Virtualisation with KVM

Or “What I did with my weekend”

Chris Wilson, Aptivate Ltd.
Presented at AfNOG 2013

Download this presentation at:
http://www.ws.afnog.org/afnog2013/sse/virtualisation

What is KVM?

- Kernel-based Virtual Machine
- Built into recent Linux versions
  - Red Hat, CentOS, Ubuntu
- Full virtualisation (supports almost any OS)
- Management tools:
  - libvirt (command-line)
  - virt-manager (GUI)
- Fast!
- No bare-metal hypervisor

How does KVM compare?

	VMware ESX	Xen	KVM	VirtualBox
Vendor	VMware	Citrix	Red Hat	Oracle
Target market	Server	Server	Server	Desktop
Bare metal	Yes	Yes	No	No
Full virt	Yes	Yes	Optional	Yes
Paravirt drivers	Yes	Yes	Yes	Yes
Runs on	Bare metal (built-in RHEL)	Linux, NetBSD, Solaris in dom0	Most Linux distributions	Some Linux distributions, Windows, OSX
PCI pass thru	Yes	Yes	Yes	No
Guest storage	IDE, SCSI, USB, FC, LVM, iSCSI, NFS, filesystems	IDE, SCSI, USB, FC, LVM, iSCSI, filesystems	IDE, SCSI, USB, FC, LVM, iSCSI, filesystems	iSCSI, filesystems
libvirt support	Yes	Yes	Yes	Yes
Price	Free (ESXi)	Free (Debian)	Free (CentOS)	Free

Why KVM?

- Advantages of KVM:
  - Officially supported on RHEL, CentOS, Fedora, Debian and Ubuntu
  - Lightweight and efficient
  - Supported by libvirt
  - Close integration with Linux, automated installation
- Disadvantages of KVM:
  - Relatively new, fewer guest OS paravirtual drivers
  - Only runs on Linux hosts (+experimental FreeBSD)
  - GUI tools are less well developed

KVM storage

- Everything supported by Qemu userspace
- Disk image files
  - virt-install options:
    - Disk image files: `--disk path=<image-file>`
    - CD-ROM ISO images: `--cdrom <image-file>`
  - formats: vvfat vpc vmdk vdi sheepdog rbd raw host_cdrom host_floppy host_device file qed qcow2 qcow parallels nbd dmg tftp ftps ftp https http cow cloop
- Partitions (raw)
  - Use LVM for flexibility!
- Host physical devices

KVM networking

- Everything supported by Qemu userspace
- Automatic mode
  - If there is a bridge device in the host with a physical interface enslaved, that will be used for connectivity.
- Bridged networking
  - virt-install option: `--network=bridge=<device>`
  - Note: *device* must be a Linux bridge device, e.g. *br0*
  - Generally does not work on wireless interfaces!
- NAT networking
  - virt-install option: `--network=network=default`

Installing the Guest OS (demo)

- `sudo virt-install --connect qemu:///system \`
  `--virt-type kvm --name FreeBSD-Demo \`
  `--os-variant=freebsd8 --ram 1024 --vcpus 1 \`
  `--disk path=FreeBSD-Demo.img,size=20 \`
  `--cdrom FreeBSD-9.1-RELEASE-amd64-dvd1.iso \`
  `--network=bridge=br219 \`
  `--graphics type=vnc,listen=0.0.0.0`
- Note: br219 is specific to our setup
  - You probably want to omit the –`network` option
- Connect to <host>:0 with VNC to drive the install
- Live demo!

Deleting a virtual machine (guest)

- If you make a mistake with *virt-install*
- `virsh destroy FreeBSD-Demo`
  - stops the guest VM with a hard virtual power-off
- `virsh undefine FreeBSD-Demo`
  - deletes the guest VM XML file from */etc/libvirt/qemu*
  - cannot start or stop the guest VM any more
- `rm FreeBSD-Demo.img`
  - deletes the disk image file (~ 20GB in this case)
- Then you can run *virt-install* to create it again

After OS installation

- The installer will eject the virtual CD-ROM
  - libvirt detaches the CD-ROM disk image
  - after reboot there will be no CD-ROM in virtual drive
  - as shown with `virsh domblklist FreeBSD-Demo`
- The installer will reboot the virtual machine
- But it won't come back up – why?
  - Look at `virsh list --all`
  - Need to start it manually
  - Configure to autostart with
    `virsh autostart FreeBSD-Demo`

Creating a Gold Image

- Shut down the gold system cleanly
  - `shutdown -p now` or `poweroff`
- Check that it's not running
  - `virsh list –all`
- Copy the image file (why?)
  - `sudo cp FreeBSD-Demo.img FreeBSD-Demo-Gold.img`

Creating a clone

- `hostname=pc$pc`
- ```
  macaddr=`echo $hostname | md5sum | sed -e \
  's/^\(..\)\(..\)\(..\)\(..\).*/52:54:\1:\2:\3:\4/'`
  ```
- `image=/data/vm/$hostname.img`
  - `sudo qemu-img create -f qcow2 -o backing_file=FreeBSD-Demo-Gold.img $image`
- ```
  virt-install --connect qemu:///system \
      --virt-type kvm --name $hostname \
      --os-variant=freebsd8 --ram 512 --vcpus 1 \
      --disk path=$image,format=qcow2 \
      --network=bridge=br219,mac=$macaddr \
      --graphics type=vnc,listen=0.0.0.0 --import
  ```
- Live demo!
  - What would I have to change, to create another one?

More clones!

```
for pc in {1..32}; do
    hostname=pc$pc
    macaddr=`echo $hostname | md5sum | sed -e 's/^\
(..\)\(..\)\(..\)\(..\).*/52:54:\1:\2:\3:\4/'`
    image=/data/vm/$hostname.img
    sudo qemu-img create -f qcow2 \
        -o backing_file=FreeBSD-SSE-Gold.img $image
    virt-install --connect qemu:///system \
        --virt-type kvm --name $hostname \
        --os-variant=freebsd8 --ram 512 --vcpus 1 \
        --disk path=$image,format=qcow2 \
        --network=bridge=br219,mac=$macaddr \
        --graphics type=vnc,listen=0.0.0.0 --import
done
```

Questions

- How big are the disk images?
  - How big could they become?
- What hostname do the machines have?
  - How would you set it automatically?
  - Use DHCP, set `hostname=""` in */etc/rc.conf*
- How can you manage them in bulk?
  - How do you deal with OS updates?
- What is the system clock set to?
- What happens to system logs?
- What are the SSH keys of these systems?

Using DHCP for fixed IP addresses

```
for i in {17..32}; do
hostname=$1
ipaddr=$2

if [ -z "$ipaddr" ]; then
echo "Usage: $0 <hostname> <ip-address>" >&2
exit 2
fi

macaddr=`echo $hostname | md5sum | sed -e 's/^\
(..\)\(..\)\(..\)\(..\).*/52:54:\1:\2:\3:\4/'`
cat <<EOF
host $hostname {
hardware ethernet $macaddr;
fixed-address $ipaddr;
}
EOF

exit 0
```

FIN

Any questions?

Virtualisation with KVM

Or “What I did with my weekend”

Chris Wilson, Aptivate Ltd.
Presented at AfNOG 2013

Download this presentation at:
http://www.ws.afnog.org/afnog2013/sse/virtualisation

What is KVM?

- Kernel-based Virtual Machine
- Built into recent Linux versions
  - Red Hat, CentOS, Ubuntu
- Full virtualisation (supports almost any OS)
- Management tools:
  - libvirt (command-line)
  - virt-manager (GUI)
- Fast!
- No bare-metal hypervisor

How does KVM compare?

	VMware ESX	Xen	KVM	VirtualBox
Vendor	VMware	Citrix	Red Hat	Oracle
Target market	Server	Server	Server	Desktop
Bare metal	Yes	Yes	No	No
Full virt	Yes	Yes	Optional	Yes
Paravirt drivers	Yes	Yes	Yes	Yes
Runs on	Bare metal (built-in RHEL)	Linux, NetBSD, Solaris in dom0	Most Linux distributions	Some Linux distributions, Windows, OSX
PCI pass thru	Yes	Yes	Yes	No
Guest storage	IDE, SCSI, USB, FC, LVM, iSCSI, NFS, filesystems	IDE, SCSI, USB, FC, LVM, iSCSI, filesystems	IDE, SCSI, USB, FC, LVM, iSCSI, filesystems	iSCSI, filesystems
libvirt support	Yes	Yes	Yes	Yes
Price	Free (ESXi)	Free (Debian)	Free (CentOS)	Free

Why KVM?

- Advantages of KVM:
  - Officially supported on RHEL, CentOS, Fedora, Debian and Ubuntu
  - Lightweight and efficient
  - Supported by libvirt
  - Close integration with Linux, automated installation
- Disadvantages of KVM:
  - Relatively new, fewer guest OS paravirtual drivers
  - Only runs on Linux hosts (+experimental FreeBSD)
  - GUI tools are less well developed

KVM storage

- Everything supported by Qemu userspace
- Disk image files
  - virt-install options:
    - Disk image files: `--disk path=<image-file>`
    - CD-ROM ISO images: `--cdrom <image-file>`
  - formats: vvfat vpc vmdk vdi sheepdog rbd raw host_cdrom host_floppy host_device file qed qcow2 qcow parallels nbd dmg tftp ftps ftp https http cow cloop
- Partitions (raw)
  - Use LVM for flexibility!
- Host physical devices

KVM networking

- Everything supported by Qemu userspace
- Automatic mode
  - If there is a bridge device in the host with a physical interface enslaved, that will be used for connectivity.
- Bridged networking
  - virt-install option: `--network=bridge=<device>`
  - Note: *device* must be a Linux bridge device, e.g. *br0*
  - Generally does not work on wireless interfaces!
- NAT networking
  - virt-install option: `--network=network=default`

Installing the Guest OS (demo)

- ```
  sudo virt-install --connect qemu:///system \
    --virt-type kvm --name FreeBSD-Demo \
    --os-variant=freebsd8 --ram 1024 --vcpus 1 \
    --disk path=FreeBSD-Demo.img,size=20 \
    --cdrom FreeBSD-9.1-RELEASE-amd64-dvd1.iso \
    --network=bridge=br219 \
    --graphics type=vnc,listen=0.0.0.0
  ```
- Note: br219 is specific to our setup
  - You probably want to omit the –`network` option
- Connect to <host>:0 with VNC to drive the install
- Live demo!


  ```

Deleting a virtual machine (guest)

- If you make a mistake with *virt-install*
- `virsh destroy FreeBSD-Demo`
  - stops the guest VM with a hard virtual power-off
- `virsh undefine FreeBSD-Demo`
  - deletes the guest VM XML file from */etc/libvirt/qemu*
  - cannot start or stop the guest VM any more
- `rm FreeBSD-Demo.img`
  - deletes the disk image file (~ 20GB in this case)
- Then you can run *virt-install* to create it again

After OS installation

- The installer will eject the virtual CD-ROM
  - libvirt detaches the CD-ROM disk image
  - after reboot there will be no CD-ROM in virtual drive
  - as shown with `virsh domblklist FreeBSD-Demo`
- The installer will reboot the virtual machine
- But it won't come back up – why?
  - Look at `virsh list --all`
  - Need to start it manually
  - Configure to autostart with
    `virsh autostart FreeBSD-Demo`

Creating a Gold Image

- Shut down the gold system cleanly
  - `shutdown -p now` or `poweroff`
- Check that it's not running
  - `virsh list –all`
- Copy the image file (why?)
  - `sudo cp FreeBSD-Demo.img FreeBSD-Demo-Gold.img`

Creating a clone

- `hostname=pc$pc`
- ```
  macaddr=`echo $hostname | md5sum | sed -e \
  's/^\(..\)\(..\)\(..\)\(..\).*/52:54:\1:\2:\3:\4/'`
  ```
- `image=/data/vm/$hostname.img`
  - `sudo qemu-img create -f qcow2 -o backing_file=FreeBSD-Demo-Gold.img $image`
- ```
  virt-install --connect qemu:///system \
      --virt-type kvm --name $hostname \
      --os-variant=freebsd8 --ram 512 --vcpus 1 \
      --disk path=$image,format=qcow2 \
      --network=bridge=br219,mac=$macaddr \
      --graphics type=vnc,listen=0.0.0.0 --import
  ```
- Live demo!
  - What would I have to change, to create another one?

More clones!

```
for pc in {1..32}; do
    hostname=pc$pc
    macaddr=`echo $hostname | md5sum | sed -e 's/^\
(..\)\(..\)\(..\)\(..\).*/52:54:\1:\2:\3:\4/'`
    image=/data/vm/$hostname.img
    sudo qemu-img create -f qcow2 \
        -o backing_file=FreeBSD-SSE-Gold.img $image
    virt-install --connect qemu:///system \
        --virt-type kvm --name $hostname \
        --os-variant=freebsd8 --ram 512 --vcpus 1 \
        --disk path=$image,format=qcow2 \
        --network=bridge=br219,mac=$macaddr \
        --graphics type=vnc,listen=0.0.0.0 --import
done
```

Questions

- How big are the disk images?
  - How big could they become?
- What hostname do the machines have?
  - How would you set it automatically?
  - Use DHCP, set `hostname=""` in */etc/rc.conf*
- How can you manage them in bulk?
  - How do you deal with OS updates?
- What is the system clock set to?
- What happens to system logs?
- What are the SSH keys of these systems?

Using DHCP for fixed IP addresses

```
for i in {17..32}; do
hostname=$1
ipaddr=$2

if [ -z "$ipaddr" ]; then
echo "Usage: $0 <hostname> <ip-address>" >&2
exit 2
fi

macaddr=`echo $hostname | md5sum | sed -e 's/^\
(..\)\(..\)\(..\)\(..\).*/52:54:\1:\2:\3:\4/'`
cat <<EOF
host $hostname {
hardware ethernet $macaddr;
fixed-address $ipaddr;
}
EOF

exit 0
```

FIN

Any questions?