virsh

This article was made for the following Ubuntu versions tested:

This article is not tested with any currently supported Ubuntu version! Please test this article for an Ubuntu version, which is currently supported. For this purpose, the instructions **for testing articles** must be observed.

Article for advanced users

This article requires more experience in dealing with Linux and is therefore only for advanced users intended.

To understand this article, the following pages are helpful:

1. **of programs**
2. **Opening a terminal**
3. **Starting a program**
4. **KVM** kernel-based virtual machines
5. **QEMU** Free emulator and virtualizer
6. **Virtualization** General Introduction to Virtualization

Table of contents

[https://media-cdn.ubuntu-de.org/wiki/attachments/07/05/libvirt-logo.png]**virsh** [http://libvirt.org/virshcmdref.html] is a tool for managing virtual machines[6] under Xen, QEMU, KVM, LXC, Test, OpenVZ, VirtualBox, OpenNebula, or VMware ESX. You can create, delete, boot, stop and manage a machine from the command line. Virsh is of interest to the advanced Linux administrator to realize simple script automatisms for virtualization. This article is especially designed for using virsh with **KVM** [https://wiki.ubuntuusers.de/KVM/]. If you have not yet dealt with virtualization[6] or if you have previously only been used to graphical interfaces such as those of Virtual Box and now want to go deeper, if possible, you should start with the pure KVM or QEMU technology on the console, in order to get to know the configuration specifications, e.g. for a Windows machine.

For stable operation, a lot to be clarified in advance on topics that are the best virtual network card or which disk file type are the best in the event[4][5]. The advantages of virsh are then in the simplified management of virtual instances.

virsh is a part of the **Libvirt project** [http://libvirt.org/], which provides a collection of commands (librus) for various programming languages that can be used to control the **hypervisors** [https://de.wikipedia.org/wiki/Hypervisor]. It provides the sets of instructions in the shell.

Installation

For virsh you need the packages [1].

- **virtinst**
- **libvirt-bin**

Command to install the packages:

```
sudo apt-get install virtinst libvirt-bin
```

Or install with **apturl** [https://wiki.ubuntuusers.de/apturl/], link: **apt://virtinst,libvirt-bin** [apt://virtinst,libvirt-bin]

Connection to the hypervisor

To connect to the **hypervisor** [https://de.wikipedia.org/wiki/Hypervisor] (this can be the local or a server accessible from the network, in most cases you use the local hypervisor), this terminal command[2] is used:

```
virsh connect qemu:///system
```

The resulting output:

```
Connecting to uri: qemu:///system
```

Create the virtual machine

Virtual machines managed with virsh are created by defining the machine in a libvirt XML file. As far as you don't know the syntax yet, you should `virt-install`create a first XML file or the virtual machine completely with it.

The following command creates a first Linux VM using virt-install. In this case **foo** [https://de.wikipedia.org/wiki/Foo], it has 512 MB of ram, has a CPU, a 12 GB hard disk and the local CDROM drive. If you have started a VM under KVM directly, you will quickly find your way around this place. More details which setting match which VM can be found at QEMU[5] or KVM[6].

```
virt-install --connect qemu:///system -n foo -r 512 --vcpus=1 -f /var/lib/libvirt/images/festplatte1.img -s 12
--vnc --cdrom /media/cdrom0 --noautoconsole --os-type linux --accelerate --network=bridge:br0,model=virtio -m
00:00:00:00:00:07 -k de
```

The virtual disk should be in the folder **/var/lib/libvirt/images**. For this folder there is an exception rule in the rules of the pre-installed **AppArmor** [https://wiki.ubuntuusers.de/AppArmor/] protection software, located in the **/etc/apparmor.d/usr.libvirt.virt-a-helper** file. Other folders always fail with `Permission denied`.

Some basic options from virt-install

Option	Statement
`-h, --help`	Shows the help notifications
`-n NAME, --name=NAME`	Name of the guest authority or in virt-install domain
`-r MEMORY, --ram=MEMORY`	Working enriched
`--vcpus=VCPUS`	Number of virtual CPUs for the VM
`-f DISKFILE, --file=DISKFILE`	File name and path to be used as a hard disk image
`-s DISKSIZE, --file-size=DISKSIZE`	Hard disk size (if it has not been created yet) in gigabytes
`--import`	Do not rebuild the disk image (record existing VMs or backups)
`-w NETWORK, --network=NETWORK`	connects the guest to a virtual network, forwardet to a physical power adapter with NAT
`--vnc`	Use VNC as the graphical interface provider of the VM
`--vncport=VNCPORT`	defines the port for VNC

Working with running virtual machines

If you first run a VM, you can edit it in different ways with virsh, as far as you are connected to the hypervisor.

- Starting the VM
  ```
  virsh start foo
  ```
- Restart the VM
  ```
  virsh reboot foo
  ```
- Turning off the VM via Acpi

```
virsh shutdown foo
```

Note:

For the shutdown to work via ACPI, it must also be installed and activated in the VM. Under Ubuntu is the package

- **acpid**

Command to install the packages:

```
sudo apt-get install acpid
```

Or install with **apturl**, link: **apt://acpid**

necessary.

Alternatively, only one

```
virsh destroy foo
```

thus pulling the plug.

- Pause or Freezing the VM

  ```
  virsh suspend foo
  ```

- Unpicking up the pause state of a VM

  ```
  virsh resume foo
  ```

- List VMs

  ```
  virsh list
  ```

```
 Id Name                 State
----------------------------------
  1 foo                  running
```

Deletion of a VM

To delete a VM, you have to destroy it first to remove the definition in libvirt.

```
virsh destroy foo_new
```

```
virsh undefine foo_new
```

Other items

XML format in libvirt or VM configuration

You can also create the VMs directly from the configuration XML file[3]. You can export such a file to **the /tmp** directory by entering the following in the console:

```
virsh dumpxml foo > /tmp/foo.xml
```

Once you have adjusted the XML file, you can create a new virtual machine from it and run it automatically:

```
virsh create /tmp/foo_new.xml
```

Alternatively, you can only define the VM without starting it:

```
virsh define /tmp/foo_new.xml
```

As far as you can control the **VIM** [https://wiki.ubuntuusers.de/VIM/] syntax, it is best to edit the XML file with the command `virsh edit`. This allows you to easily adjust the configuration. As with VIM, you can also `:i`for editing and `:wq`for saving and closing adjust the configuration.

```
virsh edit foo
```

Much more extensive options for use can be found in the **man page** [https://wiki.ubuntuusers.de/man/] of virsh.

VM console

Sometimes it is useful to use the console of a running VM to get debug information or to receive otherwise. However, please do not confuse it with the QEMU monitor:

```
virsh console foo
```

This command connects to the VM foo. Then leave the console with the keys pressed simultaneously Strg + Alt Gr + 9 9

Tools

With virt-top you can get an overview of the load balancing among the running virtual machines.

- **virt-top**

Command to install the packages:

```
sudo apt-get install virt-top
```

Or install with **apturl** [https://wiki.ubuntuusers.de/apturl/], link: **apt://virt-top** [apt://virt-top]

Then use the following command to call virt-top:

```
virt-top
```

and with Q can be left virt-top

Links

- **virt-manager** [https://wiki.ubuntuusers.de/virt-manager/] - GNOME and libvirt-based VM managers
- **Manpage** [https://manpages.ubuntu.com/cgi-bin/search.py?lr=lang_en&q=virsh]
- **Virtual machines remote control with Virsh** [http://www.linux-magazin.de/Ausgaben/2011/12/Virsh] - Article Linux Magazin, 12/2011
- **Managing guests with virsh** [http://docs.fedoraproject.org/de-DE/Fedora/12/html/Virtualization_Guide/chap-Virtualization_Guide-Managing_guests_with_virsh.html]
- **Description** [http://www.centos.org/docs/5/html/5.2/Virtualization/sect-Virtualization-Installing_guests-Create_a_guest_using_virt_install.html] - the most important options for virt-install
- **Hypervisors** [http://libvirt.org/drivers.html] supported by libvirt

---

This revision [https://wiki.ubuntuusers.de/virsh/a/revision/995465/] was released on the 2nd day. May 2021 18:26 by **noisefloor**.
The following keywords have been assigned to the article: **Emulation and virtualization** [https://wiki.ubuntuusers.de/wiki/tags/Emulation%20und%20Virtualisierung/], **server** [https://wiki.ubuntuusers.de/wiki/tags/Server/]