# Virtualization on SUSE Linux Enterprise Server for NVIDIA with Benefits and Setup

Virtualization is a technology that allows a single physical server (host) to run multiple virtual machines (guests), each with its own operating system. Use it to reduce hardware costs, save power and space, and improve infrastructure flexibility and productivity. By the end of this article, you will understand the benefits of virtualization and the basic setup of a virtual machine host and guest environment.

Publication Date: 05 Nov 2026

## Contents

1 Introduction to virtualization 3

2 Installation of virtualization components 4

3 Virtualization modes 5

4 Introduction to KVM virtualization 5

5 Installation of virtualization components 7

6 Command-line tools 11

7 Converting VMware virtual machines to KVM with **`virt-v2v`** 12

8 For more information 19

9 Legal Notice 19

A GNU Free Documentation License 20

Acronyms 28

Glossary 33

# 1 Introduction to virtualization

Virtualization is a technology that provides a way for a machine (VM Host Server) to run another operating system (VM Guest) on top of the host operating system.

## 1.1 How does virtualization work?

The primary component of VM Host Server that enables virtualization is a *hypervisor*. A hypervisor is a layer of software that runs directly on VM Host Server's hardware. It controls platform resources, sharing them among multiple VM Guests and their operating systems by presenting virtualized hardware interfaces to each VM Guest.

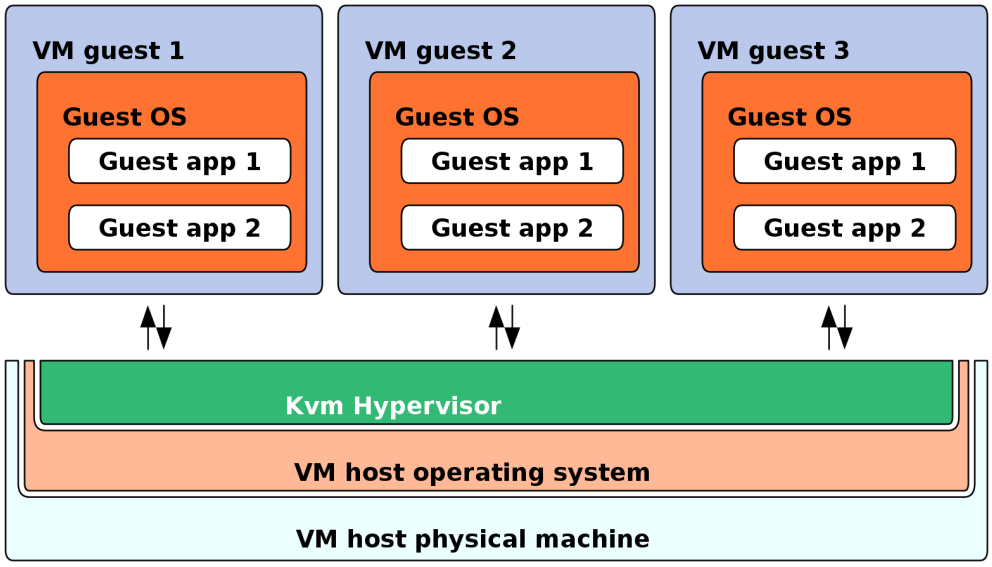



FIGURE 1: A GENERAL SCHEMA OF VIRTUALIZATION

## 1.2 Benefits of virtualization

Virtualization brings a lot of advantages while providing the same service as a hardware server.

Virtualization reduces the cost of your infrastructure. Servers are mainly used to provide a service to a customer. A virtualized operating system can provide the same service but with the following advantages:

- Less hardware: you can run several operating systems on one host, therefore all hardware maintenance is reduced.
- Less power/cooling: less hardware means you do not need to invest more in electric power, backup power, and cooling if you need more service.
- Save space: your data center space is saved because you do not need more hardware servers (fewer servers than services running).
- Less management: using a VM Guest simplifies the administration of your infrastructure.
- Agility and productivity: virtualization provides *migration* capabilities, *live migration* and *snapshots*. These features reduce downtime and bring an easy way to move your service from one place to another without any service interruption.

# 2 Installation of virtualization components

To run a virtualization server (VM Host Server) that can host multiple guest systems (VM Guests), you need to install required virtualization components on the server. These components vary depending on which virtualization technology you want to use.

You can install the virtualization tools required to run a VM Host Server either when installing the system (see the ), or from an already installed system by installing a virtualization pattern. The later option is described bellow:

```
> sudo zypper install -t pattern PATTERN_NAME
```

Replace the *PATTERN_NAME* with one of the following values:

`kvm_server`
: Installs a basic VM Host Server with the KVM and QEMU environments.

`kvm_tools`
: Installs `libvirt` tools for managing and monitoring VM Guests in the KVM environment.

# 3 Virtualization modes

Virtualization is a technology that provides a way for a machine (VM Host Server) to run another operating system (VM Guest) on top of the host operating system. There are two basic modes of hosting VM Guests on virtual machines—*full virtualization* mode and *paravirtual* mode.

**Full virtualization (FV)**

FV lets virtual machines run unmodified operating systems. It uses either Binary Translation or hardware-assisted virtualization technology, such as AMD* Virtualization or Intel* Virtualization Technology, to improve performance on processors that support it. In FV mode, VM Guest is also called the *Hardware Virtual Machine* (HVM).



## Tip

Certain guest operating systems hosted in full virtualization mode can be configured to use drivers from the SUSE Virtual Machine Drivers Pack (VMDP) instead of drivers included in the operating system. Running virtual machine drivers improves performance on guest operating systems, such as Windows Server.

**Paravirtualization (PV)**

PV normally requires that guest operating systems are modified for the virtualization environment. VM Guests running in paravirtual mode have better performance than those running under full virtualization. Operating systems currently modified to run in paravirtual mode are called *paravirtualized* operating systems and include SLE for NVIDIA.

**PV on HVM (PVHVM)**

PVHVM enhances HVM (see *Full virtualization (FV)*) with paravirtualized drivers, and handling of paravirtualized interrupts and timers.

# 4 Introduction to KVM virtualization

KVM (Kernel Virtual Machine) is a full virtualization solution that transforms the Linux kernel into a hypervisor for running multiple isolated virtual environments.

## 4.1 Basic components

KVM is a full virtualization solution for hardware architectures that support hardware virtualization.

VM Guests (virtual machines), virtual storage and virtual networks can be managed with QEMU tools directly or with the `libvirt`-based stack. The QEMU tools include **`qemu-system-`*`ARCH`***, the QEMU monitor, **`qemu-img`**, and **`qemu-nbd`**. A `libvirt`-based stack includes `libvirt` itself, along with `libvirt`-based applications such as **`virsh`**, **`virt-manager`**, **`virt-install`**, and **`virt-viewer`**.

## 4.2 KVM virtualization architecture

This full virtualization solution consists of two main components:

- A set of kernel modules (`kvm.ko`, `kvm-intel.ko`, and `kvm-amd.ko`) that provides the core virtualization infrastructure and processor-specific drivers.
- A user space program (**`qemu-system-`*`ARCH`***) 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 refer to the user space component.

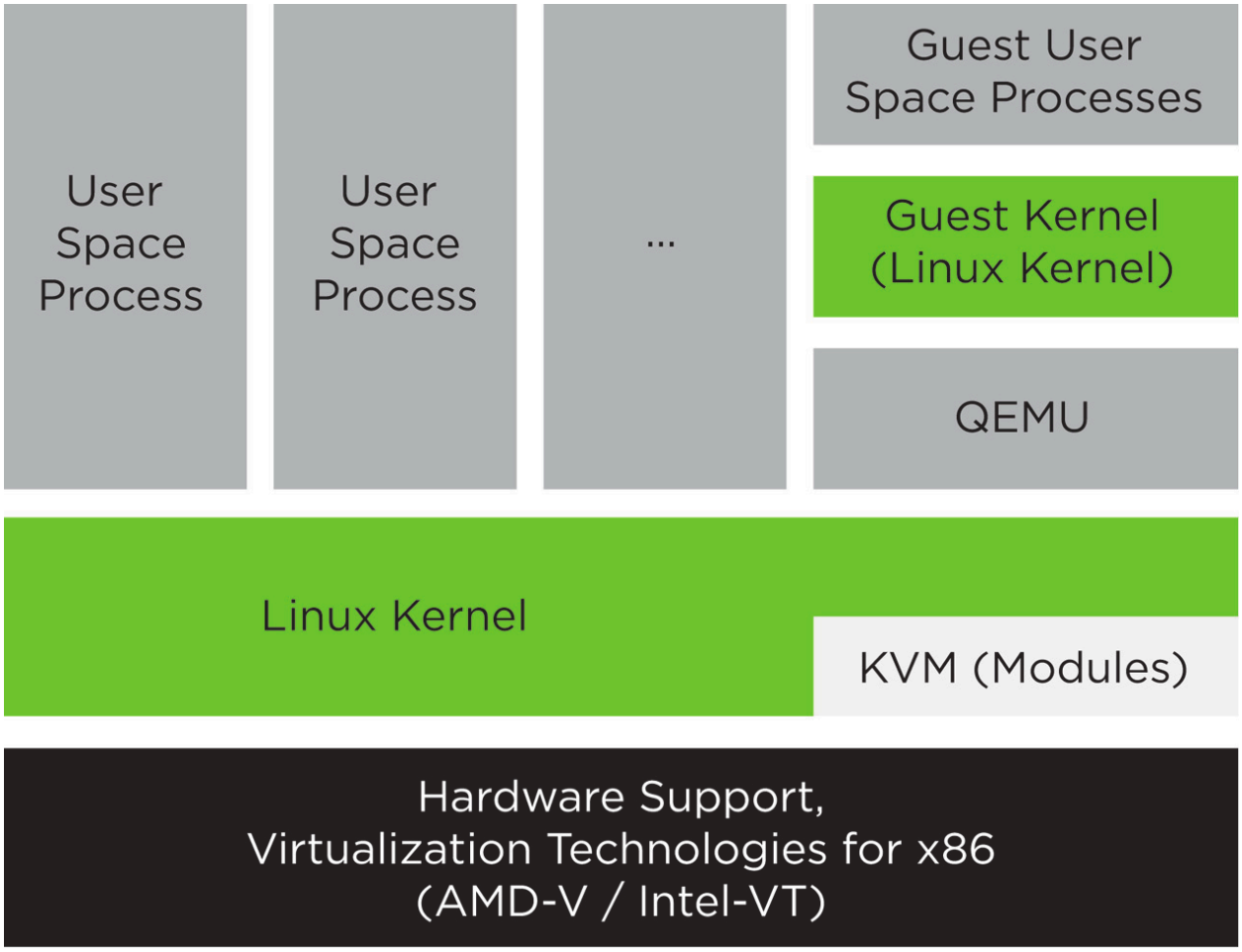



FIGURE 2: KVM VIRTUALIZATION ARCHITECTURE

# 5 Installation of virtualization components

To run a virtualization server (VM Host Server) that can host one or more guest systems (VM Guests), you need to install required virtualization components on the server.

## 5.1 Installing virtualization components

The virtualization components you need vary depending on which virtualization technology you want to use. You can install the virtualization tools required to run a VM Host Server in one of the following ways:

- By selecting a specific system role during SUSE Linux Enterprise Server for NVIDIA installation on the VM Host Server
- By installing specific installation patterns on an already installed and running SUSE Linux Enterprise Server for NVIDIA.

### 5.1.1 Specifying a system role

You can install all the tools required for virtualization during the installation of SUSE Linux Enterprise Server for NVIDIA on the VM Host Server. During the installation, you are presented with the *System Role* screen.

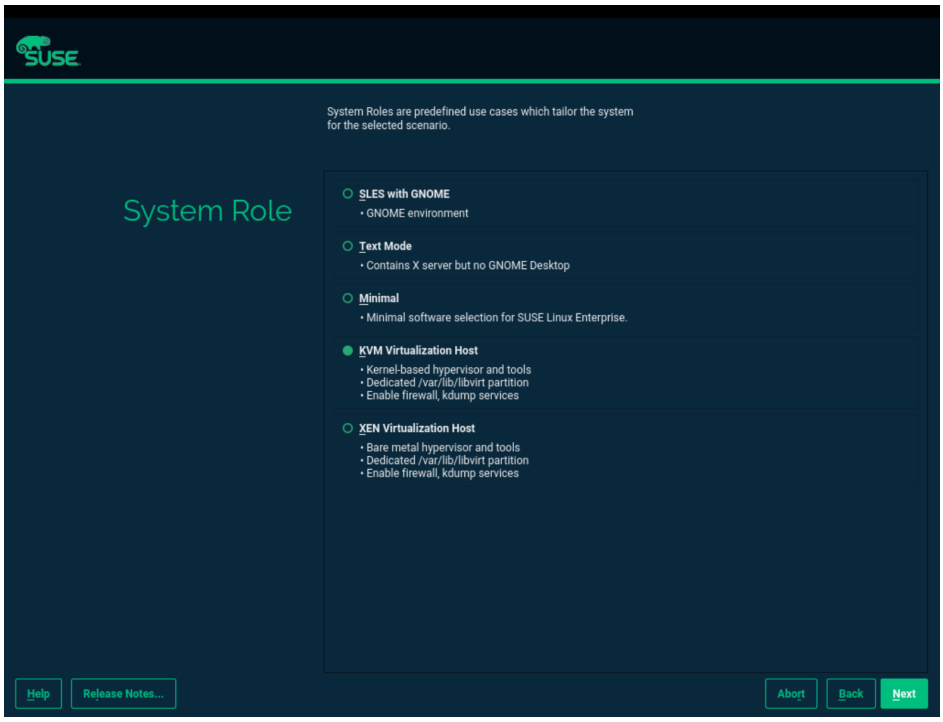



FIGURE 3: SYSTEM ROLE SCREEN

Here you can select the *KVM Virtualization Host* role. The appropriate software selection and setup is automatically performed during SUSE Linux Enterprise Server for NVIDIA installation.



### Tip

Both virtualization system roles create a dedicated `/var/lib/libvirt` partition, and enable the `firewalld` and Kdump services.

### 5.1.2 Installing specific installation patterns

Related software packages from SUSE Linux Enterprise Server for NVIDIA software repositories are organized into *installation patterns*. You can use these patterns to install specific virtualization components on an already running SUSE Linux Enterprise Server for NVIDIA. Use **`zypper`** to install them:

```
zypper install -t pattern PATTERN_NAME
```

To install the KVM environment, consider the following patterns:

`kvm_server`
: Installs basic VM Host Server with the KVM and QEMU environments.

`kvm_tools`
: Installs `libvirt` tools for managing and monitoring VM Guests in KVM environment.

## 5.2 Enable nested virtualization in KVM



### Important: Technology preview

Nested virtualization in KVM is still a technology preview. It is provided for testing purposes and is not supported.

Nested guests are KVM guests run in a KVM guest. This section describes nested guests using the following virtualization layers:

**L0**
: A bare metal host running KVM.

**L1**
: A virtual machine running on L0. Because it can run another KVM, it is called a *guest hypervisor*.

L2

A virtual machine running on L1. It is called a *nested guest*.

Nested virtualization has many advantages. You can benefit from it in the following scenarios:

- Manage your own virtual machines directly with your hypervisor of choice in cloud environments.
- Enable the live migration of hypervisors and their guest virtual machines as a single entity.



**Note**

Live migration of a nested VM Guest is not supported.

- Use it for software development and testing.

To enable nesting temporarily, remove the module and reload it with the `nested` KVM module parameter:

- For Intel CPUs, run:

```
> sudo modprobe -r kvm_intel && modprobe kvm_intel nested=1
```

- For AMD CPUs, run:

```
> sudo modprobe -r kvm_amd && modprobe kvm_amd nested=1
```

To enable nesting permanently, enable the `nested` KVM module parameter in the `/etc/modprobe.d/kvm_*.conf` file, depending on your CPU:

- For Intel CPUs, edit `/etc/modprobe.d/kvm_intel.conf` and add the following line:

```
options kvm_intel nested=1
```

- For AMD CPUs, edit `/etc/modprobe.d/kvm_amd.conf` and add the following line:

```
options kvm_amd nested=1
```

When your L0 host is capable of nesting, you can start an L1 guest in one of the following ways:

- Use the `-cpu host` QEMU command-line option.
- Add the `vmx` (for Intel CPUs) or the `svm` (for AMD CPUs) CPU feature to the `-cpu` QEMU command-line option, which enables virtualization for the virtual CPU.

### 5.2.1 VMware ESX as a guest hypervisor

If you use VMware ESX as a guest hypervisor on top of a KVM bare metal hypervisor, you may experience unstable network communication. This problem occurs especially between nested KVM guests and the KVM bare metal hypervisor or external network. The following default CPU configuration of the nested KVM guest is causing the problem:

```
<cpu mode='host-model' check='partial'/>
```

To fix it, modify the CPU configuration as follows:

```
[...]
<cpu mode='host-passthrough' check='none'>
 <cache mode='passthrough'/>
</cpu>
[...]
```

# 6 Command-line tools

`libvirt` includes several command-line utilities to manage virtual machines. A few of the command-line utilities are listed below:

`virsh` **(Package: libvirt-client)**

A command-line tool to manage VM Guests with similar functionality as the Virtual Machine Manager. Allows you to change a VM Guest's status (start, stop, pause, etc.), to set up new guests and devices, or to edit existing configurations. virsh is also useful to script VM Guest management operations.

`virt-install` **(Package: virt-install)**

A command-line tool for creating new VM Guests using the libvirt library. It supports graphical installations via VNC or SPICE protocols. Given suitable command-line arguments, virt-install can run completely unattended. This allows for easy automation of guest installs. virt-install is the default installation tool used by the Virtual Machine Manager.

`remote-viewer` **(Package: virt-viewer)**

A simple viewer of a remote desktop. It supports SPICE and VNC protocols.

`virt-clone` **(Package: virt-install)**

A tool for cloning existing virtual machine images using the libvirt hypervisor management library.

`virt-host-validate` **(Package: libvirt-client)**

A tool that validates whether the host is configured in a suitable way to run libvirt hypervisor drivers.

# 7 Converting VMware virtual machines to KVM with `virt-v2v`

**`virt-v2v`** converts virtual machines from VMware and other hypervisors to run on KVM. This guide covers the prerequisites, preparation, and conversion process, with a focus on converting Windows Server 2022 guests.

## 7.1 Overview

**`virt-v2v`** is a command-line tool for converting third-party virtual machines into a format that runs on QEMU/KVM. It is part of the **`virt-v2v`** package and uses libguestfs under the hood.

SUSE Virtualization uses **`virt-v2v`** via the Forklift migration tool to convert VMware virtual machines. The same tool can be used directly on the command line for individual conversions.



### Note: Supported source formats

**`virt-v2v`** supports conversion from VMware (VMX/VMDK), Xen, libvirt, and physical machines. This section covers VMware to KVM conversion.

## 7.2 Prerequisites

Before converting a virtual machine, ensure the following requirements are met on the KVM host:

- `virt-v2v` package installed:

```
> sudo zypper install virt-v2v
```

- `guestfs-winsupport` package installed (required for Windows guests):

```
> sudo zypper install guestfs-winsupport
```

- SUSE Linux Enterprise Virtual Machine Driver Pack ISO image. Download the latest SLE VMDP ISO from the SUSE Customer Center or the SUSE download portal.

- Source VMware VM files (VMX and VMDK) accessible on the KVM host, either locally or via NFS.
- Sufficient disk space on the target host to store the converted VM disk images.

## 7.3 Preparing the VMDP drivers

For Windows guests, **`virt-v2v`** requires virtio drivers and Windows tools to be available on the host. The SLE VMDP ISO provides a helper script that sets up the correct directory structure.

### 7.3.1 Setting up VMDP drivers

Mount the SLE VMDP ISO and run the **`virt-v2v-vmdp.sh`** script from the `v2v_scripts` directory:

1. Mount the SLE VMDP ISO:

```
> sudo mount -o loop /path/to/vmdp-VERSION.iso /mnt
```

2. Run the setup script:

```
> sudo /mnt/v2v_scripts/virt-v2v-vmdp.sh /mnt/vmdp-VERSION
```

3. Unmount the ISO:

```
> sudo umount /mnt
```

The script performs the following actions:

- Installs virtio drivers to `/usr/share/virtio-win/` in the directory layout expected by **`virt-v2v`**.
- Places `pvxsvc.exe` and other required Windows tools in `/usr/share/virt-tools/`.
- Creates placeholder files for the QEMU Guest Agent MSI and Balloon Server to suppress non-critical warnings.

### Important: Do not manually copy driver files

Do not manually extract and copy VMDP files to `/usr/share/virt-tools/` or `/usr/share/virtio-win/`. The directory structure expected by **`virt-v2v`** on SLE for NVIDIA 16.1 differs from earlier versions. Always use the **`virt-v2v-vmdp.sh`** script to ensure the correct layout.

## 7.4 Converting a VMware VM

Use the **`virt-v2v`** command to convert the VMware VM. The following example converts a Windows Server 2022 VM:

```
> sudo virt-v2v -i vmx /path/to/windows2022.vmx \
  -o local -of qcow2 -os /path/to/output/dir \
  --network vmware-network-name:target-bridge-name
```

Key command options:

`-i vmx`
: Input mode: read VMware VMX configuration file.

`-o local`
: Output mode: write disk image and XML configuration to the local filesystem. Use `-o libvirt` to define the VM directly in libvirt (requires pre-configured storage pools).

`-of qcow2`
: Output disk format. `qcow2` is recommended for space efficiency.

`-os` *`output-storage`*
: Output storage path. For `-o local`, this is a directory path. For `-o libvirt`, this is a libvirt storage pool name.

`--network` *`source:target`*
: Map a VMware network name to a target bridge or libvirt network name. If omitted, the network device mapping may require manual adjustment in the generated XML.

### 7.4.1 Using libvirt output mode

To define the converted VM directly in libvirt, use `-o libvirt`:

```
> sudo virt-v2v -i vmx /path/to/windows2022.vmx \
  -o libvirt -of qcow2 -os default \
  --network vm-network:default
```

The `-os` parameter specifies a libvirt storage pool name. Ensure the pool exists and has sufficient free space before running the conversion.

## 7.5 Post-conversion steps

After a successful conversion, follow these steps to complete the migration:

1. Start the converted VM. On first boot, the VM will install virtio drivers. The VM may reboot one or two times during this process.
2. After the VM stabilizes, power it off and change the video adapter to virtio (viogpudo) for optimal graphics performance. This can be done via **`virsh edit`** or the Virtual Machine Manager GUI.
3. Start the VM again. It will complete the driver installation and boot normally.
4. Verify that virtio drivers are active for disk, network, and graphics in the Windows Device Manager.



### Note: UEFI boot requirement

Windows Server 2022 and other modern Windows guests may require UEFI boot firmware on the target. **`virt-v2v`** detects this automatically and sets the appropriate firmware type. If the VM fails to boot, verify the firmware setting in the VM configuration.

## 7.6 Expected warnings during conversion

The following warnings are informational and can be safely ignored:

**QEMU Guest Agent MSI not found**

`virt-v2v: warning: QEMU Guest Agent MSI not found on tools ISO/directory.`

SLE VMDP does not provide a standalone QEMU Guest Agent MSI package. The guest agent is included in the VMDP `setup.exe` and is installed automatically during first boot of the converted VM.

**Balloon Server not found**

`virt-v2v: warning: Balloon Server (blnsvr.exe) not found on tools ISO/directory.`

SLE VMDP uses `pvxsvc.exe` instead of `blnsvr.exe` to provide balloon functionality. The service is installed during the first boot of the converted VM.

**Random seed could not be set**

`virt-v2v: warning: random seed could not be set for this type of guest`

This warning indicates that **virt-v2v** was unable to set a random seed for the guest. This does not affect the conversion result.

## 7.7 Troubleshooting

### 7.7.1 `Hivex.Error` during inspection

If **virt-v2v** fails with:

```
virt-v2v: error: libguestfs error: inspect_os: internal error:
inspect_os: unhandled exception thrown: Hivex.Error
```

This error occurs when libguestfs cannot parse the Windows registry hive files. Possible causes and resolutions:

- Ensure you are running the latest version of `virt-v2v` and `guestfs-winsupport`.
- Set the environment variable `LIBGUESTFS_BACKEND=direct` before running the conversion:

  ```
  > export LIBGUESTFS_BACKEND=direct
  ```

- Verify that the source VMDK disk is not corrupted by running a disk check on the VMware VM before conversion.

### 7.7.2 Virtio drivers not found

If you see the warning:

```
virt-v2v: warning: there are no virtio drivers available for this
version of Windows (10.0 x86_64 Server win2k22).  virt-v2v looks
for drivers in /usr/share/virtio-win
```

This means **virt-v2v** could not find the virtio drivers for the detected Windows version. The guest will be configured with slower emulated devices instead of virtio.

Resolution: run the **virt-v2v-vmdp.sh** script from the SLE VMDP ISO. The script installs drivers in the correct directory layout. On SLES 16, the expected driver path for Windows Server 2022 is `/usr/share/virtio-win/drivers/amd64/2k22/`.

### Important: Do not manually create symlinks as a workaround

While creating a symbolic link from `2k22` to another driver directory (e.g., `win11`) may suppress the warning, it is not a supported configuration. Always use the **`virt-v2v-vmdp.sh`** script with an up-to-date SLE VMDP ISO.

### 7.7.3 VM fails to boot after conversion (Windows Server 2019/2022/2025)

After converting Windows Server 2019, 2022, or 2025, the VM may fail to boot with an “Inaccessible Boot Device” error or a page fault blue screen.

This issue is caused by the virtio block driver installation path. On these Windows versions, the VMDP setup installs the updated virtio block driver to the driver store directory (`\windows\system32\driverstore\filerepository`) using DIRID 13, while removing the driver from `\windows\system32\drivers` (DIRID 12). Windows fails to locate the block driver on next boot.

Resolution: ensure you are using a recent SLE VMDP ISO version (2.5 or later) that includes the fix. The updated VMDP `setup.exe` copies the driver to both locations, allowing the VM to boot correctly.

If the VM fails to boot with an older VMDP version, you can manually copy the virtio block driver from the driver store to `\windows\system32\drivers` to restore boot capability.

### 7.7.4 Network mapping issues

If the `--network` mapping fails, the converted VM may not have a correctly configured network interface. To resolve:

- Verify the source network name matches the network name defined in the VMware VMX file.
- Ensure the target bridge or libvirt network exists on the host before conversion.
- If using `-o libvirt`, create a libvirt network with a matching name using **`virsh net-create`**.

### 7.7.5 Debugging **`virt-v2v`**

To enable verbose output for debugging, run **`virt-v2v`** with the `-v` option:

```
> sudo virt-v2v -v -i vmx /path/to/vm.vmx -o local -of qcow2 -os /output/dir
```

For additional libguestfs debugging, set:

```
> export LIBGUESTFS_DEBUG=1
```

```
> export LIBGUESTFS_TRACE=1
```

## 7.8 VMDP ISO contents and other uses

### 7.8.1 Installing drivers at VM creation

The VMDP ISO can be loaded as a virtual CD in a KVM VM to provide VirtIO Block or SCSI drivers during Windows installation. On the *Where do you want to install Windows* screen, select *Load driver* and follow the prompts to select the appropriate driver.

### 7.8.2 Full VMDP contents for Xen and KVM

The ISO also contains the full VMDP `setup.exe` along with expanded contents for installation flexibility. These can be used for both Xen and KVM hosted Windows VMs.

## 7.9 Complete example: Windows Server 2022

The following example demonstrates a full conversion of a Windows Server 2022 VM from VMware to KVM:

1. Mount the SLE VMDP ISO:

   ```
   > sudo mount -o loop /data/isos/vmdp/vmdp-VERSION.iso /mnt
   ```

2. Run the setup script:

   ```
   > sudo /mnt/v2v_scripts/virt-v2v-vmdp.sh /mnt/vmdp-VERSION
   ```

3. Unmount the ISO:

   ```
   > sudo umount /mnt
   ```

4. Work around the `Hivex.Error` issue by using the direct back end:

   ```
   > export LIBGUESTFS_BACKEND=direct
   ```

5. Convert the VM and define it directly in libvirt:

```
> sudo virt-v2v -i vmx /srv/vmware/windows2022/windows2022.vmx \
  -o libvirt -of qcow2 -os default \
  --network "VM Network":default
```

After conversion completes:

1. Start the VM: `virsh start windows2022`
2. Allow the VM to reboot (virtio driver installation)
3. Once stable, power off and switch video to virtio
4. Restart and verify all virtio drivers are active

# 8 For more information

For further steps in virtualization, refer to the following sources:

- Managing virtual machines with `libvirt` (https://documentation.suse.com/smart/virtualization-cloud/html/concept-manage-virtual-machines-libvirt/concept-manage-virtual-machines-libvirt.html)
- Configuring Virtual Machines with Virtual Machine Manager (https://documentation.suse.com/smart/virtualization-cloud/html/task-configure-virtual-machine-manager/task-configure-virtual-machine-manager.html)
- Assigning Host Devices to Virtual Machines (https://documentation.suse.com/smart/virtualization-cloud/html/vm-assign-pci-device/vm-assign-pci-device.html)
- Configuring a Virtual Disk Cache Mode (https://documentation.suse.com/smart/virtualization-cloud/html/virtual-disk-cache-mode-configure/virtual-disk-cache-mode-configure.html)
- Converting VMware Virtual Machines to KVM with virt-v2v (#virt-v2v-vmware-migration)

# 9 Legal Notice

Copyright© 2006– 2026 SUSE LLC and contributors. All rights reserved.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or (at your option) version 1.3; with the Invariant Section being this copyright notice and license. A copy of the license version 1.2 is included in the section entitled “GNU Free Documentation License”.

For SUSE trademarks, see https://www.suse.com/company/legal/. All other third-party trademarks are the property of their respective owners. Trademark symbols (®, ™ etc.) denote trademarks of SUSE and its affiliates. Asterisks (*) denote third-party trademarks.

All information found in this book has been compiled with utmost attention to detail. However, this does not guarantee complete accuracy. Neither SUSE LLC, its affiliates, the authors, nor the translators shall be held liable for possible errors or the consequences thereof.

# A GNU Free Documentation License

Copyright (C) 2000, 2001, 2002 Free Software Foundation, Inc. 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA. Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

### 0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or non-commercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

## 1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image for-

mats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

## 2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or non-commercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

## 3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally

prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

## 4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

## 5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

## 6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

## 7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

## 8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

## 9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

## 10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See https://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

## ADDENDUM: How to use this License for your documents

```
Copyright (c) YEAR YOUR NAME.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.2
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.
A copy of the license is included in the section entitled “GNU
Free Documentation License”.
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with...Texts.” line with this:

```
with the Invariant Sections being LIST THEIR TITLES, with the
Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

# Acronyms

## ACPI

**Advanced Configuration and Power Interface**

An open standard that provides uniform device configuration and power management capabilities directly to the operating system.

## AER

**Advanced Error Reporting**

A capability within the PCI Express specification that enables advanced reporting, logging and recovery from PCI bus errors.

## APIC

**Advanced Programmable Interrupt Controller**

An architecture of interrupt controllers designed to efficiently manage and route hardware interrupts across multiprocessing systems.

## BDF

**Bus:Device:Function**

A standard notation used to uniquely address and identify individual physical or virtual devices on a PCI/PCIe bus.

## CG

**Control Groups (cgroups)**

A Linux kernel feature used to limit, account for, isolate and prioritize the resource usage (such as CPU, memory and disk I/O) of collections of processes.

## EDF

**Earliest Deadline First**

A dynamic priority scheduling algorithm that provides intuitive, real-time CPU resource sharing based on strict time guarantees.

## EPT

**Extended Page Tables**

Performance in a virtualized environment is close to that in a native environment. However, virtualization does create overhead. It comes from the virtualization of the CPU, the *MMU*, and the I/O devices. In some recent x86 processors, AMD and Intel have begun to provide hardware extensions to help bridge

this performance gap. In 2006, both vendors introduced their first-generation hardware support for x86 virtualization with AMD-Virtualization (AMD-V) and Intel® VT-x technologies. Intel introduced its second generation of hardware support that incorporates MMU-virtualization, called Extended Page Tables (EPT). EPT-enabled systems can improve performance compared to using shadow paging for *MMU* virtualization. EPT increases memory access latencies for a few workloads. This cost can be reduced by effectively using large pages in the guest and the hypervisor.

### HVM

**Hardware Virtual Machine**

A fully virtualized operating system instance running on hardware that natively supports virtualization extensions (such as Intel VT-x or AMD-V).

### IOMMU

**Input/Output Memory Management Unit**

IOMMU (AMD* technology) is a memory management unit (*MMU*) that connects a direct memory access-capable (DMA-capable) I/O bus to the main memory.

### KSM

**Kernel Same-page Merging**

A Linux kernel feature that 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.

### MMU

**Memory Management Unit**

A hardware component responsible for handling accesses to memory requested by the CPU. Its functions include translation of virtual addresses to physical addresses (that is, virtual memory management), memory protection, cache control, bus arbitration and, in simpler computer architectures (especially 8-bit systems), bank switching.

### PAE

**Physical Address Extension**

A memory management feature that allows 32-bit x86 operating systems to address more than 4 GB of physical memory by expanding page table entries to 64 bits.

### PCID

**Process-Context Identifiers**

A processor facility that caches information for multiple linear-address spaces so that the processor may retain cached information when software switches to a different linear address space. The INVPCID instruction is used for fine-grained *TLB* flush, which benefits the kernel.

### PCIe

**Peripheral Component Interconnect Express**

A high-speed serial expansion bus standard designed to replace older PCI, PCI-X and AGP bus standards. PCIe has numerous improvements, including a higher maximum system bus throughput, a lower I/O pin count and a smaller physical footprint. Moreover, it also has a more detailed error detection and reporting mechanism (*AER*) and a native hotplug functionality. It is also backward compatible with PCI.

### PSE / PSE36

**Page Size Extension / Page Size Extension 36-Bit**

Hardware features in x86 processors that allow for pages larger than the traditional 4 KiB size. PSE-36 capability offers 4 more bits, in addition to the normal 10 bits, which are used inside a page directory entry pointing to a large page. This allows a large page to be located in a 36-bit address space.

### PT

**Page Table**

The data structure used by a virtual memory system in a computer operating system to store the mapping between virtual addresses and physical addresses. Virtual addresses are those unique to the accessing process. Physical addresses are those unique to the hardware (RAM).

### QXL

**QXL Display Device**

QXL is a cirrus VGA framebuffer (8M) driver for virtualized environment.

### RVI / NPT

**Rapid Virtualization Indexing / Nested Page Tables**

AMD's second-generation hardware-assisted virtualization technology for the processor memory management unit (*MMU*).

### SATA

**Serial ATA**

A computer bus interface that connects host bus adapters to mass storage devices, such as hard disks and optical drives.

### Seccomp2-based sandboxing

**Secure Computing Mode sandboxing**

A sandboxed environment where only predetermined system calls are permitted for added protection against malicious behavior.

### SPICE

**Simple Protocol for Independent Computing Environments**

An open-source remote display protocol designed for virtual environments to deliver high-quality graphics and audio streaming over networks.

### TCG

**Tiny Code Generator**

The core component of the QEMU emulator that acts as a just-in-time compiler, translating target CPU instructions into host system architecture instructions.

### THP

**Transparent Huge Pages**

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

### TLB

**Translation Lookaside Buffer**

A hardware cache that memory management hardware uses to improve virtual address translation speed. All current desktop, notebook and server processors use a TLB to map virtual and physical address spaces, and it is nearly always present in any hardware that uses virtual memory.

### VCPU

**Virtual Central Processing Unit**

A software-defined scheduling entity managed by a hypervisor that represents a single physical CPU core allocated to a virtual machine.

### VDI

**Virtual Desktop Infrastructure**

An IT architecture that hosts user desktop environments within virtual machines running on centralized data center servers.

### VFIO

**Virtual Function I/O**

A Linux kernel framework that provides secure, direct user-space access to PCI devices.

### VHS

**Virtualization Host Server**

A physical server hardware platform explicitly configured with a hypervisor to host, manage and run multiple virtual machines.

### VMCS

**Virtual Machine Control Structure**

VMX non-root operation and VMX transitions are controlled by a data structure called a virtual-machine control structure (VMCS). Access to the VMCS is managed through a component of processor state called the VMCS pointer (one per logical processor). The value of the VMCS pointer is the 64-bit address of the VMCS. The VMCS pointer is read and written using the instructions VMPTRST and VMPTRLD. The *VMM* configures a VMCS using the VMREAD, VMWRITE, and VMCLEAR instructions. A *VMM* could use a different VMCS for each virtual machine that it supports. For a virtual machine with multiple logical processors (virtual processors), the *VMM* could use a different VMCS for each virtual processor.

### VMDq

**Virtual Machine Device Queues**

Multi-queue network adapters that support multiple VMs at the hardware level, having separate packet queues associated with the different hosted VMs (by means of the IP addresses of the VMs).

### VMM

**Virtual Machine Monitor (Hypervisor)**

Software, firmware or hardware that creates, runs and manages virtual machines, mediating access to the underlying physical resources.

### VMX

**Virtual Machine Extensions**

Intel’s hardware-assisted virtualization technology architecture added to x86 processors to support native hypervisor execution.

### VMX Root / Non-Root Operation

Processor execution modes introduced by *VMX*. The *VMM* runs in VMX root operation, while guest software runs in VMX non-root operation. Transitions between the two modes are called VMX transitions.

### VPID

**Virtual Processor Identifiers**

New support for software control of *TLB* (VPID improves *TLB* performance with small *VMM* development effort).

### VT-d

**Virtualization Technology for Directed I/O**

Intel's (https://software.intel.com/en-us/articles/intel-virtualization-technology-for-directed-io-vt-d-enhancing-intel-platforms-for-efficient-virtualization-of-io-devices) hardware implementation of an *IOMMU*, providing device isolation and direct memory access (DMA) translation for virtualized environments.

### vTPM

**Virtual Trusted Platform Module**

Component to establish end-to-end integrity for guests via Trusted Computing.

# Glossary

# General

### Create Virtual Machine Wizard

Virtual Machine Manager provides a graphical interface to guide you through the steps to create virtual machines. It can also be run in text mode by entering **`virt-install`** at a command prompt in the host environment.

### Hardware-assisted

Intel* and AMD* provide virtualization hardware-assisted technology. This reduces the frequency of VM IN/OUT (fewer VM traps), because software is a major source of overhead, and increases the efficiency (the execution is done by the hardware). Moreover, this reduces the memory footprint, provides better resource control, and allows secure assignment of specific I/O devices.

### Host Environment

The desktop or command line environment that allows interaction with the host computer's environment. It provides a command line environment and can also include a graphical desktop, such as GNOME or IceWM. The host environment runs as a special type of virtual machine that has privileges to control and manage other virtual machines.

### Hypervisor

The software that coordinates the low-level interaction between virtual machines and the underlying physical computer hardware.

### Paravirtualized Frame Buffer

The video output device that drives a video display from a memory buffer containing a complete frame of data for virtual machine displays running in paravirtual mode.

### VHS

A Virtualization Host Server (VHS) is the physical computer running SUSE virtualization platform software. The virtualization environment consists of the hypervisor, the host environment, virtual machines and associated tools, commands and configuration files. Other commonly used terms include host, Host Computer, Host Machine (HM), Virtual Server (VS), Virtual Machine Host (VMH) and VM Host Server (VHS).

### VirtFS

VirtFS is a new paravirtualized file system interface designed for improving pass-through technologies in the KVM environment. It is based on the VirtIO framework.

### Virtual Machine

A software-defined computer environment (VM) capable of hosting an independent guest operating system and its associated applications. Also commonly referred to as a VM Guest.

### Virtual Machine Manager

A software program that provides a graphical user interface for creating and managing virtual machines.

### Virtualized

The status or condition of a guest operating system or application executing within an isolated virtual machine environment.

## CPU

### CPU capping

The resource restriction process that limits a virtual CPU's allocation window to a defined percentage (from 1 to 100 percent) of available physical CPU capacity.

### CPU hotplugging

CPU hotplugging is used to describe the functions of replacing/adding/removing a CPU without shutting down the system.

### CPU over-commitment

Virtual CPU over-commitment is the ability to assign more virtual CPUs to VMs than the actual number of physical CPUs present in the physical system. This procedure does not increase the overall performance of the system, but may be useful for testing purposes.

### CPU pinning

Also referred to as processor affinity; CPU pinning enables the binding and unbinding of a process or a thread to a central processing unit (CPU) or a range of CPUs.

## Network

### Bridged Networking

A type of network connection that lets a virtual machine be identified on an external network as a unique identity that is separate from and unrelated to its host computer.

### Empty Bridge

A type of network bridge that has no physical network device or virtual network device provided by the host. This lets virtual machines communicate with other virtual machines on the same host, but not with the host or on an external network.

### External Network

The network outside a host's internal network environment.

### Internal Network

A type of network configuration that restricts virtual machines to their host environment.

### Local Bridge

A type of network bridge that has a virtual network device but no physical network device provided by the host. This lets virtual machines communicate with the host and other virtual machines on the host. Virtual machines can communicate on an external network through the host.

### Network Address Translation (NAT)

A type of network connection that lets a virtual machine use the IP address and MAC address of the host.

### No Host Bridge

A type of network bridge that has a physical network device but no virtual network device provided by the host. This lets virtual machines communicate on an external network but not with the host. This lets you separate virtual machine network communications from the host environment.

### Traditional Bridge

A type of network bridge that has both a physical network device and a virtual network device provided by the host.

## Storage

### AHCI

The Advanced Host Controller Interface (AHCI) is a technical standard defined by Intel* that specifies the operation of Serial ATA (SATA) host bus adapters in a non-implementation-specific manner.

### Block Device

Data storage components, such as CD-ROM drives or disk drives, that move data in the form of blocks. Partitions and volumes are also considered block devices.

### File-Backed Virtual Disk

A virtual disk based on a file, also called a disk image file.

### Raw Disk

A method of accessing data on a disk at the individual byte level instead of through its file system.

### Sparse image file

A disk image file that does not reserve its entire amount of disk space but expands as data is written to it.

### xvda

The drive designation given to the first virtual disk on a paravirtual machine.