Resizing KVM qcow2 images

While expanding qcow2 images is easy, finding an easy way to shrink them was not.

Recently, my Matrix VM ran out of disk space, while my miscellaneous hosting VM had way too much space.

Since resizing images is often useful, I decided to make a fairly simple tutorial on how to expand and shrink the images.

Expanding images

This is fairly straightforward. Due to the design of qcow2 images, you don't even need to have the disk space available right away.

1. Shut down the virtual machine
2. Resize the image with

```
qemu-img resize image.qcow2 +SIZE
```

where SIZE is the size (e.g. 10G for 10 gibibytes).

3. Boot into an external live OS and resize the partition. The easiest way to do this is to use a GParted live image and virt-manager to connect to the VM.

Shrinking images

This is a bit more difficult. You will also need to have disk space to fit both images at the same time. For example, you want to resize a 100gb allocation to 50gb, your current image shows 60gb used from outside (the qcow2 image) and has 30gb actually used inside. For the change, you'll need to have 50gb space for the resized image and 60gb for the existing image, so 110gb total.

1. Resize the partition (see step #3 of expanding images)

2. If you managed to resize the partition from within the virtual machine (and thus didn't shut it down already for resizing), shut it down now.

2. KVM/QEMU images are stored in /var/lib/libvirt/images by default. It should be root-only, so sudo su is acceptable in this case.

Create the new smaller image:

```
qemu-img create -f qcow2 -o preallocation=metadata newimage.qcow2 NEW_SIZE
```

where NEW_SIZE is the size (50G for the example at the start).

4. Resize the image by copying the old image into the new one.

```
virt-resize oldimage.qcow2 newimage.qcow2
```

If the image created in the previous step is larger than the combined partitions on the old image, **virt-resize** will inform you of a surplus and create a new partition. You can still terminate the process without data loss and go back to step #3 to create a smaller image.

If the image is smaller than the partitions, **virt-resize** will fail and inform you how much space needs to be added. In this case, you must create a larger image in step #3.

5. Start your VM. There may be some disk errors related to the stored block lengths. fsck should be able to automatically fix them.

If **virt-resize** created an extra partition, you can now use a partition editor to delete it and add the space to another partition.

6. Once you have verified that the VM is working as expected, you can safely remove the old image.