Lab 12.2: Cloning a VM Image
IN720 Virtualisation

October 26, 2018

Introduction

In the last lab we create a guest domain on our Xen host. Recall that to do this we needed three things:

1. A disk, or something like one (we set up a logical volume for this);

2. An operating system kernel (we used one from the Ubuntu installer);

3. A configuration file (we wrote this).

With these resources we created a guest domain. Once it was running we installed Ubuntu onto our logical volume. After that, we did something rather tricky. We rewrote the configuration file to no longer user the kernel from the Ubuntu installer. Instead we set it to use the `pygrub` boot loader to use the kernel we installed on the logical volume. So, we actually created and ran *two different* guest domains.

As fun as it was, this process won't scale. We can't manually install an OS every time we want to create a new guest. We've already done it once; there must be a way to reuse that result. And of course there is. To run another guest domain, we need the three resources listed above. There are several ways we could do this. The other virtualisation tools we used this semester each had some clear concept of an *image*, a reusable resource we could use to create virtual instances. The bad news is that Xen doesn't have a canonical idea of an image. In fact, it has several ways to provide an image, and they're note really Xen tools. They just rely on the basic capabilities of our Dom0 system. In this lab we will employ one such method, but keep in mind that it is just one way among several.

1 Finalise our base image

We are going to use the volume from the previous lab as our base image, so begin by updating it one last time.

1. Use `xl list` to be sure your guest is running; if not, use `xl create /etc/xen/lab12cfg` to start it.

2. Connect to the guest console with `xl console lab12_guest`.

3. On the guest, run `sudo apt-get update && sudo apt-get dist-upgrade`.

4. Disconnect from the guest console and use `xl shutdown lab12_guest` to stop it.

5. Check that the guest no longer appears in the output of `xl-list`.

Since we are going to use the volume from this guest to store a base image for future guests, we do not want to modify it further. In particular, do not start a new guest attached to this volume.

2 Create copy-on-write file systems

We could simply copy the data from our base volume onto other guest volumes bit-by-bit, but doing so would result in a.lot of duplication since the resulting guest systems would be very similar. Luckily we don't have to do that. Using LVM we can create *copy-on-write* (COW) file systems. Create some now with the following commands: (In the commands below we use `dom0` as the name of our Dom0 system. You should substitute the name of your own system in your commands.)

```
modprobe dm-snapshot
lvcreate -s -L 1G -n clone1 /dev/dom0-vg/vol_lab12_vm
lvcreate -s -L 1G -n clone2 /dev/dom0-vg/vol_lab12_vm
```

The creates two new logical volumes based on our original volume. Even though the original volume was 10GB in size, these new volumes only use 1GB This is because these volumes simply use the data from the base volume and only need to store the difference between themselves and the base volume. This is also why modifying the base volume would be a Bad Idea at this point. Note that the `modprobe` command loaded a needed kernel module in the Dom0 system.

3 Create two new guests

Now that we have two new disks (which each contain kernels), we just need two new configurations to launch the guests. Simply start by copying our last configuration, `/etc/xen/lab12.cfg` into two new files, `clone1.cfg` and `clone2.cfg`. In both of those, make the following changes:

1. Change the name, since each guest needs a unique name;

2. Change the starting memory to 512, just to same system resources;

3. Change the first item in the disk triple to the path of the clone volume, e.g. `/dev/dom0-vg/clone1`

Now we can simply start each guest with the commands

```
xl create /etc/xen/clone1
xl create /etc/xen/clone2
```

Check that our new guests are running with the command `xl list`. Log into one of the commands and use `df -h` to observe the size of the root file system on the guest.

4 Resize a guest volume

Since our guests' storage is based on LVM file systems, we can resize them easily. Shut down your `clone1` guest before executing the commands:

```
lvextend -L 2G /dev/dom0-vg/clone1
resize2fs /dev/dom0-vg/clone1
```

Then restart your guest with `xl create`. It may be possible to do the above without shutting down the guest, but the guest operating system may not pick up the change. Feel free to experiment.