Creating Xen Paravirtualized Guests

0.1 Creating Para virtualized VM using virt-install

We can create Xen para virtualized guests using '`virt-install`' command using following steps:

1. Create disk image file with

   ```
   dd if=/dev/zero of=centos.img bs=1024k seek=10240
   count=0
   ```

   This should be done only on ext3 partitions, it wont work on NTFS partitions mounted using fuse.

2. Host the installation tree of OS to be installed on some NFS, HTTP or FTP tree

3. Create VM using

   ```
   virt-install --name=centos5 --ram=512 --paravirt
   --location='http://10.3.3.132/netinst'
   --file=centos.img --vnc --prompt
   ```

0.2 Managing VM using virsh commands

Similar to '`xm reboot`, '`xm create`', '`xm destroy`' etc. commands there are '`virsh reboot`', '`virsh create`', '`virsh destory`' etc. commands. '`virsh`' commands use libvirt which talks to xend and performs the same actions as '`xm`' does. But '`virsh`' has notable advantages over '`xm`' commands:

1. '`virsh`' uses libvirt which will support not only xend but other virtualization platforms also. So in future if we want to use some other virtualization platform like '`kvm`' or mixture of '`kvm`' and '`xen`' then '`virsh`' commands would still work, but '`xm`' commands will only work if we use xen.

2. '`virsh`' has more options then '`xm`' as it allows to manage virtual networks, pools and storage volumes. For example, we can specify domains which should be automatically restarted if system reboots and whenever host reboots those virtual machines are automatically restarted without user intervention.

   Internally this is accomplized by create a shortcut to VM description file in folder '`/etc/xen/auto`'. We can use command '`virsh autostart <domain_name>`' so that the shortcut gets created.

3. '`virsh`' uses more popular XML format for configuration which is more standardized then text file format used by '`xm`'. We can use utility '`virt-xml-validate`' to validate XML configuration files against standard schema definitions of domains, networks etc. to ensure that the given XML file conforms to defined standards.

4. If we create VM using '`virt-install`' then VMs configuration file gets created in '`/etc/xen`'. We can use '`virsh`' to start this domain by specifying its name, which looks for the files in '`/etc/xen`' and starts domain. This ensures all configuration files are in standard location and not spread across folders.

5. Since the domain configuration file is present in '`/etc/xen`' we can use '`virt-manager`' a GUI tool to manage VM (add/remove hardware, suspend, restore, monitor performance, etc.)

6. We can use '`virt-viewer`' to connect to vnc server of domain by supplying domain name and we need not use combination of '`ps`' and '`vncviewer`' commands to connect to correct domain.

   Also '`virt-viewer`' is more user friendly then normal '`vncviewer`' display. Try sending keys '`Ctrl + Alt + F2`' to guests first using '`vncviewer`' and later using '`virt-viewer`' to understand the difference.

7. If we try to create new domain with same name as already existing domain or try to re-use hard-disk image in new domain then '`virt-install`', '`virsh`' etc. give proper warning and prompt us to confirm whether we are ok with two virtual machines using the same hard disk image or not. This prevents us from accidently over writing an existing VM if we try to create new VM by copying its configuration file.

8. Unline '`xm`' commands which signal to xend and return before the operation is completed, '`virsh`' commands usually return only after operation is completed and hence are more convinient to use when writing scripts / programs when it is important that previous step finishes before next step is started.

9. '`virsh`' also supports a interactive terminal which can be started by using '`virsh`' command without any arguments. This terminal can be used to operate on virtual machines more efficiently so that we do not have to start each command with '`virsh `'.

0.3 Para-virtualized guests device information

0.3.1 Disk devices

For HVM guests we use device names like hda, hdb, etc. as target devices to be shown to HVM guest. But for paravirtualized guests we use names, xvda, xvdb, etc. to indicate that the disk images are paravirtualized and that guest OS can directly communicate with host OS kernel for I/O without any emulation. Also for HVM guests we use driver '`file:`' and for paravirtualized guests we use '`tap:aio:`' to indicate paravirtualized image.

For example if HVM configuration looks like

```
disk = [ "file:/home/saurabh/Desktop/vm/centos.img,hda,w"
]
```

then paravirtualized guest configuration would look like

```
disk = [ "tap:aio:/home/saurabh/Desktop/vm/centos.img,xvda,w"
]
```

assuming both guests want to use the same file as harddisk image.

0.3.2 Network devices

For HVM guests when we define virtual network interfaces we specify interface type as ioemu, but for paravirtualized guests this is not required. Hence we can just specify bridge=xenbr0 and optionally MAC address to create a network interface. This interface uses Xen paravirtualization drivers and not emulation and hence performs better then emulated interface.

For verification we can use Menu option "System → Administration → Network" to start GUI utility called '`system-config-network`'. In hardware tab of '`system-config-network`' we should see the device as 'Xen Virtual Ethernet' to verify that Guest has indeed recognized paravirtualized network device.

0.3.3 Boot loader

For HVM guests we specify kernel, builder and device_model typically as:

```
kernel = "/usr/lib/xen/boot/hvmloader"
builder='hvm'
device_model = '/usr/lib64/xen/bin/qemu-dm'
```

but for paravirtualized guests instead of these we specify bootloader as:

```
bootloader = "/usr/bin/pygrub"
```

0.4 Extra-tools

0.4.1 virt-convert

We can use '`virt-convert`' to convert VMWare (popular software for virtualization) images to 'virt-image' (Images supported by virt, that is raw format). Read '`man virt-convert`' for information on how to convert images from VMWare to virt-image format.

0.4.2 virt-clone

If we want to clone VMs then we just coping configuration file and changing disk image name is not enough. We also need to change things like MAC address, UUID, VM name, etc. which must be unique.

Hence utility named '`virt-clone`' is created which allows easy cloning of virtual machines. It supports '`--prompt`' like virt-install so that the utility will ask for information if it is not able to decide on its own or if parameters passed are ambiguous. Read '`man virt-clone`' for more information on '`virt-clone`' utility.

0.5 Lab tasks

1. Create a xen paravirtualized guest using virt-install
2. Use '`virsh help`' to see list of commands / operations supported by '`virsh`'. Try the obvious ones like start, destroy, reboot, save, restore, suspend(pause), resume(unpause) etc.
3. Use '`virsh`' to get XML configuration of domain created in this lab. (Use command '`virsh dumpxml`'). Save the configuration file for later use.
4. Use '`virsh`' to undefine domain created in part 1.
5. Modify XML file obtained by using '`virsh`' to add one more hard-disk image for paravirtualized host.
6. Create paravirtualized host using '`virsh`' and this modified XML file.
7. Start '`virsh`' in interactive mode and get XML dump of this modified guest.
8. Connect to guest using '`virt-viewer`' and login as root. Verify that the device '`/dev/xvdb`' actually exists.

9. From running domain detach '`/dev/xvdb`' using '`virsh detach-disk`' command. Example, '`virsh detach-disk centos5 xvdb`'.

10. Now dump XML file using '`virsh dumpxml`' which should show the configuration of running host (ie without xvdb) and not the configuration with which the host was booted.

11. Create XML file containing description of second disk similar to

```
<disk type='file' device='disk'>
<driver name='tap' type='aio'/>
<source file='/home/saurabh/Desktop/vm/centos2.img'/>
<target dev='xvdb' bus='xen'/>
</disk>
```

12. Use '`virsh`' to attach device '`/dev/xvdb`' in running domain using syntax '`virsh attach-device <domain_name> <xml_file_name>`'.

13. Again verify in running domain whether device has re-appeared. Again detach device '`/dev/xvdb`' and re-attach it in running domain so that you are comfortable with procedure of hotplugging and removing devices from running domains.

14. Start '`virt-manager`'. Use menu "Edit → Host details". Go to tab "Virtual Networks" and see values of various parameters.

15. Now go to command line and use command '`virsh net-dumpxml default`' to see configuration of default network in XML format.

16. Suppose the default network has configuration

```
<network>
    <name>default</name>
<uuid>d99113fb-948a-4ec0-b763-8de5a3c8023b</uuid>
<forward mode='nat'/>
<bridge name='virbr0' stp='on' forwardDelay='0' />
<ip address='192.168.122.1' netmask='255.255.255.0'>
    <dhcp>
        <range start='192.168.122.2' end='192.168.122.254' />
    </dhcp>
</ip>
</network>
```

    Modify the network configuration to get something like

```
<network>
    <name>test</name>
<forward mode='nat'/>
<bridge name='test' stp='on' forwardDelay='0' />
<ip address='192.168.100.1' netmask='255.255.255.0'>
    <dhcp>
        <range start='192.168.100.2' end='192.168.100.254' />
    </dhcp>
</ip>
</network>
```

Ensure that network ranges of default and test network do not overlap.

17. Add this virtual network using '`virsh net-define test.xml`' where '`test.xml`' is the name of file in which the new 'test' network has been defined.

18. Use '`virsh net-list --all`' to verify that network got defined.

19. Use '`virsh net-start test`' to start test virtual network.

20. Go to "Virtual networks" tab of "host details" dialog provided by '`virt-manager`'. Verify that network test is visible here and has similar properties as we defined in the XML file.

21. Use '`virsh net-destroy test`' and then '`virsh net-undfine test`' to destroy and undefine the test network.

22. List advantages of Xen over VMWare and vice-versa.

0.5.1 Optional

Figure out how we can use Xen hard-disk images and mount them on host OS for modifying copying files when guest is powered off.

0.6 Reference

You can refer to Virtualization Guide created by RedHat to learn more about virt-manger, virsh, network scripts that are used for virtualization etc. To access the Virtualization Guide one can use Menu option "System → Documentation → Virtualization Guide" option available in Cent OS distributions.