M   9% /boot
tmpfs                 980M     0  980M   0% /dev/shm
/dev/mapper/vg_splat-lv_log
                      5.9G  229M  5.3G   5% /var/log
/dev/hdb1              16G  2.3G   12G  17% /var/opt <-The /var/opt now mounted.
```

You may now copy your customer MDS backup file to a temporary directory in /var/opt and restore the MDS environment to the new large partition.

Keep in mind if you are restoring an MDS backup from a customer that runs management on Check Point appliances you will need to go through the following procedure to rename the virtual machine network interface to Mgmt or the MDS replication process will fail.

In expert mode on the replicated MDS server perform the following:

cd /etc/udev/rules.d
cat 00-OS-XX.rules

You will see the following output:
ID=="0000:02:00.0", NAME="eth0"

vi 00-OS-XX.rules
Replace eth0 with Mgmt and use :wq! to save the changes

Reboot the virtual machine and all MDS processes should come up with interfaces bound to the Mgmt interface as they would on a physical management appliance.