KVM Installation and Requirements

KVM is a full virtualization solution for x86 processors supporting hardware virtualization (Intel VT or AMD-V). It consists of two main components: A set of Kernel modules (kvm.ko, kvm-intel.ko, andkvm-amd.ko) providing the core virtualization infrastructure and processor specific drivers and a userspace program (qemu-kvm) that provides emulation for virtual devices and control mechanisms to manage VM Guests (virtual machines). The term KVM more properly refers to the Kernel level virtualization functionality, but is in practice more commonly used to reference the userspace component.

VM Guests (virtual machines), virtual storage and networks can be managed with libvirt-based and QEMU tools. libvirt is a library that provides an API to manage VM Guests based on different virtualization solutions, among them KVM and Xen. It offers a graphical user interface as well as a command line program. The QEMU tools are KVM/QEMU specific and are only available for the command line.

Hardware Requirements

Currently, SUSE only supports KVM full virtualization on x86\_64 hosts. KVM is designed around hardware virtualization features included in AMD (AMD-V) and Intel (VT-x) CPUs. It supports virtualization features of chipsets, and PCI devices, such as an I/O Memory Mapping Unit (IOMMU) and Single Root I/O Virtualization (SR-IOV)).

You can test whether your CPU supports hardware virtualization with the following command:

egrep '(vmx|svm)' /proc/cpuinfo

If this command returns no output, your processor either does not support hardware virtualization, or this feature has been disabled in the BIOS.

The following Web site identifies processors which support hardware virtualization:<http://ark.intel.com/Products/VirtualizationTechnology> (for Intel CPUs), and<http://products.amd.com/> (for AMD CPUs).

**NOTE:** The KVM Kernel modules will not load if the CPU does not support hardware virtualization or if this feature is not enabled in the BIOS.

The general minimum hardware requirements for the VM Host Server are the same as outlined in[Section 2.2, System Requirements for Operating Linux, (↑*Deployment Guide*)](https://www.suse.com/documentation/sles11/book_sle_deployment/data/sec_x86_sysreqs.html#sec_x86_sysreqs). However, additional RAM for each virtualized guest is needed. It should at least be the same amount that is needed for a physical installation. It is also strongly recommended to have at least one processor core or hyper-thread for each running guest.

# Supported Guest Operating Systems

The following table lists guest operating systems tested and their support status offered by SUSE. All guest operating systems are supported both fully virtualized and paravirtualized, with the exception of Windows guests, which are only supported fully virtualized and OES and Netware guests, which are supported only paravirtualized. All operating systems except Netware (32-bit only) are supported in both 32 and 64-bit x86 versions.

Para-virtualized drivers (PV drivers) are listed where available.

**Para-virtualized drivers for KVM**

* virtio-net: Virtual network driver.
* virtio-blk: Virtual block device driver for paravirtualized block devices.
* virtio-balloon: Memory driver for dynamic memory allocation. Allows to dynamically change the amount of memory allocated to a guest.
* virtio-scsi: Storage interface that supports advanced SCSI hardware.
* kvm-clock: Clock synchronization driver.

**Table 1-1**Supported Guest Operating Systems on KVM with SUSE Linux Enterprise Server

|  |  |  |
| --- | --- | --- |
| ****SLES 12**** | | |
|  | *PV drivers:* | kvm-clock, virtio-net, virtio-blk, virtio-balloon, virtio-console, virtio-rng, virtio-scsi |
|  | *Support Status:* | Fully supported (L3) |
| ****SLES 11 SP3 / SP4**** | | |
|  | *PV drivers:* | kvm-clock, virtio-net, virtio-blk, virtio-balloon, virtio-console, virtio-rng, virtio-scsi (SP3 and SP4 only) |
|  | *Support Status:* | Fully supported (L3) |
| ****SLES 10 SP4**** | | |
|  | *PV drivers:* | kvm-clock, virtio-net, virtio-blk, virtio-balloon, virtio-console |
|  | *Support Status:* | Fully supported (L3) |
| ****SLES 9 SP4**** | | |
|  | *PV drivers:* | n/a |
|  | *Support Status:* | Fully supported (L3) |
|  | *Mandatory boot parameters:* | * 32 bit kernel: clock=pmtmr * 64 bit kernel: ignore\_lost\_ticks |
| ****SLED 11 SP4**** | | |
|  | *PV drivers:* | kvm-clock, virtio-net, virtio-blk, virtio-balloon, virtio-console, virtio-rng |
|  | *Support Status:* | Technology Preview (L2) |
| ****RedHat Enterprise Linux 5.11+ / RHEL 6.6+ / RHEL 7.0+**** | | |
|  | *PV drivers:* | See <http://www.redhat.com/> |
|  | *Support Status:* | Best Effort (L2) |
|  | *Note:* | Refer to the RHEL Virtualization guide for more information. |
| ****Windows 2003 SP2+ / 2008 SP2+ / 2008 R2 SP1+ / 2012+ / 2012 R2+**** | | |
|  | *PV drivers:* | virtio-net, virtio-blk, virtio-balloon; drivers from the Virtual Machine Driver Pack (<http://www.suse.com/products/vmdriverpack/>) are preferred |
|  | *Support Status:* | Fully supported (L3) |
|  | *Note:* | Host processor must have constant\_tsc CPU feature (check with the following command: grep "constant\_tsc" /proc/cpuinfo). |
| ****Windows XP SP3+ / Vista SP2+ / 7 SP1+ / 8+ / 8.1+**** | | |
|  | *PV drivers:* | virtio-net, virtio-blk, virtio-balloon; drivers from the Virtual Machine Driver Pack (<http://www.suse.com/products/vmdriverpack/>) are preferred |
|  | *Support Status:* | Best effort (L2) |
| ****OES 11 SPx**** | | |
|  | *PV drivers:* |  |
|  | *Support Status:* | Fully supported (L3) |
| ****Netware 6.5 SP8 (32b only)**** | | |
|  | *PV drivers:* |  |
|  | *Support Status:* | Fully supported (L3) |

**IMPORTANT:** Guest images created under SUSE Linux Enterprise Server 11 SP1 and newer are supported, but guest images created under a previous SUSE Linux Enterprise version are not supported.

## 1.2.1Availability of Paravirtualized Drivers[#](https://www.suse.com/documentation/sles11/book_kvm/data/sec_kvm_requires_guests.html#sec_kvm_requires_guests_virt_drivers)

To improve the performance of the guest operating system, paravirtualized drivers are provided when available. Although they are not required, it is strongly recommended to use them. The paravirtualized drivers are available as follows:

SUSE Linux Enterprise Server 11 SP1 / SP2 / SP3

included in Kernel

SUSE Linux Enterprise Server 10 SP4

included in Kernel

SUSE Linux Enterprise Server 9 SP4

not available

RedHat

available in RedHat Enterprise Linux 5.4 and newer

Windows

SUSE has developed virtio based drivers for Windows, which are available in the Virtual Machine Driver Pack (VMDP). See <http://www.suse.com/products/vmdriverpack/> for more information.

# The kvm package

The kvm package provides qemu-kvm, the program that performs the I/O emulation for the VM Guest. In addition to the qemu-kvm program, the kvm package also comes with a debug level monitoring utility (kvm\_stat), firmware components, key-mapping files, and scripts. The deprecated Windows drivers (win-virtio-drivers.iso) are no longer provided. For more information, see [Deprecated features](https://www.suse.com/documentation/sles11/book_kvm/data/app_kvm_qemu-commands.html#kvm_deprecated_features).

Originally, the kvm package also provided the KVM Kernel modules. Now, these modules are included with the Kernel and only userspace components are included in the current kvm package.

Using the libvirt-based tools is the recommended way of managing VM Guests. Interoperability with other virtualization tools has been tested and is an essential part of SUSE's support stance. All tools are provided by packages carrying the tool's name.

* libvirt: A toolkit that provides management of VM Guests, virtual networks, and storage. libvirtprovides an API, a daemon, and a shell (virsh).
* virt-manager (Virtual Machine Manager): A graphical management tool for VM Guests.
* vm-install: Define a VM Guest and install its operating system.
* virt-viewer: An X viewer client for VM Guests which supports TLS/SSL encryption of x509 certificate authentication and SASL authentication.

Support for creating and manipulating file-based virtual disk images is provided by qemu-img. qemu-img is provided by the package virt-utils.

# Installing KVM

KVM is not installed by default. To install KVM and all virtualization tools, proceed as follows:

1. Start YaST and choose *Virtualization* > *Installing Hypervisor and Tools*.
2. Select *KVM* and confirm with *Accept*.
3. Confirm the list of packages that is to be installed with *Install*.
4. Agree to set up a network bridge by clicking *Yes*. It is recommended using a bridge on a VM Host Server (virtual machine host). If you prefer to manually configure a different network setup, you can safely skip this step by clicking *No*.
5. After the setup has been finished, reboot the machine as YaST suggests. Alternatively load the required kernel modules manually and start libvirtd to avoid a reboot:
6. modprobe kvm-intel # on Intel machines only
7. modprobe kvm-amd # on AMD machines only
8. modprobe vhost-net

rclibvirtd start

**NOTE: The vhost-net Kernel Module**

The vhost-net kernel module allows for a more efficient network transport to the guest. It is automatically used by libvirt if loaded, or when using the qemu-kvm command line, by adding,vhost=on to the networking option.

# KVM Limitations

Although virtualized machines behave almost like physical machines, some limitations apply. These affect both, the VM Guest as well as the VM Host Server system.

General Limitations

The following general restrictions apply when using KVM:

Overcommits

KVM allows for both memory and disk space overcommit. It is up to the user to understand the implications of doing so. However, hard errors resulting from exceeding available resources will result in guest failures. CPU overcommit is also supported but carries performance implications.

Time Synchronization

Most guests require some additional support for accurate time keeping. Where available, kvm-clock is to be used. NTP or similar network based time keeping protocols are also highly recommended (for VM Host Server and VM Guest) to help maintain a stable time. Running NTP inside the guest is not recommended when using the kvm-clock . Refer to [Section 10.8, Clock Settings](https://www.suse.com/documentation/sles11/book_kvm/data/sec_kvm_managing_clock.html) for details.

MAC addresses

If no MAC address is specified for a NIC, a default MAC address will be assigned. This may result in network problems when more than one NIC receives the same MAC address. It is recommended to always assure a unique MAC address has been assigned for each NIC.

Live Migration

Live Migration is only possible between VM Host Servers with the same CPU features. The only supported CPU model for migration is -cpu qemu64 (default) with no additional features specified. No physical devices can be passed from host to guest. Guest storage has to be accessible from both VM Host Servers and guest definitions need to be compatible. VM Host Server and VM Guests need to have proper timekeeping installed. The use of the AHCI interface, the virtfs feature, and the -mem-path command-line option are not compatible with migration. Migration from SP3 to SP2 or SP1 hosted guests is not supported.

User Permissions

The management tools (Virtual Machine Manager, virsh, vm-install) need to authenticate withlibvirt—see [Section 8.0, Connecting and Authorizing](https://www.suse.com/documentation/sles11/book_kvm/data/cha_libvirt_connect.html) for details. In order to invoke qemu-kvm from the command line, a user has to be a member of the group kvm.

Suspending/Hibernating the VM Host Server

Suspending or hibernating the VM Host Server system while guests are running is not supported.

General Limitations

The following general restrictions apply when using KVM:

Overcommits

KVM allows for both memory and disk space overcommit. It is up to the user to understand the implications of doing so. However, hard errors resulting from exceeding available resources will result in guest failures. CPU overcommit is also supported but carries performance implications.

Time Synchronization

Most guests require some additional support for accurate time keeping. Where available, kvm-clock is to be used. NTP or similar network based time keeping protocols are also highly recommended (for VM Host Server and VM Guest) to help maintain a stable time. Running NTP inside the guest is not recommended when using the kvm-clock . Refer to [Section 10.8, Clock Settings](https://www.suse.com/documentation/sles11/book_kvm/data/sec_kvm_managing_clock.html) for details.

MAC addresses

If no MAC address is specified for a NIC, a default MAC address will be assigned. This may result in network problems when more than one NIC receives the same MAC address. It is recommended to always assure a unique MAC address has been assigned for each NIC.

Live Migration

Live Migration is only possible between VM Host Servers with the same CPU features. The only supported CPU model for migration is -cpu qemu64 (default) with no additional features specified. No physical devices can be passed from host to guest. Guest storage has to be accessible from both VM Host Servers and guest definitions need to be compatible. VM Host Server and VM Guests need to have proper timekeeping installed. The use of the AHCI interface, the virtfs feature, and the -mem-path command-line option are not compatible with migration. Migration from SP3 to SP2 or SP1 hosted guests is not supported.

User Permissions

The management tools (Virtual Machine Manager, virsh, vm-install) need to authenticate withlibvirt—see [Section 8.0, Connecting and Authorizing](https://www.suse.com/documentation/sles11/book_kvm/data/cha_libvirt_connect.html) for details. In order to invoke qemu-kvm from the command line, a user has to be a member of the group kvm.

Suspending/Hibernating the VM Host Server

Suspending or hibernating the VM Host Server system while guests are running is not supported.

Hardware Limitations

The following virtual hardware limits for guests have been tested. We ensure host and VMs install and work successfully, even when reaching the limits and there are no major performance regressions (CPU, memory, disk, network) since the last release.

|  |  |
| --- | --- |
| *Max. Guest RAM Size* | 4 TB |
| *Max. Virtual CPUs per Guest* | 256 |
| *Max. Virtual Network Devices per Guest* | 8 |
| *Max. Virtual Block Devices per Guest* | 4 emulated (IDE), 20 para-virtual (using virtio-blk) |
| *Max. Number of VM Guests per VM Host Server* | Limit is defined as the total number of virtual CPUs in all guests being no greater than 8 times the number of CPU cores in the host. |

The following hardware limits for the host server have been tested.

|  |  |
| --- | --- |
| *Max. Physical CPUs* | 4096 |
| *Max. Physical Memory* | 16 TB |

Performance Limitations

Basically, workloads designed for physical installations can be virtualized and therefore inherit the benefits of modern virtualization techniques. However, virtualization comes at the cost of a slight to moderate performance impact. You should always test your workload with the maximum anticipated CPU and I/O load to verify if it is suited for being virtualized. Although every reasonable effort is made to provide a broad virtualization solution to meet disparate needs, there will be cases where the workload itself is unsuited for KVM virtualization.

We therefore propose the following performance expectations for guests performance to be used as a guideline. The given percentage values are a comparison of performance achieved with the same workload under non-virtualized conditions. The values are rough approximations and cannot be guaranteed.

|  |  |  |  |
| --- | --- | --- | --- |
| **Category** | **Fully Virtualized** | **Paravirtualized** | **Host Pass-through** |
| *CPU, MMU* | 7% | not applicable | * 97% (Hardware Virtualization with Extended Page Tables (Intel) or Nested Page Tables (AMD) * 85% (Hardware Virtualization with shadow page tables) |
| *Network I/O (1GB LAN)* | 60% (e1000 emulated NIC) | 75% (virtio-net) | 95% |
| *Disk I/O* | 40% (IDE emulation) | 85% (virtio-blk) | 95% |
| *Graphics (non-accelerated)* | 50% (VGA or Cirrus) | not applicable | not applicable |
| *Time accuracy (worst case, using recommended settings without NTP)* | 95% - 105% (where 100% = accurate) | 100% (kvm-clock) | not applicable |

Performance Limitations

Basically, workloads designed for physical installations can be virtualized and therefore inherit the benefits of modern virtualization techniques. However, virtualization comes at the cost of a slight to moderate performance impact. You should always test your workload with the maximum anticipated CPU and I/O load to verify if it is suited for being virtualized. Although every reasonable effort is made to provide a broad virtualization solution to meet disparate needs, there will be cases where the workload itself is unsuited for KVM virtualization.

We therefore propose the following performance expectations for guests performance to be used as a guideline. The given percentage values are a comparison of performance achieved with the same workload under non-virtualized conditions. The values are rough approximations and cannot be guaranteed.

|  |  |  |  |
| --- | --- | --- | --- |
| **Category** | **Fully Virtualized** | **Paravirtualized** | **Host Pass-through** |
| *CPU, MMU* | 7% | not applicable | * 97% (Hardware Virtualization with Extended Page Tables (Intel) or Nested Page Tables (AMD) * 85% (Hardware Virtualization with shadow page tables) |
| *Network I/O (1GB LAN)* | 60% (e1000 emulated NIC) | 75% (virtio-net) | 95% |
| *Disk I/O* | 40% (IDE emulation) | 85% (virtio-blk) | 95% |
| *Graphics (non-accelerated)* | 50% (VGA or Cirrus) | not applicable | not applicable |
| *Time accuracy (worst case, using recommended settings without NTP)* | 95% - 105% (where 1 |  |  |

# KVM Support Status

The following list contains features and tools as supported by SUSE—this does not necessarily reflect the support status of the software itself. For a list of qemu-kvm command switches supported by SUSE, refer to [Section A.3, QEMU Command Line Options](https://www.suse.com/documentation/sles11/book_kvm/data/app_kvm_qemu-commands.html).

Supported Features and Tools

vm-install

Define and install VM Guests via vm-install including specifying the number of virtual processors, RAM, disk type and location, video type, keyboard mapping, NIC type, binding, MAC address, and boot method.

*Restrictions:* Currently only the raw, qcow2 and qed disk formats are supported in read and write mode. The vmdk, vpc and vhd/vhdx formats are only supported in read-only mode. NIC creation is restricted to using Realtek, e1000 or virtio NICs. Sound cards are not supported.

Virtual Machine Manager

Manage guests via Virtual Machine Manager using the following functions: autostart, start, stop, restart, pause, unpause, save, restore, clone, migrate, special key sequence insertion, guest console viewers, performance monitoring, and CPU pinning. Furthermore, static modifications of CPU, RAM, boot method, disk, NIC, mouse, display, video and host PCI and USB assignments are supported.

*Restrictions:* The following features are currently not supported: sound devices, qxl, vmvga (vmware), Xen video, pcnet, ne2k\_pci, eepro100, emulated SCSI disks, Spice graphics. Raw,qed, and qcow2 are the only supported storage formats in read and write mode. The vmdk, vpcand vhd/vhdx formats are only supported in read-only mode.

virsh

Manage guests via the command line.

Most virsh subcommands are supported, including creation, modification, and destruction of guests and all life cycle operations. Any virsh subcommands which translate to unsupported qemu-kvm command-line or monitor syntax are also unsupported. Guest XML descriptions used by virsh can be created manually, using vm-install, the Virtual Machine Manager, or external tools and scripts.

qemu-kvm

Manage guests via the command line. Although managing via Virtual Machine Manager should be the preferred option, qemu-kvm may be used for greater flexibility. See [Section A.3.1, Supported qemu-kvm Command Line Options](https://www.suse.com/documentation/sles11/book_kvm/data/app_kvm_qemu-commands.html#app_kvm_qemu-commands_supported) for a list of supported options.

*Restrictions:* See [Section A.3.2, Unsupported qemu-kvm Command Line Options](https://www.suse.com/documentation/sles11/book_kvm/data/app_kvm_qemu-commands.html#app_kvm_qemu-commands_unsupported) for a list of not supported options.

kvm\_stat

Debugging and monitoring tool.

USB Host Device Pass-through

A physical USB device may be passed from the VM Host Server to the VM Guest. Given the very wide variety of USB devices available, it is anticipated that some devices may not work properly.

PCI Pass-through

PCI Pass-through improves performance of PCI devices. It requires underlying support from the hardware (e.g. Intel VT-d extensions, or AMD IOMMU extensions). VT-d requires the Kernel parameter "intel\_iommu=on". Additionally, some host hardware will require the use of the KVM Kernel module parameter: allow\_unsafe\_assigned\_interrupts=1 (enabling this parameter has security implications).

Many PCIe cards from major vendors should be supportable. Refer to system level certifications for specific details, or contact the vendor for support statements.

Device hotplugging

Dynamically adding or removing emulated or pass-through physical devices in the VM Guest is supported.

Memory ballooning

Dynamically changing the amount of memory allocated to a guest is supported.

Sharing Folders between VM Host Server and VM Guest

Sharing folders between host and VM Guest is supported via VirtFs.

Kernel Samepage Merging (KSM)

KSM allows for automatic sharing of identical memory pages between guests to save host memory. KVM is optimized to use KSM if enabled on the VM Host Server.

Transparent Huge Pages (THP)

THP allows CPUs to address memory using pages larger than the default 4 KB. This helps reducing memory consumption and CPU cache usage. KVM is optimized to use THP (via madvise and opportunistic methods) if enabled on the VM Host Server.

KVM Security

A kvm group is created by the KVM package, which permits a non-root user to access the KVM control device file (/dev/kvm). Where possible, guests should not be run as root. Steps have been taken to enable this for libvirt as well. A setuid bridge helper has been added so that a bridged network interface can be set up without needing root privileges.

Seccomp2 based sandboxing

The VM Guest can be run in a sandboxed environment where only predetermined system calls are permitted for added protection against malicious behavior.

APIC Virtualization

Hardware APIC Virtualization, allowing the processor to directly inject interrupts into the VM Guest to achieve better performance, is supported.

VirtFS (file system pass-through)

Directories in the host file system can be shared between the host and VM Guest or guests usingvirtfs. A virtfs proxy helper is provided to enable virtfs usage when KVM is used as non-root user.

Vhost-net kernel module support

The vhost-net kernel module allows for a more efficient network transport to the VM Guest. It is automatically used by libvirt if loaded, or when using the qemu-kvm command line, by addingvhost=on to the networking option.

AHCI guest storage interface

The AHCI interface for SATA storage has been recently added. It permits much higher block I/O performance than the IDE interface, and is particularly useful for use in recent Windows OS versions.

qcow2 and qed storage formats

qcow2 and qed storage formats can now be used with live migration.

Trim and Online Disk Resizing

Trim and online disk resizing support depends on the storage format used.

Virtio SCSI

Virtio SCSI allows for passing through host SCSI block or generic SCSI devices to the VM Guest, and provides additional storage options in a virtio SCSI interface within the guest.

Macvtap / vhost-net zero-copy transmits

Zero-copy packet transmits from the VM Guest are now possible using vhost-net and macvtap changes that have been added to the latest kernels.

Disk caching modes

The default caching mode for disk images is now *writeback* due to improvements in the handling of the image format. The virtio-blk back-end now automatically switches from 'writeback' to 'writethrough' if the VM Guest virtio driver does not support flushes.

Non-Uniform Memory Access (NUMA)

NUMA machines are supported. Using numactl to pin qemu-kvm processes to specific nodes is recommended.

Supported live migration scenarios

The following host operating system combinations are fully supported for live migrating guests from one host to another: SLES 11 SP3 to SLES 11 SP4, SLES 11 SP4 to SLES 11 SP4 and SLES 11 SP4 to SLES 12. When released, live migrating from SLES 11 SP4 to SLES 12 SP1 will be also supported.

Backwards migration is not supported: SLES 12 to SLES 11 SP4 and SLES 11 SP4 to SLES 11 SP3.

All supported guest systems can be migrated.