

# KVM Virt-Manager Setup



Assignment No. 2

Paravirtualization: KVM with Virt-Manager

Install KVM on any Linux platform and alongside install virt-manager and run virtualized OS on the KVM through virt-manager.

Which response do you prefer? Responses may take a moment to load.



## Response 1

To set up KVM (Kernel-based Virtual Machine) and Virt-Manager on a Linux platform, follow these steps:

1. **Verify Hardware Virtualization Support:** Ensure your CPU supports virtualization and that it's enabled in the BIOS.

```
bash

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

A result greater than 0 indicates support.

2. **Update System Packages:**

```
bash

sudo apt update && sudo apt upgrade -y
```

3. **Install KVM and Related Packages:**

```
bash

sudo apt install qemu-kvm libvirt-daemon-system libvirt-clients bridge-utils virt-m
anager -y
```

This command installs KVM, QEMU, Virt-Manager, and other necessary tools.

[Linux Tech](#)

4. **Start and Enable the Libvirt Service:**

```
bash

sudo systemctl start libvirtd
sudo systemctl enable libvirtd
```

## 5. Add User to Necessary Groups:

```
bash

sudo usermod -aG libvirt $USER
sudo usermod -aG kvm $USER
```

After running these commands, log out and back in to apply group changes.

## 6. Launch Virt-Manager: Open Virt-Manager from your application menu or by running:

```
bash

virt-manager
```

Virt-Manager provides a graphical interface to manage virtual machines.

[LinuxConfig](#)

## 7. Create a New Virtual Machine:

- Click on "Create a new virtual machine".
- Choose the installation method (e.g., local ISO image).
- Allocate CPU and memory resources.
- Create or select a disk image for storage.
- Configure network settings.
- Review the settings and finish the setup.

By following these steps, you can install KVM and Virt-Manager on your Linux system and create virtual machines efficiently.

Sources



I prefer this response



## Response 2

Here's a step-by-step guide for your assignment to install KVM and Virt-Manager on a Linux platform, followed by running a virtualized OS:

## Step 1: Install KVM

## 1. Check Hardware Virtualization Support:

- Open a terminal and run:

```
bash

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

- A non-zero output indicates your CPU supports virtualization.

## 2. Install KVM and Required Packages:

- Update your package list:

```
bash

sudo apt update
```

- Install KVM and related tools:

```
bash

sudo apt install -y qemu-kvm libvirt-daemon-system libvirt-clients bridge-util
s virt-manager
```

## 3. Verify Installation:

- Check if KVM modules are loaded:

```
bash

lsmod | grep kvm
```

- Ensure `libvirtd` service is active:

```
bash

sudo systemctl status libvirtd
```

## 4. Add User to `libvirt` Group (Optional):

- To avoid needing `sudo` for libvirt commands, add your user to the `libvirt` group:

```
bash

sudo usermod -aG libvirt $(whoami)
```

- Log out and back in for the changes to take effect.

## Step 2: Configure Virt-Manager

### 1. Launch Virt-Manager:

- Start the graphical interface:

```
bash

virt-manager
```

- You may need to run it as root if permissions aren't configured:

```
bash

sudo virt-manager
```

### 2. Connect to KVM Host:

- In Virt-Manager, confirm you're connected to the local QEMU/KVM hypervisor. The connection should appear automatically.

## Step 3: Create a Virtual Machine

### 1. Download an ISO Image:

- Download an operating system ISO (e.g., Ubuntu, Fedora) to use as the installation media.

### 2. Create a New Virtual Machine:

- Click **"New Virtual Machine"** in Virt-Manager.
- Follow the wizard:
  - Select **"Local install media (ISO)"**.
  - Browse to your downloaded ISO file.
  - Allocate CPU, RAM, and disk space as required.

### 3. Start the Virtual Machine:

- After configuration, start the virtual machine and proceed with the OS installation.

## Step 4: Manage and Use the Virtualized OS

### 1. Access the VM:

- Double-click the virtual machine in Virt-Manager to open the console.
- Complete the OS installation process.

### 2. Configure Networking:

- Ensure the VM has internet access by selecting a NAT or bridged network during setup.

### 3. Optimize Performance (Optional):

- Install guest additions or tools specific to the guest OS to improve integration and performance.

---

## Step 5: Verification

### 1. List Running VMs:

- Use the `virsh` command-line tool to verify the virtual machine is running:

```
bash
```

```
virsh list
```

### 2. Test the Virtualized OS:

- Boot into the OS and perform basic operations to ensure everything works as expected.

---

Let me know if you need detailed troubleshooting or additional tips!

I prefer this response