

Anthony Lock – Assignment 1

Table of Contents

Anthony Lock – Assignment 1	1
Introduction	2
Commands ran in order of operation to install an ubuntu image through QEMU	2
Using QEMU Utilities.....	2
Configuring Xen.....	2
Creating a virtual machine	3
QEMU versus KVM	3
QEMU Advantages:	4
QEMU Disadvantages:	4

Introduction

Our task is to document the process we undertook to use QEMU utilities to create an image file of a Linux operating system, and then convert the QEMU image to be used within Xen and then be able to manage and maintain the life cycle of the newly created virtual machine with the use of Xen utilities.

Commands ran in order of operation to install an ubuntu image through QEMU

Commands listed are in *Italic font style*, comments are normal.

Using QEMU Utilities

- *cd /var/lib/xen/images/ubuntu-network/*
 - Change the current working directory to ubuntu-network
- *sudo wget <https://releases.ubuntu.com/16.04.7/ubuntu-16.04.6-server-i386.iso>*
 - Download a ubuntu server image to the current working directory, The image of choice is Ubuntu 16.04 server edition.
- *sudo apt-get install qemu-utils -y*
 - Install the necessary tools for using QEMU.
- *sudo qemu-img create -f raw xenqemu.img 20G*
 - Running the next command, I discovered I needed to use Mobaxterm for the terminal connection because it gives me the ability to open up a UI window whereas when I was using Putty to ssh, it wouldn't let me open us the secondary window.
 - This command also creates the working virtual machine using QEMU
- *sudo qemu-system-x86_64 -m 4096 -hda xenqemu.img -cdrom /var/lib/xen/images/ubuntu-network/ubuntu-16.04.6-server-i386.iso*
 - Username for QEMU install = ubuntu
 - Password for QEMU install = qemuubuntu

Configuring Xen

Converted image (Version 1)

- *qemu-img convert xenqemu.img -O raw newqemu.img*
 - Convert the current qemu image to be able to be used with Xen tools
- *sudo lvcreate -L 30G -n qemuxen /dev/IN720AnthonyUbuntu-vg*
 - Command to create the volume of 30gb which will be used by the Xen image
- *sudo dd if=newqemu.img of=/dev/IN720AnthonyUbuntu-vg/qemuxen*
 - copy and populates the volume with the newly converted image
- *sudo cp xlexample.pvlinux qemu.cfg*
 -
- *sudo vim /etc/xen/qemu.cfg*
 - Changes that had to be made in the qemu configuration file:

- comment out the line with 'kernel'
- Change disk to '/dev/IN720UbuntuAnthony-vg/qemuxen,raw,xvda,rw', so it reads this volume group for the active disk.

Using QEMU image (Version 2)

- `sudo cp qemu-img.cfg test.cfg`
 -
- `sudo vim /etc/xen/test.cfg`

```
# Guest name
name = "xenqemu"

# 128-bit UUID for the domain as a hexadecimal number.
# Use "uuidgen" to generate one if required.
# The default behavior is to generate a new UUID each time the guest
#uuid = "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX"

# Kernel image to boot
#kernel = "/boot/vmlinuz"
bootloader = "/usr/lib/xen-4.4/bin/pygrub"
# Ramdisk (optional)
#ramdisk = "/boot/initrd.gz"

# Kernel command line options
extra = "root=/dev/xvda1"

# Initial memory allocation (MB)
memory = 2048

# Maximum memory (MB)
# If this is greater than 'memory' then the slack will start balloon
# (this assumes guest kernel support for ballooning)
#maxmem = 512

# Number of VCPUS
vcpus = 2

# Network devices
# A list of 'vifspec' entries as described in
# docs/misc/xl-network-configuration.markdown
vif = [ '' ]

# Disk Devices
# A list of 'diskspec' entries as described in
# docs/misc/xl-disk-configuration.txt
disk = [ 'file:/home/anthony/xenqemu.img,hda,rw' ]
boot = "c"
```

○

Creating a virtual machine

- `sudo xl create -c /etc/xen/test.cfg`
 - Used to launch and create the machine.
 - `sudo xl console xenqemu`
 - Connect to the machine
 - `sudo xl shutdown xenqemu`
 - Used to shutdown the virtual machine
- `Ping google.com`
 - Ping test used for testing connection to the internet.

```
64 bytes from syd15s04-in-f14.1e100.net (216.58.196.142): icmp_seq=1 ttl=114 time=47.2 ms
64 bytes from syd15s04-in-f14.1e100.net (216.58.196.142): icmp_seq=2 ttl=114 time=50.8 ms
64 bytes from syd15s04-in-f14.1e100.net (216.58.196.142): icmp_seq=3 ttl=114 time=46.9 ms
64 bytes from syd15s04-in-f14.1e100.net (216.58.196.142): icmp_seq=4 ttl=114 time=47.8 ms
^C
--- google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 46.999/48.217/50.838/1.542 ms
```

○

QEMU versus KVM

QEMU is a 'quick emulator' which runs on top of already existing software and uses a virtual computer to run, where as KVM runs on physical hardware of the machine, allowing virtual machines to access raw unaffected processing power without any latency.

QEMU Advantages:

- Quick and easy to create new images
- Dose not require a kernel module for use
- Fully open source
- It supports a wide range of processors so can run on a wide range of machines

QEMU Disadvantages:

- Only emulates on software level
- Hard to use for long term management
- Virtual processing is slow