Virtualization

From AudioWiki

Contents

KVM

- Installation (qemu-kvm and virtual manager)

```
aptitude install qemu-kvm virt-manager
```

libvirt will be installed as a dependency of the above packages. In order to use kvm as a regular user libvirt group will solve the problem.

KVM is handled from virsh in text-mode and virt-manager on graphics mode. Both use libvirt-bin package so as mentioned above $user needs to be member of libvirt group.

ethernet bridging

- We are using an ethernet bridging on HOST machine (S8) so VMs can access directly to one of the S8 ethernet interfaces. In order to perform this task the following configuration has been done:

HOST MACHINE:

- Installation bridge-utils:

```
aptitude install bridge-utils
```

- Edit */etc/network/interfaces* adding an entry called br0:

```
auto br0
iface br0 inet static
       address 192.168.122.1
       netmask 255.255.255.0
       #network 192.168.122.0
       broadcast 192.168.122.255
       #up /sbin/route add -net 192.168.122.0/24 dev br0
       #down /sbin/route del -net 192.168.122.0/24 dev br0
       #gateway 192.168.2.1
       bridge_ports eth1
       #bridge_stp on
       bridge_maxwait 0
       bridge_ageing 0
       bridge_maxage 0
       bridge_fd 0
       auto bond0
iface bond0 inet manual
       slaves eth2 eth3
       bond-mode 1
       bond-miimon 100
       bond-downdelay 200
       bond-updelay 200
       #mtu 9000
auto br1
iface br1 inet static
       address 192.168.123.1
       netmask 255.255.255.0
       #network 192.168.123.0
       broadcast 192.168.123.255
       #up /sbin/route add -net 192.168.123.0/24 gw 192.168.123.1 dev br1
       #up /sbin/route add -net 192.168.123.0/24 dev br1
       #down /sbin/route del -net 192.168.123.0/24 gw 192.168.123.1 dev br1
       #down /sbin/route del -net 192.168.123.0/24 dev br1
       #gateway 192.168.123.1
       bridge_ports bond0
       bridge_stp off
       bridge_maxwait 0
       bridge_ageing 0
       bridge_maxage 0
       bridge_fd 0
```

- NOTES ON BONDING:

st0 and st1 machines all have two ethernet cards performing alb bonding as described above. For this to work there is also another config step:

```
cat /etc/modprobe.d/bonding.conf
alias bond0 bonding
options bonding mode=6 miimon=100 downdelay=200 updelay=200
```

ARP problem

The bonding(balance-alb)+bridging has come out with an arp problem that resulted on a packet loss. In order to prevent this the solution I found is to use same mac on all slave interfaces:

Making this change permanent:

- Edit */etc/rc.local*

```
For s8
/sbin/ifconfig eth3 hw ether MAC_ADDRESS_FROM_eth2
For st0/st1
/sbin/ifconfig eth2 hw ether MAC_ADDRESS_FROM_eth0
```

- Edit /etc/sysctl.conf and add:

```
net.bridge.bridge-nf-call-ip6tables = 0
net.bridge.bridge-nf-call-iptables = 0
net.bridge.bridge-nf-call-arptables = 0
```

- Now you can ifup br0: *ifup br0*

- Allow Host machine to forward tcp packets:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

Iptables setup:

You just need to masquerade packets from our private network there's a file on root account for restoring iptables rulesets, here sentences anyway:

```
iptables -t nat -A POSTROUTING -s 192.168.122.0/24 ! -d 192.168.122.0/24 -j MASQUERADE
```

- Redirect outside 443 and 80 port to our webserver:

```
iptables -t nat -A PREROUTING -d 147.83.50.223/32 -i eth0 -p tcp -m tcp --dport 80 -j DNAT --to-destination
iptables -t nat -A PREROUTING -d 147.83.50.223/32 -i eth0 -p tcp -m tcp --dport 443 -j DNAT --to-destination
```

- Redirect port 222 to physical machine (for maintenance tasks):

```
iptables -t nat -A PREROUTING -d 147.83.50.223/32 -i eth0 -p tcp -m tcp --dport 222 -j DNAT --to-destination
```

VMs

MP0: Multipurpose0.

```
 VM seed in order to clone for creating new purpose-specific machines.
```

- - Services: sshd (SSH daemon)

```
Disk scheme:
/ = 4GB
swap = 224MB
RAM = 512 MB
```

hq

Login machine, shares user's /home along our network

- - Services: sshd,NFS,Slurmctrl, ganglia (gmond)

```
Disk scheme:
/ = 10GB
/boot = 200MB
/tmp = 1 GB
/home = 250GB
```

- Also planned to run git server
- SSH keys are copied from S8 so ssh will not complaint about authorization.

- You have to be care with free space on disk in this machine. If disk hasn't free space on it, the machine will get down and nobody can get into the smartveu.upc.edu. In special /var/log folder is the main problem, this folder increases its size and periodically you must check still there is free space on disk. This situation is common with th realm (LDAP server) machine.

web0

Web server

- - Services: sshd, apache2 running audiowiki (mediawiki), ganglia (gmetad,gmond,ganglia-webfrontend)

```
Disk scheme: same as mp0
```

realm

LDAP server

- - Services: sshd, slapd, ganglia (gmond)

```
Disk scheme: same as mp0 or web0
```

- You have to be care with free space on disk in this machine. If disk hasn't free space on it, the machine will get down and nobody can get into the smartveu.upc.edu server because LDAP server will be able to authentificate anybody. In special /var/logs folder is the main problem, this folder increases its size and periodically you must check still there is free space on disk. The same situation than hq machine.

Libvirt Networking

Libvirt networking configuration files are under /etc/libvirt/qemu/networks

This is only needed for a NAT configuration which is discouraged for long-term deployments as ours.

Virsh&Virt-clone

- virsh: enters virsh shell
- virsh --connect qemu:///system
- virsh list --all (lists all VM's)

- Clonning vm:

virt-clone --prompt (prompts for all questions about VM clonning process) virt-clone --connect=qemu:///system -o orig_name -n new_name -f path

Retrieved from "https://smartveu.upc.edu/wiki/index.php?title=Virtualization&oldid=940"

- This page was last modified on 5 December 2013, at 20:12.
- This page has been accessed 99 times.