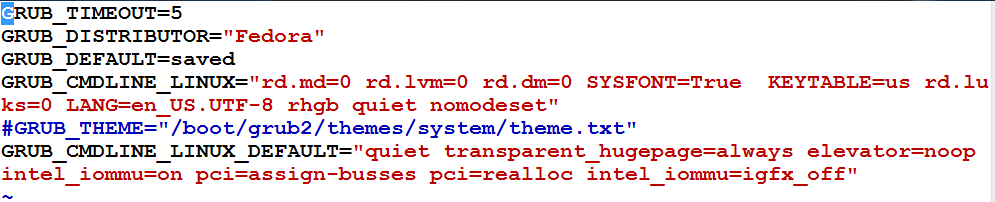
**STEPS PERFORMED FOR PCI PASSTHROUGH**

**Enable IOMMU support**

$ sudo vi /etc/default/grub

1. Add intel\_iommu=on pci=assign-busses pci=realloc intel\_iommu=igfx\_off in GRUB\_CMDLINE\_LINUX\_DEFAULT parameter.
2. Update grub with following command

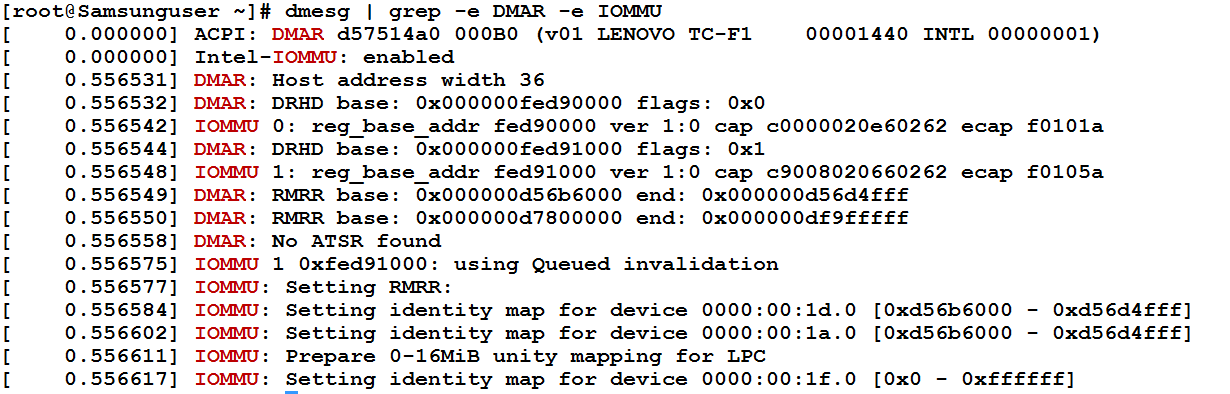
**grub2-mkconfig -o /boot/grub2/grub.cfg**



1. Reboot the system
2. Verify with following command.

**dmesg | grep -e DMAR -e IOMMU**

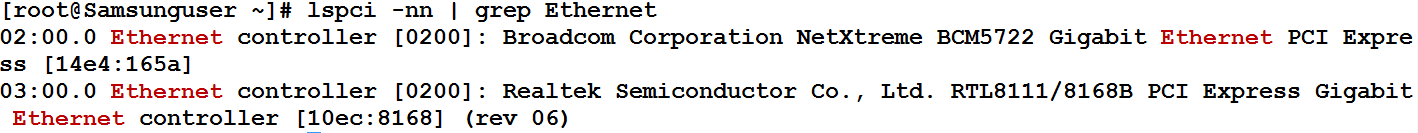
As it can be seen in the snapshot **Intel-IOMMU: enabled**



**Identify the Device you want to attach**

1. Use the following command to identify

**# lspci –nn | grep Ethernet**

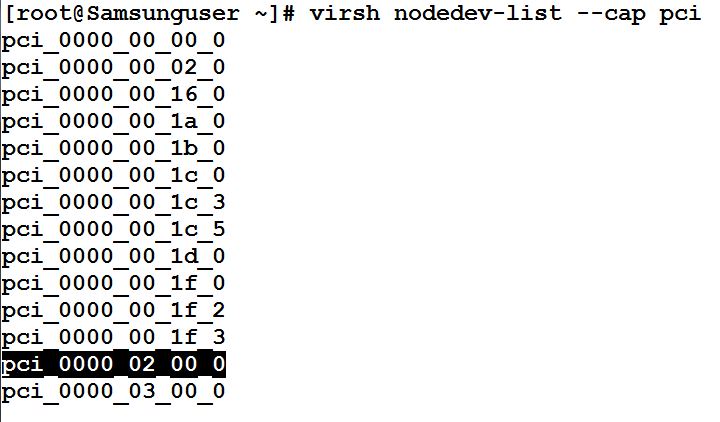


(Here we will be deattaching **02:00.0 Ethernet controller [0200]: Broadcom Corporation NetXtreme BCM5722 Gigabit Ethernet PCI Express [14e4:165a]**)

1. To find the full identifier used by virsh in order to assign PCI device use the following command.

**# virsh nodedev-list --cap pci**

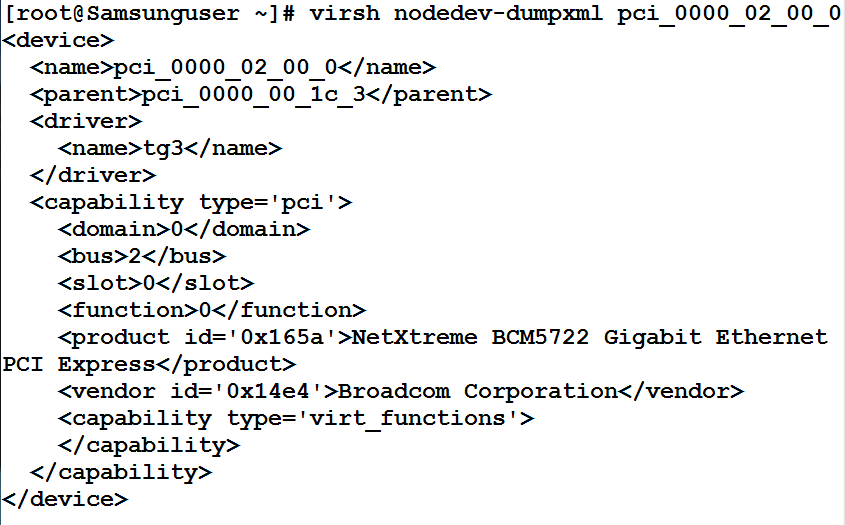
(We used the device pci\_0000\_02\_00\_0)



1. To identify the domain , bus, slot and function use the following command

**# virsh nodedev-dumpxml pci\_0000\_02\_00\_0**

(This info will be used to configure guest xml file)



(Domain , bus, slot and function info will be used to configure guest xml.)

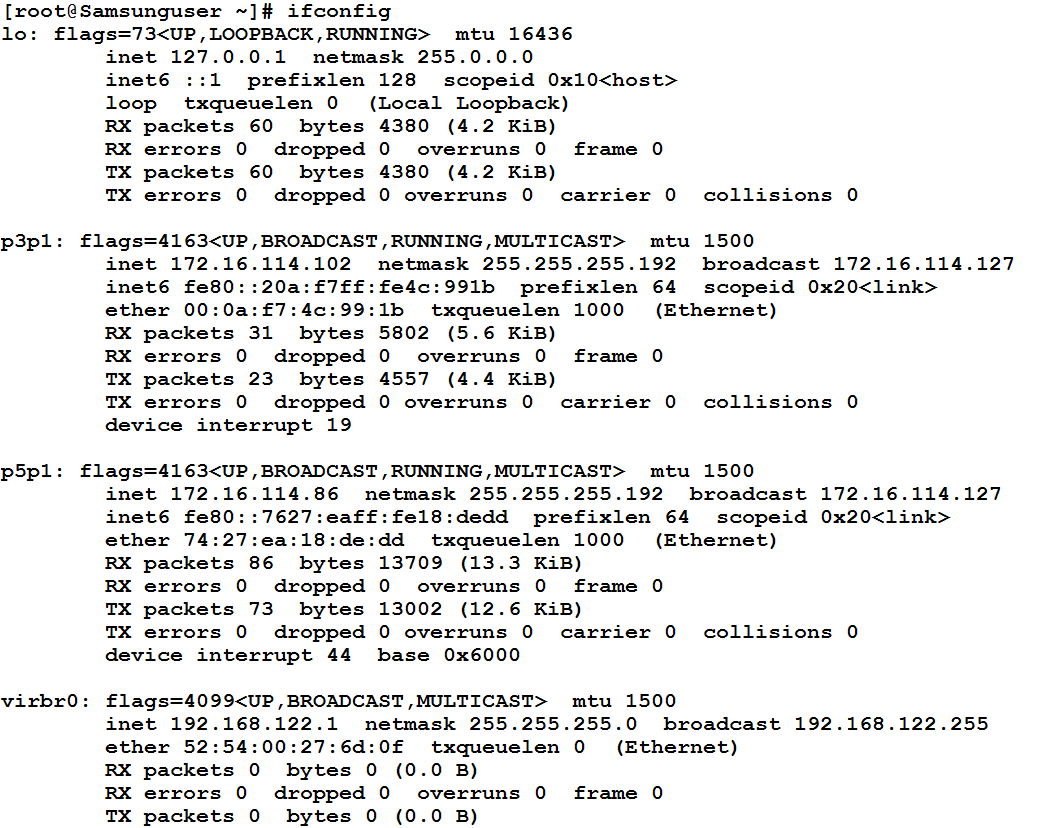
**De-attach the identified device from host.**

1. Use the following commands to de-attach the identified device( in our case it is pci\_0000\_02\_00\_0)

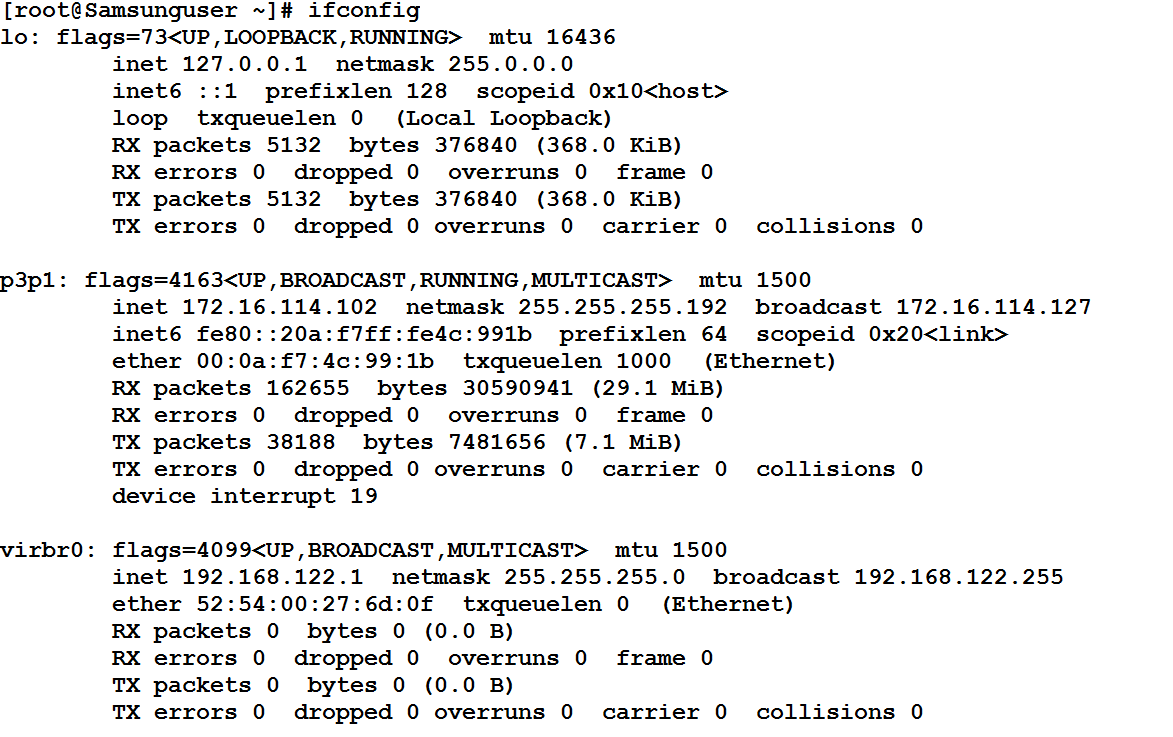
**# modprobe pci\_stub**

**# echo "14e4 165a" > /sys/bus/pci/drivers/pci-stub/new\_id   
# echo 0000:20:00.0 > /sys/bus/pci/devices/0000\:20\:00.0/driver/unbind  
# echo 0000:20:00.0 > /sys/bus/pci/drivers/pci-stub/bind**

Before Deattaching



As it can be seen our device p5p1 has been deattached.



**Configuring KVM to allow unsafe interrupts**

1. Run the following command on host machine

**echo 1 > /sys/module/kvm/parameters/allow\_unsafe\_assigned\_interrupts**

**Set an SELinux boolean to allow the management of the PCI device from the virtual machine**

1. Run the following command on host machine

**setsebool -P virt\_use\_sysfs 1**

**Methods to Spawn VM (use any one of the following.)**

1. **Through virsh**

**Changes in VM’s xml file**.

1. Add these lines to VM’s XML file (information is used from above)

**<hostdev mode='subsystem' type='pci' managed='yes'>**

**<source>**

**<address domain='0x0' bus='0x03' slot='0x0' function='0x0'/>**

**</source>**

**<rom bar='off'/>**

**</hostdev>**

Sample xml file of guest. [pci\_VM\_101.xml](file:///C:\Users\gur43505\AppData\Local\Microsoft\Windows\INetCache\Content.Outlook\E5J8PVS6\pci_VM_101.xml)

1. Create guest through virsh

**#virsh create /etc/libvirt/qemu/guest.xml**

1. **Through qemu-kvm**
2. Add **pci-assign,host=02:00.0 to the qemu-kvm command.**

**# qemu-kvm -m 1600 -name VM5 -cpu host -device pci-assign,host=02:00.0 -net none -hda /root/shivam/CRAN\_VM2.img**

**NOTE :-**

If you have successfully passed the NIC to the virtual machine and still unable to bring the Virtual Machine to the network then follow the steps given below to solve the issue :-

1. Identify the NIC driver using (on host machine)

**#lspci -v**

For example tg3 is the driver for Broadcomm NICs

1. Add the driver to blacklist (on host machine)

**# vim /etc/modprobe.d/blacklist.conf**

Add **blacklist tg3** to the endof the script

1. Reboot the host machine and perform PCI pass-through with that NIC.