EXPERIMENT NO. 2

Aim: To study and implement Hosted Virtualization using VirtualBox& KVM.

Objective: To know the concept of Virtualization along with their types, structures and mechanisms. This experiment should have demonstration of creating and running Virtual machines inside hosted hypervisors like VirtualBox and KVM with their comparison based on various virtualization parameters.

Theory:

Hosted virtualization offers many of the same characteristics and behaviors as bare-metal virtualization. The difference comes from how the system installs the hypervisor. In a hosted environment, the system installs the host OS prior to installing a suitable hypervisor -- such as VMware Workstation, KVM or Oracle VirtualBox -- atop that OS.

Once the system installs a hosted hypervisor, the hypervisor operates much like a bare-metal hypervisor. It discovers and virtualizes resources and then provisions those virtualized resources to create VMs. The hosted hypervisor and the host OS manage the connection between physical and virtual resources so that VMs -- and the software that runs within them -- only use those virtualized resources.

However, with hosted virtualization, the system can't virtualize resources for the host OS or any applications installed on it, because those resources are already in use. This means that a hosted hypervisor can only create as many VMs as there are available resources, minus the physical resources the host OS requires.

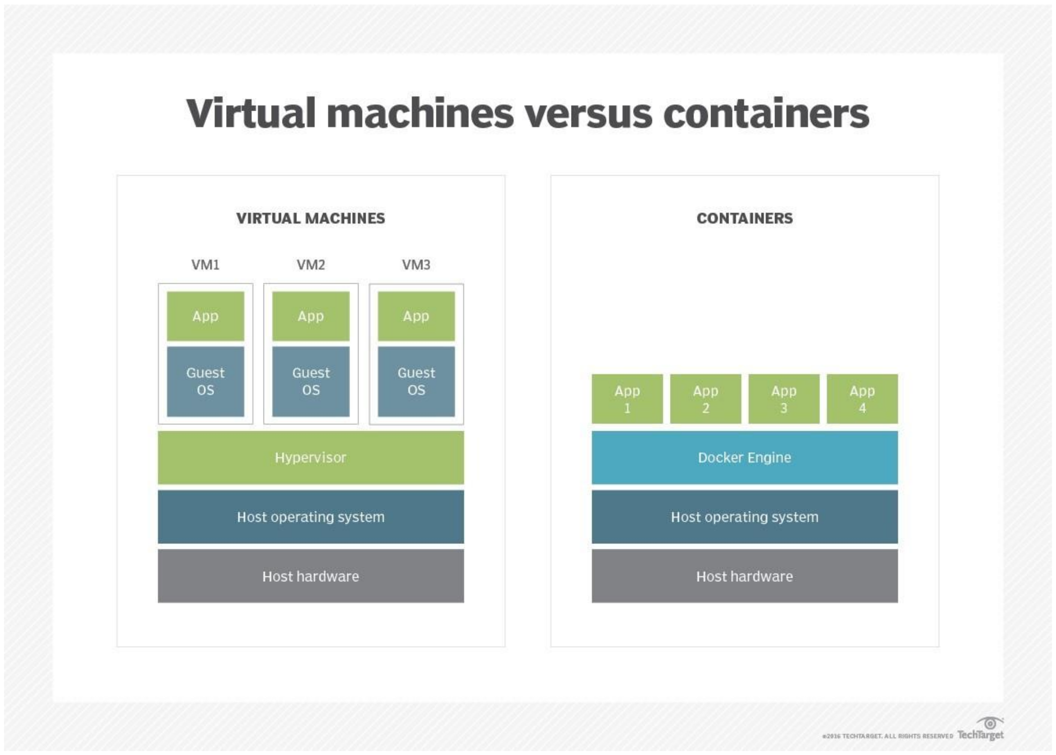
The VMs the hypervisor creates can each receive guest OSes and applications. In addition, every VM created under a hosted hypervisor is isolated from every other VM. Similar to bare-metal virtualization, VMs in a hosted system run in memory and the system can save or load them as disk files to protect, restore or duplicate the VM as desired.

Hosted hypervisors are most commonly used in endpoint systems, such as laptop and desktop PCs, to run two or more desktop environments, each with potentially different OSes. This can benefit business activities such as software development.

In spite of this, organizations use hosted virtualization less often because the presence of a host OS offers no benefits in terms of virtualization or VM performance. The host OS imposes an unnecessary layer of translation between the VMs and the underlying hardware. Inserting a common OS also poses a SPOF for the entire computer, meaning a fault in the host OS affects the hosted hypervisor and all of its VMs.

Although hosted hypervisors have fallen by the wayside for many enterprise tasks, the technology has found new life in container-based virtualization. Containers are a form of virtualization that relies on a container engine, such as Docker, LXC or Apache Mesos, as a hosted hypervisor. The container engine creates and manages virtual instances -- the containers -- that share the services of a common host OS such as Linux.

The crucial difference between hosted VMs and containers is that the system isolates VMs from each other, while containers directly use the same underlying OS kernel. This enables containers to consume fewer system resources compared to VMs. Additionally, containers can start up much faster and exist in far greater numbers than VMs, enabling for greater dynamic scalability for workloads that rely on microservice-type software architectures, as well as important enterprise services such as network load balancers.



Implementation & Output:

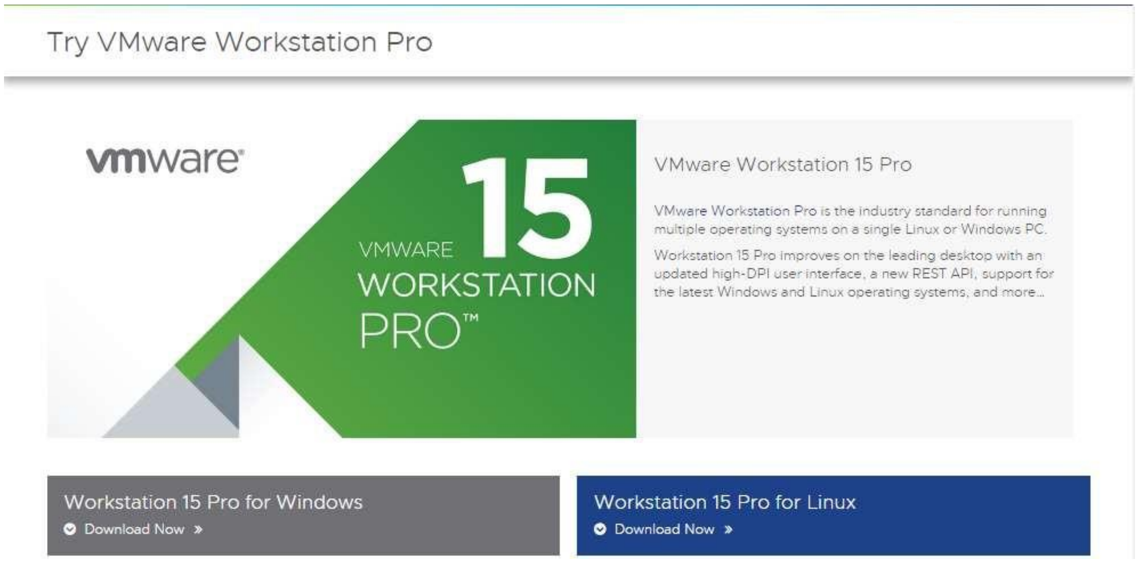
VMWARE:

Installation

Step 1. To download and install the VMware product, visit the official website of VMware.

https://www.vmware.com/in.html

Step 2. Click on Free Product Trials & Demo >> Workstation Pro. You will be redirected to the download page. (Similarly, you can select any product you want to install.)

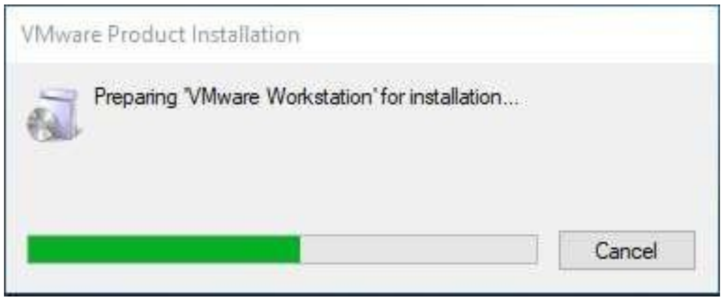

Try VMware Workstation Pro

VMware Workstation 15 Pro

VMware Workstation Pro is the industry standard for running multiple operating systems on a single Linux or Windows PC.

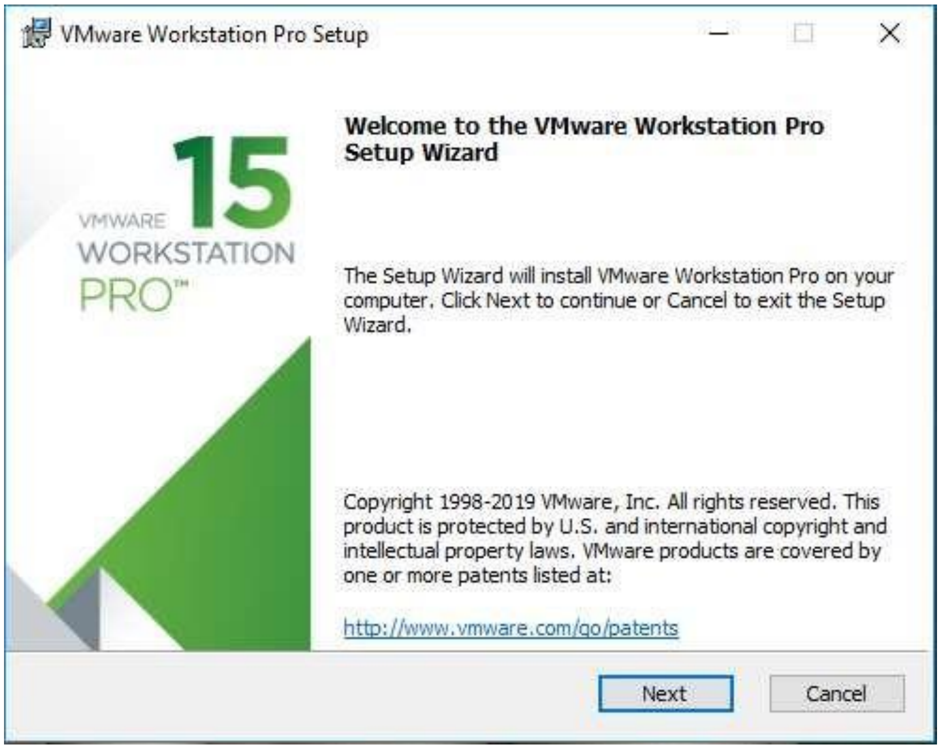
Workstation 15 Pro improves on the leading desktop with an updated high-DPI user interface, a new REST API, support for the latest Windows and Linux operating systems, and more...

Step 3. Once the download is complete, run the .exe to install VMware Workstation. A popup will appear.


VMware Product Installation

Preparing 'VMware Workstation' for installation...

Step 4. Once Initialization gets completed, Click on Next.



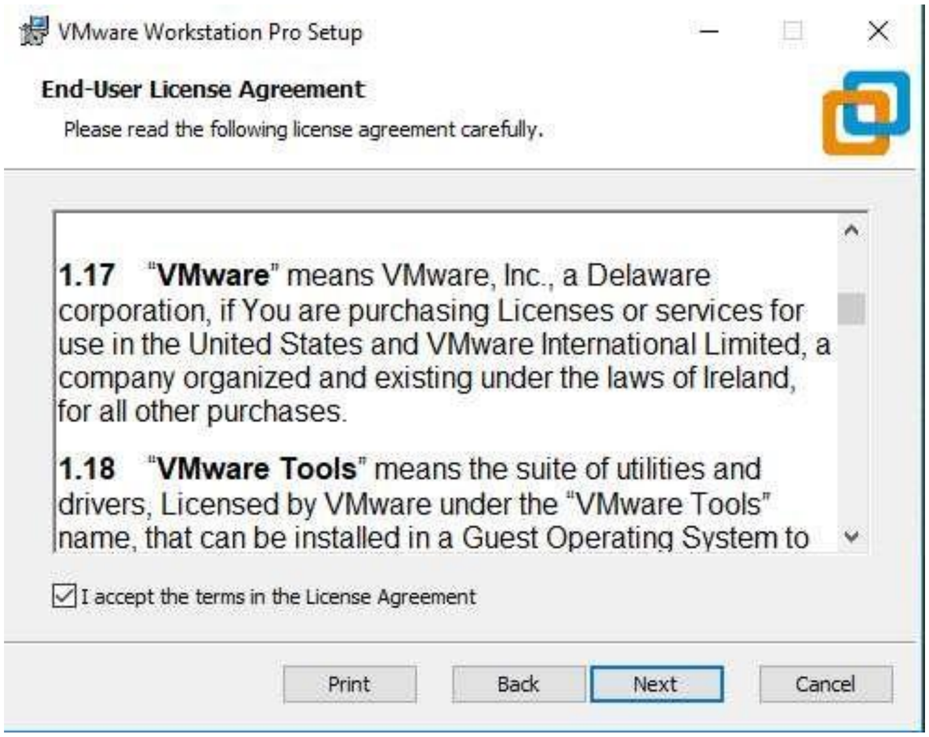
Welcome to the VMware Workstation Pro Setup Wizard

The Setup Wizard will install VMware Workstation Pro on your computer. Click Next to continue or Cancel to exit the Setup Wizard.

Copyright 1998-2019 VMware, Inc. All rights reserved. This product is protected by U.S. and international copyright and intellectual property laws. VMware products are covered by one or more patents listed at:

http://www.vmware.com/go/patents

Step 5. Accept the terms and click Next



End-User License Agreement

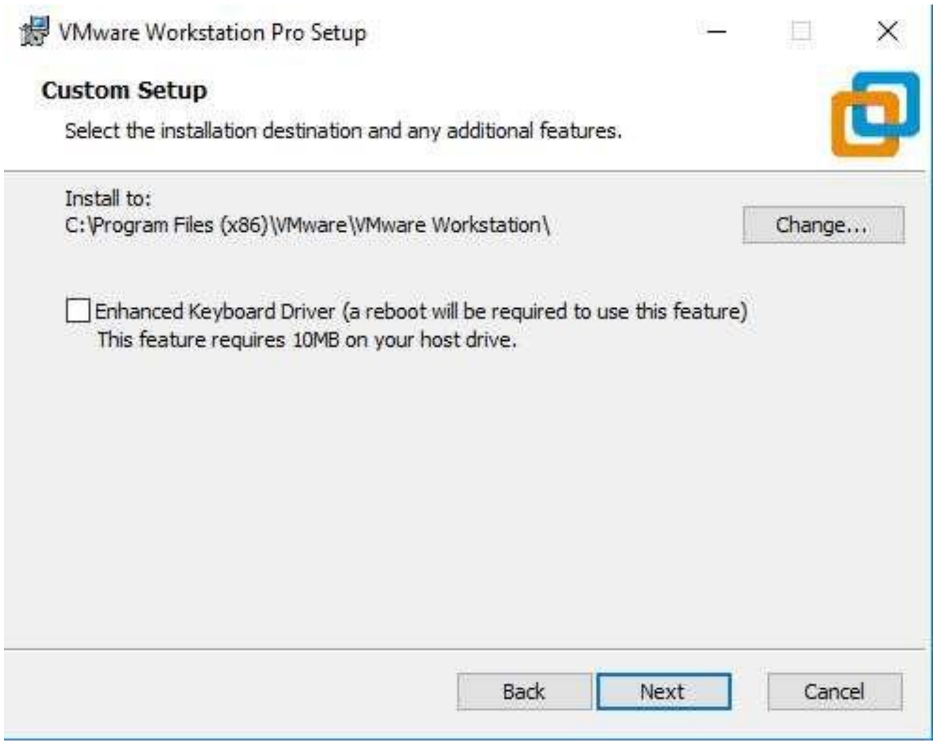
Please read the following license agreement carefully.

1.17 "**VMware**" means VMware, Inc., a Delaware corporation, if You are purchasing Licenses or services for use in the United States and VMware International Limited, a company organized and existing under the laws of Ireland, for all other purchases.

1.18 "**VMware Tools**" means the suite of utilities and drivers, Licensed by VMware under the "VMware Tools" name, that can be installed in a Guest Operating System to

☑ I accept the terms in the License Agreement

Step 6. On the next screen, It will ask for additional features; it is not mandatory to check this box. Click on Next.



VMware Workstation Pro Setup

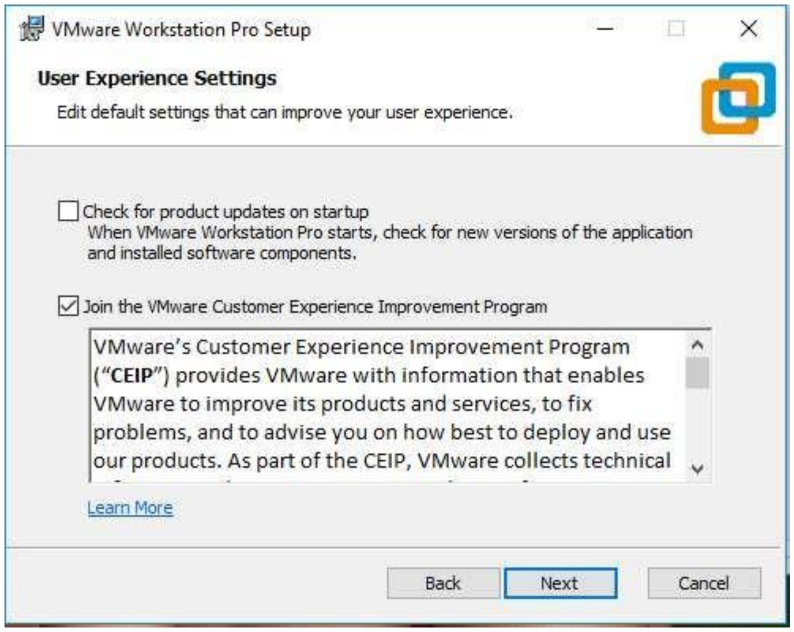
Custom Setup

Select the installation destination and any additional features.

Install to:
C:\Program Files (x86)\VMware\VMware Workstation\

☐ Enhanced Keyboard Driver (a reboot will be required to use this feature)
This feature requires 10MB on your host drive.

Step 7. On the next screen, some checkboxes are populated; check them as per your requirement. Click on Next.



VMware Workstation Pro Setup

User Experience Settings

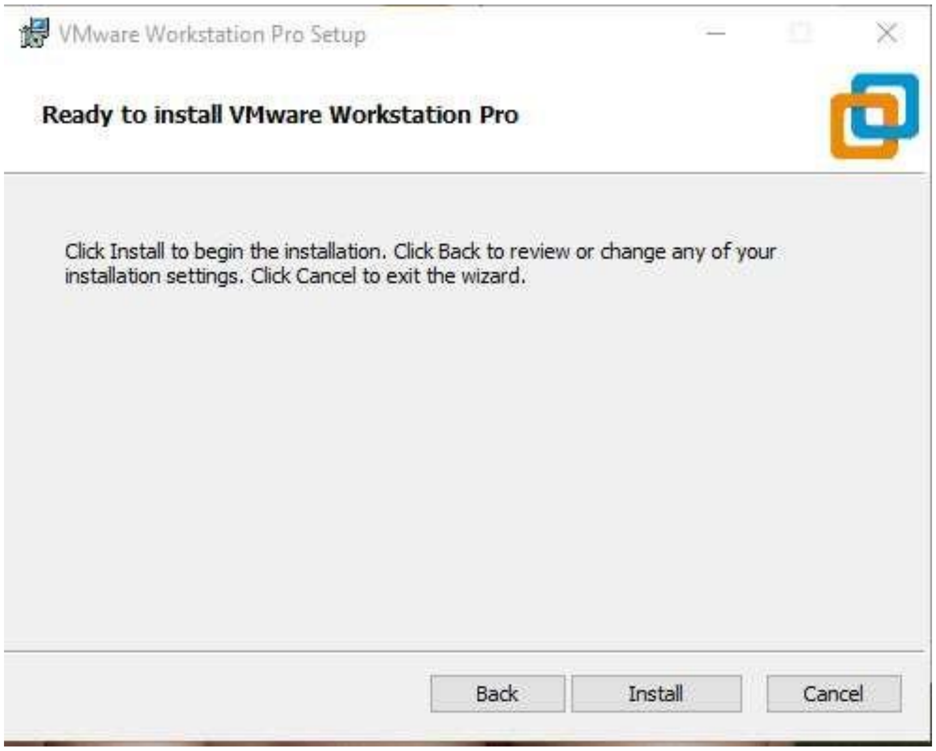
Edit default settings that can improve your user experience.

☐ Check for product updates on startup
When VMware Workstation Pro 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

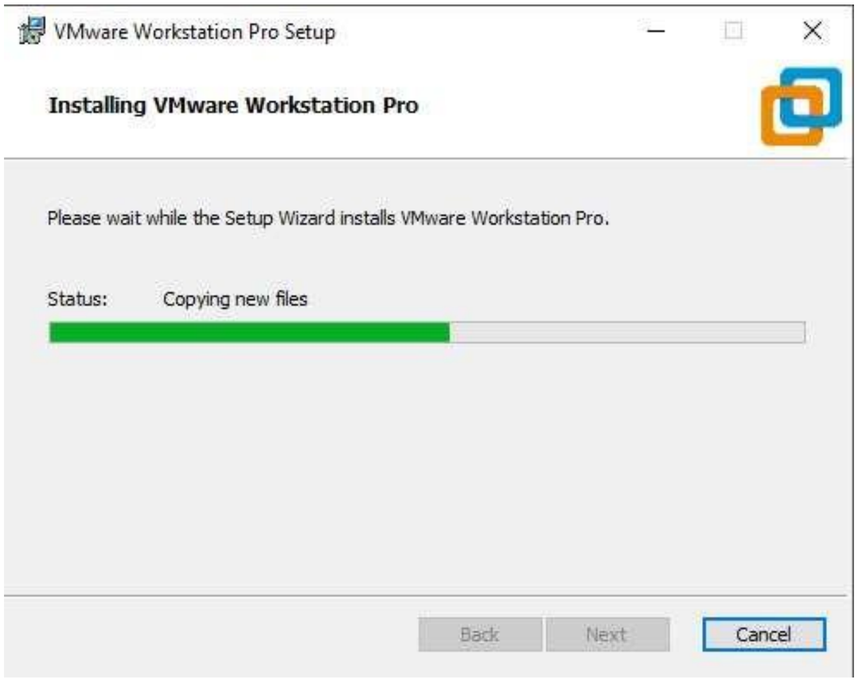
Step 8. At this step, VMware Workstation is ready to install. Click on Install.


VMware Workstation Pro Setup

Ready to install VMware Workstation Pro

Click Install to begin the installation. Click Back to review or change any of your installation settings. Click Cancel to exit the wizard.

Step 9. At this step, you can see installation taking place. The installation will take some time; wait for it to install properly.

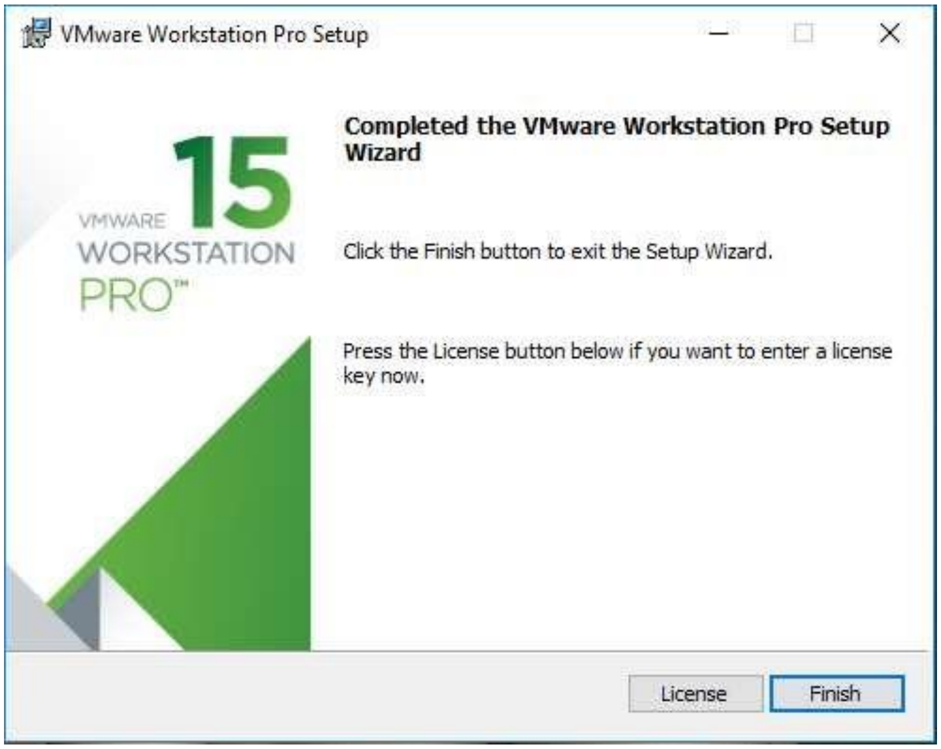

VMware Workstation Pro Setup

Installing VMware Workstation Pro

Please wait while the Setup Wizard installs VMware Workstation Pro.

Status: Copying new files

Step 10. Once the installation gets completed, you will see the following dialogue box. Click on Finish. If you have purchased the product and have a license key, click on License to enter the key.


VMware Workstation Pro Setup

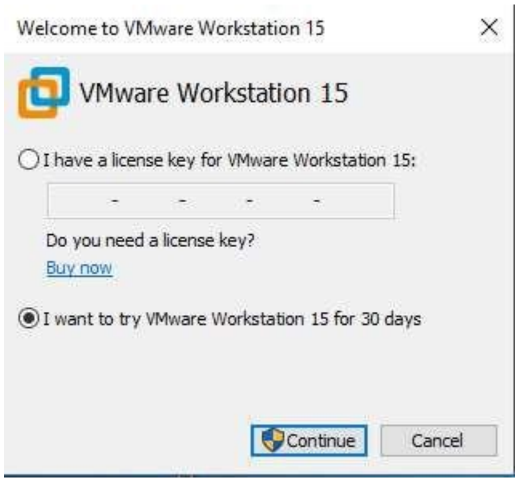
Completed the VMware Workstation Pro Setup Wizard

Click the Finish button to exit the Setup Wizard.

Press the License button below if you want to enter a license key now.

Step 11. Upon Finishing, the window will close, and You can see VMware Workstation installed icon on your Desktop.

Step 12. For the first time opening, if you have not entered the License key in step 7, it will ask for a license key. You can go for the trial version, free for 15 to 30 days. Click on Continue.


Welcome to VMware Workstation 15

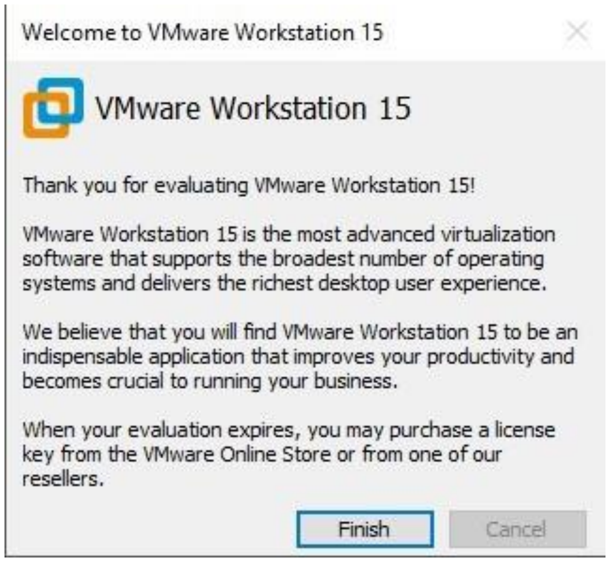
VMware Workstation 15

I have a license key for VMware Workstation 15:

Do you need a license key?
Buy now

[✓ I want to try VMware Workstation 15 for 30 days]

At this stage, you will get the final installation message. Click on Finish.



Welcome to VMware Workstation 15

VMware Workstation 15

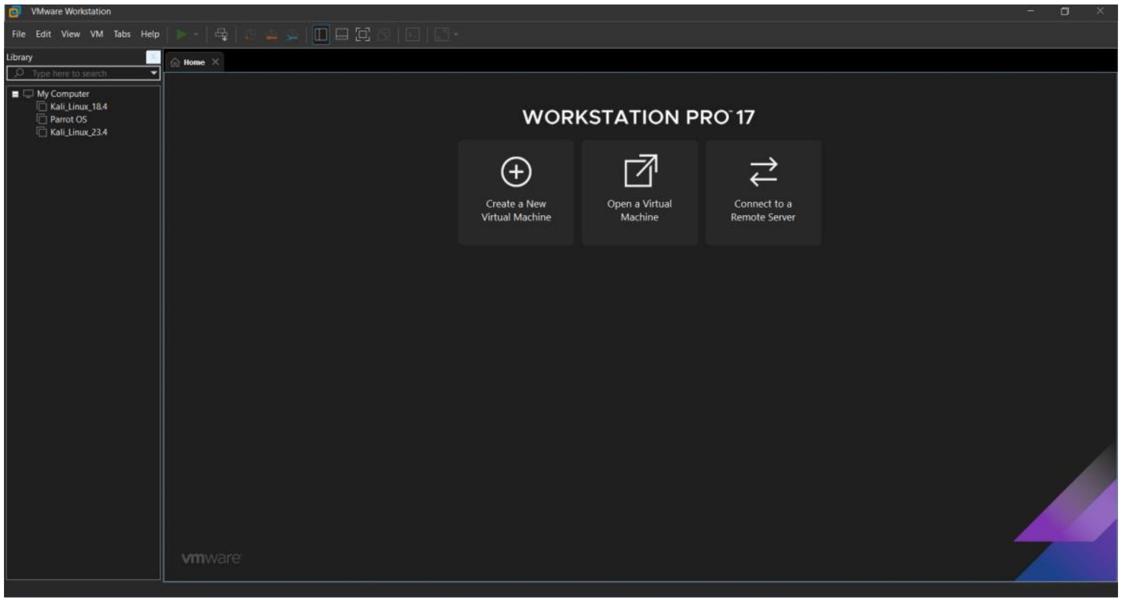
Thank you for evaluating VMware Workstation 15!

VMware Workstation 15 is the most advanced virtualization software that supports the broadest number of operating systems and delivers the richest desktop user experience.

We believe that you will find VMware Workstation 15 to be an indispensable application that improves your productivity and becomes crucial to running your business.

When your evaluation expires, you may purchase a license key from the VMware Online Store or from one of our resellers.

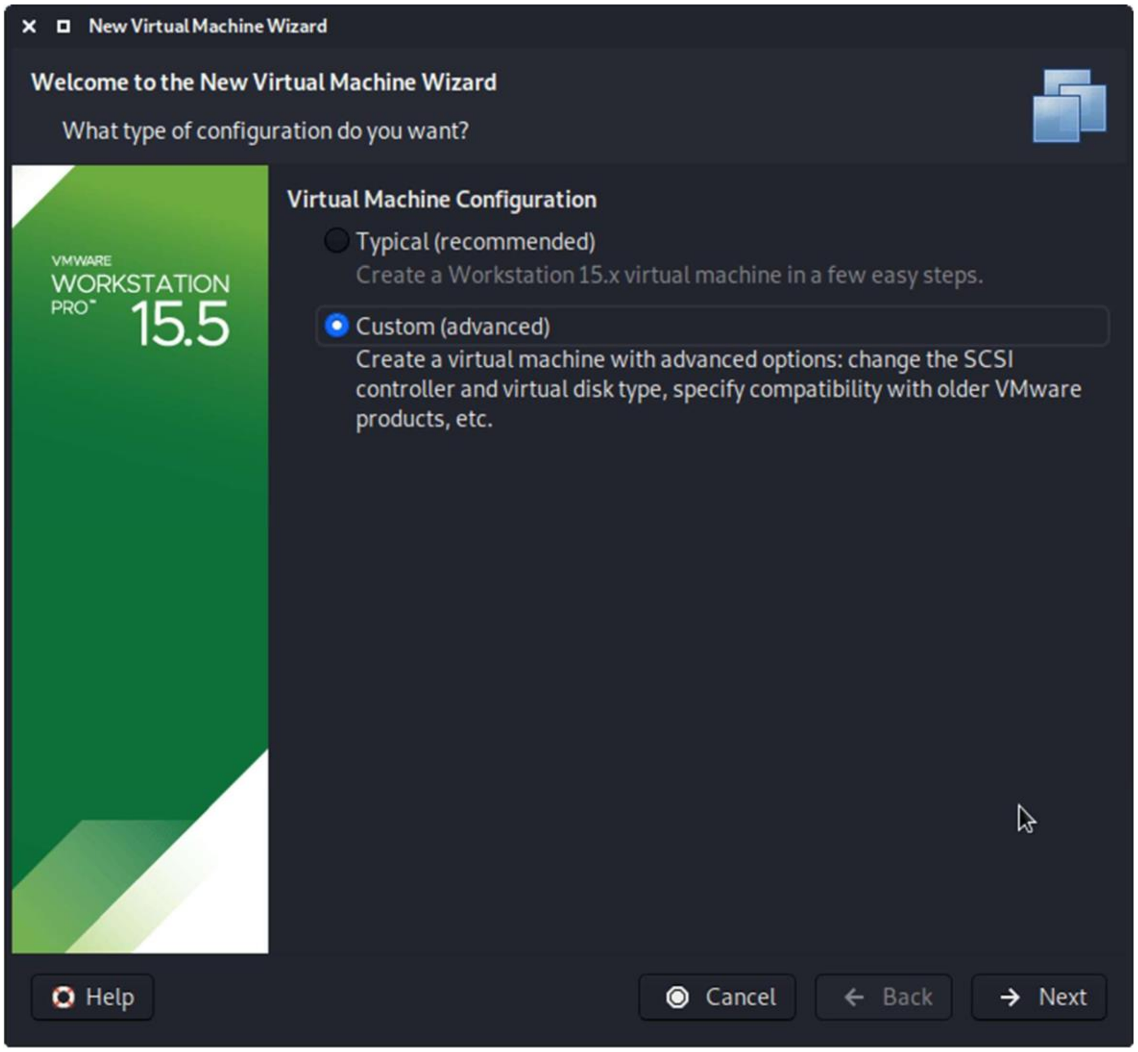
Finally, this will open a window of VMware Workstation Pro.



Installing OS on it:

Upon starting up VMware Workstation, select "**Create a New Virtual Machine**".

When you have the option, select "**Custom (advanced)**" for the Virtual Machine Configuration, as this will allow us to have more control over the creation of the VM.



New Virtual Machine Wizard

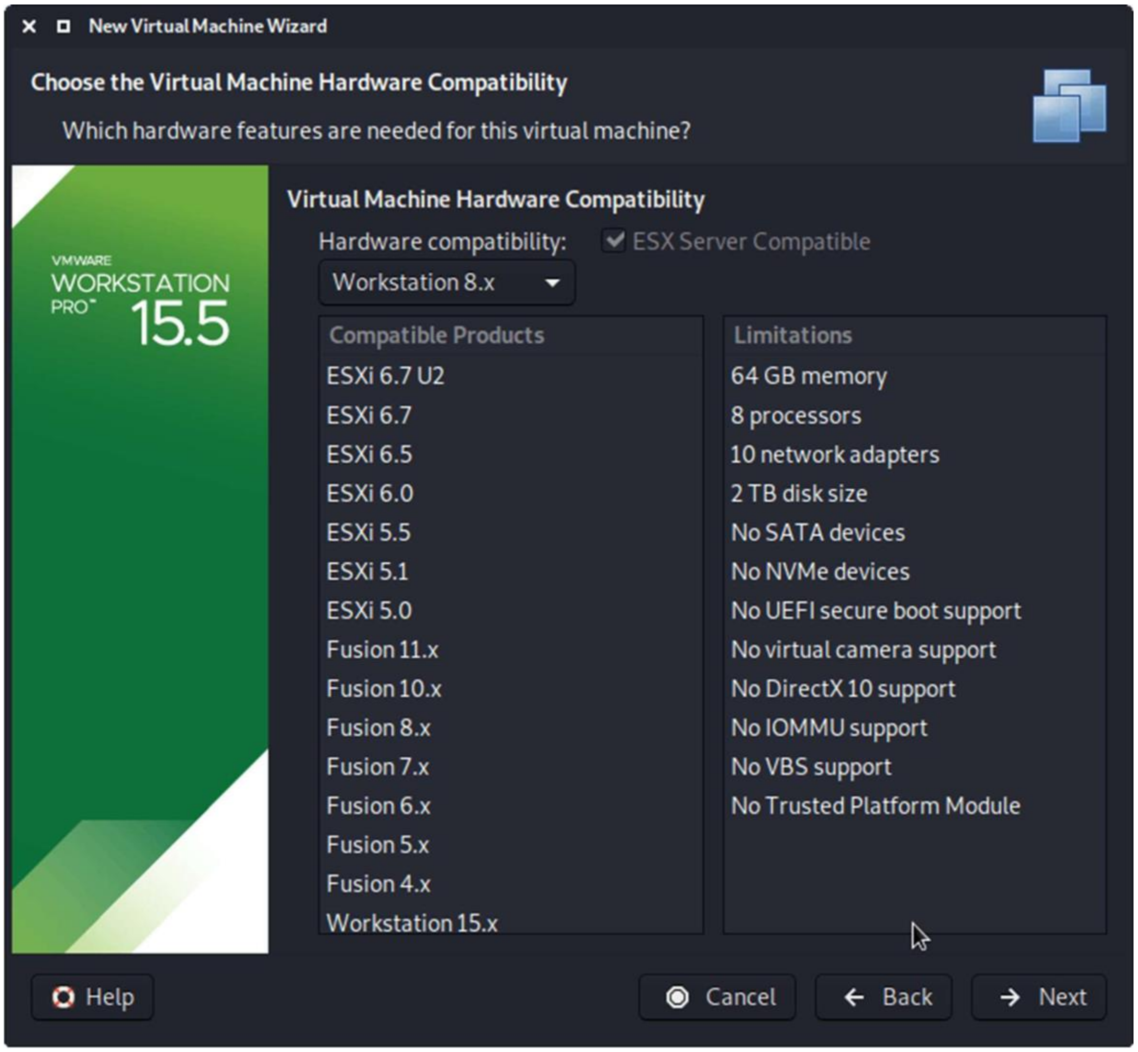
Welcome to the New Virtual Machine Wizard

What type of configuration do you want?

Virtual Machine Configuration

( ) Typical (recommended)
Create a Workstation 15.x virtual machine in a few easy steps.

[✓ Custom (advanced)]
Create a virtual machine with advanced options: change the SCSI controller and virtual disk type, specify compatibility with older VMware products, etc.


New Virtual Machine Wizard

Choose the Virtual Machine Hardware Compatibility

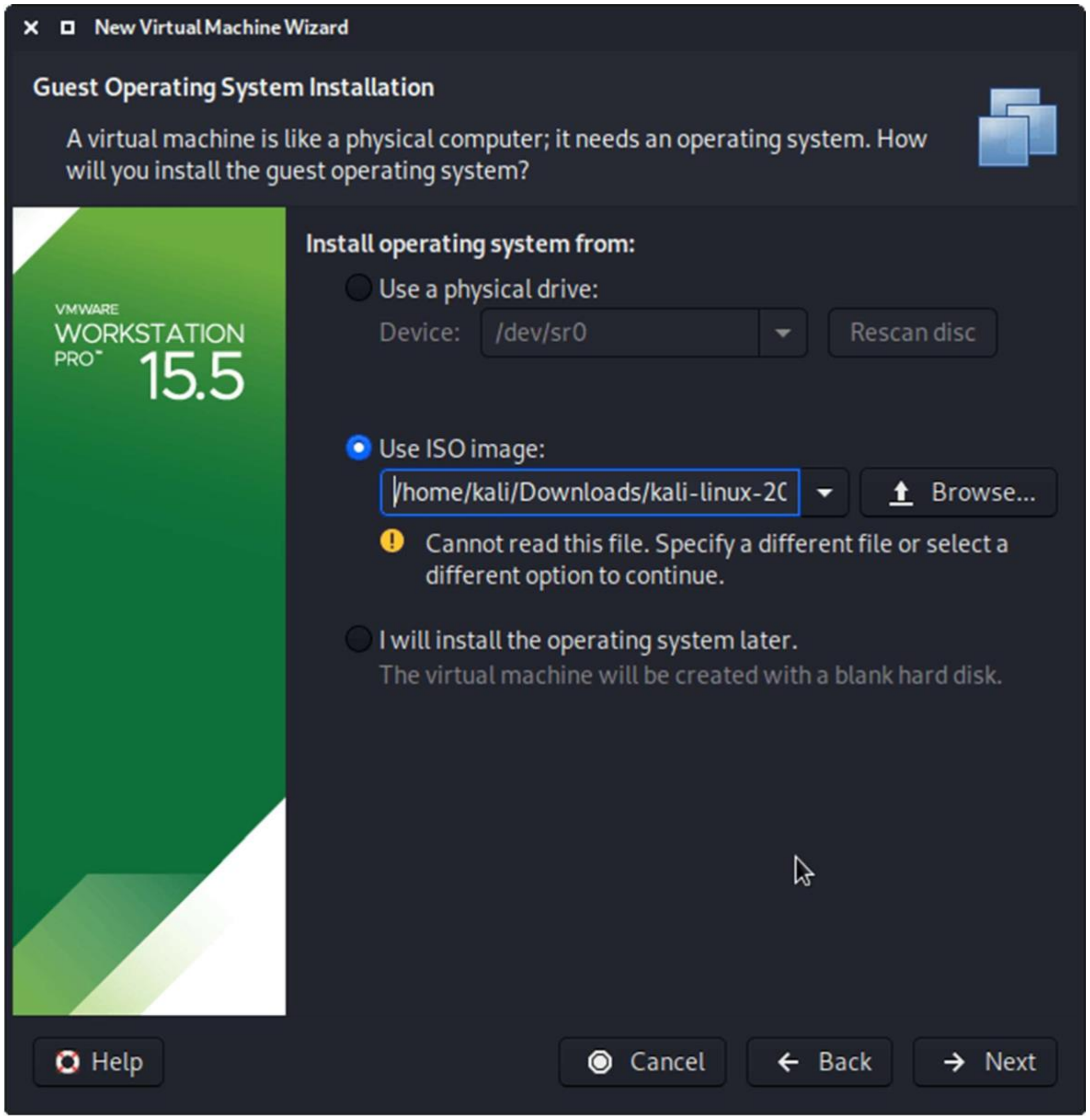
Which hardware features are needed for this virtual machine?

Virtual Machine Hardware Compatibility

Hardware compatibility: Workstation 8.x ✓ ESX Server Compatible

Compatible Products	Limitations
ESXi 6.7 U2	64 GB memory
ESXi 6.7	8 processors
ESXi 6.5	10 network adapters
ESXi 6.0	2 TB disk size
ESXi 5.5	No SATA devices
ESXi 5.1	No NVMe devices
ESXi 5.0	No UEFI secure boot support
Fusion 11.x	No virtual camera support
Fusion 10.x	No DirectX 10 support
Fusion 8.x	No IOMMU support
Fusion 7.x	No VBS support
Fusion 6.x	No Trusted Platform Module
Fusion 5.x	
Fusion 4.x	
Workstation 15.x	

On this screen, we select the Kali Linux image to use to install from. We select "**Browse**", and navigate to the location of the ISO that we downloaded. For more information on what image to download, we have written up a guide.



Guest Operating System Installation

A virtual machine is like a physical computer; it needs an operating system. How will you install the guest operating system?

Install operating system from:

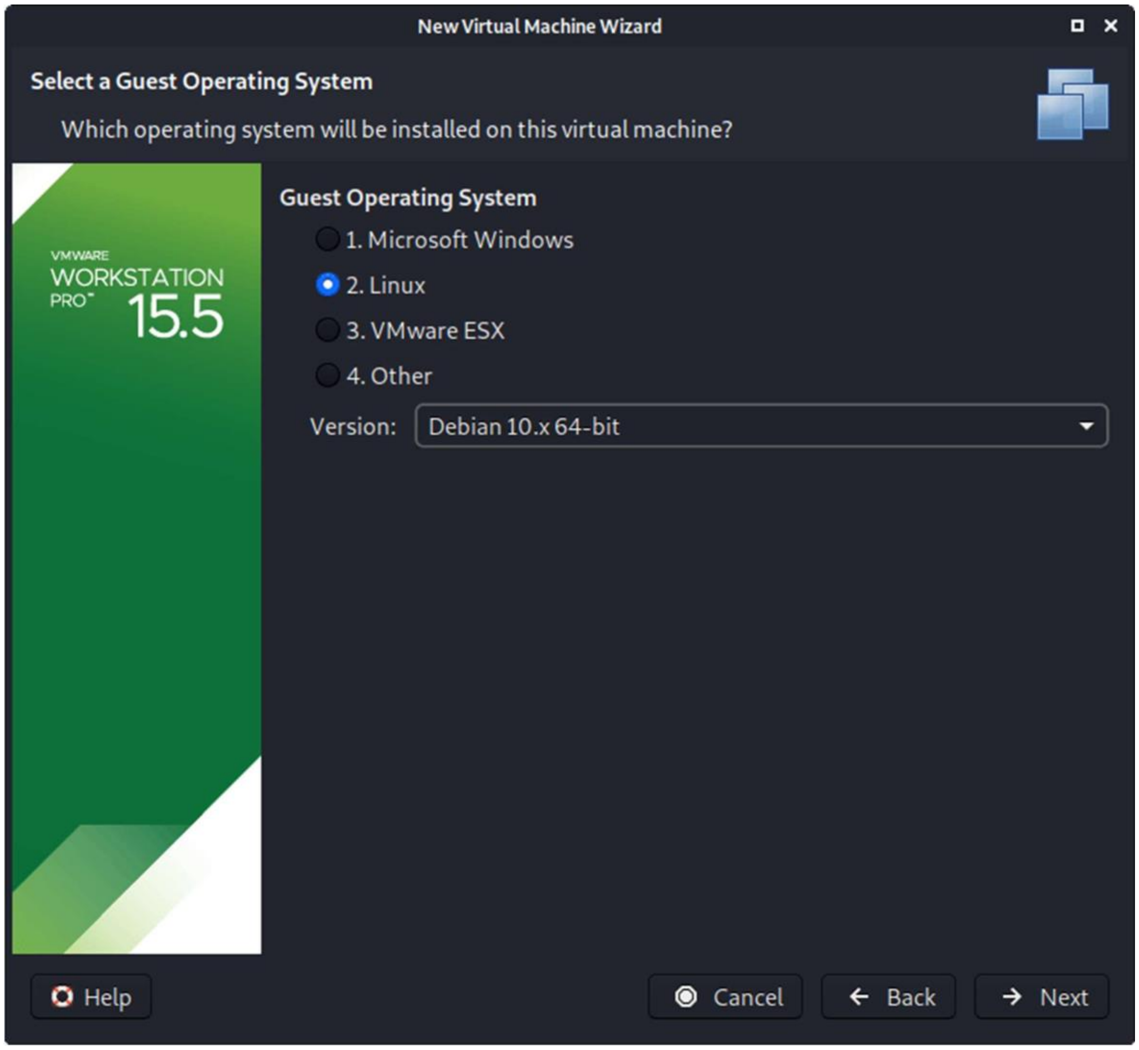
Use a physical drive:
Device: /dev/sr0

[✓ Use ISO image:]
/home/kali/Downloads/kali-linux-2C

Cannot read this file. Specify a different file or select a different option to continue.

I will install the operating system later.
The virtual machine will be created with a blank hard disk.

When you see the "Guest Operating System" screen, select "**Linux**", and then the **latest version of Debian** for the version (as Kali is based on Debian). In this example, its Debian 10. We are going to be use the x64 image to install Kali, so we have selected 64-bit.


New Virtual Machine Wizard

Select a Guest Operating System

Which operating system will be installed on this virtual machine?

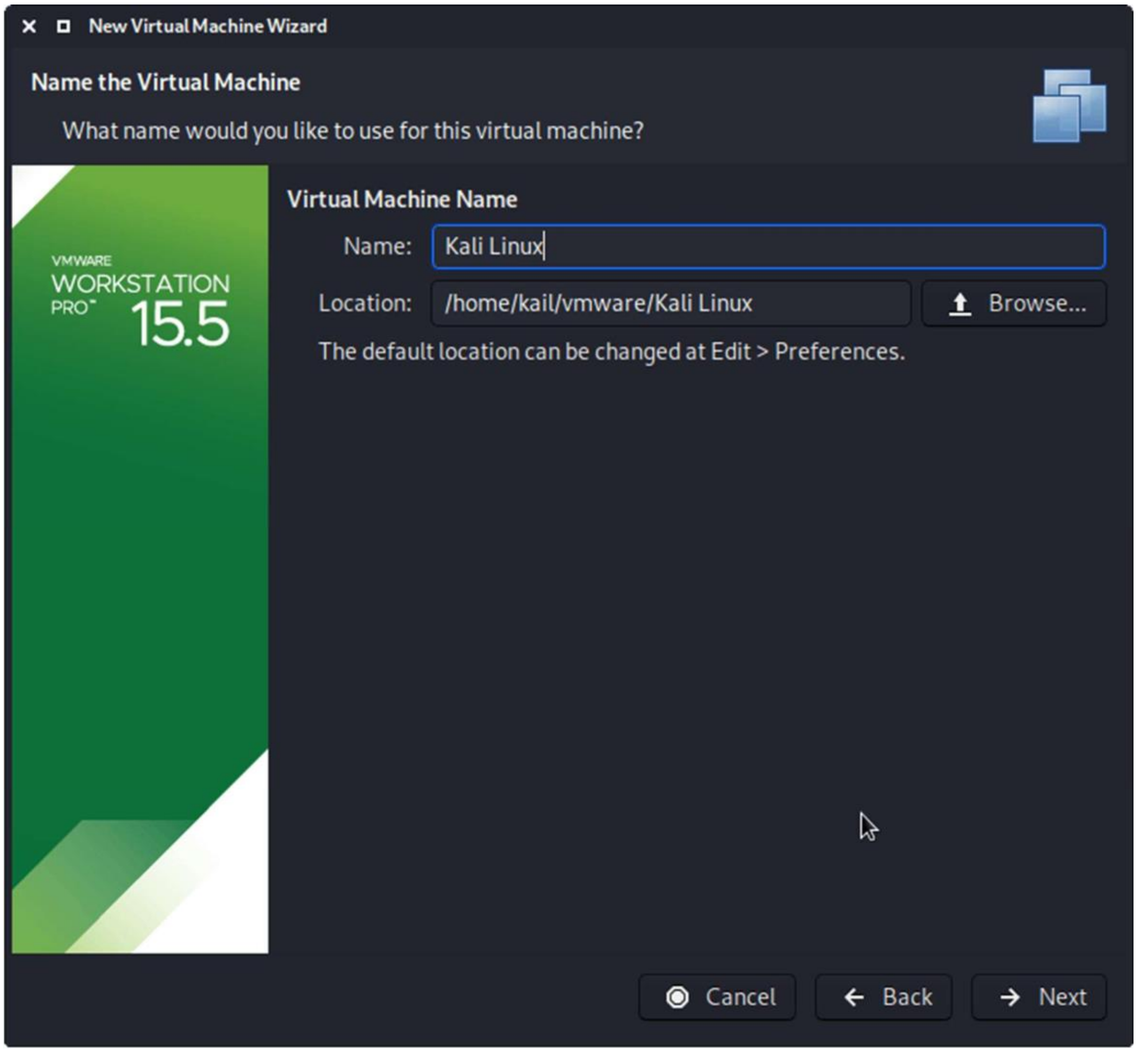
Guest Operating System

1. Microsoft Windows
[✓ 2. Linux]
3. VMware ESX
4. Other

Version: Debian 10.x 64-bit

The next screen is "Virtual Machine Name", which is where you name the VM. This name is also used as the filename (such as the configuration, hard disk and snapshot - which is not changed from this point).

We are keeping it generic in this guide, by using "**Kali Linux**" (as Kali Linux is a rolling distribution, and we update Kali Linux). However for our releases, we use the version number in the name as it is a fixed release *(kali-linux-YYYY.N-vmware-ARCH. Example: kali-linux-2024.1-vmware-amd64)*.


New Virtual Machine Wizard

Name the Virtual Machine

What name would you like to use for this virtual machine?

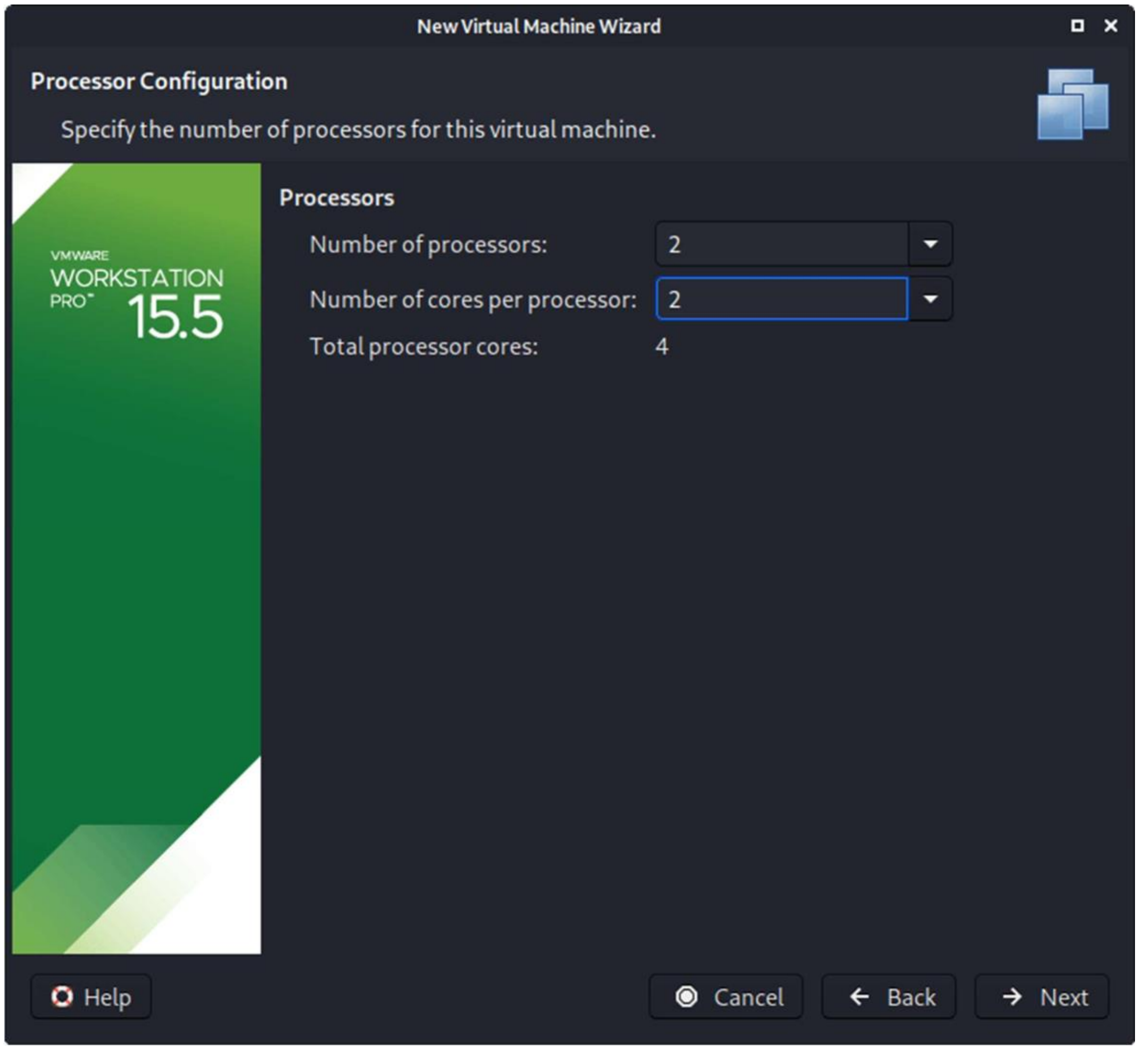
Virtual Machine Name

Name: Kali Linux

Location: /home/kail/vmware/Kali Linux

The default location can be changed at Edit > Preferences.

The next screen is "Processors". Here we can start to define how many resources we give the VM. Kali will be able to perform more tasks simultaneously and quicker if it is allocated more resources. We select "**2 processors**" and "**2 cores per processors**", giving a total of 4 cores. You may wish to use more or less depending on your system requirements.



New Virtual Machine Wizard

Processor Configuration

Specify the number of processors for this virtual machine.

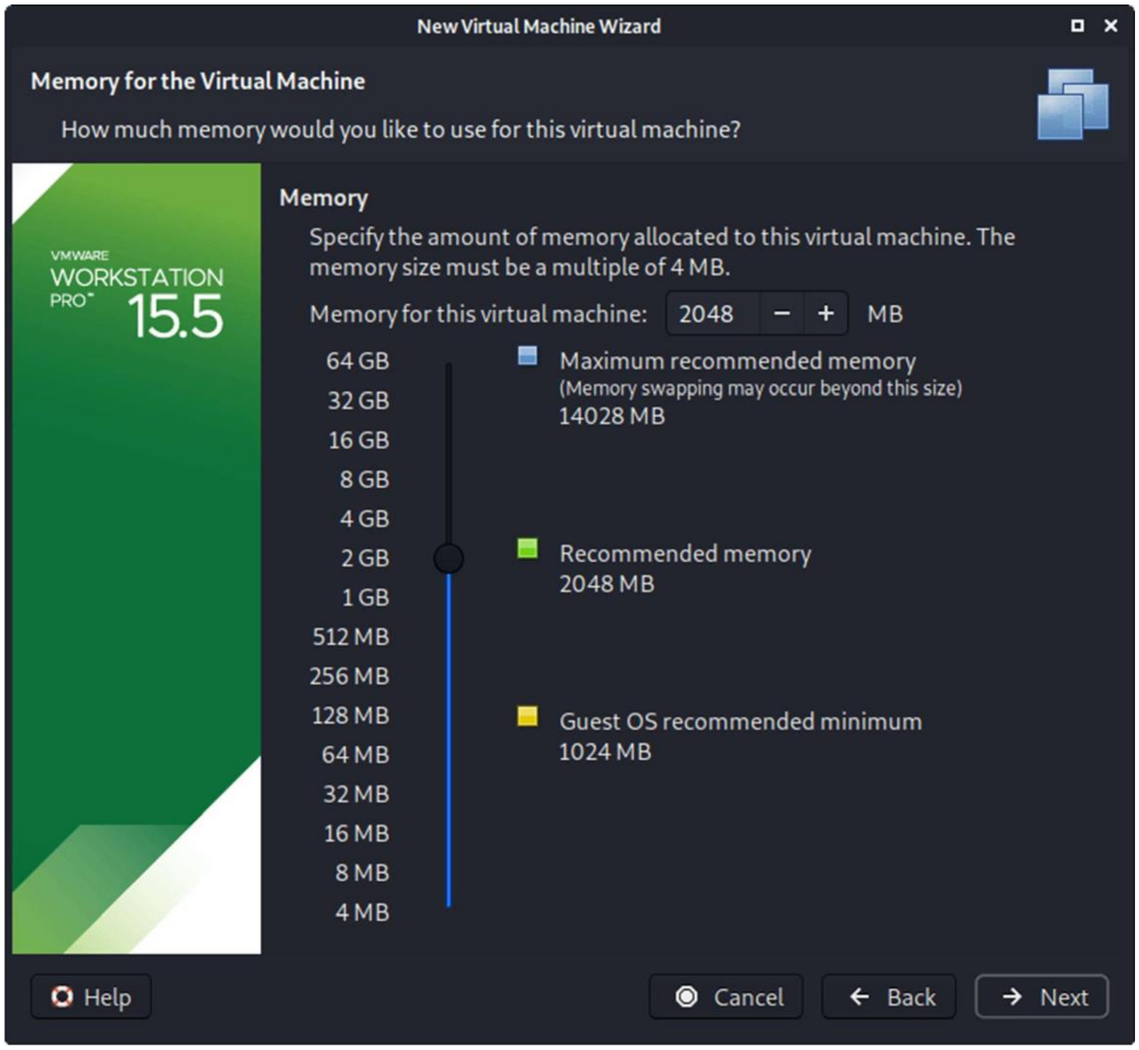
Processors

Number of processors: 2

Number of cores per processor: 2

Total processor cores: 4

"Memory" is the next section, where we can define how much RAM to use. Again, the higher amount of RAM, the more applications can be open and at increased performance. Various tools inside of Kali can be demanding of resources. When we make the general VMs, we select 2GB (**2048 MB**) for RAM, but we often increase this for our personal machines as we have high-performing devices with spare RAM which Kali can utilize.



New Virtual Machine Wizard

Memory for the Virtual Machine

How much memory would you like to use for this virtual machine?

Memory

Specify the amount of memory allocated to this virtual machine. The memory size must be a multiple of 4 MB.

Memory for this virtual machine: 2048 MB

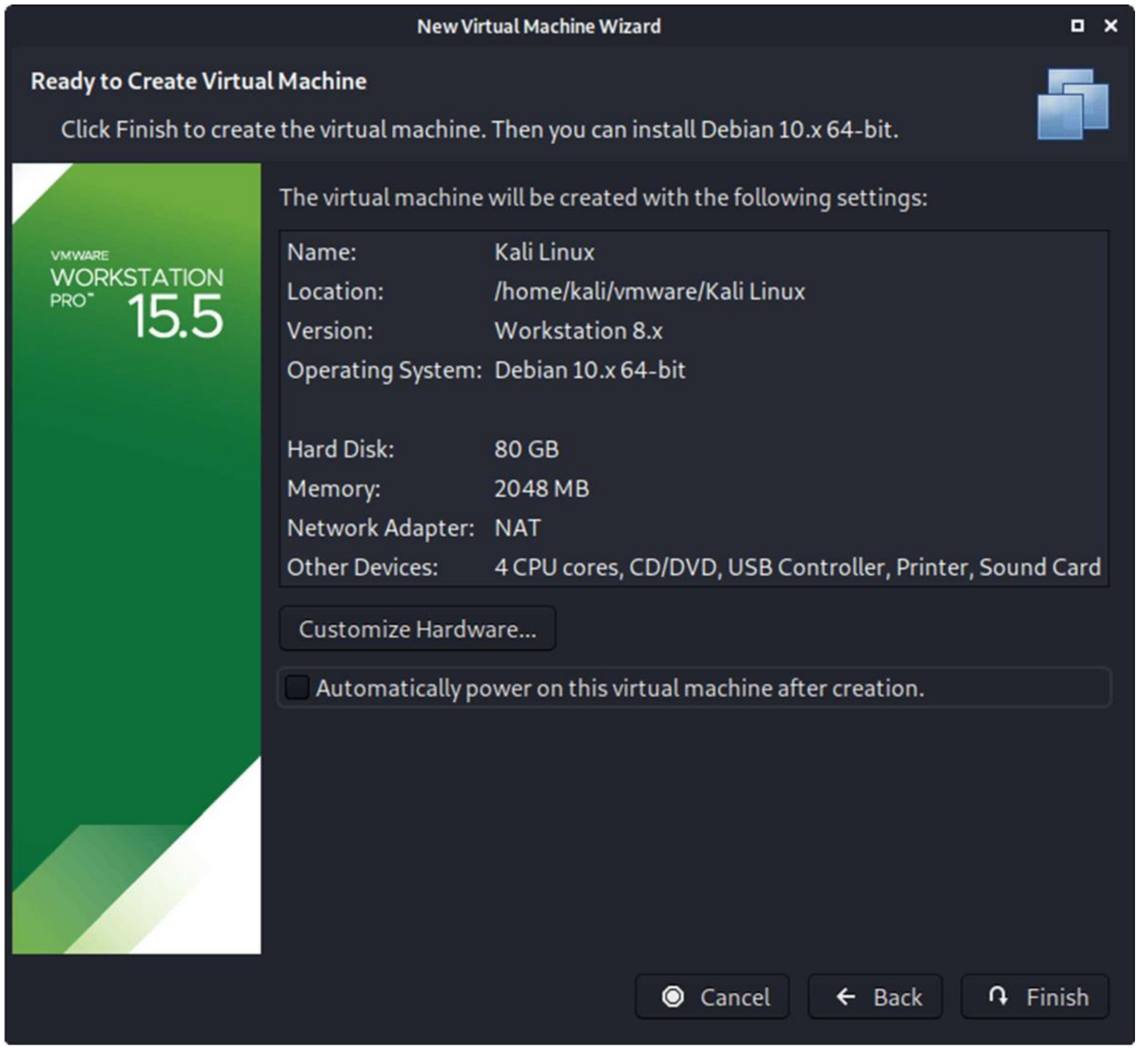
🟦 Maximum recommended memory
(Memory swapping may occur beyond this size)
14028 MB

🟩 Recommended memory
2048 MB

🟨 Guest OS recommended minimum
1024 MB

We are then presented with "Network Connection". We default to using a **NAT** connection. However, this can easy be altered (even when the VM is powered on). This allows for Kali VM to talk to the Internet, as well as the rest of the LAN connection, without it taking up an additional IP address. The downside to this is it will not be able to receive reverse shells (without port forwarding inside of VMware).

The following screen is "Disk", which allows us to "**create a new virtual disk"**



Ready to Create Virtual Machine

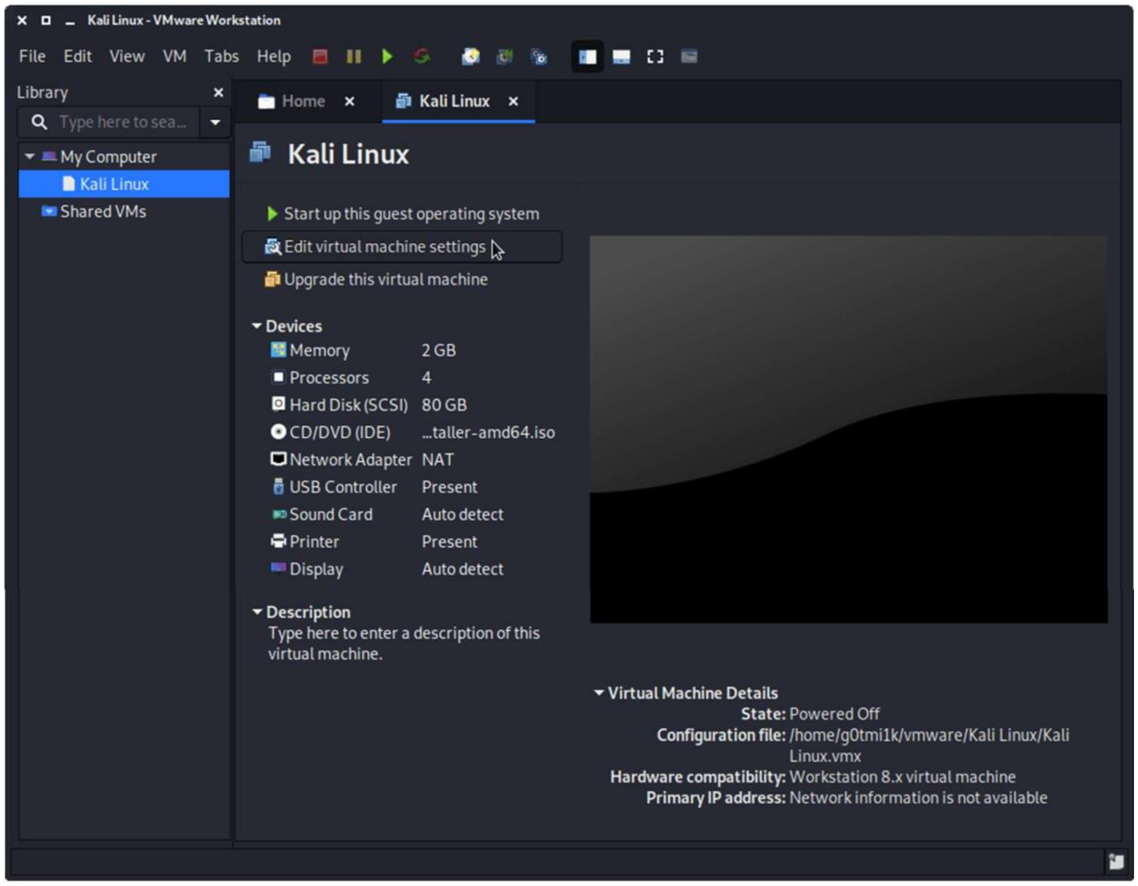
Click Finish to create the virtual machine. Then you can install Debian 10.x 64-bit.

The virtual machine will be created with the following settings:

Name:	Kali Linux
Location:	/home/kali/vmware/Kali Linux
Version:	Workstation 8.x
Operating System:	Debian 10.x 64-bit
Hard Disk:	80 GB
Memory:	2048 MB
Network Adapter:	NAT
Other Devices:	4 CPU cores, CD/DVD, USB Controller, Printer, Sound Card

☐ Automatically power on this virtual machine after creation.

If this is the first time using the wizard, you may have the following prompt explaining how installing "VMware tools" will give you a better experience when using the VM.


Kali Linux - VMware Workstation
File Edit View VM Tabs Help
Library
Type here to sea...
My Computer
Kali Linux
Shared VMs
Home
Kali Linux
Start up this guest operating system
Edit virtual machine settings
Upgrade this virtual machine

Kali Linux

Devices

Memory	2 GB
Processors	4
Hard Disk (SCSI)	80 GB
CD/DVD (IDE)	...taller-amd64.iso
Network Adapter	NAT
USB Controller	Present
Sound Card	Auto detect
Printer	Present
Display	Auto detect

Description
Type here to enter a description of this virtual machine.

Virtual Machine Details
State: Powered Off
Configuration file: /home/g0tmi1k/vmware/Kali Linux/Kali Linux.vmx
Hardware compatibility: Workstation 8.x virtual machine
Primary IP address: Network information is not available

KVM:

Install KVM on Ubuntu 20.04

To enable KVM virtualization on Ubuntu 20.04:

- Install related packages using apt
- Authorize users to run VMs
- Verify that the installation was successful

Step 1: Install KVM Packages

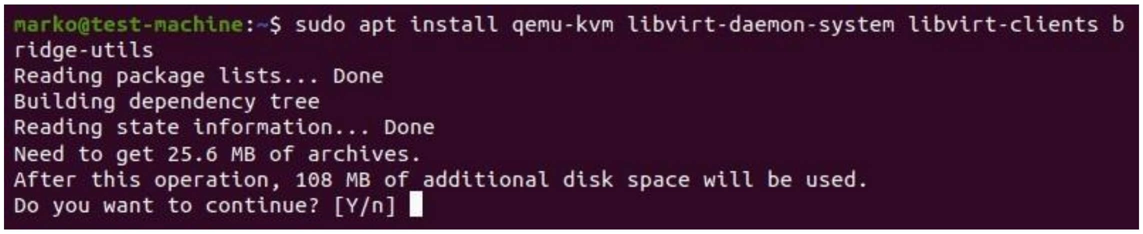
1. First, update the repositories:

sudo apt update

2. Then, install essential KVM packages with the following command:

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

This will start the installation of four KVM packages:



```
marko@test-machine:~$ sudo apt install qemu-kvm libvirt-daemon-system libvirt-clients b
ridge-utils
Reading package lists... Done
Building dependency tree
Reading state information... Done
Need to get 25.6 MB of archives.
After this operation, 108 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

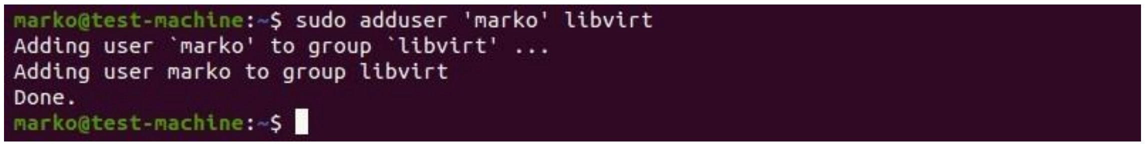
3. When prompted, type **Y**, press **ENTER**, and wait for the installation to finish.

Step 2: Authorize Users

1. Only members of the **libvirt** and **kvm** user groups can run virtual machines. Add a user to the libvirt group by typing:

sudo adduser 'username' libvirt

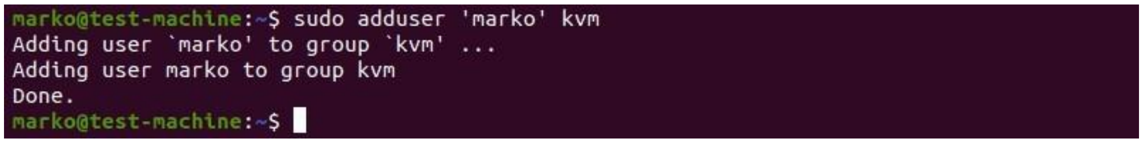
Replace *username* with the actual username.



```
marko@test-machine:~$ sudo adduser 'marko' libvirt
Adding user `marko' to group `libvirt' ...
Adding user marko to group libvirt
Done.
marko@test-machine:~$
```

2. Now do the same for the kvm group:

sudo adduser '[username]' kvm



```
marko@test-machine:~$ sudo adduser 'marko' kvm
Adding user `marko' to group `kvm' ...
Adding user marko to group kvm
Done.
marko@test-machine:~$
```

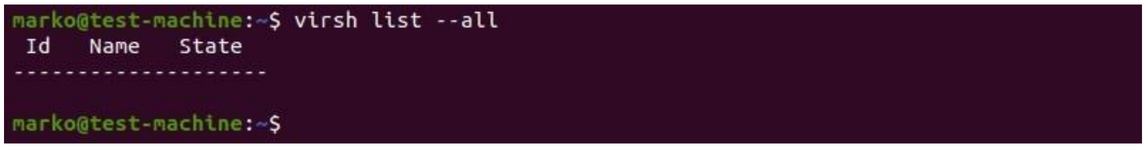
Note: If you need to remove a user from the libvirt or kvm group, just replace **adduser** with **deluser** in the command above.

Step 3: Verify the Installation

1. Confirm the installation was successful by using the **virsh** command:

virsh list --all

You can expect an output as seen below:



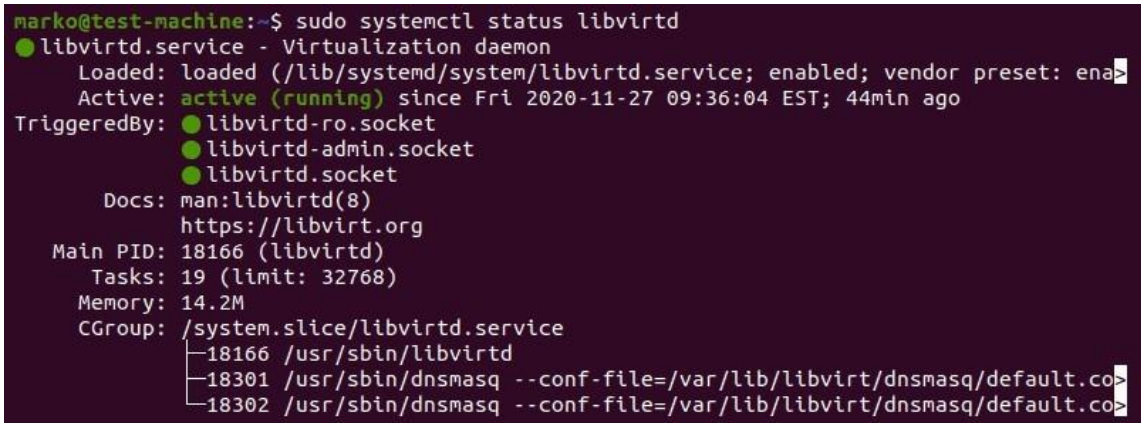
```
marko@test-machine:~$ virsh list --all
 Id   Name   State
--------------------

marko@test-machine:~$
```

2. Or use the **systemctl** command to check the status of libvirtd:

sudo systemctl status libvirtd

If everything is functioning properly, the output returns an **active (running)** status.



```
marko@test-machine:~$ sudo systemctl status libvirtd
● libvirtd.service - Virtualization daemon
     Loaded: loaded (/lib/systemd/system/libvirtd.service; enabled; vendor preset: ena>
     Active: active (running) since Fri 2020-11-27 09:36:04 EST; 44min ago
TriggeredBy: ● libvirtd-ro.socket
             ● libvirtd-admin.socket
             ● libvirtd.socket
       Docs: man:libvirtd(8)
             https://libvirt.org
   Main PID: 18166 (libvirtd)
      Tasks: 19 (limit: 32768)
     Memory: 14.2M
     CGroup: /system.slice/libvirtd.service
             ├─18166 /usr/sbin/libvirtd
             ├─18301 /usr/sbin/dnsmasq --conf-file=/var/lib/libvirt/dnsmasq/default.co>
             └─18302 /usr/sbin/dnsmasq --conf-file=/var/lib/libvirt/dnsmasq/default.co>
```

3. Press **Q** to quit the status screen.

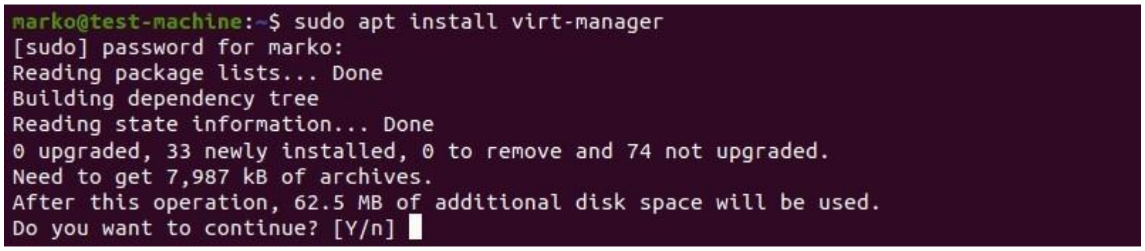
4. If the virtualization daemon is not active, activate it with the following command:

sudo systemctl enable --now libvirtd

Creating a Virtual Machine on Ubuntu 20.04

1. Before you choose one of the two methods listed below, install virt-manager, a tool for creating and managing VMs:

sudo apt install virt-manager



```
marko@test-machine:~$ sudo apt install virt-manager
[sudo] password for marko:
Reading package lists... Done
Building dependency tree
Reading state information... Done
0 upgraded, 33 newly installed, 0 to remove and 74 not upgraded.
Need to get 7,987 kB of archives.
After this operation, 62.5 MB of additional disk space will be used.
Do you want to continue? [Y/n] █
```

2. Type **Y** and press **ENTER**. Wait for the installation to finish.

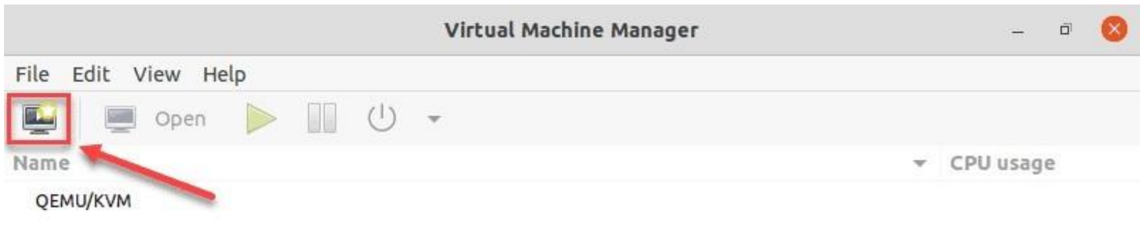
Make sure you download an ISO containing the OS you wish to install on a VM and proceed to pick an installation method.

Method 1: Virt Manager GUI

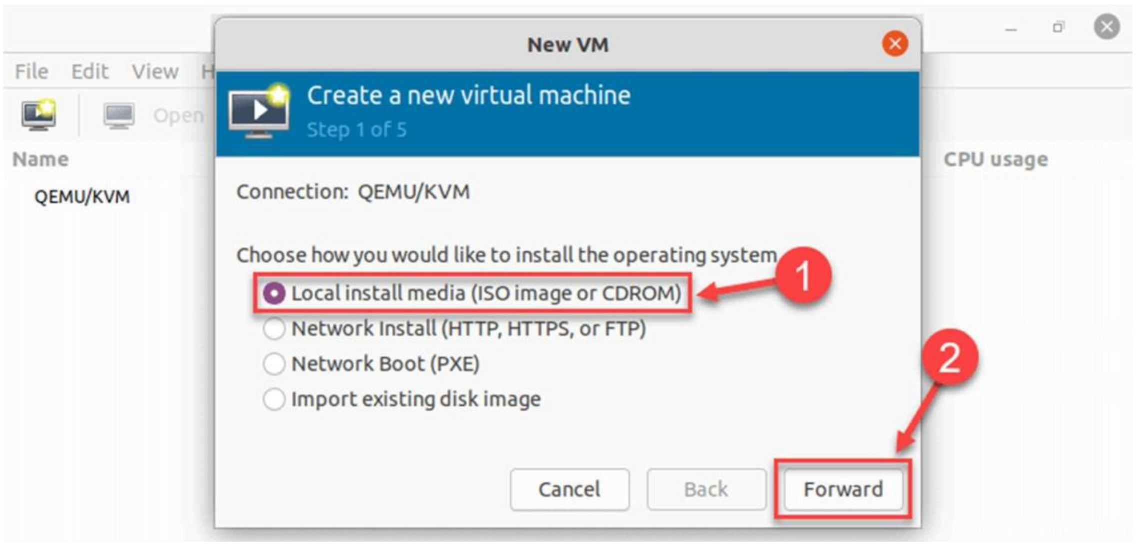
1. Start virt-manager with:

```
sudo virt-manager
```

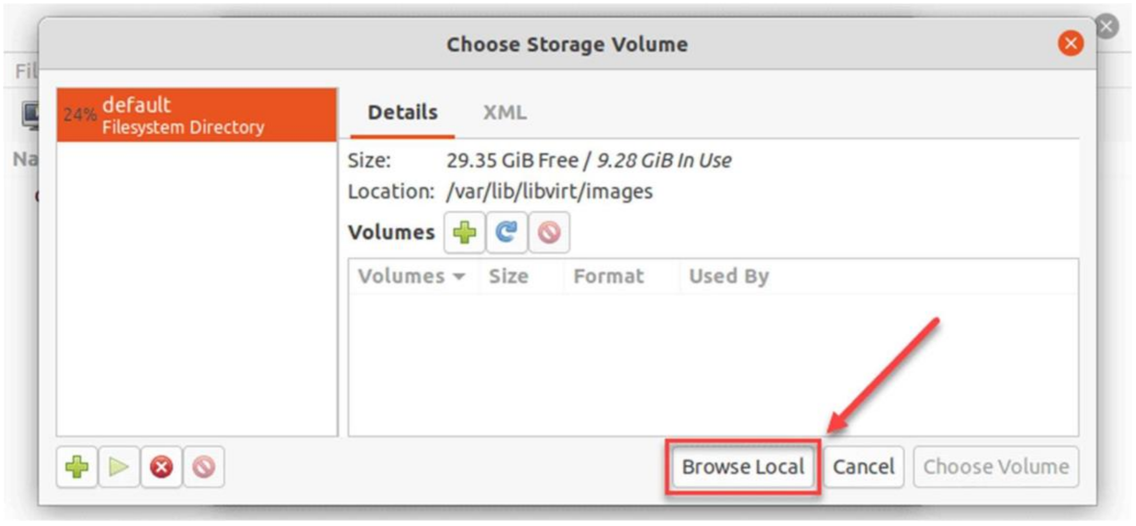
2. In the first window, click the computer icon in the upper-left corner.



3. In the dialogue box that opens, select the option to install the VM using an ISO image. Then click **Forward**.

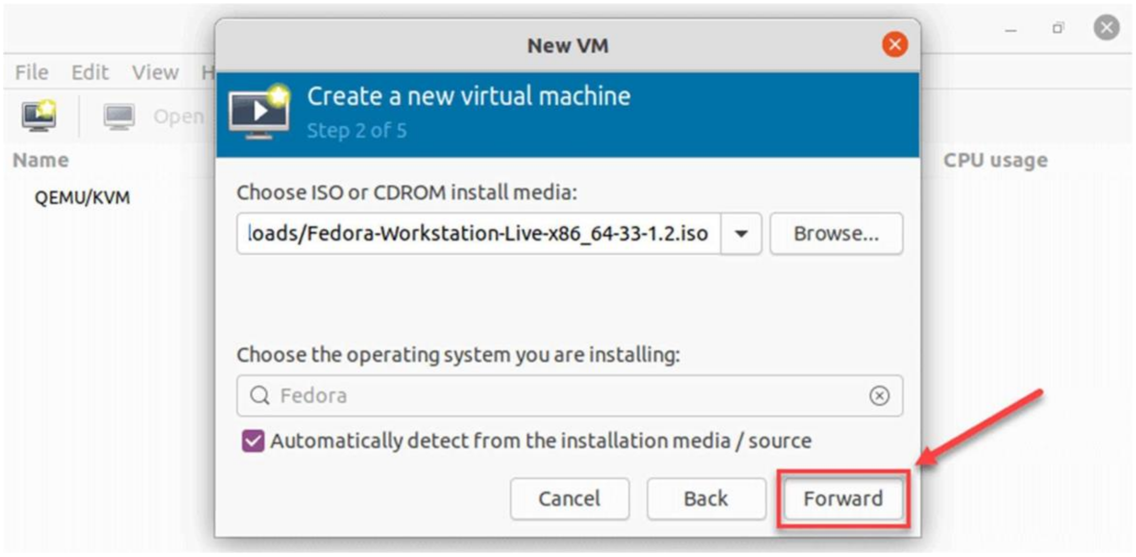

New VM
Create a new virtual machine
Step 1 of 5
Connection: QEMU/KVM
Choose how you would like to install the operating system
Local install media (ISO image or CDROM)
Network Install (HTTP, HTTPS, or FTP)
Network Boot (PXE)
Import existing disk image
1
2
Cancel
Back
Forward
CPU usage
QEMU/KVM

4. In the next dialogue, click **Browse Local** and navigate to the path where you stored the ISO you wish to install.



Size: 29.35 GiB Free / *9.28 GiB In Use*
Location: /var/lib/libvirt/images

5. The ISO you chose in the previous window populates the field in Step 2. Proceed to Step 3 by clicking **Forward**.

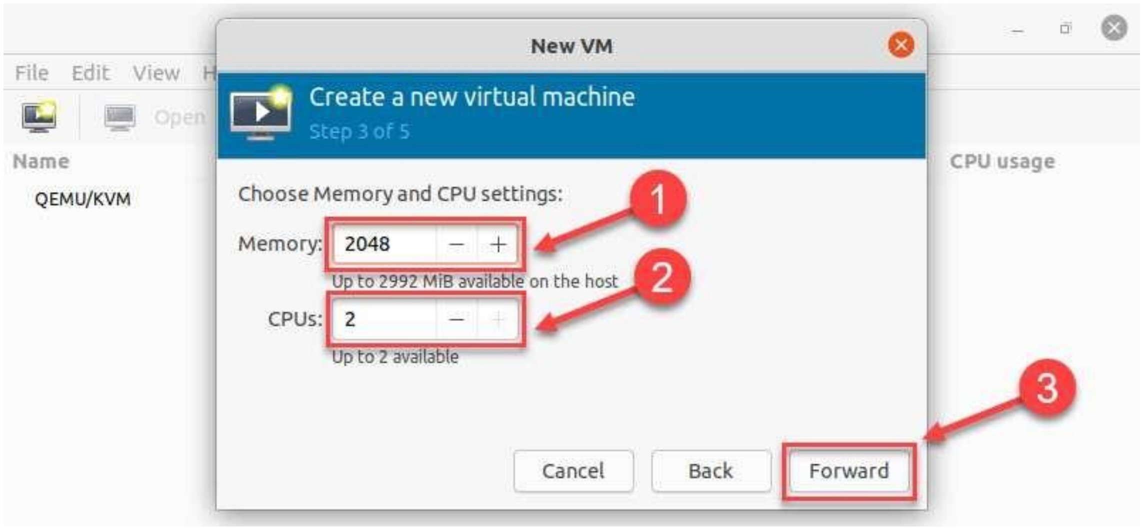


Choose ISO or CDROM install media: loads/Fedora-Workstation-Live-x86_64-33-1.2.iso

Choose the operating system you are installing: Fedora

☑ Automatically detect from the installation media / source

6. Enter the amount of RAM and the number of CPUs you wish to allocate to the VM and proceed to the next step.



New VM

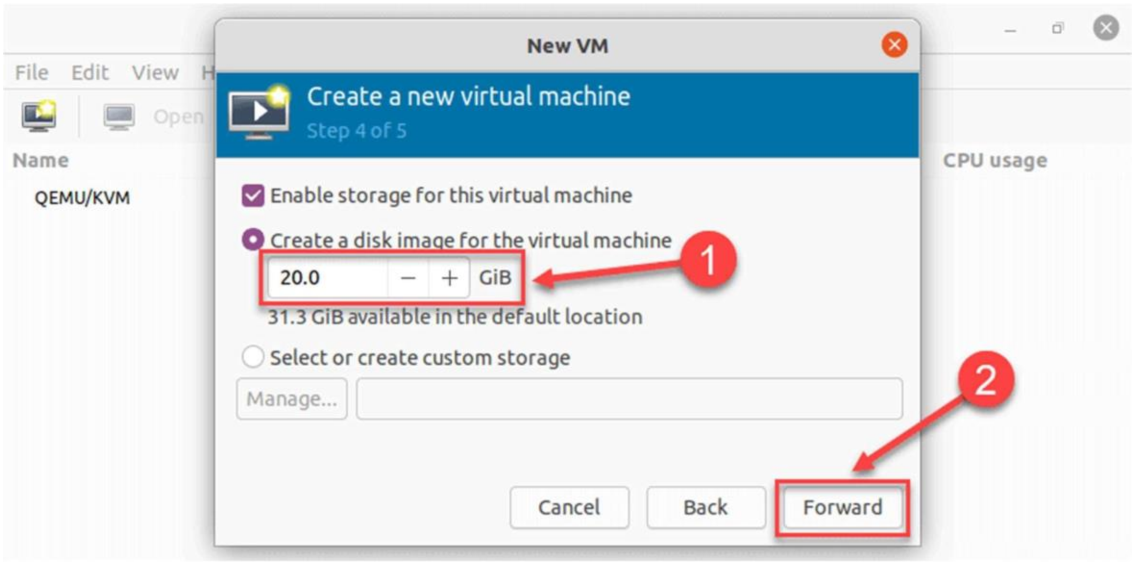
Create a new virtual machine
Step 3 of 5

Choose Memory and CPU settings:

Memory: 2048
Up to 2992 MiB available on the host

CPUs: 2
Up to 2 available

7. Allocate hard disk space to the VM. Click **Forward** to go to the last step.



New VM

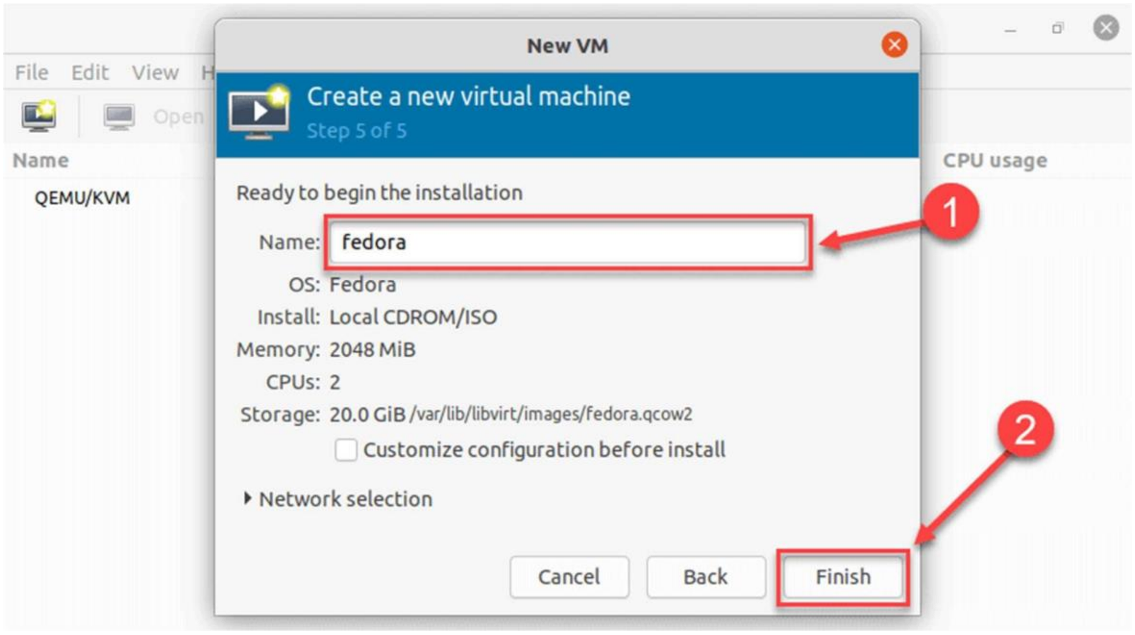
Create a new virtual machine
Step 4 of 5

[✓] Enable storage for this virtual machine

[●] Create a disk image for the virtual machine
20.0 GiB
31.3 GiB available in the default location

[ ] Select or create custom storage

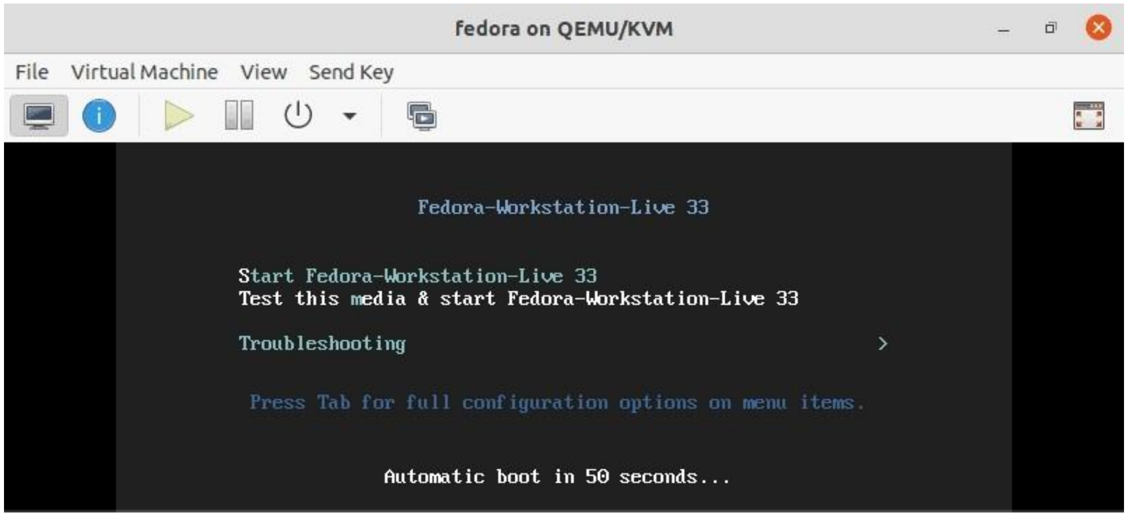
8. Specify the name for your VM and click **Finish** to complete the setup.



Ready to begin the installation

Name: fedora
OS: Fedora
Install: Local CDROM/ISO
Memory: 2048 MiB
CPUs: 2
Storage: 20.0 GiB /var/lib/libvirt/images/fedora.qcow2
[ ] Customize configuration before install

9. The VM starts automatically, prompting you to start installing the OS that's on the ISO file.



```
Fedora-Workstation-Live 33

Start Fedora-Workstation-Live 33
Test this media & start Fedora-Workstation-Live 33

Troubleshooting                                   >

Press Tab for full configuration options on menu items.

Automatic boot in 50 seconds...
```

Conclusion:

In conclusion, the experiment to study and implement Security as a Service (SECaaS) on In summary, the experiment effectively explored virtualization concepts, types, structures, and mechanisms. Through demonstrations using VirtualBox and KVM, it became evident that virtualization offers efficient resource utilization and scalability. Comparing the two hypervisors, VirtualBox excels in user-friendliness, while KVM outperforms in performance and scalability, particularly for enterprise use. This hands-on experience equips users with insights to optimize infrastructure management and enhance system efficiency in various computing environments.

****