[2CEIT603: CLOUD COMPUTING]

Practical: 7

AIM- Configuration of Private cloud (OpenNebulla).

Submitted by: 21012021001_Adeshara Brijesh

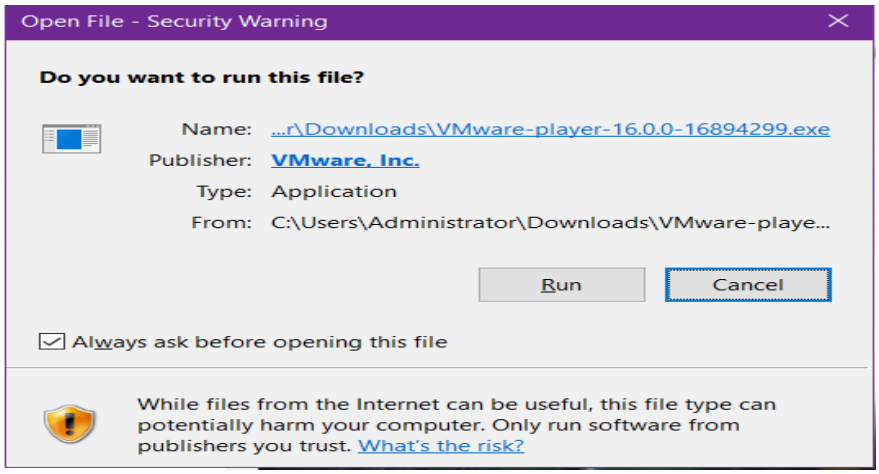
Ganpat University
॥ विद्यया समाजोत्कर्षः ॥
U.V. Patel College of Engineering

Department of Computer Engineering/Information Technology

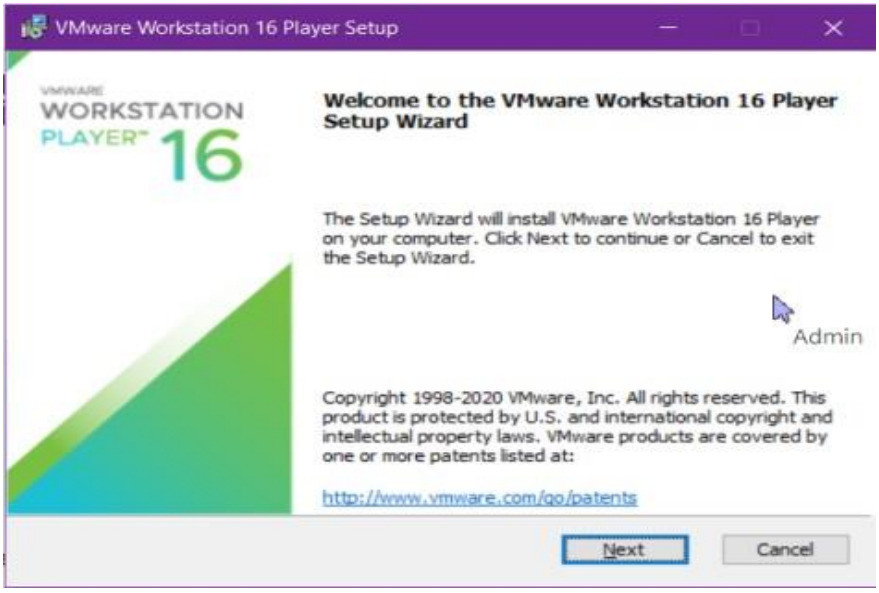
Installation of Linux.

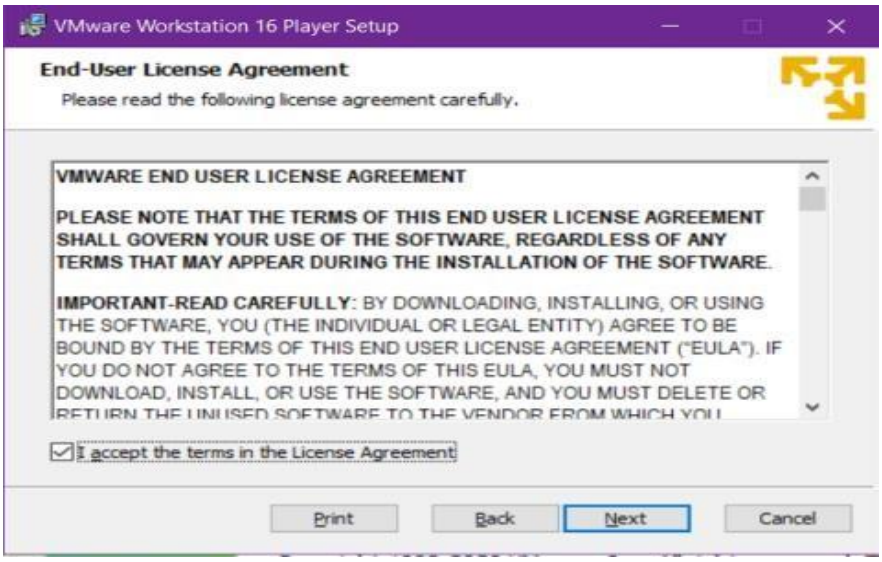
- Download ISO image file of Ubuntu 18.04 or more.
- https://releases.ubuntu.com/18.04/

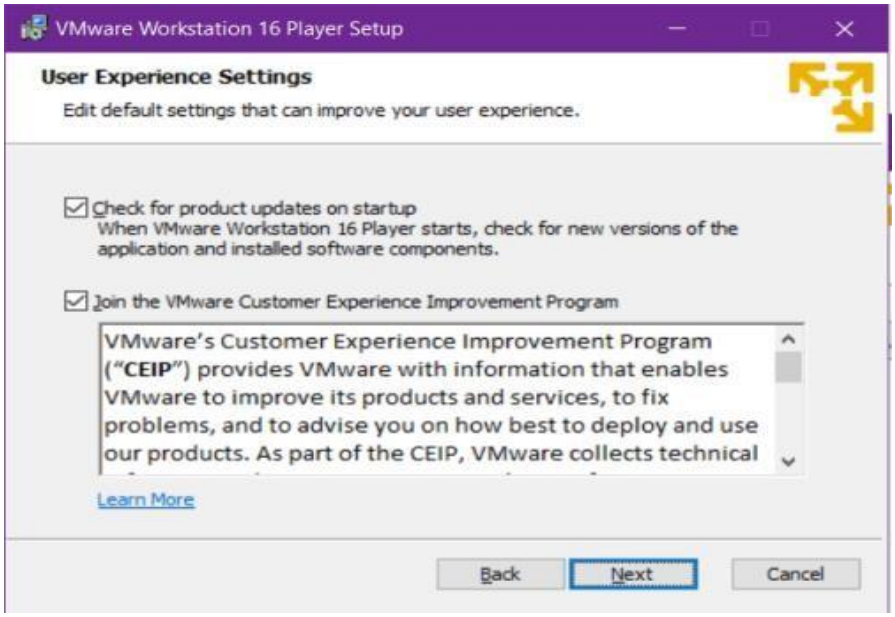
- **Installation of VMware.**
- Step 1: Download VMware software according system requirement.
- Step 2: Run file by double clicking on exe file.

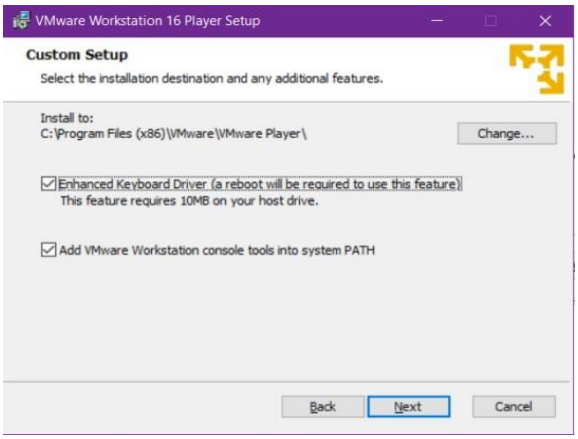


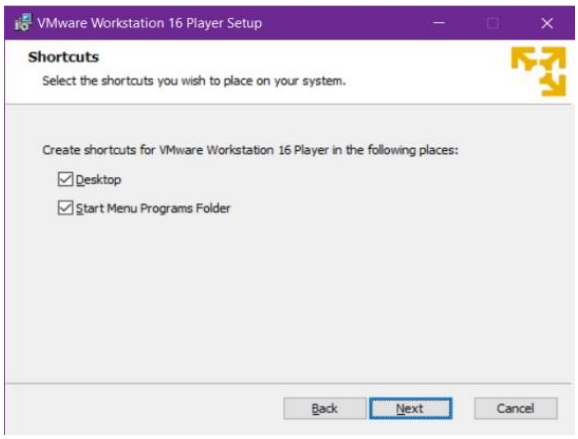
- Step 3: Click on run and setup all environment you needed.



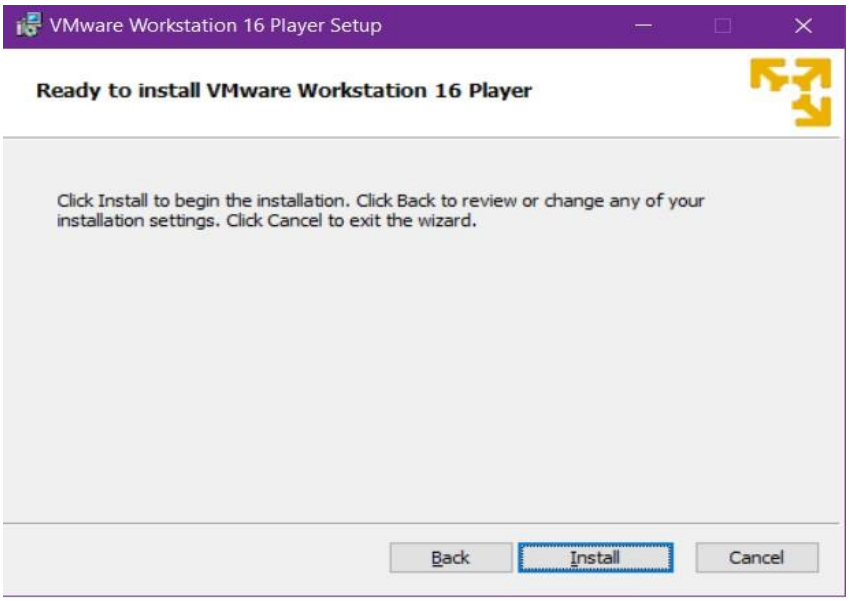

VMware Workstation 16 Player Setup
End-User License Agreement
Please read the following license agreement carefully.
VMWARE END USER LICENSE AGREEMENT
PLEASE NOTE THAT THE TERMS OF THIS END USER LICENSE AGREEMENT SHALL GOVERN YOUR USE OF THE SOFTWARE, REGARDLESS OF ANY TERMS THAT MAY APPEAR DURING THE INSTALLATION OF THE SOFTWARE.
IMPORTANT-READ CAREFULLY: BY DOWNLOADING, INSTALLING, OR USING THE SOFTWARE, YOU (THE INDIVIDUAL OR LEGAL ENTITY) AGREE TO BE BOUND BY THE TERMS OF THIS END USER LICENSE AGREEMENT ("EULA"). IF YOU DO NOT AGREE TO THE TERMS OF THIS EULA, YOU MUST NOT DOWNLOAD, INSTALL, OR USE THE SOFTWARE, AND YOU MUST DELETE OR
I accept the terms in the License Agreement
Print
Back
Next
Cancel


VMware Workstation 16 Player Setup
User Experience Settings
Edit default settings that can improve your user experience.
Check for product updates on startup
When VMware Workstation 16 Player starts, check for new versions of the application and installed software components.
Join the VMware Customer Experience Improvement Program
VMware's Customer Experience Improvement Program ("CEIP") provides VMware with information that enables VMware to improve its products and services, to fix problems, and to advise you on how best to deploy and use our products. As part of the CEIP, VMware collects technical
Learn More
Back
Next
Cancel

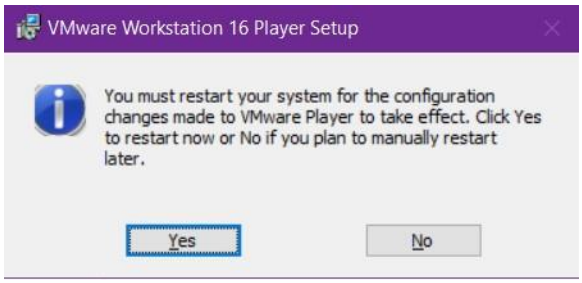

VMware Workstation 16 Player Setup
Custom Setup
Select the installation destination and any additional features.
Install to:
C:\Program Files (x86)\VMware\VMware Player\
Change...
Enhanced Keyboard Driver (a reboot will be required to use this feature)
This feature requires 10MB on your host drive.
Add VMware Workstation console tools into system PATH
Back
Next
Cancel


VMware Workstation 16 Player Setup
Shortcuts
Select the shortcuts you wish to place on your system.
Create shortcuts for VMware Workstation 16 Player in the following places:
Desktop
Start Menu Programs Folder
Back
Next
Cancel

Step 4:Once you setup all the things click on install

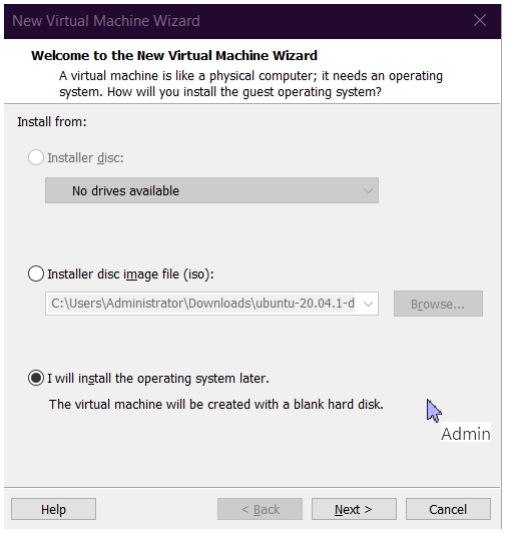


- Step 5: Now, it will install VMware in your system. After, successful installation it will show pop-up to restart your system to setting up all requirement.

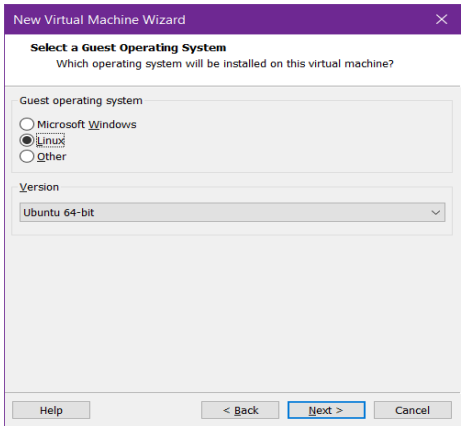


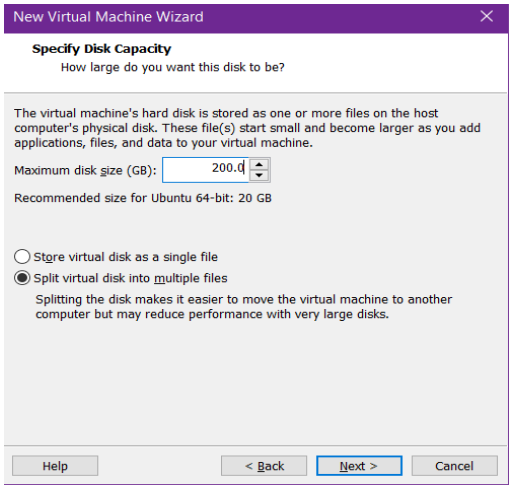
- Now, you are ready to create virtual machine on your computer

- **Create Virtual Machine**
- Step 1: open VMware and click on create virtual device.
- Step 2: select option based on your requirement. Here, we will install OS in future step

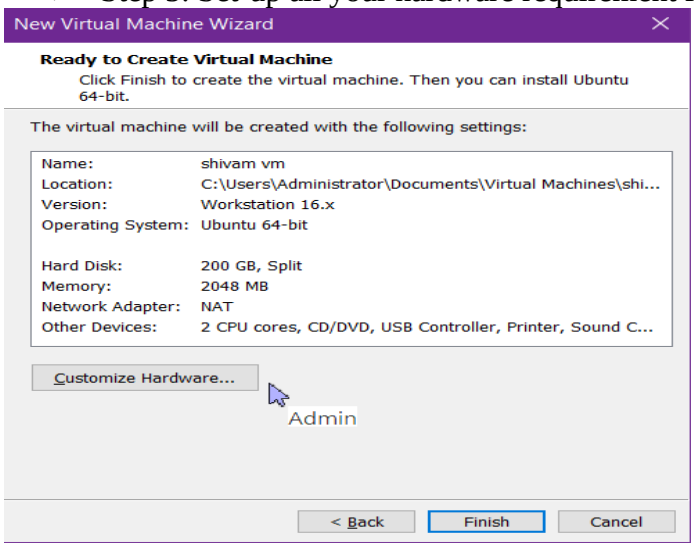


- Step 3: Select OS for which you want to create your VM. Here, we want Linux so we will selectLinux(Ubuntu).

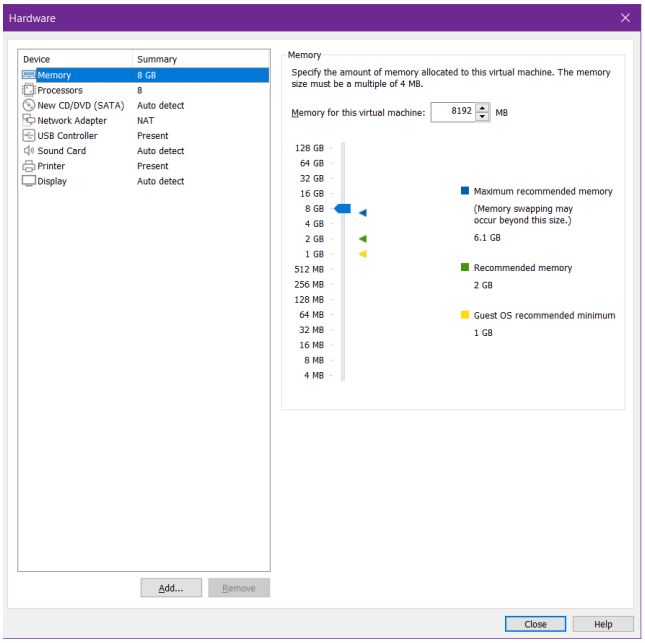




- Step 5: Set-up all your hardware requirement for your VM.



- Step 6: You can go with default hardware and can customize by clicking customize hardware



- Step 7:At last select ISO file you have download from internet to set up OS and create VM

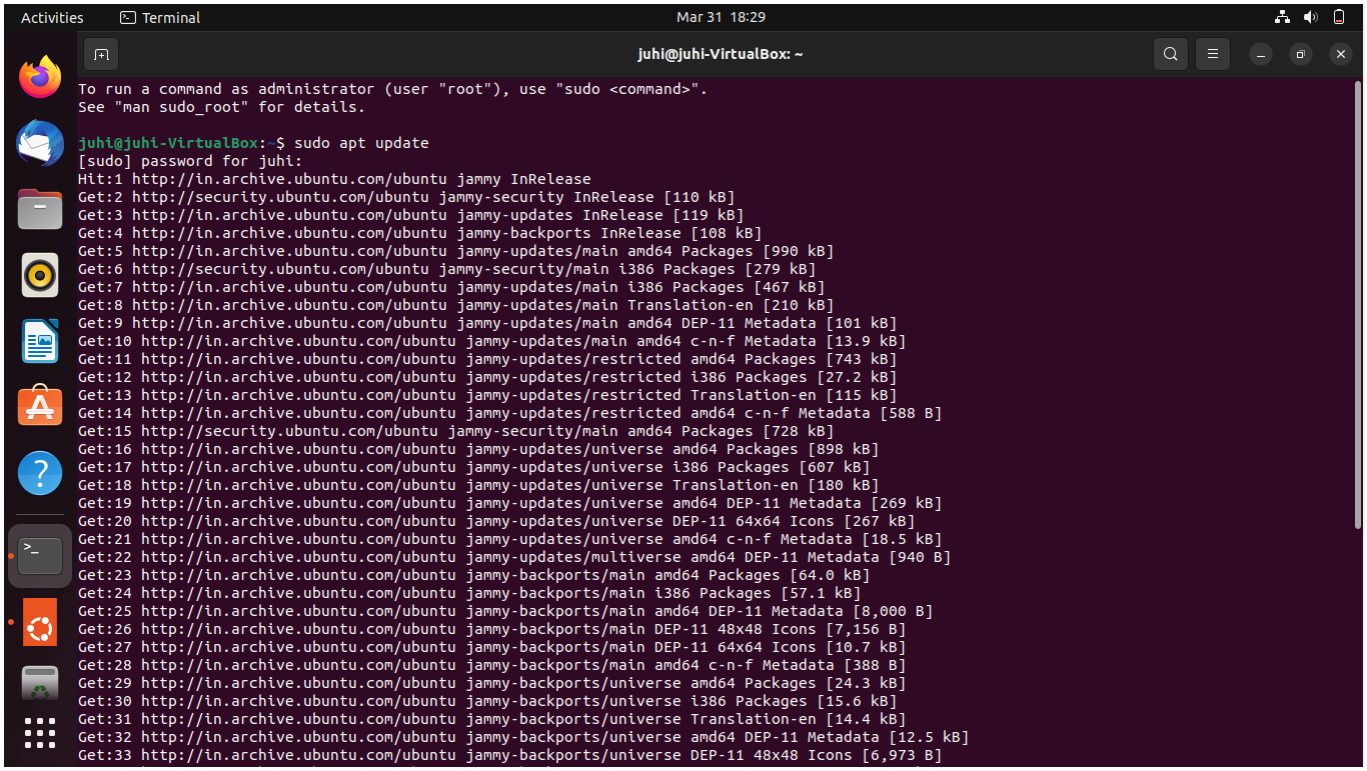
- Step 8: Now, it will install or create your VM

Step 1: Install MariaDB Database Server

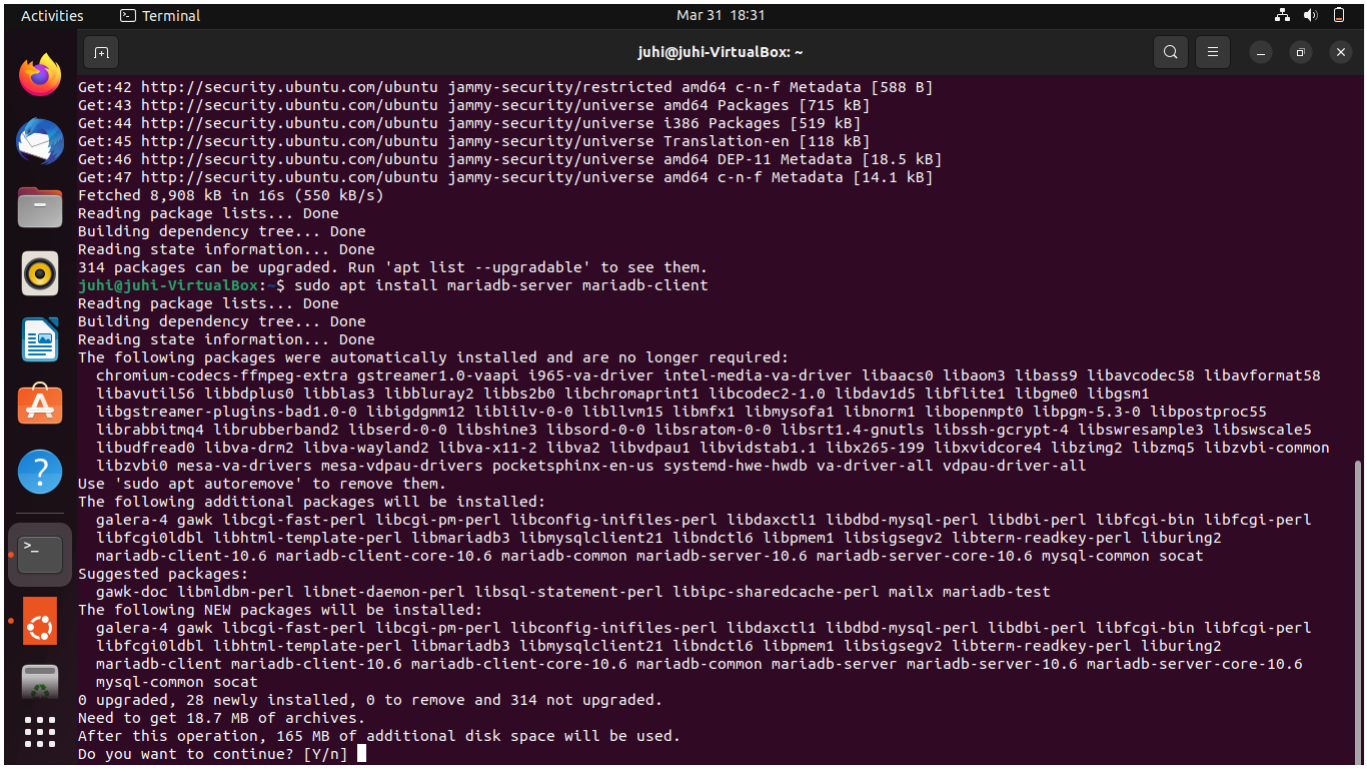
OpenNebula also needs a database server to store its content… and MariaDB database server is a greatplace to start when looking at open source database servers to use with OpenNebula.

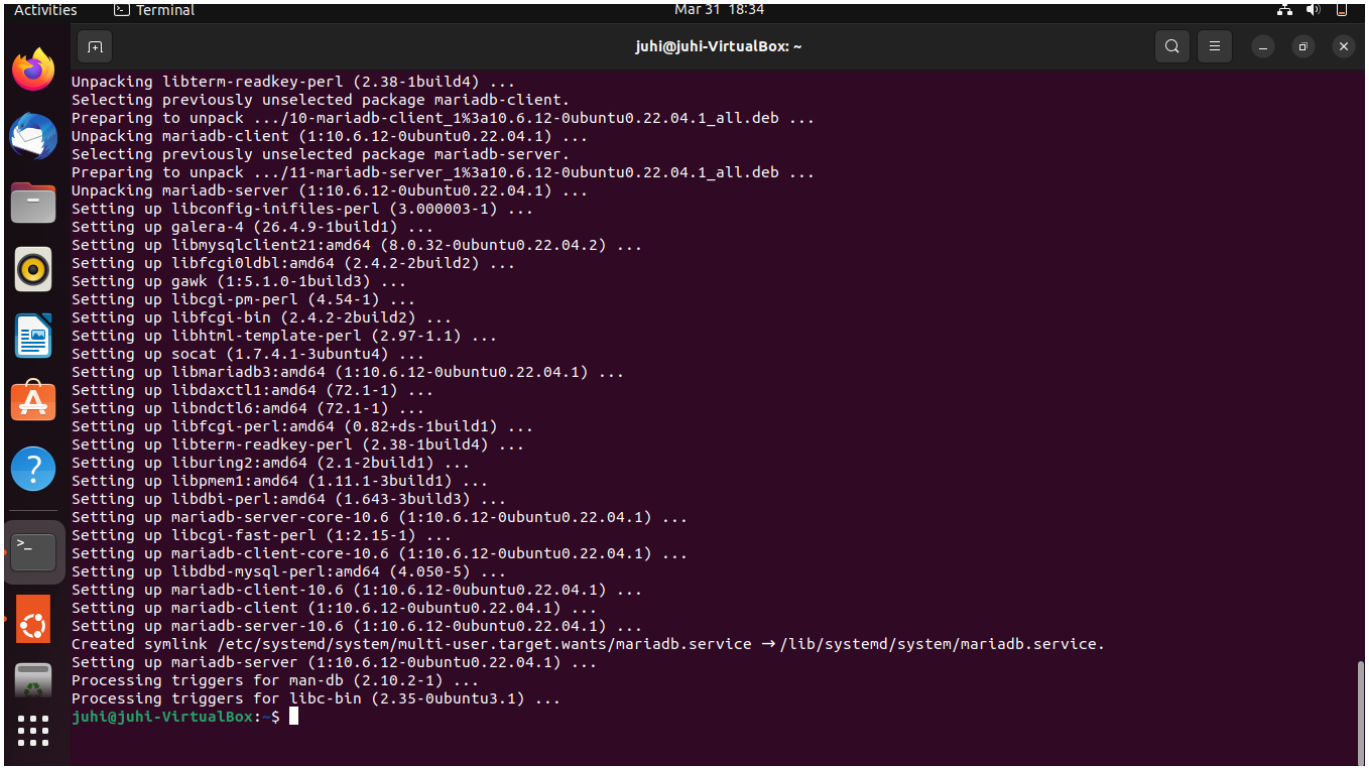
To install MariaDB run the commands below:

- sudo apt update



- sudo apt install mariadb-server mariadb-client





After installing MariaDB, the commands below can be used to stop, start and enable MariaDB service toalways start up when the server boots..

- sudo systemctl stop mariadb.service
- sudo systemctl start mariadb.service
- sudo systemctl enable mariadb.service

After that, run the commands below to secure MariaDB server by creating a root password anddisallowing remote root access.

sudo mysql_secure_installation

When prompted, answer the questions below by following the guide.Enter current password for root (enter for none): Just press the EnterSet root password? [Y/n]: Y

New password: Enter password

Re-enter new password:

Repeat passwordRemove anonymous users? [Y/n]: Y

Disallow root login

remotely? [Y/n]: Y

Remove test database and access to it?

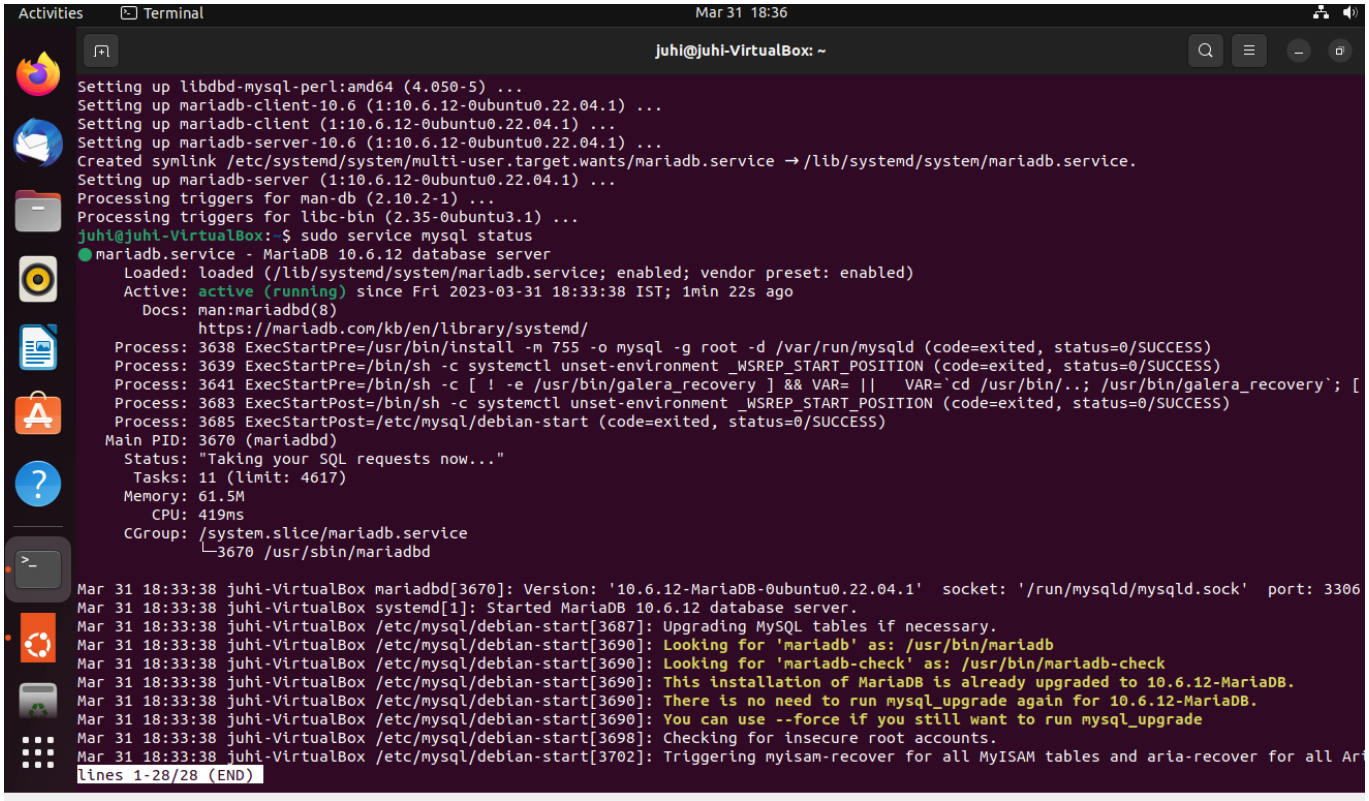
[Y/n]: YReload privilege tables now?

[Y/n]: Y

Restart MariaDB server

To test if MariaDB is installed and working, run the commands below:

- sudo systemctl status mariadb

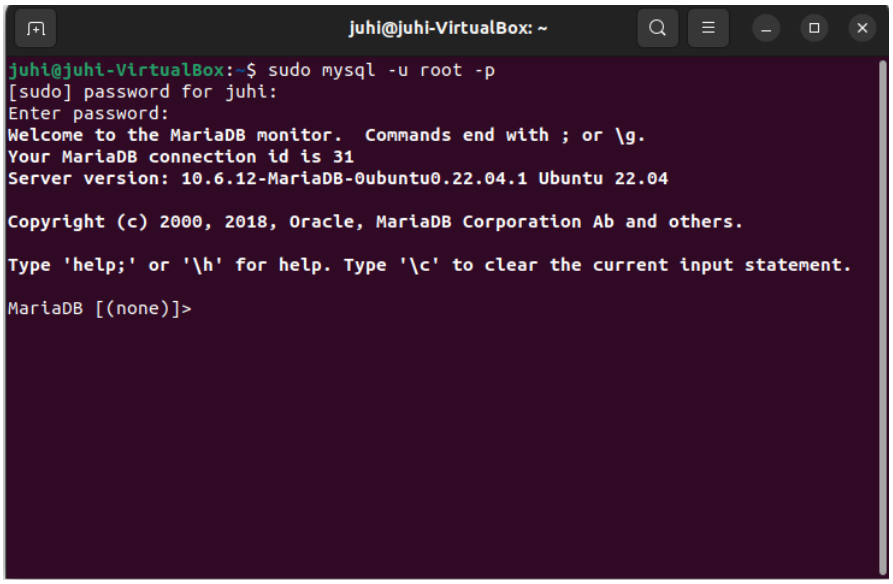


Step 2: Create OpenNebula Database

Now that you've install all the packages that are required, continue below to start configuring theservers. First create a blank database for OpenNebula to use.

To do that, run the commands below to logon to MariaDB. When prompted for a password, type theroot password you created above.

- sudo mysql -u root –p

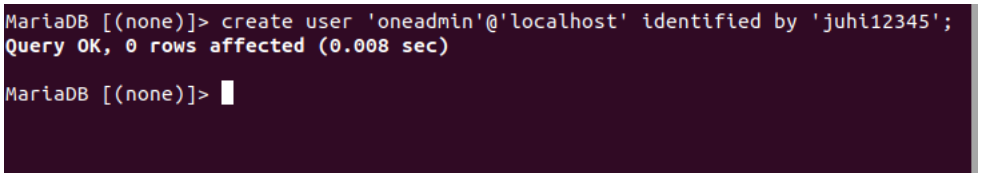


Then create a database called opennebula

- CREATE DATABASE opennebula;

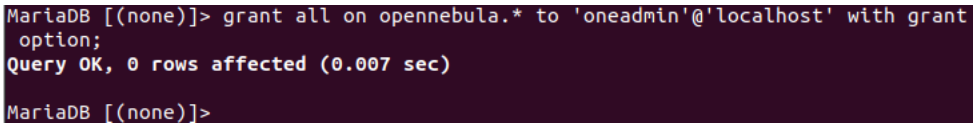
Create a database user called opennebulauser with new password

- CREATE USER 'opennebulauser'@'localhost' IDENTIFIED BY 'new_password_here';



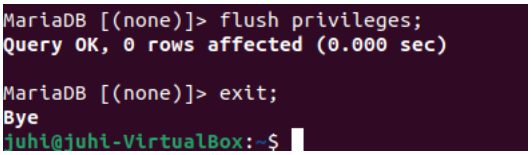
Next, grant the user full access to the cakephpuser database.

- GRANT ALL ON opennebula.* TO 'opennebulauser'@'localhost' WITH GRANT OPTION;



Finally, save your changes and exit.

- FLUSH PRIVILEGES;
- EXIT;



Now that MariaDB server is installed and a database created, now go and install OpenNebula.

Step 3: Install OpenNebula

By default, OpenNebula isn't available via Ubuntu default repositories. If you'd like to install it inUbuntu, you'll have to install it from its repository.

First, run the commands below to add its repository key to Ubuntu

- wget -q -O- https://downloads.opennebula.org/repo/repo.key | sudo apt-key add –

```
juhi@juhi-VirtualBox:~$ wget -q -O- https://downloads.opennebula.org/repo/repo.key | sudo apt-key add -
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
OK
juhi@juhi-VirtualBox:~$
```

After adding the repository's key, run the commands below to add its repository file.

- echo "deb https://downloads.opennebula.org/repo/5.12/Ubuntu/18.04 stable opennebula" |sudo tee /etc/apt/sources.list.d/opennebula.list

```
juhi@juhi-VirtualBox:~$ echo "deb https://download.opennebula.org/repo/5.12/Ubuntu/16.04 stable opennebula" | sudo tee /etc/apt/sources.list.
d/opennebula.list
deb https://download.opennebula.org/repo/5.12/Ubuntu/16.04 stable opennebula
juhi@juhi-VirtualBox:~$
```

Now that the repository's key and file are added, run the commands below to install OpenNebula.

- sudo apt update
- sudo apt install opennebula opennebula-sunstone opennebula-gate opennebula-flow

```
  rubygems-integration sharutils sqlite3
Suggested packages:
  augeas-doc gnuplot-doc apache2 | lighttpd | httpd augeas-tools
  thin-provisioning-tools mysql-server gfortran python-dev python-nose
  python-numpy-dbg python-numpy-doc debootstrap ri ruby-dev bundler bsd-mailx
  | mailx sqlite3-doc
The following NEW packages will be installed:
  aglfn augeas-lenses curl dmeventd dmsetup fonts-lato gnuplot-data
  gnuplot-nox gnuplot-tex groff javascript-common libaugeas0 libblas-common
  libblas3 libboost-random1.58.0 libboost-thread1.58.0
  libdevmapper-event1.02.1 libgfortran3 libiscsi2 libjs-jquery liblapack3
  liblua5.1-0 liblvm2app2.2 liblvm2cmd2.02 libpq5 librados2 librbd1 libruby2.3
  libzmq3-dev lvm2 opennebula opennebula-common opennebula-common-onescape
  opennebula-flow opennebula-gate opennebula-rubygems opennebula-sunstone
  opennebula-tools psutils python-numpy qemu-block-extra qemu-utils rake ruby
  ruby-did-you-mean ruby-minitest ruby-net-telnet ruby-opennebula
  ruby-power-assert ruby-test-unit ruby2.3 rubygems-integration sharutils
  sqlite3
The following packages will be upgraded:
  libcurl3-gnutls
1 upgraded, 54 newly installed, 0 to remove and 163 not upgraded.
Need to get 56.0 MB of archives.
After this operation, 441 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

```
ne
The key's randomart image is:
+---[RSA 2048]----+
|       =*+=BO++o.|
|      ..E+*X++.  |
|     o o O++=+   |
|    +   =..+. o .|
|   . .  S.o    o |
|    .  .         |
|     ..          |
|      ..         |
|      oo         |
+----[SHA256]-----+
Setting up opennebula-flow (5.12.0.3-1.ce) ...
Setting up opennebula-gate (5.12.0.3-1.ce) ...
Setting up opennebula-sunstone (5.12.0.3-1.ce) ...
Processing triggers for libc-bin (2.23-0ubuntu11.2) ...
Processing triggers for initramfs-tools (0.122ubuntu8.16) ...
update-initramfs: Generating /boot/initrd.img-4.15.0-112-generic
Processing triggers for ureadahead (0.100.0-19.1) ...
Processing triggers for systemd (229-4ubuntu21.28) ...
shivam@shivam-virtual-machine:~$
```

When you're done installing OpenNebula, run the commands below to install Ruby libraries and otherrequired packages to support OpenNebula.

```
Execution continues in 15 seconds ...
Distribution "debian" detected.
About to install these dependencies:
* gcc
* rake
* libxml2-dev
* libxslt1-dev
* patch
* g++
* libsqlite3-dev
* libcurl4-openssl-dev
* libssl-dev
* libmysqlclient-dev
* postgresql-server-dev-all
* libzmq5
* libzmq3-dev
* libaugeas-dev
* ruby-dev
* make

Press enter to continue...
```

- sudo /usr/share/one/install_gems

That should install OpenNebula on Ubuntu. After installing, the commands below can be used to startand enable OpenNebula services.

- sudo systemctl start opennebula
- sudo systemctl enable opennebula

- sudo systemctl start opennebula-sunstone
- sudo systemctl enable opennebula-sunstone

```
shivam@shivam-virtual-machine:~$ sudo systemctl start opennebula
shivam@shivam-virtual-machine:~$ sudo systemctl enable opennebula
Created symlink from /etc/systemd/system/multi-user.target.wants/opennebula.serv
ice to /lib/systemd/system/opennebula.service.
shivam@shivam-virtual-machine:~$ sudo systemctl start opennebula-sunstone
shivam@shivam-virtual-machine:~$ sudo systemctl enable opennebula-sunstone
Created symlink from /etc/systemd/system/multi-user.target.wants/opennebula-suns
tone.service to /lib/systemd/system/opennebula-sunstone.service.
```

To support MariaDB database server, run the commands below to open its configuration file

- sudo nano /etc/one/oned.conf

Then edit the highlighted lines with database name, user and password created above and save the file.SCRIPTS_REMOTE_DIR=/var/tmp/one

PORT = 2633
LISTEN_ADDRESS = "0.0.0.0"
#DB = [BACKEND = "sqlite"]
Sample configuration for MySQL
DB = [BACKEND = "mysql",
SERVER = "localhost",
PORT = 0,
USER = "opennebulauser",
PASSWD = "database_password_here",
DB_NAME = "opennebula",
CONNECTIONS = 50]
VNC_PORTS = [START = 5900
RESERVED = "6800, 6801, 6810:6820, 9869"]

```
SCRIPTS_REMOTE_DIR=/var/tmp/one


#DB = [ BACKEND = "sqlite",
#       TIMEOUT = 2500 ]

# Sample configuration for MySQL
 DB = [ BACKEND = "mysql",
        SERVER  = "localhost",
        PORT    = 0,
        USER    = "oneadmin",
        PASSWD  = "sp7283",
        DB_NAME = "opennebula",
        CONNECTIONS = 25,
        COMPARE_BINARY = "no" ]
```

Save (ctrl+o) the file and exit (ctrl+x).

To access OpenNebula portal, you'll need a credential. By default, its default username and password isstored in the credential below:

- sudo nano /var/lib/one/.one/one_auth

```
oneadmin:euwysibCeok9
```

Edit it and replace the username and password that suits you.
After that, run the commands below to check the status of OpenNebula services.

- sudo systemctl status opennebula

You should get similar lines as below:

```
opennebula.service - OpenNebula Cloud Controller Daemon
   Loaded: loaded (/lib/systemd/system/opennebula.service; enabled; vendor prese
   Active: active (running) since Wed 2021-03-10 14:34:38 IST; 5min ago
 Main PID: 39049 (oned)
   CGroup: /system.slice/opennebula.service
           ├─39049 /usr/bin/oned -f
           ├─39054 ruby /usr/lib/one/mads/one_hm.rb -p 2101 -l 2102 -b 127.0.0.1
           ├─39090 ruby /usr/lib/one/mads/one_vmm_exec.rb -t 15 -r 0 kvm
           ├─39107 ruby /usr/lib/one/mads/one_vmm_exec.rb -t 15 -r 0 lxd
           ├─39124 ruby /usr/lib/one/mads/one_vmm_exec.rb -t 15 -r 0 firecracker
           ├─39141 ruby /usr/lib/one/mads/one_vmm_exec.rb -l deploy,shutdown,reb
           ├─39158 ruby /usr/lib/one/mads/one_tm.rb -t 15 -d dummy,lvm,shared,fs
           ├─39178 ruby /usr/lib/one/mads/one_datastore.rb -t 15 -d dummy,fs,lvm
           ├─39194 ruby /usr/lib/one/mads/one_market.rb -t 15 -m http,s3,one,lin
           ├─39210 ruby /usr/lib/one/mads/one_ipam.rb -t 1 -i dummy
           ├─39222 ruby /usr/lib/one/mads/one_auth_mad.rb --authn ssh,x509,ldap,
           ├─39235 /usr/lib/one/mads/onemonitord -c monitord.conf
           ├─39236 ruby /usr/lib/one/mads/one_im_exec.rb -r 3 -t 15 -w 90 firecr
           ├─39250 ruby /usr/lib/one/mads/one_im_exec.rb -r 3 -t 15 -w 90 kvm
           ├─39264 ruby /usr/lib/one/mads/one_im_exec.rb -r 3 -t 15 -w 90 lxd
           ├─39278 ruby /usr/lib/one/mads/one_im_exec.rb -l -c -t 15 -r 0 vcente
           └─39287 ruby /var/lib/one/remotes/im/lib/vcenter_monitor.rb

lines 1-23
```

That shows the service is running.

Another Way for installation of open nebula :- Install opennebula-4.14.2-sandbox.vdi File.

To install the vdi file :-
https://drive.google.com/file/d/1HvFDzcFrgbTmRQQzjJ4qjxbl_pRKPlG9/view?usp=drive_link

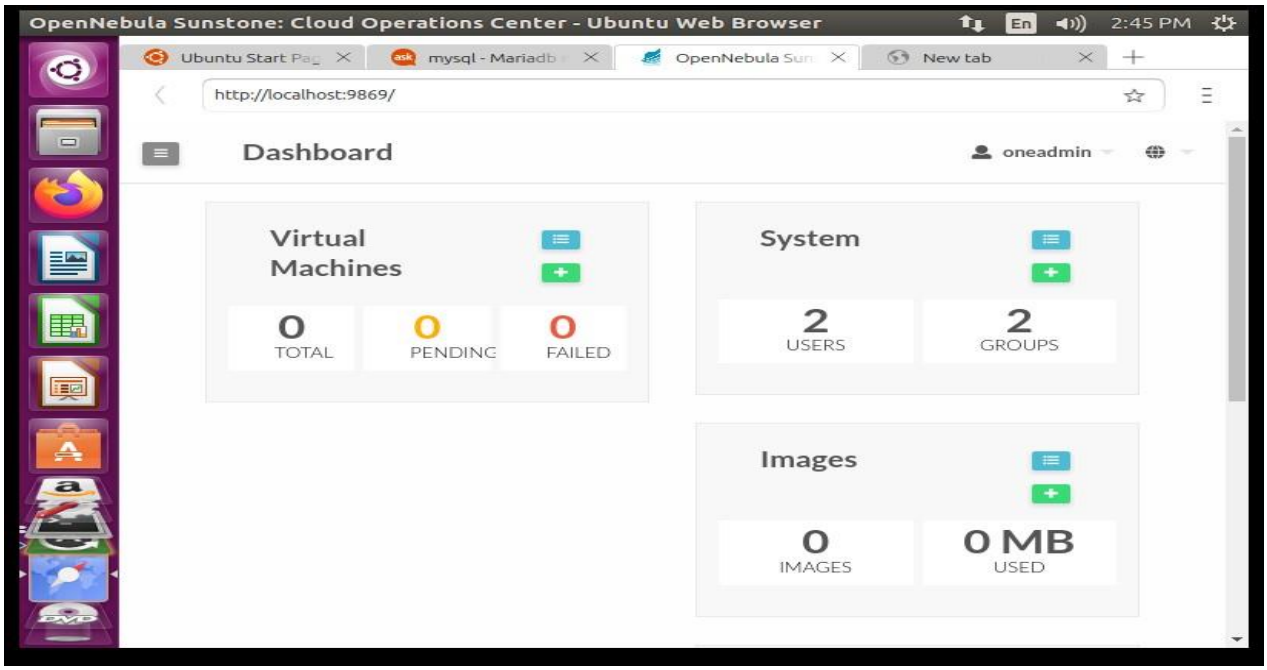
Step 4: Access OpenNebula Portal

Now that OpenNebula is installed, open your web browser ad browse to the server hostname or IPaddress followed by port 9869

http://localhost:9869

Login with the credential found in the file above.

That should redirect you to OpenNebula dashboard.



That's it! Your platform is ready, but you'll need to add VMware or KVM

nodes to manage.Setup KVM Node

Step 5: Configure KVM node

Now, that OpenNebula platform is installed and ready, you'll need to add nodes to manage. To addKVM nodes, simply run the commands below node you want to manage via OpenNebula.

wget -q -O- https://downloads.opennebula.org/repo/repo.key | sudo apt-key

add -Then add its repository

- echo "deb https://downloads.opennebula.org/repo/5.12/Ubuntu/18.04 stable opennebula" |sudo tee /etc/apt/sources.list.d/opennebulakvmnode.list

Finally, run the commands below to install the node package

- sudo apt update

```
sp7283host@sp7283host-virtual-machine:~$ sudo apt update
[sudo] password for sp7283host:
Sorry, try again.
[sudo] password for sp7283host:
Hit:1 http://in.archive.ubuntu.com/ubuntu bionic InRelease
Hit:2 http://in.archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:3 http://in.archive.ubuntu.com/ubuntu bionic-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu bionic-security InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
226 packages can be upgraded. Run 'apt list --upgradable' to see them.
sp7283host@sp7283host-virtual-machine:~$ wget -q -O- https://downloads.opennebu
la.org/repo/repo.key | sudo apt-key add -
OK
sp7283host@sp7283host-virtual-machine:~$ echo "deb https://downloads.opennebula
.org/repo/5.12/Ubuntu/18.04 stable opennebula" | sudo tee /etc/apt/sources.list
.d/opennebulakvmnode.list
deb https://downloads.opennebula.org/repo/5.12/Ubuntu/18.04 stable opennebula
sp7283host@sp7283host-virtual-machine:~$ sudo apt update
```

sudo apt-get install opennebula-nod

```
  ruby-minitest ruby-net-telnet ruby-power-assert ruby-sqlite3 ruby-test-unit
  ruby2.5 rubygems-integration seabios sharutils sqlite3 ssh-import-id vlan
Suggested packages:
  augeas-doc apache2 | lighttpd | httpd libvirt-daemon-driver-storage-gluster
  libvirt-daemon-driver-storage-sheepdog libvirt-daemon-driver-storage-zfs
  numad radvd auditd systemtap nfs-common zfsutils pm-utils
  thin-provisioning-tools molly-guard monkeysphere rssh ssh-askpass samba
  vde2 sgabios ovmf debootstrap ri ruby-dev bundler sharutils-doc bsd-mailx
  | mailx sqlite3-doc
The following NEW packages will be installed:
  augeas-lenses augeas-tools bridge-utils cpu-checker dmeventd ebtables
  fonts-lato ibverbs-providers ipset ipxe-qemu ipxe-qemu-256k-compat-efi-roms
  javascript-common libaio1 libaugeas0 libcacard0 libdevmapper-event1.02.1
  libfdt1 libibverbs1 libipset3 libiscsi7 libjs-jquery liblvm2app2.2
  liblvm2cmd2.02 libnetcf1 libnl-route-3-200 librados2 librbd1 librdmacm1
  libreadline5 libruby2.5 libsdl1.2debian libspice-server1 libusbredirparser1
  libvirt-clients libvirt-daemon libvirt-daemon-driver-storage-rbd
  libvirt-daemon-system libvirt0 libxen-4.9 libxenstore3.0 libxml2-utils lvm2
  msr-tools ncurses-term opennebula-common opennebula-common-onescape
  opennebula-node openssh-server openssh-sftp-server qemu-block-extra
  qemu-kvm qemu-system-common qemu-system-x86 qemu-utils rake ruby
  ruby-did-you-mean ruby-minitest ruby-net-telnet ruby-power-assert
  ruby-sqlite3 ruby-test-unit ruby2.5 rubygems-integration seabios sharutils
  sqlite3 ssh-import-id vlan
0 upgraded, 69 newly installed, 0 to remove and 226 not upgraded.
Need to get 28.2 MB of archives.
After this operation, 123 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

```
Done.
Adding user `oneadmin' to group `kvm' ...
Adding user oneadmin to group kvm
Done.
Processing triggers for fontconfig (2.12.6-0ubuntu2) ...
Processing triggers for ufw (0.36-0ubuntu0.18.04.1) ...
Processing triggers for ureadahead (0.100.0-21) ...
Processing triggers for install-info (6.5.0.dfsg.1-2) ...
Processing triggers for libc-bin (2.27-3ubuntu1.2) ...
Processing triggers for systemd (237-3ubuntu10.42) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for initramfs-tools (0.130ubuntu3.9) ...
update-initramfs: Generating /boot/initrd.img-5.4.0-66-generic
sp7283host@sp7283host-virtual-machine:~$
```

After installing the node package, open its configuration file to allow OpenNebula to manage it.

- sudo nano /etc/libvirt/libvirtd.conf

Then edits these lines and save.

This is restricted to'root' by default.

unix_sock_group = "oneadmin"
If not using PolicyKit and setting group ownership for access control,
then you may want to relax this too.
unix_sock_rw_perms = "0777"

```
unix_sock_group = "oneadmin"

# Set the UNIX socket permissions for the R/O socket. This is used
# for monitoring VM status only
#
# Default allows any user. If setting group ownership, you may want to
# restrict this too.
unix_sock_ro_perms = "0777"

# Set the UNIX socket permissions for the R/W socket. This is used
# for full management of VMs
#
# Default allows only root. If PolicyKit is enabled on the socket,
# the default will change to allow everyone (eg, 0777)
#
# If not using PolicyKit and setting group ownership for access
# control, then you may want to relax this too.
unix_sock_rw_perms = "0777"
```

Save the file and exit.

Restart the client services

- sudo systemctl restart libvirtd
- sudo systemctl restart libvirt-bin

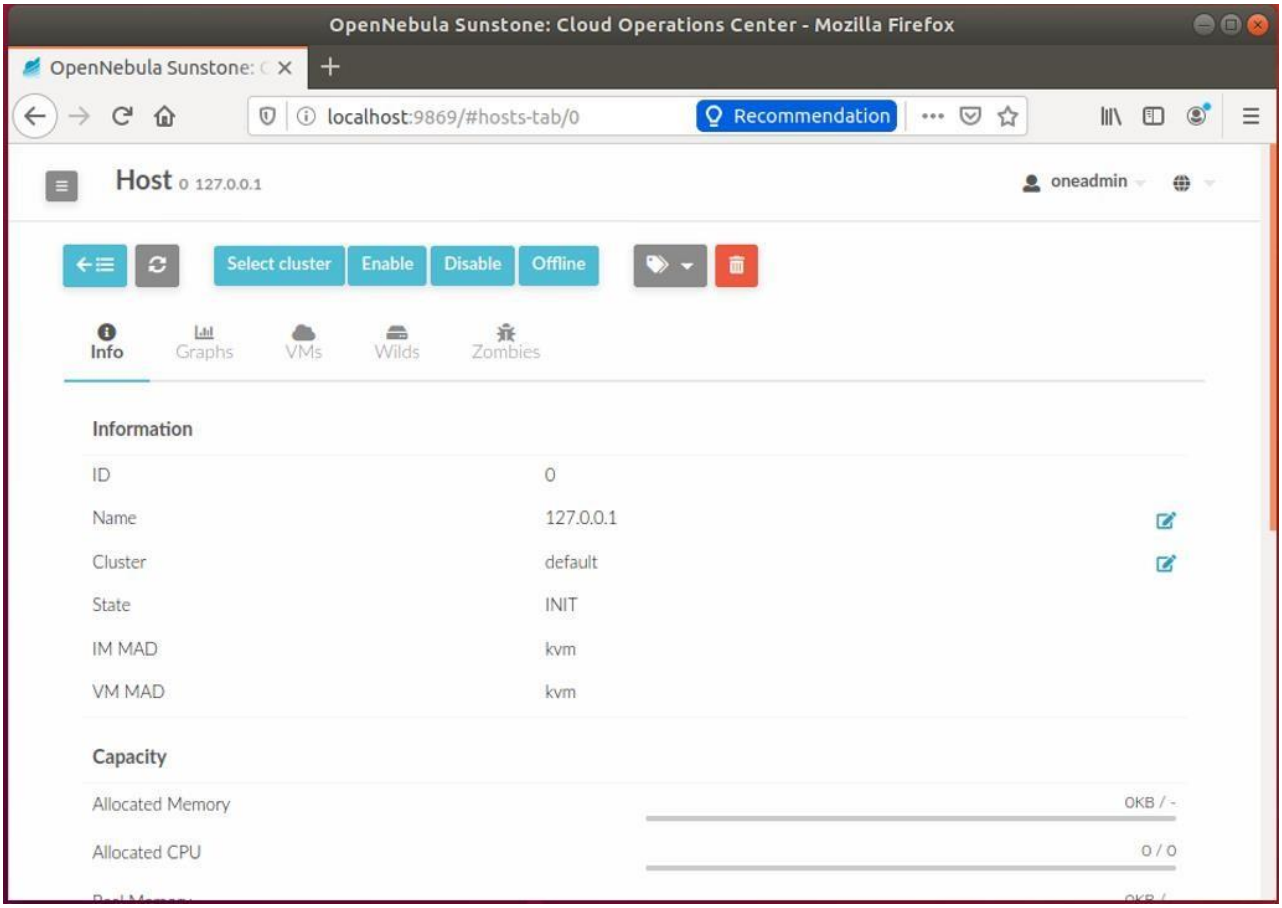
Communication is done via SSH. So you'll want install SSH and enable password-less SSH authenticationbetween the OpenNebula host and the KVM node.

After creating a password authentication, copy the server SSH key to the client known_hosts file at.

- /var/lib/one/.ssh/known_hosts
  All keys in there will be trusted.

To manage a KVM host, go to OpenNebula web interface, open Infrastructure >> Hosts and Click on the + button.

Add a new node to manage.



Step 6: Configure Passwordless SSH

OpenNebula Front-end connects to the hypervisor Hosts using SSH. You must distribute the public key of oneadmin user from all machines to the file /var/lib/one/.ssh/authorized_keys in all the machines. There are many methods to achieve the distribution of the SSH keys, ultimately the administrator should choose a method (the recommendation is to use a configuration management system). In this guide we are going to manually scp the SSH keys.

When the package was installed in the Front-end, an SSH key was generated and the authorized_keys populated. We will sync the id_rsa, id_rsa.pub and authorized_keys from the Front-end to the nodes. Additionally we need to create a known_hosts file and sync it as well to the nodes.

To create the known_hosts file, we have to execute this command as user oneadmin in theFront-end with all the node names as parameters:

- ssh-keyscan <node1> <node2> <node3> ... >> /var/lib/one/.ssh/known_hosts

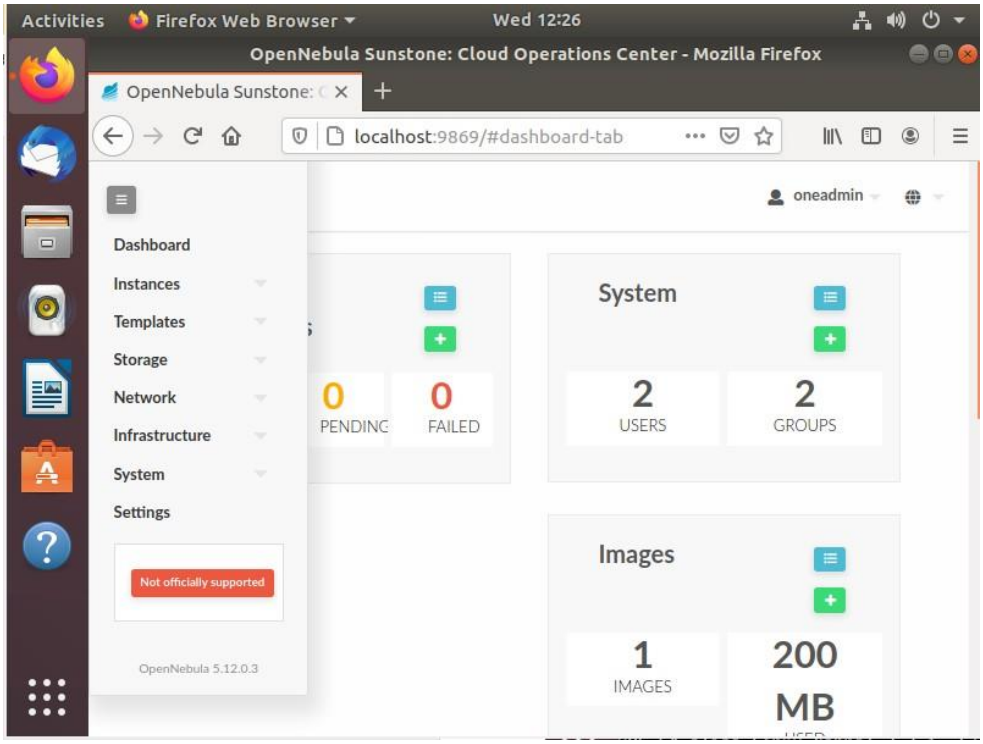
Now we need to copy the directory /var/lib/one/.ssh to all the nodes. The easiest way is to set atemporary password to oneadmin in all the hosts and copy the directory from the Front-end:

- scp -rp /var/lib/one/.ssh <node1>:/var/lib/one/

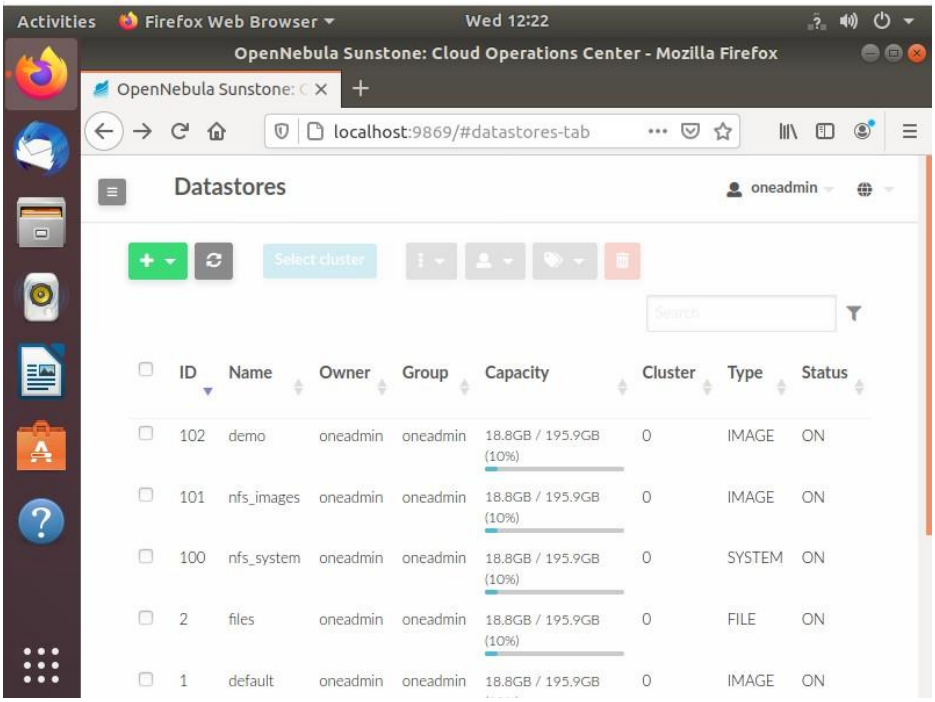
You should verify that connecting from the Front-end, as user oneadmin, to the nodes, and from thenodes to the Front-end, does not ask password:

- ssh <node1>
- ssh <frontend>
- exit

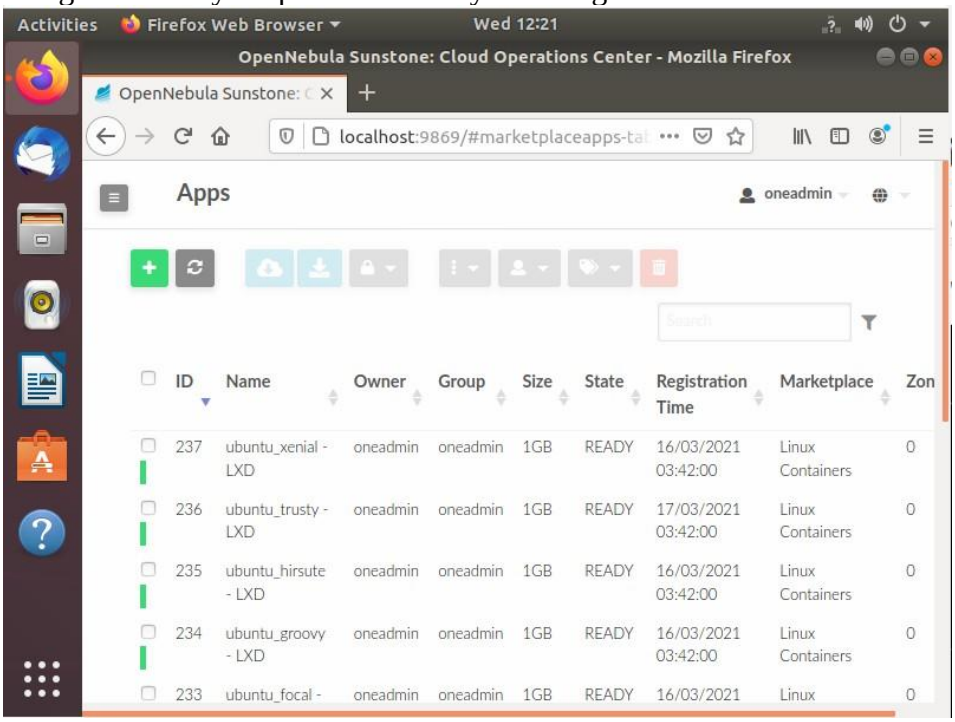
- On dashboard you will see the menu bar from that you can see your datastores from yourStorage menu bar.



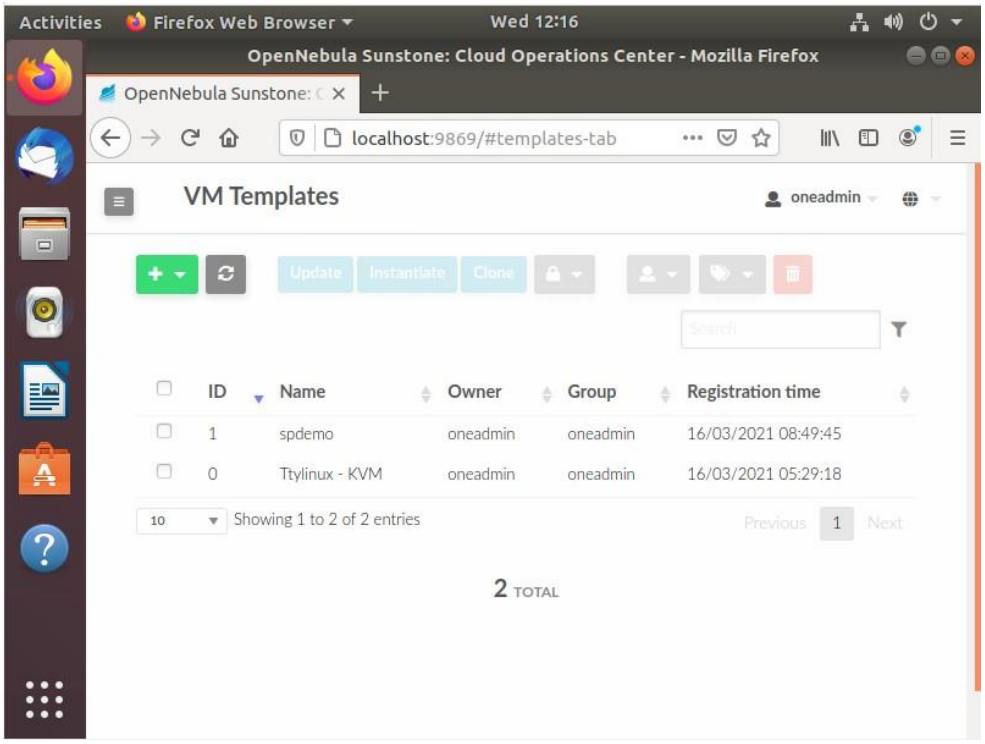
- You must have at least one datastore that can store your data you can create one from CLI aswell as from Sunstone.



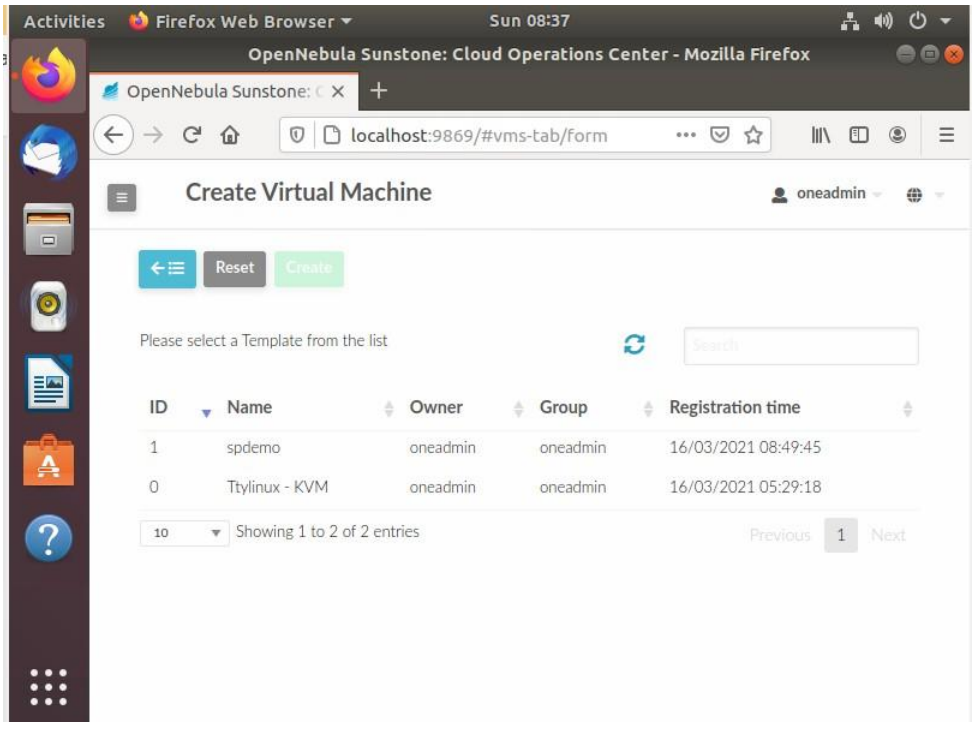
- Now, we have to add the apps to be run on our VMs. You can download the image or add toyour private cloud by selecting one.



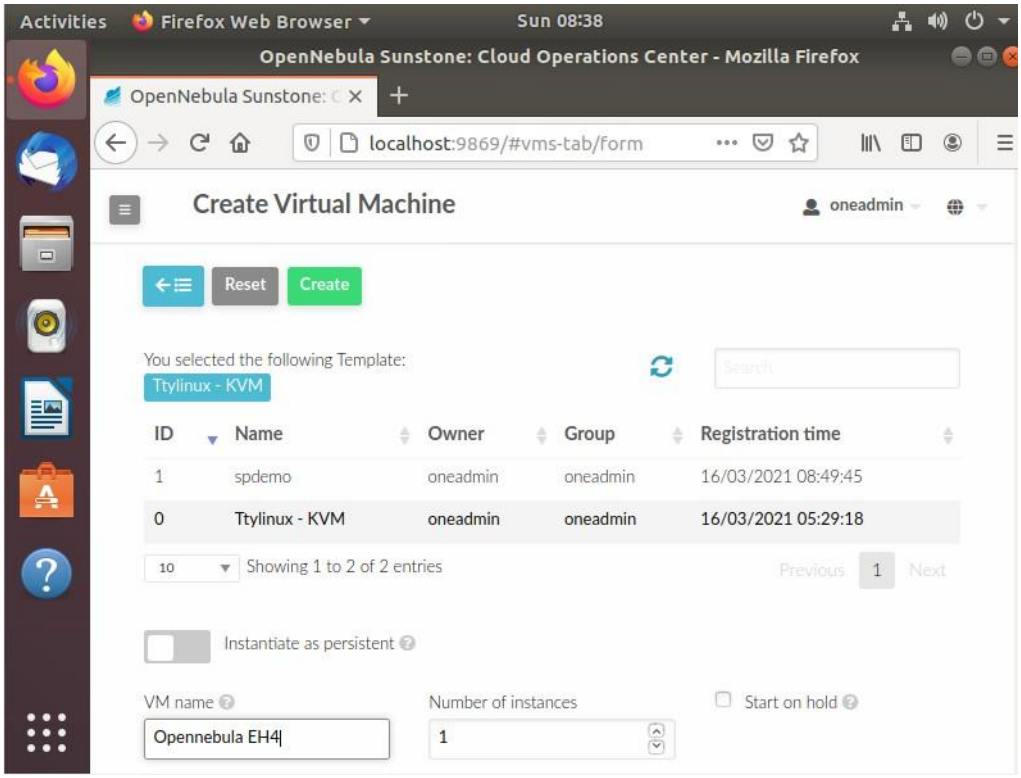
- In VM Templates you can see what apps you have added in your cloud and else you may createyour custom templates from the same.



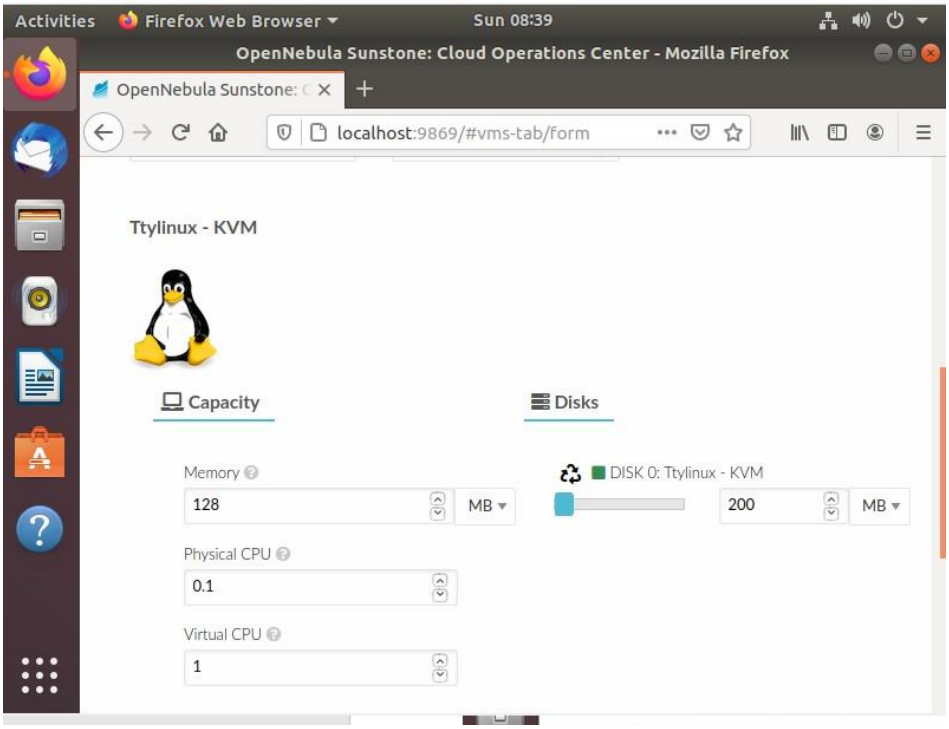
- We are done with all the thing. Now, we are good to create VM. Click on create VM.



- Once, you click on create VM you have to select the app for which you wish to create VM.

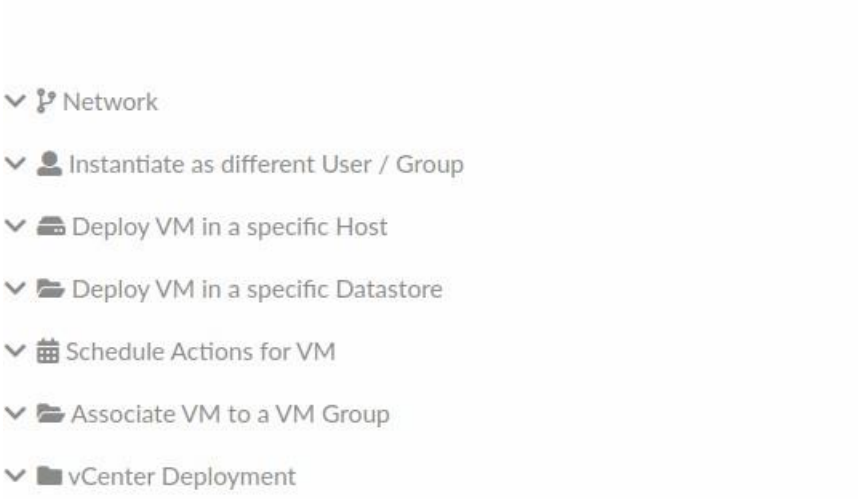


- Fill the other details. Remember Physical CPU value must be float and Virtual CPU value must beinteger.

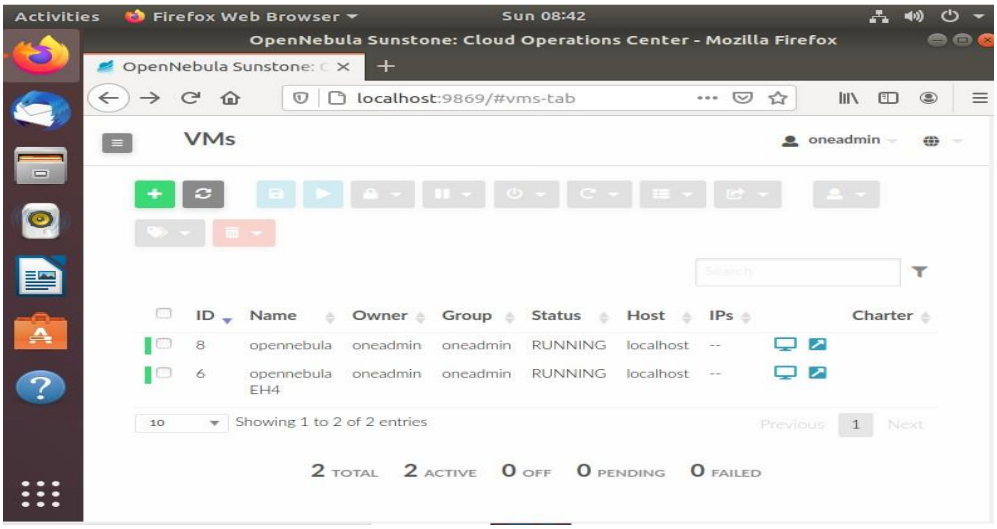


- From the same page you have many option you can select and change like on

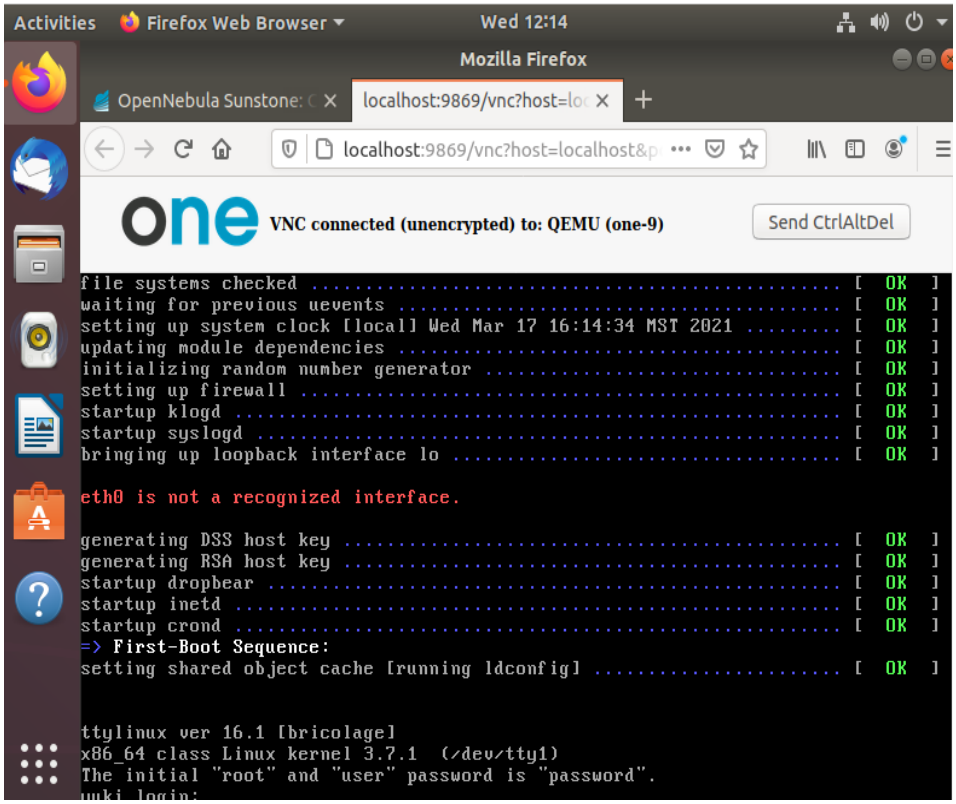
which datastoreyou wish to store your data, user, group and many more…



- Click on create now you must have this screen and all your machine must be on Running State.



- When you run machine your machine must be started on browser.



Conclusion:

Here, using open-nebula we have created our private cloud and implemented IaaS service on which wecan run machine.