

## STEPS FOR INFRASTRUCTURE

(1)

- ① Create mainBR.xml (mainBR bridge in nat mode, dhcp range of 192.168.140.2 - 192.168.140.254; mainBR dhcp server ip = 192.168.140.1)
- ② Create pns1 with veth pair p1-mbr. Attach p1 to pns1 and mbr to mainBR. DO NOT ADD ANY MASQUERADE rule.
- ③ Run dnclient on p1 inside pns1. It should get an IP as 192.168.140.x/24 and also routes must get populated.
- ④ Create c1s1 with veth pair n1-p2. Attach p2 to pns1 and n1 to c1s1. Assign IP to p2 as say 192.168.150.1/24 and to n1 as 192.168.150.2/24. DO NOT ADD MASQUERADE rule.
- ⑤ Inside c1s1:
  - ① Create a bridge vbr through brctl
  - ② Create a vxlan device with full command -  

{

ip link add vxlan0 type vxlan id 42 local <ip of n1 in current hypervisor> remote <ip of n1 in remote hypervisor> dstport 4789
  - ③ Attach vxlan0 to vbr
  - ④ Create a veth pair dns-dbr.
  - ⑤ Attach dbr to vbr
  - ⑥ Assign IP to dns - IP that customer wants say for eg:- 1.1.1.1/24
  - ⑦ Run dnsmasq command on dns interface -  

{

dnsmasq --interface dns --dhcp-range=1.1.1.2, 1.1.1.100
  - ⑧ Add default route -  

{

add default via 192.168.150.1
- ⑥ In hypervisor add the following routes -  

{

ip route add 192.168.150.0/24 via <IP of p1 of pns1>

ip route add 192.168.160.0/24 via 99.99.99.2

<Subnet of n1 of c1s1 on remote hypervisor>

⑦ For the other hypervisors, repeat steps ① to ⑥ with full changes-

- p2 ip - 192.168.160.1/24
- n1 ip - 192.168.160.2/24
- dns ip - 1.1.1.101/24 ; dhcp-range = 1.1.1.102, 1.1.1.254
- In c1s1 add default route via 192.168.160.1
- In hypervisor -
  - { ip route add 192.168.150.0/24 via 99.99.99.1
  - { ip route add 192.168.160.0/24 via <IP of p1 in pns1> }

## STEPS FOR VM CREATION & PINN

- ① Create a veth pair b1-v1. Attach b1 to vbr in c1s1.
- ② Create a VM with --network none
- ③ Attach v1 to VM using -  
 { virsh attach-interface --domain vm direct v1 --live }  
   --persistent }
- ④ Obtain mac address of the VM through domiflist and IP through virt-cat commands.
- ⑤ Repeat all the above steps on the other hypervisor
- ⑥ Ping from one VM to the other; ping must work. Remember  
 if IP of VM in hypervisor 1 - 1.1.1.22/24 and.  
 IP of VM in hypervisor 2 - 1.1.1.116/24 then, ping  
 should be - ping 1.1.1.22 / ping 1.1.1.116.

## MIGRATION STEPS

- ### MIGRATION STEPS
- ① All the VMs are directly attached to vbr through the veth pair  $b \langle x \rangle - v \langle x \rangle$ . Detail this  $v \langle x \rangle$  for the VM to be migrated using `--domain VM --type direct --config` of `virsh detach-interface`.
  - ② Shutdown the VM to be migrated.
  - ③ Copy the VM's `VM.img` and `VM.xml` files to the remote hypervisor in `/var/lib/libvirt/images` & `/etc/libvirt/qemu` respectively.



- ④ Define the VM in the remote hypervisor (virsh define).
- ⑤ Start VM (virsh start).
- ⑥ Create veth pair  $b<x> - v(x)$  { same as the local hypervisor }  
Attach  $b<x>$  to vbr in C1S1.
- ⑦ Attach  $v(x)$  to VM but this time include the --mac option -  
virsh attach-interface --domain VM direct  $v<x>$  --live  
--persistent --mac <mac of the VM in local hypervisor }  
from where we migrated the VM from >
- ⑧ The VM must be having the same mac & IP. Ping must work.
- ⑨ Destroy the shutdown VM in the local hypervisor.

THINGS TO TAKE CARE

- ① Mac resolution
- ② Copied VM.img & VM.xml might conflict with existing ones.  
so handle such situations.
- ③ Selection of  $v<x>$  &  $b<x>$  name?
- ④