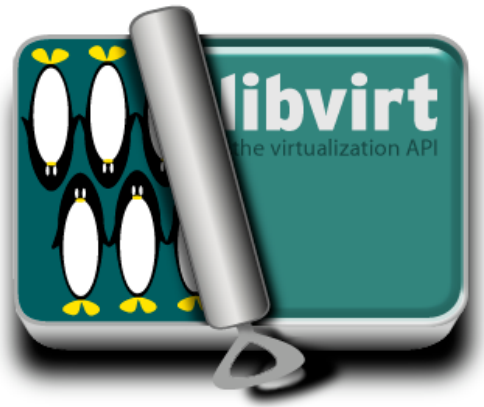
Introduction to KVM

By Sheng-wei Lee
swlee@swlee.org #20110929

Outline

- Hypervisor – KVM
- virt-manager
- Libvirt
- Migration



Outline

- How to install KVM.
- Bridged Networking
- Virsh
  How to new a VM
  How to adjust the setting of a VM.
  How to make an image of a VM
  How to new a VM using an existed image
  How to close a VM.
- Virt-manager (VMM)

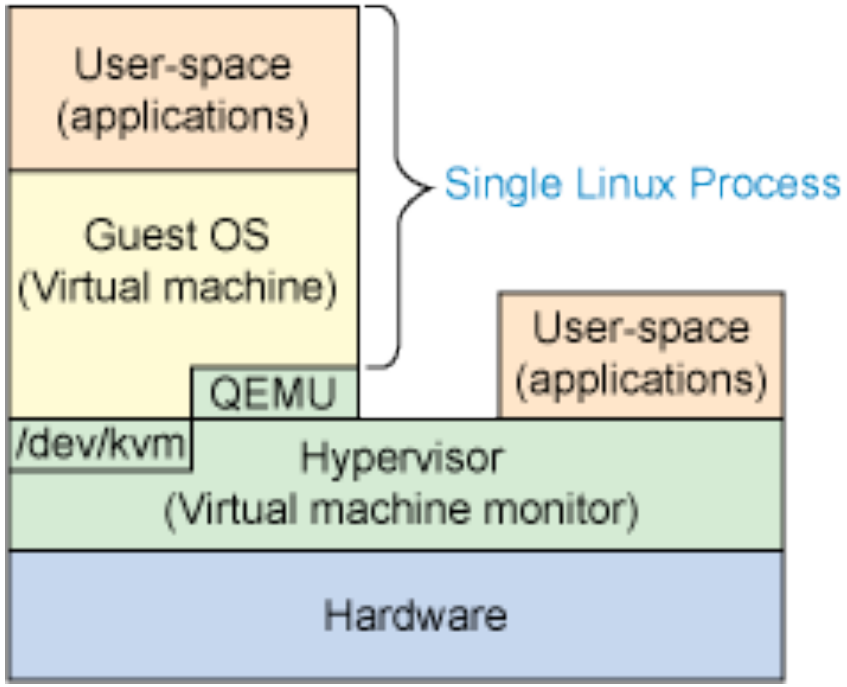
KVM – Kernel Based Virtual Machine

- KVM (for Kernel-based Virtual Machine) is a full virtualization solution for Linux on x86 hardware containing virtualization extensions (Intel VT or AMD-V). It consists of a loadable kernel module, kvm.ko, that provides the core virtualization infrastructure and a processor specific module, kvm-intel.ko or kvm-amd.ko. KVM also requires a modified QEMU although work is underway to get the required changes upstream.

KVM – Kernel Based Virtual Machine

- Using KVM, one can run multiple virtual machines running unmodified Linux or Windows images. Each virtual machine has private virtualized hardware: a network card, disk, graphics adapter, etc.
- The kernel component of KVM is included in mainline Linux, as of 2.6.20.
- KVM is open source software.

The virtualization components with KVM



Hypervisor – KVM

- **Guest Support Status**
  http://www.linux-kvm.org/page/Guest_Support_Status

Virt-manager

The **"Virtual Machine Manager"** application (virt-manager for short package name) is a desktop user interface for managing virtual machines. It presents a summary view of running domains, their live performance & resource utilization statistics. The detailed view graphs performance & utilization over time. Wizards enable the creation of new domains, and configuration & adjustment of a domain's resource allocation & virtual hardware. An embedded VNC client viewer presents a full graphical console to the guest domain.

- **libvirt supports:**
- The Xen hypervisor on Linux and Solaris hosts.
- The QEMU emulator
- The KVM Linux hypervisor
- The LXC Linux container system
- The OpenVZ Linux container system
- The User Mode Linux paravirtualized kernel
- The VirtualBox hypervisor
- The VMware ESX and GSX hypervisors
- The VMware Workstation and Player hypervisors
- Storage on IDE/SCSI/USB disks, FibreChannel, LVM, iSCSI, NFS and filesystems

- See also:

http://www.ibm.com/developerworks/linux/library/l-libvirt/index.html

- Storage drivers
- Directory backend
- Local filesystem backend
- Network filesystem backend
- Logical Volume Manager (LVM) backend
- Disk backend
- iSCSI backend
- SCSI backend
- Multipath backend

KVM – Migration

- KVM currently supports savevm/loadvm and offline or live migration Migration commands are given when in qemu-monitor (Alt-Ctrl-2). Upon successful completion, the migrated VM continues to run on the destination host.

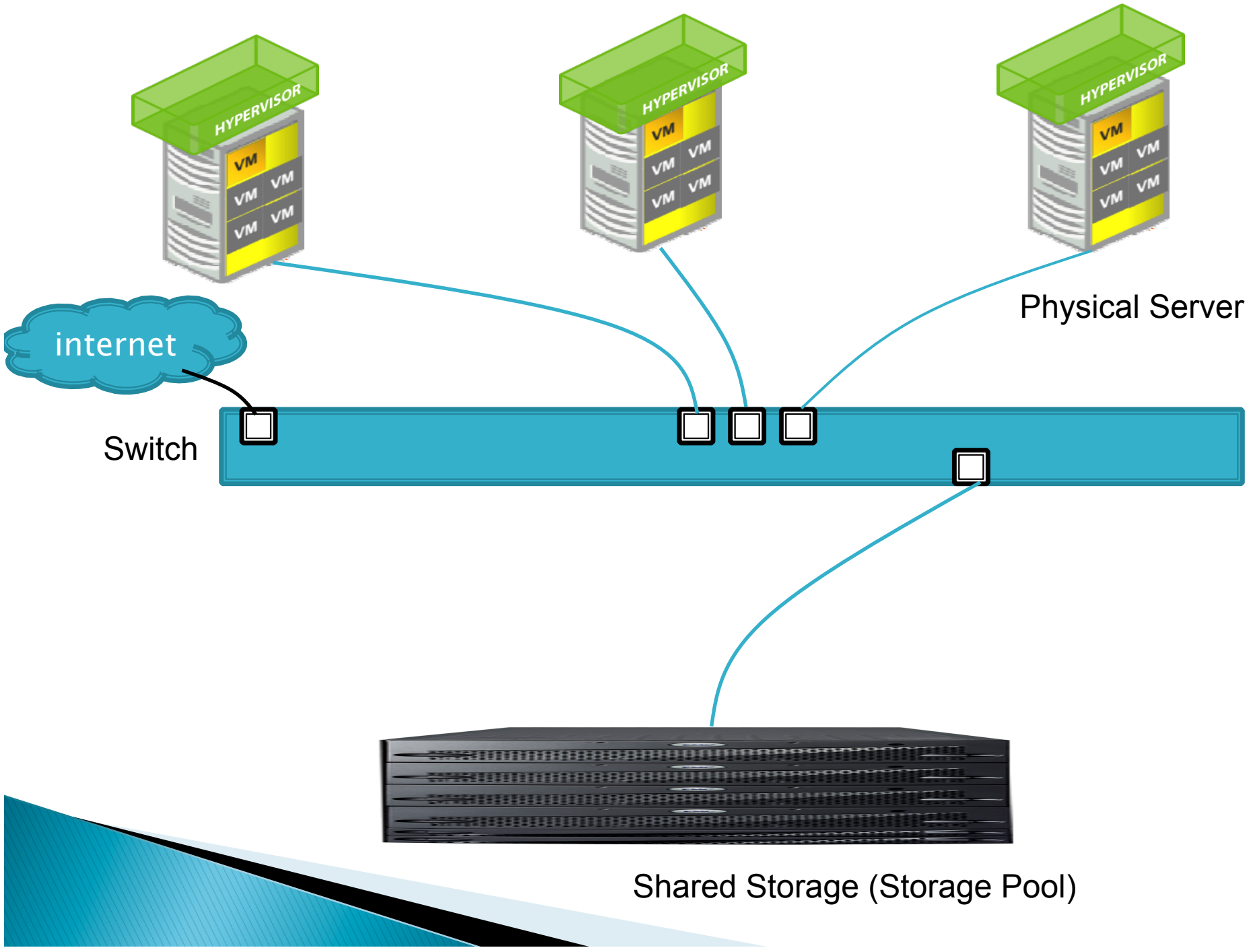
KVM – Migration

- **Note**

  You can migrate a guest between an AMD host to an Intel host and back. Naturally, a 64-bit guest can only be migrated to a 64-bit host, but a 32-bit guest can be migrated at will.

KVM – Migration

- Requirements
- The VM image is accessible on both source and destination hosts (located on a shared storage, e.g. using nfs).
- It is recommended an images-directory would be found on the same path on both hosts (for migrations of a copy-on-write image -- an image created on top of a base-image using "qemu-image create -b ...")
- The src and dst hosts must be on the same subnet (keeping guest's network when tap is used).
- Do not use -snapshot qemu command line option.
- For tcp: migration protocol
- the guest on the destination must be started the same way it was started on the source.

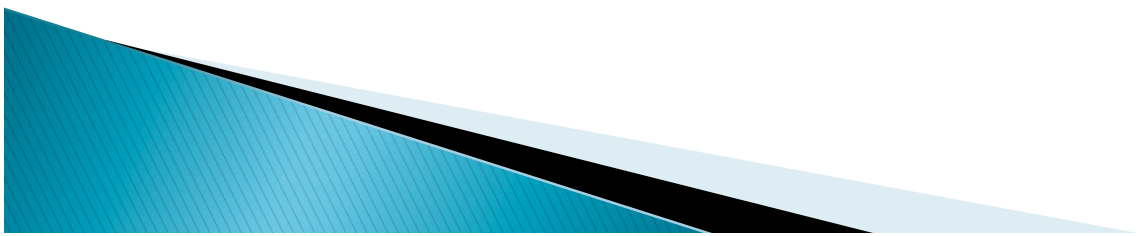

HYPERVISOR
VM
VM
VM
VM
HYPERVISOR
VM
VM
VM
VM
HYPERVISOR
VM
VM
VM
VM
Physical Server
internet
Switch
Shared Storage (Storage Pool)

How to install KVM.

- https://help.ubuntu.com/community/KVM
- Check that your CPU supports hardware virtualization
- To run KVM, you need a processor that supports hardware virtualization. Intel and AMD both have developed extensions for their processors, deemed respectively Intel VT-x (code name Vanderpool) and AMD-V (code name Pacifica). To see if your processor supports one of these, you can review the output from this command:
- egrep -c '(vmx|svm)' /proc/cpuinfo

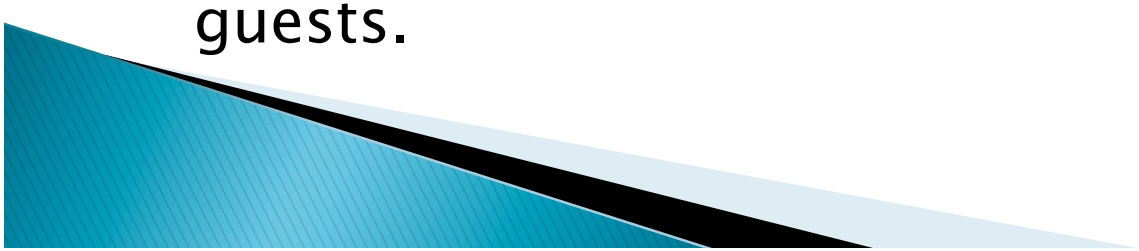
How to install KVM.

- If 0 it means that your CPU doesn't support hardware virtualization.
- If 1 (or more) it does - but you still need to make sure that virtualization is enabled in the BIOS.



How to install KVM.

- Use a 64 bit kernel (if possible)
- Running a 64 bit kernel on the host operating system is recommended but not required.
- To serve more than 2GB of RAM for your VMs, you **must** use a 64-bit kernel (see 32bit_and_64bit). On a 32-bit kernel install, you'll be limited to 2GB RAM at maximum for a given VM.
- Also, a 64-bit system can host both 32-bit and 64-bit guests. A 32-bit system can only host 32-bit guests.



How to install KVM.

- To see if your processor is 64-bit, you can run this command:
  egrep -c ' lm ' /proc/cpuinfo

- If **0** is printed, it means that your CPU is not 64-bit.
- If **1** or higher, it is.
  Note: *lm* stands for Long Mode which equates to a 64-bit CPU.

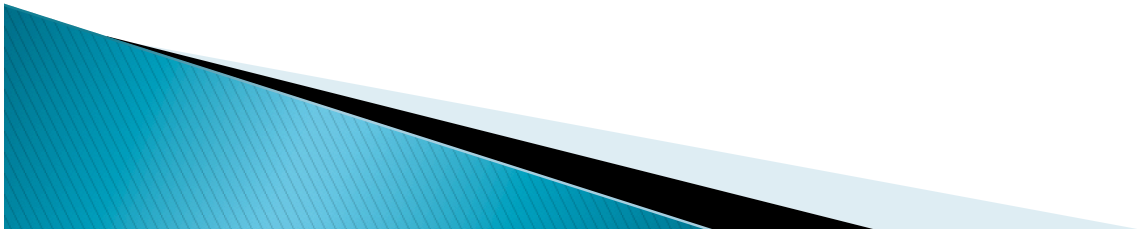
- Now see if your running kernel is 64-bit, just issue the following command:
  uname -m

- **x86_64** indicates a running 64-bit kernel. If you use see i386, i486, i586 or i686, you're running a 32-bit kernel.
- Note: x86_64 is synonymous with amd64.

How to install KVM.

- Install Necessary Packages
- For the following setup, we will assume that you are deploying KVM on a server, and therefore do not have any X server on the machine.

- **Lucid (10.04) or later**
- $ sudo apt-get install qemu-kvm libvirt-bin ubuntu-vm-builder bridge-utils



How to install KVM.

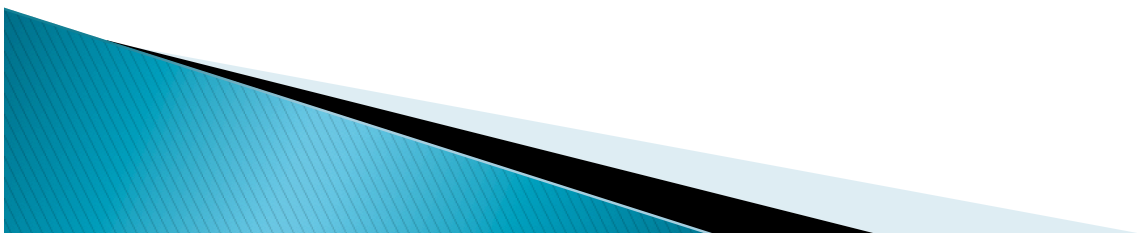
- Add Users to Groups

- To check:

```
$ groups
adm dialout cdrom floppy audio dip video plugdev fuse lpadmin
   admin sambashare kvm libvirtd
```

- To add your <username> to the groups:

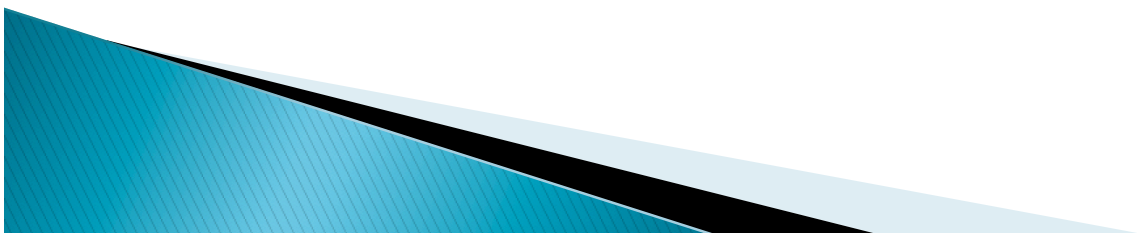
```
 $ sudo adduser `id -un` kvm
   Adding user '<username>' to group 'kvm' ...
$ sudo adduser `id -un` libvirtd
  Adding user '<username>' to group 'libvirtd' ...
```



How to install KVM.

- Verify Installation
- You can test if your install has been successful with the following command:

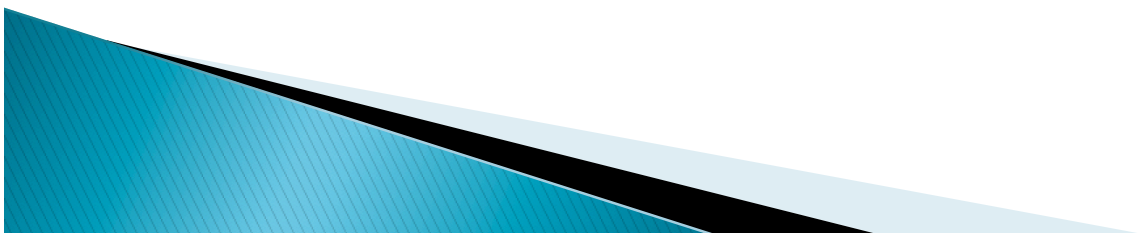
```
$ virsh -c qemu:///system list
 Id Name                                 State
 ----------------------------------------------
```



How to install KVM.

- If on the other hand you get something like this:

```
$ virsh -c qemu:///system list
libvir: Remote error : Permission denied
error: failed to connect to the hypervisor
```



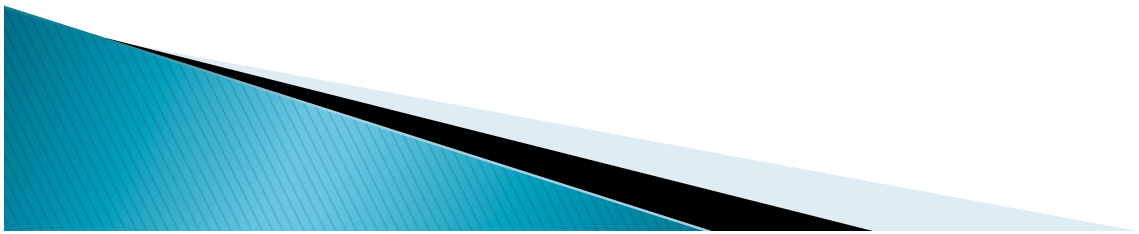
Bridged Networking

- Creating a network bridge on the host
- Install the bridge-utils package:

  $sudo apt-get install bridge-utils

- We are going to change the network configuration[1]. To do it properly, you should first stop networking[2]:

  $sudo invoke-rc.d networking stop/restart



Bridged Networking

edit /etc/network/interfaces

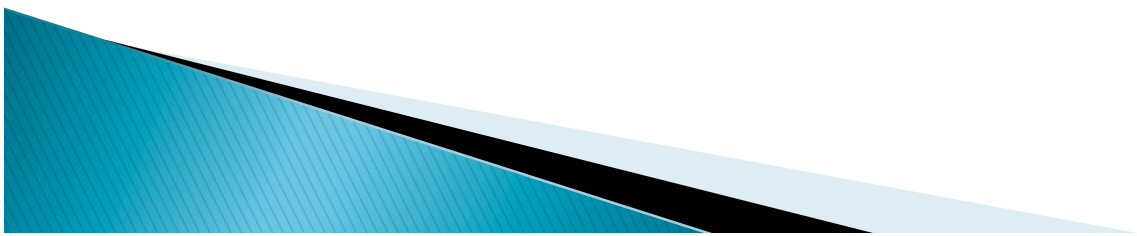
```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet manual

auto br0
iface br0 inet static
    Address  <your_IP>
    network  <network>
    netmask  <netmask>
    Broadcast  <broadcast>
    gateway  <gateway>
    bridge_ports eth0
    bridge_stp off
    bridge_fd 0
    bridge_maxwait 0
```

Bridged Networking

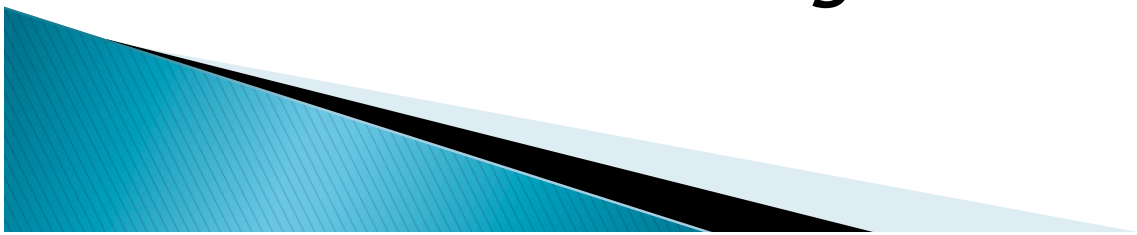
```
$ sudo /etc/init.d/networking restart
```



Virsh – How to new a VM

- **Creating a guest**
- Guests can be created from XML configuration files. You can copy existing XML from previously created guests or use the dumpxml option(refer to Creating a virtual machine XML dump(configuration file)). To create a guest with virsh from an XML file:

$ virsh create *configuration_file.xml*



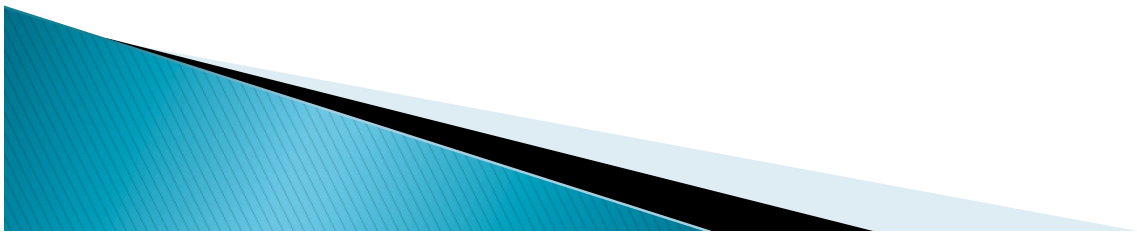
Virsh – How to new a VM

- Alternatively, if you want to define it, but not run it, you could have used:

  $ virsh define /tmp/foo_new.xml

- Once a virtual machine is running, you can manage it in many different ways, such as:

  $ virsh start foo

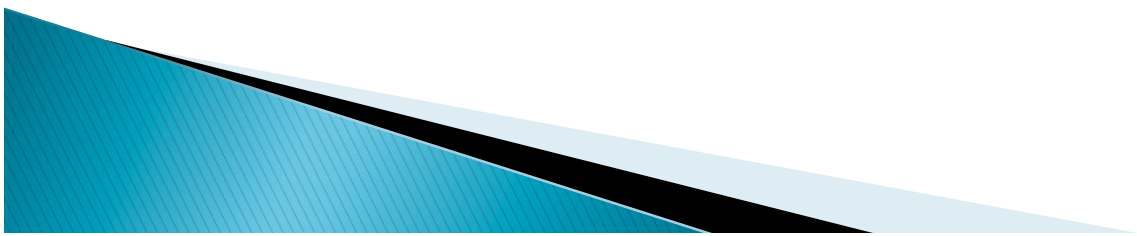


Virsh – How to new a VM

- **Creating a virtual machine XML dump(configuration file)**

- To perform a data dump for an existing guest with virsh:

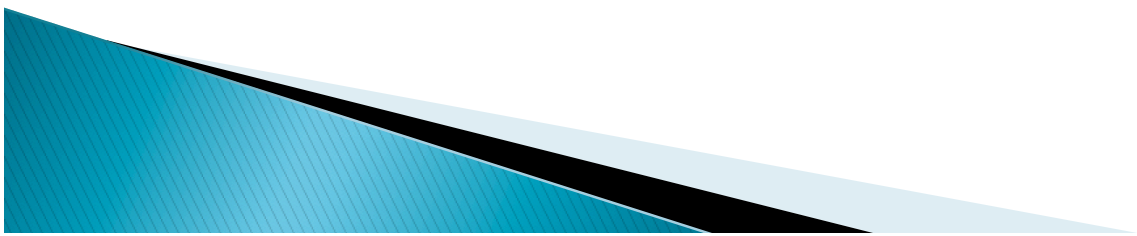
$ virsh dumpxml [domain-id, domain-name or domain-uuid] > <domain>.xml



Virsh – How to new a VM

- You can perform the following to install Ubuntu Hardy:

```
$ sudo virt-install --connect qemu:///system\
  -n hardy -r 512 -f hardy.qcow2 -s 12 /
-c hardy-server-amd64.iso --vnc --
  noautoconsole --os-type linux --os-variant
  ubuntuHardy --accelerate --
  network=network:default
```

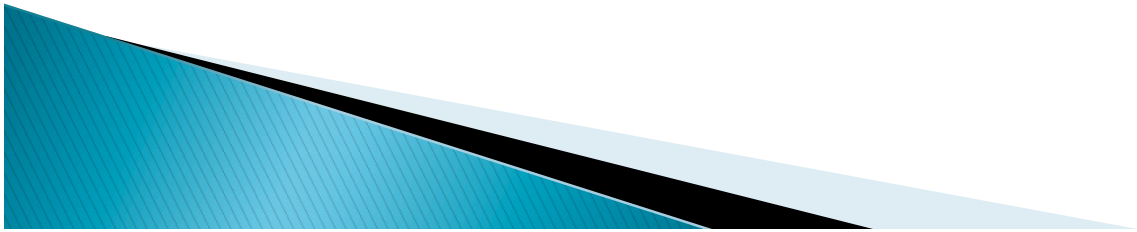


Virsh – How to adjust the setting of a VM.

- <domain type='kvm'>
- <name>Ubuntu-11.04-i686_Base</name>
- <uuid>4b4c19e8-9d76-0c9d-cbf8-12141823d393</uuid>
- <memory>524288</memory>
- <currentMemory>524288</currentMemory>
- <vcpu>2</vcpu>
- <os>
- <type arch='i686' machine='pc-0.14'>hvm</type>
- <boot dev='cdrom'/>
- <boot dev='hd'/>
- <bootmenu enable='no'/>
- </os>

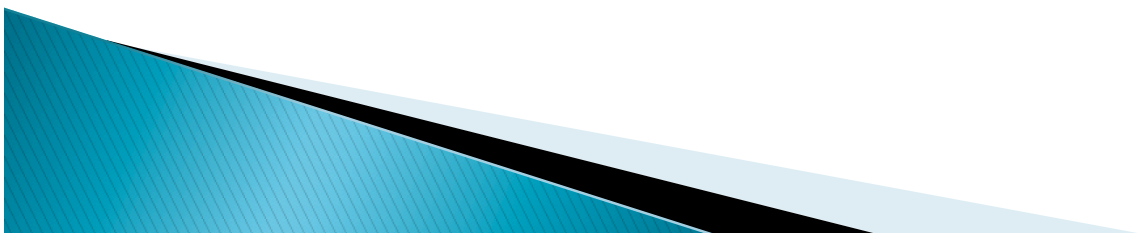
Virsh – How to adjust the setting of a VM.

```
<features>
  <acpi/>
  <apic/>
  <pae/>
 </features>
 <clock offset='utc'/>
 <on_poweroff>destroy</on_poweroff>
 <on_reboot>restart</on_reboot>
 <on_crash>restart</on_crash>
 <devices>
  <emulator>/usr/bin/kvm</emulator>
  <disk type='file' device='disk'>
   <driver name='qemu' type='qcow2'/>
   <source file='/Storage/local/Base/Ubuntu-11.04-i686_Base.qcow2'/>
   <target dev='hda' bus='ide'/>
   <address type='drive' controller='0' bus='0' unit='0'/>
  </disk>
```



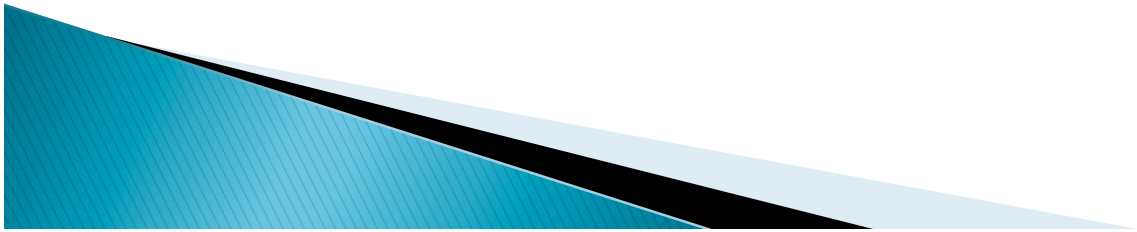
Virsh – How to adjust the setting of a VM.

- <disk type='file' device='cdrom'>
- <driver name='qemu' type='raw'/>
- <target dev='hdc' bus='ide'/>
- <readonly/>
- <address type='drive' controller='0' bus='1' unit='0'/>
- </disk>
- <controller type='ide' index='0'>
- <address type='pci' domain='0x0000' bus='0x00' slot='0x01' function='0x1'/>
- </controller>
- <interface type='network'>
- <mac address='52:54:00:4a:9a:02'/>
- <source network='default'/>
- <address type='pci' domain='0x0000' bus='0x00' slot='0x03' function='0x0'/>
- </interface>
-



Virsh – How to adjust the setting of a VM.

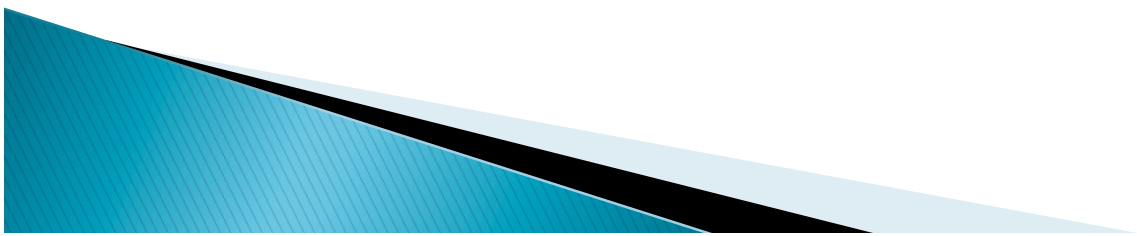
```
<serial type='pty'>
      <target port='0'/>
    </serial>
    <console type='pty'>
      <target type='serial' port='0'/>
    </console>
    <input type='mouse' bus='ps2'/>
    <graphics type='vnc' port='-1' autoport='yes'/>
    <sound model='ac97'>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x04' function='0x0'/>
    </sound>
    <video>
      <model type='cirrus' vram='9216' heads='1'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x02' function='0x0'/>
    </video>
    <memballoon model='virtio'>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x05' function='0x0'/>
    </memballoon>
  </devices>
</domain>
```



Virsh – How to make an image of a VM

- Create the hard drive image with qcow2 format:

$ qemu-img create -f qcow2 <image name>.qcow2



Virsh – How to new a VM using an existed image

- Cloning a virtual machine
- You can clone an existing virtual machine using the virt-clone tool. This duplicates the disk image and sets up the virtual machine domain configuration.
- If you wish to clone a virtual machine named *srchost* to a new machine *newhost*, ensure that the virtual machine *srchost* is not running and execute the following command.

```
$ virt-clone --connect=qemu:///system -o srchost -n newhost -
  f /path/to/newhost.qcow2
```

Virsh – How to shut a VM

$ virsh shutdown foo

$ virsh suspend foo

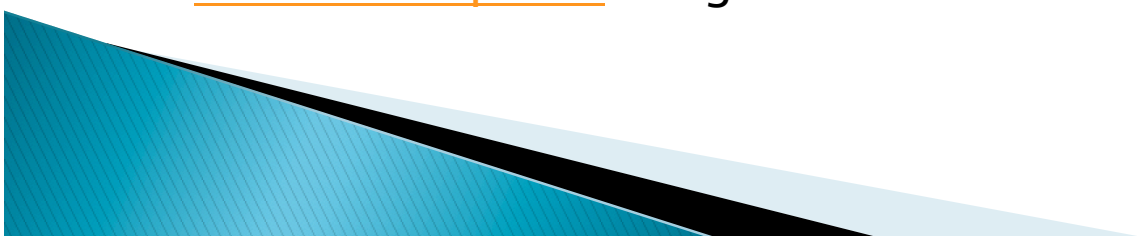
$ virsh resume foo

$ virsh save foo state-file

To save the current state of a guest to a file using the virsh command

$virsh restore foo stat-file

To restore a guest that you previously saved with the virsh save option using the virsh command



How to install VMM.

- Virt-Manager

- If you are working on a desktop computer you might want to install a GUI tool to manage virtual machines.

  $ sudo apt-get install virt-manager