NAME

systemd-detect-virt − Detect execution in a virtualized environment

SYNOPSIS

systemd−detect−virt [OPTIONS...]

DESCRIPTION

systemd−detect−virt detects execution in a virtualized environment. It identifies the virtualization technology and can distinguish full machine virtualization from container virtualization. systemd−detect−virt exits with a return value of 0 (success) if a virtualization technology is detected, and non−zero (error) otherwise. By default, any type of virtualization is detected, and the options **−−container** and **−−vm** can be used to limit what types of virtualization are detected.

When executed without **−−quiet** will print a short identifier for the detected virtualization technology. The following technologies are currently identified:

Table 1. Known virtualization technologies (both VM, i.e. full hardware virtualization, and container, i.e. shared kernel virtualization)

Type	**ID**	**Product**
VM	*qemu*	QEMU software virtualization, without KVM
	kvm	Linux KVM kernel virtual machine, with whatever software, except Oracle Virtualbox
	zvm	s390 z/VM
	vmware	VMware Workstation or Server, and related products
	microsoft	Hyper−V, also known as Viridian or Windows Server Virtualization
	oracle	Oracle VM VirtualBox (historically marketed by innotek and Sun Microsystems), for legacy and KVM hypervisor
	xen	Xen hypervisor (only domU, not dom0)
	bochs	Bochs Emulator
	uml	User−mode Linux
	parallels	Parallels Desktop, Parallels Server
	bhyve	bhyve, FreeBSD hypervisor
	qnx	QNX hypervisor
	acrn	**ACRN hypervisor**[1]
Container	*openvz*	OpenVZ/Virtuozzo
	lxc	Linux container implementation by LXC
	lxc−libvirt	Linux container implementation by libvirt
	systemd−nspawn	systemd's minimal container implementation, see **systemd-nspawn**(1)
	docker	Docker container manager
	podman	**Podman**[2] container manager
	rkt	rkt app container runtime
	wsl	**Windows Subsystem for Linux**[3]

If multiple virtualization solutions are used, only the "innermost" is detected and identified. That means if both machine and container virtualization are used in conjunction, only the latter will be identified (unless **−−vm** is passed).

Windows Subsystem for Linux is not a Linux container, but an environment for running Linux userspace applications on top of the Windows kernel using a Linux−compatible interface. WSL is categorized as a container for practical purposes. Multiple WSL environments share the same kernel and services should generally behave like when being run in a container.

OPTIONS

The following options are understood:

−c, **−−container**
Only detects container virtualization (i.e. shared kernel virtualization).

−v, **−−vm**
Only detects hardware virtualization.

−r, **−−chroot**
Detect whether invoked in a **chroot**(2) environment. In this mode, no output is written, but the return value indicates whether the process was invoked in a **chroot()** environment or not.

−−private−users
Detect whether invoked in a user namespace. In this mode, no output is written, but the return value indicates whether the process was invoked inside of a user namespace or not. See **user_namespaces**(7) for more information.

−q, **−−quiet**
Suppress output of the virtualization technology identifier.

−−list
Output all currently known and detectable container and VM environments.

−h, **−−help**
Print a short help text and exit.

−−version
Print a short version string and exit.

EXIT STATUS

If a virtualization technology is detected, 0 is returned, a non−zero code otherwise.

SEE ALSO

systemd(1), **systemd-nspawn**(1), **chroot**(2), **namespaces**(7)

NOTES

1. ACRN hypervisor
   https://projectacrn.org

2. Podman
   https://podman.io

3. Windows Subsystem for Linux
   https://docs.microsoft.com/en-us/windows/wsl/about