

PDFVCE



Choose the version that fits your needs	PDF Version	Desktop Test Engine	Online Test Engine
Latest and Up-to-Date exam dumps with real exam questions answers.	✓	✓	✓
Get 12-Months free updates without any extra charges.	✓	✓	✓
Experience same exam environment before appearing in the certification exam.	✗	✓	✓
100% exam passing guarantee in the first attempt.	✓	✓	✓
20% discount on more than one license and 30% discount on 5+ license purchases.	✗	✓	✓
100% secure purchase on SSL.	✓	✓	✓
Completely private purchase without sharing your personal info with anyone.	✓	✓	✓

<http://www.pdfvce.com>

Highly Efficiently Exam Tool and Effective Exam Practice Materials

Exam : **305-300J**

Title : LPIC-3 Exam 305:
Virtualization and
Containerization (305-
300日本語版)

Vendor : Lpi

Version : DEMO

QUESTION NO: 1

Xen は次のタイプのゲスト システムのうちどれをサポートしますか？

(2つお選びください。)

- A. 外国建築ゲスト (FA)
- B. 準仮想化クエスト (PVI)
- C. エミュレートされたゲスト
- D. コンテナ仮想化ゲスト
- E. 完全に仮想化されたゲスト

Answer: B E

Explanation:

Xen supports two types of guest systems: paravirtualized guests (PV) and fully virtualized guests (HVM).

* Paravirtualized guests (PV) are guests that have been modified to run on the Xen hypervisor. They use a special kernel that communicates with the hypervisor through hypercalls, and use paravirtualized drivers for I/O devices. PV guests can run faster and more efficiently than HVM guests, but they require the guest operating system to be ported to Xen and to support the Xen ABI¹².

* Fully virtualized guests (HVM) are guests that run unmodified operating systems on the Xen hypervisor. They use hardware virtualization extensions, such as Intel VT-x or AMD-V, to create a virtual platform for the guest. HVM guests can run any operating system that supports the hardware architecture, but they incur more overhead and performance penalties than PV guests. HVM guests can also use paravirtualized drivers for I/O devices to improve their performance¹².

The other options are not correct. Xen does not support foreign architecture guests (FA), emulated guests, or container virtualized guests.

* Foreign architecture guests (FA) are guests that run on a different hardware architecture than the host.

For example, running an ARM guest on an x86 host. Xen does not support this type of virtualization, as it would require emulation or binary translation, which are very complex and slow techniques³.

* Emulated guests are guests that run on a software emulator that mimics the hardware of the host or another platform. For example, running a Windows guest on a QEMU emulator. Xen does not support this type of virtualization, as it relies on the emulator to provide the virtual platform, not the hypervisor. Xen can use QEMU to emulate some devices for HVM guests, but not the entire platform¹⁴.

* Container virtualized guests are guests that run on a shared kernel with the host and other guests, using namespaces and cgroups to isolate them. For example, running a Linux guest on a Docker container. Xen does not support this type of virtualization, as it requires the guest operating system to be compatible with the host kernel, and does not provide the same level of isolation and security as hypervisor-based virtualization⁵⁶.

:

Xen Project Software Overview - Xen

Xen ARM with Virtualization Extensions - Xen

Xen Project Beginners Guide - Xen

QEMU - Xen

Docker overview | Docker Documentation

What is a Container? | App Containerization | VMware

QUESTION NO: 2

イメージ構築に Packer を使用する利点は何ですか？

(該当するものをすべて選択してください)

- A. 組み込みの監視とログ記録
- B. 自動パッチ適用によるセキュリティの向上
- C. 仮想マシンのプロビジョニングの高速化
- D. 異なる環境間でのマシンイメージの一貫性

Answer: C D

Explanation:

Packer is designed to automate machine image creation, which leads to faster provisioning of virtual machines (C) because prebuilt images eliminate repetitive configuration steps at deployment time. Another primary benefit is consistency across environments (D), ensuring that development, testing, and production systems are built from the same image.

Packer does not provide built-in monitoring, and while it can apply patches during image creation, automated patching is not its primary or guaranteed function.

Thus, the correct answers are C and D.

QUESTION NO: 3

VirtualBox に関して正しいのは次のどれですか？

A. これは、最初の通常のオペレーティング

システムが起動する前に起動される特別なカーネルとして設計されたハイパーバイザーです。

B. ゲスト オペレーティング システムとして Linux のみをサポートし、仮想マシン内で Windows を実行することはできません。

C. 仮想マシンのディスク イメージを仮想化ホストのブロック

デバイスにローカルに保存できないため、専用の共有ストレージが必要です。

D. 仮想マシンを管理するためのグラフィカル ユーザー インターフェイスとコマンド ライン ツールの両方を提供します。

E. Linux でのみ使用でき、現在実行中の Linux カーネルのソース コードが使用可能である必要があります。

Answer: D

Explanation:

VirtualBox is a hosted hypervisor, which means it runs as an application on top of an existing operating system, not as a special kernel that is booted before the first regular operating system starts¹. VirtualBox supports a large number of guest operating systems, including Windows, Linux, Solaris, OS/2, and OpenBSD¹. VirtualBox does not require dedicated shared storage, as it can store virtual machine disk images locally on block devices of the virtualization host, or on network shares, or on iSCSI targets¹. VirtualBox provides both a graphical user interface (GUI) and command line tools (VBoxManage) to administer virtual machines¹. VirtualBox is available for Windows, Linux, macOS, and Solaris hosts¹, and does not require the source code of the currently running Linux kernel to be available. References:

* Oracle VM VirtualBox: Features Overview

QUESTION NO: 4

次のコマンドを 2 回連続して実行するとどうなりますか？

```
docker run -tid -v data:/data debian bash
```

- A. 2 回目の呼び出しで生成されたコンテナは、/data/の内容を読み取ることのみが可能で、変更することはできません。
- B. 各コンテナには独自の独立したデータ ボリュームが装備されており、各コンテナの /data/で利用できます。
- C. 両方のコンテナはデータ ボリュームのコンテンツを共有し、そのコンテンツを変更するための完全な権限を持ち、それぞれの変更を相互に確認します。
- D. コンテナ イメージデータの元のコンテンツは両方のコンテナで利用できますが、変更は各コンテナ内でローカルに残ります。
- E. 2 番目のコマンド呼び出しは、ボリュームデータが実行中のコンテナにすでに関連付けられていることを示すエラーで失敗します。

Answer: C

Explanation:

The command `docker run -tid -v data:/data debian bash` creates and runs a new container from the `debian` image, with an interactive terminal and a detached mode, and mounts a named volume `data` at `/data` in the container^{1,2}. If the volume `data` does not exist, it is created automatically³. If the command is executed twice in succession, two containers are created and run, each with its own terminal and process ID, but they share the same volume data. This means that both containers can access, modify, and see the contents of the `data` volume, and any changes made by one container are reflected in the other container. Therefore, the statement C is true and the correct answer. The statements A, B, D, and E are false and incorrect, as they do not describe the behavior of the command or the volume correctly. References:

* 1: `docker run` | Docker Docs.

* 2: Docker run reference | Docker Docs - Docker Documentation.

* 3: Use volumes | Docker Documentation.

* [4]: How to Use Docker Run Command with Examples - phoenixNAP.

QUESTION NO: 5

LXC について正しいのは次のうちどれですか？

- A. Windowsのみ対応
- B. Linuxカーネルは使用しません
- C. コンテナランタイムです
- D. 主に仮想マシンの管理に使用されます

Answer: C

Explanation:

LXC (Linux Containers) is a container runtime that uses Linux kernel features such as namespaces and cgroups to provide lightweight, isolated environments. According to

containerization documentation, LXC allows multiple Linux systems (containers) to run on a single host while sharing the same kernel.

LXC is Linux-specific, uses the Linux kernel directly, and is not a virtual machine manager. Therefore, the correct answer is C.

QUESTION NO: 6

パッカーとは何ですか？

- A. コンテナオーケストレーションツール
- B. 画像構築ツール
- C. 構成管理ツール
- D. 仮想化プラットフォーム

Answer: B

Explanation:

Packer is an image automation and building tool developed by HashiCorp. According to virtualization and DevOps documentation, Packer is used to create machine images for multiple platforms from a single configuration file.

Packer is not an orchestration platform, configuration management tool, or hypervisor. Its primary role is to ensure consistent, repeatable, and automated image creation, making B the correct answer.

QUESTION NO: 7

KVM

仮想化ホストをセットアップする場合、次のコンポーネントのうちどれが必須ではありませんか？

- A. ブリッジユーティリティ
- B. kvm カーネルモジュール
- C. qemu
- D. libvirt
- E. virsh

Answer: E

Explanation:

When configuring a KVM-based virtualization host, several core components are mandatory to enable and manage virtual machines. According to KVM and virtualization documentation, KVM kernel modules are essential because they provide hardware-assisted virtualization support within the Linux kernel. QEMU is required to perform hardware emulation and manage virtual machine execution. Libvirt acts as the virtualization management API, enabling centralized and secure control of virtual machines. Additionally, bridge-utils is commonly required to configure network bridges, allowing virtual machines to communicate with external networks.

However, virsh is not strictly required. Virsh is a command-line utility that interacts with libvirt to manage virtual machines, but it is only a management interface, not a core dependency.

Virtual machines can still be created and managed using alternative tools such as virt-manager, Ansible, OpenStack, or custom API-based solutions without virsh being installed. Virtualization documentation clearly distinguishes between essential backend

components(KVM, QEMU, libvirt) and optional management tools(virsh). Therefore, while virsh is widely used and highly recommended for administrative convenience, it is not a mandatory component for a functional KVM virtualization host.

QUESTION NO: 8

次のツールのうち、VMのプロビジョニングとイメージの作成に通常使用されないものはどれですか。

- A. Kubernetes
- B. 放浪者
- C. パッカー
- D. クラウド初期化

Answer: A

Explanation:

VM provisioning and image creation tools are used to build, configure, and initialize virtual machine images. According to virtualization documentation, Packer is specifically designed for automated image creation, Vagrant is used for provisioning reproducible development environments, and cloud-init is used to configure instances at first boot.

Kubernetes, however, is a container orchestration platform, not a VM provisioning or image creation tool.

It manages containerized workloads after infrastructure has already been provisioned.

Documentation clearly separates infrastructure provisioning tools from application orchestration platforms

. As such, Kubernetes is not typically used for VM provisioning or image creation.

Therefore, the correct answer is A.

QUESTION NO: 9

コマンド `virsh vol-list vms` は次のエラーを返します。

エラー: プール「vms」の取得に失敗しました

エラー: ストレージ プールが見つかりません: 一致する名前 'vms' のストレージ プールがありません ディレクトリ /vms exists

の場合、次のコマンドのどれがこの問題を解決しますか?

- A. `dd if=/dev/zero of=/vms bs=1 count=0 flags=name:vms`
- B. `libvirt-poolctl new --name=/vms --type=dir --path=/vms`
- C. `qemu-img` プール `vms:/vms`
- D. `virsh pool-create-as vms dir --target /vms`
- E. `/vms/.libvirtpool` をタッチします

Answer: D

Explanation:

The command `virsh pool-create-as vms dir --target /vms` creates and starts a transient storage pool named `vms` of type `dir` with the target directory `/vms`. This command resolves the issue of the storage pool not found error, as it makes the existing directory `/vms` visible to libvirt as a storage pool. The other commands are invalid because:

* `dd if=/dev/zero of=/vms bs=1 count=0 flags=name:vms` is not a valid command syntax. The `dd` command does not take a `flags` argument, and the output file `/vms` should be a regular

file, not a directory³.

* `libvirt-poolctl new --name=/vms --type=dir --path=/vms` is not a valid command name. There is no such command as `libvirt-poolctl` in the `libvirt` package⁴.

* `qemu-img pool vms:/vms` is not a valid command syntax. The `qemu-img` command does not have a `pool` subcommand, and the `vms:/vms` argument is not a valid image specification⁵.

* `touch /vms/.libvirtpool` is not a valid command to create a storage pool. The `touch` command only creates an empty file, and the `.libvirtpool` file is not recognized by `libvirt` as a storage pool configuration file⁶.

References:

1: `virsh` - difference between `pool-define-as` and `pool-create-as` - Stack Overflow

2: `dd(1)` - Linux manual page - man7.org

3: 12.3.3. Creating a Directory-based Storage Pool with `virsh` - Red Hat Customer Portal

4: `libvirt` - Linux Man Pages (3)

5: `qemu-img(1)` - Linux manual page - man7.org

6: `touch(1)` - Linux manual page - man7.org

QUESTION NO: 10

同じ Linux ベースのホスト

システム上で実行されている複数の仮想マシン間でデータを共有するにはどうすればよいですか？

A. 同じホスト システム上のすべての仮想マシンが同じファイル

システムを使用するため、データをファイル システムに書き込むことによって行われます。

B. 他の仮想マシンのファイル システムを `/dev/virt-disks/remote/` からマウントする。

C. 1 つの仮想マシンに RAM ディスクをセットアップし、その UUID を使用して他の VM にマウントします。

D. ネットワーク ファイル システムまたはファイル転送プロトコルを使用します。

E. 同じ仮想ハードディスクをすべての仮想マシンに接続し、その仮想マシン上で EXT4 共有拡張機能をアクティブ化します。

Answer: D

Explanation:

The correct way to share data between several virtual machines running on the same Linux-based host system is by using a network file system or file transfer protocol. A network file system (NFS) is a distributed file system protocol that allows a user on a client computer to access files over a network in a manner similar to how local storage is accessed¹. A file transfer protocol (FTP) is a standard network protocol used for the transfer of computer files between a client and server on a computer network². Both methods allow data to be shared between virtual machines regardless of their underlying file systems or virtualization technologies. The other options are incorrect because they either do not work or are not feasible. Option A is wrong because each virtual machine has its own file system that is not directly accessible by other virtual machines. Option B is wrong because there is no such device as `/dev/virt-disks/remote/` that can be used to mount other virtual machines' file systems. Option C is wrong because a `ramdisk` is a volatile storage device that is not suitable for sharing data between virtual machines. Option E is wrong because attaching the same virtual hard disk to multiple virtual machines can cause data corruption and conflicts, and

EXT4 does not have any sharing extensions that can prevent this.

References:<https://kb.vmware.com/s/article/1012706>

<https://bing.com/search?q=data+sharing+between+virtual+machines>

QUESTION NO: 11

Kubernetes の thekubletservice の目的は何ですか？

- A. Kubernetes を管理するためのコマンドライン インターフェイスを提供します。
- B. Dockerfile で指定されたとおりにコンテナ イメージをビルドします。
- C. Kubernetes API を操作する際のユーザーの権限を管理します。
- D. Kubernetes 構成に従ってワーカー ノード上でコンテナを実行します。
- E. Kubernetes 構成データを保存および複製します。

Answer: D

Explanation:

The purpose of the kubelet service in Kubernetes is to run containers on the worker nodes according to the Kubernetes configuration. The kubelet is an agent or program that runs on each node and communicates with the Kubernetes control plane. It receives a set of PodSpecs that describe the desired state of the pods that should be running on the node, and ensures that the containers described in those PodSpecs are running and healthy. The kubelet also reports the status of the node and the pods back to the control plane. The kubelet does not manage containers that were not created by Kubernetes. References:

* Kubernetes Docs - kubelet

* Learn Steps - What is kubelet and what it does: Basics on Kubernetes

QUESTION NO: 12

XenStore と対話するために使用されるツールは次のどれですか？

- A. xendo
- B. oxs
- C. xstore
- D. xentore-ls
- E. xl store

Answer: D

Explanation:

XenStore is a shared configuration and state database used by Xen domains. According to Xen documentation, tools such as xentore-ls, xentore-read, and xentore-write are used to interact directly with XenStore.

Although option D contains a minor typographical error ("xentore-ls" instead of xentore-ls), it clearly refers to the correct and documented utility. None of the other options represent standard XenStore interaction tools.

Therefore, despite the typo, D is the correct answer.