**EX.NO:1,C**
DATE:
1. Virtualization

c. Create a VM clone and attach virtual block to the cloned virtual machine and check whether it holds the data even after the release of the virtual machine.
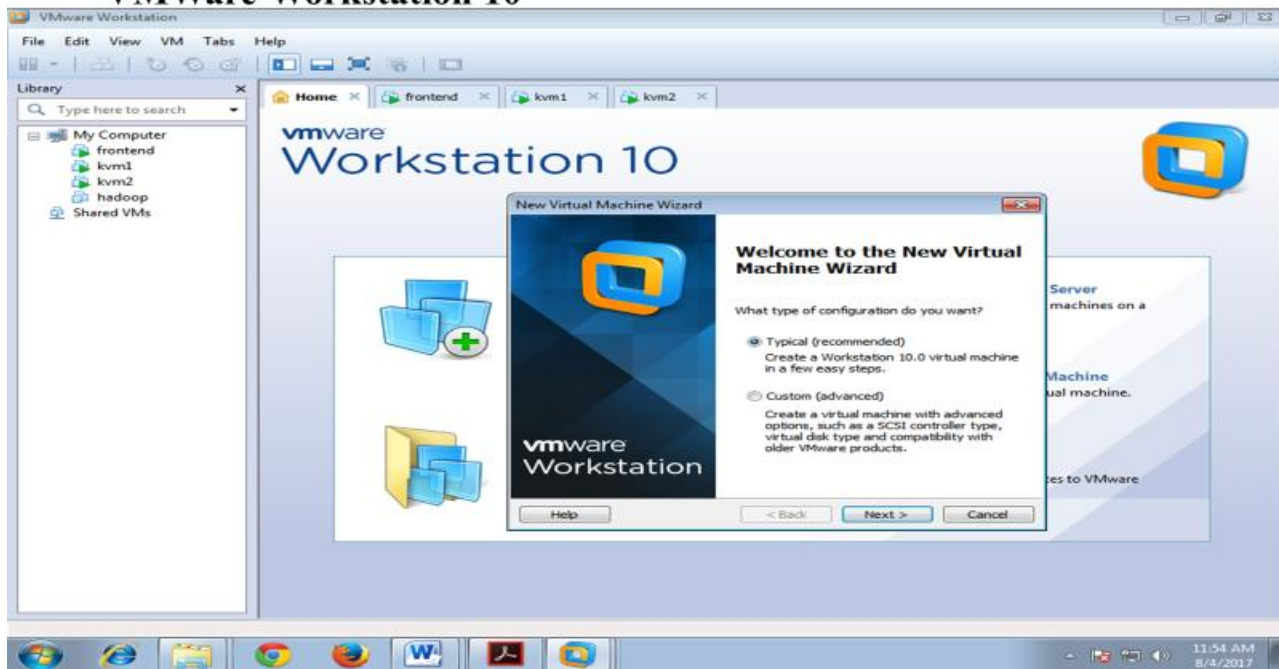
AIM:

To Install VMware workstation ,VM clone and attach virtual block to the cloned virtual machine and check whether it holds the data even after the release of the virtual machine.
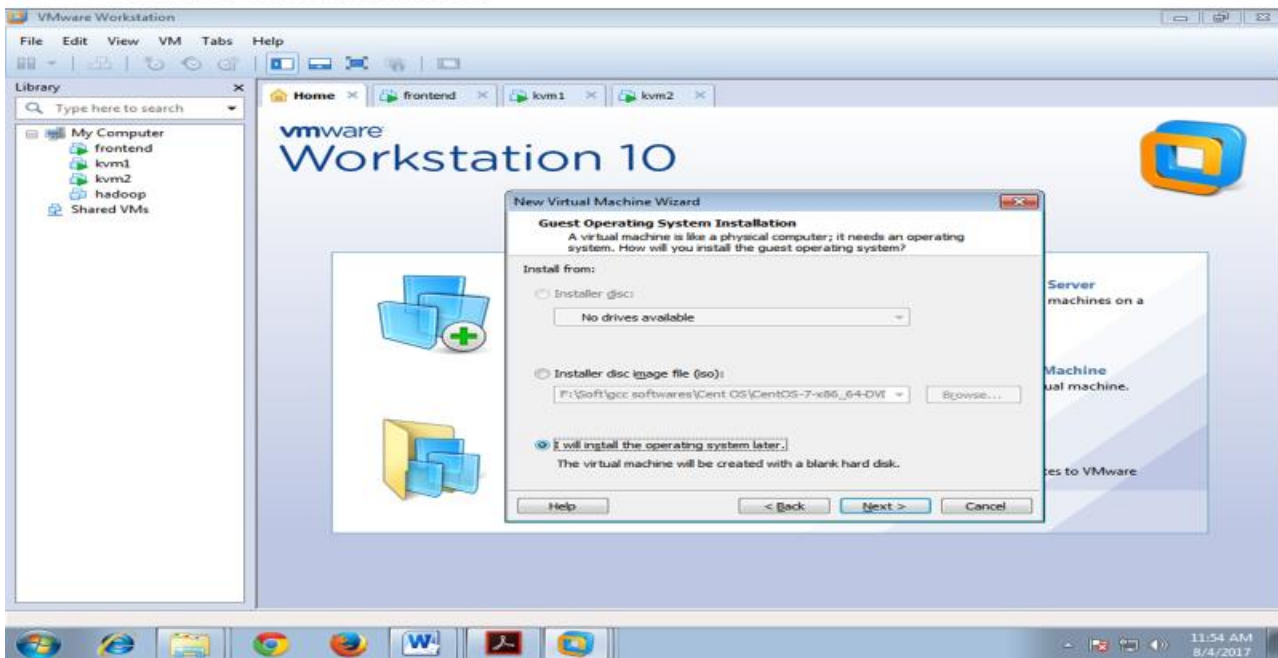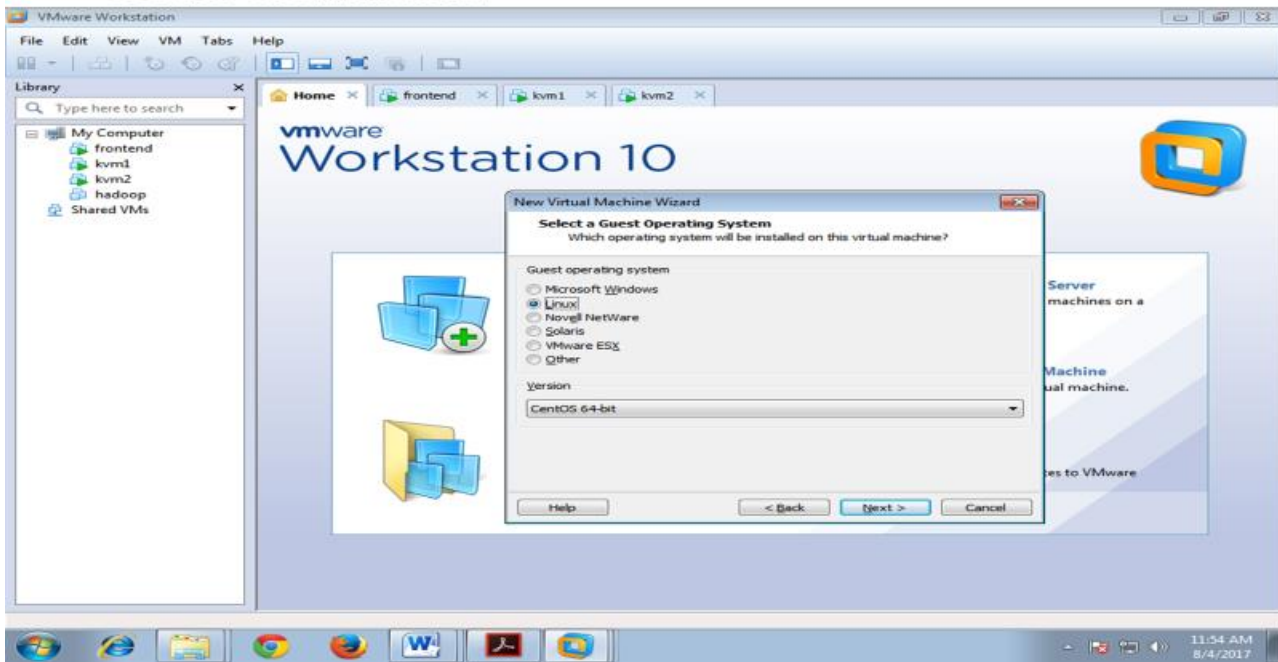
PROCEDURE:

VMWare Workstation 10
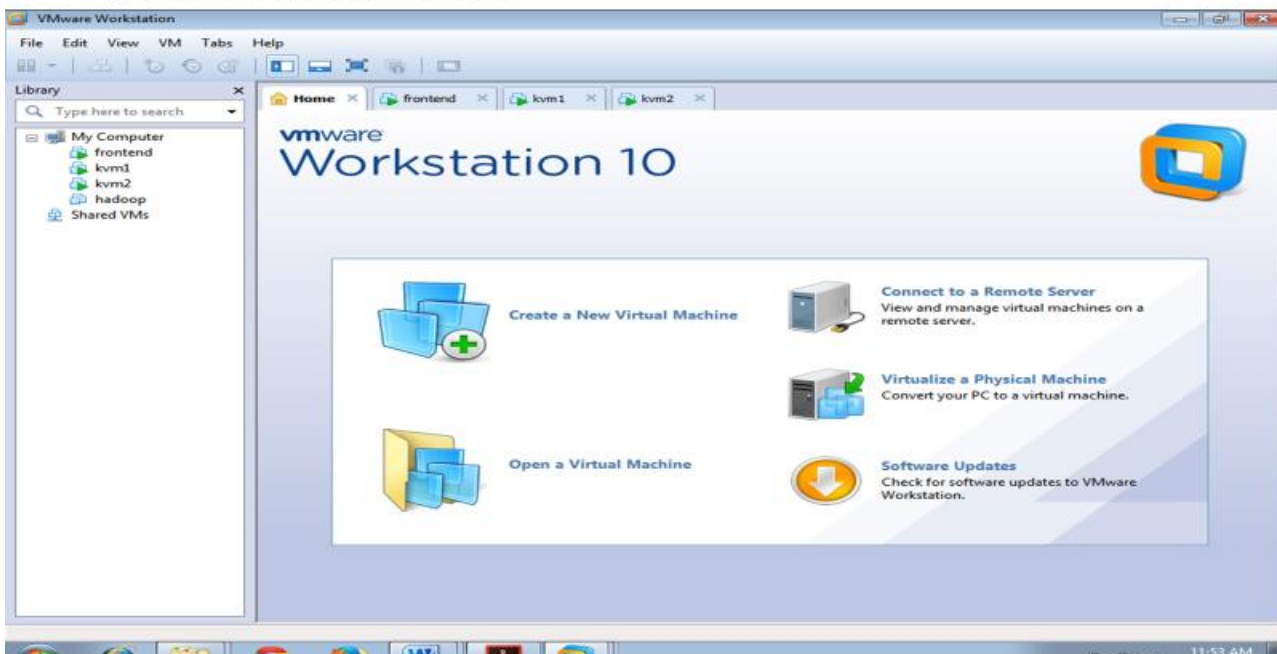
VMWare Workstation 10

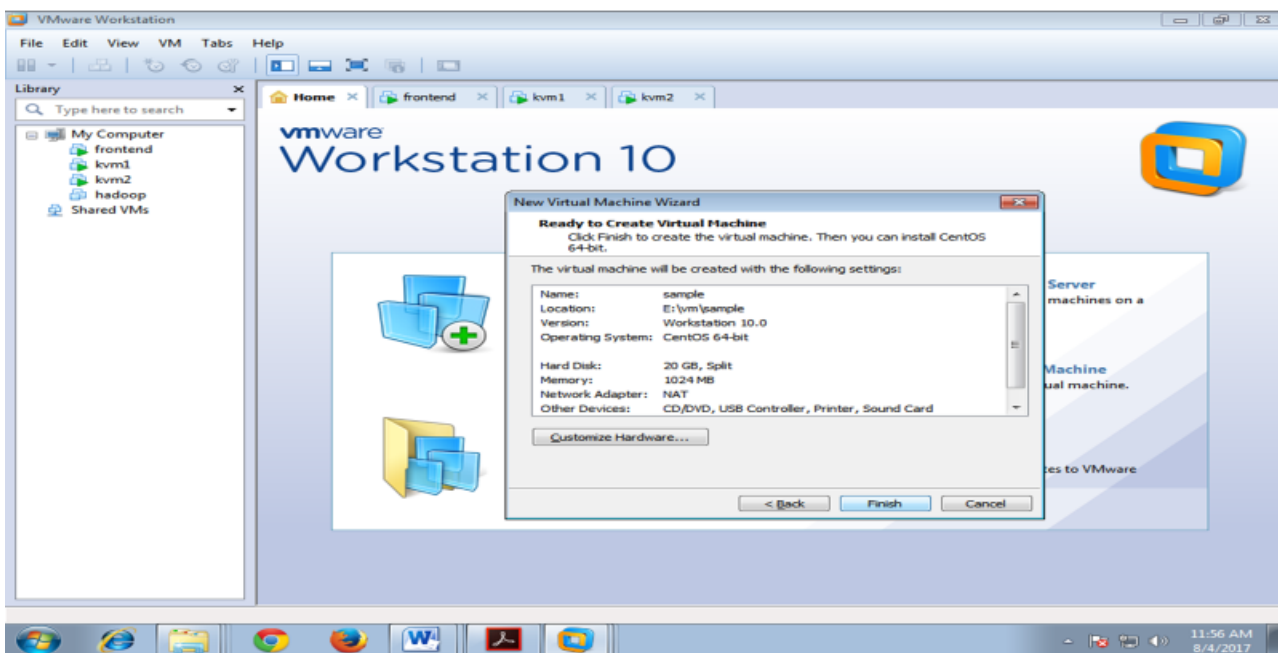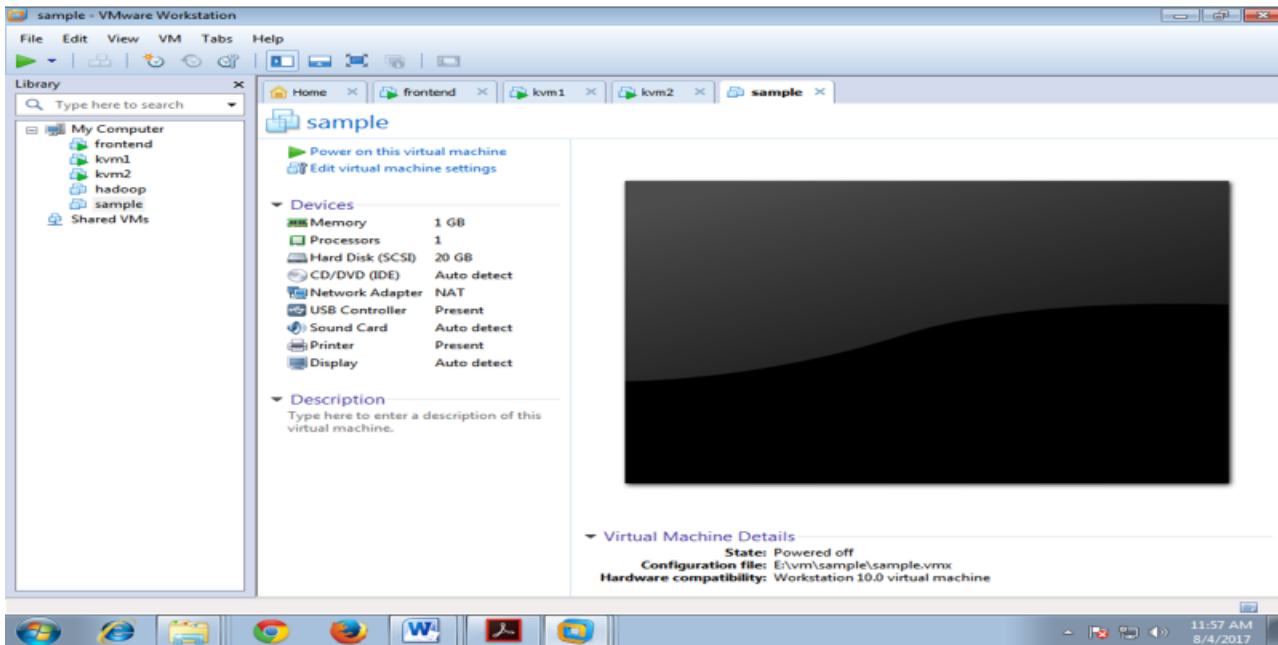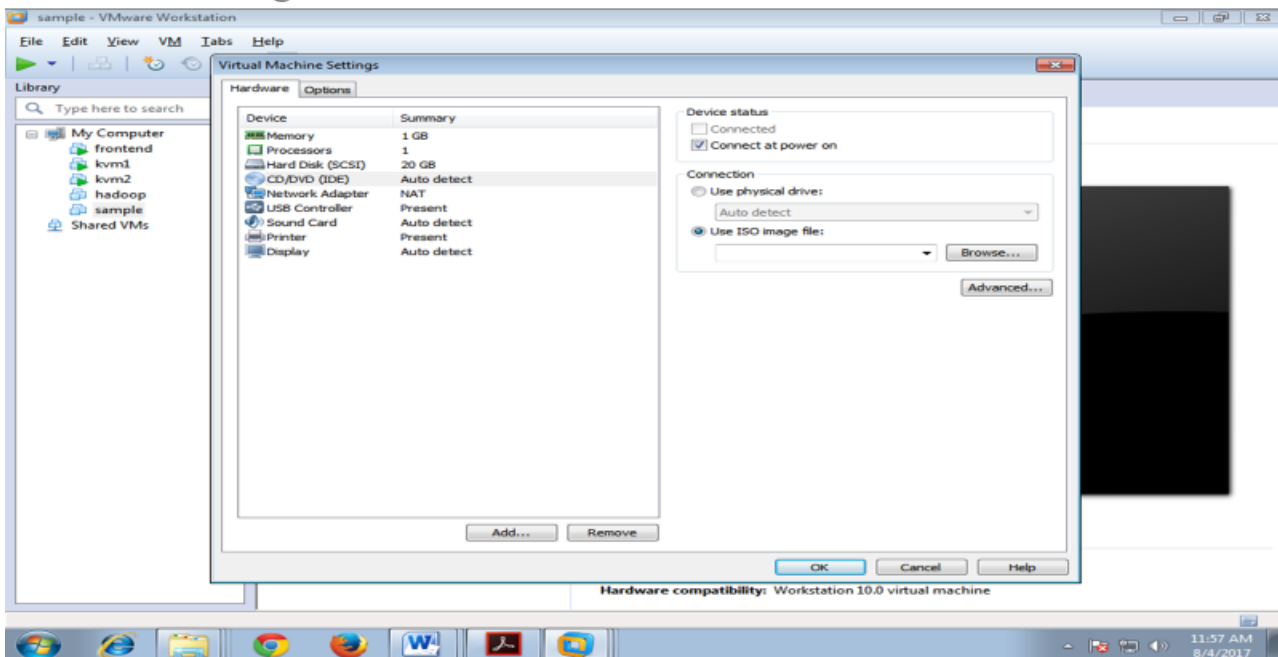## GUEST OS Installation

GUEST OS Installation

VM Creation

VM Creation

VM Settings

VM Settings

CentOS 7 Installation

CentOS 7 Installation

CentOS 7 Installation

CentOS 7 Installation

CentOS 7 Installation

CentOS 7 Installation

1.1 Host file Entry in centos 7

***** Add following entry in /etc/hosts file on kvm1 kvm2 frontend *****

#vi /etc/hosts

192.168.35.135 frontend1.cnl.com frontend1

192.168.35.136 kvm2.cnl.com kvm2

192.168.35.137 kvm1.cnl.com kvm1
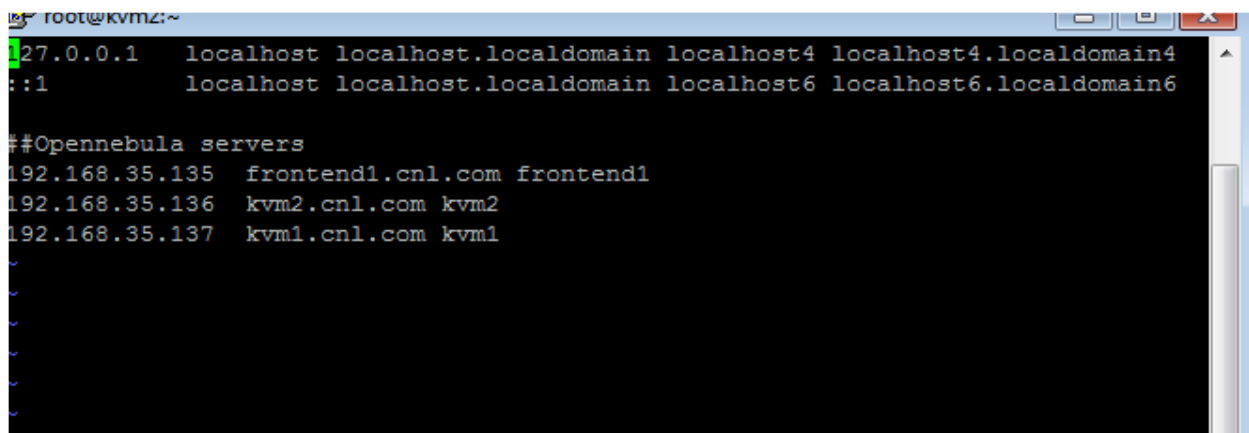
1.2 Check system is enabled VT or Not in both servers

grep -E 'svm|vmx' /proc/cpuinfo

```
Using username "root".
root@192.168.35.135's password:
Last login: Sun Jul  3 22:22:57 2016
[root@frontend1 ~]# grep -E 'svm|vmx' /proc/cpuinfo
flags           : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov
pat pse36 clflush dts mmx fxsr sse sse2 ss syscall nx pdpe1gb rdtscp lm constant
_tsc arch_perfmon pebs bts nopl xtopology tsc_reliable nonstop_tsc aperfmperf ea
gerfpu pni pclmulqdq vmx ssse3 fma cx16 pcid sse4_1 sse4_2 x2apic movbe popcnt a
es xsave avx f16c rdrand hypervisor lahf_lm arat epb xsaveopt pln pts dtherm tpr
_shadow vnmi ept vpid fsgsbase smep
[root@frontend1 ~]#
```

1.3 System must be enabled with KVM mode on both servers

lsmod | grep -i kvm

```
Using username "root".
root@192.168.35.135's password:
Last login: Sun Jul  3 22:43:36 2016 from 192.168.35.1
[root@frontend1 ~]# lsmod | grep -i kvm
kvm_intel             148081  0
kvm                   461126  1 kvm_intel
[root@frontend1 ~]#
```

1.4 Modify selinux mode on both servers

 #vi /etc/selinux/config□SELINUX=permissive

#vi /etc/sysconfig/selinux□□SELINUX=permissive

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#       enforcing - SELinux security policy is enforced.
#       permissive - SELinux prints warnings instead of enforcing.
#       disabled - No SELinux policy is loaded.
SELINUX=permissive
# SELINUXTYPE= can take one of three two values:
#       targeted - Targeted processes are protected,
#       minimum - Modification of targeted policy. Only selected processes are protected.
#       mls - Multi Level Security protection.
```

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#       enforcing - SELinux security policy is enforced.
#       permissive - SELinux prints warnings instead of enforcing.
#       disabled - No SELinux policy is loaded.
SELINUX=permissive
# SELINUXTYPE= can take one of three two values:
#       targeted - Targeted processes are protected,
#       minimum - Modification of targeted policy. Only selected processes are protected.
#       mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

1.5 Enable nfs home directory usage in selinux on both servers

#getsebool -a | grepuse_nfs_home_dirs

#setsebool -P use_nfs_home_dirs 1

```
[root@frontend1 ~]# getsebool -a | grep use_nfs_home_dirs
use_nfs_home_dirs --> off
[root@frontend1 ~]# setsebool -P use_nfs_home_dirs 1
[root@frontend1 ~]#
```

1.6 Disable firewalld in both servers

systemctl disable firewalld

systemctl stop firewalld

systemctl status firewalld

```
[root@frontend1 ~]# setsebool -P use_nfs_home_dirs 1
[root@frontend1 ~]# systemctl disable firewalld
[root@frontend1 ~]# systemctl stop firewalld
[root@frontend1 ~]# systemctl status firewalld
firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; disabled)
   Active: inactive (dead)

Jul 03 23:26:23 frontend1.cnl.com systemd[1]: Stopped firewalld - dynamic firewall daemon.
[root@frontend1 ~]#
```

1.7 Reboot all 3 servers

shutdown –r now

1.8 Check nfs SE Boolean settings only in kvm1 and kvm2 servers

systemctl status firewalld|grep -i active

****Active: inactive (dead)****

getsebool -a | grep use_nfs_home_dirs

1.9 Install EPEL repo on all the servers

yum install epel-release

Add the OpenNebula repository in all three machines:

 # cat<< EOT > /etc/yum.repos.d/opennebula.repo

[opennebula]

name=opennebula

baseurl=http://downloads.opennebula.org/repo/4.8/CentOS/7/x86_64

enabled=1

gpgcheck=0

EOT /

```
[opennebula]
name=opennebula
baseurl=http://downloads.opennebula.org/repo/4.8/CentOS/7/x86_64/
enabled=1
gpgcheck=0
```

1.10 Install the required packages in frontend1 server

 # yum -y install opennebula-server opennebula-sunstone

1.11 Install the required packages in kvm1 and kvm2 servers

 # yum -y install opennebula-node-kvm

1.12 Install the gems package and dependencies in frontend1 servers

On frontend1, now run install_gems to install all the gem dependencies (Select Cent-OS/Redhat)

 # /usr/share/one/install_gems

****lsb_release command not found. If you are using a RedHat based distribution install redhat-lsb****

****Select your distribution or press enter to continue without installing dependencies****

0. Ubuntu/Debian

1. CentOS/RedHat

2. SUSE

******PRESS 1 AND ENTER (because we are using Centos 7)******

****Press Enter wherever asked, and Y when asked to install via yum (will be asked multiple times)****

1.13 On Frontend1 server, configure and start the services

There are two main processes that must be started, the main OpenNebula daemon: oned, and the graphical user interface: sunstone.

Sunstone listens only in the loopback interface by default for security reasons. To change it edit /etc/one/sunstone-server.conf and change :host: 127.0.0.1 to :host: 0.0.0.0.****on 31th line

Now we can start the services

```
:one_xmlrpc: http://localhost:2633/RPC2

# Server Configuration
#
:host: 0.0.0.0
:port: 9869
```

 # serviceopennebula-sunstone start

Check to see service is enabled for different runlevels to start on reboot;

 # chkconfig --list 2>/dev/null|grep -i open

opennebula 0:off 1:off 2:on 3:on 4:on 5:on 6:off

opennebula-sunstone 0:off 1:off 2:on 3:on 4:on 5:on 6:off

1.14 On Frontend1 server, configure NFS service

Export /var/lib/one/ from the frontend to the worker nodes. To do so add the following to the /etc/exports file in the frontend:

#vi /etc/exports

/var/lib/one/ *(rw,sync,no_subtree_check,no_root_squash,insecure)

 # exportfs –ra On KVM1 and KVM2 servers ,check NFS

 # showmount -e frontend1

****Refresh the NFS exports by doing on FRONT end 1****

#systemctl status nfs.service

#systemctl start nfs.service

#systemctl enable nfs-server.service

****On kvm1 and kvm2****

#systemctl status nfs-client.target

#systemctl start nfs-client.target

      #systemctl enable nfs-client.target

1.15 On frontend1 server ,check NFS

#systemctl| grep -i nfs

```
[root@frontend1 ~]# systemctl |grep -i nfs
proc-fs-nfsd.mount                                    loaded active mounted   NFSD configuration filesystem
var-lib-nfs-rpc_pipefs.mount                          loaded active mounted   RPC Pipe File System
nfs-config.service                                    loaded active exited    Preprocess NFS configuration
nfs-idmapd.service                                    loaded active running   NFSv4 ID-name mapping service
nfs-mountd.service                                    loaded active running   NFS Mount Daemon
nfs-server.service                                    loaded active exited    NFS server and services
rpc-statd.service                                     loaded active running   NFS status monitor for NFSv2/3 locking.
```

1.16 On KVM1 and KVM2 servers ,check NFS

#systemctl | grep -i nfs

```
[root@kvm1 ~]# systemctl |grep -i nfs
proc-fs-nfsd.mount                                loaded active mounted   NFSD configuration filesystem
var-lib-nfs-rpc_pipefs.mount                      loaded active mounted   RPC Pipe File System
nfs-config.service                                loaded active exited    Preprocess NFS configuration
rpc-statd.service                                 loaded active running   NFS status monitor for NFSv2/3 locking.
nfs-client.target                                 loaded active active    NFS client services
[root@kvm1 ~]#
```

1.17 On KVM1 and KVM2 servers , mount /var/lib/one from frontend1

**** opennebulaoneadminhomedir****

#vi /etc/fstab

      frontend1.cnl.com:/var/lib/one/ /var/lib/one/ nfssoft,intr,rsize=8192,wsize=8192 0 0

```
#
# /etc/fstab
# Created by anaconda on Sun Jul  3 21:54:08 2016
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
UUID=034285d9-806c-4e88-9079-f70dfee763c6 /                         xfs     defaults        0 0
UUID=0e568dde-99b1-4e64-982c-c12dbf9201f8 /boot                     xfs     defaults        0 0
UUID=249f4d92-37e0-45e7-b7a5-4425a4cb0619 swap                      swap    defaults        0 0

# New DataStore
/dev/sdb1                /var/lib/one/datastores/3                   xfs     defaults        0 0
~
~
~
```

 # mount -a -t nfs

 # df -h /var/lib/one (check to see if it is mounted)

Reboot kvm1 and kvm2 to see if homedirs are mounted on reboot

 # df -h /var/lib/one (check to see if it is mounted)

1.18.On frontend1 server configure ssh public key

OpenNebula will need to SSH passwordlessly from any node (including the frontend) to any other node.

Add the following snippet to ~/.ssh/config as oneadmin so it doesn't prompt to add the keys to the known_hosts file:

 # su - oneadmin

$ cat<< EOT > ~/.ssh/config

Host *

StrictHostKeyChecking no

UserKnownHostsFile /dev/null

EOT

```
Host *
    StrictHostKeyChecking no
    UserKnownHostsFile /dev/null
```

On KVM1 and KVM2 servers ,start the services:

 # systemctl status messagebus.service

 # systemctl status libvirtd.service

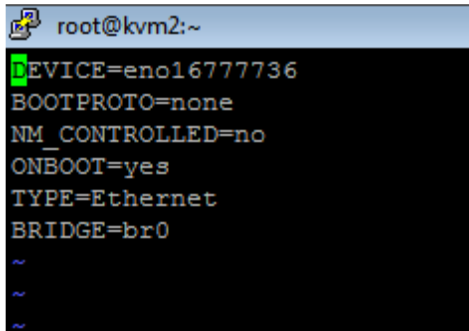**# systemctl start messagebus.service**

 # systemctl start libvirtd.service

On KVM1 and KVM2 servers ,configure the bridging for instances:

You will need to have your main interface connected to a bridge. We will do the following example with ens3 but the name of the interface may vary. An OpenNebularequirements is that the name of the bridge should be the same in all nodes.

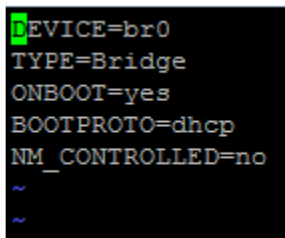**#vi /etc/sysconfig/network-scripts/ifcfg-eno16777736**

DEVICE=eno16777736

BOOTPROTO=none

NM_CONTROLLED=no

ONBOOT=yes

TYPE=Ethernet

```
root@kvm2:~
DEVICE=eno16777736
BOOTPROTO=none
NM_CONTROLLED=no
ONBOOT=yes
TYPE=Ethernet
BRIDGE=br0
~
~
~
```

****BRIDGE=br0****

#vi /etc/sysconfig/network-scripts/ifcfg-br0

DEVICE=br0

TYPE=Bridge

ONBOOT=yes

BOOTPROTO=dhcp

NM_CONTROLLED=no

```
DEVICE=br0
TYPE=Bridge
ONBOOT=yes
BOOTPROTO=dhcp
NM_CONTROLLED=no
~
~
```

****Reboot kvm1 and kvm2 to see if devices are configured****

ip route show | grep -i " br0"

****Using browser, open http://frontend1:9869****

```
←  →  C  ⌂  ⎙ http://192.168.35.135:9869/
```
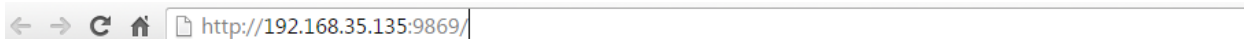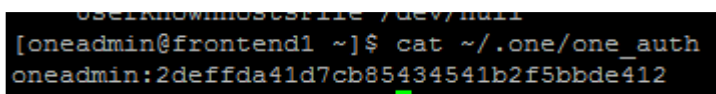
****Password is here****

****On frontend1****

#**su - oneadmin**

#**cat ~/.one/one_auth**

```
UserKnownHostsFile /dev/null
[oneadmin@frontend1 ~]$ cat ~/.one/one_auth
oneadmin:2deffda41d7cb85434541b2f5bbde412
```

****Copy that key into your browser password field****

Open Frontend, kvm1, kvm2 (Pwd: Redhat)

Goto FrontEnd ->Right Click-> Open In Terminal

[frontend@frontend Desktop]$ **su**

Password: **redhat**

[root@frontend Desktop]# **su – oneadmin**

OUTPUT:

Last login: Wed Aug 21 09:27:42 IST 2019 on pts/0

[oneadmin@frontend ~]$ **onehost list**

OUTPUT:

ID NAME	CLUSTER	RVM	ALLOCATED_CPU	ALLOCATED_MEM STAT
0 kvm1.saec.com -		0	0 / 100 (0%)	0K / 986.7M (0%) on
1 kvm2.saec.com -		0	0 / 100 (0%)	0K / 986.7M (0%) on

++

RESULT:

Thus procedure to launch VM clone and attach virtual block to the cloned virtual machine and check whether it holds the data even after the release of the virtual machine was done successfully.