


EXPAND THE ACTOR STORAGE

As you use Mandiant Security Validation (MSV), you may come to a point where the default storage space for the Actor is not enough and you need to expand it. The overall process is similar for all versions of the Actor but is impacted by the form factor and where the Actor is installed.

Default Actor requirements

The following requirements must be met for the Security Validation Director to function properly. These requirements are the same for virtual appliance and installable software Actors:

Specification	Form Factor	Actor Minimum Requirement
<code>/opt</code>	Installable software	a minimum of 20 GB

 Newly-released Actor images have `/opt` set at 20 GB of space. However, existing Actor images cannot be automatically updated. If you're having space issues, we encourage you to increase the size of the Actor's `/opt` partition from 10 to 20 GB.

Change the Actor's local storage

For most Actors, you can use the same procedure, whether the Actor was installed from an appliance or from the installable software.

The process involves two steps:

1. Create a new partition.
2. Increase logical volume.



- This process is completed from the command line using an account that has root access.
- This process is only possible if you are on a virtual infrastructure that allows for expansion of the disk and use a Logical Volume Manager (LVM) storage setup.

Create a new partition

1. Connect to the actor using SSH.
2. Promote yourself to root using the following command:

```
sudo su -
```

3. Check to see if the current disk was expanded or if a second hard disk was added:

```
fdisk -l
```



- If two disks are shown, a second disk was added. If only one disk appears, the current disk was expanded.
- Note the name given of the disk that was either added or expanded. Use that for subsequent steps.

4. Execute the following to create a new partition to add to the LVM:

```
fdisk /dev/DISK_NAME
```

Where *DISK_NAME* is the name of the disk that appeared after you ran `fdisk -l` in the preceding step.

5. Use the following options (in the given order) to configure the partition:

step	Key you press	Explanation
1.	n	New partition
2.	p	New primary
3.	3	Third primary partition
4.	Enter	Default value for first sector
5.	Enter	Default value for last sector
6.	t	Set type
7.	3	Select partition 3
8.	8e	Select partition type of Linux LVM
9.	w	Write the partition table

Example output:

```
Command (m for help): n
Partition type
  p   primary (2 primary, 0 extended, 2 free)
  e   extended (container for logical partitions)
Select (default p): p
Partition number (3,4, default 3): 3
First sector (737280000-946995199, default 737280000):
Last sector, +sectors or +size{K,M,G,T,P} (737280000-946995199, default 946995199):

Created a new partition 3 of type 'Linux' and of size 100 GiB.

Command (m for help): t
Partition number (1-3, default 3): 3
Hex code (type L to list all codes): 8e

Changed type of partition 'Linux' to 'Linux LVM'.

Command (m for help): w
The partition table has been altered.
Syncing disks.
```

6. Verify the partition was created:

```
fdisk -l
```

Example output:

```
[root@actor ~]# fdisk -l
Disk /dev/vda: 451.6 GiB, 484861542400 bytes, 946995200 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x46-5f922
```

Disk identifier: 0x00031023

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/vda1	*	2048	2099199	2097152	1G	83	Linux
/dev/vda2		2099200	737279999	735180800	350.6G	8e	Linux LVM
/dev/vda3		737280000	946995199	209715200	100G	8e	Linux LVM

Disk /dev/mapper/VolGroup0-root: 10 GiB, 10737418240 bytes, 20971520 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/mapper/VolGroup0-swap: 8 GiB, 8589934592 bytes, 16777216 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/mapper/VolGroup0-opt: 182.6 GiB, 196020797440 bytes, 382853120 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/mapper/VolGroup0-home: 5 GiB, 5368709120 bytes, 10485760 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/mapper/VolGroup0-vartmp: 1 GiB, 1073741824 bytes, 2097152 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/mapper/VolGroup0-var: 80 GiB, 85899345920 bytes, 167772160 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/mapper/VolGroup0-tmp: 50 GiB, 53687091200 bytes, 104857600 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/mapper/VolGroup0-audit: 10 GiB, 10737418240 bytes, 20971520 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/mapper/VolGroup0-varlog: 4 GiB, 4294967296 bytes, 8388608 sectors
Units: sectors of 1 * 512 = 512 bytes

```
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

7. Rescan the volume:

```
partprobe -s
```

Example output:

```
[root@actor ~]# partprobe -s
/dev/vda: msdos partitions 1 2 3
```

Increase the logical volume

1. Rescan the disks:

```
partprobe -s
```

2. Create the physical volume:

```
pvcreate /dev/vda3
```

Example output:

```
[root@actor ~]# pvcreate /dev/vda3
Physical volume "/dev/vda3" successfully created.
```

3. Get the name of the volume group you are expanding:

```
vgdisplay
```

When working with a virtual appliance Actor, this value is `VolGroup0` .

```
[root@actor ~]# vgdisplay
--- Volume group ---
VG Name          VolGroup0
System ID
Format           lvm2
Metadata Areas   1
Metadata Sequence No 10
VG Access        read/write
VG Status        resizable
MAX LV           0
Cur LV          9
Open LV          9
Max PV           0
Cur PV          1
Act PV           1
VG Size          < 57.59 GiB
PE Size          4.00 MiB
Total PE         14743
Alloc PE / Size  14743 / < 57.59 GiB
Free PE / Size   0 / 0
VG UUID          rOqPYv-jjDW-IKwI-OoEf-k3lg-VTGT-49cpxA
```

Save the name and exact spelling for the item listed in **VG Name**.

4. Extend the physical volume group:

```
vgextend VolGroup0 /dev/vda3
```

Example output:

```
[root@actor ~]# vgextend VolGroup0 /dev/vda3
Volume group "VolGroup0" successfully extended
```

5. Scan the new disk:

```
pvscan
```


Example output:

```
PV /dev/vda2  VG VolGroup0   lvm2 [< 350.56 GiB / 0   free]
PV /dev/vda3  VG VolGroup0   lvm2 [< 100.00 GiB / < 100.00 GiB free]
Total: 2 [450.55 GiB] / in use: 2 [450.55 GiB] / in no VG: 0 [0   ]
```

6. Get the path of the logical volume:

```
lvdisplay
```

For a virtual appliance Actor, this value is `/dev/VolGroup0/root` .

 If expanding `/var` or `/opt` depending on needs, replace references to `root` with `var` or `opt` in the remaining commands, such as `/dev/VolGroup0/var` .

Example output:

```
[root@actor ~]# lvdisplay

.....SEVERAL LOGICAL VOLUMES THAT CAN BE IGNORED.....

--- Logical volume ---
LV Path          /dev/VolGroup0/root
LV Name          root
VG Name          VolGroup0
LV UUID          8mEqh1-clJl-URC7-ZyU8-ZQhz-358N-xtuNoz
LV Write Access  read/write
LV Creation host, time localhost.localdomain, 2024-05-15 14:18:10 +0000
LV Status        available
# open           1
LV Size          10.00 GiB
Current LE       2560
Segments         1
Allocation       inherit
Read ahead sectors auto
- currently set to 256
Block device     253:0
```

7. Extend the logical volume:

```
lvextend /dev/VolGroup0/root /dev/vda3
```

Example output:

```
[root@actor ~]# lvextend /dev/VolGroup0/root /dev/vda3
Size of logical volume VolGroup0/root changed from 10.00 GiB (2560 extents) to < 110.00 GiB (28159 extents)
.
Logical volume VolGroup0/root successfully resized.
```

8. Expand the disk size:

```
xfs_growfs /dev/VolGroup0/root
```

Example output:

```
[root@actor ~]# xfs_growfs /dev/VolGroup0/root
meta-data=/dev/mapper/VolGroup0-root isize=512  agcount=4, agsize=655360 blks
          =             sectsz=512   attr=2, projid32bit=1
          =             crc=1      finobt=1, sparse=1, rmapbt=0
          =             reflink=1   bigtime=0 inobtcount=0
data      =             bsize=4096  blocks=2621440, imaxpct=25
          =             sunit=0    swidth=0 blks
naming    =version 2          bsize=4096  ascii-ci=0, ftype=1
log       =internal log      bsize=4096  blocks=2560, version=2
          =             sectsz=512  sunit=0 blks, lazy-count=1
realtime  =none              extsz=4096  blocks=0, rtextents=0
data blocks changed from 2621440 to 28834816
```

9. Verify the disk is expanded:

```
df -h
```

Before expansion:

```
[root@actor nodeone]# df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        3.8G   0 3.8G   0% /dev
tmpfs           3.8G 296K 3.8G   1% /dev/shm
tmpfs           3.8G 8.9M 3.8G   1% /run
tmpfs           3.8G   0 3.8G   0% /sys/fs/cgroup
/dev/mapper/VolGroup0-root 10G 5.5G 4.6G 55% /
/dev/sda1       1014M 277M 738M 28% /boot
/dev/mapper/VolGroup0-var 80G 1.6G 79G  2% /var
/dev/mapper/VolGroup0-home 5.0G 2.7G 2.4G 53% /home
/dev/mapper/VolGroup0-vartmp 1014M 40M 975M 4% /var/tmp
/dev/mapper/VolGroup0-opt 183G 17G 166G 10% /opt
/dev/mapper/VolGroup0-varlog 4.0G 227M 3.8G 6% /var/log
/dev/mapper/VolGroup0-audit 10G 109M 9.9G 2% /var/log/audit
/dev/mapper/VolGroup0-tmp 50G 390M 50G 1% /tmp
shm             63M 64K 63M  1% /usr/lib/msi/containers/overlay-containers/75c47f103d920230a218ff
b327cfe5d339322bff5ecee5f4f824419e983faebf/userdata/shm
overlay        10G 5.5G 4.6G 55% /usr/lib/msi/containers/overlay/5bf832397b4d01ed643d5615c487b0
0ec85d6f7605b52fdf95a171b71f048625/merged
tmpfs          769M   0 769M  0% /run/user/1000
```

After expansion:

```
[root@actor nodeone]# df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        3.8G  0  3.8G  0% /dev
tmpfs           3.8G  296K  3.8G  1% /dev/shm
tmpfs           3.8G  8.9M  3.8G  1% /run
tmpfs           3.8G  0  3.8G  0% /sys/fs/cgroup
/dev/mapper/VolGroup0-root  10G  5.5G  4.6G  55% /
/dev/sda1       1014M  277M  738M  28% /boot
/dev/mapper/VolGroup0-var  80G  1.6G  79G  2% /var
/dev/mapper/VolGroup0-home  5.0G  2.7G  2.4G  53% /home
/dev/mapper/VolGroup0-vartmp 1014M  40M  975M  4% /var/tmp
/dev/mapper/VolGroup0-opt  331G  18G  314G  6% /opt
/dev/mapper/VolGroup0-varlog 4.0G  227M  3.8G  6% /var/log
/dev/mapper/VolGroup0-audit 10G  109M  9.9G  2% /var/log/audit
/dev/mapper/VolGroup0-tmp  50G  390M  50G  1% /tmp
shm             63M  64K  63M  1% /usr/lib/msi/containers/overlay-containers/75c47f103d920230a218ffb327cfe5d339322bff5ecee5f4f824419e983faebf/userdata/shm
overlay        10G  5.5G  4.6G  55% /usr/lib/msi/containers/overlay/5bf832397b4d01ed643d5615c487b00ec85d6f7605b52fdf95a171b71f048625/merged
tmpfs          769M  0  769M  0% /run/user/1000
```