# **GVT-d Testenv setup on CentOS7.3**

**12/1/2017**

## **Install host CentOS7.3**

Host we used is SKL Kontron server.

Use USB or DVD to install CentOS7.3 with GUI server enabled and Development tools installed.

## **Configure and setup host**

### **Set proxy for host**

1. #sed -i '$ a\\ export https\_proxy=https://proxy-prc.intel.com:911\n export http\_proxy=http://proxy-prc.intel.com:911' ~/.bashrc
2. #sed -i '$ a\\ export https\_proxy=https://proxy-prc.intel.com:911\n export http\_proxy=http://proxy-prc.intel.com:911' /home/media/.bashrc
3. #sed -i '$ a\\ proxy=https://proxy-prc.intel.com:911\n proxy=http://proxy-prc.intel.com:911' /etc/yum.conf
4. #reboot

### **Upgrade kernel of Host server**

1. **Copy PV3\_GVT package to your host user package.**

You can get it from [\\SHWDEJOINTD026\Media\_validation\GVT](file:///\\SHWDEJOINTD026\Media_validation\GVT)

1. **Switch to root user and update Kernel**
   1. $su
   2. # cd /home/media/PV3\_GVT/kernel/20171027/
   3. #ls

kernel-4.14.0\_rc5-1.x86\_64.rpm

* 1. #rpm -Uvh kernel-4.14.0\_rc5-1.x86\_64.rpm --force --nodeps
  2. #reboot

1. **After host boot up, check the kernel if is 4.14.0**
2. $ uname -a

Linux localhost.localdomain 4.14.0-rc5 #1 SMP Thu Oct 19 04:30:33 CST 2017 x86\_64 x86\_64 x86\_64 GNU/Linux

1. **Install needed Dependecies**
   1. # yum install -y git vim redhat-lsb SDL-devel.x86\_64 SDL.x86\_64 zip.x86\_64 zlib.x86\_64 zlib-devel.x86\_64 glib2-devel.x86\_64 pixman-devel.x86\_64 spice-server.x86\_64 spice-server-devel.x86\_64 spice-protocol.noarch uuid
2. **Edit Host Grub**
   1. #vim /boo t/grub2/grub.cfg

### END /etc/grub.d/01\_users ###

### BEGIN /etc/grub.d/10\_linux ###

menuentry 'CentOS Linux (4.14.0-rc5) 7 (Core)' --class centos --class gnu-linux --class gnu --class os --unrestricted $menuentry\_id\_option 'gnulinux-3.10.0-514.el7.x86\_64-advanced-5d745ca1-c790-4ec9-833f-8911414adc9c' {

load\_video

set gfxpayload=keep

insmod gzio

insmod part\_msdos

insmod xfs

set root='hd0,msdos1'

if [ x$feature\_platform\_search\_hint = xy ]; then

search --no-floppy --fs-uuid --set=root --hint-bios=hd0,msdos1 --hint-efi=hd0,msdos1 --hint-baremetal=ahci0,msdos1 --hint='hd0,msdos1' 2b8e7114-4aad-489e-a4d3-7f45fa66c5fc

else

search --no-floppy --fs-uuid --set=root 2b8e7114-4aad-489e-a4d3-7f45fa66c5fc

fi

linux16 /vmlinuz-4.14.0-rc5 root=/dev/mapper/cl-root ro crashkernel=auto rd.lvm.lv=cl/root rd.lvm.lv=cl/swap rhgb quiet LANG=zh\_CN.UTF-8 console=ttyS0,115200n8 drm.debug=0x2 intel\_iommu=on

initrd16 /initramfs-4.14.0-rc5.img

}

* 1. #reboot

**NOTE:Marked yellow need to be added to grub file.**

1. **Build QEMU for KVM**

(as normal user)

* 1. Download qemu-2.10.1.tar.xz
  2. $ tar –xzvf qemu-2.10.1.tar.xz
  3. $ cd qemu-2.10.1/
  4. $ git submodule update --init dtc
  5. $ git submodule update --init roms/seabios
  6. $ ./configure --prefix=/usr --target-list=x86\_64-softmmu
  7. $ make -j8
  8. $ cd roms/seabios
  9. $make -j8
  10. $ cd -
  11. $ sudo make install
  12. $ sudo cp `pwd`/roms/seabios/out/bios.bin /usr/bin/bios.bin
  13. Copy bios.bin to /usr/bin
* $cd ~/PV3\_GVT/seabios
* $sudo cp bios.bin /usr/bin/
* $ sudo chmod 755 /usr/bin/ bios.bin
  1. Copy qemu-ifup and qemu-ifdown to your host /etc/

You can get from [\\SHWDEJOINTD026\Media\_validation\GVT](file:///\\SHWDEJOINTD026\Media_validation\GVT)

 

$sudo cp qemu-if\* /etc/

1. **Setup a proxy for external source download**

(as root user)

* 1. # touch /usr/bin/git-proxy
  2. # chmod 755 /usr/bin/git-proxy
  3. # cat /usr/bin/git-proxy

[core]

proxy=proxy-shz.intel.com

exec socat STDIO SOCKS4:$proxy:$1:$2

* 1. # touch /home/media/.gitconfig
  2. # cat .gitconfig

[core]

gitproxy = none for intel.com

gitproxy = git-proxy

1. **Load GVT module**

(as normal user)

* 1. Add below command in /etc/dracut.conf under line“# additional kernel modules to the default”.

add\_­drivers+="kvmgt vfio-iommu-type1 vfio-mdev"

* 1. $sudo mkinitrd /boot/initramfs-4.11.0+.img 4.11.0+ --force
  2. $sudo reboot
  3. After reboot, check the path **/sys/bus/pci/devices/0000:00:02.0/mdev\_supported\_types** if exist.

## **Install guest and configure guest**

### **Install guest**

1. Download Centos7.3 from mirrors.163.com or other source.

[**http://mirrors.163.com/centos/7.3.1611/isos/x86\_64/CentOS-7-x86\_64-DVD-1611.iso**](http://mirrors.163.com/centos/7.3.1611/isos/x86_64/CentOS-7-x86_64-DVD-1611.iso)

[**https://mirrors.aliyun.com/centos/7.3.1611/isos/x86\_64/CentOS-7-x86\_64-DVD-1611.iso**](https://mirrors.aliyun.com/centos/7.3.1611/isos/x86_64/CentOS-7-x86_64-DVD-1611.iso)

1. Use following command to install guest
2. # qemu-img create -f qcow2 CentOS73.qcow 50G
3. # cat installGuest.sh

#!/bin/bash

qemu-system-x86\_64 \

-m 2048 -smp 2 -M pc \

-name kvmgt -cpu host -hda /root/CentOS73.qcow \

-net nic -net tap,script=/etc/qemu-ifup \

-enable-kvm \

-machine kernel\_irqchip=on \

-net nic,model=e1000,macaddr=00:FE:EA:84:4F:dd \

-cdrom CentOS-7-x86\_64-DVD-1611.iso

NOTE: Every GUEST should have no conflict macaddr

1. **Boot up guest and install mssdriver**
   1. **Build bridge on host**
      1. **Copy the crbr.sh to host**



You can also get it from

[\\SHWDEJOINTD026\Media\_validation\GVT](file:///\\SHWDEJOINTD026\Media_validation\GVT)

* + 1. **Build bridge**
       1. #sh crbr.sh sw0 add enp1s0f0
       2. #dhclient sw0

NOTE: The enp1s0f0 can be changed to your own machine network card. Everytime you reboot the host you need repeat the steps above.

* 1. **Boot up guest system**

#cat bootCentOS.sh

#!/bin/bash

/usr/bin/qemu-system-x86\_64 \

-m 4096 -smp 4 -M pc \

-name CentOS -cpu host -hda /home/media/ CentOS73.qcow \

-net nic -net tap,script=/etc/qemu-ifup \

--enable-kvm \

-vga cirrus \

-device vfio-pci,host=00:02.0,id=hostdev0,bus=pci.0,addr=0x06 \

-usb -usbdevice tablet \

-machine kernel\_irqchip=on \

-net nic,model=e1000,macaddr=00:DE:EF:14:32:12

* 1. **Copy PV3 related packages to guest**
     1. **Update guest kernel to 4.14.0 as root user**

a) $su

b) # cd /home/media/PV3\_GVT/kernel/20171027/

c) #ls

kernel-4.14.0\_rc5-1.x86\_64.rpm

d) #rpm -Uvh kernel-4.14.0\_rc5-1.x86\_64.rpm --force –nodeps

e) #reboot

* + 1. **Update Guest Grub**

#cat /boot/grub2/grub.cfg

### END /etc/grub.d/01\_users ###

### BEGIN /etc/grub.d/10\_linux ###

menuentry 'CentOS Linux (4.14.0-rc5) 7 (Core)' --class centos --class gnu-linux --class gnu --class os --unrestricted $menuentry\_id\_option 'gnulinux-3.10.0-514.el7.x86\_64-advanced-913d10e7-c055-45d8-8a91-184b746a8930' {

load\_video

set gfxpayload=keep

insmod gzio

insmod part\_msdos

insmod xfs

set root='hd0,msdos1'

if [ x$feature\_platform\_search\_hint = xy ]; then

search --no-floppy --fs-uuid --set=root --hint-bios=hd0,msdos1 --hint-efi=hd0,msdos1 --hint-baremetal=ahci0,msdos1 --hint='hd0,msdos1' c13106cb-9128-4f3c-af0c-a715660c26ee

else

search --no-floppy --fs-uuid --set=root c13106cb-9128-4f3c-af0c-a715660c26ee

fi

linux16 /vmlinuz-4.14.0-rc5 root=/dev/mapper/cl-root ro crashkernel=auto rd.lvm.lv=cl/root rd.lvm.lv=cl/swap rhgb quiet LANG=en\_US.UTF-8 console=ttyS0,115200n8 ignore\_loglevel i915.enable\_hangcheck=0

initrd16 /initramfs-4.14.0-rc5.img

}

NOTE:Marked yellow need to be added to grub file which enable serial port, guest log output

* + 1. Install MSS driver and OPENCL

(as root user)

* + - 1. #cd /home/media/PV3\_GVT/MSS\_PV3
      2. #tar zxvf PV3.tar.gz
      3. #cd MediaServerStudioEssentials2017R3/SDK2017Production16.5.2/Generic
      4. #./install\_media.sh
      5. #tar xvf intel-opencl-cpu-r5.0-63503.x86\_64.tar.xz
      6. #tar xvf intel-opencl-r5.0-63503.x86\_64.tar.xz
      7. #tar xvf tar xvf intel-opencl-devel-r5.0-63503.x86\_64.tar.xz
      8. #cp -raf etc/\* /etc/
      9. #cp -raf opt/intel/opencl/ /opt/intel/
      10. #ldconfig
      11. Close qemu tool on host

NOTE:When execute step4, if the machine is SKL, choose y, other choose n, after the guest started normally again, the vainfo command may show unconnect info, try command “unset DISPLAY”

1. **Do It First After Reboot The Host**

Make a device assignable by using vfio driver. For example, if you wanted to make the device at BDF 00:02.0 available for guests, you could type the following:

#modprobe vfio

#modprobe vfio\_pci

#echo 0000:00:02.0 > /sys/bus/pci/devices/0000\:00\:02.0/driver/unbind

Start on VNC windows:

#lspci -Dnn | egrep "^0000:00:02"

0000:00:02.0 VGA compatible controller [0300]: Intel Corporation Device [8086:193d] (rev 09)

#echo 8086 193d > /sys/bus/pci/drivers/vfio-pci/new\_id

#ls /dev/vfio

2 vfio

## On The VNC **Windows**

We can start the guest os in VNC windows.

#cat bootCentOS.sh

#!/bin/bash

qemu-system-x86\_64 \

-m 4096 -smp 4 -M pc \

-k en-us \

-sdl \

-name CentOS -cpu host -hda /home/media/CentOS73.qcow \

-net nic -net tap,script=/etc/qemu-ifup \

-bios /usr/bin/bios.bin \

--enable-kvm \

-vga cirrus \

-device vfio-pci,host=00:02.0,id=hostdev0,bus=pci.0,addr=0x06 \

-usb -usbdevice tablet \

-machine kernel\_irqchip=on \

-net nic,model=e1000,macaddr=00:DE:EF:14:32:12

### **Mount test system on host and run case**

1. **Setup on host**

(as root user)

* 1. #yum install nfs-utils -y
  2. #vim /etc/exports

/home/media/test\_system \*(rw,sync,no\_root\_squash)

* 1. # systemctl enable rpcbind.service
  2. #systemctl enable nfs-server.service
  3. #systemctl start rpcbind.service
  4. #systemctl start nfs-server.service
  5. #iptables -X
  6. #iptables –F
  7. systemctl status nfs-server

(as normal user)­

* 1. $cd /home/media/ PV3\_GVT/
  2. $ unzip test\_system.zip -d /home/media/test\_system
  3. $chmod 775 -R /home/media/test\_system
  4. Copy sample app to test\_system
     1. $cd ~/PV3\_GVT/MSS\_PV3/MediaServerStudioEssentials2017R3/MediaSamples\_Linux\_2017R3\_b698/samples/\_bin/x64
     2. $sudo cp \*.so ocl\_rotate.cl sample\_\* /home/media/test\_system/build/lin\_x64/bin
  5. Copy 6.0 tool to test\_system

You can get from [\\SHWDEJOINTD026\Media\_validation\validation\_tools\6.0](file:///\\SHWDEJOINTD026\Media_validation\validation_tools\6.0)

$sudo cp -r 6.0 /home/media/test\_system

**NOTE: Everytime you reboot your host, you need repeat step g and h, or you will not find the mountserver.**

1. **Setup on guest**

(as normal user)

* 1. $mkdir -p /home/media/ws/msdk\_validation/mediasdk\_streams
  2. $mkdir -p /home/media/ws/msdk\_validation/test\_system
  3. $chmod 775 -R /home/media/ws
  4. $showmount -e HOST\_IP

Export list for HOST\_IP:

/home/media/test\_system \*

(as root user)

* 1. #mount -t nfs -o nolock,nfsvers=3,vers=3 -o proto=tcp  HOST\_IP:/home/media/test\_system /home/media/ws/msdk\_validation/test\_system/

#####

#mount -t nfs -o vers=3 10.67.116.91:/datadisk/streams /home/media/ws/msdk\_validation/mediasdk\_streams

1. **Run msdk case on guest**
   1. Enter the test\_system package on guest

#cd /home/media/ws/msdk\_validation/test\_system

* 1. #export MEDIASDK\_STREAMS=/home/media/ws/msdk\_validation/mediasdk\_streams
  2. #export MEDIASDK\_ROOT=/home/media/ws/msdk\_validation/test\_system
  3. #export LD\_LIBRARY\_PATH=./:/user/local/lib:/usr/lib64:/opt/intel/mediasdk/lib64:/opt/intel/common/mdf/lib64:/opt/intel/opencl:/home/media/ws/msdk\_validation/test\_system/build/lin\_x64/bin
  4. # perl test\_driver.pl -s h264d\_common -p c7.3\_skl\_64\_server --copy\_missed\_streams