

# Practical: 7

**AIM- Configuration of Private cloud (OpenNebulla).**

**Submitted by:** 21012021001\_Adeshara Brijesh



**Ganpat  
University**

॥ विद्यया समाजोत्कर्षः ॥

**U.V. Patel  
College of  
Engineering**

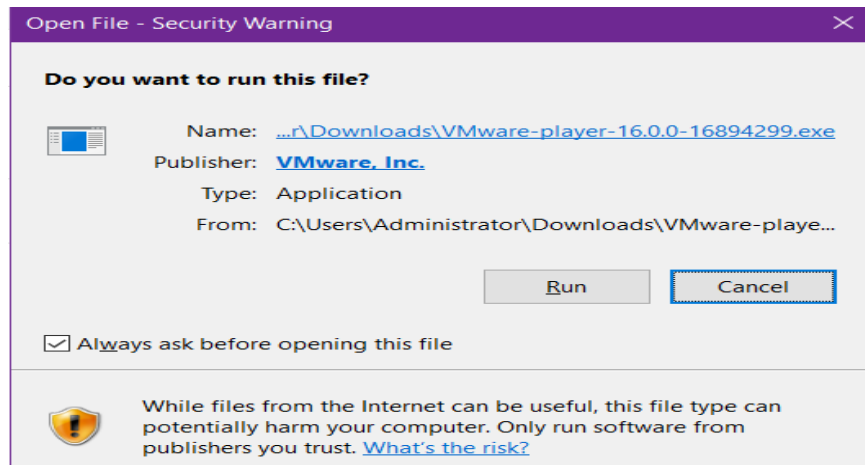
**Department of Computer  
Engineering/Information Technology**

## Practical: 7

---

### Installation of Linux.

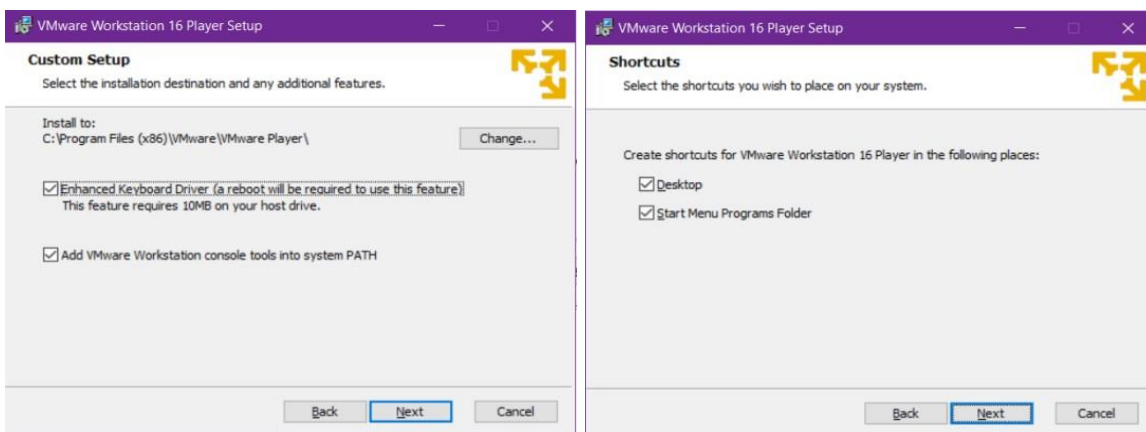
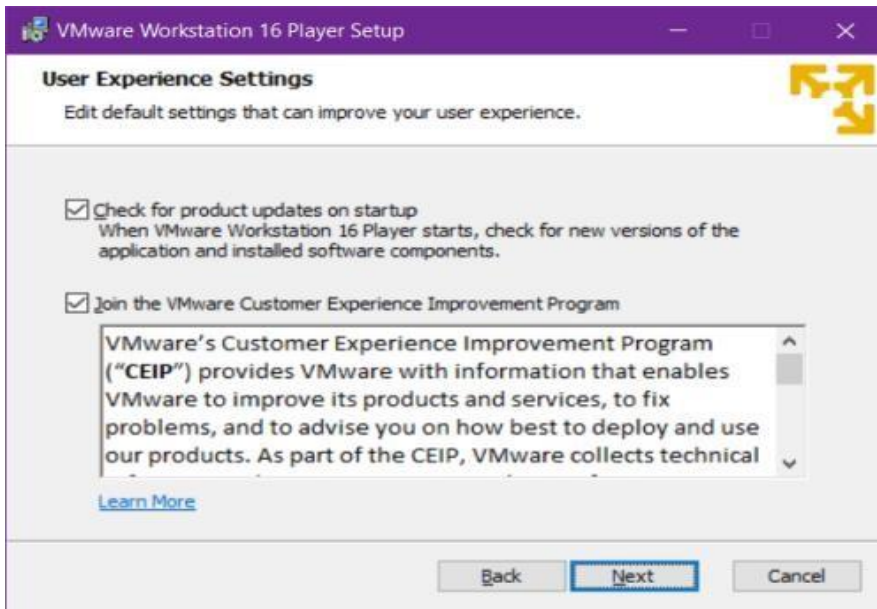
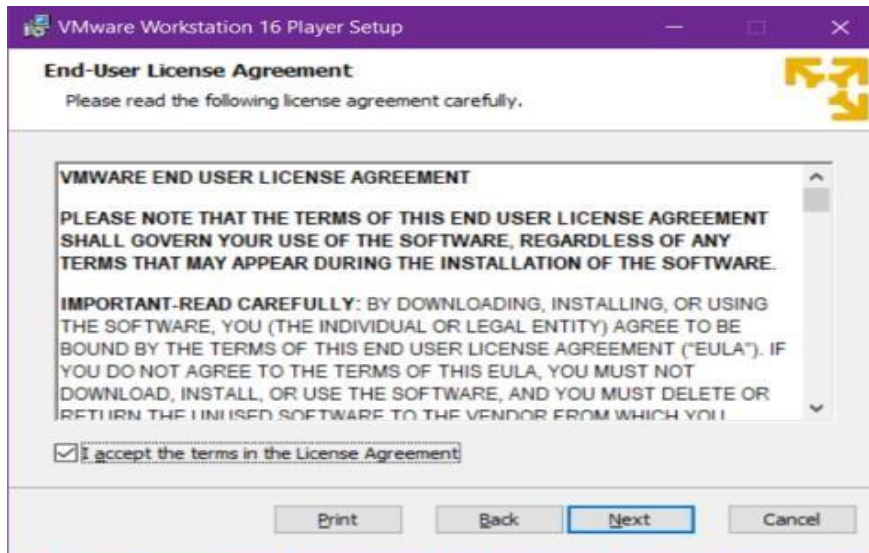
- Download ISO image file of Ubuntu 18.04 or more.
- <https://releases.ubuntu.com/18.04/>
- **Installation of VMware.**
- Step 1: Download VMware software according system requirement.
- Step 2: Run file by double clicking on exe file.



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



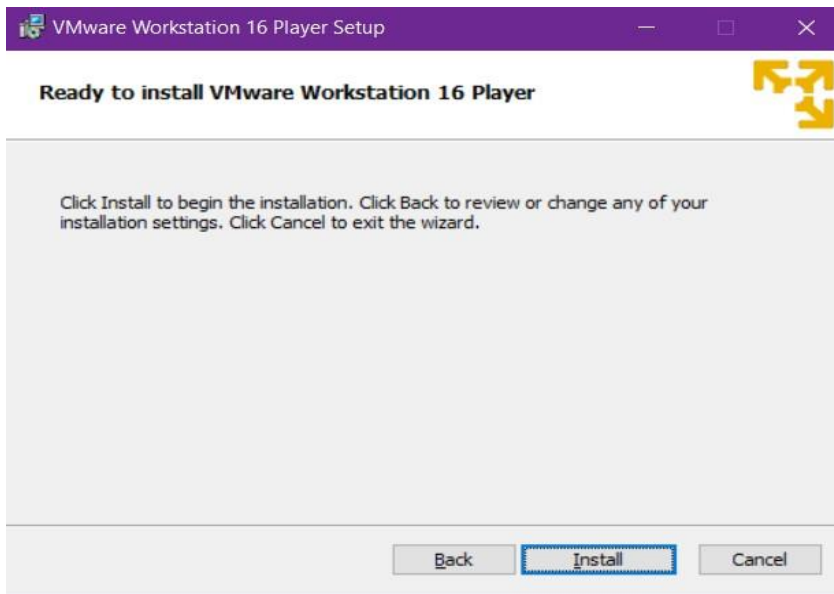
## Practical: 7



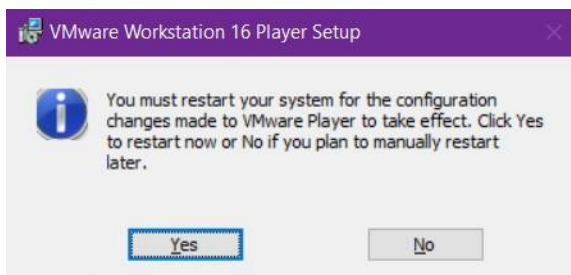
## Practical: 7

---

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

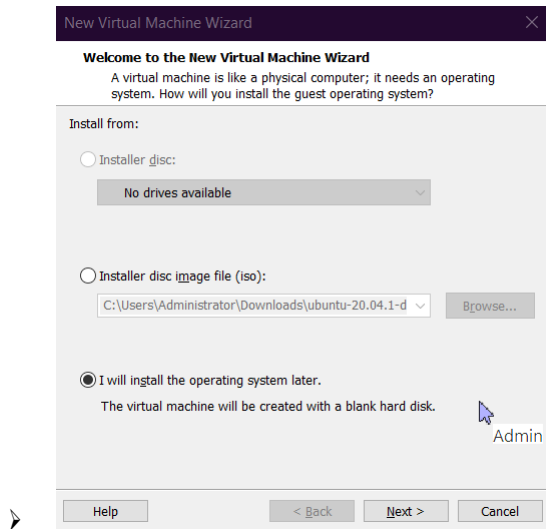


- Step 5: Now, it will install VMware in your system. After, successful installation it will show pop-up to restart your system to setting up all requirement.

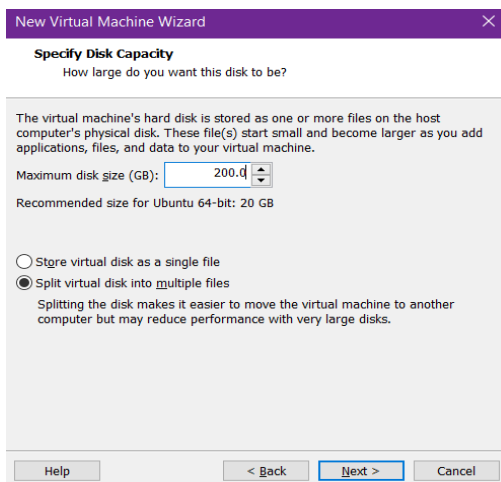
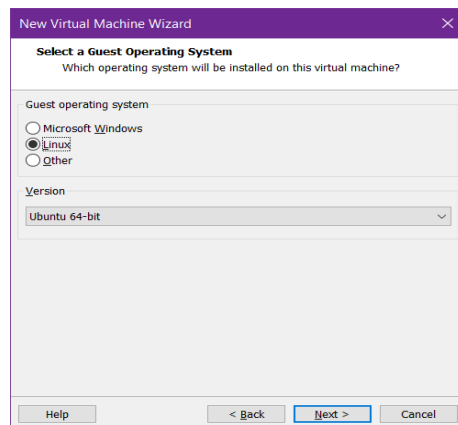


- Now, you are ready to create virtual machine on your computer
- **Create Virtual Machine**
- Step 1: open VMware and click on create virtual device.
- Step 2: select option based on your requirement. Here, we will install OS in future step

## Practical: 7

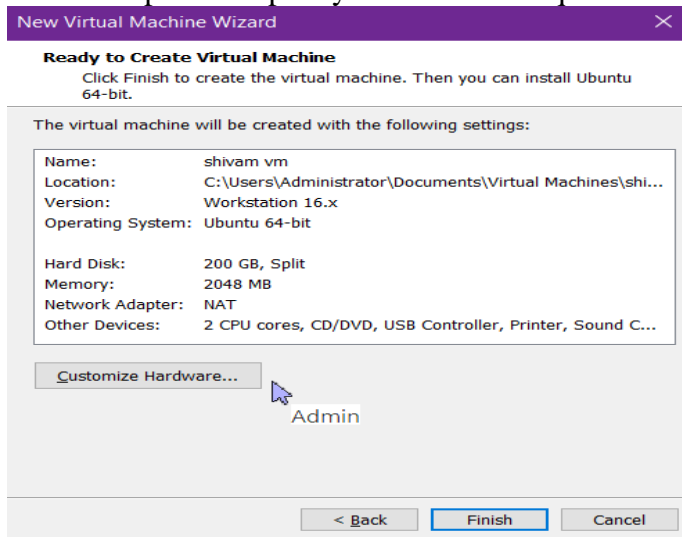


- Step 3: Select OS for which you want to create your VM. Here, we want Linux so we will select Linux(Ubuntu).

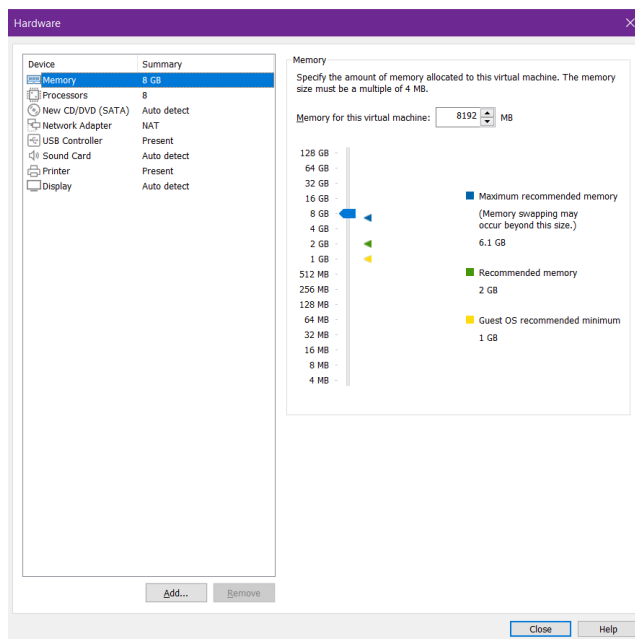


## Practical: 7

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

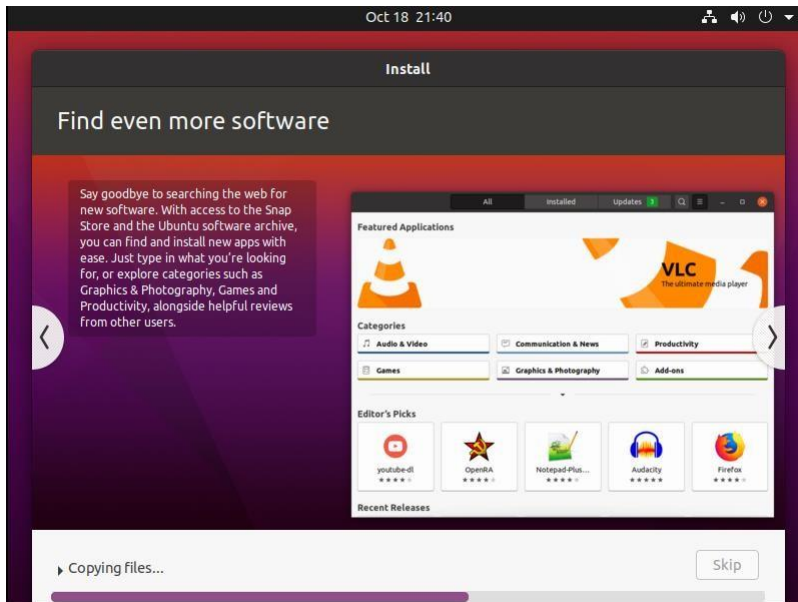


- Step 6: You can go with default hardware and can customize by clicking customize hardware



- 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

## Practical: 7



### Step 1: Install MariaDB Database Server

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

To install MariaDB run the commands below:

⇒ `sudo apt update`

## Practical: 7

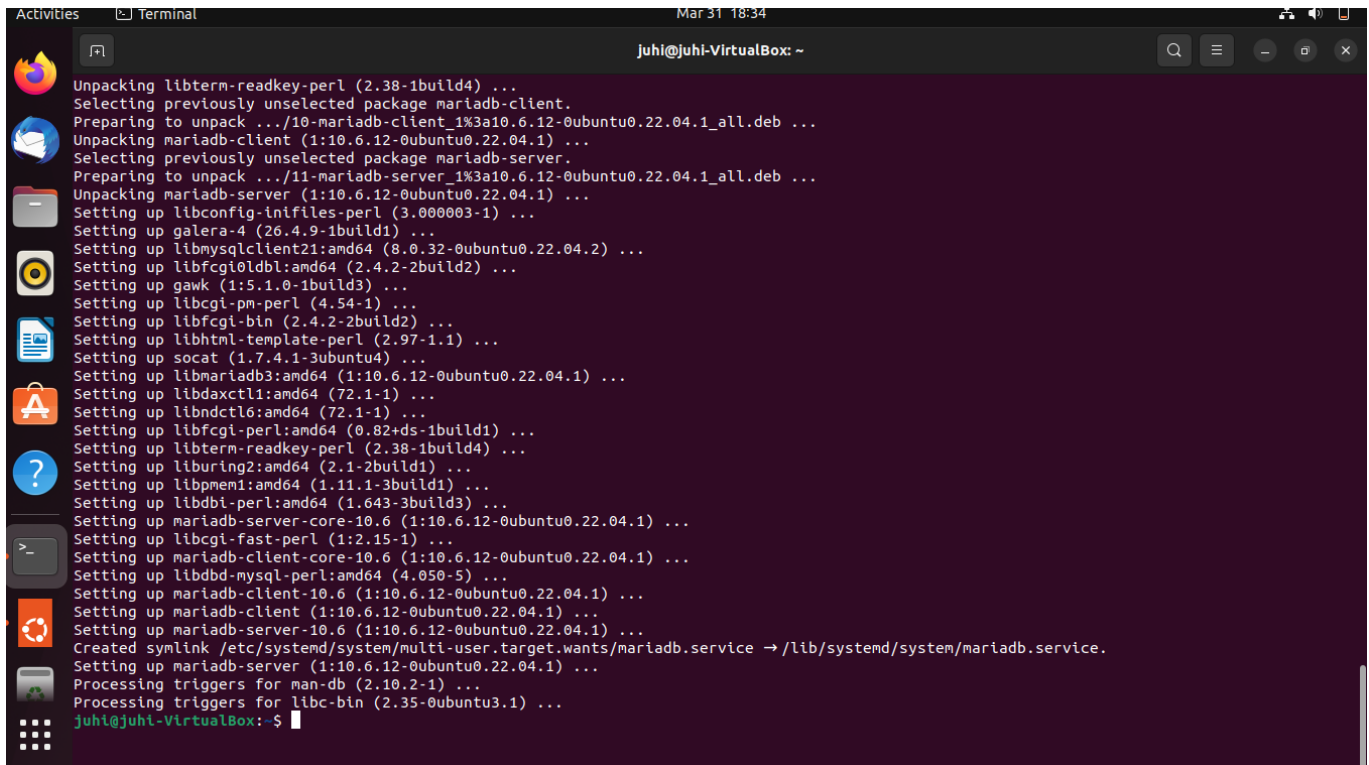
```
Activities Terminal Mar 31 18:29 juhi@juhi-VirtualBox: ~  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
juhi@juhi-VirtualBox:~$ sudo apt update  
[sudo] password for juhi:  
Hit:1 http://in.archive.ubuntu.com/ubuntu jammy InRelease  
Get:2 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]  
Get:3 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]  
Get:4 http://in.archive.ubuntu.com/ubuntu jammy-backports InRelease [108 kB]  
Get:5 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [990 kB]  
Get:6 http://security.ubuntu.com/ubuntu jammy-security/main i386 Packages [279 kB]  
Get:7 http://in.archive.ubuntu.com/ubuntu jammy-updates/main i386 Packages [467 kB]  
Get:8 http://in.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [210 kB]  
Get:9 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 DEP-11 Metadata [101 kB]  
Get:10 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 c-n-f Metadata [13.9 kB]  
Get:11 http://in.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [743 kB]  
Get:12 http://in.archive.ubuntu.com/ubuntu jammy-updates/restricted i386 Packages [27.2 kB]  
Get:13 http://in.archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [115 kB]  
Get:14 http://in.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 c-n-f Metadata [588 B]  
Get:15 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [728 kB]  
Get:16 http://in.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [898 kB]  
Get:17 http://in.archive.ubuntu.com/ubuntu jammy-updates/universe i386 Packages [607 kB]  
Get:18 http://in.archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [180 kB]  
Get:19 http://in.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 DEP-11 Metadata [269 kB]  
Get:20 http://in.archive.ubuntu.com/ubuntu jammy-updates/universe DEP-11 64x64 Icons [267 kB]  
Get:21 http://in.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 c-n-f Metadata [18.5 kB]  
Get:22 http://in.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 DEP-11 Metadata [940 B]  
Get:23 http://in.archive.ubuntu.com/ubuntu jammy-backports/main amd64 Packages [64.0 kB]  
Get:24 http://in.archive.ubuntu.com/ubuntu jammy-backports/main i386 Packages [57.1 kB]  
Get:25 http://in.archive.ubuntu.com/ubuntu jammy-backports/main amd64 DEP-11 Metadata [8,000 B]  
Get:26 http://in.archive.ubuntu.com/ubuntu jammy-backports/main DEP-11 48x48 Icons [7,156 B]  
Get:27 http://in.archive.ubuntu.com/ubuntu jammy-backports/main DEP-11 64x64 Icons [10.7 kB]  
Get:28 http://in.archive.ubuntu.com/ubuntu jammy-backports/main amd64 c-n-f Metadata [388 B]  
Get:29 http://in.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [24.3 kB]  
Get:30 http://in.archive.ubuntu.com/ubuntu jammy-backports/universe i386 Packages [15.6 kB]  
Get:31 http://in.archive.ubuntu.com/ubuntu jammy-backports/universe Translation-en [14.4 kB]  
Get:32 http://in.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 DEP-11 Metadata [12.5 kB]  
Get:33 http://in.archive.ubuntu.com/ubuntu jammy-backports/universe DEP-11 48x48 Icons [6,973 B]
```

⇒ sudo apt install mariadb-server mariadb-client

```
Activities Terminal Mar 31 18:31 juhi@juhi-VirtualBox: ~  
Get:42 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 c-n-f Metadata [588 B]  
Get:43 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [715 kB]  
Get:44 http://security.ubuntu.com/ubuntu jammy-security/universe i386 Packages [519 kB]  
Get:45 http://security.ubuntu.com/ubuntu jammy-security/universe Translation-en [118 kB]  
Get:46 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 DEP-11 Metadata [18.5 kB]  
Get:47 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 c-n-f Metadata [14.1 kB]  
Fetched 8,908 kB in 16s (550 kB/s)  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
314 packages can be upgraded. Run 'apt list --upgradable' to see them.  
juhi@juhi-VirtualBox:~$ sudo apt install mariadb-server mariadb-client  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following packages were automatically installed and are no longer required:  
chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi i965-va-driver intel-media-va-driver libaacs0 libaom3 libass9 libavcodec58 libavformat58  
libavutil56 libbdplus0 libblas3 libbluray2 libbs2b0 libchromaprint1 libcodec2-1.0 libdavid5 libflite1 libgme0 libgsm1  
libgstreamer-plugins-bad1.0-0 libigdgmm12 libllv-0-0 libllvm15 libmfx1 libmysofa1 libnorm1 libopenmpt0 libpgm-5.3-0 libpostproc55  
librabbitmq4 librubberband2 libserd-0-0 libshine3 libsord-0-0 libsratom-0-0 libstr1.4-gnutls libssh-gcrypt-4 libswresample3 libswscale5  
libudfread0 libva-drm2 libva-wayland2 libva-x11-2 libva2 libvdpau1 libvidstab1.1 libx265-199 libxvidcore4 libzimg2 libzmq5 libzvb1-common  
libzvb10 mesa-va-drivers mesa-vdpau-drivers pocketphinx-en-us systemd-hwe-hwdb va-driver-all vdpau-driver-all  
Use 'sudo apt autoremove' to remove them.  
The following additional packages will be installed:  
galera-4 gawk libcgi-fast-perl libcgi-pm-perl libconfig-inifiles-perl libdaxctl1 libdbd-mysql-perl libdbi-perl libfcgi-bin libfcgi-perl  
libfcgi10db1 libhtml-template-perl libmariadb3 libmysqlclient21 libndctl6 libpnm1 libsigsegv2 libterm-readkey-perl liburing2  
mariadb-client-10.6 mariadb-client-core-10.6 mariadb-common mariadb-server-10.6 mariadb-server-core-10.6 mysql-common socat  
Suggested packages:  
gawk-doc libmldbm-perl libnet-daemon-perl libsql-statement-perl libipc-sharedcache-perl mailx mariadb-test  
The following NEW packages will be installed:  
galera-4 gawk libcgi-fast-perl libcgi-pm-perl libconfig-inifiles-perl libdaxctl1 libdbd-mysql-perl libdbi-perl libfcgi-bin libfcgi-perl  
libfcgi10db1 libhtml-template-perl libmariadb3 libmysqlclient21 libndctl6 libpnm1 libsigsegv2 libterm-readkey-perl liburing2  
mariadb-client mariadb-client-10.6 mariadb-client-core-10.6 mariadb-common mariadb-server mariadb-server-10.6 mariadb-server-core-10.6  
mysql-common socat  
0 upgraded, 28 newly installed, 0 to remove and 314 not upgraded.  
Need to get 18.7 MB of archives.  
After this operation, 165 MB of additional disk space will be used.  
Do you want to continue? [Y/n]
```



## Practical: 7



```
Activities Terminal Mar 31 18:34 juhi@juhi-VirtualBox: ~
Unpacking libterm-readkey-perl (2.38-1build4) ...
Selecting previously unselected package mariadb-client.
Preparing to unpack .../10-mariadb-client_1%3a10.6.12-0ubuntu0.22.04.1_all.deb ...
Unpacking mariadb-client (1:10.6.12-0ubuntu0.22.04.1) ...
Selecting previously unselected package mariadb-server.
Preparing to unpack .../11-mariadb-server_1%3a10.6.12-0ubuntu0.22.04.1_all.deb ...
Unpacking mariadb-server (1:10.6.12-0ubuntu0.22.04.1) ...
Setting up libconfig-inifiles-perl (3.000003-1) ...
Setting up galera-4 (26.4.9-1build1) ...
Setting up libmysqlclient21:amd64 (8.0.32-0ubuntu0.22.04.2) ...
Setting up libfcgi0ldbl:amd64 (2.4.2-2build2) ...
Setting up gawk (1:5.1.0-1build3) ...
Setting up libfcgi-pm-perl (4.54-1) ...
Setting up libfcgi-bin (2.4.2-2build2) ...
Setting up libhtml-template-perl (2.97-1.1) ...
Setting up socat (1.7.4.1-3ubuntu4) ...
Setting up libmariadb3:amd64 (1:10.6.12-0ubuntu0.22.04.1) ...
Setting up libdaxctl1:amd64 (72.1-1) ...
Setting up libndctl6:amd64 (72.1-1) ...
Setting up libfcgi-perl:amd64 (0.82+ds-1build1) ...
Setting up libterm-readkey-perl (2.38-1build4) ...
Setting up liburing2:amd64 (2.1-2build1) ...
Setting up libpmem1:amd64 (1.11.1-3build1) ...
Setting up libdbi-perl:amd64 (1.643-3build3) ...
Setting up mariadb-server-core-10.6 (1:10.6.12-0ubuntu0.22.04.1) ...
Setting up libfcgi-fast-perl (1:2.15-1) ...
Setting up mariadb-client-core-10.6 (1:10.6.12-0ubuntu0.22.04.1) ...
Setting up libdbd-mysql-perl:amd64 (4.050-5) ...
Setting up mariadb-client-10.6 (1:10.6.12-0ubuntu0.22.04.1) ...
Setting up mariadb-client (1:10.6.12-0ubuntu0.22.04.1) ...
Setting up mariadb-server-10.6 (1:10.6.12-0ubuntu0.22.04.1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/mariadb.service → /lib/systemd/system/mariadb.service.
Setting up mariadb-server (1:10.6.12-0ubuntu0.22.04.1) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for libc-bin (2.35-0ubuntu3.1) ...
juhi@juhi-VirtualBox:~$
```

After installing MariaDB, the commands below can be used to stop, start and enable MariaDB service to always 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 and disallowing 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 Enter

Set root password? [Y/n]: Y

New password: Enter password

Re-enter new password:

Repeat password

Remove anonymous users? [Y/n]: Y

## Practical: 7

Disallow root login

remotely? [Y/n]: Y

Remove test database and access to it?

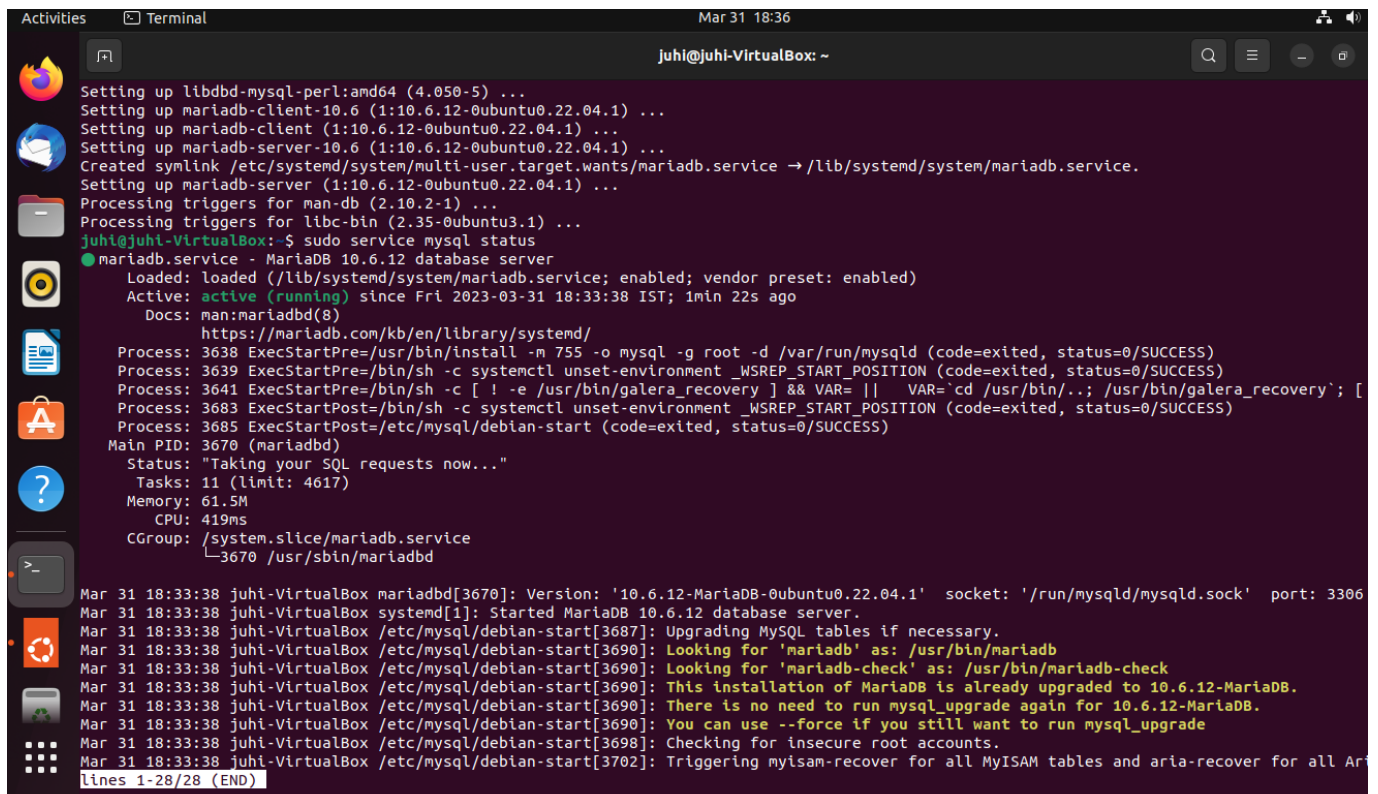
[Y/n]: Y Reload privilege tables now?

[Y/n]: Y

Restart MariaDB server

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

➤ `sudo systemctl status mariadb`



```
Setting up libdbd-mysql-perl:amd64 (4.050-5) ...
Setting up mariadb-client-10.6 (1:10.6.12-0ubuntu0.22.04.1) ...
Setting up mariadb-client (1:10.6.12-0ubuntu0.22.04.1) ...
Setting up mariadb-server-10.6 (1:10.6.12-0ubuntu0.22.04.1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/mariadb.service → /lib/systemd/system/mariadb.service.
Setting up mariadb-server (1:10.6.12-0ubuntu0.22.04.1) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for libc-bin (2.35-0ubuntu3.1) ...
juhi@juhi-VirtualBox: ~$ sudo service mysql status
● mariadb.service - MariaDB 10.6.12 database server
   Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2023-03-31 18:33:38 IST; 1min 22s ago
     Docs: man:mariadb(8)
           https://mariadb.com/kb/en/library/systemd/
   Process: 3638 ExecStartPre=/usr/bin/install -m 755 -o mysql -g root -d /var/run/mysql (code=exited, status=0/SUCCESS)
   Process: 3639 ExecStartPre=/bin/sh -c systemctl unset-environment _WSREP_START_POSITION (code=exited, status=0/SUCCESS)
   Process: 3641 ExecStartPre=/bin/sh -c [ ! -e /usr/bin/galera_recovery ] && VAR= || VAR='cd /usr/bin/.; /usr/bin/galera_recovery'; [
   Process: 3683 ExecStartPost=/bin/sh -c systemctl unset-environment _WSREP_START_POSITION (code=exited, status=0/SUCCESS)
   Process: 3685 ExecStartPost=/etc/mysql/debian-start (code=exited, status=0/SUCCESS)
  Main PID: 3670 (mariadb)
    Status: "Taking your SQL requests now..."
     Tasks: 11 (limit: 4617)
    Memory: 61.5M
       CPU: 419ms
    CGroup: /system.slice/mariadb.service
            └─3670 /usr/sbin/mariadb

Mar 31 18:33:38 juhi-VirtualBox mariadb[3670]: Version: '10.6.12-MariaDB-0ubuntu0.22.04.1' socket: '/run/mysql/mysql.sock' port: 3306
Mar 31 18:33:38 juhi-VirtualBox systemd[1]: Started MariaDB 10.6.12 database server.
Mar 31 18:33:38 juhi-VirtualBox /etc/mysql/debian-start[3687]: Upgrading MySQL tables if necessary.
Mar 31 18:33:38 juhi-VirtualBox /etc/mysql/debian-start[3690]: Looking for 'mariadb' as: /usr/bin/mariadb
Mar 31 18:33:38 juhi-VirtualBox /etc/mysql/debian-start[3690]: Looking for 'mariadb-check' as: /usr/bin/mariadb-check
Mar 31 18:33:38 juhi-VirtualBox /etc/mysql/debian-start[3690]: This installation of MariaDB is already upgraded to 10.6.12-MariaDB.
Mar 31 18:33:38 juhi-VirtualBox /etc/mysql/debian-start[3690]: There is no need to run mysql_upgrade again for 10.6.12-MariaDB.
Mar 31 18:33:38 juhi-VirtualBox /etc/mysql/debian-start[3690]: You can use --force if you still want to run mysql_upgrade
Mar 31 18:33:38 juhi-VirtualBox /etc/mysql/debian-start[3698]: Checking for insecure root accounts.
Mar 31 18:33:38 juhi-VirtualBox /etc/mysql/debian-start[3702]: Triggering myisam-recover for all MyISAM tables and aria-recover for all Ar
Lines 1-28/28 (END)
```

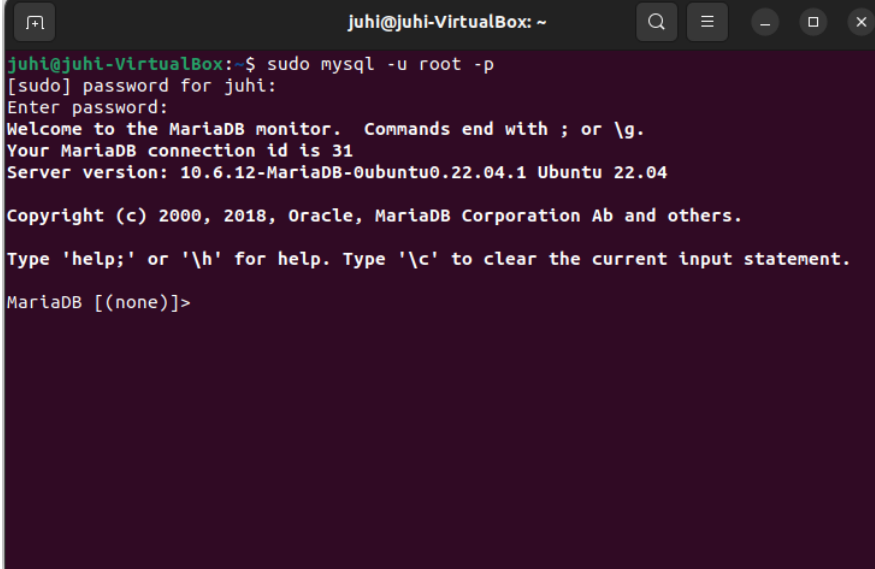
### Step 2: Create OpenNebula Database

Now that you've install all the packages that are required, continue below to start configuring the servers. 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 the root password you created above.

➤ `sudo mysql -u root -p`

## Practical: 7

A terminal window titled 'juhi@juhi-VirtualBox: ~' showing the installation and login process for MariaDB. The user runs 'sudo mysql -u root -p', enters the password, and is greeted by the MariaDB monitor. The monitor displays the welcome message, connection ID (31), server version (10.6.12-MariaDB-0ubuntu0.22.04.1), and copyright information. The prompt is 'MariaDB [(none)]>'.

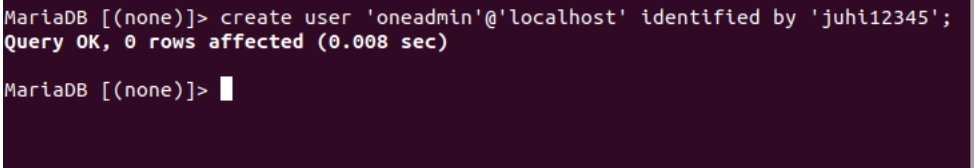
```
juhi@juhi-VirtualBox: ~  
juhi@juhi-VirtualBox:~$ sudo mysql -u root -p  
[sudo] password for juhi:  
Enter password:  
Welcome to the MariaDB monitor.  Commands end with ; or \g.  
Your MariaDB connection id is 31  
Server version: 10.6.12-MariaDB-0ubuntu0.22.04.1 Ubuntu 22.04  
  
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
MariaDB [(none)]>
```

Then create a database called opennebula

⇒ CREATE DATABASE opennebula;

Create a database user called opennebulauser with new password

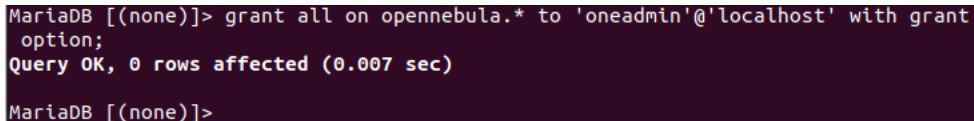
⇒ CREATE USER 'opennebulauser'@'localhost' IDENTIFIED BY 'new\_password\_here';

A terminal window showing the execution of the 'CREATE USER' command. The user enters 'create user 'oneadmin'@'localhost' identified by 'juhi12345';' and the output is 'Query OK, 0 rows affected (0.008 sec)'. The prompt is 'MariaDB [(none)]>'.

```
MariaDB [(none)]> create user 'oneadmin'@'localhost' identified by 'juhi12345';  
Query OK, 0 rows affected (0.008 sec)  
  
MariaDB [(none)]>
```

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

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

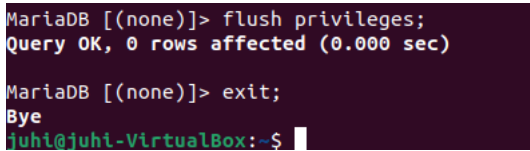
A terminal window showing the execution of the 'GRANT ALL' command. The user enters 'grant all on opennebula.\* to 'oneadmin'@'localhost' with grant option;' and the output is 'Query OK, 0 rows affected (0.007 sec)'. The prompt is 'MariaDB [(none)]>'.

```
MariaDB [(none)]> grant all on opennebula.* to 'oneadmin'@'localhost' with grant  
option;  
Query OK, 0 rows affected (0.007 sec)  
  
MariaDB [(none)]>
```

Finally, save your changes and exit.

⇒ FLUSH PRIVILEGES;

⇒ EXIT;

A terminal window showing the execution of 'flush privileges;' and 'exit;'. The output for 'flush privileges;' is 'Query OK, 0 rows affected (0.000 sec)'. The output for 'exit;' is 'Bye'. The prompt returns to the shell 'juhi@juhi-VirtualBox:~\$'.

```
MariaDB [(none)]> flush privileges;  
Query OK, 0 rows affected (0.000 sec)  
  
MariaDB [(none)]> exit;  
Bye  
juhi@juhi-VirtualBox:~$
```

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

## Practical: 7

### Step 3: Install OpenNebula

By default, OpenNebula isn't available via Ubuntu default repositories. If you'd like to install it in Ubuntu, 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
```

## Practical: 7

```
ne
The key's randomart image is:
+---[RSA 2048]---+
  =*+=B0++o.|
  ..E+*X++.|
  o o 0++=+
  +  =..+. 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 other required 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 start and enable OpenNebula services.

☞ `sudo systemctl start opennebula`  
☞ `sudo systemctl enable opennebula`

## Practical: 7

---

- ☞ `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.service 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-sunstone.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 is stored in the credential below:



## Practical: 7

☞ `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/one_monitord -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](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 and browse to the server hostname or IP address followed by port 9869

`http://localhost:9869`

## Practical: 7



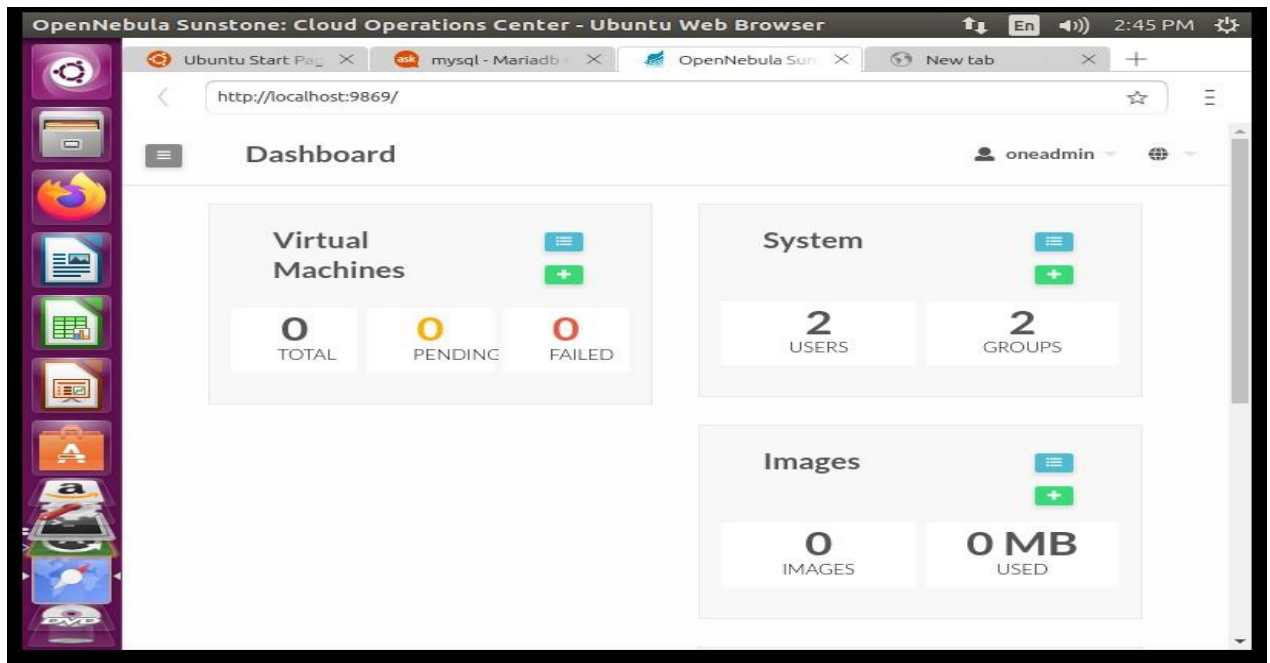
Login with the credential found in the file above.



That should redirect you to OpenNebula dashboard.



## Practical: 7



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 add KVM 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
```

## Practical: 7

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.opennebula.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`

```
The following NEW packages will be installed:
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 libsd1.2debian 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]
```

## Practical: 7

---

```
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 edit 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.

## Practical: 7

Restart the client services

- 👁️ `sudo systemctl restart libvirtd`
- 👁️ `sudo systemctl restart libvirt-bin`

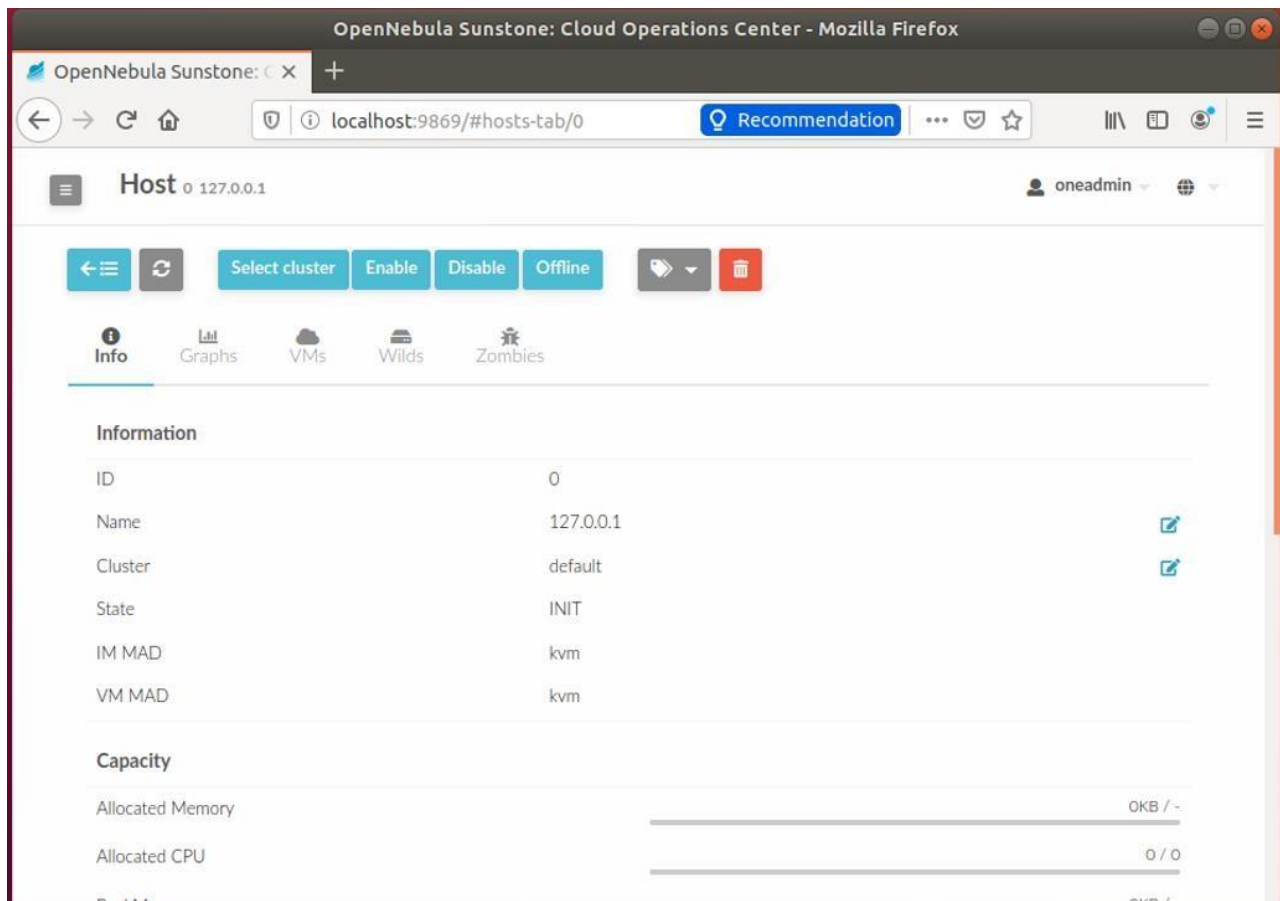
Communication is done via SSH. So you'll want install SSH and enable password-less SSH authentication between the OpenNebula host and the KVM node.

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.



## Practical: 7

---

### 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 the Front-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 a temporary password to **oneadmin** in all the hosts and copy the directory from the Front-end:

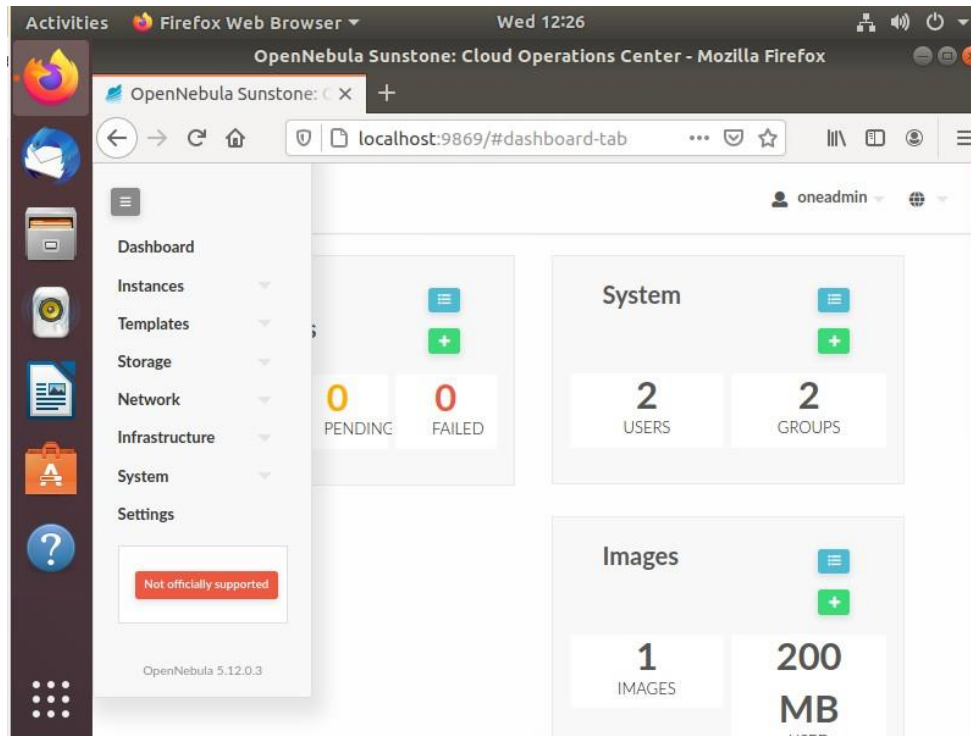
```
⇒ scp -rp /var/lib/one/.ssh <node1>:/var/lib/one/
```

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

```
⇒ ssh <node1>
⇒ ssh <frontend>
⇒ exit
```

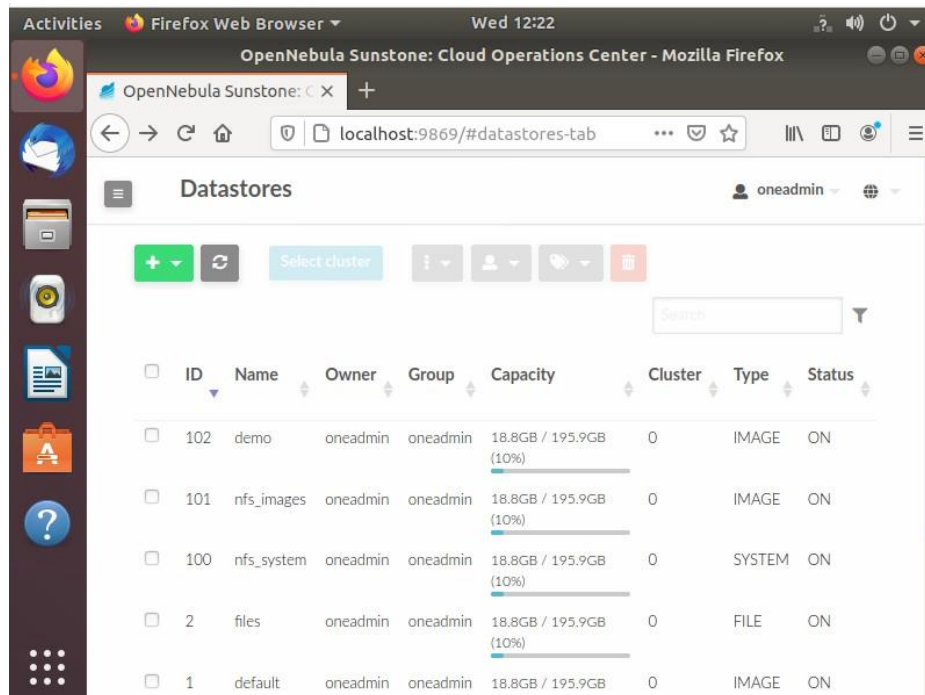
- On dashboard you will see the menu bar from that you can see your datastores from yourStorage menu bar.

## Practical: 7

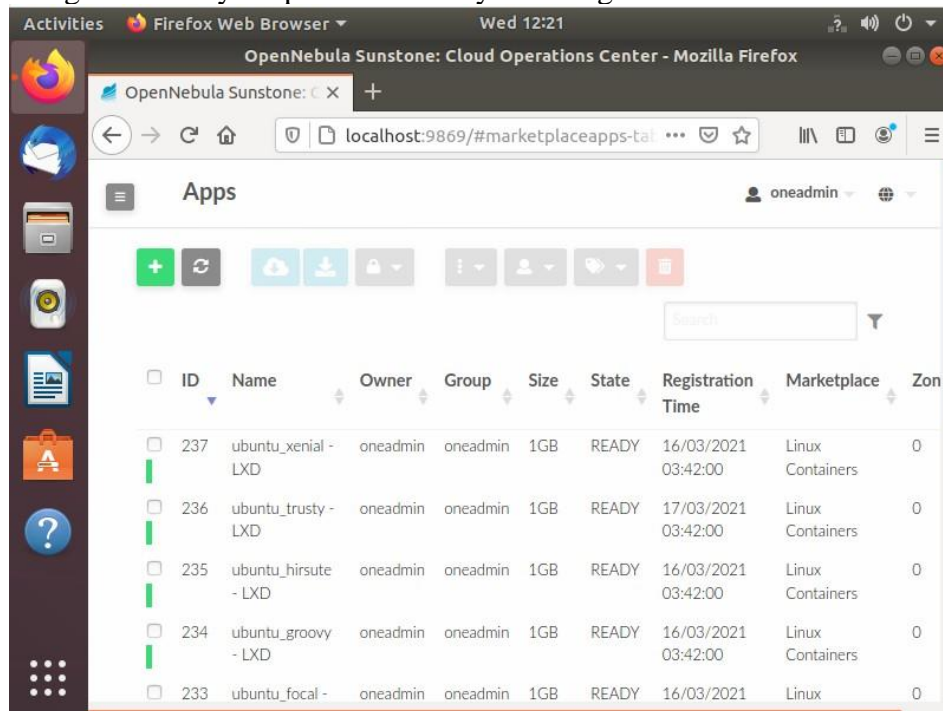


- You must have at least one datastore that can store your data you can create one from CLI as well as from Sunstone.

## Practical: 7



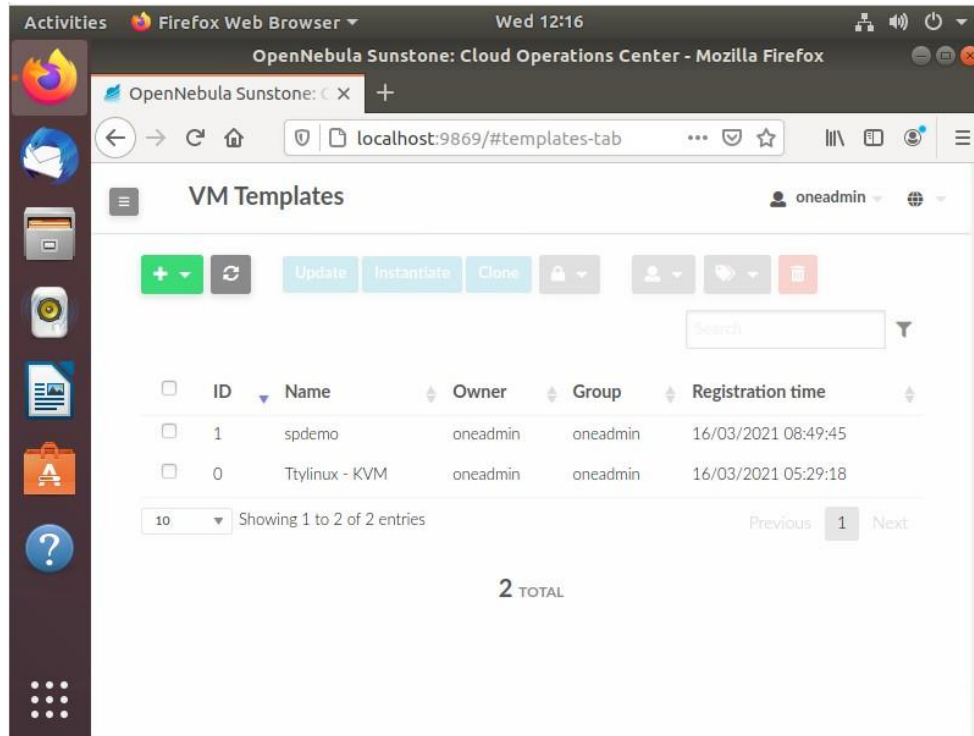
- Now, we have to add the apps to be run on our VMs. You can download the image or add to your private cloud by selecting one.



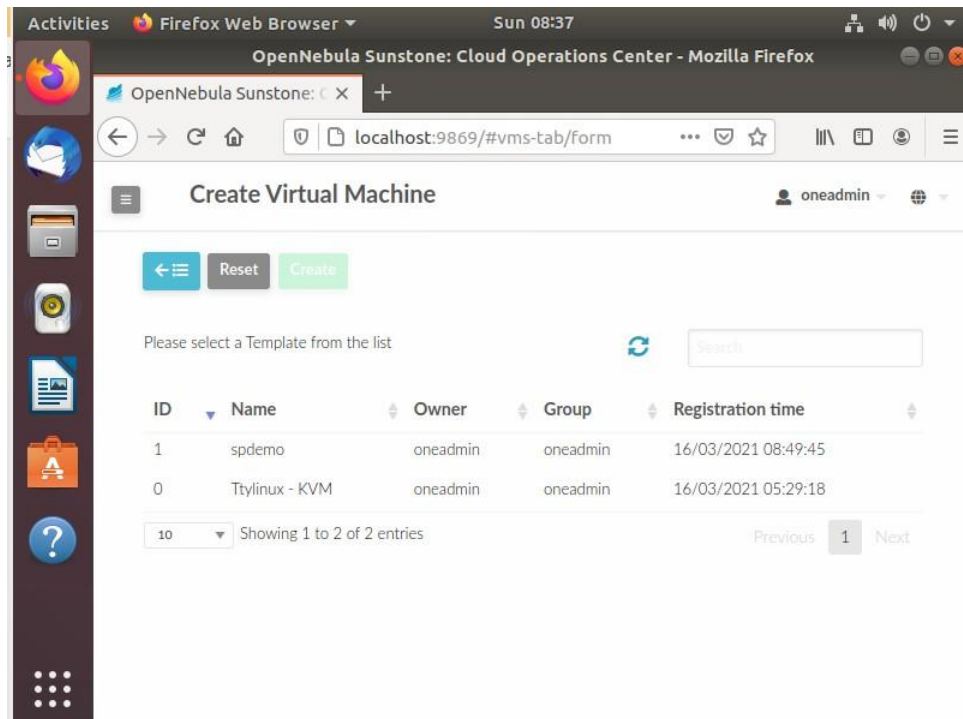
- In VM Templates you can see what apps you have added in your cloud and else you may create your custom templates from the same.



## Practical: 7



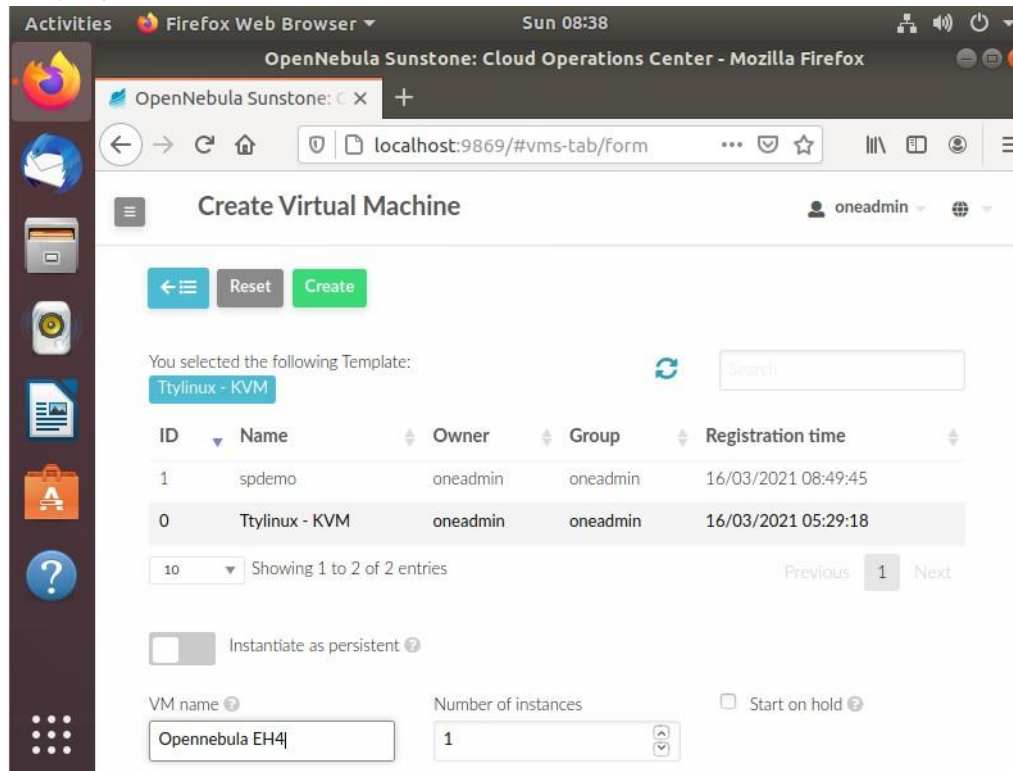
- We are done with all the thing. Now, we are good to create VM. Click on create VM.



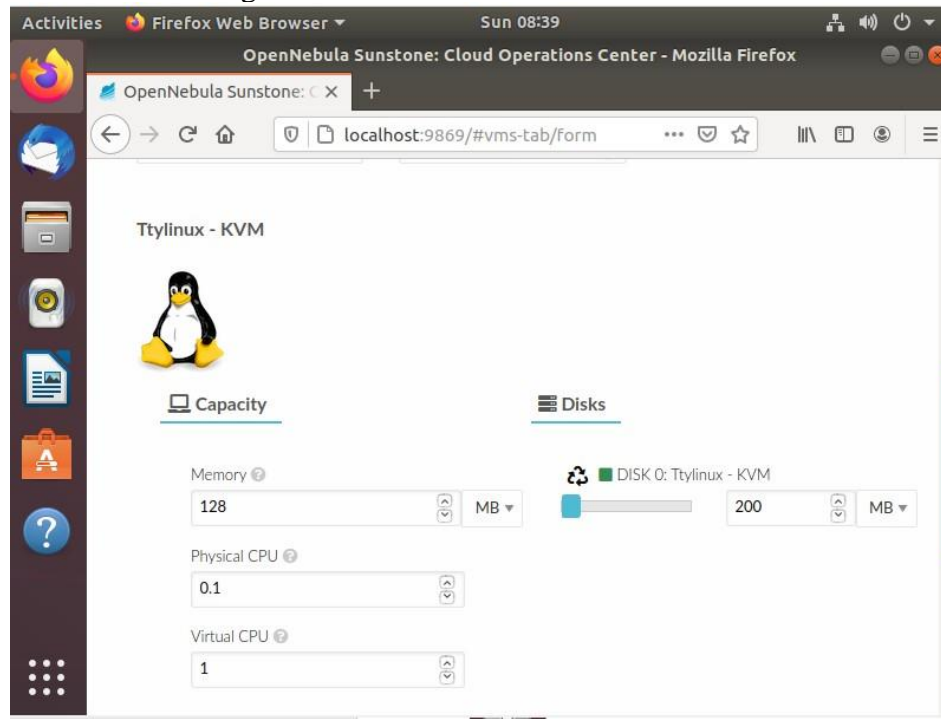


## Practical: 7

- 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 be integer.



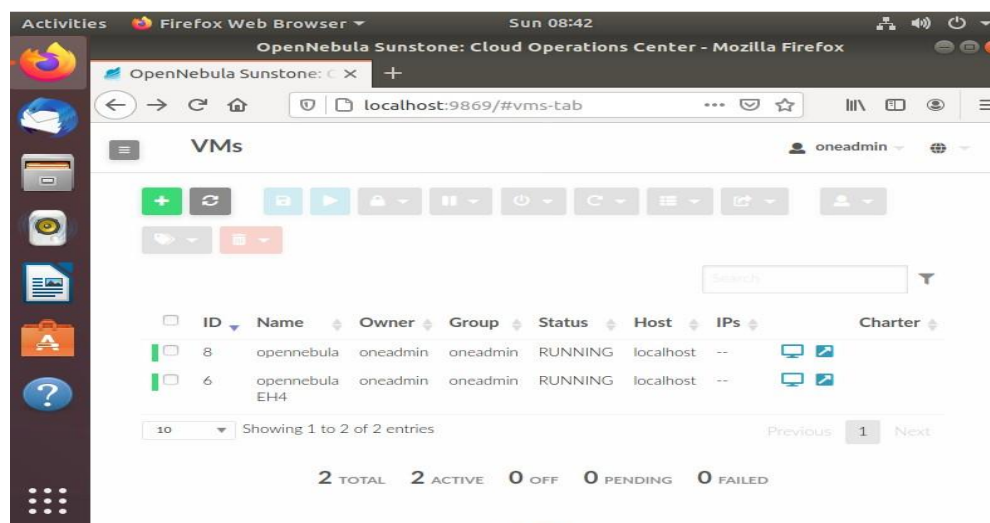
- From the same page you have many option you can select and change like on

## Practical: 7

which datastore you wish to store your data, user, group and many more...

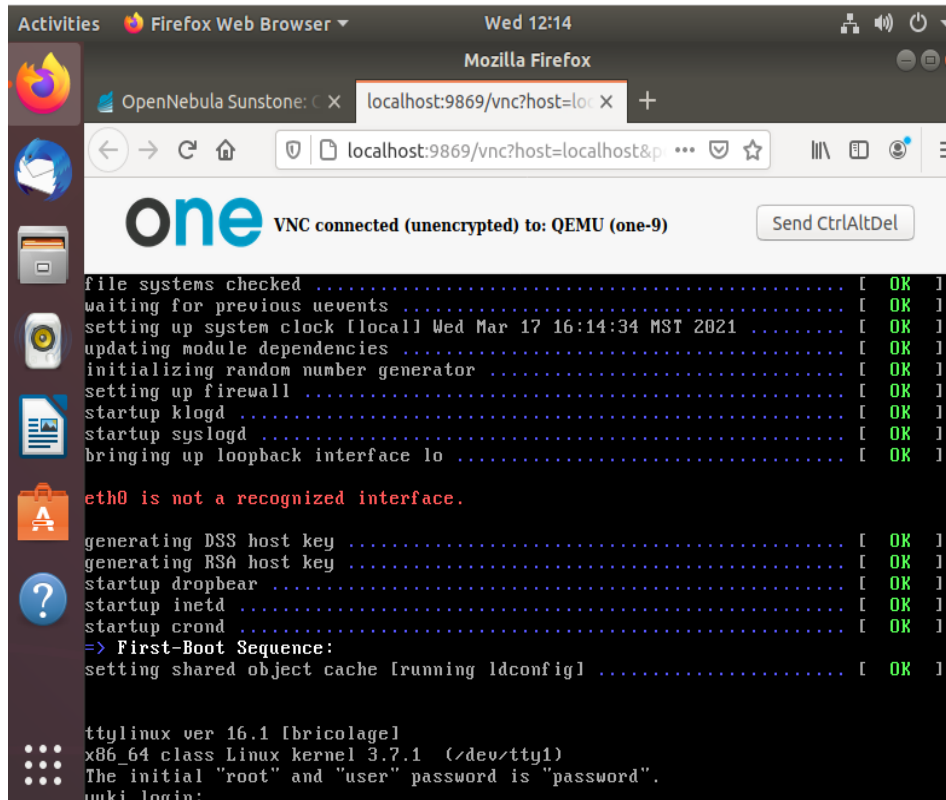
- ✓ Network
- ✓ Instantiate as different User / Group
- ✓ Deploy VM in a specific Host
- ✓ Deploy VM in a specific Datastore
- ✓ Schedule Actions for VM
- ✓ Associate VM to a VM Group
- ✓ vCenter Deployment

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



## Practical: 7

- 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 we can run machine.