Working with KVM Hypervisors

David Clinton
LINUX SYSTEM ADMINISTRATOR

@davidbclinton | www.bootstrap-it.com | www.bootstrap-it.com/blog

KVM Features

Xen	KVM
Controls hardware	Controls software tools

KVM Features

Xen	KVM
Controls hardware	Controls software tools
HVM and PV	Only HVM

KVM Features

Xen	KVM
Controls hardware	Controls software tools
HVM and PV	Only HVM
	Leverages QEMU as emulator

KVM Features

Xen	KVM
Controls hardware	Controls software tools
HVM and PV	Only HVM
	Leverages QEMU as emulator
	Requires integrated hardware virtualization

KVM Resources

/etc/libvirt/qemu	.XML definition files
/var/lib/libvirt/images	OS host source files
/var/log/libvirt/qemu	Logs
~/.virtinst/	Logs

KVM Resources

/etc/libvirt/qemu	.XML definition files
/var/lib/libvirt/images	OS host source files
/var/log/libvirt/qemu	Logs
~/.virtinst/	Logs

KVM Resources

/etc/libvirt/qemu	.XML definition files
/var/lib/libvirt/images	OS host source files
/var/log/libvirt/qemu	Logs
~/.virtinst/	Logs

KVM Resources

/etc/libvirt/qemu	.XML definition files
/var/lib/libvirt/images	OS host source files
/var/log/libvirt/qemu	Logs
~/.virtinst/	Logs

KVM Management Tools: libvirt

KVM Managers

- libvirt (virsh)
- virt-manager
  - Non-GUI version; including virt-install
- vmbuilder
- KVM

KVM Management Tools: virt-manager

virt-install Parameters

```
sudo virt-install -n ubuntu-vm \
 --connect qemu:///system \
 --description "Ubuntu 14.04 VM" \
 --os-type=Linux \
 --ram=1024 \
 --vcpus=2 \
 --disk path=/var/lib/libvirt/images/ubuntu-vm.img,bus=virtio,size=4 \
 --graphics none \
 --location /home/ubuntu/ubuntu-14.04.4-server-amd64.iso \
 --extra-args='console=tty0 console=ttyS0,115200n8 serial' \
 --network bridge:virbr0
```

virt-install Parameters

```
sudo virt-install -n ubuntu-vm \
 --connect qemu:///system \
 --description "Ubuntu 14.04 VM" \
 --os-type=Linux \
 --ram=1024 \
 --vcpus=2 \
 --disk path=/var/lib/libvirt/images/ubuntu-vm.img,bus=virtio,size=4 \
 --graphics none \
 --location /home/ubuntu/ubuntu-14.04.4-server-amd64.iso \
 --extra-args='console=tty0 console=ttyS0,115200n8 serial' \
 --network bridge:virbr0
```

virt-install Parameters

```
sudo virt-install -n ubuntu-vm \
 --connect qemu:///system \
 --description "Ubuntu 14.04 VM" \
 --os-type=Linux \
 --ram=1024 \
 --vcpus=2 \
 --disk path=/var/lib/libvirt/images/ubuntu-vm.img,bus=virtio,size=4 \
 --graphics none \
 --location /home/ubuntu/ubuntu-14.04.4-server-amd64.iso \
 --extra-args='console=tty0 console=ttyS0,115200n8 serial' \
 --network bridge:virbr0
```

virt-install Parameters

```
sudo virt-install -n ubuntu-vm \
 --connect qemu:///system \
 --description "Ubuntu 14.04 VM" \
 --os-type=Linux \
 --ram=1024 \
 --vcpus=2 \
 --disk path=/var/lib/libvirt/images/ubuntu-vm.img,bus=virtio,size=4 \
 --graphics none \
 --location /home/ubuntu/ubuntu-14.04.4-server-amd64.iso \
 --extra-args='console=tty0 console=ttyS0,115200n8 serial' \
 --network bridge:virbr0
```

virt-install Parameters

```
sudo virt-install -n ubuntu-vm \
 --connect qemu:///system \
 --description "Ubuntu 14.04 VM" \
 --os-type=Linux \
 --ram=1024 \
 --vcpus=2 \
 --disk path=/var/lib/libvirt/images/ubuntu-vm.img,bus=virtio,size=4 \
 --graphics none \
 --location /home/ubuntu/ubuntu-14.04.4-server-amd64.iso \
 --extra-args='console=tty0 console=ttyS0,115200n8 serial' \
 --network bridge:virbr0
```

virt-install Parameters

```
sudo virt-install -n ubuntu-vm \
 --connect qemu:///system \
 --description "Ubuntu 14.04 VM" \
 --os-type=Linux \
 --ram=1024 \
 --vcpus=2 \
 --disk path=/var/lib/libvirt/images/ubuntu-vm.img,bus=virtio,size=4 \
 --graphics none \
 --location /home/ubuntu/ubuntu-14.04.4-server-amd64.iso \
 --extra-args='console=tty0 console=ttyS0,115200n8 serial' \
 --network bridge:virbr0
```

virt-install Parameters

```
sudo virt-install -n ubuntu-vm \
 --connect qemu:///system \
 --description "Ubuntu 14.04 VM" \
 --os-type=Linux \
 --ram=1024 \
 --vcpus=2 \
 --disk path=/var/lib/libvirt/images/ubuntu-vm.img,bus=virtio,size=4 \
 --graphics none \
 --location /home/ubuntu/ubuntu-14.04.4-server-amd64.iso \
 --extra-args='console=tty0 console=ttyS0,115200n8 serial' \
 --network bridge:virbr0
```

virt-install Parameters

```
sudo virt-install -n ubuntu-vm \
 --connect qemu:///system \
 --description "Ubuntu 14.04 VM" \
 --os-type=Linux \
 --ram=1024 \
 --vcpus=2 \
 --disk path=/var/lib/libvirt/images/ubuntu-vm.img,bus=virtio,size=4 \
 --graphics none \
 --location /home/ubuntu/ubuntu-14.04.4-server-amd64.iso \
 --extra-args='console=tty0 console=ttyS0,115200n8 serial' \
 --network bridge:virbr0
```

virt-install Parameters

```
sudo virt-install -n ubuntu-vm \
 --connect qemu:///system \
 --description "Ubuntu 14.04 VM" \
 --os-type=Linux \
 --ram=1024 \
 --vcpus=2 \
 --disk path=/var/lib/libvirt/images/ubuntu-vm.img,bus=virtio,size=4 \
 --graphics none \
 --location /home/ubuntu/ubuntu-14.04.4-server-amd64.iso \
 --extra-args='console=tty0 console=ttyS0,115200n8 serial' \
 --network bridge:virbr0
```

virt-install Parameters

```
sudo virt-install -n ubuntu-vm \
 --connect qemu:///system \
 --description "Ubuntu 14.04 VM" \
 --os-type=Linux \
 --ram=1024 \
 --vcpus=2 \
 --disk path=/var/lib/libvirt/images/ubuntu-vm.img,bus=virtio,size=4 \
 --graphics none \
 --location /home/ubuntu/ubuntu-14.04.4-server-amd64.iso \
 --extra-args='console=tty0 console=ttyS0,115200n8 serial' \
 --network bridge:virbr0
```

KVM Management Tools: vmbuilder

vmbuilder Parameters

```
sudo vmbuilder kvm ubuntu \
    --name new-ubuntu-vm \
    --suite trusty \
    --flavour virtual \
    --addpkg=linux-image-generic \
    --addpkg=unattended-upgrades \
    --addpkg openssh-server \
    --addpkg=acpid \
    --arch amd64 \
    --libvirt qemu:///system \
    --user ubuntu \
    --name mypassword \
    --hostname=test \
    --pass default
```

vmbuilder Parameters

```
sudo vmbuilder kvm ubuntu \
    --name new-ubuntu-vm \
    --suite trusty \
    --flavour virtual \
    --addpkg=linux-image-generic \
    --addpkg=unattended-upgrades \
    --addpkg openssh-server \
    --addpkg=acpid \
    --arch amd64 \
    --libvirt qemu:///system \
    --user ubuntu \
    --name mypassword \
    --hostname=test \
    --pass default
```

vmbuilder Parameters

```
sudo vmbuilder kvm ubuntu \
    --name new-ubuntu-vm \
    --suite trusty \
    --flavour virtual \
    --addpkg=linux-image-generic \
    --addpkg=unattended-upgrades \
    --addpkg openssh-server \
    --addpkg=acpid \
    --arch amd64 \
    --libvirt qemu:///system \
    --user ubuntu \
    --name mypassword \
    --hostname=test \
    --pass default
```

vmbuilder
.cfg file

```
[DEFAULT]
arch = i386
ip = 10.0.4.100
part = vmbuilder.partition
user = ubuntu
name = ubuntu
pass = default
tmpfs = -
firstboot = boot.sh
firstlogin = login.sh

[ubuntu]
mirror = http://ca.archive.ubuntu.com/ubuntu/
suite = trusty
flavour = virtual
addpkg = openssh-server, apache2, apache2-utils, apache2.2-common,libapache2-mod-php5, php5-cli, php5-gd, php5-ldap, php5-mysql, mysql-server, unattended-upgrades, acpid
```

vmbuilder
.cfg file

```
[DEFAULT]
arch = i386
ip = 10.0.4.100
part = vmbuilder.partition
user = ubuntu
name = ubuntu
pass = default
tmpfs = -
firstboot = boot.sh
firstlogin = login.sh

[ubuntu]
mirror = http://ca.archive.ubuntu.com/ubuntu/
suite = trusty
flavour = virtual
addpkg = openssh-server, apache2, apache2-utils, apache2.2-
common,libapache2-mod-php5, php5-cli, php5-gd, php5-ldap, php5-mysql,
mysql-server, unattended-upgrades, acpid
```

KVM Management Tools: KVM

KVM Parameters

```
sudo kvm -name "my-VM" -M pc -m 768 \
-smp 2 -boot d \
-drive file=/var/lib/libvirt/images/my-ub,if=virtio,index=0,media=disk,format=raw \
-drive file=/home/ubuntu/ubuntu-14.04.4-server-amd64.iso,index=1,media=cdrom \
-net nic,model=virtio,macaddr=52:54:00:05:11:11 \
-net bridge,vlan=0,br=virbr0 \
-vga none -balloon virtio
```

KVM Parameters

```
sudo kvm -name "my-VM" -M pc -m 768 \
-smp 2 -boot d \
-drive file=/var/lib/libvirt/images/my-ub,if=virtio,index=0,media=disk,format=raw \
-drive file=/home/ubuntu/ubuntu-14.04.4-server-amd64.iso,index=1,media=cdrom \
-net nic,model=virtio,macaddr=52:54:00:05:11:11 \
-net bridge,vlan=0,br=virbr0 \
-vga none -balloon virtio
```

KVM Parameters

```
sudo kvm -name "my-VM" -M pc -m 768 \
-smp 2 -boot d \
-drive file=/var/lib/libvirt/images/my-ub,if=virtio,index=0,media=disk,format=raw \
-drive file=/home/ubuntu/ubuntu-14.04.4-server-amd64.iso,index=1,media=cdrom \
-net nic,model=virtio,macaddr=52:54:00:05:11:11 \
-net bridge,vlan=0,br=virbr0 \
-vga none -balloon virtio
```

KVM Parameters

```
sudo kvm -name "my-VM" -M pc -m 768 \
-smp 2 -boot d \
-drive file=/var/lib/libvirt/images/my-ub,if=virtio,index=0,media=disk,format=raw \
-drive file=/home/ubuntu/ubuntu-14.04.4-server-amd64.iso,index=1,media=cdrom \
-net nic,model=virtio,macaddr=52:54:00:05:11:11 \
-net bridge,vlan=0,br=virbr0 \
-vga none -balloon virtio
```

KVM Parameters

```
sudo kvm -name "my-VM" -M pc -m 768 \
-smp 2 -boot d \
-drive file=/var/lib/libvirt/images/my-ub,if=virtio,index=0,media=disk,format=raw \
-drive file=/home/ubuntu/ubuntu-14.04.4-server-amd64.iso,index=1,media=cdrom \
-net nic,model=virtio,macaddr=52:54:00:05:11:11 \
-net bridge,vlan=0,br=virbr0 \
-vga none -balloon virtio
```

KVM Parameters

```
sudo kvm -name "my-VM" -M pc -m 768 \
-smp 2 -boot d \
-drive file=/var/lib/libvirt/images/my-ub,if=virtio,index=0,media=disk,format=raw \
-drive file=/home/ubuntu/ubuntu-14.04.4-server-amd64.iso,index=1,media=cdrom \
-net nic,model=virtio,macaddr=52:54:00:05:11:11 \
-net bridge,vlan=0,br=virbr0 \
-vga none -balloon virtio
```

KVM Parameters

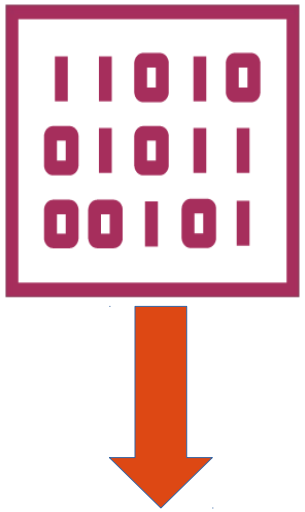
```
sudo kvm -name "my-VM" -M pc -m 768 \
-smp 2 -boot d \
-drive file=/var/lib/libvirt/images/my-ub,if=virtio,index=0,media=disk,format=raw \
-drive file=/home/ubuntu/ubuntu-14.04.4-server-amd64.iso,index=1,media=cdrom \
-net nic,model=virtio,macaddr=52:54:00:05:11:11 \
-net bridge,vlan=0,br=virbr0 \
-vga none -balloon virtio
```

KVM Parameters

```
sudo kvm -name "my-VM" -M pc -m 768 \
-smp 2 -boot d \
-drive file=/var/lib/libvirt/images/my-ub,if=virtio,index=0,media=disk,format=raw \
-drive file=/home/ubuntu/ubuntu-14.04.4-server-amd64.iso,index=1,media=cdrom \
-net nic,model=virtio,macaddr=52:54:00:05:11:11 \
-net bridge,vlan=0,br=virbr0 \
-vga none -balloon virtio
```

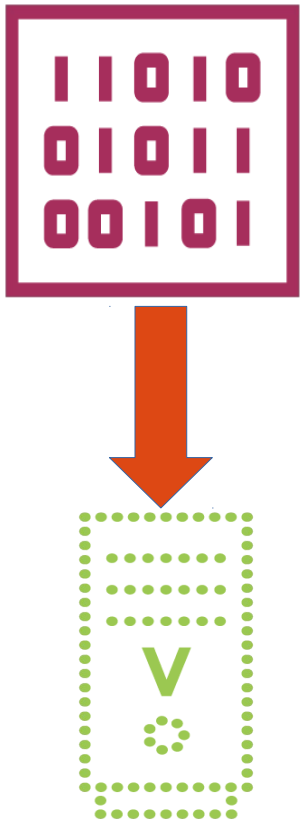
The KVM Process

STEP ONE:
Create or modify image with qemu-img



The KVM Process

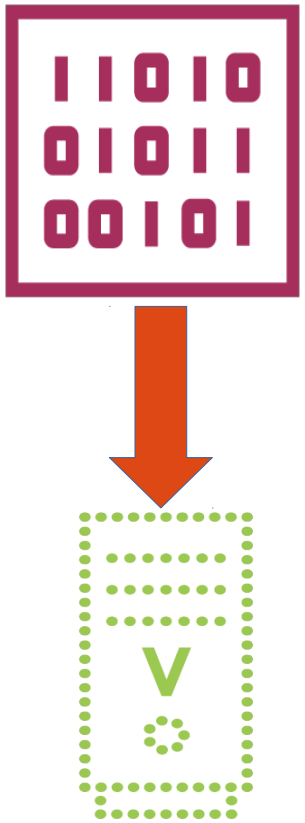
STEP ONE:
Create or modify image with qemu-img



STEP TWO:
Use image to start installation with qemu-kvm

The KVM Process

STEP ONE:
Create or modify image with qemu-img



STEP TWO:
Use image to start installation with qemu-kvm

NOTE: qemu-kvm = qemu-system-x86_64 and the kvm wrapper

Access Console from Shell

Access KVM console:	CTRL+ALT and SHIFT+2
Close console:	SHIFT+1
Launch from command line:	qemu-system-x86_64 -monitor stdio

Access Console from Shell

Access KVM console: CTRL+ALT and SHIFT+2

Close console: SHIFT+1

Launch from command line: qemu-system-x86_64 -monitor stdio

Access Console from Shell

Access KVM console:	CTRL+ALT and SHIFT+2
Close console:	SHIFT+1
Launch from command line:	qemu-system-x86_64 -monitor stdio

KVM: Networking and Storage

/etc/network/
interfaces
Bridged
Configuration

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet manual

auto br0
iface br0 inet dhcp
        bridge_ports eth0
        bridge_stp off
        bridge_fd 0
        bridge_maxwait 0
```

/etc/network/
interfaces
Bridged
Configuration

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet manual

auto br0
iface br0 inet dhcp
        bridge_ports eth0
        bridge_stp off
        bridge_fd 0
        bridge_maxwait 0
```

Add a Bridge to .XML Configuration

```
...
  <devices>
    <interface type='network'>
      <source network='default'/>
    </interface>
    ...
    <interface type='network'>
      <source network='default' portgroup='engineering'/>
      <target dev='vnet7'/>
      <mac address="00:11:22:33:44:55"/>
      <virtualport>
        <parameters instanceid='09b11c53-8b5c-4eeb-8f00-
d84eaa0aaa4f'/>
      </virtualport>


    </interface>
```

Add a Bridge to .XML Configuration

```
...
  <devices>
    <interface type='network'>
      <source network='default'/>
    </interface>
    ...
    <interface type='network'>
      <source network='br0' portgroup='engineering'/>
      <target dev='vnet7'/>
      <mac address="00:11:22:33:44:55"/>
      <virtualport>
        <parameters instanceid='09b11c53-8b5c-4eeb-8f00-
d84eaa0aaa4f'/>
      </virtualport>


    </interface>
```

Associate Bridge to Client

Use:

```
sudo qemu-system-x86_64 -net bridge,vlan=0,br=virbr0
```

Or:

```
sudo virt-install [...]--network bridge:virbr0
```

Sample Pool Entry from KVM .XML Configuration

```
<pool type="netfs">
        <name>virtimages</name>
        <source>
          <host name="nfs.example.com"/>
          <dir path="/home/datauser/current-files"/>
          <format type='nfs'/>
        </source>
        <target>
          <path>/var/current-files</path>
        </target>
      </pool>
```

Sample Pool Entry from KVM .XML Configuration

```
<pool type="netfs">
      <name>virtimages</name>
      <source>
        <host name="nfs.example.com"/>
        <dir path="/home/datauser/current-files"/>
        <format type='nfs'/>
      </source>
      <target>
        <path>/var/current-files</path>
      </target>
    </pool>
```

Sample Pool Entry from KVM .XML Configuration

```
<pool type="netfs">
      <name>virtimages</name>
      <source>
        <host name="nfs.example.com"/>
        <dir path="/home/datauser/current-files"/>
        <format type='nfs'/>
      </source>
      <target>
        <path>/var/current-files</path>
      </target>
    </pool>
```

Sample Pool Entry from KVM .XML Configuration

```
<pool type="netfs">
      <name>virtimages</name>
      <source>
        <host name="nfs.example.com"/>
        <dir path="/home/datauser/current-files"/>
        <format type='nfs'/>
      </source>
      <target>
        <path>/var/current-files</path>
      </target>
    </pool>
```

Sample Pool Entry from KVM .XML Configuration

```
<pool type="netfs">
      <name>virtimages</name>
      <source>
        <host name="nfs.example.com"/>
        <dir path="/home/datauser/current-files"/>
        <format type='nfs'/>
      </source>
      <target>
        <path>/var/current-files</path>
      </target>
    </pool>
```

Valid KVM/libvirt Pool Types

Netfs	Network File System
DIR	Directory
ISCSI	iSCSI server
Logical	logical volume storage pool

KVM Storage

Review

kvm-ok (or egrep -c '(vmx|svm)' /proc/cpuinfo)

modprobe kvm

modprobe kvm-intel

virsh list --all

virt-install -n ubuntu-vm ...

--extra-args='console=tty0 console=ttyS0,115200n8 serial'

virsh start ubuntu-vm

vmbuilder kvm ubuntu --name new-ubuntu-vm --suite trusty ...

vmbuilder kvm ubuntu -c my-guest-file.cfg

qemu-img create -f qcow2 /var/lib/libvirt/images/my-disk.img 6G

kvm -name "my-VM" -M pc -m 768 ...

qemu-system-x86_64 -monitor stdio

qemu-system-x86_64 -net bridge,vlan=0,br=virbr0

virsh pool-define ~/shared_files_disk.xml

Review

kvm-ok (or egrep -c '(vmx|svm)' /proc/cpuinfo)

modprobe kvm

modprobe kvm-intel

virsh list --all

virt-install -n ubuntu-vm ...

--extra-args='console=tty0 console=ttyS0,115200n8 serial'

virsh start ubuntu-vm

vmbuilder kvm ubuntu --name new-ubuntu-vm --suite trusty ...

vmbuilder kvm ubuntu -c my-guest-file.cfg

qemu-img create -f qcow2 /var/lib/libvirt/images/my-disk.img 6G

kvm -name "my-VM" -M pc -m 768 ...

qemu-system-x86_64 -monitor stdio

qemu-system-x86_64 -net bridge,vlan=0,br=virbr0

virsh pool-define ~/shared_files_disk.xml

Review

kvm-ok (or egrep -c '(vmx|svm)' /proc/cpuinfo)

modprobe kvm

modprobe kvm-intel

virsh list --all

virt-install -n ubuntu-vm ...

--extra-args='console=tty0 console=ttyS0,115200n8 serial'

virsh start ubuntu-vm

vmbuilder kvm ubuntu --name new-ubuntu-vm --suite trusty ...

vmbuilder kvm ubuntu -c my-guest-file.cfg

qemu-img create -f qcow2 /var/lib/libvirt/images/my-disk.img 6G

kvm -name "my-VM" -M pc -m 768 ...

qemu-system-x86_64 -monitor stdio

qemu-system-x86_64 -net bridge,vlan=0,br=virbr0

virsh pool-define ~/shared_files_disk.xml

Review

kvm-ok (or egrep -c '(vmx|svm)' /proc/cpuinfo)

modprobe kvm

modprobe kvm-intel

virsh list --all

virt-install -n ubuntu-vm ...

--extra-args='console=tty0 console=ttyS0,115200n8 serial'

virsh start ubuntu-vm

vmbuilder kvm ubuntu --name new-ubuntu-vm --suite trusty ...

vmbuilder kvm ubuntu -c my-guest-file.cfg

qemu-img create -f qcow2 /var/lib/libvirt/images/my-disk.img 6G

kvm -name "my-VM" -M pc -m 768 ...

qemu-system-x86_64 -monitor stdio

qemu-system-x86_64 -net bridge,vlan=0,br=virbr0

virsh pool-define ~/shared_files_disk.xml

Review

kvm-ok (or egrep -c '(vmx|svm)' /proc/cpuinfo)

modprobe kvm

modprobe kvm-intel

virsh list --all

virt-install -n ubuntu-vm ...

--extra-args='console=tty0 console=ttyS0,115200n8 serial'

virsh start ubuntu-vm

vmbuilder kvm ubuntu --name new-ubuntu-vm --suite trusty ...

vmbuilder kvm ubuntu -c my-guest-file.cfg

qemu-img create -f qcow2 /var/lib/libvirt/images/my-disk.img 6G

kvm -name "my-VM" -M pc -m 768 ...

qemu-system-x86_64 -monitor stdio

qemu-system-x86_64 -net bridge,vlan=0,br=virbr0

virsh pool-define ~/shared_files_disk.xml

Review

kvm-ok (or egrep -c '(vmx|svm)' /proc/cpuinfo)

modprobe kvm

modprobe kvm-intel

virsh list --all

virt-install -n ubuntu-vm ...

--extra-args='console=tty0 console=ttyS0,115200n8 serial'

virsh start ubuntu-vm

vmbuilder kvm ubuntu --name new-ubuntu-vm --suite trusty ...

vmbuilder kvm ubuntu -c my-guest-file.cfg

qemu-img create -f qcow2 /var/lib/libvirt/images/my-disk.img 6G

kvm -name "my-VM" -M pc -m 768 ...

qemu-system-x86_64 -monitor stdio

qemu-system-x86_64 -net bridge,vlan=0,br=virbr0

virsh pool-define ~/shared_files_disk.xml

Review

kvm-ok (or egrep -c '(vmx|svm)' /proc/cpuinfo)

modprobe kvm

modprobe kvm-intel

virsh list --all

virt-install -n ubuntu-vm ...

--extra-args='console=tty0 console=ttyS0,115200n8 serial'

virsh start ubuntu-vm

vmbuilder kvm ubuntu --name new-ubuntu-vm --suite trusty ...

vmbuilder kvm ubuntu -c my-guest-file.cfg

qemu-img create -f qcow2 /var/lib/libvirt/images/my-disk.img 6G

kvm -name "my-VM" -M pc -m 768 ...

qemu-system-x86_64 -monitor stdio

qemu-system-x86_64 -net bridge,vlan=0,br=virbr0

virsh pool-define ~/shared_files_disk.xml

Review

kvm-ok (or egrep -c '(vmx|svm)' /proc/cpuinfo)

modprobe kvm

modprobe kvm-intel

virsh list --all

virt-install -n ubuntu-vm ...

--extra-args='console=tty0 console=ttyS0,115200n8 serial'

virsh start ubuntu-vm

vmbuilder kvm ubuntu --name new-ubuntu-vm --suite trusty ...

vmbuilder kvm ubuntu -c my-guest-file.cfg

qemu-img create -f qcow2 /var/lib/libvirt/images/my-disk.img 6G

kvm -name "my-VM" -M pc -m 768 ...

qemu-system-x86_64 -monitor stdio

qemu-system-x86_64 -net bridge,vlan=0,br=virbr0

virsh pool-define ~/shared_files_disk.xml

Review

kvm-ok (or egrep -c '(vmx|svm)' /proc/cpuinfo)

modprobe kvm

modprobe kvm-intel

virsh list --all

virt-install -n ubuntu-vm ...

--extra-args='console=tty0 console=ttyS0,115200n8 serial'

virsh start ubuntu-vm

vmbuilder kvm ubuntu --name new-ubuntu-vm --suite trusty ...

vmbuilder kvm ubuntu -c my-guest-file.cfg

qemu-img create -f qcow2 /var/lib/libvirt/images/my-disk.img 6G

kvm -name "my-VM" -M pc -m 768 ...

qemu-system-x86_64 -monitor stdio

qemu-system-x86_64 -net bridge,vlan=0,br=virbr0

virsh pool-define ~/shared_files_disk.xml

Review

kvm-ok (or egrep -c '(vmx|svm)' /proc/cpuinfo)

modprobe kvm

modprobe kvm-intel

virsh list --all

virt-install -n ubuntu-vm ...

--extra-args='console=tty0 console=ttyS0,115200n8 serial'

virsh start ubuntu-vm

vmbuilder kvm ubuntu --name new-ubuntu-vm --suite trusty ...

vmbuilder kvm ubuntu -c my-guest-file.cfg

qemu-img create -f qcow2 /var/lib/libvirt/images/my-disk.img 6G

kvm -name "my-VM" -M pc -m 768 ...

qemu-system-x86_64 -monitor stdio

qemu-system-x86_64 -net bridge,vlan=0,br=virbr0

virsh pool-define ~/shared_files_disk.xml

Review

kvm-ok (or egrep -c '(vmx|svm)' /proc/cpuinfo)

modprobe kvm

modprobe kvm-intel

virsh list --all

virt-install -n ubuntu-vm ...

--extra-args='console=tty0 console=ttyS0,115200n8 serial'

virsh start ubuntu-vm

vmbuilder kvm ubuntu --name new-ubuntu-vm --suite trusty ...

vmbuilder kvm ubuntu -c my-guest-file.cfg

qemu-img create -f qcow2 /var/lib/libvirt/images/my-disk.img 6G

kvm -name "my-VM" -M pc -m 768 ...

qemu-system-x86_64 -monitor stdio

qemu-system-x86_64 -net bridge,vlan=0,br=virbr0

virsh pool-define ~/shared_files_disk.xml

Review

kvm-ok (or egrep -c '(vmx|svm)' /proc/cpuinfo)

modprobe kvm

modprobe kvm-intel

virsh list --all

virt-install -n ubuntu-vm ...

--extra-args='console=tty0 console=ttyS0,115200n8 serial'

virsh start ubuntu-vm

vmbuilder kvm ubuntu --name new-ubuntu-vm --suite trusty ...

vmbuilder kvm ubuntu -c my-guest-file.cfg

qemu-img create -f qcow2 /var/lib/libvirt/images/my-disk.img 6G

kvm -name "my-VM" -M pc -m 768 ...

qemu-system-x86_64 -monitor stdio

qemu-system-x86_64 -net bridge,vlan=0,br=virbr0

virsh pool-define ~/shared_files_disk.xml

Review

kvm-ok (or egrep -c '(vmx|svm)' /proc/cpuinfo)

modprobe kvm

modprobe kvm-intel

virsh list --all

virt-install -n ubuntu-vm ...

--extra-args='console=tty0 console=ttyS0,115200n8 serial'

virsh start ubuntu-vm

vmbuilder kvm ubuntu --name new-ubuntu-vm --suite trusty ...

vmbuilder kvm ubuntu -c my-guest-file.cfg

qemu-img create -f qcow2 /var/lib/libvirt/images/my-disk.img 6G

kvm -name "my-VM" -M pc -m 768 ...

qemu-system-x86_64 -monitor stdio

qemu-system-x86_64 -net bridge,vlan=0,br=virbr0

virsh pool-define ~/shared_files_disk.xml

Review

kvm-ok (or egrep -c '(vmx|svm)' /proc/cpuinfo)

modprobe kvm

modprobe kvm-intel

virsh list --all

virt-install -n ubuntu-vm ...

--extra-args='console=tty0 console=ttyS0,115200n8 serial'

virsh start ubuntu-vm

vmbuilder kvm ubuntu --name new-ubuntu-vm --suite trusty ...

vmbuilder kvm ubuntu -c my-guest-file.cfg

qemu-img create -f qcow2 /var/lib/libvirt/images/my-disk.img 6G

kvm -name "my-VM" -M pc -m 768 ...

qemu-system-x86_64 -monitor stdio

qemu-system-x86_64 -net bridge,vlan=0,br=virbr0

virsh pool-define ~/shared_files_disk.xml

Review

kvm-ok (or egrep -c '(vmx|svm)' /proc/cpuinfo)

modprobe kvm

modprobe kvm-intel

virsh list --all

virt-install -n ubuntu-vm ...

--extra-args='console=tty0 console=ttyS0,115200n8 serial'

virsh start ubuntu-vm

vmbuilder kvm ubuntu --name new-ubuntu-vm --suite trusty ...

vmbuilder kvm ubuntu -c my-guest-file.cfg

qemu-img create -f qcow2 /var/lib/libvirt/images/my-disk.img 6G

kvm -name "my-VM" -M pc -m 768 ...

qemu-system-x86_64 -monitor stdio

qemu-system-x86_64 -net bridge,vlan=0,br=virbr0

virsh pool-define ~/shared_files_disk.xml