Kvm: Manage VM’s like a PRO

Check GitHub for helpful DevOps tools:

Michael Robotics

Hi, I'm Michal. I'm a Robotics Engineer and DevOps enthusiast. My mission is to create skill-learning platform that combats information overload by adhering to the set of principles: simplify, prioritize, and execute.

https://github.com/MichaelRobotics

Ask Personal AI Document assistant to learn interactively (FASTER)!

1 Copy link

2 Go to website

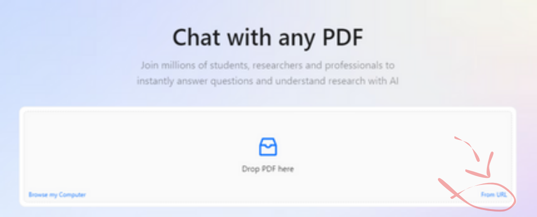
3 Paste URL

4 Chat with Document

1 https://github.com/MichaelRobotics/DevOpsTools/KVMisntall.pdf

Click there to go to website 2

3



Chat with any PDF

Join millions of students, researchers and professionals to instantly answer questions and understand research with AI

Ask questions about document! 4

Complety new to VM’s and Linux?

If you are completely new to this topic, using a document assistant to understand the many definitions can be helpful. However, the best way to start is by watching this video, which I believe provides the best explanation for beginners starting their journey with Linux and virtualization:

QEMU/KVM for absolute beginners

On this episode of Veronica Explains, I explain the absolute basics of hypervisors generally, KVM specifically, and virt-manager graphically.

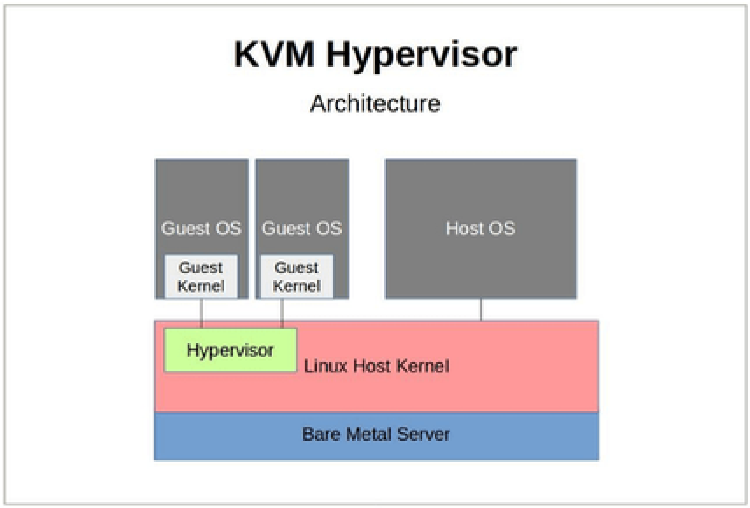
https://youtu.be/VeronicaExplains

What is KVM?

Kernel-based Virtual Machine (KVM) is an open-source virtualization technology built into Linux kernel, enabling a host machine to run multiple, isolated virtual environments (VMs) while leveraging Linux features, updates, and security.

How KVM work?

KVM transforms Linux into a type-1 (bare-metal) hypervisor, utilizing Linux's existing components like memory management, process scheduling, I/O stack, and device drivers to run VMs. Each VM functions as a regular Linux process with dedicated virtual hardware.



KVM: Why and When

Use KVM when you need a cost-effective, open-source virtualization solution that integrates seamlessly with the Linux ecosystem, providing high performance and robust security features without additional licensing fees. Compared to commercial solutions like VMware ESXi, KVM is ideal for organizations seeking flexibility and scalability while leveraging community support and avoiding vendor lock-in.

KVM works best for businesses needing to deploy numerous virtual machines on existing Linux infrastructure, maximizing resource utilization and cost efficiency without sacrificing performance.

System Requirements

- 8 gb ram
- 50 free gb storage
- ubuntu 22.04

If you want to install it on a different Linux distro, ask in the comments and I will write an Ansible playbook or bash script.

KVM: Main components & packages

- libvirt-daemon - runs virtualization in background
- qemu-kvm – An opensource emulator and virtualization package that provides hardware emulation.
- virt-manager – A Qt-based graphical interface for managing virtual machines via the libvirt daemon.
- libvirt-daemon-system – A package that provides configuration files required to run the libvirt daemon.
- virtinst – A set of command-line utilities for provisioning and modifying virtual machines.
- libvirt-clients – A set of client-side libraries and APIs for managing and controlling virtual machines & hypervisors from the command line.
- bridge-utils – A set of tools for creating and managing bridge devices.

KVM: How to install

1) Check if Virtualization is enabled

Before you proceed any further, you need to check if your CPU supports KVM virtualization. For this to be possible, your system needs to either have a VT-x(vmx) Intel processor or an AMD-V (svm) processor.

This is achieved by running the following command. if the output is greater than 0, then virtualization is enabled. Otherwise, virtualization is disabled and you need to enable it.

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

If Virtualization is not enabled, be sure to enable the virtualization feature in your system's BIOS settings.

2) Install KVM

Next, run the command below to install KVM and additional virtualization packages.

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

3) Add the Users to KVM and Libvirt Group

This step helps you add your Users to KVM and Libvirt groups without any errors. Follow the command to add the user to KVM:

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

Add the user to the Libvirt group with the command:

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

4) Virtualization services

The ‘libvirtd’ command activates the services that aid Virtualization. The command is as follows:

```
$ sudo systemctl enable libvirtd
```

Run the following command to initiate the services:

```
$ sudo systemctl start libvirtd
```

Now, to verify the status of the services, run the below command. The output ‘active’ confirms the live status of the services.

```
$ sudo systemctl status libvirtd
```

How to manage VM using CLI (Like SysAdmin)

1) Setup storage for VM’s ISO images and VM systems

Create directories for storage

```
$ sudo mkdir -p /media/isos
$ sudo mkdir -p /media/qemu
```

Change group ownership to libvirt

```
$ sudo chown :libvirt /media/isos
$ sudo chown :libvirt /media/qemu
```

Set permissions to allow read/write access to the group

```
$ sudo chmod 775 /media/isos
$ sudo chmod 775 /media/qemu
```

Set the setgid bit to ensure new files inherit group ownership

```
$ sudo chmod g+s /media/isos
sudo chmod g+s /media/qemu
```

Add your user to the libvirt group

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

2) Download ISO image Link and move image into /media/isos/

```
$ virt-install \
 --name guest1-ubuntu24.04 \
 --memory 8192 \
 --vcpus 2 \
 --disk path=/media/qemu/ubuntu-vm.qcow2,size=15 \
 --cdrom /media/isos/ubuntu-24.04-desktop-amd64.iso \
```

3) Configure new VM

After starting the VM installation, a virt-viewer window should pop up displaying the VM's graphical user interface (GUI). Follow the instructions provided by the installer to complete the setup.

4) basic commands to manage VM’s

Open VM GUI

```
$ virt-viewer --connect qemu:///system <vm_name>
```

Get acces VM to CLI

There are two possibilities 1) install ssh server on vm and connect 2) use virsh

```
$ virsh console vm-name
```

I had issues with this one, to resolve go to ***common troubleshooting*** part of this pdf

Function to list all VMs

```
$ virsh list --all
```

Function to start a VM

```
$ virsh start <vm_name>
```

Function to shutdown a VM

```
$ virsh shutdown <vm_name>
```

Function to force stop a VM

```
$ virsh destroy <vm_name>
```

Function to reboot a VM

```
$ virsh reboot <vm_name>
```

Function to suspend a VM

```
$ virsh suspend <vm_name>
```

Function to resume a VM

```
$ virsh resume <vm_name>
```

Function to delete a VM

```
$ virsh undefine <vm_name>
```

Function to get VM info

```
$ virsh dominfo <vm_name>
```

How to manage VM using GUI (Casual way)

Most common virtual machine manager is VirtualBox, and it compares poorly to KVM virt-manager. QEMU, especially when combined with KVM, provides extensive architecture support and near-native performance, making it ideal for advanced and diverse virtualization needs. Its flexibility and lightweight nature offer more low-level customization options compared to VirtualBox.

There is a lot of tutorials how to manage VM's using virt-manager, this is short and great:

QEMU/KVM for absolute beginners

On this episode of Veronica Explains, I explain the absolute basics of hypervisors generally, KVM specifically, and virt-manager graphically.

https://youtu.be/VeronicaExplains

Common troubleshooting

1) waiting for display

Essentially, you need to disable the screen timeout feature in your VM's power saving settings.

2) **virsh cannot connect to CLI**

Great thread, look for answer from user292283:

virsh: VM console does not show any output

Working example of using Debian jessie as host and guest operating system.
1. create a VM using virt-install or virt-manager In any case you will....

https://serverfault.com/questions

StackExchange

3) **Check the man page**

4) **If everything is a complete mess**

delete KVM and install again

KVM: How to remove

3) Step 1: Stop the libvirt service

As the first step, you need to stop the libvirt service:

```
$ sudo systemctl stop libvirtd
```

4) Step 2: Remove the KVM packages

Next, remove the KVM packages by utilizing the command below, in the terminal:

```
$ sudo apt-get remove qemu-kvm virt-manager libvirt-daemon-system
virtinst libvirt-clients bridge-utils
```

Step 3: Remove the dependencies

```
$ sudo apt-get autoremove -y
```

Learn more about KVM

Check RedHat, they have great docs

Virtualization Deployment and Administration Guide

Installing, configuring, and managing virtual machines on a RHEL physical machine

https://docs.redhat.com/en

Share , ask, and check GitHub for scripts and playbooks created to automate instalation.

Check my GitHub

Michael Robotics

Hi, I'm Michal. I'm a Robotics Engineer and DevOps enthusiast. My mission is to create skill-learning platform that combats skill information overload by adhering to the set of principles: simplify, prioritize, and execute.

https://github.com/MichaelRobotics

PS.

If you need playbook/bash script to install KVM on specific linux distro, ask me in commnets or DM me and i will add it into my Git folder!