

Resize Disk Image

There are several disk images which can be easily extended. But pay attention to the overall consumed space of your sparse disk images.

Private disk image

1048576 MB is the maximum size which can be assigned to a private storage through qubes-manager.

To grow the private disk image of a AppVM beyond this limit [qubes-grow-private](#) can be used:

```
qvm-grow-private <vm-name> <size>
```

Note: Size is the target size (i.e. 4096MB or 16GB, ...), not the size to add to the existing disk.

Shrinking private disk image (Linux VM)

This operation is dangerous and this is why it isn't available in standard Qubes tools. If you have enough disk space, it is safer to create new VM with smaller disk and move the data.

The basic idea is to:

1. Shrink filesystem on the private disk image.
2. Then shrink the image.

Ext4 does not support online shrinking, so can't be done as convenient as image grown. Note that we don't want to touch the VM filesystem directly in dom0 for security reasons. First you need to start VM without `/rw` mounted. One of the possibility is to interrupt its normal startup by adding `rd.break` kernel option:

```
qvm-prefs -s <vm-name> kernelopts rd.break  
qvm-start --no-guid <vm-name>
```

And wait for qrexec connect timeout (or simply press Ctrl-C). Then you can connect to VM console and shrink the filesystem:

```
sudo xl console <vm-name>  
# you should get dracut emergency shell here  
mount --bind /dev /sysroot/dev  
chroot /sysroot  
mount /proc  
e2fsck -f /dev/xvdb  
resize2fs /dev/xvdb <new-desired-size>  
umount /proc
```

```
exit
umount /sysroot/dev
poweroff
```

Now you can resize the image:

```
truncate -s <new-desired-size> /var/lib/qubes/appvms/<vm-name>/private.img
```

It is critical to use the same (or bigger for some safety margin) size in truncate call compared to resize2fs call. Otherwise you will loose your data! Then reset kernel options back to default:

```
qvm-prefs -s <vm-name> kernelopts default
```

Done.

Template disk image

If you want install a lot of software in your TemplateVM, you may need to increase the amount of disk space your TemplateVM can use.

1. Make sure that all the VMs based on this template are shut off (including netvms etc).
2. Sanity check: verify that none of loop device are pointing at this template root.img: `sudo losetup -a`
3. Resize root.img file using `truncate -s <desired size>` (the root.img path can be obtained from qvm-prefs).
4. If any netvm/proxyvm used by this template is based on it, set template netvm to none.
5. Start the template.
6. Execute `sudo resize2fs /dev/mapper/dmroot` in the template.
7. Verify available space in the template using `df -h`
8. Shutdown the template.
9. Restore original netvm setting (if changed), check firewall settings (setting netvm to none causes firewall reset to “block all”)

HVM disk image

In this example we will grow the disk image of an HVM to 30GB.

First, stop/shutdown the HVM.

Then, from a Dom0 terminal (in KDE: System Tools -> Terminal Emulator) do the following:

```
cd /var/lib/qubes/appvms/<yourHVM>/
ls -lh root.img  (<--verify current size of disk image)
truncate -s 30GB root.img
ls -lh root.img  (<--verify new size of disk image)
```

The partition table and file-system must be adjusted after this change. Use tools appropriate to the OS in your HVM. Brief instructions for Windows 7, FreeBSD, and Linux are provided below.

Windows 7

1. Click Start
2. type “diskmgmt.msc” - this takes you to Disk Management
3. Right-click on your existing volume, select “Extend Volume...”
4. Click through the wizard.

No reboot required.

FreeBSD

```
gpart recover ada0
sysctl kern.geom.debugflags=0x10
gpart resize -i index ada0
zpool online -e poolname ada0
```

Linux

You will see that there is unallocated free space at the end of your primary disk. You can use standard linux tools like fdisk and mkfs to make this space available.