

第 8 章 Qemu/KVM 虚拟机

Qemu (Qemu 模拟器的简称) 是一个开源的虚拟机管理软件，主要功能是模拟物理计算机。在运行 Qemu 的主机看来，Qemu 就是一个普通的用户进程，将主机拥有的硬盘分区、文件、网卡等本地资源虚拟成物理硬件设备并映射给模拟计算机使用。

模拟计算机的操作系统访问这些虚拟硬件时，就好像在访问真正的物理硬件设备一样。例如，当你设置 Qemu 参数向模拟计算机映射一个 ISO 镜像时，模拟计算机的操作系统就会看到一个插在 CD 驱动器里的 CDROM 光盘。

Qemu 能够模拟包括从 ARM 到 sparc 在内的一大批硬件设备，但 Proxmox VE 仅仅使用了其中的 32 位和 64 位 PC 平台模拟硬件，而这也是当前绝大部分服务器所使用的硬件环境。此外，借助 CPU 的虚拟化扩展功能，Qemu 模拟相同架构硬件环境的速度可以被大大提高，虚拟 PC 硬件也是当前 Qemu 支持的运行速度最快的虚拟硬件环境。

➤ 注意

后续章节你可能会看到 KVM (Kernel-based Virtual Machine) 一词。这是指 Qemu 借助 Linux 的 kvm 内核模块在 CPU 虚拟化扩展的支持下运行。在 Proxmox VE 里，Qemu 和 KVM 这两个词完全可以互换使用，因为 Qemu 总是尝试使用 kvm 模块。

为方便访问块存储设备和 PCI 硬件，在 Proxmox VE 里 Qemu 进程总是以 root 权限运行。

8.1 虚拟化硬件和半虚拟化硬件

Qemu 模拟的 PC 硬件设备包括主板、网卡控制器、scsi 控制器、ide 控制器、sata 控制器、串口等 (完整列表参见 man kvm(1)手册)，这些都是以软件模拟方式实现的虚拟化硬件。换句话说，这些虚拟化硬件都是和对应硬件设备完全相当的软件，如果客

户机操作系统安装了对应的驱动程序，客户机就可以像驱动真实物理硬件一样驱动这些虚拟化硬件。这样，Qemu 就可以直接运行客户机而无需修改客户机操作系统。

但这种方式的缺点就是性能损耗较大，因为 CPU 必须耗费大量计算能力才能以软件方式模拟硬件操作。为提高性能，可以 Qemu 还提供有半虚拟化硬件，这时客户机操作系统会感知到 Qemu 环境的存在，并直接和虚拟机管理器配合工作。

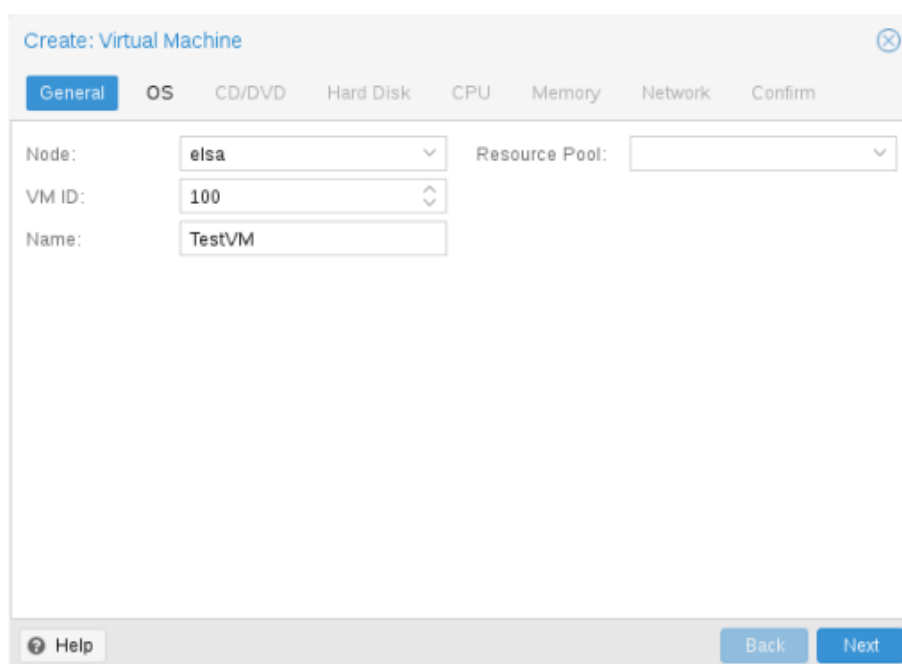
Qemu 的半虚拟化硬件采用了 virtio 标准，并以 virtio 半虚拟化硬件形式实现，具体包括半虚拟化硬盘控制器，半虚拟化网卡，半虚拟化串口，半虚拟化 SCSI 控制器等。

鉴于其所提供的高性能，我们强烈推荐优先使用 virtio 半虚拟硬件。在使用 bonnie++(8)进行的连续写测试中，virtio 半虚拟磁盘控制器的性能是模拟 IDE 控制器的 2 倍。而在基于 iperf 的测试中，virtio 半虚拟网卡的性能是模拟 Intel E1000 虚拟网卡的 3 倍。

8.2 虚拟机配置

一般来说，Proxmox VE 默认提供的虚拟机硬件配置就是最佳选择。当你确实需要改变 Proxmox VE 默认的虚拟机配置时，确保你确实清楚修改的原因及后果，否则可能会导致性能下降或者数据丢失风险。

8.2.1 通用配置



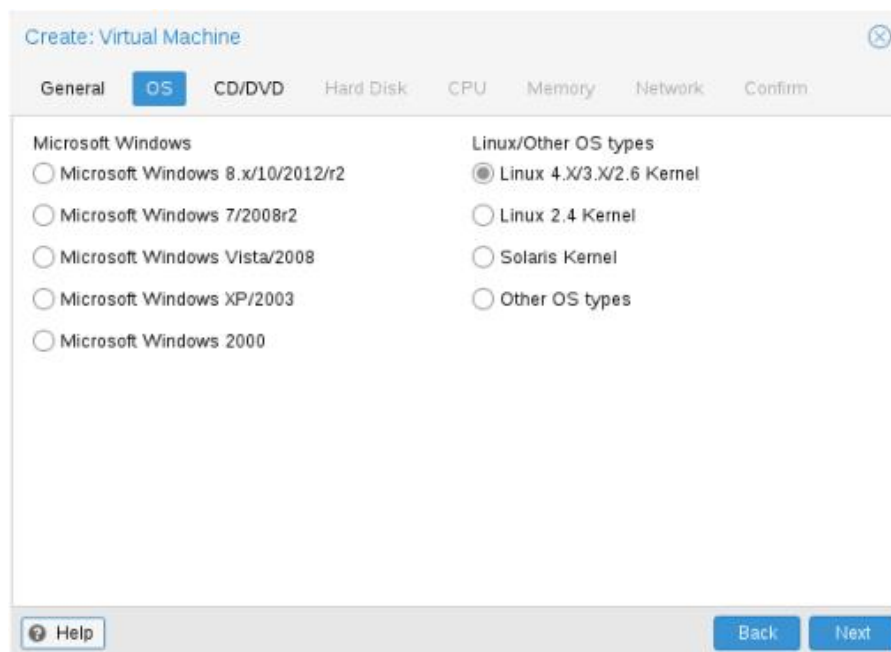
The screenshot shows the 'Create: Virtual Machine' dialog box in Proxmox VE, with the 'General' tab selected. The dialog has a title bar with a close button. Below the title bar is a tabbed interface with tabs for 'General', 'OS', 'CD/DVD', 'Hard Disk', 'CPU', 'Memory', 'Network', and 'Confirm'. The 'General' tab is active, showing fields for 'Node:' (set to 'elisa'), 'VM ID:' (set to '100'), 'Name:' (set to 'TestVM'), and 'Resource Pool:' (empty). At the bottom, there is a 'Help' button with a question mark icon, and 'Back' and 'Next' buttons.

Field	Value
Node:	elisa
VM ID:	100
Name:	TestVM
Resource Pool:	

虚拟机通用配置包括：

- 节点：虚拟机所处的物理服务器名。
- VM ID：Proxmox VE 用于标识虚拟机的一个唯一编号。
- 名称：虚拟机名称，用于描述虚拟机的字符串。
- 资源池：虚拟机所处的逻辑组。

8.2.2 操作系统配置



在创建虚拟机时，设置合适的操作系统版本能够帮助 Proxmox VE 优化虚拟机底层配置。例如，Windows 操作系统将期望 BIOS 时钟基于本地时间，而 Unix 类操作系统将期望 BIOS 时钟使用 UTC 时间。

8.2.3 硬盘

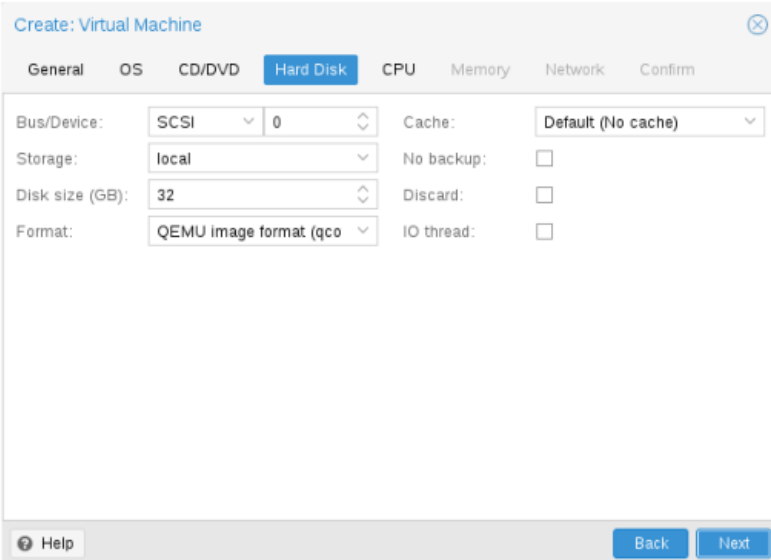
Qemu 能够模拟多种存储控制器：

- IDE 控制器，最早可追溯到 1984 年的 PC/AT 硬盘控制器。尽管后来又出现了更多更新的控制器设计，但基本上你能想到的每种操作系统都会支持 IDE 控制器。当你的虚拟机使用 2003 年以前开发的操作系统时，使用 IDE 控制器将是最佳选择。该控制器上最多可挂载 4 个设备。

- SATA 控制器出现于 2003 年，采用了更为现代化的设计，不仅提供了更高的数据传输速率，并且支持挂载更多的设备。该控制器上最多可挂载 6 个设备。
- SCSI 控制器设计于 1985 年，通常用于服务器级硬件，最多可挂载 14 个设备。默认情况下 Proxmox VE 模拟的 SCSI 控制器型号为 LSI 53C895A。

如果你想追求更高的虚拟硬盘性能，可以选择使用 virtio 类型的 SCSI 控制器。事实上，Proxmox VE 4.3 开始将该类型 SCSI 控制器用于 Linux 虚拟机的默认配置。Linux 于 2012 年开始支持该控制器，而 FreeBSD 则于 2014 年开始支持。对于 Windows 类操作系统，你需要在安装操作系统时使用专门的驱动光盘安装驱动程序后才可以使

- Virtio 控制器，为区别于 virtio SCSI 控制器，也称为 virtio-blk 控制器，是一种老式的半虚拟化控制器，后来被更新的 Virtio SCSI 控制器取代。



The image shows a 'Create: Virtual Machine' window with several tabs: General, OS, CD/DVD, Hard Disk (selected), CPU, Memory, Network, and Confirm. The 'Hard Disk' tab contains the following configuration options:

- Bus/Device: SCSI (dropdown), 0 (spin box)
- Storage: local (dropdown)
- Disk size (GB): 32 (spin box)
- Format: QEMU image format (qcow) (dropdown)
- Cache: Default (No cache) (dropdown)
- No backup: ☐
- Discard: ☐
- IO thread: ☐

At the bottom of the window, there are buttons for 'Help', 'Back', and 'Next'.

以上每种控制器都支持同时挂载多个虚拟硬盘设备，虚拟硬盘可以基于一个文件，也可以基于某种存储服务提供的块存储设备。而所选择的存储服务类型决定了虚拟硬盘镜像能采用的数据格式。块存储服务（LVM，ZFS，Ceph）上只能保存 raw 格式虚拟硬盘，文件系统存储服务（Ext4，NFS，GlusterFS）则允许你选择使用 raw 格式或 QEMU 镜像格式。

- QEMU 镜像格式是一种基于“写时复制”的虚拟硬盘格式，支持虚拟硬盘快照和薄模式存储。
- Raw 格式硬盘镜像是一种逐个 bit 存储数据的硬盘镜像格式，具体和你在 Linux 上用 dd 命令创建的镜像格式很像。这种镜像格式本身不具有创建快照或薄模式存储的功能，而需要下层存储服务支持才可以实现这些功能。但是其速度比 QEMU 镜像格式快 10%。

- VMware 镜像格式仅供用于从其他类型虚拟机系统导入/导出硬盘镜像时使用。

虚拟硬盘的 Cache 模式设置会影响 Proxmox VE 主机系统向虚拟机操作系统返回数据块写操作完成通知的时机。设置为 No cache 是指在所有数据块都已写入物理存储设备写队列后，再向虚拟机发出写操作完成通知，而忽略主机页缓存机制。该方式将能较好地平衡数据安全性和写入性能。

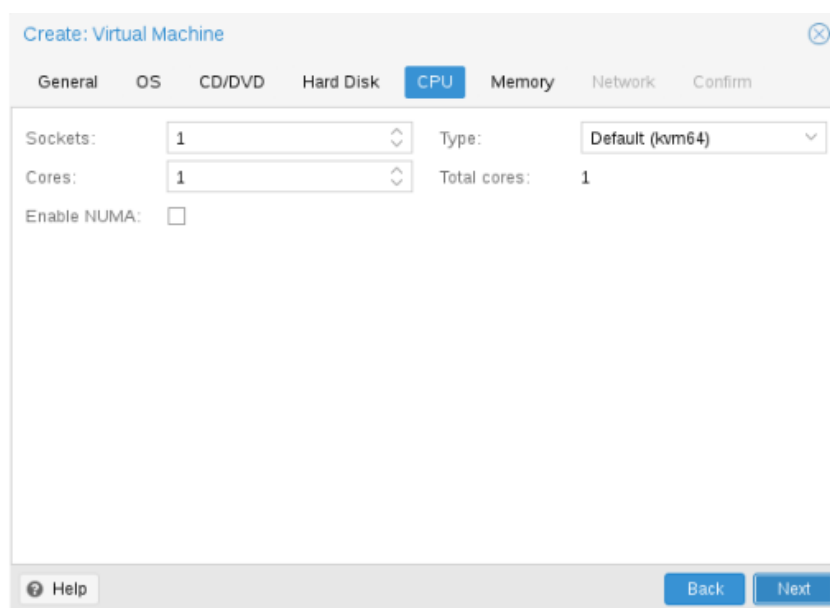
如果你希望在备份某个虚拟机时指定 Proxmox VE 备份管理器跳过某个虚拟硬盘，可以在该虚拟硬盘上启用 No backup 配置。

如果存储服务支持薄模式存储（参见存储服务一章），并且虚拟机配置了 SCSI 控制器，你可以在 SCSI 所连接的虚拟硬盘上启用 Discard 配置。当虚拟机不再使用相关硬盘，并将相关硬盘标记为 unused 之后，SCSI 控制器能够自动通知底层存储服务释放相关硬盘占用的空间。

IO Thread

仅在启用 Virtio 控制器或 Virtio SCSI 控制器时才可以启用 IO Thread 配置。启用后 IO Thread 后，Qemu 将为每一个虚拟硬盘分配一个读写线程，于之前所有虚拟硬盘共享一个线程相比，这将能够大大提高多硬盘虚拟机的性能。注意，IO Thread 配置并不能提高虚拟机备份的速度。

8.2.4 CPU



The screenshot shows the 'Create: Virtual Machine' dialog box with the 'CPU' tab selected. The dialog has tabs for General, OS, CD/DVD, Hard Disk, CPU, Memory, Network, and Confirm. The CPU tab contains the following settings:

Parameter	Value
Sockets	1
Cores	1
Enable NUMA	<input type="checkbox"/>
Type	Default (kvm64)
Total cores	1

At the bottom of the dialog, there is a 'Help' button with a question mark icon, and 'Back' and 'Next' buttons.

CPU Socket 指 PC 主板上的 CPU 芯片插槽。每个 CPU 可以有一个或多个核心（core），每个核心都是一个独立的处理单元。为虚拟机配置 1 个 4 核心虚拟 CPU 和配置 2 个 2 核心 CPU 在性能上区别不大。但某些软件是基于 Socket 授权，这时先按照软件授权设置 Socket 数量，再配置较多核心数量就是改善虚拟机性能的一种有效方法了。

通常增加虚拟机的虚拟 CPU 数量都可以改善性能，但最终改善程度还依赖于虚拟机对 CPU 的使用方式。每增加 1 个虚拟 CPU，Qemu 都会在 Proxmox VE 主机上增加一个处理线程，从而改善多线程应用的性能。如果你不确定虚拟机的具体负载，可以先为虚拟机配置 2 个虚拟 CPU，通常情况下这是比较安全的配置方法。

✓ 注意

在 Proxmox VE 主机上可以为多个虚拟机分配总数超过物理 CPU 核心数量的虚拟 CPU（比如在 8 核心主机上同时运行 4 台 4 核心的虚拟机）。Proxmox VE 主机会自动平衡各物理 CPU 核心上的 Qemu 处理线程数量，其效果就和调度一般的多线程应用程序一样。但是，Proxmox VE 不允许为一台虚拟机分配超过物理 CPU 核心数的虚拟 CPU，这只会白白增加不必要的调度数量，并导致性能低下。

Qemu 可以模拟包括从 486 到最新 Xeon 处理器在内的多种 CPU 硬件。模拟更新的 CPU 意味着模拟更多功能特性，比如硬件 3D 渲染，随机数生成器，内存保护等等。通常，你应该选择和主机 CPU 最接近的虚拟机 CPU 类型，这可以让你的虚拟机访问使用主机 CPU 的功能特性（也称为 CPU flags），你也可以将 CPU 类型设置为 host，这样虚拟机的虚拟 CPU 就和主机物理 CPU 完全一致。

这种配置方法最大的问题在于，如果你需要将一个虚拟机在线迁移到另一台物理服务器，虚拟机可能会因为两台物理服务器的 CPU 类型不同而崩溃。如果 CPU flag 不一致，Qemu 进程会直接停止运行。为避免该问题，Qemu 专门提供了一种名为 kvm64 的虚拟 CPU，这也是 Proxmox VE 默认使用的 CPU。大致上 kvm64 是一种类似于 Pentium 4 的虚拟 CPU，具有较少的 CPU flag，但具有最好的兼容性。

简而言之，如果你需要确保虚拟机的在线迁移能力，最好使用默认的 kvm64 虚拟 CPU。如果不在乎在线迁移，可以设置虚拟 CPU 类型为 host，以获得最好的性能。

此外还可以选择在虚拟机上启用 NUMA 架构模拟功能。NUMA 架构的基本设计是，抛弃了以往多个内核共同使用一个大内存池的设计，而将内存按照 Socket 分配给每个 CPU 插槽。NUMA 能有效解决共用一个大内存池时的内存总线瓶颈问题，大大改善系统性能。如果你的物理服务器支持 NUMA 架构，我们推荐启用该配置，从而更合理地

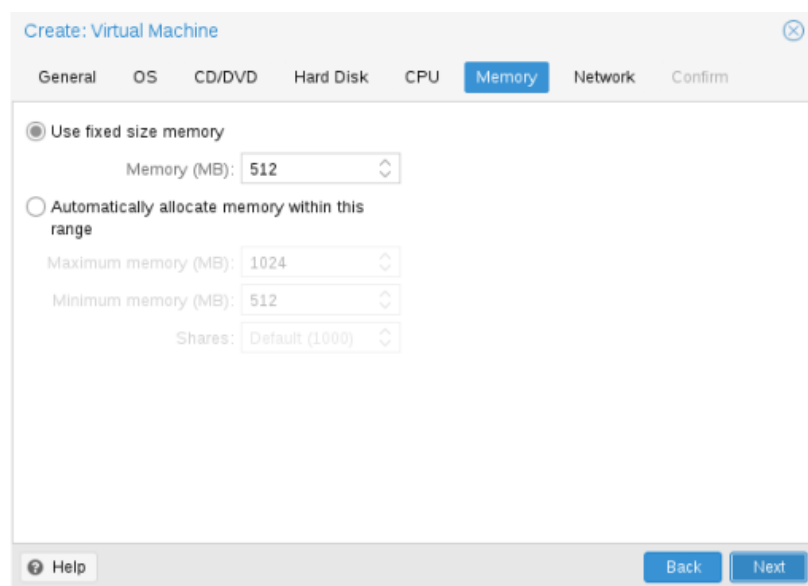
在物理服务器上分配虚拟机工作负载。此外，如果要使用虚拟机的 CPU 和内存热插拔，也需要启用该项配置。

如果启用了 NUMA，建议为虚拟机分配和物理服务器一致的 Socket 数量。

8.2.5 内存

为虚拟机分配内存时，你可以选择分配固定容量内存或让 Proxmox VE 根据主机内存使用情况动态分配内存。

分配固定容量内存



Create: Virtual Machine

General OS CD/DVD Hard Disk CPU **Memory** Network Confirm

☒ Use fixed size memory

Memory (MB): 512

☐ Automatically allocate memory within this range

Maximum memory (MB): 1024

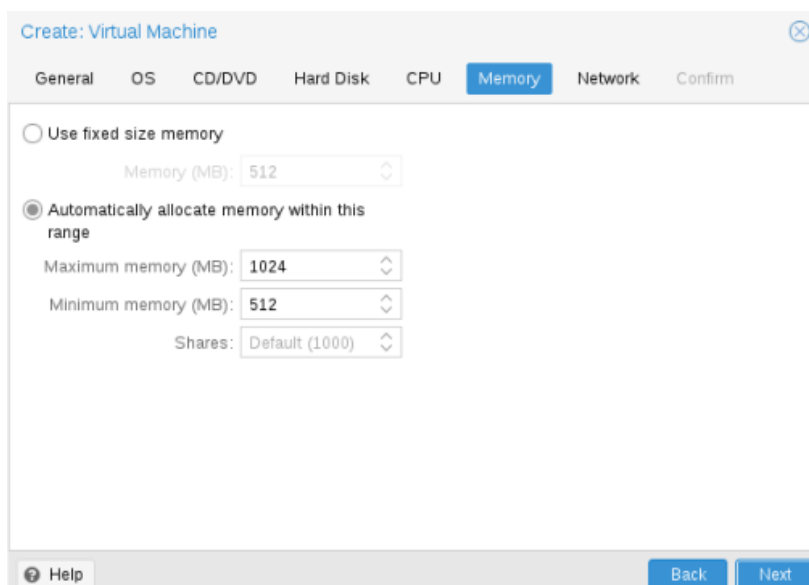
Minimum memory (MB): 512

Shares: Default (1000)

Help Back Next

当选择分配固定容量内存时，Proxmox VE 将为虚拟机直接分配相应容量的内存。

自动分配内存



Create: Virtual Machine

General OS CD/DVD Hard Disk CPU **Memory** Network Confirm

☐ Use fixed size memory

Memory (MB): 512

☒ Automatically allocate memory within this range

Maximum memory (MB): 1024

Minimum memory (MB): 512

Shares: Default (1000)

Help Back Next

当选择自动分配内存时，Proxmox VE 将为虚拟机至少分配设置的最小容量内存，并在物理服务器内存占用率达到 80%之前根据虚拟机需要动态分配内存，直到达到设置的最大内存分配量。

当物理服务器内存不足时，Proxmox VE 将开始回收分配给虚拟机的内存，并在必要时启动 SWAP 分区，如果仍然不能满足需要，最终将启动 oom 进程杀掉部分进程以释放内存。物理服务器和虚拟机之间的内存分配和释放通过虚拟机内的 balloon 驱动完成，该驱动主要用于从主机抓取或向主机释放内存页面。

当有多台虚拟机使用自动内存分配方式时，可以通过配置 “shares” 参数，在多个虚拟机之间分配可用内存份额。比如，假定现在有 4 台虚拟机，其中 3 台为 HTTP 服务器，1 台为数据库服务器。为了让数据库服务器能够使用更多内存缓存数据库数据，你会希望能优先给数据库服务器分配更多内存。为达此目的，可以设置数据库虚拟机的 Shares 为 3000，并设置其他 3 个 HTTP 虚拟机的 Shares 为默认值 1000。如果物理服务器有 32GB 内存，且目前已使用 16GB，那么可以提供给这 4 台虚拟机分配使用的物理内存有 $32 \times 80 / 100 - 16 = 9\text{GB}$ 。数据库虚拟机能获得的内存容量为 $9 \times 3000 / (3000 + 1000 + 1000 + 1000) = 4.5\text{GB}$ ，而每个 HTTP 虚拟机能获得 1.5GB。

2010 年以后，所有的 Linux 发行版默认都安装了 balloon 驱动。对于 Windows 系统，则需要手工安装 balloon 驱动，并且可能会导致系统性能降低，所以我们不建议在重要的 Windows 系统上安装 balloon 驱动。

当为虚拟机分配内存时，至少要为主机保留 1GB 可用内存。

8.2.6 网卡

Create: Virtual Machine

General OS CD/DVD Hard Disk CPU Memory **Network** Confirm

☒ Bridged mode

VLAN Tag: no VLAN

Bridge: vmbr0

Firewall: ☐

☐ NAT mode

☐ No network device

Model: VirtIO (paravirtualized)

MAC address: auto

Rate limit (MB/s): unlimited

Multiqueues: 1

Disconnect: ☐

Help Back Next

虚拟机可以配置多个网卡，共有以下四种类型虚拟网卡可以选择使用：

- Intel E1000 是默认配置的网卡类型，模拟了 Intel 千兆网卡设备。
- VirtIO 是半虚拟化网卡，具有较高的性能。但和其他 VirtIO 虚拟设备一样，虚拟机必须安装 virtio 驱动程序。
- Realtek 8139 模拟了旧的 100Mb/s 的网卡。当虚拟机使用旧版操作系统（2002 年以前发行）时，可以使用该类型虚拟网卡。
- Vmxnet3 是另一种半虚拟化网卡。可用于从其他类型虚拟化平台导入的虚拟机。

Proxmox VE 会为每一块虚拟网卡生成一个随机的 MAC 地址，以便虚拟机网络通信使用。

虚拟网卡的工作模式分为以下两种：

- 桥接模式下，每个虚拟网卡的底层都使用物理服务器上的 tap 设备（软件实现的 loopback 物理网卡设备）实现。该 tap 设备被添加到虚拟交换机上，如 Proxmox VE 默认的 vbr0，以便虚拟机直接访问物理服务器所连接的局域网 LAN。
- NAT 模式下，虚拟网卡将只能和 Qemu 的网络协议栈通信，并在内嵌的路由服务和 DHCP 服务的帮助下进行网络通信。内嵌的 DHCP 服务会在 10.0.2.0/24 范围内分配 IP 地址。由于 NAT 模式的性能远低于桥接模式，所以一般仅用于测试环境。

你可以在创建虚拟机时通过选择 No network device 跳过网络设备添加环节。

Multiqueue

如果你配置了 VirtIO 网卡，可以同时配置使用 Multiqueue 功能。启用 Multiqueue 可以让虚拟机同时使用多个虚拟 CPU 处理网络数据包，从而提高整体网络数据包处理能力。

在 Proxmox VE 下使用 VirtIO 网卡时，每个虚拟网卡的收发队列都传递给内核处理，每个收发队列的数据包都由虚拟主机驱动创建的一个内核线程负责处理。当启用 Multiqueue 后，可以为每个虚拟网卡创建多个收发队列交由主机内核处理。

使用 Multiqueue 时，推荐设置虚拟机收发队列数量和虚拟 CPU 数量一致。此外，还需要为每个虚拟 VirtIO 网卡设置多功能通道数量，命令如下：

```
ethtool -L eth0 combined X
```

其中 X 指虚拟机的虚拟 CPU 数量。

需要注意，当设置 `multiqueue` 参数值大于 1 时，网络流量增大会引发主机 CPU 和虚拟机 CPU 负载的升高。我们推荐仅在虚拟机需要处理大量网络数据包时启用该配置，例如用作路由器、反向代理或高负载 HTTP 服务器时。

8.2.7 USB 直通

Proxmox VE 支持两种 USB 直通方法：

- 基于主机的 USB 直通
- 基于 SPICE 协议的 USB 直通

基于主机的 USB 直通是将主机上的一个 USB 设备分配给虚拟机使用。具体可以通过指定厂商 ID 和设备 ID 分配，也可以通过指定主机总线号和端口号分配。

厂商/设备 ID 格式为：0123:abcd。其中 0123 为厂商 ID，abcd 为设备 ID，这意味着同样型号的 USB 设备将具有同样的 ID。

总线/端口编号格式为：1-2.3.4。其中 1 为总线号，2.3.4 为端口路径。合起来标识了主机上的一个物理端口（取决于 USB 控制器的内部顺序）。

即使虚拟机配置中的 USB 直通设备并未连接到物理服务器，虚拟机也可以顺利启动。在主机上指定的直通设备不可访问时，虚拟机会做跳过处理。

☒ 警告

由于 USB 直通设备只在当前主机上具备，所以使用 USB 直通的虚拟机将无法在线迁移到其他物理服务器。

第二种直通方式基于 SPICE 协议。这种直通方式需要 SPICE 客户端的支持。如果你给虚拟机添加了 SPICE USB 端口，那么就可以直接将 SPICE 客户端上的 USB 设备直通给虚拟机使用（例如输入设备或硬件加密狗）。

8.2.8 BIOS 和 UEFI

为了完美模拟计算机硬件，QEMU 也使用了固件。QEMU 默认使用开源 x86 BIOS 固件 SeaBIOS。大多数情况下，SeaBIOS 都是不错的选择。

当然，也有 BIOS 不能正常引导启动场景。比如，在配置 VGA 直通时。此时，使用开源 UEFI 固件 OVMF 更好。

使用 OVMF 有以下几点需要注意：

为了保存启动顺序等配置，需要为虚拟机添加一个 EFI 硬盘，并纳入备份和快照管理范围，并且只能有一块 EFI 硬盘。

EFI 硬盘添加命令如下：

```
qm set <vmid> -efidisk0 <storage>:1,format=<format>
```

其中<storage>是 Proxmox VE 存储服务名，<format>是存储格式。你也可以在 Web 控制台提供的虚拟机硬件配置界面通过添加 EFI 硬盘来完成该操作。

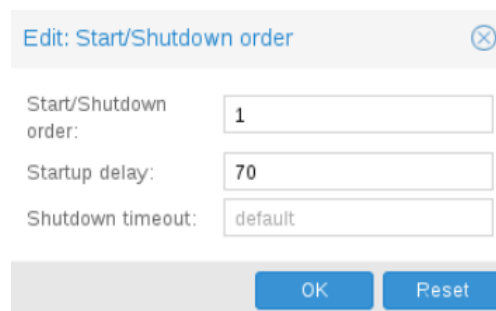
在 OVMF 下使用虚拟显示器时（而非通过 VGA 直通），你需要在 OVMF 菜单（在虚拟机启动时按 ESC 键可调出该菜单）中配置终端显示器的分辨率，或者选择使用 SPICE 显示器。

8.2.9 虚拟机自启动和自关闭

创建虚拟机后，如果需要虚拟机在 Proxmox VE 物理服务器开机后自动运行，需要在 Web 控制台的虚拟机 Option 选项卡中选择 “Start at boot ”，或者运行以下命令：

```
qm set <vmid> -onboot 1
```

开机顺序和关机顺序



某些场景下，你可能需要仔细调整各个虚拟机的启动顺序，比如为其他虚拟机提供防火墙或 DHCP 服务的虚拟机应该先启动。可以设置以下参数调整开关机顺序。

- Start/Shutdown order：用于设置开机优先级。例如，设置为 1 表示该虚拟机需要第一个被启动（关机顺序和开机顺序相反，所以设置为 1 的虚拟机会最后被关闭）。

- Startup delay：用于设置当前虚拟机开机后到下一个虚拟机开机前的时间间隔。例如，设置为 240 表示时间间隔为 240 秒。
- Shutdown timeout：用于设置关机命令发出后 Proxmox VE 等待虚拟机关机的时间间隔。该参数默认值为 60，也就是说 Proxmox VE 发出关机命令后会花 60 秒时间等待虚拟机完成关机操作，如果 60 秒后虚拟机仍未完成关机，Proxmox VE 会通知管理员关机失败。

需要注意，未设置 Start/Shutdown order 参数的虚拟机总是在设置了该参数的虚拟机之后启动，而且该参数仅能影响同一 Proxmox VE 服务器上虚拟机的启动顺序，其作用域局限于单一 Proxmox VE 服务器内部，而非整个集群。

8.3 虚拟机管理命令 qm

命令 qm 是 Proxmox VE 提供的用于管理 Qemu/KVM 虚拟机的命令行工具。通过该命令，可以创建或销毁虚拟机，也可以控制虚拟机运行状态（启动/停止/挂起/恢复）。此外，还可以利用 qm 设置虚拟机配置文件中的参数，创建或删除虚拟磁盘。

8.3.1 命令行示例

新建一个虚拟机，配置 4GB 的 IDE 虚拟硬盘。

```
qm create 300 -ide0 4 -net0 e1000 -cdrom proxmox-mailgateway_2.1.iso
```

启动新建虚拟机。

```
qm start 300
```

发出关机命令，并等待直到虚拟机关机。

```
qm shutdown 300 && qm wait 300
```

发出关机命令，并等待 40 秒。

```
qm shutdown 300 && qm wait 300 -timeout 40
```

8.4 虚拟机迁移

在 Promxox VE 集群中，可以将虚拟机迁移到其他服务器运行。命令如下：

```
qm migrate <vmid> <target>
```

如果虚拟机没有配置使用 Proxmox VE 服务器的本地资源（例如 local 存储上的虚拟磁盘，直通物理设备等），你可以增加-online 参数发起在线迁移命令，也就是在虚拟机开机运行状态下进行迁移操作。

即使虚拟机使用了 Proxmox VE 服务器本地资源，但只要虚拟硬盘所处的存储服务在源服务器和目的服务器都有配置，仍然可以离线迁移虚拟机。迁移操作中，Proxmox VE 会通过网络将虚拟机硬盘镜像复制到目标服务器。

8.5 虚拟机配置文件

虚拟机配置文件保存在 Proxmox 集群文件系统中，并可以通过路径/etc/pve/qemu-server/<VMID>.conf 访问。和/etc/pve 下的其他文件一样，虚拟机配置文件会自动同步复制到集群的其他节点。

✓ 注意

小于 100 的 VMID 被 Proxmox VE 保留内部使用，并且在集群内的 VMID 不能重复。

虚拟机配置文件示例

```
cores: 1
sockets: 1
memory: 512
name: webmail
ostype: l26
bootdisk: virtio0
net0: e1000=EE:D2:28:5F:B6:3E,bridge=vmbro
virtio0: local:vm-100-disk-1,size=32G
```

虚拟机配置文件就是普通文本文件，可以直接使用常见文本编辑器（vi，nano 等）编辑。这也是日常对虚拟机配置文件进行细微调整的一般做法。但是务必注意，必须彻底关闭虚拟机，然后再启动虚拟机，修改后的配置才能生效。

因此，更好的做法是使用 `qm` 命令或 WebGUI 来创建或修改虚拟机配置文件。
Proxmox VE 能够直接将大部分变更直接应用到运行中的虚拟机，并即时生效。该特性称为“热插拔”，并无需重启虚拟机。

8.5.1 配置文件格式

虚拟机配置文件使用英文冒号字符 “:” 为分隔符的键/值格式。格式如下：

```
# this is a comment
```

```
OPTION: value
```

空行会被自动忽略，以字符 “#” 开头的行按注释处理，也会被自动忽略。

8.5.2 虚拟机快照

创建虚拟机快照后，`qm` 会在配置文件中创建一个小节，专门保存创建虚拟机快照时的虚拟机配置。例如，创建名为 “testsnapshot” 的虚拟机快照后，虚拟机配置文件内容可能会像下面这样：

创建快照后的虚拟机配置文件示例

```
memory: 512
```

```
swap: 512
```

```
parent: testsnaphot
```

```
...
```

```
[testsnaphot]
```

```
memory: 512
```

```
swap: 512
```

```
snaptime: 1457170803
```

```
...
```

其中 `parent` 和 `snaptime` 是和虚拟机快照相关的配置属性。属性 `parent` 用于保存快照之间的父/子关系，属性 `snaptime` 是创建快照的时间戳（Unix epoch）。

8.5.3 虚拟机配置项目

- `acpi` : <boolean> (default = 1)
启用/禁用 ACPI。
- `agent` : <boolean> (default = 0)
启用/禁用 Qemu GuestAgent.
- `args` : <string>
传递给 kvm 的任意参数，例如：
`args: -no-reboot -no-hpet`

✓ 注意

配置项目 `args` 仅供专家使用。

- `autostart` : <boolean> (default = 0)
虚拟机崩溃后自动启动（目前该属性会被自动忽略）
- `balloon` : <integer> (0 -N)
为虚拟机配置的目标内存容量，单位为 MB。设为 0 表示禁用 balloon 驱动程序。
- `bios` : <ovmf | seabios> (default = seabios)
设置 BIOS 类型。
- `boot` : [acdn]{1,4} (default = cdn)
虚拟机启动顺序，软驱（a），硬盘（c），光驱（d），或网络（n）。
- `bootdisk` : (ide|sata|scsi|virtio)\d+
指定硬盘为系统启动盘。
- `cdrom` : <volume>
相当于-ide2 的别名。
- `cores` : <integer> (1 -N) (default = 1)
每个虚拟 CPU 的核心数。

- `cpu : [cputype=]<enum> [,hidden=<1|0>]`

模拟 CPU 类型。

`cputype = <486 | Broadwell | Broadwell-noTSX | Conroe | Haswell |
Haswell-noTSX | IvyBridge | Nehalem | Opteron_G1 | Opteron_G2 |
Opteron_G3 | Opteron_G4 | Opteron_G5 | Penryn | SandyBridge |
Westmere | athlon | core2duo | coreduo | host | kvm32 | kvm64 |
pentium | pentium2 | pentium3 | phenom | qemu32 | qemu64> (default
= kvm64)`

以上为可选的模拟 CPU 类型值。

`hidden = <boolean> (default = 0)`

设为 1 表示不标识为 KVM 虚拟机。

- `cpulimit : <number> (0 -128) (default = 0)`

CPU 配额上限值。

✓ 注意

如果一台计算机有 2 个 CPU，那么该计算机一共有 2 份额的 CPU 时间片可以分配。设为 0 表示不限制 CPU 配额。

- `cpuunits : <integer> (0 -500000) (default = 1024)`

虚拟机的 CPU 时间片分配权重值。该参数供内核的公平调度器使用。设定的权重值越大，虚拟机得到的 CPU 时间片越多。最终分配得到的时间片由该虚拟机权重和所有其他虚拟机权重总和之比决定。

✓ 注意

可以将该值设为 0，从而禁用内核的公平调度器。

- description : <string>

虚拟机描述信息。仅供 WebGUI 使用。在虚拟机配置文件中以注释形式保存。

- efidisk0 : [file=]<volume> [,format=<enum>] [,size=<DiskSize>]

配置 EFI 类型虚拟硬盘。

file = <volume>

EFI 虚拟硬盘所基于的存储服务卷名称。

format = <cloop | cow | qcow | qcow2 | qed | raw | vmdk>

EFI 虚拟硬盘所采用的存储格式。

size = <DiskSize>

EFI 虚拟硬盘容量。仅供显示使用，并不能影响实际容量大小。

- freeze : <boolean>

虚拟机启动时自动冻结 CPU（使用监视器命令 c 可继续启动过程）。

- hostpci[n] : [host=]<HOSTPCIID[;HOSTPCIID2...]> [,pcie=<1|0>] [,rombar=<1|0>] [,x-vga=<1|0>]

将物理主机 PCI 设备映射给虚拟机。

✓ 注意

该属性允许虚拟机直接访问物理主机硬件。启用后将不能再进行虚拟机迁移操作，因此使用时务必小心。

☒ 警告

该特性仍处于试验阶段，有用户报告该属性会导致故障和问题。

host = <HOSTPCIID[;HOSTPCIID2...]>

将 PCI 设备直通虚拟机使用。可指定一个或一组设备的 PCI ID。HOSTPCIID 格式为“总线号:设备号.功能号”（16 进制数字表示），具体可使用 lspci 命令查看。

pcie = <boolean> (default = 0)

标明是否是 PCI-express 类型总线（用于 q35 类型计算机）。

rombar = <boolean> (default = 1)

标明是否将设备 ROM 映射至虚拟机内存空间。

x-vga = <boolean> (default = 0)

标明是否启用 vfio-vga 设备支持。

- hotplug : <string> (default = network,disk,usb)

设置启用的热插拔设备类型。启用热插拔的设备类型之间用英文逗号字符分隔，

可选参数值包括 network , disk , cpu , memory 和 usb。设为 0 表示禁用热插拔，

设为 1 表示启用默认值 network,disk,usb.

- hugepages : <1024 | 2 | any>

启用/禁用巨型页。

- ide[n] : [file=<volume> [,aio=<native|threads>] [,backup=<1|0>]
[,bps=<bps>] [,bps_max_length=<seconds>] [,bps_rd=<bps>]
[,bps_rd_length=<seconds>] [,bps_wr=<bps>] [,bps_wr_length=<seconds>]
[,cache=<enum>] [,cyls=<integer>] [,detect_zeroes=<1|0>]
[,discard=<ignore|on>] [,format=<enum>] [,heads=<integer>] [,iops=<iops>]
[,iops_max=<iops>] [,iops_max_length=<seconds>] [,iops_rd=<iops>]
[,iops_rd_length=<seconds>] [,iops_rd_max=<iops>] [,iops_wr=<iops>]
[,iops_wr_length=<seconds>] [,iops_wr_max=<iops>] [,mbps=<mbps>]
[,mbps_max=<mbps>] [,mbps_rd=<mbps>] [,mbps_rd_max=<mbps>]
[,mbps_wr=<mbps>] [,mbps_wr_max=<mbps>] [,media=<cdrom|disk>]
[,model=<model>] [,rerror=<ignore|report|stop>] [,secs=<integer>]
[,serial=<serial>] [,size=<DiskSize>] [,snapshot=<1|0>]
[,trans=<none|lba|auto>] [,werror=<enum>]

配置 IDE 类型虚拟硬盘或光驱（n 的值为 0-3）。

aio = <native | threads>

指定 AIO 类型。

backup = <boolean>

设置虚拟硬盘在进行虚拟机备份时是否被纳入备份范围。

bps = <bps>

最大读写操作速度，单位为字节/秒。

bps_max_length = <seconds>

突发读写操作最大时间长度，单位为秒。

bps_rd = <bps>

最大读操作速度，单位为字节/秒。

bps_rd_length = <seconds>

突发读操作最大时间长度，单位为秒。

bps_wr = <bps>

最大写操作速度，单位为字节/秒。

bps_wr_length = <seconds>

突发写操作最大时间长度，单位为秒。

cache = <directsync | none | unsafe | writeback | writethrough>

虚拟硬盘缓存工作模式。

cyls = <integer>

强制指定虚拟硬盘物理几何参数中的 cylinder 值。

detect_zeroes = <boolean>

设置是否检测并优化零写入操作。

discard = <ignore | on>

设置是否向下层存储服务传递 discard/trim 操作请求。

file = <volume>

IDE 虚拟硬盘所基于的存储服务卷名称。

format = <cloop | cow | qcow | qcow2 | qed | raw | vmdk>

IDE 虚拟硬盘所采用的存储格式。

heads = <integer>

强制指定虚拟硬盘物理几何参数中的 head 值。

iops = <iops>

最大读写 I/O 速度，单位为个/秒。

iops_max = <iops>

最大无限制读写 I/O 速度，单位为个/秒。

iops_max_length = <seconds>

突发读写操作最大时间长度，单位为秒。

iops_rd = <iops>

最大读 I/O 速度，单位为个/秒。

iops_rd_length = <seconds>

突发读操作最大时间长度，单位为秒。

iops_rd_max = <iops>

最大无限制读 I/O 速度，单位为个/秒。

iops_wr = <iops>

最大写 I/O 速度，单位为个/秒。

iops_wr_length = <seconds>

突发写操作最大时间长度，单位为秒。

iops_wr_max = <iops>

最大无限制写 I/O 速度，单位为个/秒。

mbps = <mbps>

最大读写操作速度，单位为 MB/秒。

mbps_max = <mbps>

最大无限制读写操作速度，单位为 MB/秒。

mbps_rd = <mbps>

最大读操作速度，单位为 MB/秒。

mbps_rd_max = <mbps>

最大无限制读操作速度，单位为 MB/秒。

mbps_wr = <mbps>

最大写操作速度，单位为 MB/秒。

mbps_wr_max = <mbps>

最大无限制写操作速度，单位为 MB/秒。

media = <cdrom | disk> (default = disk)

虚拟硬盘驱动器介质类型。

model = <model>

虚拟硬盘的模型名，基于 url 编码格式，最大 40 字节。

error = <ignore | report | stop>

读错误处理方式。

secs = <integer>

强制指定虚拟硬盘物理几何参数中的 sector 值。

serial = <serial>

虚拟硬盘的序列号，基于 url 编码格式，最大 20 字节。

size = <DiskSize>

虚拟硬盘容量。仅供显示使用，并不能影响实际容量大小。

snapshot = <boolean>

设置虚拟硬盘在创建虚拟机快照时是否被纳入快照范围。

trans = <auto | lba | none>

设置虚拟硬盘几何参数地址 bios 解释模式。

werror = <enospc | ignore | report | stop>

写错误处理方式。

- keyboard : <da | de | de-ch | en-gb | en-us | es | fi | fr | fr-be | fr-ca | fr-ch | hu | is | it | ja | lt | mk | nl | no | pl | pt | pt-br | sl | sv | tr> (default = en-us)

键盘布局设置，用于 vnc 服务器。默认采用/etc/pve/datacenter.conf 中的设置值。

- kvm : <boolean> (default = 1)

启用/禁用 KVM 硬件虚拟化。

- localtime : <boolean>

设置虚拟机时间是否采用服务器本地时间。如虚拟机操作系统类型为 Microsoft OS , 则默认启用该项。

- lock : <backup | migrate | rollback | snapshot>

锁定/解锁虚拟机。

- machine : (pc|pc(-i440fx)?-\\d+\\.\\d+(\\.pxe)?|q35|pc-q35-\\d+\\.\\d+(\\.pxe)?)

设置 Qemu 虚拟机类型。

- memory : <integer> (16 -N) (default = 512)

设置虚拟机内存容量，单位为 MB。启用 balloon 驱动时，该值为最大可用内存值。

- migrate_downtime : <number> (0 -N) (default = 0.1)

设置虚拟机在线迁移时最大停机时间（单位为秒）。

- migrate_speed : <integer> (0 -N) (default = 0)

设置虚拟机迁移时最大数据传输速度（单位为 MB/s）。设为 0 表示不限速。

- name : <string>

设置虚拟机名称。仅用于 WebGUI 界面。

- net[n] : [model=<enum> [,bridge=<bridge>] [,firewall=<1|0>] [,link_down=<1|0>] [,macaddr=<XX:XX:XX:XX:XX:XX>] [,queues=<integer>] [,rate=<number>] [,tag=<integer>] [,trunks=<vlanid[;vlanid...]>] [,<model>=<macaddr>]

设置虚拟网络设备。

bridge = <bridge>

虚拟网络设备桥接的虚拟交换机。Proxmox VE 默认创建虚拟交换机名为 vbr0。

如未指定虚拟交换机，Proxmox VE 将创建 KVM 网络设备（NAT），并提供 DHCP

和 DNS 服务。具体地址如下：

10.0.2.2 Gateway

10.0.2.3 DNS Server

10.0.2.4 SMB Server

其中 DHCP 服务器将从 10.0.2.15 开始分配 IP 地址

firewall = <boolean>

设置虚拟网络设备是否被防火墙保护。

link_down = <boolean>

设置虚拟网络设备是否可以被断开连接（类似于拔掉网线）。

macaddr = <XX:XX:XX:XX:XX:XX>

MAC 地址。不能和已有 MAC 地址重复。如未指定，系统将自动生成一个。

model = <e1000 | e1000-82540em | e1000-82544gc | e1000-82545em |
i82551 | i82557b | i82559er | ne2k_isa | ne2k_pci | pcnet |
rtl8139 | virtio | vmxnet3>

虚拟网卡类型。其中 virtio 性能最好。如虚拟机不支持 virtio，最好使用 e1000。

queues = <integer> (0 -16)

设置虚拟网卡的包队列数量。

rate = <number> (0 -N)

虚拟网卡最大传输速度，单位为 Mb/s，值为浮点数。

tag = <integer> (1 -4094)

虚拟网卡在数据包上自动标记的 VLAN 号。

trunks = <vlanid[;vlanid...]>

虚拟网卡所连接的 VLAN 号。

- numa : <boolean> (default = 0)

启用/禁用 NUMA。

- numa[n] : cpus=<id[-id];...> [,hostnodes=<id[-id];...>] [,memory=<number>]
[,policy=<preferred|bind|interleave>]

NUMA 拓扑结构。

cpus = <id[-id];...>

当前 NUMA 节点上的 CPU 列表。

hostnodes = <id[-id];...>

所采用的主机 NUMA 节点。

memory = <number>

NUMA 节点所提供的内存容量。

policy = <bind | interleave | preferred>

NUMA 分配策略。

- onboot : <boolean> (default = 0)

设置虚拟机是否在物理服务器启动时自动启动。

- ostype : <l24 | l26 | other | solaris | w2k | w2k3 | w2k8 | win10 | win7 | win8 | wvista | wxp>

虚拟机操作系统类型。用于启用针对操作系统的优化和功能特性。可选值如下：

other	unspecified OS
wxp	Microsoft Windows XP
w2k	Microsoft Windows 2000
w2k3	Microsoft Windows 2003
w2k8	Microsoft Windows 2008
wvista	Microsoft Windows Vista
win7	Microsoft Windows 7
win8	Microsoft Windows 8/2012
l24	Linux 2.4 Kernel
l26	Linux 2.6/3.X Kernel
solaris	Solaris/OpenSolaris/OpenIndiana kernel

- parallel[n] : /dev/parport\d+ | /dev/usb/lp\d+

将物理主机并口设备映射给虚拟机(n 的值为 0-2)。

✓ 注意

该属性允许虚拟机直接访问物理主机硬件。启用后将不能再进行虚拟机迁移操作，因此使用时务必小心。

☒ 警告

该特性仍处于试验阶段，有用户报告该属性会导致故障和问题。

-
- protection : <boolean> (default = 0)

设置虚拟机保护标识。启用后将禁止删除虚拟机或虚拟机硬盘。

- reboot : <boolean> (default = 1)

允许虚拟机重启。设为 0 后，虚拟机重启时将自动关闭。

- sata[n] : [file=]<volume> [,aio=<native|threads>] [,backup=<1|0>]
[,bps=<bps>] [,bps_max_length=<seconds>] [,bps_rd=<bps>]
[,bps_rd_length=<seconds>] [,bps_wr=<bps>] [,bps_wr_length=<seconds>]
[,cache=<enum>] [,cyls=<integer>] [,detect_zeroes=<1|0>]
[,discard=<ignore|on>] [,format=<enum>] [,heads=<integer>] [,iops=<iops>]
[,iops_max=<iops>] [,iops_max_length=<seconds>] [,iops_rd=<iops>]
[,iops_rd_length=<seconds>] [,iops_rd_max=<iops>] [,iops_wr=<iops>]
[,iops_wr_length=<seconds>] [,iops_wr_max=<iops>] [,mbps=<mbps>]
[,mbps_max=<mbps>] [,mbps_rd=<mbps>] [,mbps_rd_max=<mbps>]
[,mbps_wr=<mbps>] [,mbps_wr_max=<mbps>] [,media=<cdrom|disk>]
[,error=<ignore|report|stop>] [,secs=<integer>] [,serial=<serial>]
[,size=<DiskSize>] [,snapshot=<1|0>] [,trans=<none|lba|auto>]
[,werror=<enum>]

配置 SATA 类型虚拟硬盘或光驱（n 的值为 0-5）。

aio = <native | threads>

指定 AIO 类型。

backup = <boolean>

设置虚拟硬盘在进行虚拟机备份时是否被纳入备份范围。

bps = <bps>

最大读写操作速度，单位为字节/秒。

bps_max_length = <seconds>

突发读写操作最大时间长度，单位为秒。

bps_rd = <bps>

最大读操作速度，单位为字节/秒。

bps_rd_length = <seconds>

突发读操作最大时间长度，单位为秒。

bps_wr = <bps>

最大写操作速度，单位为字节/秒。

bps_wr_length = <seconds>

突发写操作最大时间长度，单位为秒。

cache = <directsync | none | unsafe | writeback | writethrough>

虚拟硬盘缓存工作模式。

cyls = <integer>

强制指定虚拟硬盘物理几何参数中的 cylinder 值。

detect_zeroes = <boolean>

设置是否检测并优化零写入操作。

discard = <ignore | on>

设置是否向下层存储服务传递 discard/trim 操作请求。

file = <volume>

IDE 虚拟硬盘所基于的存储服务卷名称。

format = <cloop | cow | qcow | qcow2 | qed | raw | vmdk>

IDE 虚拟硬盘所采用的存储格式。

heads = <integer>

强制指定虚拟硬盘物理几何参数中的 head 值。

iops = <iops>

最大读写 I/O 速度，单位为个/秒。

iops_max = <iops>

最大无限制读写 I/O 速度，单位为个/秒。

iops_max_length = <seconds>

突发读写操作最大时间长度，单位为秒。

iops_rd = <iops>

最大读 I/O 速度，单位为个/秒。

iops_rd_length = <seconds>

突发读操作最大时间长度，单位为秒。

iops_rd_max = <iops>

最大无限制读 I/O 速度，单位为个/秒。

iops_wr = <iops>

最大写 I/O 速度，单位为个/秒。

iops_wr_length = <seconds>

突发写操作最大时间长度，单位为秒。

iops_wr_max = <iops>

最大无限制写 I/O 速度，单位为个/秒。

mbps = <mbps>

最大读写操作速度，单位为 MB/秒。

mbps_max = <mbps>

最大无限制读写操作速度，单位为 MB/秒。

mbps_rd = <mbps>

最大读操作速度，单位为 MB/秒。

mbps_rd_max = <mbps>

最大无限制读操作速度，单位为 MB/秒。

mbps_wr = <mbps>

最大写操作速度，单位为 MB/秒。

mbps_wr_max = <mbps>

最大无限制写操作速度，单位为 MB/秒。

media = <cdrom | disk> (default = disk)

虚拟硬盘驱动器介质类型。

error = <ignore | report | stop>

读错误处理方式。

secs = <integer>

强制指定虚拟硬盘物理几何参数中的 sector 值。

serial = <serial>

虚拟硬盘的序列号，基于 url 编码格式，最大 20 字节。

size = <DiskSize>

虚拟硬盘容量。仅供显示使用，并不能影响实际容量大小。

snapshot = <boolean>

设置虚拟硬盘在创建虚拟机快照时是否被纳入快照范围。

trans = <auto | lba | none>

设置虚拟硬盘几何参数地址 bios 解释模式。

werror = <enospc | ignore | report | stop>

写错误处理方式。

- scsi[n] : [file=]<volume> [,aio=<native|threads>] [,backup=<1|0>]
[,bps=<bps>] [,bps_max_length=<seconds>] [,bps_rd=<bps>]
[,bps_rd_length=<seconds>] [,bps_wr=<bps>] [,bps_wr_length=<seconds>]
[,cache=<enum>] [,cyls=<integer>] [,detect_zeroes=<1|0>]
[,discard=<ignore|on>] [,format=<enum>] [,heads=<integer>] [,iops=<iops>]
[,iops_max=<iops>] [,iops_max_length=<seconds>] [,iops_rd=<iops>]
[,iops_rd_length=<seconds>] [,iops_rd_max=<iops>] [,iops_wr=<iops>]
[,iops_wr_length=<seconds>] [,iops_wr_max=<iops>] [,iothread=<1|0>]
[,mbps=<mbps>] [,mbps_max=<mbps>] [,mbps_rd=<mbps>]
[,mbps_rd_max=<mbps>] [,mbps_wr=<mbps>] [,mbps_wr_max=<mbps>]
[,media=<cdrom|disk>] [,queues=<integer>] [,secs=<integer>]
[,serial=<serial>] [,size=<DiskSize>] [,snapshot=<1|0>]
[,trans=<none|lba|auto>] [,werror=<enum>]

配置 SCSI 类型虚拟硬盘或光驱（n 的值为 0-13）。

aio = <native | threads>

指定 AIO 类型。

backup = <boolean>

设置虚拟硬盘在进行虚拟机备份时是否被纳入备份范围。

bps = <bps>

最大读写操作速度，单位为字节/秒。

bps_max_length = <seconds>

突发读写操作最大时间长度，单位为秒。

bps_rd = <bps>

最大读操作速度，单位为字节/秒。

bps_rd_length = <seconds>

突发读操作最大时间长度，单位为秒。

bps_wr = <bps>

最大写操作速度，单位为字节/秒。

bps_wr_length = <seconds>

突发写操作最大时间长度，单位为秒。

cache = <directsync | none | unsafe | writeback | writethrough>

虚拟硬盘缓存工作模式。

cyls = <integer>

强制指定虚拟硬盘物理几何参数中的 cylinder 值。

detect_zeroes = <boolean>

设置是否检测并优化零写入操作。

discard = <ignore | on>

设置是否向下层存储服务传递 discard/trim 操作请求。

file = <volume>

IDE 虚拟硬盘所基于的存储服务卷名称。

format = <cloop | cow | qcow | qcow2 | qed | raw | vmdk>

IDE 虚拟硬盘所采用的存储格式。

heads = <integer>

强制指定虚拟硬盘物理几何参数中的 head 值。

iops = <iops>

最大读写 I/O 速度，单位为个/秒。

iops_max = <iops>

最大无限制读写 I/O 速度，单位为个/秒。

iops_max_length = <seconds>

突发读写操作最大时间长度，单位为秒。

iops_rd = <iops>

最大读 I/O 速度，单位为个/秒。

iops_rd_length = <seconds>

突发读操作最大时间长度，单位为秒。

iops_rd_max = <iops>

最大无限制读 I/O 速度，单位为个/秒。

iops_wr = <iops>

最大写 I/O 速度，单位为个/秒。

iops_wr_length = <seconds>

突发写操作最大时间长度，单位为秒。

iops_wr_max = <iops>

最大无限制写 I/O 速度，单位为个/秒。

iothread = <boolean>

设置是否启用 iothreads。

mbps = <mbps>

最大读写操作速度，单位为 MB/秒。

mbps_max = <mbps>

最大无限制读写操作速度，单位为 MB/秒。

mbps_rd = <mbps>

最大读操作速度，单位为 MB/秒。

mbps_rd_max = <mbps>

最大无限制读操作速度，单位为 MB/秒。

mbps_wr = <mbps>

最大写操作速度，单位为 MB/秒。

mbps_wr_max = <mbps>

最大无限制写操作速度，单位为 MB/秒。

media = <cdrom | disk> (default = disk)

虚拟硬盘驱动器介质类型。

error = <ignore | report | stop>

读错误处理方式。

queues = <integer> (2 -N)

设置队列数量。

secs = <integer>

强制指定虚拟硬盘物理几何参数中的 sector 值。

serial = <serial>

虚拟硬盘的序列号，基于 url 编码格式，最大 20 字节。

size = <DiskSize>

虚拟硬盘容量。仅供显示使用，并不能影响实际容量大小。

snapshot = <boolean>

设置虚拟硬盘在创建虚拟机快照时是否被纳入快照范围。

trans = <auto | lba | none>

设置虚拟硬盘几何参数地址 bios 解释模式。

werror = <enospc | ignore | report | stop>

写错误处理方式。

- scsihw : <lsi | lsi53c810 | megasas | pvscsi | virtio-scsi-pci | virtio-scsi-single>
(default = lsi)

设置 SCSI 控制器类型。

- serial[n] : (/dev/.+|socket)

在虚拟机内创建串口设备（n 的值为 0-3），并直通到物理服务器的串口设备（如 /dev/ttyS0）或主机上创建的 unix socket（可使用 qm terminal 打开终端连接）。

✓ 注意

该属性允许虚拟机直接访问物理主机硬件。启用后将不能再进行虚拟机迁移操作，因此使用时务必小心。

☒ 警告

该特性仍处于试验阶段，有用户报告该属性会导致故障和问题。

- shares : <integer> (0 -50000) (default = 1000)

用于 auto-ballooning 的共享内存容量。设置的值越大，虚拟机得到的内存越多。具体分配到的内存由当前虚拟机的权重和所有其他虚拟机权重之和的比例决定。设置为 0 表示禁用 auto-ballooning。

- smbios1 : [family=<string>] [,manufacturer=<string>] [,product=<string>] [,serial=<string>] [,sku=<string>] [,uuid=<UUID>] [,version=<string>]

设置 SMBIOS 的类型 1 字段。

family = <string>

设置 SMBIOS1 家族名称。

manufacturer = <string>

设置 SMBIOS1 厂商名。

product = <string>

设置 SMBIOS1 产品 ID。

serial = <string>

设置 SMBIOS1 序列号。

sku = <string>

设置 SMBIOS1 的 SKU 字符串。

uuid = <UUID>

设置 SMBIOS1 的 UUID。

version = <string>

设置 SMBIOS1 版本。

- smp : <integer> (1 -N) (default = 1)

设置 CPU 数量。请使用选项-sockets 代替。

- sockets : <integer> (1 -N) (default = 1)

设置 CPU 的 sockets 数量。

- startdate : (now | YYYY-MM-DD | YYYY-MM-DDTHH:MM:SS) (default = now)

设置系统时钟的初始时间。日期格式示例为：now 或 2006-06-17T16:01:21 或 2006-06-17。

- startup : `[order=]\d+` [,up=\d+] [,down=\d+] `

开机或关机操作。参数 order 是一个非负整数值，用于指定虚拟机开机启动顺序。

关机顺序和开机顺序相反。此外还可以设置开机或关机的延迟时间，也就是当前虚拟机启动或关机后下一个虚拟机开机或关机的等待时间。

- tablet : <boolean> (default = 1)

启用/禁用 USB 指针设备。该设备一般用来允许在 VNC 中使用鼠标的绝对坐标。

否则鼠标将和 VNC 终端内的位置不同步。如果你在一台主机上运行了很多通过 VNC 终端访问的虚拟机，可以考虑禁用该参数以节省不必要的上下文切换。该项在 spice 终端下默认是禁用的 (-vga=gxl)。

- tdf : <boolean> (default = 0)

启用/禁用时间漂移修正。

- template : <boolean> (default = 0)

启用/禁用模板。

- unused[n] : <string>

用于标识未使用的存储卷。该参数仅供内部使用，不要手工编辑该参数。

- usb[n] : [host=]<HOSTUSBDEVICE|spice> [,usb3=<1|0>]

设置 USB 设备 (n 的值为 0 到 4)。

host = <HOSTUSBDEVICE|spice>

主机 USB 设备或端口或值 spice。HOSTUSBDEVICE 的配置语法为：

' bus-port(.port) * ' (十进制数) 或

' vendor_id:product_id' (十六进制数) 或

' spice'

可使用命令 `lsusb -t` 列出当前的 usb 设备。

✓ 注意

该属性允许虚拟机直接访问物理主机硬件。启用后将不能再进行虚拟机迁移操作，因此使用时务必小心。

值 `spice` 用于设置 `usb` 设备到 `spice` 的重定向。

`usb3 = <boolean> (default = 0)`

标识是否为 USB3 设备或端口（该参数在 `spice` 重定向下不稳定，将被忽略）。

- `vcpus : <integer> (1 -N) (default = 0)`

可热插拔的虚拟 `cpu` 数量。

- `vga : <cirrus | qxl | qxl2 | qxl3 | qxl4 | serial0 | serial1 | serial2 | serial3 | std | vmware>`

设置 `VGA` 显示器类型。如果使用高分辨率显示器（`>=1280x1024x16`），应该设

置为 `std` 或 `vmware`。默认情况下，操作系统 `win8/win7/win2k8` 使用 `std`，其他操

作系统使用 `cirrus`。设置为 `qxl` 将启用 `SPICE` 显示服务器。对于 `Win` 系列操作系统，

可以设置多个独立显示器，对于 `Linux` 系统可以自行添加显示器。也可以设置不使

用图形显示卡，而使用串口设备作为终端。

- `virtio[n] : [file=<volume> [,aio=<native|threads>] [,backup=<1|0>] [,bps=<bps>] [,bps_max_length=<seconds>] [,bps_rd=<bps>] [,bps_rd_length=<seconds>] [,bps_wr=<bps>] [,bps_wr_length=<seconds>] [,cache=<enum>] [,cyls=<integer>] [,detect_zeroes=<1|0>] [,discard=<ignore|on>] [,format=<enum>] [,heads=<integer>] [,iops=<iops>] [,iops_max=<iops>] [,iops_max_length=<seconds>] [,iops_rd=<iops>] [,iops_rd_length=<seconds>] [,iops_rd_max=<iops>] [,iops_wr=<iops>] [,iops_wr_length=<seconds>] [,iops_wr_max=<iops>] [,iothread=<1|0>] [,mbps=<mbps>] [,mbps_max=<mbps>] [,mbps_rd=<mbps>] [,mbps_rd_max=<mbps>] [,mbps_wr=<mbps>] [,mbps_wr_max=<mbps>] [,media=<cdrom|disk>] [,rerror=<ignore|report|stop>] [,secs=<integer>] [,serial=<serial>] [,size=<DiskSize>] [,snapshot=<1|0>] [,trans=<none|lba|auto>] [,werror=<enum>]`

配置 `VIRTIO` 类型虚拟硬盘（`n` 的值为 `0-15`）。

`aio = <native | threads>`

指定 `AIO` 类型。

backup = <boolean>

设置虚拟硬盘在进行虚拟机备份时是否被纳入备份范围。

bps = <bps>

最大读写操作速度，单位为字节/秒。

bps_max_length = <seconds>

突发读写操作最大时间长度，单位为秒。

bps_rd = <bps>

最大读操作速度，单位为字节/秒。

bps_rd_length = <seconds>

突发读操作最大时间长度，单位为秒。

bps_wr = <bps>

最大写操作速度，单位为字节/秒。

bps_wr_length = <seconds>

突发写操作最大时间长度，单位为秒。

cache = <directsync | none | unsafe | writeback | writethrough>

虚拟硬盘缓存工作模式。

cyls = <integer>

强制指定虚拟硬盘物理几何参数中的 cylinder 值。

detect_zeroes = <boolean>

设置是否检测并优化零写入操作。

discard = <ignore | on>

设置是否向下层存储服务传递 discard/trim 操作请求。

file = <volume>

IDE 虚拟硬盘所基于的存储服务卷名称。

format = <cloop | cow | qcow | qcow2 | qed | raw | vmdk>

IDE 虚拟硬盘所采用的存储格式。

heads = <integer>

强制指定虚拟硬盘物理几何参数中的 head 值。

iops = <iops>

最大读写 I/O 速度，单位为个/秒。

iops_max = <iops>

最大无限制读写 I/O 速度，单位为个/秒。

iops_max_length = <seconds>

突发读写操作最大时间长度，单位为秒。

iops_rd = <iops>

最大读 I/O 速度，单位为个/秒。

iops_rd_length = <seconds>

突发读操作最大时间长度，单位为秒。

iops_rd_max = <iops>

最大无限制读 I/O 速度，单位为个/秒。

iops_wr = <iops>

最大写 I/O 速度，单位为个/秒。

iops_wr_length = <seconds>

突发写操作最大时间长度，单位为秒。

iops_wr_max = <iops>

最大无限制写 I/O 速度，单位为个/秒。

iothread = <boolean>

设置是否启用 iothreads。

mbps = <mbps>

最大读写操作速度，单位为 MB/秒。

mbps_max = <mbps>

最大无限制读写操作速度，单位为 MB/秒。

mbps_rd = <mbps>

最大读操作速度，单位为 MB/秒。

mbps_rd_max = <mbps>

最大无限制读操作速度，单位为 MB/秒。

mbps_wr = <mbps>

最大写操作速度，单位为 MB/秒。

mbps_wr_max = <mbps>

最大无限制写操作速度，单位为 MB/秒。

media = <cdrom | disk> (default = disk)

虚拟硬盘驱动器介质类型。

error = <ignore | report | stop>

读错误处理方式。

secs = <integer>

强制指定虚拟硬盘物理几何参数中的 sector 值。

serial = <serial>

虚拟硬盘的序列号，基于 url 编码格式，最大 20 字节。

size = <DiskSize>

虚拟硬盘容量。仅供显示使用，并不能影响实际容量大小。

snapshot = <boolean>

设置虚拟硬盘在创建虚拟机快照时是否被纳入快照范围。

trans = <auto | lba | none>

设置虚拟硬盘几何参数地址 bios 解释模式。

werror = <enospc | ignore | report | stop>

写错误处理方式。

- watchdog : [[model=]<i6300esb|ib700>] [,action=<enum>]

创建虚拟看门狗设备。一旦启用（由虚拟机），虚拟机必须定期重置看门狗，否则虚拟机将自动重启（或执行指定的操作）。

action = <debug | none | pause | poweroff | reset | shutdown>

看门狗超时后所要进行的操作。

```
model = <i6300esb | ib700> (default = i6300esb )
```

虚拟看门狗类型。

8.6 锁

在线迁移，创建快照和备份（vzdump），Proxmox VE 都会对虚拟机加锁，以防止对虚拟机不恰当的并发操作。有时，你需要手工移除锁（例如，意外断电）。命令如下：

```
qm unlock <vmid>
```

☒ 警告

执行该操作前，务必确保设置锁的操作已经停止运行。
