

# Lpi 305-300日本語版問題解説 & 305-300合格対策



P.S. ShikenPASSがGoogle Driveで共有している無料かつ新しい305-300ダンプ: <https://drive.google.com/open?id=1ay4lIzqgVqxuYw6vQkpXDEvQ244J0irt>

ShikenPASS Lpiの305-300試験問題集は実践の検査に合格しますから、広い研究と実際に基づいている経験を提供できます。ShikenPASSはIT領域の10年以上の認定経験を持っていますから、問題と解答に含まれています。305-300試験に準備するためにインターネットで色々なトレーニングツールを見つけることができますが、ShikenPASSの305-300試験資料は最も良いトレーニング資料です。、弊社は最全面的な認証試験問題と解答を提供するだけでなく、一年間の無料更新サービスも提供いたします。

もちろん、資格試験を審査するとき、非公開にすることはできません。テスト305-300認定に関連する新しいポリシーと情報に注意する必要があります。ユーザーの便宜を図るため、ホームページで305-300テスト資料を更新し、資格試験に関連する情報をタイムリーに更新します。年次認定試験は、内容はほぼ同じですが、各年のポリシーとして、対応する試験パターンのグレーディング基準とホットスポットが変更されます。305-300テスト準備は、ユーザーが最短時間で合格するのに役立ちます。試験。

>> Lpi 305-300日本語版問題解説 <<

## 305-300試験の準備方法 | 権威のある305-300日本語版問題解説試験 | 完璧なLPIC-3 Exam 305: Virtualization and Containerization合格対策

我々は305-300試験を準備しているあなたに便利をもたらすために、PDF版、ソフト版、オンライン版の3つの異なるバージョンを提供しています。PDF版の305-300問題集を利用したら、紙でプリントすることができて読みやすいです。ソフト版であなたは試験の環境で305-300模擬試験をすることができて複数のパソコンで使用することができます。また、オンライン版を通して、どの電子製品でも使うことができ、オンライン版の機能はソフト版のと大体同じです。

Lpi 305-300試験は、Linuxエキスパート向けのプロフェッショナル認定であるLPIC-3認定プログラムの一部で

す。この試験は、仮想化およびコンテナ化に焦点を当てており、どのLinuxプロフェッショナルにとっても必須のスキルです。LPIC-3認定プログラムは、Linux管理における高度なスキルと知識を検証するために設計されており、305-300試験に合格することは、この認定を取得するための要件の1つです。

Lpiの305-300試験に向けて準備するには、Linuxシステム管理および仮想化およびコンテナ化技術についての堅実な理解が必要です。候補者は、コマンドラインインターフェースでの作業に慣れており、Linuxベースの仮想マシンおよびコンテナの管理の経験を持っている必要があります。候補者は、公式Lpi試験目標、練習問題、および学習ガイドなどのリソースを使用して自習するか、Lpiの認定トレーニングパートナーが提供するトレーニングコースを通じて試験に向けて準備することができます。この試験に合格することは、仮想化およびコンテナ化技術に高度な専門知識を持っていることを示すため、Linuxベースの仮想化およびコンテナ化技術と一緒に働くITプロフェッショナルにとって貴重な認定資格です。

## Lpi LPIC-3 Exam 305: Virtualization and Containerization 認定 305-300 試験問題 (Q61-Q66):

### 質問 # 61

Which of the following commands moves the libvirt domain web1 from the current host system to the host system host2?

- A. `virsh node-update host1=-domrweb1 host2=+domrweb1`
- B. `virsh pool-add host2 web1`
- C. `virsh patch web1 .Domain.Node=host2`
- D. `virsh cp .:web1 host2:web1`
- E. `virsh migrate web1 qemu+ssh://host2/system`

正解: E

解説:

Explanation

The correct command to move the libvirt domain web1 from the current host system to the host system host2 is `virsh migrate web1 qemu+ssh://host2/system`. This command uses the `virsh migrate` command, which initiates the live migration of a domain to another host1. The first argument is the name of the domain to migrate, which in this case is web1. The second argument is the destination URI, which specifies the connection to the remote host and the hypervisor to use2. In this case, the destination URI is `qemu+ssh://host2/system`, which means to use the QEMU driver and connect to host2 via SSH, and use the system instance of libvirt3. The other options are incorrect because they either use invalid commands or arguments, such as `node-update`, `pool-add`, `patch`, or `cp`, or they do not specify the destination URI correctly.

References:

<https://balamuruhan.github.io/2019/01/09/kvm-migration-with-libvirt.html>

<http://libvirt.org/migration.html>

### 質問 # 62

What is QEMU primarily used for?

- A. Managing virtual machines
- B. Monitoring virtualized environments
- C. Creating virtual disk images
- D. Emulating hardware to run virtual machines

正解: D

解説:

QEMU is primarily used for hardware emulation, enabling virtual machines to run by emulating CPU, memory, storage, and peripheral devices. Virtualization documentation describes QEMU as a generic and open-source machine emulator and virtualizer. When used alone, QEMU can fully emulate hardware, allowing operating systems compiled for one architecture to run on another. In modern Linux virtualization environments, QEMU is most commonly paired with KVM (Kernel-based Virtual Machine). In this configuration, KVM provides CPU virtualization using hardware extensions, while QEMU supplies device emulation and virtual hardware components. This division of responsibility allows near-native performance while maintaining flexibility.

Although QEMU includes utilities such as `qemu-img` for disk image creation, that is not its primary purpose.

VM management is handled by tools like `libvirt`, `virsh`, and `virt-manager`, and monitoring is performed by separate observability tools. Virtualization notes consistently emphasize that QEMU's core function is hardware emulation, making option D the correct answer.

### 質問 # 63

Which of the following values are valid in the `firmware` attribute of a `<os>` element in a libvirt domain definition? (Choose two.)

- A. `pcie`
- B. `efi`
- C. `virtio`
- D. `bios`
- E. `scsi`

正解: B、D

解説:

The `firmware` attribute of the `<os>` element in a libvirt domain definition specifies the type of firmware used to boot the virtual machine. The valid values for this attribute are `efi` and `bios`, which correspond to the Extensible Firmware Interface (EFI) and the Basic Input/Output System (BIOS) respectively. EFI is a newer standard that supports more features and security than BIOS, such as Secure Boot and faster boot times. BIOS is an older standard that is widely compatible and supported by most hypervisors and operating systems. The other values, `scsi`, `virtio`, and `pcie`, are not related to firmware, but to different types of devices or drivers that can be used in a virtual machine. References: 1 (search for `firmware` enum)

### 質問 # 64

Which value must be set in the option builder in the configuration file of a Xen guest domain in order to create a fully virtualized machine instead of a paravirtualized one?

正解:

解説:

`builder = "hvm"`

Explanation:

In Xen virtualization, guest domains can be created using either paravirtualization (PV) or hardware-assisted full virtualization, also known as HVM (Hardware Virtual Machine). According to official Xen documentation, the `builder` option in a Xen guest domain configuration file determines the type of virtualization used when creating a virtual machine.

To create a fully virtualized (HVM) guest, the configuration file must explicitly set the value of the `builder` option to `"hvm"`:

`builder = "hvm"`

This instructs the Xen toolstack to use hardware-assisted virtualization features provided by the CPU, such as Intel VT-x or AMD-V, allowing unmodified guest operating systems to run. Fully virtualized guests behave like physical machines and can run operating systems that are unaware of Xen.

If the `builder` option is omitted or set to the default paravirtualized builder, Xen expects a PV-capable kernel, which requires guest operating system modifications. HVM guests, by contrast, rely on device emulation and hardware virtualization support.

Virtualization documentation clearly distinguishes between PV and HVM guests and identifies `builder =`

`"hvm"` as the required configuration for full virtualization. This setting is fundamental when deploying operating systems such as Windows or standard Linux distributions without Xen-specific kernels.

Therefore, the correct and documented value is `builder = "hvm"`.

### 質問 # 65

What is the name of the kernel module that is required to be loaded in order to use KVM on an Intel CPU architecture? (Specify ONLY the module name without any path information and with or without the module suffix.) Solution:

`kvm-intel.ko` - or - `kvm-intel` - or - `kvm_intel.ko` - or - `kvm_intel`

Determine whether the given solution is correct?

- A. Incorrect
- B. Correct

正解: B

解説:

Kernel-based Virtual Machine (KVM) relies on hardware-assisted virtualization features provided by modern CPUs. On Intel CPU architectures, this support is enabled through the `kvm_intel` kernel module. Official KVM and Linux virtualization documentation

