



## OpenNebula R 3.4.1- Create CENTOS VM With contextualization

### Table of Contents

Creating CENTOS Desktop image file {using virsh}:	2
OpenNebula setup and VM Creation:	3
.....	3



This guide assumes that, OpenNebula is already installed and configured [For OpenNebula installation refer my guide [<http://cloudblab.files.wordpress.com/2012/05/opennebula-3-4-1-in-ubuntu-12-04-precises-single-machine-installation-for-learning-and-testing-purpose1.pdf>]

You are going to create CENTOS Desktop Virtual machines using OpenNebula .

Before we go in, I would like you to familiarize few terms used in this guide:

OneHost : OpenNebula Server machine [ip:192.168.1.174

VMHost : OpenNebula Virtual Machine Host (node) [ip: 192.168.1.95]

: A free, Java-based programming framework that supports the processing of large data sets in a clustered computing environment.

## Creating CENTOS Desktop image file {using virsh}:

1. Login to VMHost
2. Create an image file for installing CENTOS 32 bit Desktop.

```
qemu-img create -f raw /var/lib/one/centos/centos.img 8G
```

3. Create a domain creation script [/var/lib/one/template/deployment\_1]for Centos image and save it in /var/lib/one/template/ folder
4. Store the CENTOS ISO file in /var/lib/one/iso/ folder

```
<name>centos-70</name>
<memory>1048576</memory>
<os>
  <type arch='i686'>hvm</type>
  <boot dev='hd' />
  <boot dev='cdrom' />
</os>
<on_reboot>restart</on_reboot>
<on_crash>restart</on_crash>
<devices>
  <emulator>/usr/bin/kvm</emulator>
  <disk type='file' device='disk'>
    <source file='/var/lib/one/centos/centos.img' />
    <target dev='hda' />
    <driver name='qemu' type='raw' cache='default' />
  </disk>
  <disk type='file' device='cdrom'>
    <driver name='qemu' type='raw' />
    <target dev='hdc' bus='ide' />
    <readonly />
    <source file='/var/lib/one/iso/centos.iso' />
    <address type='drive' controller='0' bus='1' unit='0' />
  </disk>
  <controller type='ide' index='0'>
    <address type='pci' domain='0x0000' bus='0x00' slot='0x01' function='0x1' />
  </controller>
  <!--use one network -->
  <interface type='network'>
    <source network='default' />
  </interface>
  <graphics type='vnc' port='5970' />
</devices>
<features>
  <acpi />
</features>
```



```
</domain>
```

5. Start virtual shell by typing “virsh” on the shell prompt.
6. Start the installation of CENTOSSQ 32 bit Server image by typing below in virsh # prompt

```
virsh # create /var/lib/one/template/deployment_1
```

You will get an output like the blow:

```
Domain centos-70 created from /var/lib/one/template/deployment_1
```

7. Connect to the virtual instance using vnc, [192.168.1:174:70] and complete the installation.
  - Install only the CENTOS server
  - Use only “OpenSSH” as the additional package
  - Note the hostname as in /etc/hosts file. You may need it below.
  - Have automatic login enabled

### Update rc.local with

```
mount -t iso9660 /dev/hdb /mnt
if [ -f /mnt/context.sh ];
then
. /mnt/init.sh
fi
umount /mnt
```

That's it. We have created a image file with an update to rc.local.  
Now let's move to OpenNebula server and proceed with further setup

## OpenNebula setup and VM Creation:

1. Login to OpenNebula OneHost
2. Create two VM Templates for server and node respectively.

1. Create a Network definition template using following script  
name:ubuntu.net

```
NAME = "ubuntu-net"
TYPE = RANGED
BRIDGE = br0
NETWORK_SIZE = C
#NETWORK_ADDRESS= 192.168.2.0/24
IP_START = 192.168.1.30
IP_END = 192.168.1.49
VLAN = NO
NETWORK_MASK = 255.255.255.0
# Custom Attributes to be used in Context
GATEWAY = 192.168.1.1
DNS = 192.168.1.1
```

2. Before creating a image definition template , copy Centos image to /var/lib/one/vardatastores

```
cp centos.img /var/lib/one/var/datastores/centos.img
```



```
chown oneadmin:oneadmin /var/lib/one/var/datastores/centos.img
```

3. Create a image definition template using following script

```
/var/lib/one/template/debain.img
```

```
NAME      = "centos desktop"
SOURCE    = //var/lib/one/vardatastores/centos.img
TYPE      = OS
```

4. Create “one”images using above definitions and change permissions.

```
oneimage create debain.img -d default
```

5. Check the status using “oneimage list” command. You should get an **output** like below. The STAT should be 'rdy' instead of 'err'.

ID	USER	GROUP	NAME	SIZE	TYPE	REGTIME	PUB	PER	STAT	RVMS
10	oneadmin	oneadmin	centos_deskt	0M	OS	06/22 19:53:03	No	No	rdy	1

6. Make the images persistent [make your choice. Why?, Read further...] . You can make it persistent during image creation itself. The reason I am doing it as a separate step is, you should make persistent images only after you are through with both opennebula processes. [Be sure: you have the original image file intact. Do not delete it. Persistent images will have everything you do in an instance saved in it, provided you followed the correct “one” processes [graceful shutdown etc.]]. Anytime if you need to start everything from the beginning , you need the original image file intact. Do not even create a dummy oneimage using the original image file.

```
oneimage persistent 10
```

7. Check the status now using “oneimage list” command and check the value of “PER” column.

ID	USER	GROUP	NAME	SIZE	TYPE	REGTIME	PUB	PER	STAT	RVMS
10	oneadmin	oneadmin	centos desk	0M	OS	06/22 19:55:03	No	Yes	rdy	1

8. Create a VM Template file “debain.one” and store the following content in it

```
CONTEXT = [ hostname = centos$VMID, ip_public = "$NIC[IP, NETWORK=\"ubuntu-net\"]", username =
centos$VMID, dns = "$NETWORK[DNS, NETWORK_ID=1]", password = "password",
files = "/var/lib/one/.ssh/id_rsa.pub /var/lib/one/centos/init.sh"
]
```

```
#CAPACITY Definition
```

```
NAME=server
```

```
CPU=1
```

```
MEMORY=1024
```

```
# OS image, mapped to hda.
```

```
DISK=[ DRIVER=raw, READONLY=no, IMAGE_ID = 8, TARGET=hda, TYPE=disk ]
```

```
FEATURES=[ ACPI=yes ]
```

```
# I/O Devices Section
```

```
GRAPHICS=[ TYPE=vnc ]
```

```
#NETWORK Section:
```

```
NIC=[ IP=192.168.1.47,
```

```
network = "ubuntu-net" ]
```

```
#OS and BOOT Options Section
```



```
OS=[ ARCH=i686, BOOT=hd ]
```

```
#RAW Section  
RAW=[ TYPE=kvm ]
```

- In order to automate, let's use following shell script. The script is self explanatory. This script will take the CONTEXT information as an input.
- Create a shell script to upload context values to the Debain Desktop

**file: /var/lib/one/CENTOS/init.sh**

```
echo nameserver $DNS > /etc/resolv.conf  
chown -R $USERNAME /home/$USERNAME  
  
# replace the default hostname [name taken during image file creation] with name of Ubuntu server  
# You may change the value .node616. with the hostname specified during Ubuntu image creation.  
  
#sed -i -e 's/node616/"${HOSTNAME}"/g' /etc/hosts  
#comment line 127.0.1.1 server in /etc/hosts  
#sed -i -e 's/"127.0.1.1\ \ \ node616"/"#127.0.1.1\ \ \ node616"/g' /etc/hosts  
  
#update host file with IP address of Ubuntu server  
echo $IP_PUBLIC $HOSTNAME >> /etc/hosts  
  
#update /etc/network/interfaces file with static IP. You may change the Gateway as in your environment.  
  
#sed -i -e 's/dhcp/static/g' /etc/sysconfig/network-scripts/ifcfg-eth0  
sed -i -e 's/IPADDR/IPADDR $IP_PUBLIC/g' /etc/sysconfig/network-scripts/ifcfg-eth0  
sed -i -e 's/NETMASK/NETMASK 255.255.255.0/g' /etc/sysconfig/network-scripts/ifcfg-eth0  
#echo IPADDR $IP_PUBLIC >> /etc/sysconfig/network-scripts/ifcfg-eth0  
##echo NETMASK 255.255.255.0 >> /etc/sysconfig/network-scripts/ifcfg-eth0  
#echo gateway 192.168.1.1 >> /etc/sysconfig/network-scripts/ifcfg-eth0  
  
# update /etc/resolv.conf with dns. You may change the DNS address as in your environment.  
  
sed -i -e 's/192.168.122.1/192.168.1.1/g' /etc/resolv.conf  
  
# restart networking  
/etc/init.d/network restart
```

- Create a VM for server

```
onevm create CENTOS.one
```

- ssh to the instance. You know the IP of both server[192.168.2.50]

```
ssh <username>@192.168.1.47
```

- Check if everything as per the context section and init scripts has been executed. In short make a check for the following. Make corrections if required. The image is persistent. So whatever changes you do will be saved. You can make corrections if any to the context section and/or the init scripts for a later execution.

```
In server machine:  
cat /etc/hosts - should have server and node1 host name entries along with IPs. IP 127.0.0.1 is only for localhost.
```

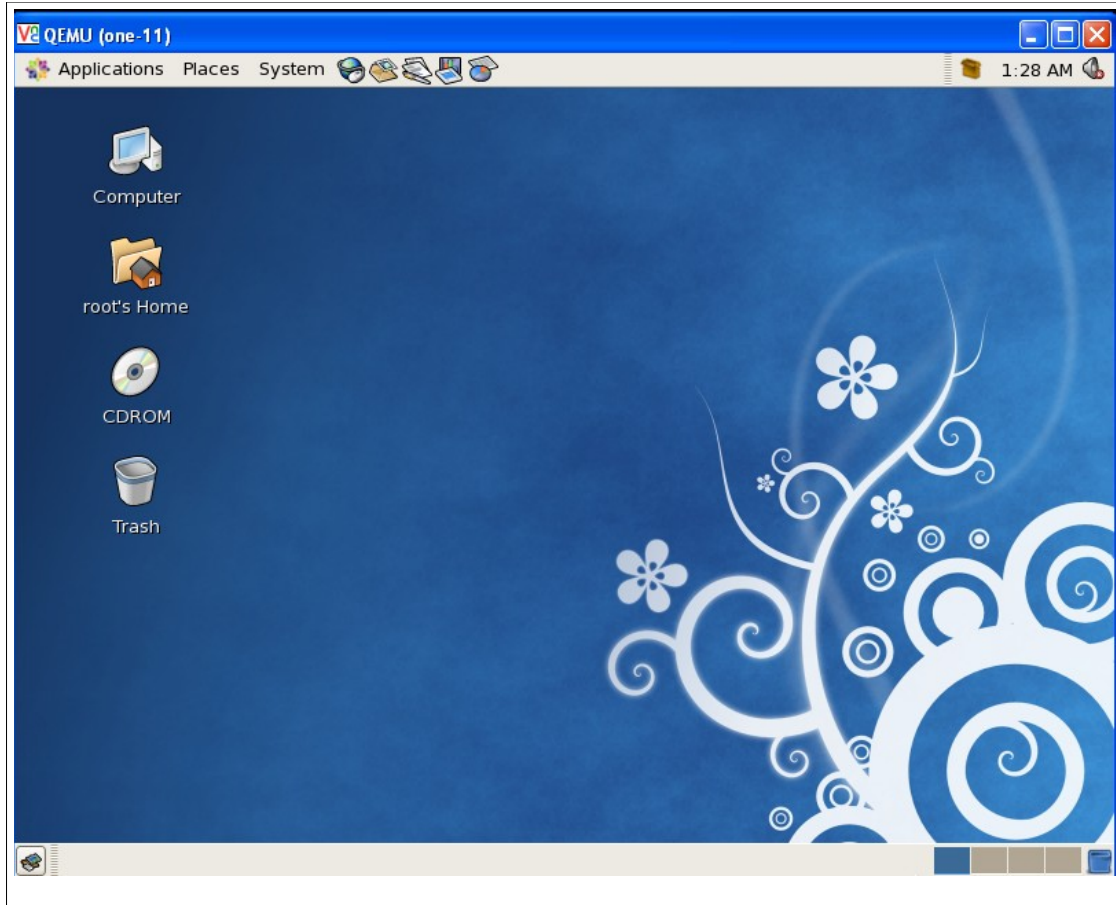


cat /etc/resolv.conf – should have correct DNS  
cat /etc/network/interfaces – should have static IP setting with gateway

```
root@centos11:~ (on westel.com)
File Edit View Terminal Tabs Help
[root@westel ~]# ./mnt/init.sh
bash: /etc/init.d/networking: No such file or directory
[root@westel ~]# ifconfig
eth0      Link encap:Ethernet  HWaddr 02:00:C0:A8:01:2F
          inet addr:192.168.1.47  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::c0ff:fea8:12f/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2609 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1409 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2930768 (2.7 MiB)  TX bytes:102662 (100.2 KiB)
          Interrupt:10 Base address:0xc000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:1165 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1165 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:3253624 (3.1 MiB)  TX bytes:3253624 (3.1 MiB)

[root@westel ~]# hostname
centos11
[root@westel ~]#
Display all 2544 possibilities? (y or n)
[root@westel ~]#
[root@westel ~]#
Display all 2544 possibilities? (y or n)
```



**If you liked this tutorial , post your comments to [cloud.b.lab@zoho.com](mailto:cloud.b.lab@zoho.com)**