

**Open Nebula 3.6.0**

**Create a Windows Server 2008 VM**

**Frontend and multiple nodes**

**INDEX**

Preparing of Front end and nodes:.....2

Prepare OpenNebula Frontend and Node.....3

    CONFIGURING OneServer.....3

    INSTALL and CONFIGURE OpenNebula in OneServer.....4

    Prepare OpenNebula Node [OneVMHost1 and oneVMHost2].....6

    CONFIGURING oneVMHost(s).....8

    ADMINSTRING OpenNebula :.....9

    A) Configure for Sunstone web interface.....10

Creating a Windows Server 2000 R2 VM .....11

By

ANIL KUMAR

[cloud.b.lab@zoho.com](mailto:cloud.b.lab@zoho.com)

[www.cloud-b-lab.com](http://www.cloud-b-lab.com)

[www.cloud-b-lab.co.in](http://www.cloud-b-lab.co.in)

Chennai and Trivandrum , India



Requirements

1. OpenNebula Front-end machine
2. OpenNebula Node : A VT Enabled machine

### **Preparing of Front end and nodes:**

3. Install Ubuntu 12.04 64 bit server in all machines
4. 2 Ubuntu VMs will be used as Nodes.
5. 1 Ubuntu VM will be prepared as OpenNebula Front end.
6. During Ubuntu installation, once you set the host name and select eth0 as Primary n/w, while DHCP process runs, press cancel button, which will enable you to enter IP address ,netmask, gateway manually.
7. All Vms to have at least 1GB RAM
8. **VM-1**

Hostname	oneVMHost1	For node1
IP eth0	192.168.1.94	
Gateway	192.168.1.1	
Domain/DNS	Example.com / 192.168.1.1	
HDD Partitioning		
Partition-1	100 MB +	
Additional pkg	Openssh server	

- **VM-2**

Hostname	OneVMHost2	For node2
IP eth0	192.168.1.97	
Gateway	192.168.1.1	
Domain/DNS	Example.com / 192.168.1.1	
HDD Partitioning		
Partition-1	100 MB+	
Username	localadmin	
Additional pkg	Openssh server	

- **VM-3**

Hostname	OneServer	Frontend
IP eth0	192.168.1.200	
Gateway	192.168.1.1	
Domain/DNS	Example.com / 192.168.1.1	
HDD Partitioning		
Partition-1	Single Partition wth whatever size [ say 60 GB ]	
Username	localadmin	

Additional pkg	Openssh server	
----------------	----------------	--

## Prepare OpenNebula Frontend and Node

1. Setup a bridge in OneServer: Install bridge-utils using below command

```
sudo apt-get install bridge-utils
```

2. Edit "/etc/network/interfaces" file to add a "bridge". Replace the contents as given in **Table VM-T2** and **restart networking**

### Sample network setup for Frontend and Node machines

Table VM-T2:

```
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet manual
auto br0
iface br0 inet static
#[Change IP for node machines]
address 192.168.1.200
netmask 255.255.255.0
network 192.168.1.0
broadcast 192.168.1.255
gateway 192.168.1.1
# dns-* options are implemented by the resolvconf package, if installed
dns-search example.com
bridge_ports eth0
bridge_fd 9
bridge_hello 2
bridge_maxage 12
bridge_stp off

sudo /etc/init.d/networking restart

/etc/resolv.conf

nameserver 192.168.1.1
search westel.com
```

## CONFIGURING OneServer

Note: [either work directly on the server console or connect to OneHost using SSH if you have a third machine with putty or any other SSH client]

1. Create a folder "/var/lib"[if doesn't exist] and create a group named "oneadmin"

```
sudo mkdir -p /var/lib/
```

```
sudo groupadd -g 10000 oneadmin
```

2. Create a user "oneadmin" , add user to group "oneadmin" and have /var/lib/one as home folder.

```
sudo useradd -u 10000 -m oneadmin -d /var/lib/one -s /bin/bash -g oneadmin
```

3. Setup password for "oneadmin" and make oneadmin owner of "/var/lib/one"

```
sudo passwd oneadmin
sudo chown -R oneadmin:oneadmin /var/lib/one
```

4. Test by logging as user "oneadmin" and exit

```
su -l oneadmin
exit
```

5. Install Network file Server [NFS]

```
sudo apt-get install nfs-kernel-server
```

6. Edit /etc/exports and add the following line to make folder /var/lib/one/

```
/var/lib/one
192.168.1.0/24(rw,fsid=0,nohide,sync,root_squash,no_subtr
ee_check)
```

7. . Restart NFS server

```
sudo /etc/init.d/nfs-kernel-server start
```

8. create a SSH key for oneadmin and disable host key checking else make all hostkeys known on the OpenNebula node.

```
su -l oneadmin
ssh-keygen
{Note - all defaults, and no passphrase.}
cat ~/.ssh/id_rsa.pub > ~/.ssh/authorized_keys
nano ~/.ssh/config
[add below two lines to SSH config file]
Host *
StrictHostKeyChecking no
exit
```

9. Exit from editor and try ssh OneVMHost, it should connect with no password

## INSTALL and CONFIGURE OpenNebula in OneServer.

1. Login to OneServer and download OpenNebula Release 3.6.0

```
su -l oneadmin
Download Latest release of OpenNebula [opennebula-3.4.1.tar] from http://downloads.opennebula.org/
```

2. Un-tar the build

```
tar xzf opennebula-3.6.0.tar.gz
```

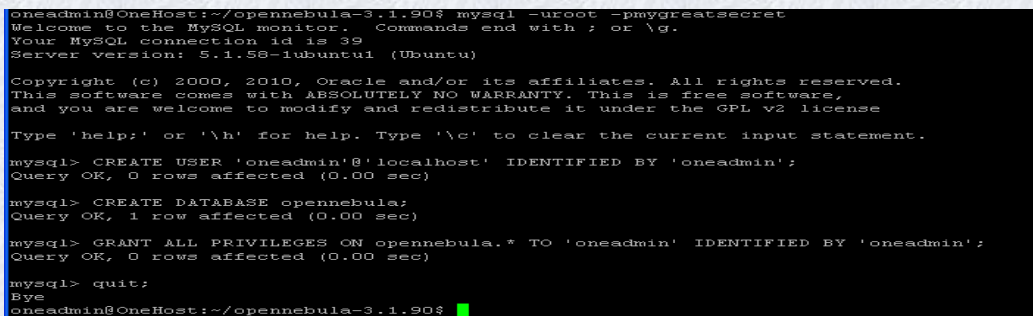
```
cd opennebula-3.6.0/
```

### 3. Before installing OpenNebula, install all pre-requisite packages

```
sudo apt-get install libcurl3 libmysqlclient18 libruby1.8 libsqlite3-ruby libsqlite3-ruby1.8 libxmlrpc-c3-dev libxmlrpc-core-c3 mysql-
common ruby ruby1.8
sudo apt-get install libxml2-dev libmysqlclient-dev libmysql++-dev libsqlite3-ruby libexpat1-dev
sudo apt-get install libc6 libgcc1 libmysqlclient18 libpassword-ruby libsequel-ruby libsqlite3-0 libssl0.9.8 libstdc++6 libxml2
sudo apt-get install ruby rubygems libmysql-ruby libsqlite3-ruby libamazonec2-ruby
sudo apt-get install libsqlite3-dev libxmlrpc-c3-dev g++ ruby libopenssl-ruby libssl-dev ruby-dev
sudo apt-get install libxml2-dev libmysqlclient-dev libmysql++-dev libsqlite3-ruby libexpat1-dev
sudo apt-get install rake rubygems libxml-parser-ruby1.8 libxslt1-dev genisoimage scons
sudo gem install nokogiri rake xmlparser
sudo apt-get install mysql-server [ set the password when asked. I normally give "mygreatsecret" as the pwd]
```

#### **configure MYSQL: <refer below screen shot in case of any doubt>**

```
mysql -uroot -pmygreatsecret
CREATE USER 'oneadmin'@'localhost' IDENTIFIED BY 'oneadmin';
CREATE DATABASE opennebula;
GRANT ALL PRIVILEGES ON opennebula.* TO 'oneadmin' IDENTIFIED BY 'oneadmin';
quit;
```



```
oneadmin@OneHost:~/opennebula-3.1.90$ mysql -uroot -pmygreatsecret
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 39
Server version: 5.1.58-1ubuntu1 (Ubuntu)

Copyright (c) 2000, 2010, Oracle and/or its affiliates. All rights reserved.
This software comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to modify and redistribute it under the GPL v2 license

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE USER 'oneadmin'@'localhost' IDENTIFIED BY 'oneadmin';
Query OK, 0 rows affected (0.00 sec)

mysql> CREATE DATABASE opennebula;
Query OK, 1 row affected (0.00 sec)

mysql> GRANT ALL PRIVILEGES ON opennebula.* TO 'oneadmin' IDENTIFIED BY 'oneadmin';
Query OK, 0 rows affected (0.00 sec)

mysql> quit;
Bye
oneadmin@OneHost:~/opennebula-3.1.90$
```

[you may drag corners to expand]

### 4. Before installing OpenNebula, configure mysql support.

```
cd ~/opennebula-3.6.0 [change your folder to opennebula source]
scons sqlite=no mysql=yes
```

### 5. Install opennebula in /var/lib/one accessible by group oneadmin and as user "oneadmin"

```
./install.sh -u oneadmin -g oneadmin -d /var/lib/one
```

### 6. Create a profile file[~/bash\_profile] to set ENVIRONMENT VARIABLES required to start and use services rendered by "one"

```
nano ~/.bash_profile
export ONE_LOCATION=/var/lib/one
export ONE_AUTH=$ONE_LOCATION/one/one_auth
export ONE_XMLRPC=http://localhost:2633/RPC2
export PATH=$ONE_LOCATION/bin:/usr/local/bin:/var/lib/gems/1.8/bin:/var/lib/gems/1.8:$PATH
```

### 7. execute the profile file and set the environment variables

```
source ~/.bash_profile
```

[Note: Anytime you open a new SSH window for OneHost, change user to "oneadmin" and source

`~/bash_profile before issuing any "one" command]`

8. Create and store OpenNebula user and password in a file. Substitute <TYPE THE PASSWORD HERE> with value

```
mkdir ~/.one
echo "oneadmin:<TYPE THE PASSWORD HERE>" > ~/.one/one_auth
```

9. Make required changes in OpenNebula configuration file `~/etc/oned.conf`

```
nano ~/etc/oned.conf

a. comment following line # Line 58 or near by [c change if your password for oneadmin is some different]
#DB = [ backend = "sqlite" ]

b. Set SQL as MYSQL-uncomment #lines 61 through 66 or near by
DB = [ backend = "mysql",
server = "localhost",
port = 0,
user = "oneadmin",
passwd = "oneadmin",
db_name = "opennebula" ]
```

10. Start Nebula

```
one start { Note: it should start with no error messages }
```

11. Now You can test OpenNebula services

```
onevm list - this command should execute with no errors. (The list will be empty for now)
```

```
oneadmin@onevmhost:~/ttylinux$ onevm list
ID USER GROUP NAME STAT CPU MEM HOSTNAME TIME
```

Now login/move back / to Terminal windows of Server 1 [192.168.1.94]  
check the contents of folder `/var/lib/one`  
You will see , all change made above stored in `/var/lib/one`.

## Prepare OpenNebula Node [OneVMHost1 and oneVMHost2]

Install Ubuntu Server 12.04 64 bit software in VMHost with following parameters.

Ubuntu 12.04 installation steps are same as OneServer, except the values as given in **Table VM-T1**

Do not set the IP address during installation. Let have DHCP. Even if you set it does not matter. We can use/change it post installation.

**Table VM-T1:**

<b>Partition</b>	You need at least one dedicated partition [e.g ID: 83 System: linux]with 100+GB of free space. Better go for automated partitioning. In case of
------------------	---

	specific partition choices, go for manual one.
<b>Hostname</b>	OneVMHost1
<b>Private N/W- bridge</b>	<b>Just setup eth0 with DHCP during install.</b> Post installation add a bridge <b>br0</b> as given below [Table: <b>Sample network setup in oneVMHost1</b> ]
IP[eth0]	DHCP
<b>Username</b>	localadmin [ <i>or have your chosen one</i> ]
<b>Additional software selection</b>	OpenSSH server alone
<b>POST INSTALLATION N/W SETUP:- BRIDGE</b>	
IP[br0]	192.168.1.94[ <i>or a different one as per your setup</i> ]
Netmask	255.255.255.0
Gateway	192.168.1.1
Domain	westell.com

**Repeat same for as many nodes as you want with different IP addresses and host name.**

Once installation is over, login to oneVMHost using server console/SSH. You should be able to ping Onevmhost and connect to internet.

Install bridge-utils using below command

```
sudo apt-get install bridge-utils
```

Edit "/etc/network/interfaces" file to add a "bridge".

**Sample network setup for oneVMHost 1 and oneVMHost2**

```
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet manual
auto br0
iface br0 inet static
#change IP address for OneVmHost2
address 192.168.1.94
netmask 255.255.255.0
network 192.168.1.0
broadcast 192.168.1.255
gateway 192.168.1.1
# dns-* options are implemented by the resolvconf package, if installed
dns-nameservers 192.168.1.1
dns-search westell.com
bridge_ports eth0
bridge_fd 9
bridge_hello 2
bridge_maxage 12
bridge_stp off

sudo /etc/init.d/networking restart
```

Add onevmhost,oneserver to the hosts file [/etc/hosts] of oneVMHost1

```
192.168.1.200 oneserver.westell.com oneserver
192.168.1.97 onevmhost2.westell.com onevmhost2
```

Add onevmhost,oneserver to the hosts file [/etc/hosts] of oneVMHost2

```
192.168.1.200 oneserver.westell.com oneserver
192.168.1.94 onevmhost1.westell.com onevmhost1
```

Add onevmhost to the hosts file [/etc/hosts] of oneserver

```
192.168.1.200 oneserver.westell.com oneserver
192.168.1.97 onevmhost1.westell.com onevmhost1
192.168.1.94 onevmhost2.westell.com onevmhost2
```

## CONFIGURING oneVMHost(s)

- Before configuring oneVMHost1 check if you are able to ping OneVMHost and the gateway
- Now try to ping 192.168.1.1 , 192.168.1.95,192.168.1.200 from oneVMHost1
- Also check the internet connectivity in VMHost by pinging *www.ubuntu.com*

Install "NFS common" to enable access to the folder "/var/lib/one" of OneVMHost. [ if run as "oneadmin" . First add oneadmin to SUDOERS file]

```
sudo apt-get update
sudo apt-get install nfs-common
```

Edit /etc/fstab and add an NFS entry for /var/lib/one. You will be using the NFS share from replication Server

```
192.168.1.200:/var/lib/one /var/lib/one nfs defaults 0 0
```

create folder structure /var/lib/one in VMHost and mount it as per "fstab" entry

```
sudo mkdir -p /var/lib/one
```

Create user "oneadmin" , group oneadmin as you did in "oneHost"

```
sudo groupadd -g 10000 oneadmin
sudo useradd -u 10000 -g oneadmin -m oneadmin -s /bin/bash
sudo usermod -d /var/lib/one oneadmin
sudo usermod -a -G oneadmin,root oneadmin
sudo passwd oneadmin
sudo chown oneadmin:oneadmin /var/lib/one
sudo mount /var/lib/one
mount
```

**Note:** You should see the below line

```
192.168.1.200:/var/lib/one on /var/lib/one type nfs (rw,vers=4,addr=192.168.1.200)
```

Install KVM hypervisor [it will take around 2 minutes or less based on your internet speed]

```
Sudo apt-get install qemu-kvm libvirt-bin ubuntu-vm-builder bridge-utils ruby
```

Libvirt needs to be configured to enable users of group "oneadmin" to manage the Vms and to allow VNC connections. Edit "/etc/libvirt/libvirtd.conf" and make the following two changes

```
unix_sock_group = "oneadmin"
(Search for string "unix_sock", if commented, uncomment this line and change the existing value to "oneadmin").
```

Edit /etc/libvirt/qemu.conf and uncomment vnc\_listen line and restart libvirt

```
vnc_listen = "0.0.0.0"
```

```
sudo service libvirt-bin restart
```



## Configure libvirt to allow access from the members of group "oneadmin"

```
sudo chown :oneadmin /var/run/libvirt/libvirt-sock
```

**Repeat same steps above for OneVmHost2.**

That's it. We have configured a host machines. Now lets add this host [onevmhost1,onevmhost2] to OpenNebula Front end.

Check if password less ssh is possible to onevmhost. You should be able to connect to onevmhost with no password , before adding it to OpenNebula front-end as a new host.

### ADMINSTRING OpenNebula :

```
onehost create onevmhost1 --im im_kvm --vm vmm_kvm --net dummy
onehost create onevmhost2 --im im_kvm --vm vmm_kvm --net dummy
```

```
oneadmin@onevmhost:~$ onehost list
```

ID	NAME	CLUSTER	RVM	TCPU	FCPU	ACPU	TMEM	FMEM	AMEM	STAT
0	onevmhost1	-	0	800	796	600	7.5G	6.7G	5.5G	on
1	onevmhost2	-	0	800	796	600	7.5G	6.7G	5.5G	on

once you register a Host check the STAT flag. It should display "on".

You may need to debug log files if Value "Err" is display for STAT.

**Note :**Common cause of "Err" flag will be either password less connection to OneVMHost is lost or VMHost is not available to OneHost.

**Hint:** Just type onehost and press enter to get all available parameters.

**Command "onehost top"** will display the output of "onehost list" continuously.

**In case of any errors just check ~/var/oned.log**

Now let's create a cluster named "1cluster"

```
onecluster create 1cluster
```

Let's get the ID of 1cluster

```
oneadmin@onevmhost:/home/localadmin$ onecluster list
```

ID	NAME	HOSTS	VNETS	DATASTORES
100	1cluster	0	0	0

Add both the hosts to 1cluster

```
onecluster addhost 100 0
```

```
onecluster addhost 100 1
```

- To obtain detailed information about the registered host use the "show" function of "onehost" command

```
onehost show <host ID> /<host_Name>
e.g onehost show 0 or onehost show onevmhost1
```

```
oneadmin@OneServer:~$ onehost show 0
HOST 0 INFORMATION
ID      : 0
NAME    : onevmhost1
CLUSTER : -
STATE   : MONITORING
IM_MAD  : im_kvm
VM_MAD  : vmm_kvm
VN_MAD  : dummy
```

```
LAST MONITORING TIME : 1338727143
```

```
HOST SHARES
MAX MEM      : 8172324
USED MEM (REAL) : 240480
USED MEM (ALLOCATED) : 0
MAX CPU      : 800
USED CPU (REAL) : 0
USED CPU (ALLOCATED) : 0
MAX DISK     : 0
USED DISK (REAL) : 0
USED DISK (ALLOCATED) : 0
RUNNING VMS  : 0
```

```
MONITORING INFORMATION
```

```
ARCH="x86_64"
CPUSPEED="1998"
FREECPU="800.0"
FREEMEMORY="7931844"
HOSTNAME="ONEVMHOST1"
HYPERVISOR="kvm"
MODELNAME="Intel(R) Xeon(R) CPU E5440 @ 2.83GHz"
NETRX="0"
NETTX="0"
TOTALCPU="800"
TOTALMEMORY="8172324"
USEDGPU="0.0"
USEDMEMORY="240480"
oneadmin@OneServer:~$
```

## A) Configure for Sunstone web interface

Ensure that you have the following installed

```
sudo apt-get install rails thin
sudo gem install json sinatra thin
```

Due to certain versions of Ruby, anytime if you get a date format error, you can remove it by

```
sudo sed -i 's/ 00:00:00.000000000Z//' /var/lib/gems/1.8/specifications/*
```

```
ln -s /usr/bin/rackup1.8 /usr/bin/rackup
```

Edit `/var/lib/one/etc/sunstone-server.conf` and make following changes

```
# Server Configuration
:host: 192.168.1.200
:port: 9869
```

### Install novnc

```
$cd /var/lib/one/share
$./install_novnc.sh -d /var/lib/one
You should get echo "Installation successful"
```

```
oneadmin@onehost:~/template$ sunstone-server start
sunstone-server started
```

Connect to SunStone interface using the url <http://192.168.1.200:9869/>

The screenshot displays the OpenNebula Sunstone interface. At the top, there's a login form with the following fields and options:

- Username:** oneadmin
- Password:** .....
- Remember me
- Login** button

Below the login form is the dashboard, which includes a sidebar menu and several monitoring widgets:

- Hosts:** Total Hosts: 1, State: ON - 1 (100%). Includes a 'Global CPU Usage' gauge and 'Used vs. Max CPU' and 'Used vs. Max Memory' bar charts.
- Virtual Machines:** Total VMs: 4, State: ACTIVE - 4 (100%). Shows 'Bandwidth - Upload' (174.1KB/s) and 'Bandwidth - Download' (55.3KB/s) gauges, along with 'Global transfer rates' and 'System Information' (Total Users: 3, Total Groups: 2).
- Clusters:** A table showing 'Allocated CPU per cluster'.

## Creating a Windows Server 2008 R2 VM

- Have a Windows Server 2008 R2 install disk as an ISO file and store it in `/var/lib/image/iso` folder to create an ISO from CD/DVD ROM, `$dd if=/dev/dvd of=win2008.iso`

Create a folder `/var/lib/images` and make `oneadmin` as the owner of the folder

```
mkdir /var/lib/image
chown -R oneadmin /var/lib/image
```

Create an empty Image file of 15 G

```
qemu-img create -f raw win2008-1.img 15G
```

Download virtio drivers and store it in `/var/lib/one/template/` folder

```
wget http://alt.fedoraproject.org/pub/alt/virtio-win/latest/images/bin/virtio-win-0.1-22.iso
```

Create a Libvirt deployment file [`/var/lib/image/win-deployment`] and store below content.

```
nano /var/lib/image/win2008-deployment
```

```

<domain type='kvm' xmlns:qemu='http://libvirt.org/schemas/domain/qemu/1.0'>
  <name>win2008</name>
  <memory>1048576</memory>
  <os>
    <type arch='x86_64'>hvm</type>
    <boot dev='cdrom' />
    <boot dev='hd' />
  </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/images/win2008-1.img' />
      <target dev='vda' bus='virtio' />
      <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/win2008.iso' />
      <address type='drive' controller='0' bus='1' unit='0' />
    </disk>
    <disk type='file' device='cdrom'>
      <driver name='qemu' type='raw' />
      <source file='/var/lib/one/template/virtio-win-0.1-22.iso' />
      <target dev='hdd' bus='ide' />
      <readonly />
    </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='5950' />
  </devices>
  <features>
    <acpi />
  </features>
</domain>

```

Instead of using virtio, if you want to go ahead with hda,  
Just replace the <disk> virto section with the following line highlighted in red.

```

<disk type='file' device='disk'>
<source file='/var/lib/one/images/win2008.img' />
<target dev='hda' />
<driver name='qemu' type='raw' cache='default' />
</disk>

```

Start virsh by typing “virsh” on the \$prompt . You will be taken to Virtual-shell  
On the virsh # prompt type the below and press enter  
`virsh # create /var/lib/image/win_deployment`

**You will get an output similar to below**

```

virsh # create /var/lib/image/win_deployment
Domain win2008 created from /var/lib/image/win_deployment

```

You will be able to monitor the Windows Installation through a VNC console. To get the VNC console PORT # , type below commands [list and then vncdisplay ] in Virsh# prompt

```

virsh # list
Id Name      State

```

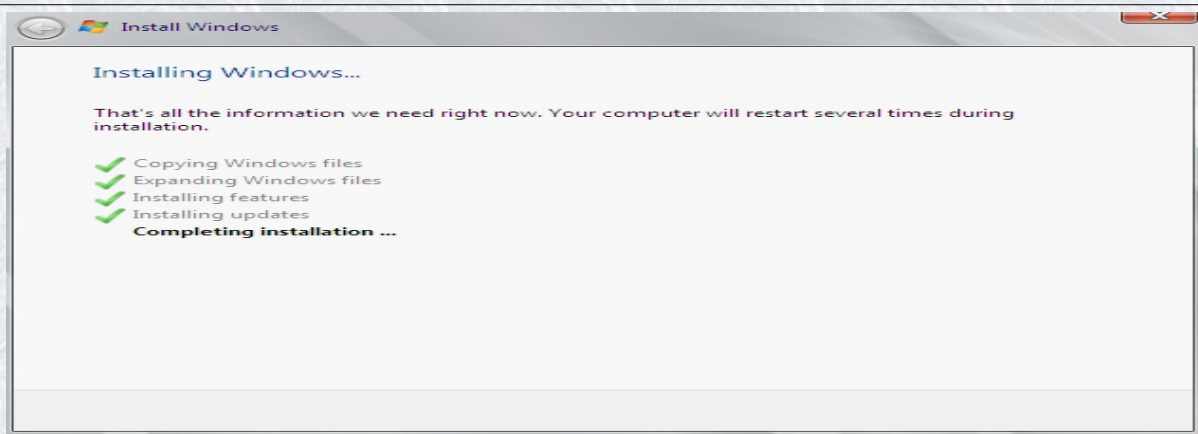
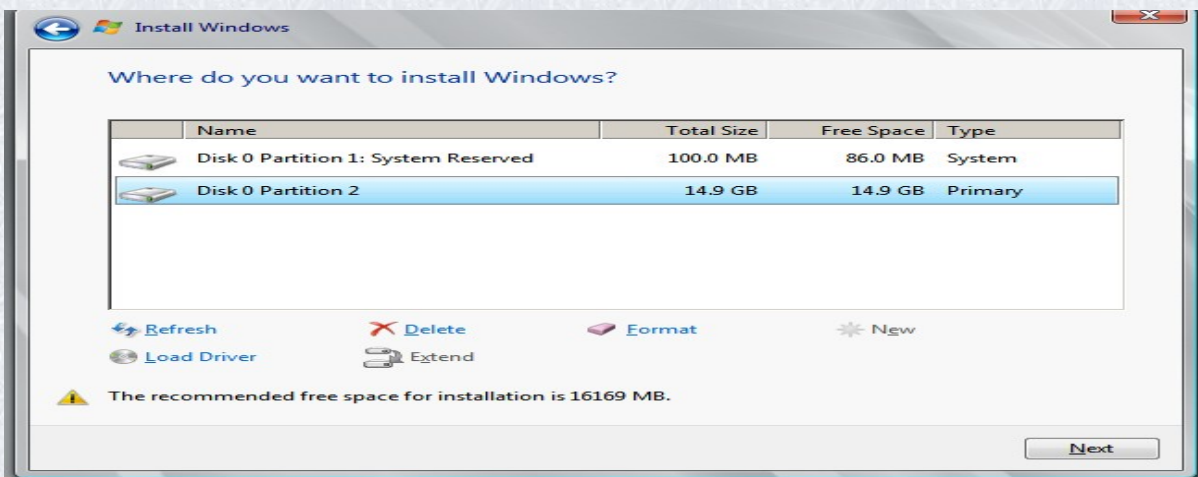
```
9 win2008      running
```

```
virsh # vncdisplay 9
:50
```

Monitor Windows Server 2008 R2 installation through VNC viewer, with the IP → 192.168.1.200:50

When asked

- When asked , install the VIRTIO drivers from CDROM
- Partition the disk by clicking "new" and go for the default partition

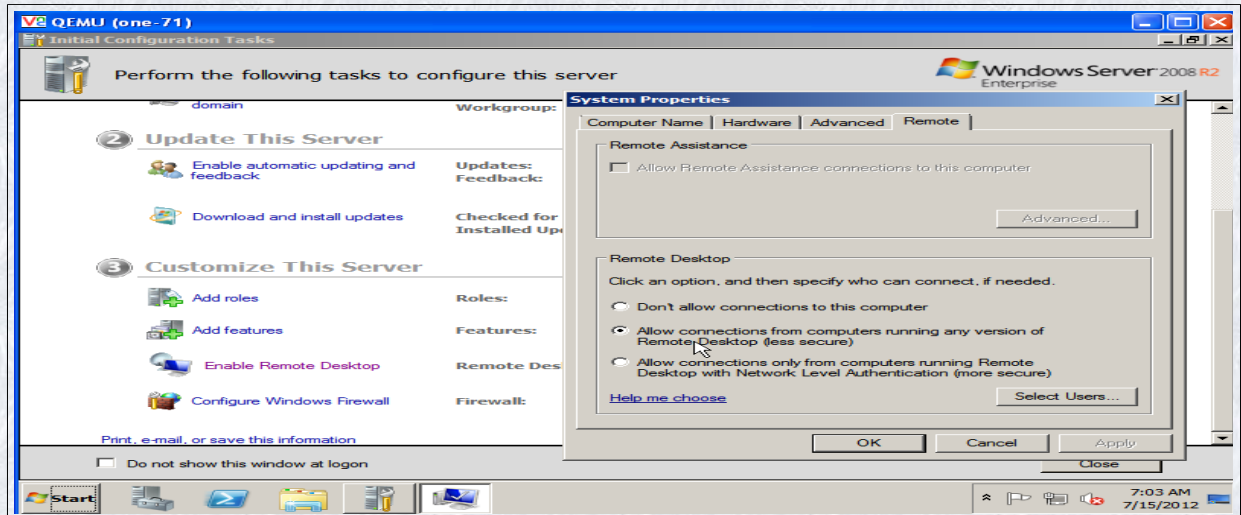


- Provide password for Administrator.

**Complete the Windows Server 2008 R2 installation**

Enable remote desktop access [Control Panel-System-Remote tab]

Disable Windows Firewall -click the icon "Windows Security alerts" in the taskbar and switch firewall to OFF state in the resulting window.



Just to test netsh, open a command window and type  
 netsh int show interface  
 This will show the network interfaces available in the system.

Create two new folders in C drive , admin and autorun [within admin]  
 mkdir admin  
 cd admin  
 mkdir autorun  
 cd autorun

Create a **StartupScript.bat** in `c:\admin\autorun` folder, and store following content in it.

#### **StartupScript.bat**

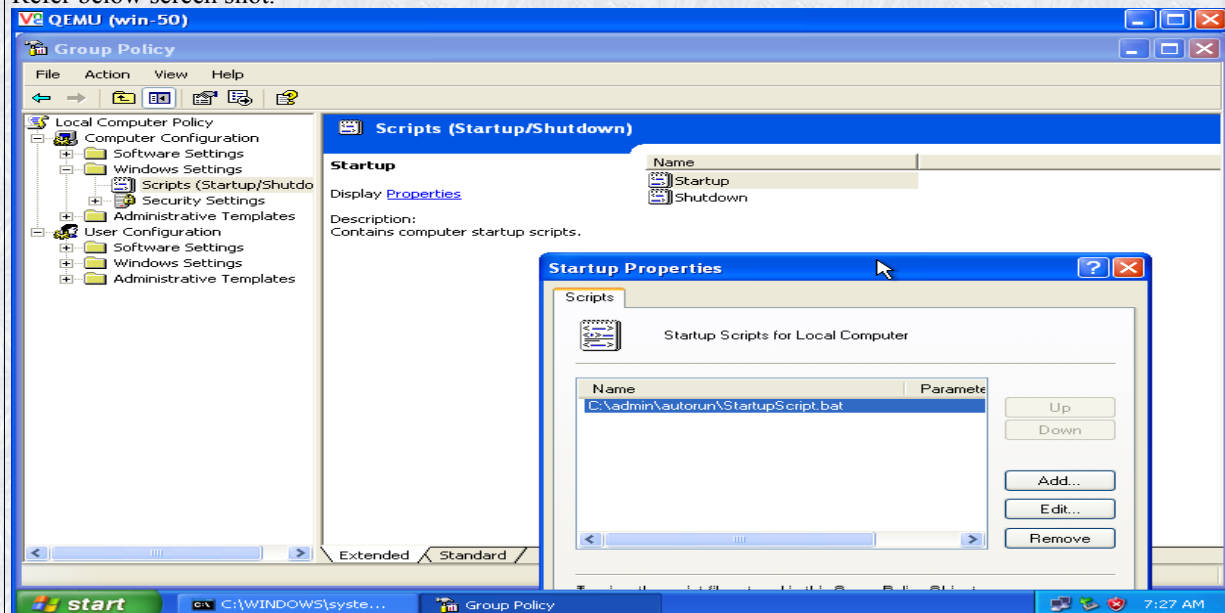
```
:: Put whatever you want here. Just leave the bottom line untouched.
c:\Windows\System32\spoolsv.exe recycle apppool "DefaultAppPool"
:: Below file will be available only when WINDOWS instance starts as a guest OS in side VM. You just specify it for now.
call d:\setcontextvals.bat
```

Currently you donot have [d:\setcontextvals.bat](#) file. But do not bother, we will soon create it with in OpenNebula front end and pass it on to the Windows instance.

Schedule **StartupScript.bat** to run on System startup:

2. Start -> Run -> "gpedit.msc"
3. Expand Computer Configuration\Windows Settings\Scripts

4. Double Click Startup
  5. Click Add...
  6. Enter c:\admin\autorun\ StartupScript.bat for the Script Name
  7. Leave Script Parameters blank
  8. Click the OK button
- Refer below screen shot.



Shutdown Windows.

Now, convert the image file from “raw” to “qcow” so that one will have no issues in understanding it. Should be run from VMHost.

```
qemu-img convert -O qcow2 win2008-1.img win2008-2.qcow2
```

On successful completion, you will have a new file “win2008-2.qcow2” created. The size will be much smaller and will be around 8G compared to the img file which was 15G

Make a copy of this file and preserve it somewhere.

Move the win2008-2.qcow2 file to /var/lib/one/var/datastores/ folder

```
mv /var/lib/image/ win-2008 Server R2-2.qcow2 /var/lib/one/var/datastores/ win2008-2.qcow2
```

Let's now move to OpenNebula.

List the available datastores:

```
oneadmin@onevmhost:~$ onedatastore list
ID NAME      CLUSTER IMAGES TYPE TM
0 system    -      0      -  shared
1 default   -      0      fs  shared
```

Create a folder “template” with in /var/lib/one to store all template files

```
mkdir /var/lib/one/template
```

Create a image definition template file /var/lib/one/template/Winxp.img and store below content in it. Save and Exit

```
NAME = "Win2008-NoPersistence"
SOURCE = /var/lib/one/var/datastores/win2008-2.qcow2
TYPE = OS
```

PUBLIC = YES

Create an image with default datastore  
 oneimage create win2008.img -d default

List the image to check the status of it. On successful creation It should show “Rdy” status.

```
oneadmin@onehost:~$ oneimage list
  ID USER  GROUP  NAME  DATASTORE  SIZE TYPE PER STAT RVMS
    15 oneadmin oneadmin Win2008-NoPers default 0M OS No rdy 0
oneadmin@onehost:~$ oneimage show 6
ID          : 15
NAME       : Win2K8R2-3-NoPersistence
USER      : oneadmin
GROUP     : oneadmin
DATASTORE  : production
TYPE      : OS
REGISTER TIME : 07/15 13:15:34
PERSISTENT : No
SOURCE     : /var/lib/one/var/datastores/win2008-2.qcow2
SIZE      : 0M
STATE     : rdy
RUNNING_VMS : 0

PERMISSIONS
OWNER     : um-
GROUP    : ---
OTHER    : ---

IMAGE TEMPLATE
DEV_PREFIX="hd"
PUBLIC="YES"
oneadmin@onehost:~/template$
```

In case of any errors during image creation, cross check the following,

1. Syntax errors in image creation template
2. Ownership of the qcow2 file should be with “oneadmin”

List the datastore again, you will see the value for “image” changes to “1” for default DS

```
oneadmin@onevmhost:~$ onedatastore list
  ID NAME  CLUSTER IMAGES TYPE TM
  0 system - 0 - shared
  1 default - 1 fs shared
```

Create a Vnet definition template /var/lib/one/template/winxp.net and store following content in it. Save exit

```
NAME = "private-win-d"
TYPE = RANGED
BRIDGE = br0
NETWORK_SIZE = C
IP_START = 192.168.1.2
IP_END = 192.168.1.25
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
```

Create a Vnet,  
 onevnet create /var/lib/one/template/winxp.net

```
oneadmin@onevmhost:/home/localadmin$ onevnet list
  ID USER  GROUP  NAME  CLUSTER  TYPE BRIDGE LEASES
    0 oneadmin oneadmin private-win-d - R br0 0
```

Create a VM template file /var/lib/one/template/win2008.one and store following content in it. Check for the IMAGE ID and VnetID.

Note: If you are using “hda” instead of virtio, replace **TARGET=vda** to **TARGET=hda**.



```

HOSTNAME = onevmhost1
#CONTEXT definition section
CONTEXT=[FILES="/var/lib/one/.ssh/id_rsa.pub /var/lib/one/images/setcontextvals.bat /var/lib/one/images/sethostname.vbs",
HOSTNAME=Win2008-$VMID,
IP_PUBLIC="$NIC[IP, NETWORK="private-win-d"]",
PASSWORD=Redhat123456,
ROOT_PUBKEY=id_rsa.pub,
USERNAME=Administrator]
#CAPACITY Definition
NAME=Windows2K8R2-NoPers
CPU=1
MEMORY=2048
# OS image, mapped to hda.
DISK=[ DRIVER=qcow2, READONLY=no, IMAGE_ID = 15, TARGET=vda, TYPE=disk ]
FEATURES=[ ACPI=yes ]
# I/O Devices Section
GRAPHICS=[ TYPE=vnc ]
#NETWORK Section:
NIC=[network = "private-win-d" ]
#OS and BOOT Options Section
OS=[ ARCH=x86_64, BOOT=hd ]
#RAW Section
RAW=[ TYPE=kvm ]

```

Create the files specified in the CONTEXT section

- Create a folder /var/lib/one/images  
`mkdir /var/lib/one/images`
- Create a file /var/lib/one/images/setcontextvals.bat and store below content in it . This batch script will extract the IP address and HOSTNAME from the context.sh file. Context.sh file will be automatically created by OpenNebula based on the CONTEXT section in the VM Template file, VM Creation process, stores the context.sh file in a Virtual CDROM (for Windows instance , assume the CDROM is “D” drive) with the VM instance.

```

@echo off
setlocal ENABLEDELAYEDEXPANSION
for /F "skip=3 eol= tokens=*" %%S in (d:\context.sh) do (set line2=%%S&goto:forend2)
:forend2
REM echo line2=%line2%
set IPADD=%line2:~11,13%
REM echo %IPADD%
for /F "skip=2 eol= tokens=*" %%S in (d:\context.sh) do (set line1=%%S&goto:forend3)
:forend3
REM echo line1=%line1%
rem set hostname=!line1:~10,8!
rem echo hostname is !hostname!

setlocal
set str=%line1:HOSTNAME=%
set str=%str:,=%
rem set hostname=%str:~-1,-1%
REM echo %hostname%
call :dequote %str%
REM echo ret=%ret%
endlocal
goto :eof

:dequote
setlocal
rem The tilde in the next line is the really important bit.
set thestring=%~1
endlocal&set ret=%thestring%
call d:\sethostname.vbs %ret%
netsh int ip set address "Local Area Connection" static %IPADD% 255.255.255.0 192.168.1.1 1
netsh int ip set dns "Local Area Connection" static 192.168.1.1 primary

```

```
goto :eof
```

- Create a file /var/lib/one/images/sethostname.vbs. Save and exit.
- This VB Script will change the hostname to the value passed by the CONTEXT section. [e.g. Win2008-72 ]

```
nHOSTNAME = Wscript.Arguments(0)
sNewName = nHOSTNAME
Set oShell = CreateObject ("WScript.shell")
sCCS = "HKLM\SYSTEM\CurrentControlSet\"
sTcpipParamsRegPath = sCCS & "Services\Tcpip\Parameters\"
nHOSTNAME = Wscript.Arguments(0)
sNewName = nHOSTNAME
Set oShell = CreateObject ("WScript.shell")
sCCS = "HKLM\SYSTEM\CurrentControlSet\"
sTcpipParamsRegPath = sCCS & "Services\Tcpip\Parameters\"
sCompNameRegPath = sCCS & "Control\ComputerName\"
With oShell
.RegDelete sTcpipParamsRegPath & "Hostname"
.RegDelete sTcpipParamsRegPath & "NV Hostname"
.RegWrite sCompNameRegPath & "ComputerName\ComputerName", sNewName
.RegWrite sCompNameRegPath & "ActiveComputerName\ComputerName", sNewName
.RegWrite sTcpipParamsRegPath & "Hostname", sNewName
.RegWrite sTcpipParamsRegPath & "NV Hostname", sNewName
End With ' oShell
rem Dim objShell
rem Set objShell = WScript.CreateObject("WScript.Shell")
rem objShell.Run "C:\WINDOWS\system32\shutdown.exe -r -t 0"
```

Now let's add our VNET [ID 0] to the cluster  
 onecluster addvnet 100 0

Add the default datastore to the 1cluster  
 onecluster adddatastore 100 1

Now list the 1cluster again  
 oneadmin@onevmhost:/home/localadmin\$ onecluster list

ID	NAME	HOSTS	VNETS	DATASTORES
100	1cluster	2	1	1

List the VMs running currently  
 oneadmin@onevmhost:/home/localadmin\$ onevm list

ID	USER	GROUP	NAME	STAT	CPU	MEM	HOSTNAME	TIME
None:								

Store/preserve the VM template in OpenNebula using onetemplate command  
 onetemplate create ~/template/win2008.one

Test the creation using List and show commands  
 oneadmin@onevmhost:~\$ onetemplate list

ID	USER	GROUP	NAME	REGTIME
0	oneadmin	oneadmin	Windows2K8R2-No	05/28 09:50:44

Instantiate the VM using onetemplate command  
 onetemplate instantiate 0 or  
 onevm create win2008.one

Monitor the progress of VM creation using onevm top command  
 onevm top

ID	USER	GROUP	NAME	STAT	CPU	MEM	HOSTNAME	TIME
72	oneadmin	oneadmin	Windows2K8R2-No	runn	23	2G	onehost	0d 01h00

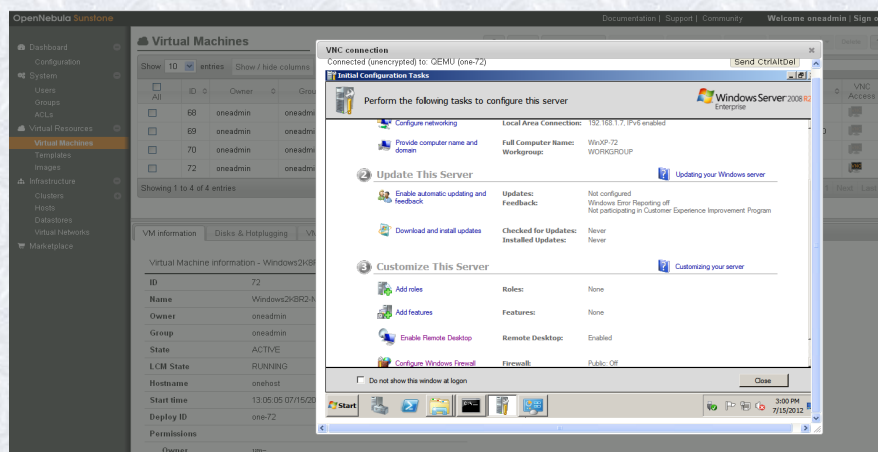
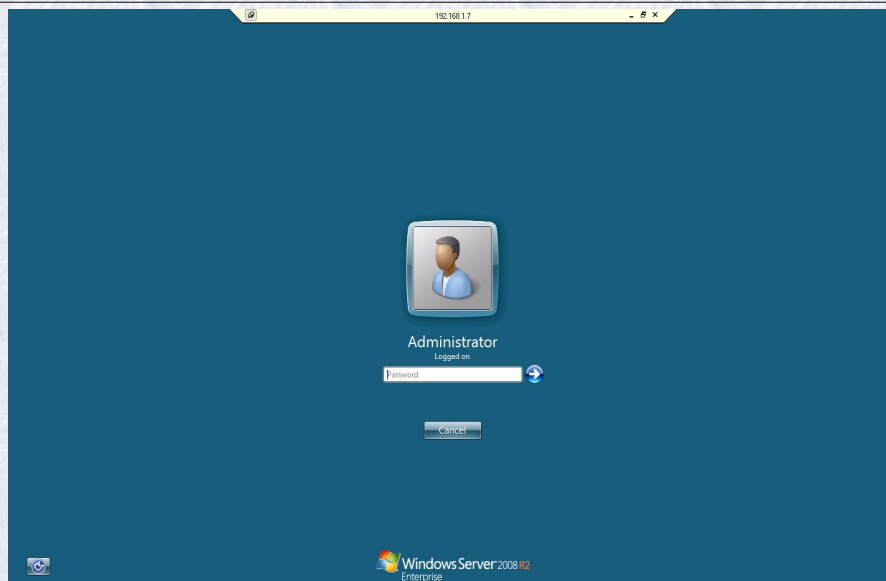
Get the IP address and PORT of the Running VM

```
oneadmin@onehost:~$ onevm show 72 | grep IP
IP_PUBLIC="192.168.1.7",
IP="192.168.1.7",
oneadmin@onehost:~$ onevm show 72 | grep PORT
PORT="5972",
```

Test by pinging the IP 192.168.1.7. It will Ping. If it does not Ping, Check through VNC console that whether you have disabled the Windows Firewall. Disable it.

Connect to Windows instance through Remote desktop.

Check to see if the IP address is set to 192.168.1.7 and hostname is changed to WinXP-39.



You can also try Migration, live migration etc with two nodes. [refer my tutorial posted on OpenNebula wiki or wordpress]

Using onevm deploy command, you can deploy the VM on a specific host like with onevmhost1 or onevmHost2

That's it. You have successfully created a Windows2008 Server R2 and set the IP address and hostname as passed by OpenNebula CONTEXT section

**If you liked this tutorial just put a comment to [cloud.b.lab@zoho.com](mailto:cloud.b.lab@zoho.com)**