

How to detect the type of virtualization from the command line

Author : Dan Nanni

Categories : [Shell](#), [Virtualization](#)

Tagged as : [cli](#), [docker](#), [kvm](#), [lxc](#), [qemu](#), [virtualboxxen](#)

Question: I am trying to write a script in which I need to detect whether the Linux server on which the script is running is a virtual machine, a container or actual bare-metal hardware. This is because the script must behave differently depending on whether the underlying server is virtual or physical. Is there an easy way to detect (from the command line) whether I am in a virtualized environment, and if so what type of virtualization is used?

If you want to find out whether the Linux OS is running inside a virtualized environment or on a real hardware computer, there are various heuristics you can try, depending on the type of hypervisor/container used underneath. Different hypervisors and container technologies may introduce different fingerprints within their instances (e.g., processor manufacturer, special file in /proc, virtualized NIC driver name). Also the boot sequence shown by `dmesg` can reveal some clues whether the OS is virtualized.

```
[ 0.000000] Initializing cgroup subsys cpu
[ 0.000000] Initializing cgroup subsys cpuacct
[ 0.000000] Linux version 3.13.0-24-generic (buildd@panlong) (gcc version 4.8.2 (Ubuntu 4.8.2-19u
-Ubuntu SMP Thu Apr 10 19:11:08 UTC 2014 (Ubuntu 3.13.0-24.46-generic 3.13.9)
[ 0.000000] Command line: BOOT_IMAGE=/boot/vmlinuz-3.13.0-24-generic root=UUID=05088af6-6cef-4516
2ea0a ro
[ 0.000000] KERNEL supported cpus:
[ 0.000000]   Intel GenuineIntel
[ 0.000000]   AMD AuthenticAMD
[ 0.000000]   Centaur CentaurHauls
[ 0.000000] e820: BIOS-provided physical RAM map:
[ 0.000000] BIOS-e820: [mem 0x0000000000000000-0x0000000000009fbff] usable
[ 0.000000] BIOS-e820: [mem 0x0000000000009fc00-0x0000000000009ffff] reserved
[ 0.000000] BIOS-e820: [mem 0x000000000000f0000-0x000000000000ffffff] reserved
[ 0.000000] BIOS-e820: [mem 0x00000000000100000-0x0000000000007fffdfff] usable
[ 0.000000] BIOS-e820: [mem 0x000000000007ffffe000-0x000000000007fffffff] reserved
[ 0.000000] BIOS-e820: [mem 0x000000000feffc000-0x000000000fefffffff] reserved
[ 0.000000] BIOS-e820: [mem 0x000000000fffc0000-0x000000000ffffffff] reserved
[ 0.000000] NX (Execute Disable) protection: active
[ 0.000000] SMBIOS 2.4 present.
[ 0.000000] DMI: QEMU Standard PC (i440FX + PIIX, 1996), BIOS Bochs 01/01/2011
[ 0.000000] Hypervisor detected: KVM
[ 0.000000] e820: update [mem 0x000000000-0x00000fff] usable ==> reserved
[ 0.000000] e820: remove [mem 0x000a0000-0x000ffff] usable
[ 0.000000] No AGP bridge found
```

Fortunately, there are several command-line tools that make it far easier for end users to detect the type of virtualization technology used underneath Linux OS.

Method One: dmidecode

One way to detect the type of underlying virtualization is via `dmidecode` command, which is originally designed to show information about system's BIOS and hardware components. In particular, use the following command to detect virtualization:

```
$ sudo dmidecode -s system-manufacturer
```

If you are running this command on a physical server, the output will be the actual name of the hardware manufacturer (e.g., "Dell Inc."). On the other hand, if you run the command on a virtual machine, it will show the name of virtualization technology (e.g., "QEMU", "Xen", "VirtualBox", "VMware, Inc.").

Note that the `dmidecode` command does not work within containers as container-based virtualization does not create device node entries in the container's `/dev` directory which is used by `dmidecode`.

Method Two: systemd

If you use `systemd` on your Linux system, you can use a command-line tool called `systemd-detect-virt` which comes with `systemd`. This command can detect both hypervisor-based (e.g., KVM, QEMU, VMware, Xen, Oracle VM, VirtualBox, UML) and container-based (e.g., LXC, Docker, OpenVZ) virtualization.

```
$ systemd-detect-virt
```

On a physical server, the output of `systemd-detect-virt` will be "none". When run inside a virtual machine or a container, `systemd-detect-virt` will show the name of the virtualization technology used (e.g., "lxc", "kvm").

Method Three: virt-what

Another way to detect a virtualized environment within a terminal is by using `virt-what`. This command is actually a shell-script that uses various heuristics to identify the type of a virtualized environment you are in. It can detect QEMU/KVM, VMware, Hyper-V, VirtualBox, OpenVZ/Virtuozzo, Xen, LXC, IBM PowerVM, Parallels, etc.

To install `virt-what` on Debian-based system:

```
$ sudo apt-get install virt-what
```

To install `virt-what` on Red Hat-based system:

```
$ sudo yum install virt-what
```

Ask Xmodulo

Find answers to commonly asked Linux questions

<http://ask.xmodulo.com>

To determine whether you are on a physical server or a virtual server, run the command with root privilege:

```
$ sudo virt-what
```

If you are on a physical server, the command will print nothing. If you are on a virtual machine or a container, it will print the type of virtualization.