

Install KVM on Ubuntu Linux

Kernel-based Virtual Machine (KVM) is a Type-1 hypervisor. It operates directly on the host hardware without the need for a host operating system. KVM is integrated into the Linux kernel, and it takes advantage of hardware virtualization extensions (such as Intel VT-x and AMD-V) to provide efficient virtualization.

As a Type-1 hypervisor, KVM is well-suited for server virtualization scenarios where performance, resource efficiency, and the ability to run multiple virtual machines directly on the hardware are crucial. It is commonly used in data centers and cloud environments to provide virtualization services on Linux-based systems.

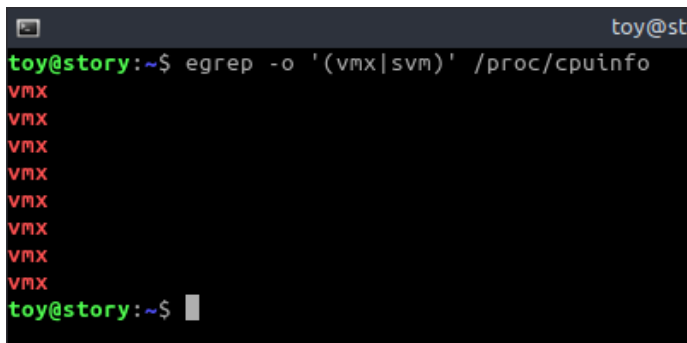
A Type-2 hypervisor is a virtualization solution that runs on top of a host operating system (OS). Type-2 hypervisors operate as applications within a host operating system. They are typically used for desktop or development environments where the emphasis is on ease of use and flexibility rather than maximum performance and resource utilization. Example: Virtualbox.

Installation Steps

1. Ensure that your CPU supports hardware virtualization. ("vmx" for Intel & "svm" for AMD):

```
egrep -o '(vmx|svm)' /proc/cpuinfo
```

vmx indicates support for virtualization.

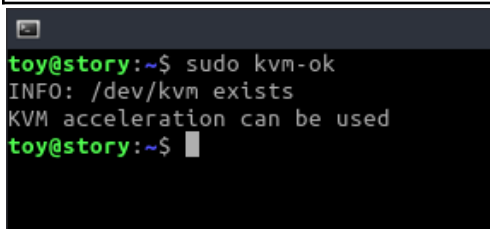


```
toy@story:~$ egrep -o '(vmx|svm)' /proc/cpuinfo
vmx
vmx
vmx
vmx
vmx
vmx
vmx
vmx
vmx
vmx
toy@story:~$
```

2. Ensure system supports KVM Virtualization

```
$ sudo kvm-ok
```

If kvm-ok has not been installed, install it using `sudo apt install cpu-checker`.



```
toy@story:~$ sudo kvm-ok
INFO: /dev/kvm exists
KVM acceleration can be used
toy@story:~$
```

KVM support for current OS.

3. Install KVM

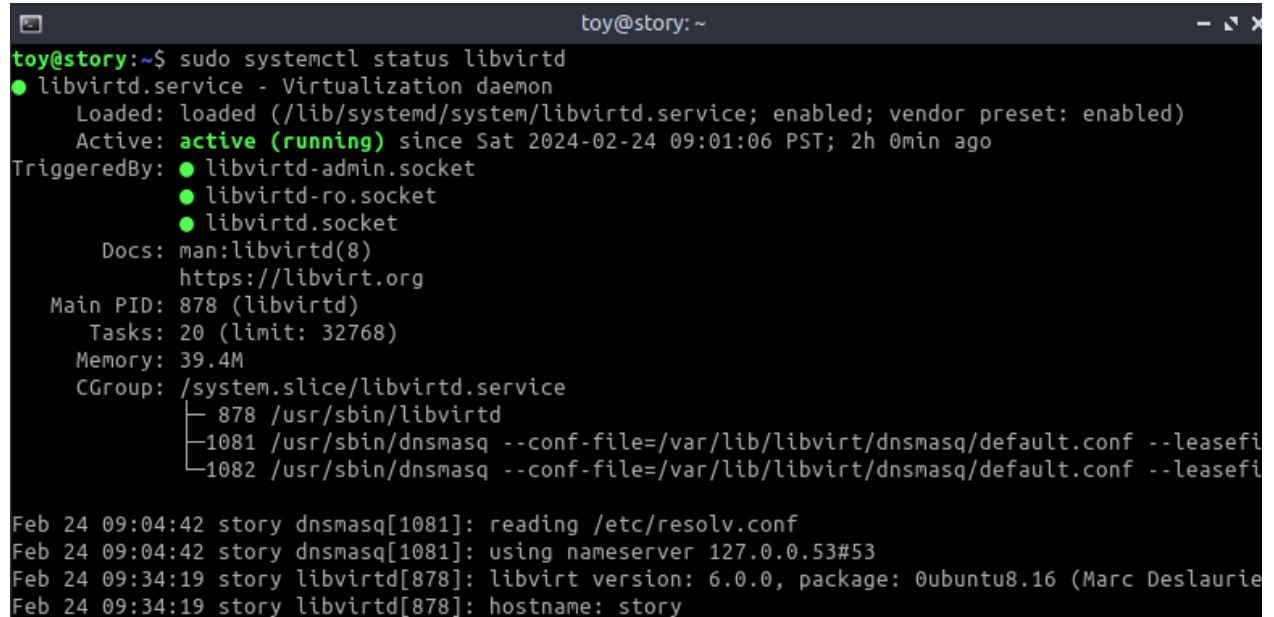
```
$ sudo apt install -y qemu qemu-kvm libvirt-daemon libvirt-clients bridge-utils virt-manager
```

- The **qemu** (quick emulator) is an application that allows you to perform hardware virtualization.
- The **qemu-kvm** is the main KVM package.
- The **libvirt-daemon** is the virtualization daemon.
- The **bridge-utils** create a bridge connection to allow other users to access a virtual machine other than the host system.

- The **virt-manager** is a graphical user interface application to manage VMs.

4. Confirm virtualization daemon is running

```
$ sudo systemctl status libvirtd
```



```
toy@story: ~  
toy@story:~$ sudo systemctl status libvirtd  
● libvirtd.service - Virtualization daemon  
   Loaded: loaded (/lib/systemd/system/libvirtd.service; enabled; vendor preset: enabled)  
   Active: active (running) since Sat 2024-02-24 09:01:06 PST; 2h 0min ago  
TriggeredBy: ● libvirtd-admin.socket  
              ● libvirtd-ro.socket  
              ● libvirtd.socket  
   Docs: man:libvirtd(8)  
         https://libvirt.org  
 Main PID: 878 (libvirtd)  
    Tasks: 20 (limit: 32768)  
  Memory: 39.4M  
   CGroup: /system.slice/libvirtd.service  
           └─ 878 /usr/sbin/libvirtd  
             1081 /usr/sbin/dnsmasq --conf-file=/var/lib/libvirt/dnsmasq/default.conf --leasefi  
             1082 /usr/sbin/dnsmasq --conf-file=/var/lib/libvirt/dnsmasq/default.conf --leasefi  
  
Feb 24 09:04:42 story dnsmasq[1081]: reading /etc/resolv.conf  
Feb 24 09:04:42 story dnsmasq[1081]: using nameserver 127.0.0.53#53  
Feb 24 09:34:19 story libvirtd[878]: libvirt version: 6.0.0, package: 0ubuntu8.16 (Marc Deslaurie  
Feb 24 09:34:19 story libvirtd[878]: hostname: story
```

Options: Boot virtualization daemon by default \$ **sudo systemctl enable --now libvirtd**

5. Confirm KVM modules are loaded

```
$ lsmod | grep -i kvm
```

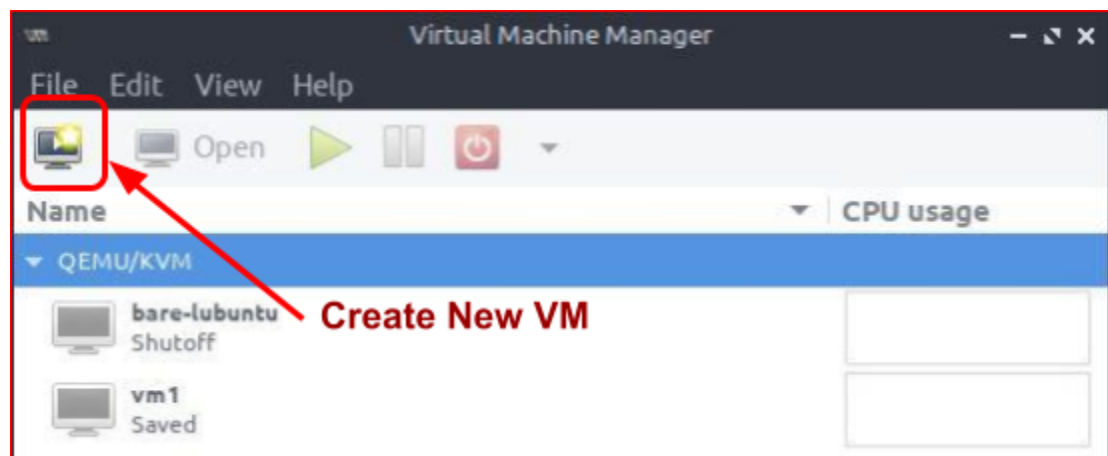


```
toy@story: ~  
toy@story:~$ lsmod | grep -i kvm  
kvm_intel          376832    0  
kvm                1015808    1 kvm_intel  
toy@story:~$
```

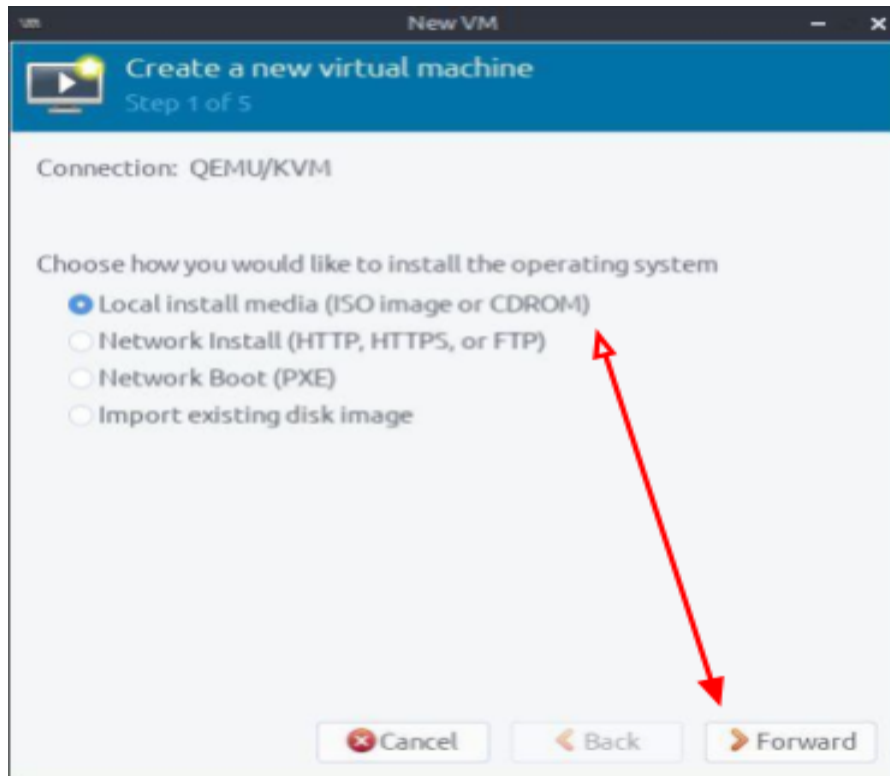
6. Create a VM by virt-manager

```
sudo virt-manager
```

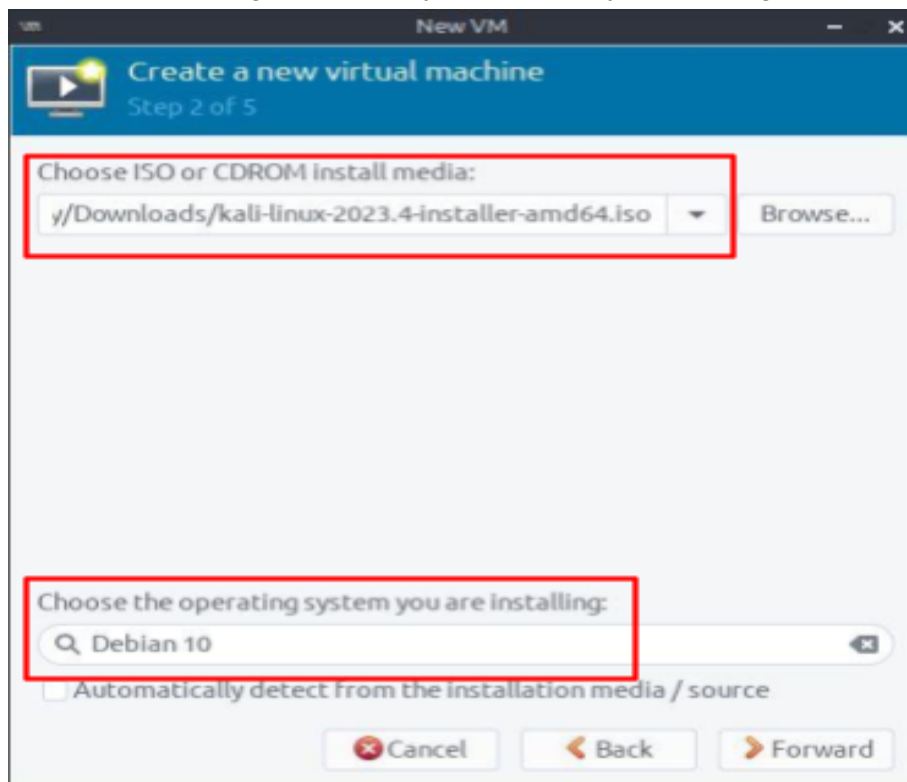
- a. Click on the “Create a new virtual machine” button



- b. Select the first option – Local Install Media (ISO image or CDROM). Next, click the ‘Forward’ button to continue



- c. Select the ISO image and specify the OS family of the image.



- d. Select the number of CPUs and set the memory size

The screenshot shows the 'New VM' window at Step 3 of 5, titled 'Create a new virtual machine'. The main heading is 'Choose Memory and CPU settings:'. Below this, there are two settings: 'Memory' and 'CPUs'. The 'Memory' setting is set to '4096' with a range of 'Up to 7756 MiB available on the host'. The 'CPUs' setting is set to '2' with a range of 'Up to 4 available'. At the bottom, there are three buttons: 'Cancel', 'Back', and 'Forward'.

- e. Specify a name for your virtual machine and click on the 'Finish' button.

The screenshot shows the 'New VM' window at Step 5 of 5, titled 'Create a new virtual machine'. The main heading is 'Ready to begin the installation'. Below this, there is a 'Name' field with the value 'kali'. Other settings are displayed: 'OS: Debian 10', 'Install: Local CDROM/ISO', 'Memory: 4096 MiB', 'CPUs: 2', and 'Storage: 50.0 GiB /var/lib/libvirt/images/kali.qcow2'. There is a checkbox for 'Customize configuration before install' which is unchecked. Under 'Network selection', it shows 'Virtual network \'default\' : NAT'. At the bottom, there are three buttons: 'Cancel', 'Back', and 'Finish'.