Alright!
Let's go step-by-step for your task:
"Installation and Configuration of virtualization using KVM"
(Assuming you are doing this on **Ubuntu/Linux**.)

📚 What is KVM?

- **KVM (Kernel-based Virtual Machine)** is a Linux kernel module that allows your Linux machine to act as a **hypervisor** — meaning, it can host virtual machines (VMs).
- It is used for virtualization at the hardware level (just like VirtualBox, VMware, etc.).

🛠️ Steps to Install and Configure KVM

1. Check if your system supports virtualization

First, make sure your system (CPU) supports hardware virtualization (**Intel VT-x** or **AMD-V**).

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

- If you get output **1** or more → **Good, supported.**
- If you get output **0** → **Not supported** or virtualization is disabled in BIOS (you need to enable it).

2. Update your system

Always update your system first:

```
sudo apt update
```

```
sudo apt upgrade -y
```

3. Install KVM and related packages

Install the necessary KVM components:

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

Package	Purpose
`qemu-kvm`	The KVM module (core virtualization)
`libvirt-daemon-system`	Allows managing KVM as a service
`libvirt-clients`	Client utilities to interact with libvirt
`bridge-utils`	Needed for network bridging
`virt-manager`	GUI (graphical interface) to create and manage VMs easily

4. Verify KVM installation

Check if the **kvm** modules are loaded:

```
lsmod | grep kvm
```

You should see something like:

```
kvm_intel    (or kvm_amd)
kvm
```

Also, check if the KVM service is running:

```
sudo systemctl status libvirtd
```

If it's inactive, start and enable it:

```
sudo systemctl start libvirtd
sudo systemctl enable libvirtd
```

5. Add your user to the libvirt group

To manage VMs without sudo:

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

Important: Log out and log back in (or reboot) to apply group changes.

6. Create and manage Virtual Machines

Now, you can launch the **Virtual Machine Manager** (GUI):

```
virt-manager
```

From here:

- Click **Create New Virtual Machine**.
- Select installation type (ISO file or network installation).
- Assign CPU, RAM, storage.
- Launch your VM!

That's it! KVM virtualization is now installed and working.

⚡ Common Errors and Their Solutions

Error	Reason	Solution
`/dev/kvm permission denied`	User not in libvirt group	Add user to `libvirt` group and reboot
`Virtualization not supported`	CPU doesn't support VT-x/AMD-V or it's disabled	Enable virtualization in BIOS
`virt-manager: command not found`	virt-manager not installed	`sudo apt install virt-manager`
`No Virtual Networks`	Default network not active	`sudo virsh net-start default`

📌 Quick Summary Diagram (ASCII Style)

```
+--------------------+

| Check Virtualization|----> (egrep vmx|svm)

+--------------------+

          |

          v

+--------------------+

| Install KVM Packages |

+--------------------+
```

```
          |
          v
+--------------------+
| Start libvirtd Service |
+--------------------+
          |
          v
+--------------------+
| Add User to libvirt Group |
+--------------------+
          |
          v
+--------------------+
| Create VMs (virt-manager) |
+--------------------+
```

Your Task (Simple Version)

Step	Command
Check virtualization	`egrep -c '(vmx
Update packages	`sudo apt update && sudo apt upgrade -y`
Install KVM	`sudo apt install qemu-kvm libvirt-daemon-system libvirt-clients bridge-utils virt-manager -y`
Start service	`sudo systemctl start libvirtd && sudo systemctl enable libvirtd`
Add user to group	`sudo usermod -aG libvirt $(whoami)`
Launch GUI	`virt-manager`

Would you also like me to give a **ready practical file/handwritten notes format**? (useful if you need to submit it!)

Would you also like a **sample VM creation example** (like Ubuntu server)?