LS Lab Assignment: Virtualization[*]

Paul Danielse   Maurice Mouw   Shashikant Ilager

Feedback deadline:
November 1, 2024 23:59 CET

Abstract

Virtual machines (VMs) and virtual networks are a great tool for conducting network experiments. In this lab session you will get familiar with KVM and Libvirt. KVM is a hypervisor which supports both para-virtualization and hardware assisted virtualization. To make installation and configuration easy, you will use precompiled packages provided by the Linux distribution. Before starting fully read this document to get an overview of what is to come. Explain and describe every step you took to solve each task.

Preparing the environment

During this lab assignment, you might have to reinstall your server. Backup all data you want to preserve before continuing this lab assignment! If you want to deviate in any way from the suggested setup, you are allowed to do so as long as you first consult with the lab teacher and explain your reason and describe the changes on your wiki. Note that if you are using a non-standard setup, you may not receive help in debugging any problems that might occur!

Task 1. Make sure that Ubuntu 22.04 64-bit is installed on your server. Use the PXE service to start the installation process if needed. Partition your disks such that you have at least 200 GB of unpartitioned space.

Logical Volume Management

LVM (*Logical Volume Managment*) Hasenstein [2001] allows the grouping of storage resources, called *physical volumes*, into larger pools called *volume groups*. Each volume group can be then partitioned just as if it would be a single large drive. These partitions are called *logical volumes*. There are a lot of advantages of using LVM over the basic storage approach. LVM can be also combined with RAID solutions to provide resilience or to increase performance.

Task 2. Install the **lvm2** package and create a physical volume using 100 GB of the 200 GB free space reserved before. On top of the physical volume create a volume group called `VolumeGroupKVM`. Here you will store the virtual machine images. We will create the logical volumes later.
Hints: `pvcreate, vgcreate, pvdisplay`.

[*]Based on an earlier document by Cosmin Dumitru, Naod Jebessa, Jeroen van der Ham, Péter Prjevara, Arno Bakker, Vincent Breider. Version October 28, 2024.

Installing Libvirt

Before starting with KVM, make sure that you have some idea of how this works. Please read through Mastering KVM Virtualization chapter 1 paragraph 'Introducing KVM' on page 17 and familiarize yourself with the terminology and the general concepts.

KVM is by default part of the Linux Kernel and this does not require any additional installation. To use KVM however we do need to require some additional packages and make sure virtualization is enabled on our system. We are going to install the Libvirt and Qemu packages.

Task 3. Verify if your server has virtualization and KVM support enabled on your system. Install the **cpu-checker qemu-kvm**, **libvirt-daemon-system** and **libvirt-clients** packages. Make sure the libvirt daemon is started on boot, start the daemon and ensure your current user is allowed to manage kvm and libvirt (add your user to the required groups).

Explain the difference between KVM, libvirt en Qemu.

Hint: *Your CPU can tell you if virtualization is enabled on your system using cpu-checker, the kvm-ok command will tell you if you can use KVM on your system.*

You can test that Libvirt is fully functional by issuing `virsh version` and `virsh list` and checking if the daemon is running and does not show any errors.

Networking

The goal of the next set of tasks is to create the networking environment for your virtual machines. First, we create an virtual Ethernet network to which the VMs will be connected. In Linux we can create such a network using a virtual Ethernet bridge.

Task 4. Since `brctl` is deprecated, we will use the `iproute2` package to create a bridge. Using the `ip` command, manually create a bridge named `kvmbr0`. Do not add any interfaces to it; we will use routing instead of switching to connect the VMs to the Internet.

Task 5. When creating the bridge, Linux will also create a network interface called `kvmbr0` that connects your server to that bridge. The IPv4 addresses to use for your VMs are those in the /28 subnet which is routed to your server (see SNE students mailing list). Assign the first free IPv4 address from your /28 subnet to this `kvmbr0` network interface using `ip addr`. The first free address is your subnet is the address at the start of the range plus 1. The starting address of the range is reserved to act as a network address (e.g. 145.100.104.0) in currrent Internet practice. The last address in the range is reserved to act as the broadcast address (see RFC3021). The address you assign will act as the IP gateway address for your virtual machines.

Task 6. Test whether you can reach the address on the bridge interface from outside of your machine. You may have to enable IPv4 and IPv6 routing (*Hint: `sysctl.conf`*). Make sure that you test using `ping -n` from your workstation or any other machine connected to the Internet. Note that reverse DNS for your /28 subnet is also delegated to you. *Make sure you don't have any firewall filters that prevent forwarding IP traffic.*

Task 7. Edit `/etc/netplan/00-installer-config.yaml` such that `kvmbr0` will persist across system reboots.

KVM Virtual Machines

Task 8. Define the earlier created **VolumeGroupKVM** as a storage pool so newly created VMs can use it. Verify the pool exists and make sure the pool is started, also after a reboot.

Hint: You might find useful to inspect the man page of `virsh`, you need to define, build and start the storage pool before it can be used. `https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/9/html/configuring_and_managing_virtualization/managing-storage-for-virtual-machines_configuring-and-managing-virtualization#creating-lvm-based-storage-pools-using-the-cli_assembly_managing-virtual-machine-storage-pools-using-the-cli`

Task 9. Create a new volume in the pool **VolumeGroupKVM** called **guest01volume**, download the latest ubuntu Yammy cloud-init image (https://cloud-images.ubuntu.com/jammy/current/), convert into to a raw image and upload the image to the created volume.

Hint: *wget the image with -kvm in the name. The qemu-img convert command can help you with converting the downloaded qcow2 image. e.g. qemu-img convert ¡source¿ -O raw ¡deationation¿.raw virsh provides an command that allows you to upload the raw disk image directly onto the LVM parition, or you could use the dd command line tool.*

Task 10. Use `virt-install` to create a Ubuntu virtual machine that has the following characteristics:

- Hostname: **Guest-01**
- RAM: 2048MB
- Disk size: 20GB
- VolumeGroup: VolumeGroupKVM
- VolumeName: guest01volume
- Distribution: Ubuntu Jammy
- Virtual CPUs: 2
- IP: `< an IP from your own range>`

Hint: You might find useful to inspect the man page of `virt-install`.
In the previous task we uploaded a image to the LVM disk we setup for the VM, *cloud-images* tend to run a tool called cloud-init to allow customization of an image. Without any customization you will not be able to login into a deployed VM. With the proper options you can provide details to cloud-init to customize your VM via a set of YAMLfiles. You need to define at least 2 cloud-init files, lets call the first `network.yaml` and the second `userdata.yml`. Note that creating/updating users should be enought for this task. See the list below for some usefull resources:

- `https://cloudinit.readthedocs.io/en/latest/explanation/introduction.html#introduction`
- `https://cloudinit.readthedocs.io/en/latest/reference/examples.html#including-users-and-groups`

- `https://cloudinit.readthedocs.io/en/latest/reference/network-config-format-v2.html`

Note this virtual machine will be reachable from the Internet, so don't use dummy passwords and protect your ssh daemon (man hosts.allow).

If all goes well, you can find the details of the vm (by default) in /etc/libvirt/qemu or you can use the *virsh dumpxml* command to view the (xml) data that details the created VM.

Task 11. The MAC address starts with `52:54:00`. Explain why this prefix is used.

Task 12. Start the virtual machine and login to its console and test network connectivity *Hint `virsh console <vm name>`*. Exit by hitting `CTRL + ]`

Task 13. Use `virsh list` to find information about the running VM and then stop it and start it again.

Task 14. Configure your system such that **Guest-01** is auto-started after a reboot.

Task 15. `autoinstall`, `preseed`, `cloud-init` and `kickstart` can be used to aid in bootstrapping of virtual machine images. In fact, with virt-install you can utilize kickstart or cloud-init directly. When would you use any of the aforementioned methods?

Task 16. How do you think that the virtual machine communicates with the outside network in your setup? Draw a simple network diagram showing at least the network cards, the bridges and any routers that might be present. Don't forget to label everything with IP addresses and names.

References

M. Hasenstein. The logical volume manager (lvm), 2001.