

## Practical 5

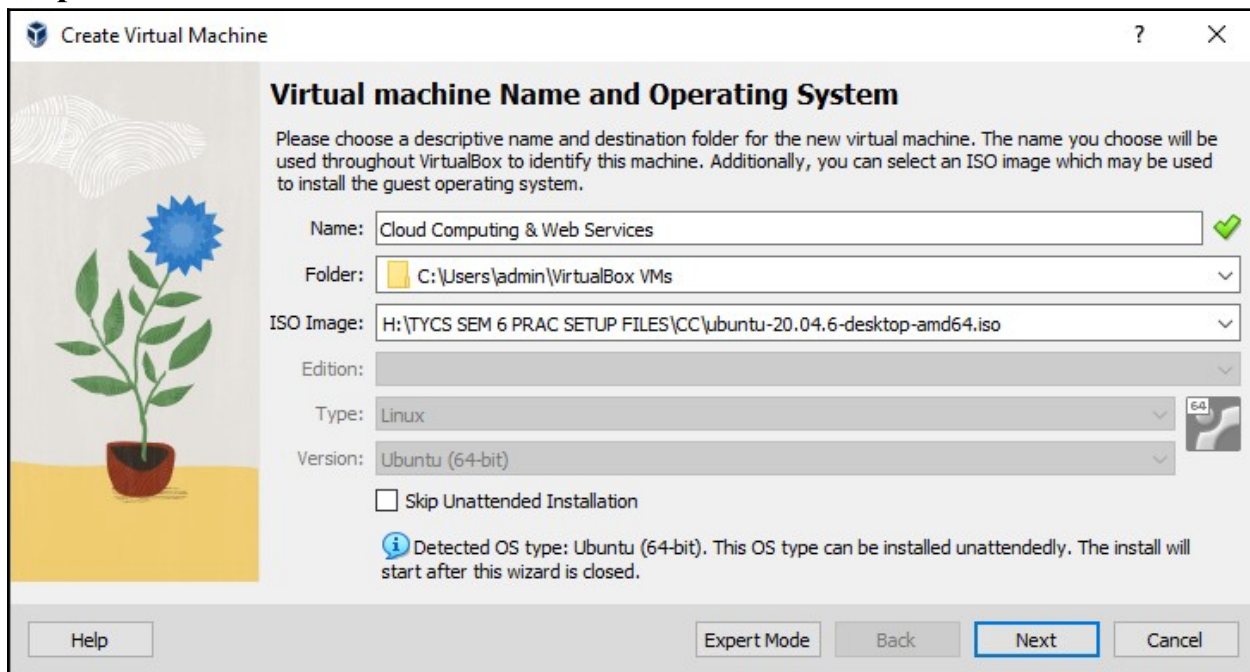
### Aim: Installation and Configuration of virtualization using KVM.


#### Theory :

- KVM (for Kernel-based Virtual Machine) is a full virtualization solution for Linux on x86 hardware containing virtualization extensions (Intel VT or AMD-V). It consists of a loadable kernel module, `kvm` that provides the core virtualization infrastructure and a processor specific module, `kvm-intel` or `kvm-amd`.
- Using KVM, one can run multiple virtual machines running unmodified Linux or Windows images. Each virtual machine has private virtualized hardware: a network card, disk, graphics adapter, etc.
- KVM is open source software. The kernel component of KVM is included in mainline Linux, as of 2.6.20. The user space component of KVM is included in mainline QEMU, as of 1.3.

**Requirements:** Virtual Box, Ubuntu ISO file (latest version)

#### Steps:





## Create Virtual Machine

### Unattended Guest OS Install Setup

You can configure the unattended guest OS install by modifying username, password, and hostname. Additionally you can enable guest additions install. For Microsoft Windows guests it is possible to provide a product key.

Username and Password

Username:  ✓

Password:  ✕

Repeat Password:  ✕

Additional Options

Product Key:

Hostname:  ✓

Domain Name:


☐ Install in Background

☐ Guest Additions

Guest Additions ISO:

Help

Back Next Cancel



## Create Virtual Machine

### Hardware

You can modify virtual machine's hardware by changing amount of RAM and virtual CPU count. Enabling EFI is also possible.

Base Memory:  4096 MB

4 MB 16384 MB

Processors:  3

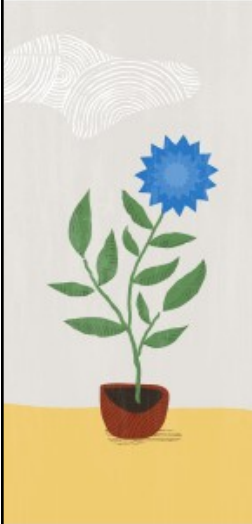
1 CPU 8 CPUs

☐ Enable EFI (special OSes only)

Help

Back Next Cancel

Create Virtual Machine



### Virtual Hard disk

If you wish you can add a virtual hard disk to the new machine. You can either create a new hard disk file or select an existing one. Alternatively you can create a virtual machine without a virtual hard disk.

☒ Create a Virtual Hard Disk Now

Disk Size:

4.00 MB

2.00 TB

25.00 GB

☐ Pre-allocate Full Size

☐ Use an Existing Virtual Hard Disk File

Windows 7.vdi (Normal, 32.00 GB)

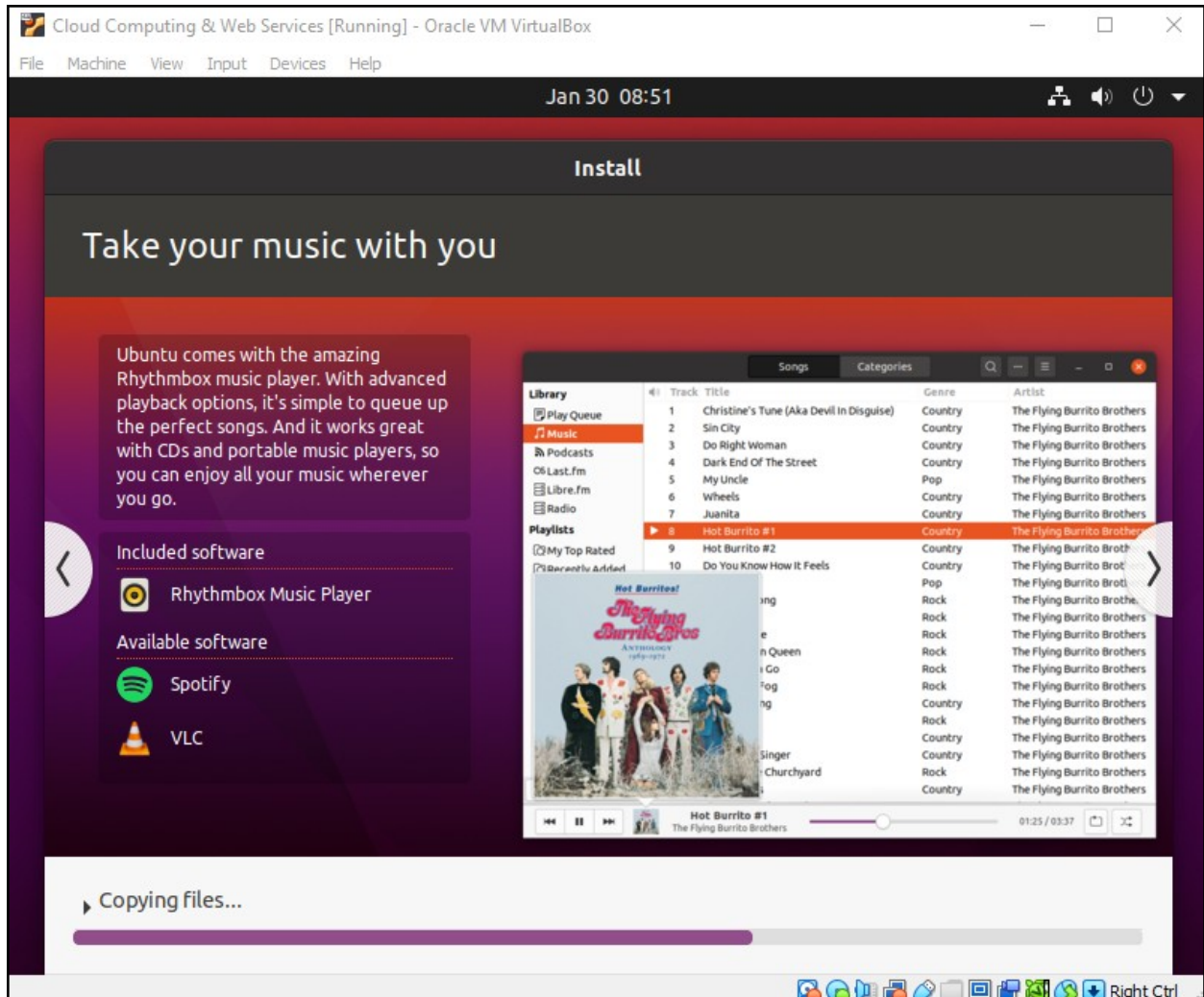
☐ Do Not Add a Virtual Hard Disk

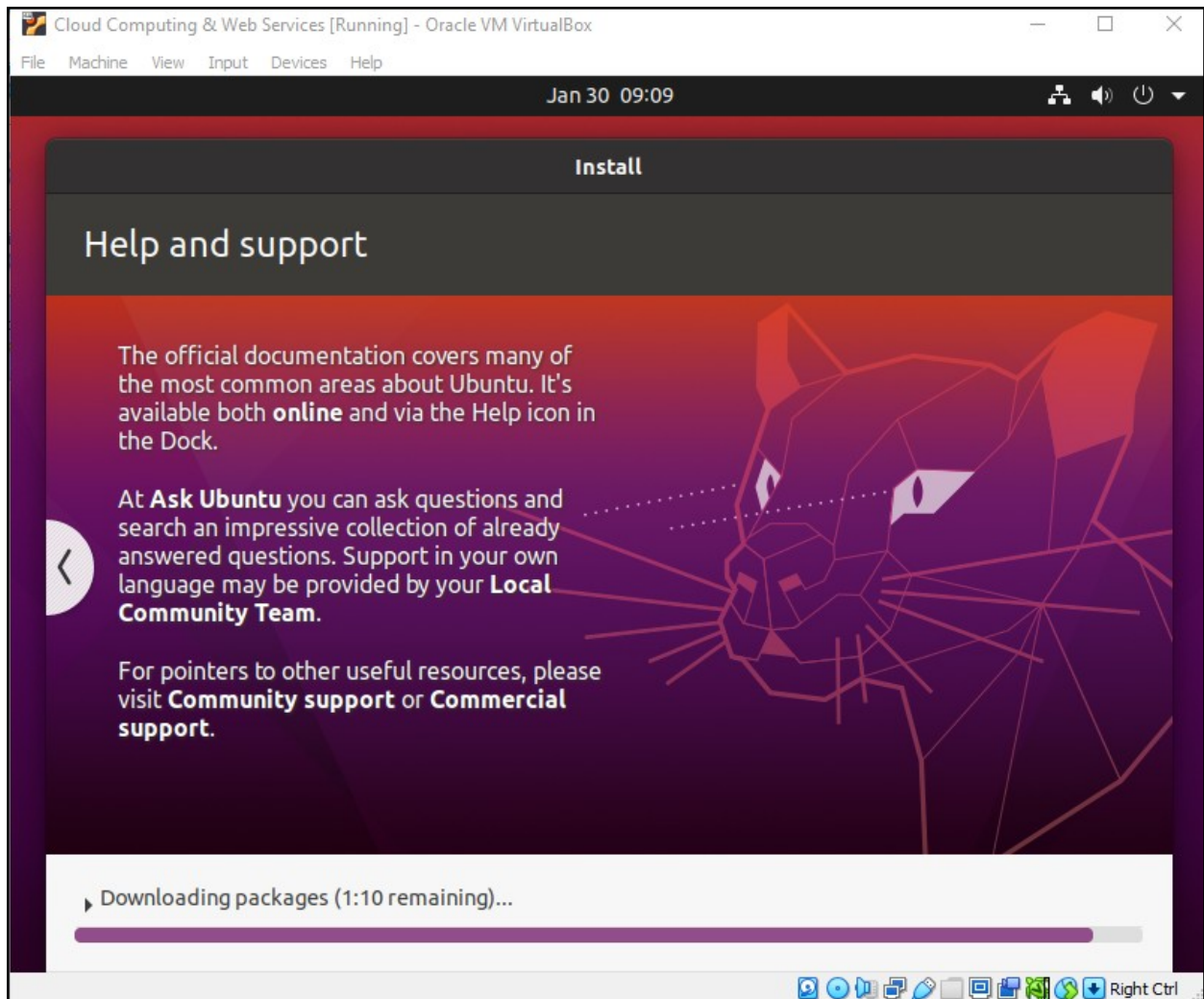
Help

Back

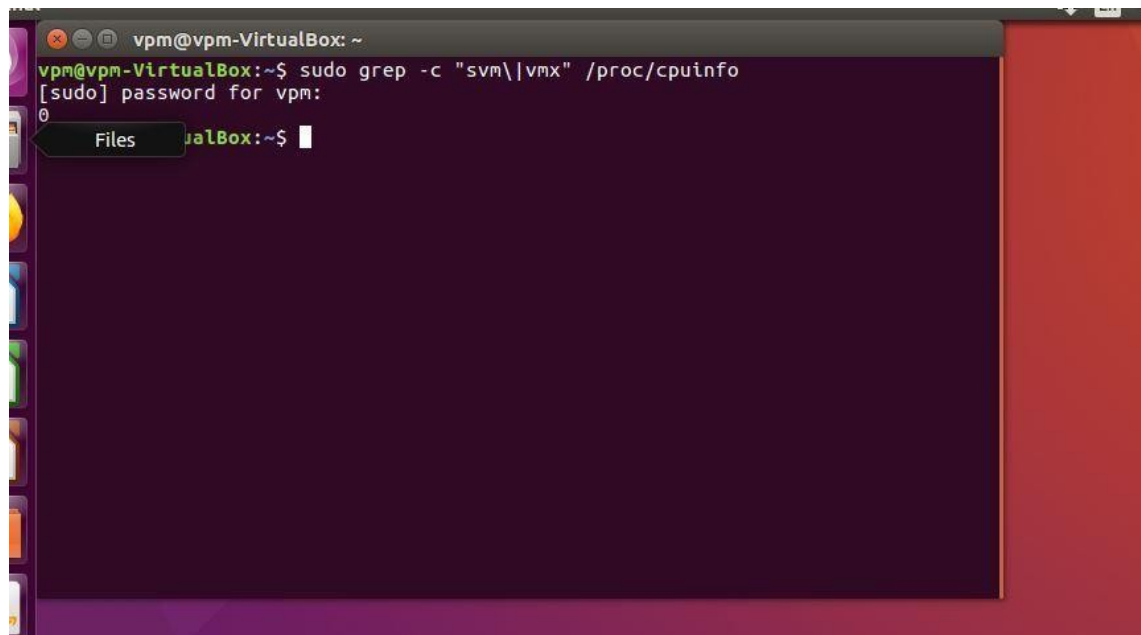
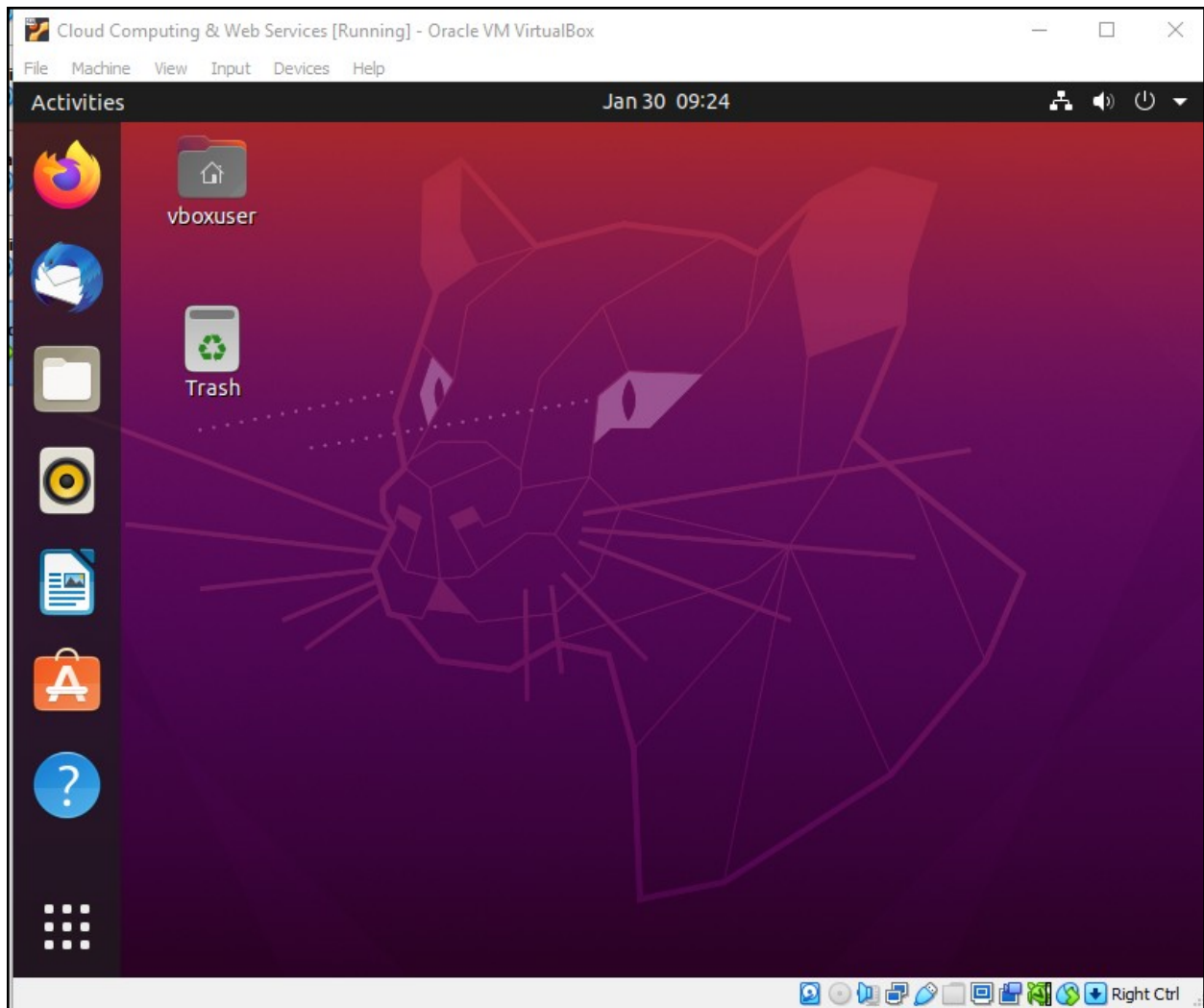
Next

Cancel





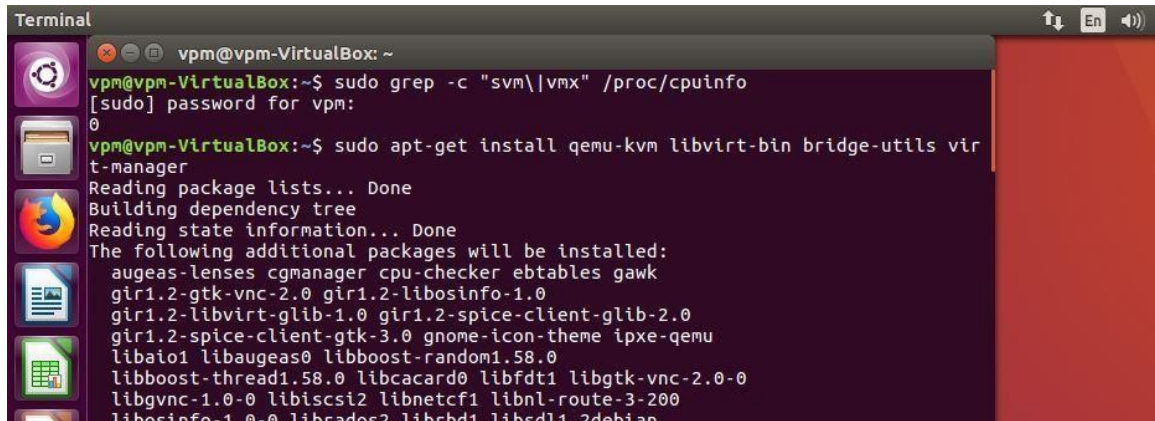




`sudo apt-get install qemu-kvm libvirt-bin bridge-utils virt-manager`

OR

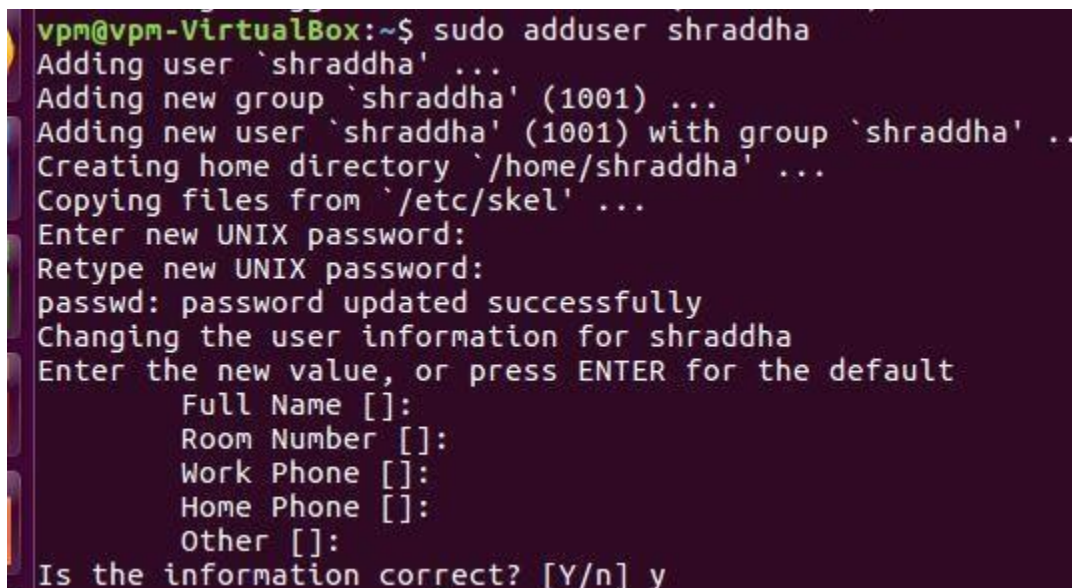
`sudo apt-get install qemu-kvm libvirt-daemon-system libvirt-clients bridge-utils`

A terminal window titled 'Terminal' with a dark background. The prompt is 'vpm@vpm-VirtualBox: ~'. The user enters 'sudo grep -c "svm|vmx" /proc/cpuinfo'. The prompt changes to '[sudo] password for vpm:' and the user enters '0'. Then the user enters 'sudo apt-get install qemu-kvm libvirt-bin bridge-utils virt-manager'. The terminal shows the progress: 'Reading package lists... Done', 'Building dependency tree', 'Reading state information... Done'. It then lists additional packages to be installed: 'augeas-lenses cgmanager cpu-checker ebttables gawk gir1.2-gtk-vnc-2.0 gir1.2-libosinfo-1.0 gir1.2-libvirt-glib-1.0 gir1.2-spice-client-glib-2.0 gir1.2-spice-client-gtk-3.0 gnome-icon-theme ipxe-qemu libaio1 libaugeas0 libboost-random1.58.0 libboost-thread1.58.0 libcacard0 libfdt1 libgtk-vnc-2.0-0 libgvnc-1.0-0 libiscsi2 libnetcf1 libnl-route-3-200 libosinfo-1.0-0 librados2 librbdl1 libstd1.2debian'.

```
Terminal
vpm@vpm-VirtualBox: ~
vpm@vpm-VirtualBox:~$ sudo grep -c "svm|vmx" /proc/cpuinfo
[sudo] password for vpm:
0
vpm@vpm-VirtualBox:~$ sudo apt-get install qemu-kvm libvirt-bin bridge-utils vir
t-manager
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
augeas-lenses cgmanager cpu-checker ebttables gawk
gir1.2-gtk-vnc-2.0 gir1.2-libosinfo-1.0
gir1.2-libvirt-glib-1.0 gir1.2-spice-client-glib-2.0
gir1.2-spice-client-gtk-3.0 gnome-icon-theme ipxe-qemu
libaio1 libaugeas0 libboost-random1.58.0
libboost-thread1.58.0 libcacard0 libfdt1 libgtk-vnc-2.0-0
libgvnc-1.0-0 libiscsi2 libnetcf1 libnl-route-3-200
libosinfo-1.0-0 librados2 librbdl1 libstd1.2debian
```

If the above commands give error then run  
`sudo apt-get update`

`Sudo adduser <yourname>`

A terminal window showing the execution of 'sudo adduser shraddha'. The output shows the user being added, a new group 'shraddha' (1001) being created, and a new user 'shraddha' (1001) being created with that group. It then shows the creation of a home directory and copying files from '/etc/skel'. The user is prompted to enter a new UNIX password, which is then confirmed. Finally, the user information is changed, and the user is prompted to enter a new value for various fields (Full Name, Room Number, Work Phone, Home Phone, Other), all of which are left blank. The process concludes with 'Is the information correct? [Y/n] y'.

```
vpm@vpm-VirtualBox:~$ sudo adduser shraddha
Adding user `shraddha' ...
Adding new group `shraddha' (1001) ...
Adding new user `shraddha' (1001) with group `shraddha' ..
Creating home directory `/home/shraddha' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for shraddha
Enter the new value, or press ENTER for the default
    Full Name []:
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n] y
```

`Sudo adduser <yourname> libvirtd`

```
Is the information correct: [Y/n] y
vpm@vpm-VirtualBox:~$ sudo adduser shraddha libvirt
Adding user `shraddha' to group `libvirt' ...
Adding user shraddha to group libvirt
Done.
vpm@vpm-VirtualBox:~$
```

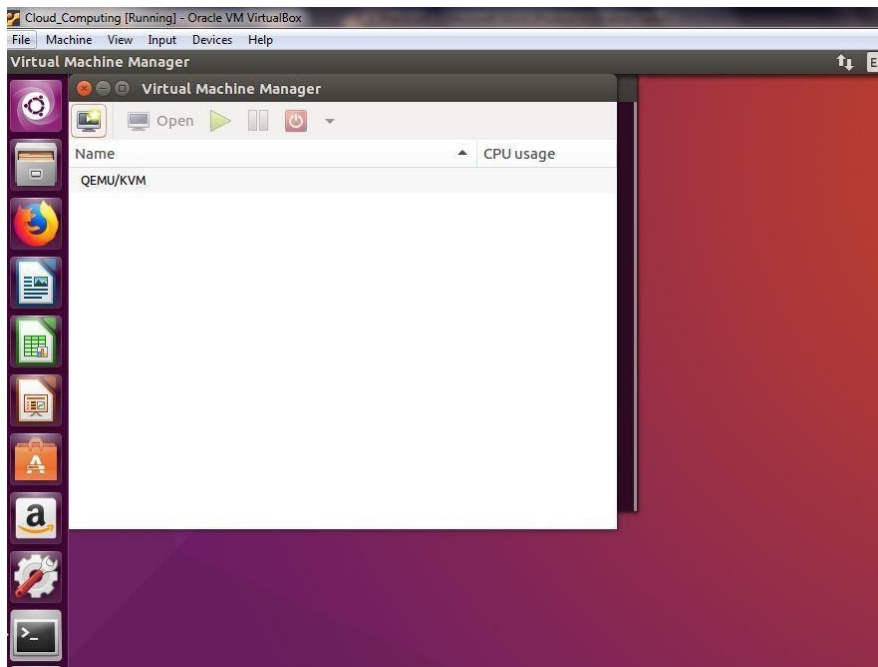
Sudo su <username>

virsh -c qemu:///system list

```
shraddha@vpm-VirtualBox:~$ virsh -c qemu:///system list
Id      Name                                     State
-----
shraddha@vpm-VirtualBox:~$
```

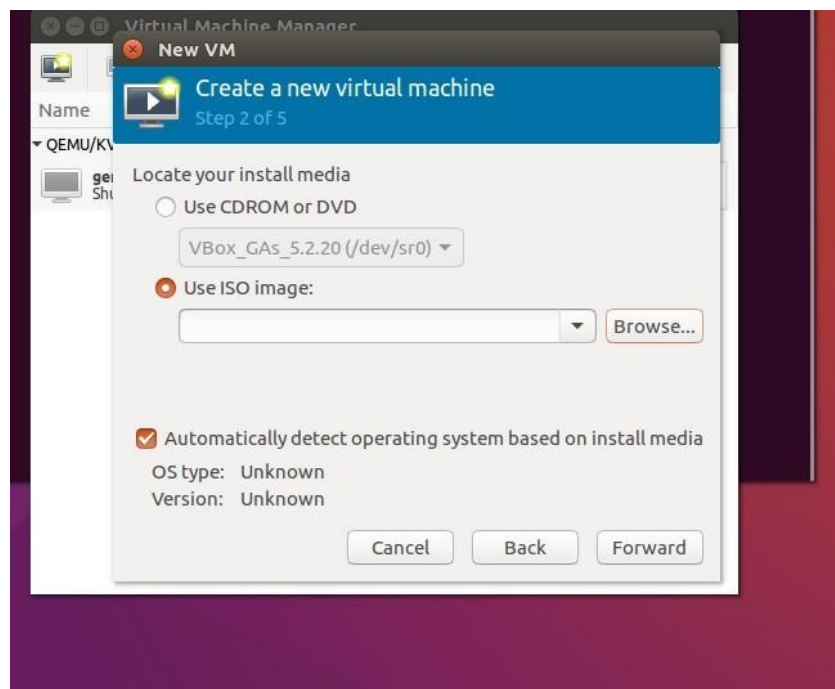
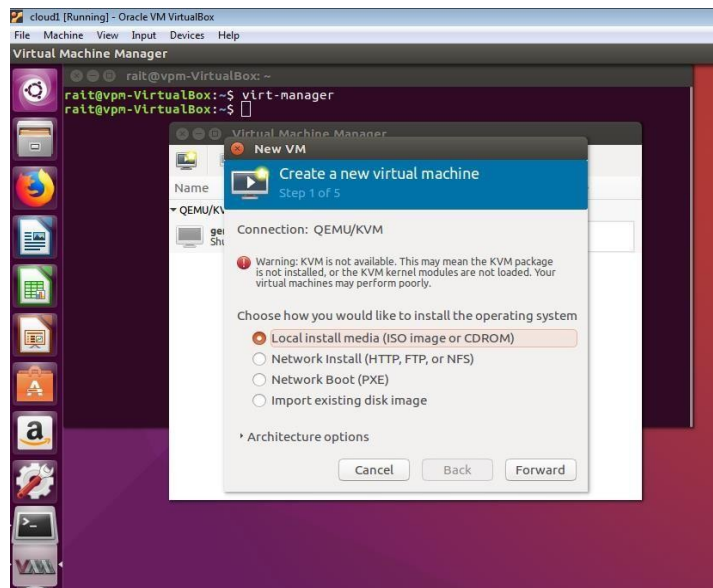
Virt-manager

```
shraddha@vpm-VirtualBox:~$ virt-manager
shraddha@vpm-VirtualBox:~$
```



Click on new virtual machine in qemu and perform the following step





## Final output

