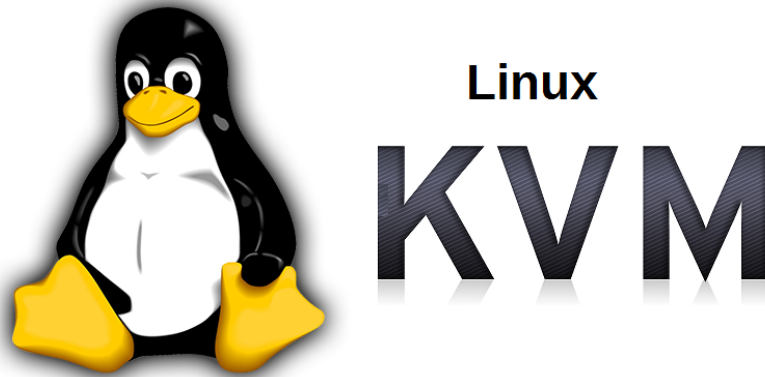


Linux: How to set KVM Virtual Machine to autostart on system boot

Author : admin



Recently I've written a short article on [how to make auto boot OpenVZ container Virtual Machine if system gets rebooted](#) as I had the task to do so in daily job as sysadmin.

Few days ago after some power outage in one of Frankfurt based Data Centers, some Rack tech equipment has been temporary powered of and once the electricity was up again, some Linux servers running **Kernel Based Virtualization (KVM)** some of the virtual machines did not managed to automatically start up and we had to manually start them one by one.

To manually start each of the machines that did not start up had to do the trivial:

```
[root@hypervisor ~]# virsh list
```

```
Id   Name                               State
```

```
-----  
3    VM500                             running
```

```
[root@hypervisor ~]# virsh dominfo VM500
```

```
Id:      3
```

```
Name:    VM500
```

```
UUID:      82075903-a4ce-5ec0-a3d8-9ee83d85fc75
OS Type:    hvm
State:      running
CPU(s):     2
CPU time:   247407.9s
Max memory: 2097152 KiB
Used memory: 2097152 KiB
Persistent: yes
Autostart:  disable
Managed save: no
Security model: selinux
Security DOI: 0
Security label: system_u:system_r:svirt_t:s0:c447,c723 (permissive)
```

```
[root@hypervisor ~]# virsh start VM_domain_Name
```

Of course logically to prevent future issues in case if Linux server gets suddenly rebooted due to whatever reason, we had to configure the machines in questions to automatically boot on OS system boot.

In some rare cases if above start command does not help you might want to check what is the status of **libvirtd** and investigate the logs in **/var/log/libvirt/** i.e. **/var/log/libvirt/libvirt.log**

An alternative but more complicated way to set virtual machine to automatically start on boot is by using default location for automatic VM start just like OpenXEn has its **/etc/xen/auto/** directory from where each soft symlinked VM configuration from **/etc/xen/VM_name.cfg** is started in KVM Hypervisor hosts to auto boot a certain **vm** you have to link **/etc/libvirt/qemu/VM-to-autoboot-name.xml** to **/etc/libvirt/qemu/autostart/VM-to-autoboot-name.xml**.

```
[root@hypervisor ~]# systemctl status libvirtd
? libvirtd.service - Virtualization daemon
   Loaded: loaded (/usr/lib/systemd/system/libvirtd.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2021-06-29 10:21:30 CEST; 2 weeks 2 days ago
     Docs: man:libvirtd(8)
           https://libvirt.org
   Main PID: 1809 (libvirtd)
```

```
Tasks: 22 (limit: 32768)
CGroup: /system.slice/libvirtd.service
    ??1809 /usr/sbin/libvirtd
    ??2335 /usr/sbin/dnsmasq --conf-file=/var/lib/libvirt/dnsmasq/default.conf --leasefile-ro
--dhcp-script=/us...
    ??2336 /usr/sbin/dnsmasq --conf-file=/var/lib/libvirt/dnsmasq/default.conf --leasefile-ro
--dhcp-script=/us...
    ??2386 /usr/sbin/dnsmasq --conf-file=/var/lib/libvirt/dnsmasq/Host-Only.conf --leasefile-ro
--dhcp-script=/...
    ??2387 /usr/sbin/dnsmasq --conf-file=/var/lib/libvirt/dnsmasq/Host-Only.conf --leasefile-ro
--dhcp-script=/...
```

If for some reason libvirtd is not running or disabled state you will have to enable it with:

```
[root@hypervisor ~]# systemctl enable libvirtd
```

If the virtualization is running on some RPM based distribtuion OpenSuse whatever and **libvirtd** is controlled via **chkconfig** (redhat runlevel alternative system), you will have to :

```
[root@hypervisor ~]# chkconfig libvirtd on
```

Disabling KVM Virtual Machine from auto start on server boot

```
[root@hypervisor ~]# virsh autostart Debian10 --disable
Domain Debian10 unmarked as autostarted
```