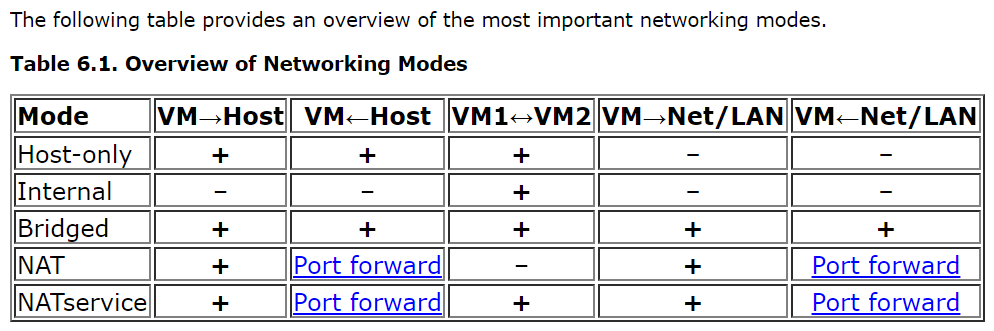
* All the virtual machines on physical machines are handled by libvirt. Whether its kvm, xen, vmware esx what ever.
* To give libvirt a command we need CLI or GUI like virt-manager.
* By default VMs managed by libvirt are hidden from internet/LAN of host. It can only communicate to other VMs and host.
* If you have server machine at remote location and you want to manage your VM from your home then you need to bridge the VMs on host machine so that they can be accessed from internet.
* Virt-manager or virsh can setup the bridge network so that all the VMs managed by libvirt on physical machines use the same bridge and IPs etc. range is provided and VMs are assigned IPs from dhcp range mentioned in virt-manager.



* What is port forwarding see this <https://www.youtube.com/watch?v=-K6jMYBfuIY>
* In this assignment we have to create a custom virtual manager which starts and stop VMs dynamically based on conditions i.e. load balancer.
* Virt-manager is already built and has GUI so we can also build our own app with GUI and give features like create VMs, add bridge, stop VMs etc. actually virt manager is built using Libvert so we can do same.
* Libvert separates the VMs based on Hypervisors like for KVM/QEMU, Xen, VMWare ESX etc. we have separate configurations in virt-manager and we need to select the hypervisor first then do configuration.
* We create vcpu for VM using virt-manager but these vcpu would not necessarily hog the physical cpu of host machine.
* By default virt-manager uses virtio see: <https://www.redhat.com/en/blog/introduction-virtio-networking-and-vhost-net>
* Virtual NIC is used and data plane(data packets) directly go from host to guest QEMU process is not involved. Otherwise vmexit is needed if we transfer host -> QEMU -> guest but control plane is through QEMU process. Virtio used here is developed by red hat.
* By default libvirt uses DHCP server to assign IP address to guest VMs and they are static you can see the DHCP server range from where it assigns. Also if there is any problem you can specify guest VM MAC(which is fixed) and static IP in the DHCP server configuration so that it will assigns same IP specified MAC. NOTE: this is libvirt DHCP server hence it used static IP assignment to guest VMs but if you take your VMs on bridge (network mode bridge not NAT(default)) then public DHCP server will assign IP that will depends. See this: <https://www.cyberciti.biz/faq/linux-kvm-libvirt-dnsmasq-dhcp-static-ip-address-configuration-for-guest-os/>
* If you want to access your guest VMs from internet you just need to do port forwarding from your router, just open jiodongle administration and do port forwarding.