Environment Introduce && Guide

Docker Container with mediaSDK Test:

http://los-vmm.sc.intel.com/wiki/Docker_Container

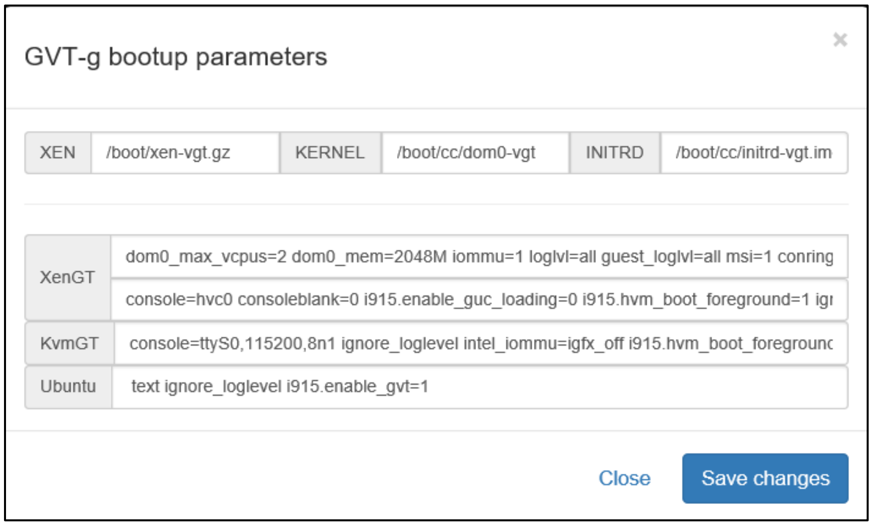
Docker in Clear Container with Yami Test:

http://los-vmm.sc.intel.com/wiki/Docker_in_Container_with_Yami_Test

Name: lujie-cc-testing
User：root   Pwd：123456
Purpose:

1. Create Clear Container in Host;
2. Create Docker Container in Clear Container;
3. Test the performance with Yami;

Details:
Glab Grub parameters:



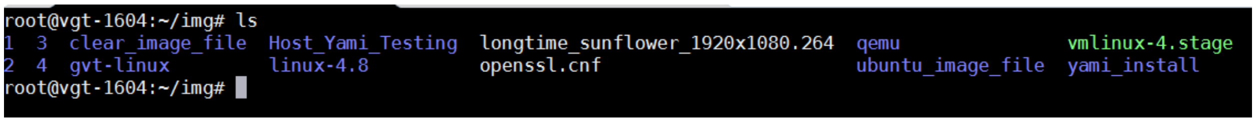
GVT-g bootup parameters

XEN	/boot/xen-vgt.gz	KERNEL	/boot/cc/dom0-vgt	INITRD	/boot/cc/initrd-vgt.im

XenGT	dom0_max_vcpus=2 dom0_mem=2048M iommu=1 loglvl=all guest_loglvl=all msi=1 conring
	console=hvc0 consoleblank=0 i915.enable_guc_loading=0 i915.hvm_boot_foreground=1 igı
KvmGT	console=ttyS0,115200,8n1 ignore_loglevel intel_iommu=igfx_off i915.hvm_boot_foregrounc
Ubuntu	text ignore_loglevel i915.enable_gvt=1

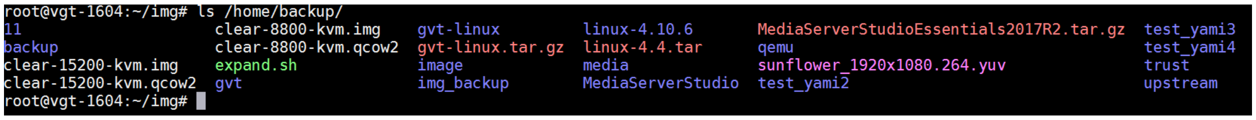
```
console=ttyS0,115200,8n1 ignore_loglevel intel_iommu=igfx_off i915.hvm_boot_foreground=1
log_buf_len=128M drm.debug=0 i915.enable_gvt=1
```

/root/img/ :



```
root@vgt-1604:~/img# ls
1  3  clear_image_file  Host_Yami_Testing  longtime_sunflower_1920x1080.264  qemu               vmlinux-4.stage
2  4  gvt-linux         linux-4.8          openssl.cnf                       ubuntu_image_file  yami_install
root@vgt-1604:~/img# 
```

/home/backup/ :

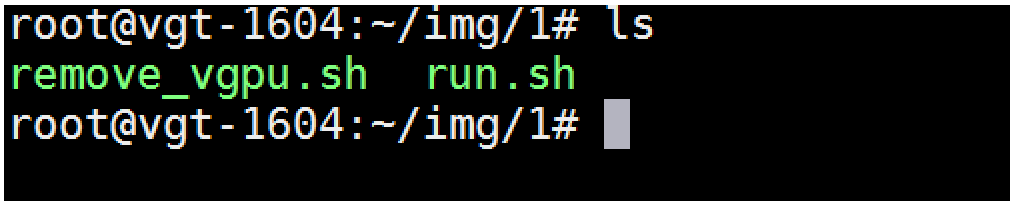


```
root@vgt-1604:~/img# ls /home/backup/
11                     clear-8800-kvm.img    gvt-linux         linux-4.10.6       MediaServerStudioEssentials2017R2.tar.gz  test_yami3
backup                 clear-8800-kvm.qcow2  gvt-linux.tar.gz  linux-4.4.tar      qemu                                      test_yami4
clear-15200-kvm.img    expand.sh             image             media              sunflower_1920x1080.264.yuv               trust
clear-15200-kvm.qcow2  gvt                   img_backup        MediaServerStudio  test_yami2                                upstream
root@vgt-1604:~/img# 
```

All things are in the /root/img directory and /home/backup directories:

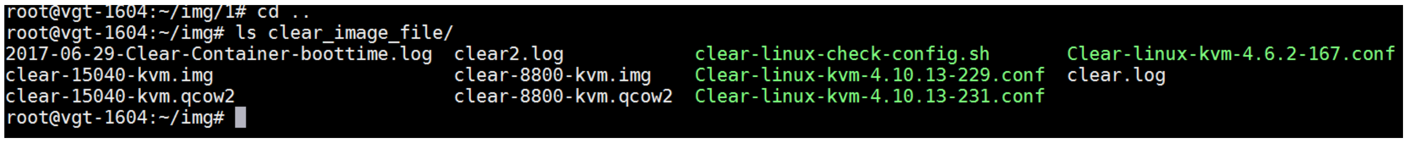
For /root/img directory in Host:

1. The 1/, 2/, 3/, 4/ directories is used to create clear containers named 1 to 4. You just need to run the script named "run.sh" to create a container. After you exit the clear container,you need to run the script named "remove_vgpu.sh" to release the vgpu.



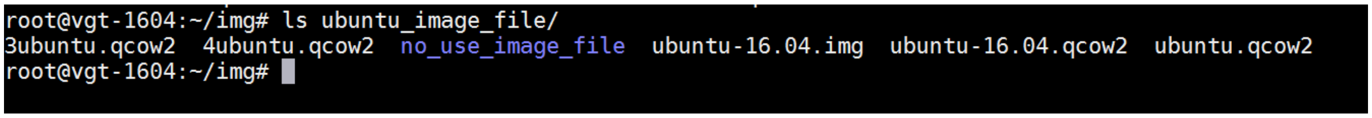
```
root@vgt-1604:~/img/1# ls
remove_vgpu.sh  run.sh
root@vgt-1604:~/img/1# 
```

2. The "clear_image_file" directory is used to save the clear image which I have ever tried to use. Although I failed, I think maybe it's worth to save.



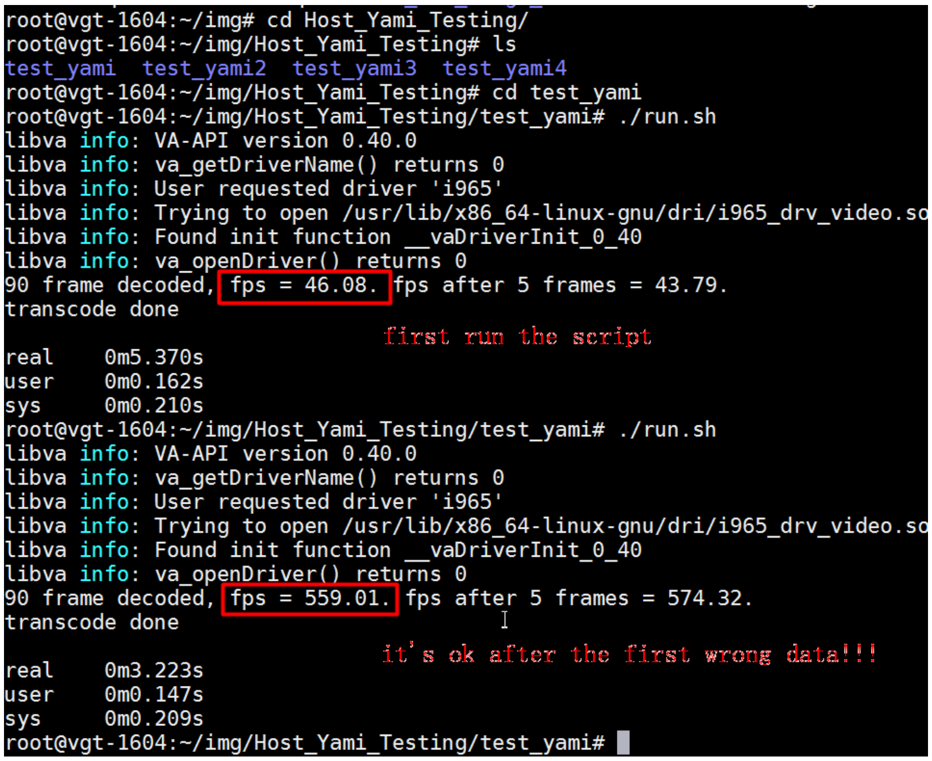
```
root@vgt-1604:~/img/1# cd ..
root@vgt-1604:~/img# ls clear_image_file/
2017-06-29-Clear-Container-boottime.log  clear2.log            clear-linux-check-config.sh       Clear-linux-kvm-4.6.2-167.conf
clear-15040-kvm.img                      clear-8800-kvm.img    Clear-linux-kvm-4.10.13-229.conf  clear.log
clear-15040-kvm.qcow2                    clear-8800-kvm.qcow2  Clear-linux-kvm-4.10.13-231.conf
root@vgt-1604:~/img# 
```

3. The "ubuntu_image_file" directory is used to save the ubuntu image which we need.



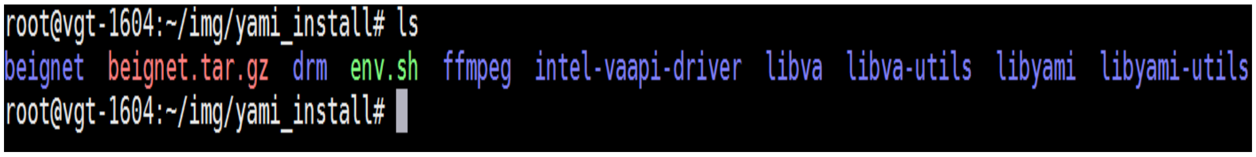
```
root@vgt-1604:~/img# ls ubuntu_image_file/
3ubuntu.qcow2  4ubuntu.qcow2  no_use_image_file  ubuntu-16.04.img  ubuntu-16.04.qcow2  ubuntu.qcow2
root@vgt-1604:~/img# 
```

4. The "Host_Yami_Testing" directory is used to test the perfrmance in Host. Pay attention to the wrong data created by running the script at first. Luckily, It is ok when you try again.



```
root@vgt-1604:~/img# cd Host_Yami_Testing/
root@vgt-1604:~/img/Host_Yami_Testing# ls
test_yami  test_yami2  test_yami3  test_yami4
root@vgt-1604:~/img/Host_Yami_Testing# cd test_yami
root@vgt-1604:~/img/Host_Yami_Testing/test_yami# ./run.sh
libva info: VA-API version 0.40.0
libva info: va_getDriverName() returns 0
libva info: User requested driver 'i965'
libva info: Trying to open /usr/lib/x86_64-linux-gnu/dri/i965_drv_video.so
libva info: Found init function __vaDriverInit_0_40
libva info: va_openDriver() returns 0
90 frame decoded, fps = 46.08. fps after 5 frames = 43.79.
transcode done
                              first run the script
real    0m5.370s
user    0m0.162s
sys     0m0.210s
root@vgt-1604:~/img/Host_Yami_Testing/test_yami# ./run.sh
libva info: VA-API version 0.40.0
libva info: va_getDriverName() returns 0
libva info: User requested driver 'i965'
libva info: Trying to open /usr/lib/x86_64-linux-gnu/dri/i965_drv_video.so
libva info: Found init function __vaDriverInit_0_40
libva info: va_openDriver() returns 0
90 frame decoded, fps = 559.01. fps after 5 frames = 574.32.
transcode done
                              it's ok after the first wrong data!!!
real    0m3.223s
user    0m0.147s
sys     0m0.209s
root@vgt-1604:~/img/Host_Yami_Testing/test_yami#
```

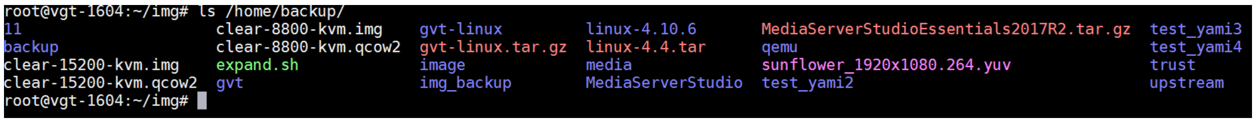
5. The “yami_install” directory is used to install the yami.



```
root@vgt-1604:~/img/yami_install# ls
beignet  beignet.tar.gz  drm  env.sh  ffmpeg  intel-vaapi-driver  libva  libva-utils  libyami  libyami-utils
root@vgt-1604:~/img/yami_install#
```

For /home/backup/ :

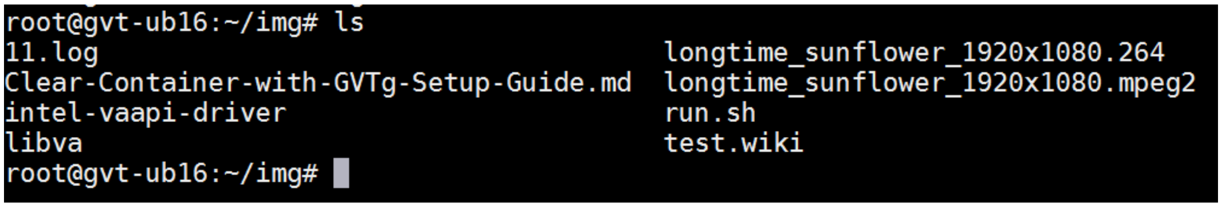
The /home/backup directory is based on a Hard Disk. I backuped the gvt-linux kernel files and qemu-lite files(named “qemu” in this derictory) there.



```
root@vgt-1604:~/img# ls /home/backup/
11                    clear-8800-kvm.img    gvt-linux         linux-4.10.6        MediaServerStudioEssentials2017R2.tar.gz  test_yami3
backup                clear-8800-kvm.qcow2  gvt-linux.tar.gz  linux-4.4.tar       qemu                                      test_yami4
clear-15200-kvm.img   expand.sh             image             media               sunflower_1920x1080.264.yuv               trust
clear-15200-kvm.qcow2 gvt                   img_backup        MediaServerStudio   test_yami2                                upstream
root@vgt-1604:~/img#
```

For in the clear container:

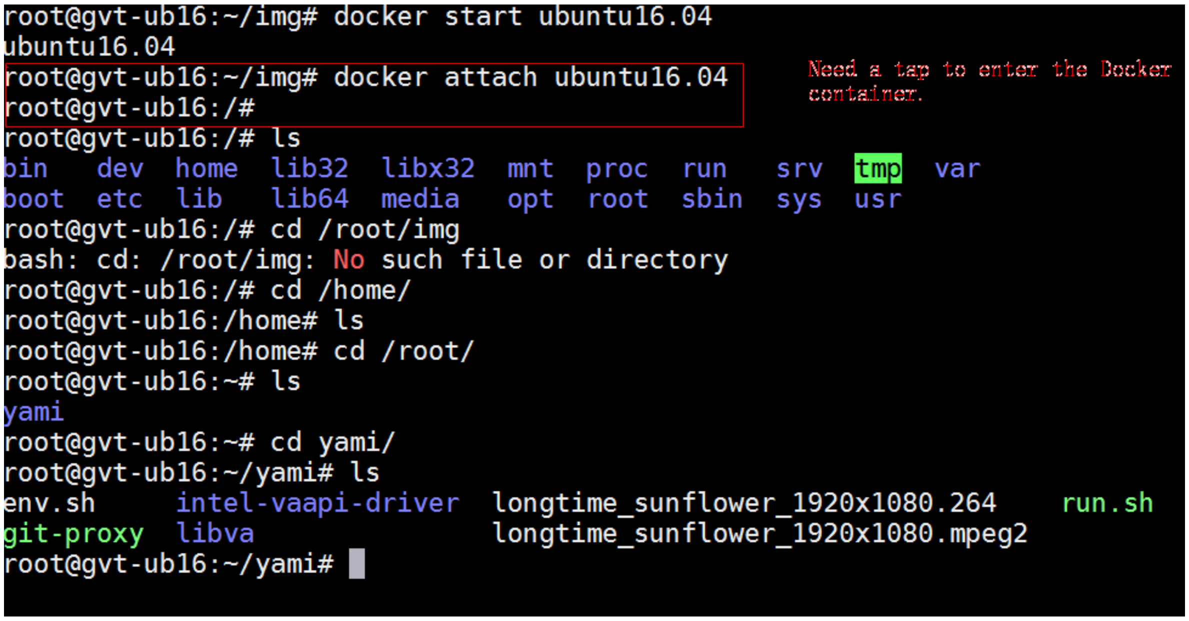
You can run the script named “run.sh” in /root/img directory to test the performance in the clear container.



```
root@gvt-ub16:~/img# ls
11.log                                    longtime_sunflower_1920x1080.264
Clear-Container-with-GVTg-Setup-Guide.md  longtime_sunflower_1920x1080.mpeg2
intel-vaapi-driver                        run.sh
libva                                     test.wiki
root@gvt-ub16:~/img#
```

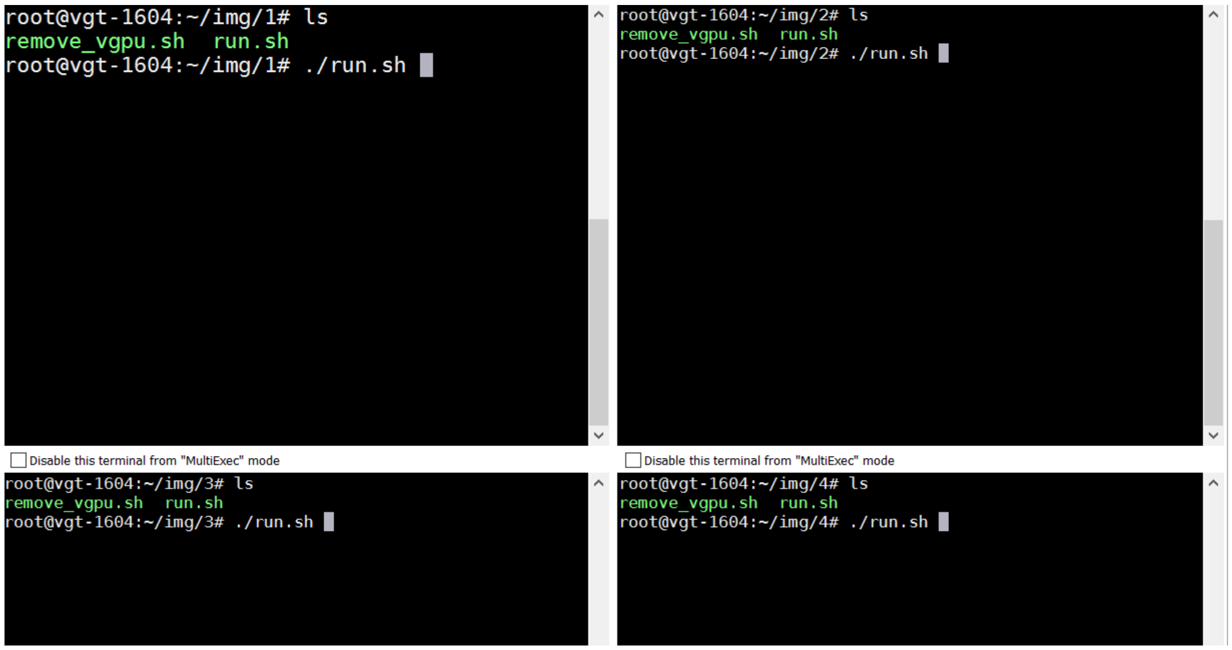
For creating Docker Container in the Clear container：

When you attach a Docker Container,maybe there needs a “Enter” tap to enter the Docker Container.Beacuse I used the host mode network, the terminal is the same as the Clear Container, but the other parts are absolutely different. You can enter the /root/yami directory to test the performance with “run.sh” script.



```
root@gvt-ub16:~/img# docker start ubuntu16.04
ubuntu16.04
root@gvt-ub16:~/img# docker attach ubuntu16.04
root@gvt-ub16:/#
root@gvt-ub16:/# ls
bin   dev  home  lib32  libx32  mnt  proc  run   srv  tmp  var
boot  etc  lib   lib64  media   opt  root  sbin  sys  usr
root@gvt-ub16:/# cd /root/img
bash: cd: /root/img: No such file or directory
root@gvt-ub16:/# cd /home/
root@gvt-ub16:/home# ls
root@gvt-ub16:/home# cd /root/
root@gvt-ub16:~# ls
yami
root@gvt-ub16:~# cd yami/
root@gvt-ub16:~/yami# ls
env.sh     intel-vaapi-driver  longtime_sunflower_1920x1080.264   run.sh
git-proxy  libva               longtime_sunflower_1920x1080.mpeg2
root@gvt-ub16:~/yami#
```

Ps:if you want to create four clear containers and test their performances at the same time, you can use a software named “MobaXterm” to connect your Machine. And use the “MultiExec” to do that.



```
root@vgt-1604:~/img/1# ls
remove_vgpu.sh  run.sh
root@vgt-1604:~/img/1# ./run.sh
```

```
root@vgt-1604:~/img/2# ls
remove_vgpu.sh  run.sh
root@vgt-1604:~/img/2# ./run.sh
```

Disable this terminal from "MultiExec" mode
Disable this terminal from "MultiExec" mode

```
root@vgt-1604:~/img/3# ls
remove_vgpu.sh  run.sh
root@vgt-1604:~/img/3# ./run.sh
```

```
root@vgt-1604:~/img/4# ls
remove_vgpu.sh  run.sh
root@vgt-1604:~/img/4# ./run.sh
```

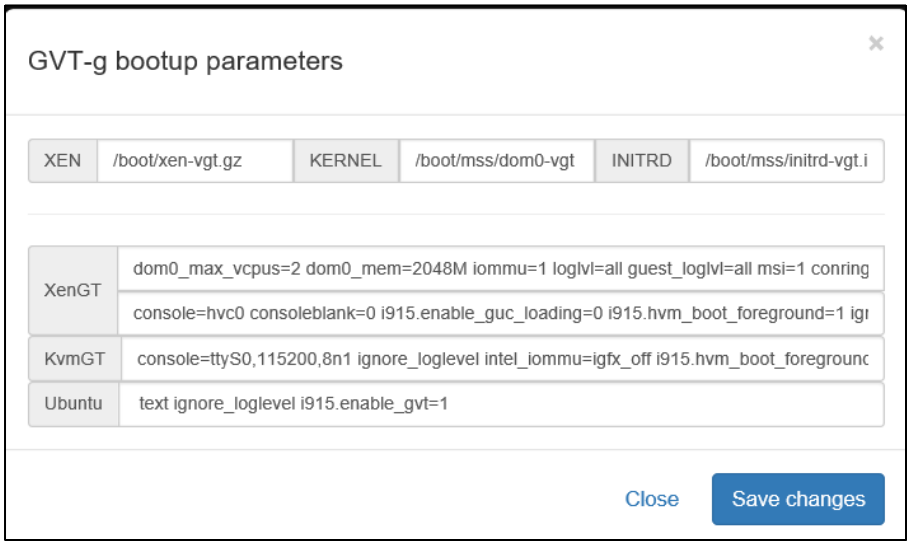
Name:lj-container

User： root Pwd： 123456

Purpose: MediaSDK Test in Host and Docker Container

Details:

Glab grub parameters:

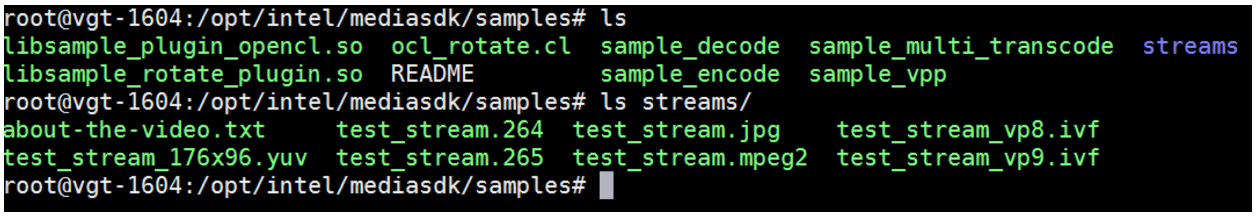


GVT-g bootup parameters

XEN	/boot/xen-vgt.gz	KERNEL	/boot/mss/dom0-vgt	INITRD	/boot/mss/initrd-vgt.i

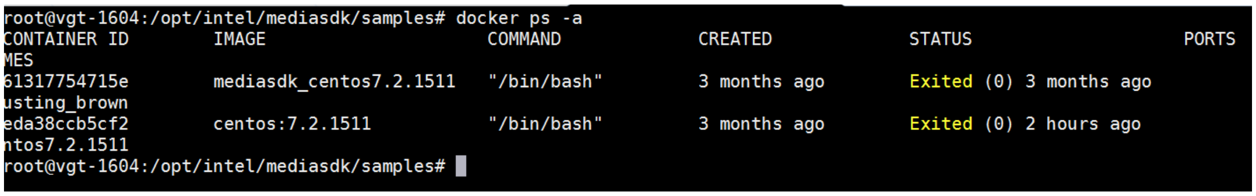
XenGT	dom0_max_vcpus=2 dom0_mem=2048M iommu=1 loglvl=all guest_loglvl=all msi=1 conring
	console=hvc0 consoleblank=0 i915.enable_guc_loading=0 i915.hvm_boot_foreground=1 igı
KvmGT	console=ttyS0,115200,8n1 ignore_loglevel intel_iommu=igfx_off i915.hvm_boot_foregrounc
Ubuntu	text ignore_loglevel i915.enable_gvt=1

For Host MediaSDK Test:



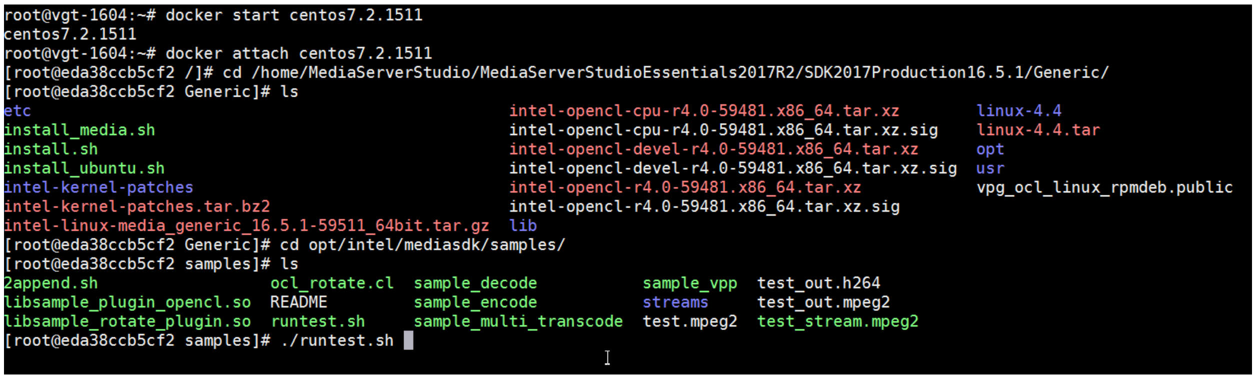
```
root@vgt-1604:/opt/intel/mediasdk/samples# ls
libsample_plugin_opencl.so  ocl_rotate.cl  sample_decode  sample_multi_transcode  streams
libsample_rotate_plugin.so  README         sample_encode  sample_vpp
root@vgt-1604:/opt/intel/mediasdk/samples# ls streams/
about-the-video.txt     test_stream.264  test_stream.jpg    test_stream_vp8.ivf
test_stream_176x96.yuv  test_stream.265  test_stream.mpeg2  test_stream_vp9.ivf
root@vgt-1604:/opt/intel/mediasdk/samples# 
```

Docker Containers(already installed mediaSDK):



```
root@vgt-1604:/opt/intel/mediasdk/samples# docker ps -a
CONTAINER ID        IMAGE                     COMMAND             CREATED             STATUS                     PORTS
MES
61317754715e        mediasdk_centos7.2.1511   "/bin/bash"         3 months ago        Exited (0) 3 months ago
usting_brown
eda38ccb5cf2        centos:7.2.1511           "/bin/bash"         3 months ago        Exited (0) 2 hours ago
ntos7.2.1511
root@vgt-1604:/opt/intel/mediasdk/samples# 
```

You can test the mediaSDK performance with the script named “runtest.sh”.



```
root@vgt-1604:~# docker start centos7.2.1511
centos7.2.1511
root@vgt-1604:~# docker attach centos7.2.1511
[root@eda38ccb5cf2 /]# cd /home/MediaServerStudio/MediaServerStudioEssentials2017R2/SDK2017Production16.5.1/Generic/
[root@eda38ccb5cf2 Generic]# ls
etc                                                  intel-opencl-cpu-r4.0-59481.x86_64.tar.xz        linux-4.4
install_media.sh                                     intel-opencl-cpu-r4.0-59481.x86_64.tar.xz.sig    linux-4.4.tar
install.sh                                           intel-opencl-devel-r4.0-59481.x86_64.tar.xz      opt
install_ubuntu.sh                                    intel-opencl-devel-r4.0-59481.x86_64.tar.xz.sig  usr
intel-kernel-patches                                 intel-opencl-r4.0-59481.x86_64.tar.xz            vpg_ocl_linux_rpmdeb.public
intel-kernel-patches.tar.bz2                         intel-opencl-r4.0-59481.x86_64.tar.xz.sig
intel-linux-media_generic_16.5.1-59511_64bit.tar.gz  lib
[root@eda38ccb5cf2 Generic]# cd opt/intel/mediasdk/samples/
[root@eda38ccb5cf2 samples]# ls
2append.sh                   ocl_rotate.cl  sample_decode           sample_vpp   test_out.h264
libsample_plugin_opencl.so   README         sample_encode           streams      test_out.mpeg2
libsample_rotate_plugin.so   runtest.sh     sample_multi_transcode  test.mpeg2   test_stream.mpeg2
[root@eda38ccb5cf2 samples]# ./runtest.sh
```