

# KVM

Command	Description
virsh capabilities	shows capabilities
virsh connect qemu:///system	connect locally as root
virsh connect qemu:///session	connect locally as normal user
virsh hostname	print the hypervisor hostname
virsh sysinfo	print the hypervisor sysinfo
virsh uri	print the hypervisor conanical URI
virsh list	list active domains
virsh list --all	list all domains, either active or inactive
virsh list --autostart	list autostart domains
virsh nodeinfo	display host information
virsh autostart <domain>	domain starts automatically at boot time
virsh autostart --disable <domain>	domain does NOT start at boot time
virsh start --console <domain>	start domain and attach to the console after creation
virsh save --running <domain>	save running domain state
virsh restore domain.xml	restore domain from 'virsh save' command
virsh suspend <domain>	suspend the domain
virsh resume <domain>	restart the domain from the point that was suspended
virsh define domain.xml	define domain from xml file. It register the domain <b>but not</b> start it.
virsh create domain.xml	define domain from xml file. It register the domain <b>and</b> start it.
virsh dumpxml <domain>	print domain configuration in xml
virsh shutdown --mode { acpi   initctl   signal   agent   paravirt } <domain>	shutdown domain
virsh destroy [--graceful] <domain>	shutdown domain <b>immediately</b> .
virsh save-image-dumpxml [--security-info] domain.xml	dump saved image [security] information
virsh save-image-edit { --running   --paused } domain.xml	change the xml configuration file state.
virsh restart <domain>	restart domain
virsh net-list --all	display all network interfaces
virsh net-dumpxml <interface>	dump interface information
virsh domrename <domain> <new_domain>	Rename the domain name. Do this only when guest is shutdown
virsh domname <domain_uuid>	display domain name
virsh domid <domain>	display domain ID
virsh domuuid <domain>	display domain UUID
virsh dominfo <domain>	display information about domain
virsh vcpuinfo <domain>	display information about virtual cpus
virsh vcpus <domain>	display virtual cpus infromation for a guest
virsh setvcpus <domain> NUM	modify domain virtual cpus number
virsh setmem <domain> VALUE	modify domain memory value
virsh domiflist <domain>	display list of network interfaces for specified domain

virsh domifstat <domain> <interface>	display domain interface status
virt-install \ --name <domain> \ --memory 2048 \ --vcpus 2 \ --disk size=10 \ --cdrom /path/to/iso	create domain from iso file
virt-install \ --name <domain> \ --memory 2048 \ --vcpus 2 \ --disk /path/to/imported/machine \ <b>--import</b>	create domain from imported machine
virt-install \ --name <domain> \ --memory 2048 \ --vcpus 2 \ --disk size=10 \ --location <a href="https://path/to/machine">https://path/to/machine</a>	create domain from remote container
virt-clone \ --connect { qemu:///system   qemu:///session } \ --original <domain> \ --name <new_domain> \ --file /var/lib/libvirt/images/new_domain.qcow2	clone existing machine