



Winter Semester 2017-18

# **Live Migration on KVM**

## **PROJECT REPORT**

**Submitted for CAL in B.Tech Virtualization(CSE4011)**

**By**

**Tushar Pahuja                      15BCE1252**

**Osho Agyeya                      15BCE1326**

**Shivam Attree                      15BCE1044**

**Faculty: Prof Kumar R.  
(SCHOOL OF COMPUTING SCIENCE AND ENGINEERING)**

# CERTIFICATE

This is to certify that the Project work entitled "***LIVE MIGRATION ON KVM***" that is being submitted by "***TUSHAR PAHUJA, OSHO AGYEYA, SHIVAM ATTREE***" for CAL in B.Tech Virtualisation (CSE4011) is a record of bonafide work done under my supervision.

Place: Chennai

Date: 05/04/2018

## **Signature of Students:**

TUSHAR PAHUJA

OSHO AGYEYA

SHIVAM ATTREE

## **Signature of Faculty:**

Prof Kumar R

# ACKNOWLEDGEMENTS

We thank VIT University (**SCHOOL OF COMPUTING SCIENCE AND ENGINEERING**) for giving us the opportunity to conduct this project and experiment. We also thank our guide for project Dr. R Kumar for his constant, good and knowledgeable guidance for the project. Through this project, we learned many of new things about Live Migration and Virtualisation which will be definitely useful for us.

OSHO AGYEYA  
Reg. No. 15BCE1326

TUSHAR PAHUJA  
Reg. No. 15BCE1252

SHIVAM ATTREE  
Reg. No. 15BCE1044

# ABSTRACT

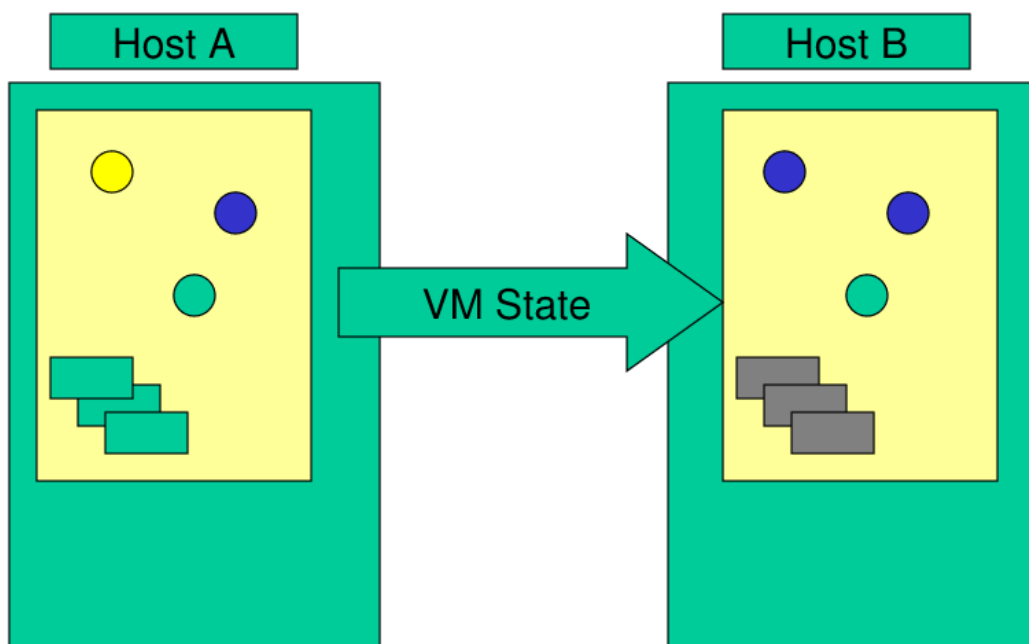
Live migration enables the movement of virtual machines (VMs) from one physical host to another without downtime. It's often used when administrators need to perform hardware maintenance on a physical host. You can just live-migrate all VMs off the host, then live-migrate them back once maintenance is completed. Migration describes the process of moving a guest virtual machine from one host physical machine to another. This is possible because guest virtual machines are running in a virtualized environment instead of directly on the hardware. Performing a KVM live migration requires the the same kind of CPU on the hosts, the same kernel version and shared storage among the migrating VMs. Shared storage ensures that the host computers can access the storage back end during a KVM live migration.

This project is aimed at performing live migration of a linuxmint virtual machine between two Ubuntu physical hosts using Virtual Machine Manager. Along with this, all the necessary system setup and installation has been described in detail.

---

# Introduction

Live migration refers to the process of moving a running virtual machine or application between different physical machines without disconnecting the client or application. Memory, storage, and network connectivity of the virtual machine are transferred from the original guest machine to the destination.



## **REQUIREMENTS:**

1. OS – Ubuntu 16.04 (any OS can be used but all the host involved in Live Migration should have the same OS).
2. Hypervisor – KVM (any linux based hypervisor can be used eg. Xen for managing the VMs)
3. Shared Storage – NFS (most important requirement for the Live Migration)
4. Virtual Machines – Any number of VMs can be created and can be managed by using KVM (Kernel Virtual Machine)

## CONCEPTS:

### 1. VIRTUAL MACHINE

A Virtual Machine is an Operating System or application environment that is installed on software which imitates dedicated hardware. The end user has the same experience on a Virtual Machine as they would have on dedicated hardware.

### 2. LIVE MIGRATION

Live Migration refers to the process of moving a running Virtual Machine or application between different physical machines without disconnecting the client or application.

### 3. KVM (Kernel Virtual Machine)

Kernel-based Virtual Machine (KVM) is a virtualization infrastructure for the linux kernel that turns it into a hypervisor.

### 4. NFS

NFS, or Network File System, is a distributed file system protocol that allows you to mount remote directories on your server. NFS provides a relatively quick and easy way to access systems over a network and works well in situations where the shared resources will be accessed regularly.

Migration works by sending the state of the guest virtual machine's memory and any virtualized devices to a destination host physical machine. It is recommended to use shared, networked storage to store the guest virtual machine's images to be migrated. It is also recommended to use libvirt-managed storage pools for shared storage when migrating virtual machines.

---

# MOTIVATION

Migration is useful for:

- Load balancing - guest virtual machines can be moved to host physical machines with lower usage when their host physical machine becomes overloaded, or another host physical machine is under-utilized.
- Hardware independence - when we need to upgrade, add, or remove hardware devices on the host physical machine, we can safely relocate guest virtual machines to other host physical machines. This means that guest virtual machines do not experience any downtime for hardware improvements.
- Energy saving - guest virtual machines can be redistributed to other host physical machines and can thus be powered off to save energy and cut costs in low usage periods.
- Geographic migration - guest virtual machines can be moved to another location for lower latency or in serious circumstances.

Migrations can be performed live or not.

In a **live migration**, the guest virtual machine continues to run on the source host physical machine while its memory pages are transferred, in order, to the destination host physical machine. During migration, KVM monitors the source for any changes in pages it has already transferred and begins to transfer these changes when all of the initial pages have been transferred. KVM also estimates transfer speed during migration, so when the remaining amount of data to transfer will take a certain configurable period of time, KVM suspends the original guest virtual machine, transfers the remaining data, and resumes the same guest virtual machine on the destination host physical machine.

A migration that is **not performed live**, suspends the guest virtual machine, then moves an image of the guest virtual machine's memory to the destination host physical machine. The guest virtual machine is then resumed on the destination host physical machine and the memory the guest virtual machine used on the source host physical machine is freed. The time it takes to complete such a migration depends on network bandwidth and latency. If the network is experiencing heavy use or low bandwidth, the migration will take much longer.

If the original guest virtual machine modifies pages faster than KVM can transfer them to the destination host physical machine, offline migration must be used, as live migration would never complete.

---

# IMPLEMENTATION DETAILS

## STEPS INVOLVED:

### 1.) INSTALLATION OF UBUNTU ON PHYSICAL HOSTS

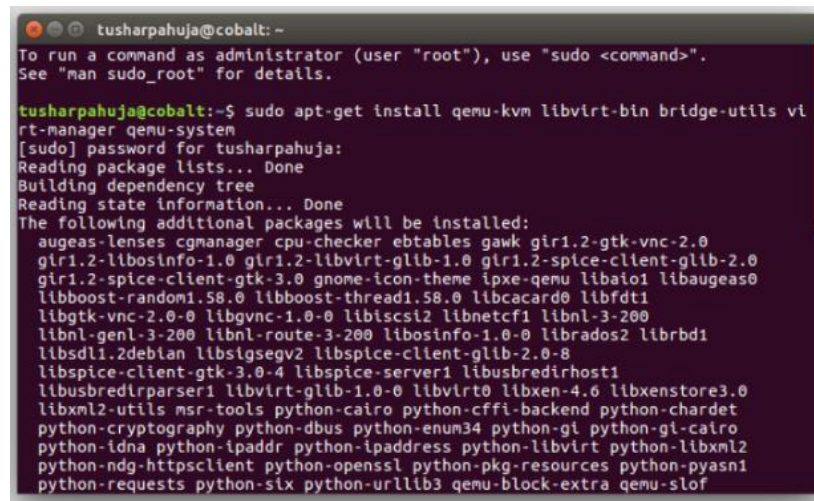
The first step is to ensure that Ubuntu 16.04 is properly installed on both physical machines and is up and running.

The image can be obtained from <http://releases.ubuntu.com/16.04.4/>

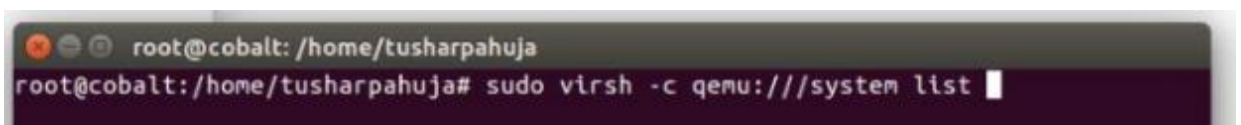
The steps below are supposed to be executed on both the hosts.

### 2.) INSTALLING OF KVM & VERIFICATION

The following screenshots showcase the installation and verification of KVM:



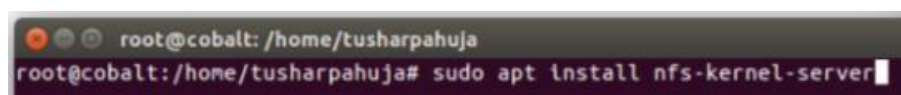
```
tusharpahuja@cobalt: ~  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
  
tusharpahuja@cobalt:~$ sudo apt-get install qemu-kvm libvirt-bin bridge-utils vi  
rt-manager qemu-system  
[sudo] password for tusharpahuja:  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following additional packages will be installed:  
  augeas-lenses cgmanager cpu-checker ebttables gawk gir1.2-gtk-vnc-2.0  
  gir1.2-libosinfo-1.0 gir1.2-libvirt-glib-1.0 gir1.2-spice-client-glib-2.0  
  gir1.2-spice-client-gtk-3.0 gnome-icon-theme ipxe-qemu libaio1 libaugeas0  
  libboost-random1.58.0 libboost-thread1.58.0 libcacard0 libfdt1  
  libgtk-vnc-2.0-0 libgvnc-1.0-0 libiscsi2 libnetcf1 libnl-3-200  
  libnl-genl-3-200 libnl-route-3-200 libosinfo-1.0-0 librados2 librbdl1  
  libsd1.2debian libsigsegv2 libspice-client-glib-2.0-8  
  libspice-client-gtk-3.0-4 libspice-server1 libusbredirhost1  
  libusbredirparser1 libvirt-glib-1.0-0 libvirt0 libxen-4.6 libxenstore3.0  
  libxml2-utils msr-tools python-cairo python-ccfi-backend python-chardet  
  python-cryptography python-dbus python-enum34 python-gi python-gi-cairo  
  python-idna python-ipaddr python-ipaddress python-libvirt python-libxml2  
  python-ndg-httpsclient python-openssl python-pkg-resources python-pyasn1  
  python-requests python-six python-urllib3 qemu-block-extra qemu-slof
```



```
root@cobalt: /home/tusharpahuja  
root@cobalt:/home/tusharpahuja# sudo virsh -c qemu:///system list
```

### 3.) NFS MOUNTING

- 1.) This needs to be done carefully. Both machines act as separate hosts.
- 2.) NFS is a common directory on both the host 1 and host 2.
- 3.) Host 1 ip address: 192.168.43.121 (Tushar's PC)
- 4.) Host 2 ip address: 192.168.43.121 (Osho's PC)
- 5.) The following screenshots showcase the installation:



```
root@cobalt: /home/tusharpahuja  
root@cobalt:/home/tusharpahuja# sudo apt install nfs-kernel-server
```



6.) Make a directory for NFS:

```
root@cobalt:/home/tusharpahuja# sudo mkdir /var/lib/libvirt/images -p
root@cobalt:/home/tusharpahuja# ls -la /var/lib/libvirt/images
total 1638616
drwx--x--x 2 root      root      4096 Mar 29 16:46 .
drwxr-xr-x 7 root      root      4096 Mar 29 16:25 ..
-rw----- 1 libvirt-qemu kvm 10739318784 Mar 29 16:46 generic-tushar.qcow2
-rw-r--r-- 1 libvirt-qemu kvm 1676083200 Jun 28 2017 linuxmint-18.2-cinnamon-64bit.iso
root@cobalt:/home/tusharpahuja#
```

7.) Permit everyone to connect from any group:

```
root@cobalt:/home/tusharpahuja# sudo chown nobody:nogroup /var/lib/libvirt/images
root@cobalt:/home/tusharpahuja#
```

8.) Map the NSF path in /etc/export folder:

```
Last login: Mon Mar 19 23:29:13 2018 from 192.168.43.95
root@osho-GL552VW:~# mount -tnfs 192.168.43.95:/var/lib/libvirt/images /var/lib/libvirt/images
root@osho-GL552VW:~# cat /etc/exports
# /etc/exports: the access control list for filesystems which may be exported
#                 to NFS clients.  See exports(5).
#
# Example for NFSv2 and NFSv3:
# /srv/homes      hostname1(rw,sync,no_subtree_check) hostname2(ro,sync,no_subtree_check)
#
# Example for NFSv4:
# /srv/nfs4       gss/krb5i(rw,sync,fsid=0,crossmnt,no_subtree_check)
# /srv/nfs4/homes gss/krb5i(rw,sync,no_subtree_check)
/var/lib/libvirt/images 192.168.122.84(rw,sync,no_root_squash)
root@osho-GL552VW:~#
```

9.) Restart the nfs-kernel-server

```
root@cobalt:/home/tusharpahuja# sudo chown nobody:nogroup /var/lib/libvirt/images
root@cobalt:/home/tusharpahuja# sudo view /etc/exports
root@cobalt:/home/tusharpahuja# sudo systemctl restart nfs-kernel-server
```

10.) Check listening port after allowing from any to any port NFS

```
root@cobalt:/home/tusharpahuja/Desktop# ufw status
Status: inactive
root@cobalt:/home/tusharpahuja/Desktop# sudo ufw allow from any to any port nfs
Skipping adding existing rule
Skipping adding existing rule (v6)
root@cobalt:/home/tusharpahuja/Desktop# ufw status
Status: inactive
root@cobalt:/home/tusharpahuja/Desktop# clear

root@cobalt:/home/tusharpahuja# sudo ufw status
Status: active

To Action From
--
2049 ALLOW Anywhere
2049 (v6) ALLOW Anywhere (v6)
root@cobalt:/home/tusharpahuja#
```

11.) Mount the 2<sup>nd</sup> host in the directory of 1<sup>st</sup> host.

```
root@osho-GL552VW: /home/osho
osho@osho-GL552VW:~$ su
Password:
root@osho-GL552VW:/home/osho# virt-manager
root@osho-GL552VW:/home/osho# sudo mount 192.168.43.121:/var/lib/libvirt/images
/var/lib/libvirt/images
root@osho-GL552VW:/home/osho#
```

12.) Verification of step 11

```
root@osho-GL552VW: /home/osho
root@osho-GL552VW:/home/osho# sudo mount 192.168.43.121:/var/lib/libvirt/images /var/lib/libvirt/images
root@osho-GL552VW:/home/osho# df -h
Filesystem                                Size  Used Avail Use% Mounted on
udev                                     7.8G   0 7.8G   0% /dev
tmpfs                                    1.6G  18M  1.6G   2% /run
/dev/sda8                                36G   13G   22G  37% /
tmpfs                                    7.8G   26M  7.8G   1% /dev/shm
tmpfs                                    5.0M   4.0K  5.0M   1% /run/lock
tmpfs                                    7.8G   0 7.8G   0% /sys/fs/cgroup
/dev/sda1                               256M   34M  223M  14% /boot/efi
cgfs                                     100K   0 100K   0% /run/cgmanager/fs
tmpfs                                    1.6G   76K  1.6G   1% /run/user/1000
/dev/sda4                               215G  187G   29G  87% /media/osho/Local Disk
/dev/sda3                               118G   86G   33G  73% /media/osho/O5
/dev/sda6                               559G  332G  228G  60% /media/osho/EEEC94B3EC947793
192.168.43.121:/var/lib/libvirt/images  59G   7.6G   48G  14% /var/lib/libvirt/images
root@osho-GL552VW:/home/osho#
```

13.) Checking daemon activity of libvirtd

```
root@cobalt: /home/tusharpahuja
root@cobalt:/home/tusharpahuja# systemctl status libvirtd
● libvirtd.service - Virtualization daemon
   Loaded: loaded (/lib/systemd/system/libvirtd.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2018-03-29 16:25:31 IST; 2h 40min ago
     Docs: man:libvirtd(8)
           http://libvirt.org
   Main PID: 4427 (libvirtd)
   CGroup: /system.slice/libvirtd.service
           └─4427 /usr/sbin/libvirtd
             └─5002 /usr/sbin/dnsmasq --conf-file=/var/lib/libvirt/dnsmasq/default
              └─5003 /usr/sbin/dnsmasq --conf-file=/var/lib/libvirt/dnsmasq/default

Mar 29 16:47:45 cobalt dnsmasq-dhcp[5002]: DHCPACK(virbr0) 192.168.122.115 52:54:00:12:34:56
Mar 29 17:51:36 cobalt dnsmasq[5002]: reading /etc/resolv.conf
Mar 29 17:51:36 cobalt dnsmasq[5002]: using nameserver 127.0.1.1#53
Mar 29 18:42:15 cobalt dnsmasq-dhcp[5002]: DHCPDISCOVER(virbr0) 52:54:00:12:34:56
Mar 29 18:42:15 cobalt dnsmasq-dhcp[5002]: DHCPOFFER(virbr0) 192.168.122.26 52:54:00:12:34:56
Mar 29 18:42:15 cobalt dnsmasq-dhcp[5002]: DHCPREQUEST(virbr0) 192.168.122.26 52:54:00:12:34:56
Mar 29 18:42:15 cobalt dnsmasq-dhcp[5002]: DHCPACK(virbr0) 192.168.122.26 52:54:00:12:34:56
Mar 29 18:46:27 cobalt libvirtd[4427]: libvirt version: 1.3.1, package: 1ubuntu1
Mar 29 18:46:27 cobalt libvirtd[4427]: hostname: cobalt
Mar 29 18:46:27 cobalt libvirtd[4427]: End of file while reading data: Input/output error
lines 1-21/21 (END)
```

14.) Checking daemon activity of ssh server & completion of ssh:

```
root@cobalt: /home/tusharpahuja
root@cobalt:/home/tusharpahuja# systemctl start sshd
root@cobalt:/home/tusharpahuja# systemctl enable sshd
Failed to execute operation: Too many levels of symbolic links
root@cobalt:/home/tusharpahuja# systemctl status sshd
● ssh.service - OpenSSH Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2018-03-29 19:14:01 IST; 1h 10min ago
     Process: 12579 ExecReload=/bin/kill -HUP $MAINPID (code=exited, status=0/SUCCESS)
   Main PID: 11261 (sshd)
   CGroup: /system.slice/ssh.service
           └─11261 /usr/sbin/sshd -D

Mar 29 20:09:07 cobalt sshd[11261]: Received SIGHUP; restarting.
Mar 29 20:09:07 cobalt systemd[1]: Reloaded OpenSSH Secure Shell server.
Mar 29 20:09:07 cobalt sshd[11261]: Server listening on 0.0.0.0 port 22.
Mar 29 20:09:07 cobalt sshd[11261]: Server listening on :: port 22.
Mar 29 20:09:07 cobalt systemd[1]: Reloading OpenSSH Secure Shell server.
Mar 29 20:09:07 cobalt sshd[11261]: Received SIGHUP; restarting.
Mar 29 20:09:07 cobalt systemd[1]: Reloaded OpenSSH Secure Shell server.
Mar 29 20:09:07 cobalt sshd[11261]: Server listening on 0.0.0.0 port 22.
Mar 29 20:09:07 cobalt sshd[11261]: Server listening on :: port 22.
Mar 29 20:23:40 cobalt systemd[1]: Started OpenSSH Secure Shell server.
lines 1-18/18 (END)
```

```

root@cobalt: /home/tusharpahuja
root@cobalt:/home/tusharpahuja# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
/root/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:hLr8yKL0Mjed6HabbXuEBiqYuIW4wFExpkwzEc5+0fo root@cobalt
The key's randomart image is:
+---[RSA 2048]----+
| *o+.              |
| = =.o .           |
| =.. .. .          |
| .o o. .           |
| *+ooo S           |
| X.o+.o            |
| +=o +E            |
| ==o*++            |
| +o=B0 .           |
+---[SHA256]-----+
root@cobalt:/home/tusharpahuja#

```

```

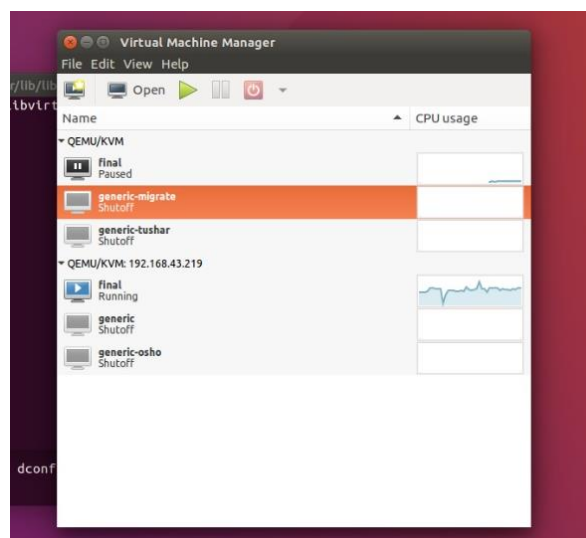
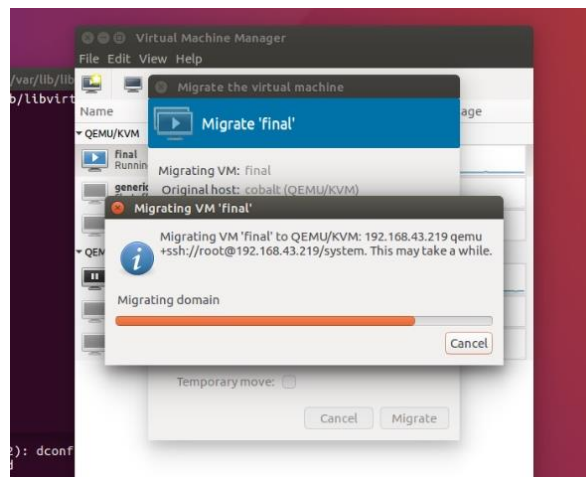
osho@osho-GL552VM:~$ exit
logout
Connection to 192.168.43.242 closed.
root@cobalt:/var/www# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
/root/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:ZUqGZA8gHGBVuNVPYf8L4HPrd/BlS0nNZ0uj5NPVIt8 root@cobalt
The key's randomart image is:
+---[RSA 2048]----+
|.++o+o+          |
|. .o o +          |
|. . . + o         |
|. o +             |
|. o .S .o|        |
|. .o. . . BB|     |
|.ooo =.Bo0|       |
|.oo.o..*.*|       |
|.oooo.o E|        |
+---[SHA256]-----+
root@cobalt:/var/www#

```

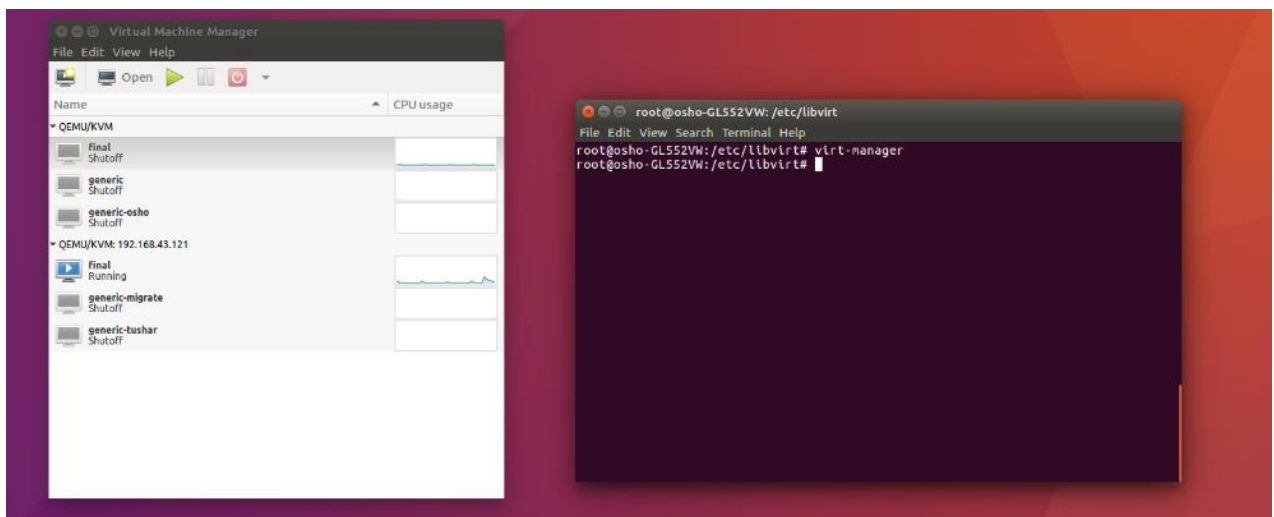
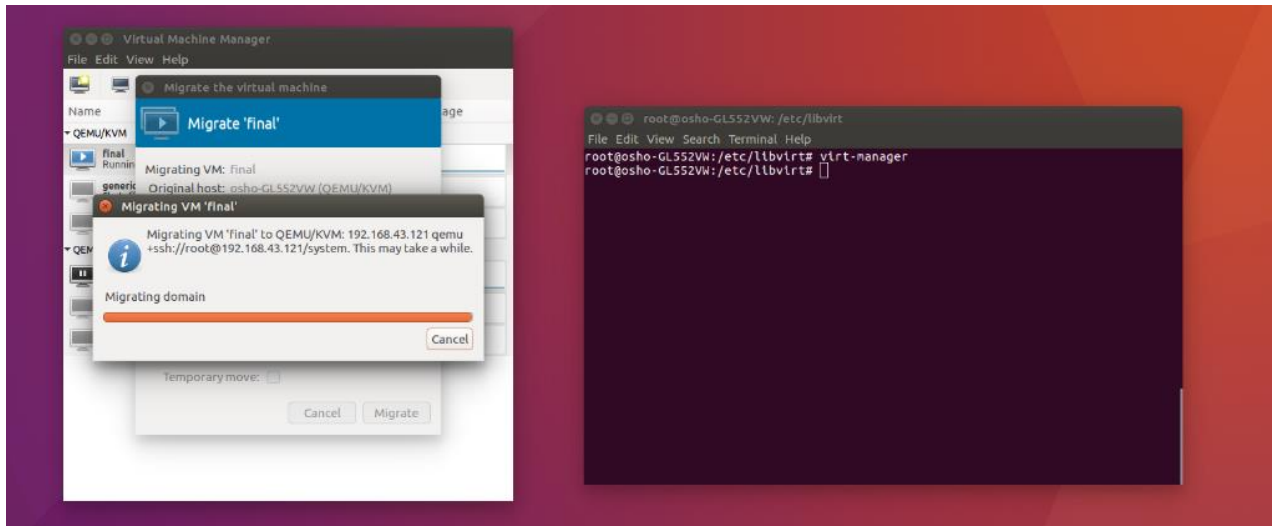
# EXPERIMENTATION RESULTS

The Live migration was carried out of the VM named “final” from 192.168.43.121 to 192.168.43.219 and vice versa after establishing connection:

This screenshot shows the 'Connect to remote host' dialog box in Virtual Machine Manager. The 'Hypervisor' is set to 'QEMU/KVM'. The 'Connect to remote host' checkbox is checked. The 'Method' is 'SSH'. The 'Username' is 'root'. The 'Hostname' is 'virtlab22'. The 'Autoconnect' checkbox is unchecked. The 'Generated URI' is 'qemu+ssh://root@virtlab22/system'. There are 'Cancel' and 'Connect' buttons at the bottom.



From 192.168.43.219 to 192.168.43.121 -----



# CONCLUSION

It was observed that in order to make the live migration work perfectly, certain daemons like libvirtd, apparmor, openssh need to be properly configured. These are implicit barriers which need to be resolved. The status of these daemons is to be checked using systemctl command as they need to be started, exited and again restarted.

This was a quick overview of the way live migration works for QEMU-KVM virtual machines. There are several details available in the archives on the QEMU list.

Following references were used:

<https://developers.redhat.com/blog/2015/03/24/live-migrating-qemu-kvm-virtual-machines/>

<http://www.admin-magazine.com/Archive/2016/33/Live-migration-of-virtual-machines-with-KVM>

[https://fedoraproject.org/wiki/QA:Testcase Live Migration using libvirt/virsh](https://fedoraproject.org/wiki/QA:Testcase_Live_Migration_using_libvirt/virsh)

[https://access.redhat.com/documentation/en-us/red\\_hat\\_enterprise\\_linux/6/html/virtualization\\_administration\\_guide/sect-virtualization-kvm live migration-migrating with virt manager](https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/6/html/virtualization_administration_guide/sect-virtualization-kvm_live_migration-migrating_with_virt_manager)

---



# APPENDIX-COMMAND HISTORY

```
root@cobalt: /home/tusharpahuja
156 systemctl start sshd
157 systemctl enable sshd
158 systemctl status sshd
159 systemctl enable ssh.service
160 systemctl enable sshd
161 systemctl enable ssh.service
162 ssh 192.168.43.219
165 ssh 192.168.43.219
166 ssh root@192.168.43.219
167 ssh root@192.168.43.219
169 ssh-keygen -t rsa
171 ssh-copy-id -i /root/.ssh/id_rsa.pub root@192.168.43.219
173 cat /root/.ssh/known_hosts
174 gedit /root/.ssh/known_hosts
176 rm /root/.ssh/known_hosts
177 ssh 192.168.43.219
179 ssh root@192.168.43.219
184 ssh-keygen -t rsa
186 ssh-copy-id -i /root/.ssh/id_rsa.pub root@192.168.43.219
188 sudo gedit /etc/ssh/sshd_config
190 gedit /etc/ssh/sshd_config
192 ssh root@192.168.43.219
193 service ssh restart
194 ssh root@192.168.43.219
195 ssh 192.168.43.219
199 ssh-copy-id -i /root/.ssh/id_rsa.pub root@192.168.43.219
202 cd /etc/ssh/
203 nano sshd_config
204 sudo service ssh restart
205 ssh-keygen
207 ssh-copy-id 192.168.43.219
212 cd /root/.ssh/
216 ssh-copy-id -i root@192.168.43.219
221 ssh osho@192.168.43.242
222 ssh osho@192.168.43.242 -p 22
223 ssh-keygen -t rsa
225 ssh-copy-id -i /root/.ssh/id_rsa.pub osho@192.168.43.242
226 ssh-keygen -t rsa
227 ssh-copy-id -i /root/.ssh/id_rsa.pub root@192.168.43.242
228 ssh root@192.168.43.242
241 ssh 192.168.43.219
264 ssh root@192.168.43.219
266 ssh-keygen -t rsa
268 ssh-copy-id -i /root/.ssh/id_rsa.pub root@192.168.43.219
285 gedit /etc/ssh
286 gedit /etc/ssh/sshd_config
310 cd /etc/ssh
311 cat sshd_config
312 nano sshd_config
313 systemctl ssh restrat
314 systemctl restart ssh
316 cat ssh_config
317 nano ssh
318 nano ssh_config
321 cd /etc/ssh/
323 cat ssh_config
324 nano ssh_config
325 cat sshd_config
326 nano sshd_config
327 service ssh restart
449 history | grep ssh
root@cobalt: /home/tusharpahuja#
```

```
root@cobalt: /home/tusharpahuja
root@cobalt:/home/tusharpahuja# history | grep virt
5 libvirt-manager
7 libvirt
9 libvirt
11 virt-manager
16 sudo mkdir /var/lib/libvirt/images -p
17 ls -la /var/lib/libvirt/images
19 sudo chown nobody:nogroup /var/lib/libvirt/images
134 virt-manager
137 systemctl status libvirtd
249 mkdir /var/lib/libvirt/images -p
250 ls -la /var/lib/libvirt/images
251 chown nobody:nogroup /var/lib/libvirt/images
278 virt-manager
282 libvirt
283 libvirtd
288 cd /etc/libvirt/
290 gedit libvirtd.conf
292 service libvirt restart
293 service libvirtd restart
294 service libvirtd stop
295 service libvirtd status
297 cd /etc/libvirt-bin/
298 cd /etc/libvirt-bin/
300 cd /etc/libvirt-bin/
301 cd /etc/default/libvirt-bin/
302 cd /etc/default/libvirt-bin
304 service libvirtd restart
305 virt-manager
306 service libvirtd status
308 virt-manager
329 virt-manager
333 cd /etc/libvirt
334 cd /etc/libvirt
336 cat libvirt.conf
340 virt-manager
345 virt-manager
351 systemctl status libvirtd
352 systemctl restart libvirtd
353 systemctl status libvirtd
370 virt-manager
373 history | grep libvirtd
375 service libvirtd statu
377 service libvirtd status
380 cd /etc/libvirt/
383 gedit libvirt.conf
384 gedit libvirtd.conf
396 cd /var/lib/libvirt/images
410 virt-manager
416 virt-manager
417 cd /var/lib/libvirt/images
426 virt-manager
427 history | grep virt
451 history | grep virt
root@cobalt: /home/tusharpahuja#
```

```
root@cobalt: /home/tusharpahuja
root@cobalt:/home/tusharpahuja# history | grep server
14 sudo apt install nfs-kernel-server
21 sudo systemctl restart nfs-kernel-server
36 apt-get install openssh-server
49 sudo apt-get install openssh-server
130 systemctl restart nfs-kernel-server
253 sudo systemctl restart nfs-kernel-server
453 history | grep server
root@cobalt: /home/tusharpahuja#
```

```
tusharpahuja@cobalt: ~
tusharpahuja@cobalt:~$ history | grep ssh
69 sudo apt-get install openssh-server
70 ssh root@192.168.43.219
71 ssh -p 22 root@192.168.43.210
72 sudo service ssh status
73 cd /etc/ssh/
74 sudo gedit sshd_config
75 sudo service ssh reload
96 ssh osho@192.168.43.219
100 ssh 192.168.43.219
102 ssh osho@192.168.43.219
118 cd /etc/ssh/
157 history | grep ssh
tusharpahuja@cobalt:~$
```

```
tusharpahuja@cobalt: ~
tusharpahuja@cobalt:~$ history | grep nautilus
143 nautilus
159 history | grep nautilus
tusharpahuja@cobalt:~$
```

tusharpahuja@cobalt: ~

```
tusharpahuja@cobalt:~$ history | grep libvirt
62  cd libvirt
88  sudo apt-get install qemu-kvm libvirt-bin bridge-utils virt-manager qemu-system
121 cd /etc/libvirt/qemu/
138 cd /var/lib/libvirt
161 history | grep libvirt
tusharpahuja@cobalt:~$
```

tusharpahuja@cobalt: ~

```
tusharpahuja@cobalt:~$ history | grep server
69  sudo apt-get install openssh-server
163 history | grep server
tusharpahuja@cobalt:~$ h4
h4: command not found
tusharpahuja@cobalt:~$ 4
4: command not found
tusharpahuja@cobalt:~$
```

root@cobalt: /home/tusharpahuja

```
tusharpahuja@cobalt:~$ su
Password:
root@cobalt:/home/tusharpahuja# history | grep virt
5  libvirt-manager
7  libvirt
9  libvirt
11 virt-manager
16 sudo mkdir /var/lib/libvirt/images -p
17 ls -la /var/lib/libvirt/images
19 sudo chown nobody:nogroup /var/lib/libvirt/images
134 virt-manager
137 systemctl status libvirtd
249 mkdir /var/lib/libvirt/images -p
250 ls -la /var/lib/libvirt/images
251 chown nobody:nogroup /var/lib/libvirt/images
278 virt-manager
282 libvirt
283 libvirtd
288 cd /etc/libvirt/
290 gedit libvirtd.conf
292 service libvirt restart
293 service libvirtd restart
294 service libvirtd stop
295 service libvirtd status
297 cd /etc/libvirt-bin
298 cd /etc/libvirt-bin/
300 cd /etc/libvirt-bin/
301 cd /etc/default/libvirt-bin/
302 cd /etc/default/libvirt-bin
304 service libvirtd restart
305 virt-manager
306 service libvirtd status
308 virt-manager
329 virt-manager
333 cd /etc/libvirtd
334 cd /etc/libvirt
336 cat libvirt.conf
340 virt-manager
345 virt-manager
351 systemctl status libvirtd
352 systemctl restart libvirtd
353 systemctl status libvirtd
370 virt-manager
373 history | grep libvirtd
375 service libvirtd statu
377 service libvirtd status
380 cd /etc/libvirt/
383 gedit libvirt.conf
384 gedit libvirtd.conf
396 cd /var/lib/libvirt/images
410 virt-manager
416 virt-manager
417 cd /var/lib/libvirt/images
426 virt-manager
427 history | grep virt
root@cobalt:/home/tusharpahuja#
```



```

root@cobalt: /home/tusharpahuja
root@cobalt:/home/tusharpahuja# history | grep nfs
14  sudo apt install nfs-kernel-server
21  sudo systemctl restart nfs-kernel-server
24  sudo ufw allow from any to any port nfs
30  sudo ufw allow from any to any port nfs
115 sudo ufw allow from any to any port nfs
129 systemctl restart nfs-kernel-serve
130 systemctl restart nfs-kernel-server
132 sudo ufw allow from any to any port nfs
253 sudo systemctl restart nfs-kernel-server
256 ufw allow from any to any port nfs
429 history | grep nfs
root@cobalt:/home/tusharpahuja#

```

```

root@cobalt: /home/tusharpahuja
root@cobalt:/home/tusharpahuja# history | grep conf
33  ifconfig | grep inet
62  cat sshd_config
63  gedit sshd_config
88  sudo nano sshd_config
112 ifconfig | grep inet
120 grep ^ENABLED /etc/ufw/ufw.conf
125 cat /etc/init/ufw.conf
188 sudo gedit /etc/ssh/sshd_config
190 gedit /etc/ssh/sshd_config
203 nano sshd_config
286 gedit /etc/ssh/sshd_config
290 gedit libvirtd.conf
311 cat sshd_config
312 nano sshd_config
316 cat ssh_config
318 nano ssh_config
323 cat ssh_config
324 nano ssh_config
325 cat sshd_config
326 nano sshd_config
336 cat libvirt.conf
337 cat qemu.conf
338 nano qemu.conf
347 nano qemu.conf
354 nano qemu.conf
371 ifconfig
383 gedit libvirt.conf
384 gedit libvirtd.conf
387 gedit qemu.conf
431 history | grep conf
root@cobalt:/home/tusharpahuja#

```

```

root@cobalt: /home/tusharpahuja
root@cobalt:/home/tusharpahuja# history | grep ssh
36  apt-get install openssh-server
37  ssh 192.168.43.219
39  ssh
40  ssh root@192.168.43.219
42  ssh oshog192.168.43.219
43  ssh root@192.168.43.219 -p 22
44  ssh root@192.168.43.219 -p22
46  ssh localhost
48  ssh localhost
49  sudo apt-get install openssh-server
50  sudo service ssh status
52  ssh localhost
55  ssh localhost
57  ssh 192.168.43.219
59  ssh-keygen -t rsa
60  ssh-copy-id -i ~/.ssh/id_rsa.pub root@192.168.43.219
61  cd /etc/ssh
62  cat sshd_config
63  gedit sshd_config
64  service ssh restart
66  ssh root@192.168.43.219
67  ssh 192.168.43.219
74  ssh-keygen -t dsa -f 192.168.43.219
79  ssh 192.168.43.219
80  ssh root@192.168.43.219
84  ssh root@192.168.43.219
85  ssh -p 22 root@192.168.43.219
86  ssh
87  cd /etc/ssh/
88  sudo nano sshd_config
89  sudo service ssh restart
90  ssh root@192.168.43.219
91  ssh oshog192.168.43.219
94  ssh localhost
139 systemctl status sshd
140 ssh 192.168.43.219
144 systemctl start sshd
145 ssh
146 systemctl enable ssh
147 sudo systemctl enable sshd
148 sudo systemctl enable ssh
149 apt-get install openssh
150 apt-get install ssh
152 apt-get install ssh
154 systemctl enable sshd
156 systemctl start sshd
157 systemctl enable sshd
158 systemctl status sshd
159 systemctl enable ssh.service
160 systemctl enable sshd
161 systemctl enable ssh.service
162 ssh 192.168.43.219
165 ssh 192.168.43.219
166 ssh root@192.168.43.219
167 ssh root@192.168.43.219
169 ssh-keygen -t rsa
171 ssh-copy-id -i /root/.ssh/id_rsa.pub root@192.168.43.219
173 cat /root/.ssh/known_hosts

```

```

root@cobalt: /home/tusharpahuja
root@cobalt:/home/tusharpahuja# history | grep apparmor
279 apt-get install apparmor
280 apt-get install apparmor-profiles
330 apt-get install apparmor
331 apt-get install apparmor-profiles
433 history | grep apparmor
root@cobalt:/home/tusharpahuja#

```

```

root@cobalt: /home/tusharpahuja
root@cobalt:/home/tusharpahuja# history | grep mount
260 showmount -e osho
269 showmount
270 showmount -e 192.168.43.219
274 showmount -e 192.168.43.121
346 showmount -e localhost
420 showmount -e
447 history | grep mount
root@cobalt:/home/tusharpahuja#

```