**VIRTUALISATION**

This document explains complete procedure of creating a bridge networking and tuning the host machine for best performance. All the steps have been successfully performed on Fedora 17 (Beefy Miracle).

**PREREQUISITE**

1. Your processor should support VT-x(or Virtualization) technology and it should be enabled in BIOS.
2. Installing virtualization packeges:-

When installing Fedora, the virtualization packages can be installed by selecting **Virtualization** in the Base Group in the installer. For existing Fedora installations, QEMU, KVM, and other virtualization tools can be installed by running the following command which installs the virtualization group:

**#yum install @virtualization**

This will install below Mandatory, Default and Optional packages.

$ yum groupinfo @virtualization

Group: Virtualisation

Group-Id: virtualization

Description: These packages provide a virtualisation environment.

Mandatory Packages:

=virt-install

Default Packages:

=libvirt-daemon-config-network

=libvirt-daemon-kvm

=qemu-kvm

=virt-manager

=virt-viewer

Optional Packages:

guestfs-browser

libguestfs-tools

python-libguestfs

virt-top

**#systemctl start libvirtd**

1. Verify that the kvm kernel modules were properly loaded:

**# lsmod | grep kvm**

kvm\_amd 55563 0

kvm 419458 1 kvm\_amd

**Bridge Creation:-**

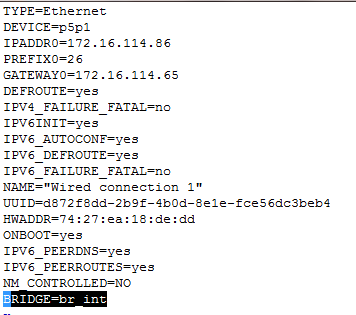
1. Install bridge utility by using the below command :-

**#Yum install brctl bridge-utils**

1. Verify by using

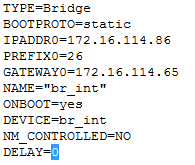
**# brctl show**

1. Changes in network scripts
   1. **# cd /etc/sysconfig/network-scripts/**
   2. **# vim ifcfg-Wired\_connection\_1**

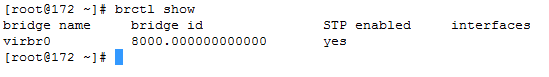
****

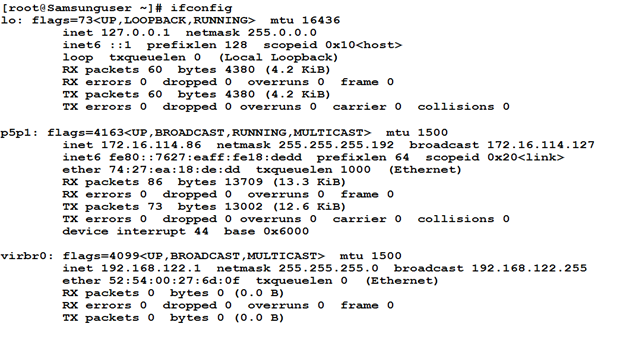
Add **BRIDGE= br\_int** in the script

* 1. **Create another script in the same folder with name “ifcfg-br\_int”.content of the script is as follows :-**

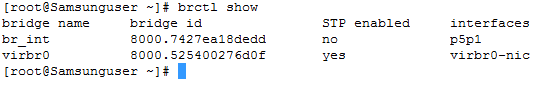
****

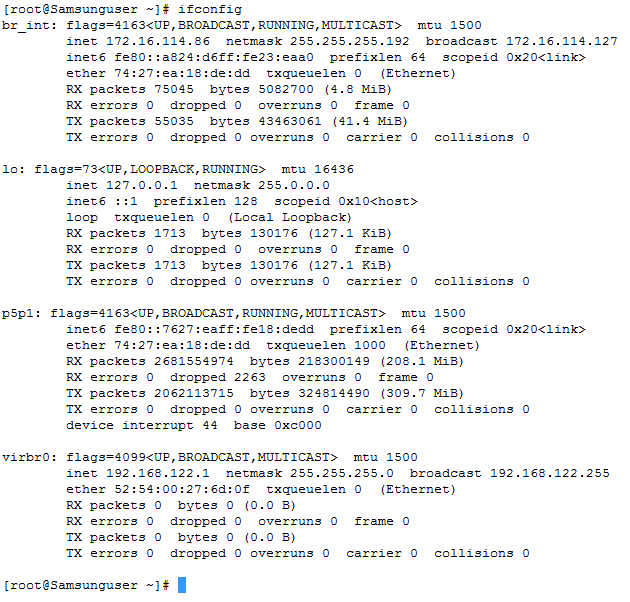
**Before making changes in the script**





**After making changes in the script**





1. Create a ifup script

**# vim /etc/ovs-ifup**

Add following lines and save

#!/bin/sh

switch='br\_int'

brctl addif ${switch} $1

/sbin/ifconfig $1 0.0.0.0 up

**Virtual Machine creation :-**

1. Create image of the Virtual machine :-

**#qemu-img create -f qcow2 v1disk.img 32G**

1. Follow the installation steps after using using the below mentioned command.

**# qemu-kvm -m 512 -name Guest1 -net nic,macaddr=52:54:00:19:A9:22 -net tap,ifname=tap0,script=/etc/ovs-ifup,downscript=/etc/ovs-ifdown -cdrom Fedora-17-i386-DVD.iso -hda v1disk.img**

**Spawning the virtual machine :-**

1. **Using qemu-kvm :-**Use the below mentioned command to spawn the virtual machine using the below mentioned command

**#taskset -c 0,1,2,3 qemu-kvm -m 1600 -name VM4 -nographic -cpu host -net nic,model=virtio,macaddr=52:54:00:19:A9:22 -net tap,ifname=tap5,script=/etc/ovs-ifup -hda /root/shivam/newVM2.img**

1. **Using Virsh :-**Use the xml file attached below to spawn the VM.
2. Create the xml file in **/etc/libvirt/qemu**

Sample xml file :- [VM\_96.xml](file:///C:\Users\gur43505\AppData\Local\Microsoft\Windows\INetCache\Content.Outlook\E5J8PVS6\VM_96.xml)

1. Use the below mentioned command to spawn the VM.

**#virsh create <filename>**

1. Use the below mentioned command to destroy the VM.

**#virsh destroy <domain name>**

Domain name is present in the xml file.

**PERFORMANCE TUNING**

1. For better performance use updated version of qemu-kvm and kernel
2. In BIOS disable IEST (Intel Enhanced Speed Technology)
3. In BIOS disable Enhanced Power Saving Mode.
4. For better networking performance use vhost\_net

In qemu-kvm command add this

**model=virtio vhost=on**

(note: by default vhost is on when spawning VM through virsh)