

# Complete Installation of KVM, QEMU and Virt Manager on Arch Linux and Manjaro | ComputingForGeeks

Here is the Complete guide on how to install KVM, QEMU and Virt Manager on Arch Linux and Manjaro. KVM is one of the most used Virtualization software in Linux World. In fact, most cloud providers use KVM as their Hypervisor of choice. Big projects including Openstack use KVM as default Virtualization tool.

In this tutorial, we'll install KVM on Arch Linux and set Kernel modules required to automatically load at boot. We'll also install a guest machine running CentOS 7.

Our previous tutorials on KVM include:

[How to Create and Configure Bridge Networking For KVM in Linux](#)

[How To Clone and Use KVM Virtual Machine in Linux](#)

[How to extend/increase KVM Virtual Machine \(VM\) disk size](#)

## Step 1: Install KVM packages

First step is installing all packages needed to run KVM:

```
sudo pacman -S qemu virt-manager virt-viewer dnsmasq vde2 bridge-utils openbsd-netcat
```

Also install *ebtables* and *iptables* packages:

```
sudo pacman -S ebtables iptables
```

## Step 2: Install libguestfs on Arch Linux / Manjaro

libguestfs is a set of tools used to access and modify virtual machine (VM) disk images. You can use this for:

- viewing and editing files inside guests
- scripting changes to VMs
- monitoring disk used/free statistics
- creating guests

- P2V, V2V
- performing backup e.t.c

First enable AUR on */etc/pacman.conf*

```
$ sudo vim /etc/pacman.conf

# Should have below lines
[archlinuxfr]
SigLevel = Never
Server = http://repo.archlinux.fr/$arch
```

Then update your Pacman databases.

```
sudo pacman -Syy
```

Install libguestfs tools with AUR helper.

```
sudo pacman -S yaourt
yaourt -S --noconfirm --needed libguestfs
```

## Step 3: Start KVM libvirt service

Once the installation is done, start and enable libvirtd service to start at boot:

```
sudo systemctl enable libvirtd.service
sudo systemctl start libvirtd.service
```

Status show it's running:

```
$ sudo systemctl status libvirtd.service
● libvirtd.service - Virtualization daemon
   Loaded: loaded (/usr/lib/systemd/system/libvirtd.service; enabled; vendor
  preset: disabled)
   Active: active (running) since Thu 2019-04-18 20:55:34 EAT; 11h ago
     Docs: man:libvirtd(8)
           https://libvirt.org
  Main PID: 646 (libvirtd)
    Tasks: 26 (limit: 32768)
   Memory: 74.0M
    CGroup: /system.slice/libvirtd.service
            └─ 646 /usr/bin/libvirtd
            └─ 754 /usr/bin/dnsmasq
--conf-file=/var/lib/libvirt/dnsmasq/default.conf --leasefile-ro
--dhcp-script=/usr/lib/libvirt/libvirt_leases>
            └─ 755 /usr/bin/dnsmasq
--conf-file=/var/lib/libvirt/dnsmasq/default.conf --leasefile-ro
--dhcp-script=/usr/lib/libvirt/libvirt_leases>
            └─ 777 /usr/bin/dnsmasq --conf-file=/var/lib/libvirt/dnsmasq/docker-
machines.conf --leasefile-ro --dhcp-script=/usr/lib/libvirt/libvir>
            └─ 778 /usr/bin/dnsmasq --conf-file=/var/lib/libvirt/dnsmasq/docker-
machines.conf --leasefile-ro --dhcp-script=/usr/lib/libvirt/libvir>
            └─25924 /usr/bin/dnsmasq --conf-file=/var/lib/libvirt/dnsmasq/vagrant-
libvirt.conf --leasefile-ro --dhcp-script=/usr/lib/libvirt/libvir>
```

```
└─25925 /usr/bin/dnsmasq --conf-file=/var/lib/libvirt/dnsmasq/vagrant
└─25960 /usr/bin/dnsmasq
--conf-file=/var/lib/libvirt/dnsmasq/fed290.conf --leasefile-ro
--dhcp-script=/usr/lib/libvirt/libvirt_leasesh>
```

## Step 4: Enable normal user account to use KVM

Since we want to use our standard Linux user account to manage KVM, let's configure KVM to allow this.

Open the file `/etc/libvirt/libvirtd.conf` for editing.

```
sudo pacman -S vim
sudo vim /etc/libvirt/libvirtd.conf
```

Set the UNIX domain socket group ownership to libvirt, (around line **85**)

```
unix_sock_group = "libvirt"
```

Set the UNIX socket permissions for the R/W socket (around line **102**)

```
unix_sock_rw_perms = "0770"
```

Add your user account to libvirt group.

```
sudo usermod -a -G libvirt $(whoami)
newgrp libvirt
```

Restart libvirt daemon.

```
sudo systemctl restart libvirtd.service
```

## Step 5: Enable Nested Virtualization (Optional)

Nested Virtualization feature enables you to run Virtual Machines inside a VM. Enable Nested virtualization for `kvm_intel` by enabling kernel module as shown.

```
sudo modprobe -r kvm_intel
sudo modprobe kvm_intel nested=1
```

To make this configuration persistent, run:

```
echo "options kvm-intel nested=1" | sudo tee /etc/modprobe.d/kvm-intel.conf
```

Confirm that Nested Virtualization is set to Yes:

```
$ systool -m kvm_intel -v | grep nested
    nested                = "Y"
    nested_early_check    = "N"
$ cat /sys/module/kvm_intel/parameters/nested
Y
```

## Step 6: Using KVM on Arch Linux / Manjaro

You have successfully installed KVM, QEMU and Virt Manager on Arch Linux and Manjaro. You are now good to go. You can install a Linux or Windows OS Virtual Machine and build your study labs.



Here are our few guides to get you started

[Using Vagrant with Libvirt on Linux.](#)

[How to Provision VMs on KVM with Terraform](#)

[How to Create CentOS / Fedora / RHEL VM Templates on KVM](#)

[RHEL and CentOS Kickstart on KVM Automated Installation With virt-install](#)

Other interesting guides.

[How to mount VM virtual disk on KVM hypervisor](#)