Lpi.305-300.v2026-04-16.q56

Exam Code:	305-300
Exam Name:	LPIC-3 Exam 305: Virtualization and Containerization
Certification Provider:	Lpi
Free Question Number:	56
Version:	v2026-04-16
# of views:	130
# of Questions views:	560
https://www.freepdfdumps.com/Lpi.305-300.v2026-04-16.q56.html	

NEW QUESTION: 1
Which of the following statements are true about container-based virtualization? (Choose two.)
A. Each container runs its own operating system kernel.
B. Different containers may use different distributions of the same operating system.
C. Container-based virtualization relies on hardware support from the host system's CPU.
D. All containers run within the operating system kernel of the host system.
E. Linux does not support container-based virtualization because of missing kernel APIs.
Answer: B,D (LEAVE A REPLY)
Container-based virtualization is a method of operating system-level virtualization that allows multiple isolated user spaces (containers) to run on the same host system1. Each container shares the same operating system kernel as the host, but has its own file system, libraries, and processes2. Therefore, the statements A and C are false, as containers do not run their own kernels or rely on hardware support from the CPU. The statement E is also false, as Linux does support container-based virtualization through various technologies, such as cgroups, namespaces, LXC, Docker, etc12. The statement B is true, as different containers may use different distributions of the same operating system, such as Debian, Ubuntu, Fedora, etc., as long as they are compatible with the host kernel3. The statement D is also true, as all containers run within the operating system kernel of the host system, which provides isolation and resource management for them12. References:
* 1: Containerization (computing) - Wikipedia.
* 2: What are containers? | Google Cloud.
* 3: What is Container-Based Virtualization? - StackHowTo.

NEW QUESTION: 2
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. Correct
B. Incorrect
Answer: (SHOW ANSWER)
Kernel-based Virtual Machine (KVM) relies on hardware-assisted virtualization features provided by modern CPUs. OnIntel CPU architectures, this support is enabled through thekvm_intelkernel module. Official KVM and Linux virtualization documentation clearly states that the required module for Intel processors is namedkvm_intel, with the optional .ko suffix when referring to the kernel object file.
The provided solution lists multiple acceptable representations of the module name, includingkvm_inteland kvm_intel.ko, both of which are valid and correct. Linux kernel module naming conventions allow the module to be referenced with or without the .ko suffix when loading it using tools such as modprobe or lsmod.
Although variants using a hyphen (kvm-intel) are not the canonical kernel module name, the solution explicitly includes the correct and documented module name. Therefore, the solutioncorrectly identifies the required kernel modulefor enabling KVM on Intel CPUs.
Virtualization documentation emphasizes that KVM functionality requires both the generic kvm module and the CPU-specific module (kvm_intel for Intel or kvm_amd for AMD). Hence, the determination that the solution iscorrectaligns with verified documentation.

NEW QUESTION: 3
How does cloud-init enhance the customization of cloud instances?
A. It automatically configures all aspects of the cloud instance
B. It allows you to specify scripts and user data
C. It integrates with popular containerization tools
D. It provides a GUI for configuration
Answer: B (LEAVE A REPLY)
Cloud-init enhances cloud instance customization by allowing users to provideuser data and scriptsthat are executed during thefirst bootof an instance. According to cloud-init documentation, these scripts and configuration files enable tasks such as installing packages, configuring services, setting hostnames, managing users, and injecting SSH keys.
Cloud-init does not automatically configure every aspect of an instance, nor does it provide a graphical interface. It also does not directly integrate with containerization tools. Its strength lies inautomation and repeatability, making instance initialization consistent across environments.
Thus, the correct answer isB.

NEW QUESTION: 4
FILL BLANK

What is the default path to the Docker daemon configuration file on Linux? (Specify the full name of the file, Including path.)

Answer:

/etc/docker/daemon.json

Explanation:

The default path to the Docker daemon configuration file on Linux is /etc/docker/daemon.json. This file is a JSON file that contains the settings and options for the Docker daemon, which is the service that runs on the host operating system and manages the containers, images, networks, and other Docker resources. The /etc
/docker/daemon.json file does not exist by default, but it can be created by the user to customize the Docker daemon behavior. The file can also be specified by using the --config-file flag when starting the Docker daemon. The file must be a valid JSON object and follow the syntax and structure of the dockerd reference docs12. References:

* Docker daemon configuration file - Medium3
* Docker daemon configuration overview | Docker Docs4
* docker daemon | Docker Docs5

NEW QUESTION: 5

Which of the following statements are true of full virtualization? (Select THREE correct answers)

A. Full virtualization is faster than paravirtualization

B. Full virtualization requires no modification to the Guest OS kernel

C. Full virtualization works through CPU emulation

D. Full virtualization has superior I/O performance through emulated device drivers

E. Full virtualization requires time and system resources for emulation

Answer: B,C,E (LEAVE A REPLY)

Full virtualization allowsunmodified guest operating systemsto run in virtual machines, making statementB correct. The hypervisor presents a complete hardware abstraction to the guest OS. Historically and conceptually, full virtualization relies onCPU emulation or hardware-assisted virtualization, making statementCcorrect. Even with hardware extensions, some degree of emulation is involved.

Full virtualization alsorequires additional system resources and processing timeto emulate hardware and manage isolation, making statementEcorrect.

Statement A is incorrect because paravirtualization often offers better performance. Statement D is incorrect because emulated I/O devices generally perform worse than paravirtualized drivers. Therefore, the correct answers areB, C, and E.

NEW QUESTION: 6

Which of the following services can QEMU provide in a user network? (Choose three.)

A. DHCP

B. BGP

C. CIFS

D. AppleTalk
E. TFTP
Answer: A,E (LEAVE A REPLY)
QEMU can provide some network services in a user network, which is a mode of networking that does not require any administrator privilege to run. The user network uses the SLIRP TCP/IP emulator to create a virtual NAT'ted subnet, with a DHCP server started by QEMU that gives out IP addresses to the guest machines and puts the host on 10.0.2.21. QEMU can also provide a TFTP server in the user network, which can be used to boot the guest machines from a network image. The TFTP server can be configured with the - tftp option2. QEMU does not provide BGP, CIFS, or AppleTalk services in the user network. BGP is a routing protocol that is used to exchange routing information between autonomous systems on the Internet3. CIFS is a file-sharing protocol that is used to access files and printers on a network4. AppleTalk is a deprecated network protocol suite that was used by Apple devices5. These services require more advanced networking features than the user network can offer, such as bridging, routing, or tunneling.
:
Documentation/Networking - QEMU
QEMU/Networking - Wikibooks, open books for an open world
Border Gateway Protocol - Wikipedia
Common Internet File System - Wikipedia
AppleTalk - Wikipedia

NEW QUESTION: 7
Which sub-command of xl changes the media inside a virtual CD-ROM drive of a Xen guest domain?
(Specify ONLY the sub-command without any path or parameters.)
Solution:
block-attach - or - xl cd-insert - or - cd-eject - or - xl cd-eject - or - cd-insert - or - xl block-attach - or - block-detach - or - xl block-detach Determine whether the given solution is correct?
A. Correct
B. Incorrect
Answer: B (LEAVE A REPLY)
According to official Xen xl toolstack documentation, thecorrect sub-commandsused to change media in a virtual CD-ROM drive arecd-insertandcd-eject. These commands are specifically designed for inserting or ejecting ISO media from a Xen guest's virtual CD-ROM device.
While commands such as block-attach and block-detach are valid xl sub-commands, they are used for attaching and detaching block devices in general and arenot specific to CD-ROM media operations. The provided solution lists multiple commands, including incorrect and unrelated ones, rather than specifying the correct sub-command precisely.
Because the solution does not accurately and exclusively identify the correct xl sub-command, it is considered incorrect. Therefore, the correct determination isB.

NEW QUESTION: 8
Which file format is used by libvirt to store configuration data?
A. INI-style text files
B. SQLite databases
C. XML files
D. Java-like properties files
E. Text files containing key/value pairs
Answer: C (LEAVE A REPLY)
Libvirt uses XML files to store configuration data for objects in the libvirt API, such as domains, networks, storage, etc. This allows for ease of extension in future releases and validation of documents prior to usage.
Libvirt does not use any of the other file formats listed in the question. References:
* libvirt: XML Format
* LPIC-3 Virtualization and Containerization: Topic 305.1: Virtualization Concepts and Theory

NEW QUESTION: 9
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?
Answer:
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".

NEW QUESTION: 10
What types of cloud-init data sources can be used to configure cloud instances? (Select all that apply)
A. Cloud-based configuration management tools
B. Metadata service
C. QR codes
D. User data scripts
Answer: B,D (LEAVE A REPLY)
Cloud-init is a widely used initialization system for cloud instances, allowing configuration at first boot.
According to cloud-init documentation, the two primary data sources used to configure instances aremetadata servicesanduser data scripts.
Ametadata serviceprovides instance-specific information such as hostname, instance ID, networking details, and SSH keys.User data scriptsallow administrators to supply custom configuration instructions, commonly written as shell scripts or cloud-config YAML.
Cloud-based configuration management tools like Ansible or Puppet are not cloud-init data sources; instead, cloud-init may install or trigger them.QR codesare not supported data sources in cloud-init.
Virtualization and cloud documentation clearly identifies metadata and user data as the supported configuration mechanisms. Therefore, the correct answers areB and D.

NEW QUESTION: 11
What is the purpose ofcloud-init?
A. Replace common Linux inic systems, such as systemd or SysV init.
B. Assign an IaaS instance to a specific computing node within a cloud.
C. Standardize the configuration of infrastructure services, such as load balancers or virtual firewalls in a cloud.
D. Orchestrate the creation and start of multiple related IaaS instances.
E. Prepare the generic image of an IaaS instance to fit a specific instance's configuration.
Answer: E (LEAVE A REPLY)
Cloud-init is a tool that processes configurations and runs through five stages during the initial boot of Linux VMs in a cloud. It allows users to customize a Linux VM as it boots for the first time, by applying user data to the instance. User data can include scripts, commands, packages, files, users, groups, SSH keys, and more.
Cloud-init can also interact with various cloud platforms and services, such as Azure, AWS, OpenStack, and others. The purpose of cloud-init is to prepare the generic image of an IaaS instance to fit a specific instance's configuration, such as hostname, network, security, and application settings. References:
* Cloud-init - The standard for customising cloud instances

* Understanding cloud-init - Azure Virtual Machines
* Tutorial - Customize a Linux VM with cloud-init in Azure - Azure Virtual Machines

NEW QUESTION: 12
Which of the following statements in aDockerfileleads to a container which outputs hello world? (Choose two.)
A. ENTRYPOINT "echo Hello World"
B. ENTRYPOINT ["echo hello world"]
C. ENTRYPOINT ["echo", "hello", "world"]
D. ENTRYPOINT echo Hello World
E. ENTRYPOINT "echo", "Hello", "World*
Answer: B,C (LEAVE A REPLY)
The ENTRYPOINT instruction in a Dockerfile specifies the default command to run when a container is started from the image. The ENTRYPOINT instruction can be written in two forms: exec form and shell form. The exec form uses a JSON array to specify the command and its arguments, such as ["executable",
"param1", "param2"]. The shell form uses a single string to specify the command and its arguments, such as
"executable param1 param2". The shell form is converted to the exec form by adding /bin/sh -c to the beginning of the command. Therefore, the following statements in a Dockerfile are equivalent and will lead to a container that outputs hello world:
ENTRYPOINT ["echo hello world"] ENTRYPOINT ["/bin/sh", "-c", "echo hello world"] ENTRYPOINT
"echo hello world" ENTRYPOINT ["echo", "hello", "world"] ENTRYPOINT ["/bin/sh", "-c", "echo", "hello", "world"] ENTRYPOINT "echo hello world"
The other statements in the question are invalid or incorrect. The statement A. ENTRYPOINT "echo Hello World" is invalid because it uses double quotes to enclose the entire command, which is not allowed in the shell form. The statement D. ENTRYPOINT echo Hello World is incorrect because it does not use quotes to enclose the command, which is required in the shell form. The statement E. ENTRYPOINT "echo", "Hello",
"World" is invalid because it uses double quotes to separate the command and its arguments, which is not allowed in the exec form. References:
* Dockerfile reference | Docker Docs
* Using the Dockerfile ENTRYPOINT and CMD Instructions - ATA Learning
* Difference Between run, cmd and entrypoint in a Dockerfile

NEW QUESTION: 13
Which of the following types of guest systems does Xen support? (Choose two.)
A. Foreign architecture guests (FA)
B. Paravirtualized quests (PVI
C. Emulated guests

D. Container virtualized guests
E. Fully virtualized guests
Answer: B,E (LEAVE A REPLY)
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

NEW QUESTION: 14

Which disk image formats are commonly used in Linux-based virtualization environments? (Select all that apply)

A. RAW

B. VMDK

C. QCOW2

D. VHD

Answer: (SHOW ANSWER)

Linux-based virtualization environments support a wide range of disk image formats to ensure compatibility with multiple hypervisors and cloud platforms. According to virtualization documentation,RAW, VMDK, QCOW2, and VHDare all commonly used formats.

RAWimages are simple, unstructured disk files that offer maximum performance due to minimal overhead.

QCOW2(QEMU Copy-On-Write version 2) is the most widely used format in KVM environments because it supports advanced features such as snapshots, thin provisioning, compression, and encryption.VMDKis the native disk format for VMware products but is frequently used in Linux environments for interoperability and migration purposes.VHDis commonly associated with Microsoft Hyper-V but is also supported by QEMU and cloud platforms.

Virtualization notes emphasize that modern Linux virtualization tools like QEMU and libvirt are designed to work across multiple disk formats. This flexibility enables administrators to migrate workloads between different hypervisors and cloud providers without rebuilding virtual machines. Therefore, all listed disk image formats are valid and commonly supported in Linux-based virtualization environments.

NEW QUESTION: 15

Which of the following tasks are part of a hypervisor's responsibility? (Choose two.)

A. Create filesystems during the installation of new virtual machine quest operating systems.

B. Provide host-wide unique PIDs to the processes running inside the virtual machines in order to ease inter-process communication between virtual machines.

C. Map the resources of virtual machines to the resources of the host system.

D. Manage authentication to network services running inside a virtual machine.

E. Isolate the virtual machines and prevent unauthorized access to resources of other virtual machines.

Answer: C,E (LEAVE A REPLY)

A hypervisor is a software that creates and runs virtual machines (VMs) by separating the operating system and resources from the physical hardware. One of the main tasks of a hypervisor is to map the resources of VMs to the resources of the host system, such as CPU, memory, disk, and network. This allows the hypervisor to allocate and manage the resources among multiple VMs and ensure that they run efficiently and independently123. Another important task of a hypervisor is to isolate the VMs and prevent unauthorized access to resources of other VMs. This ensures the security and privacy of the VMs and their data, as well as the stability and

performance of the host system. The hypervisor can use various techniques to isolate the VMs, such as virtual LANs, firewalls, encryption, and access control145.
The other tasks listed are not part of a hypervisor's responsibility, but rather of the guest operating system or the application running inside the VM. A hypervisor does not create filesystems during the installation of new VMs, as this is done by the installer of the guest operating system6. A hypervisor does not provide host-wide unique PIDs to the processes running inside the VMs, as this is done by the kernel of the guest operating system7. A hypervisor does not manage authentication to network services running inside a VM, as this is done by the network service itself or by a directory service such as LDAP or Active Directory8. References: 1 (search for "What is a hypervisor?"), 2 (search for "How does a hypervisor work?"), 3 (search for "The hypervisor gives each virtual machine the resources that have been allocated"), 4 (search for "Benefits of hypervisors"), 5 (search for "Isolate the virtual machines and prevent unauthorized access"), 6 (search for "Create filesystems during the installation of new virtual machine quest operating systems"), 7 (search for "Provide host-wide unique PIDs to the processes running inside the virtual machines"), 8 (search for "Manage authentication to network services running inside a virtual machine").

NEW QUESTION: 16
What is the primary purpose of Vagrant's "Vagrantfile" configuration file?
A. To provide a list of available software packages
B. To define the virtual machine's hardware requirements
C. To specify the developer's name and contact information
D. To specify cloud provider details
Answer: B (LEAVE A REPLY)

NEW QUESTION: 17
How can data be shared between several virtual machines running on the same Linux-based host system?
A. By writing data to the file system since all virtual machines on the same host system use the same file system.
B. By mounting other virtual machines' file systems from /dev/virt-disks/remote/.

C. By setting up a ramdisk in one virtual machine and mounting it using its UUID in the other VMs.
D. By using a network file system or file transfer protocol.
E. By attaching the same virtual hard disk to all virtual machines and activating EXT4 sharing extensions on it.
Answer: D (LEAVE A REPLY)
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

NEW QUESTION: 18
Virtualization of which hardware component is facilitated by CPUs supporting nested page table extensions, such as Intel Extended Page Table (EPT) or AMD Rapid Virtualization Indexing (RVI)?
A. Memory
B. Network Interfaces
C. Host Bus Adapters
D. Hard Disks
E. IO Cache
Answer: A (LEAVE A REPLY)
Nested page table extensions, such as Intel Extended Page Table (EPT) or AMD Rapid Virtualization Indexing (RVI), are hardware features that facilitate the virtualization of memory. They allow the CPU to perform the translation of guest virtual addresses to host physical addresses in a single step, without the need for software-managed shadow page tables. This reduces the overhead and complexity of memory management for virtual machines, and improves their performance and isolation. Nested page table extensions do not directly affect the virtualization of other hardware components, such as network interfaces, host bus adapters, hard disks, or IO cache.

:
Second Level Address Translation - Wikipedia
c - What is use of extended page table? - Stack Overflow
Hypervisor From Scratch - Part 4: Address Translation Using Extended ...

NEW QUESTION: 19
What is Docker Hub?
A. A Linux distribution
B. A container image registry
C. A container orchestration platform
D. A container runtime
Answer: B (LEAVE A REPLY)
Docker Hub is acloud-based container image registryused to store, manage, and distribute Docker container images. According to container documentation, it hosts bothofficial imagesand user-contributed images.
Docker Hub is not a Linux distribution, orchestration platform, or runtime. Therefore, the correct answer isB.

NEW QUESTION: 20
Which of the following is true about LXC?
A. It is only compatible with Windows
B. It does not use the Linux kernel
C. It is a container runtime
D. It is primarily used for virtual machine management
Answer: C (LEAVE A REPLY)
LXC (Linux Containers) is acontainer runtimethat uses Linux kernel features such asnamespaces and cgroupsto 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 isC.

NEW QUESTION: 21
What does LXC stand for?
A. Legacy Xenon Container
B. Linux Container
C. Linux Container Xenon
D. Lightweight Xenon Container
Answer: (SHOW ANSWER)
LXCstands forLinux Containers, a lightweight virtualization technology that enables multiple isolated Linux systems (containers) to run on a single Linux host. Virtualization and

containerization documentation describes LXC as acontainer-based virtualization solutionthat uses Linux kernel features such as namespaces, control groups (cgroups), and capabilitiesto provide process isolation without requiring a full hypervisor.
Unlike traditional virtual machines, LXC containers share the host kernel, which significantly reduces overhead and improves performance. Each container behaves like an independent Linux system with its own process tree, network interfaces, and filesystem, while still relying on the host kernel for execution.
LXC is often referred to as"system containers", as it allows running full Linux distributions inside containers, unlike application-focused containers such as Docker. It serves as the foundation for higher-level container platforms, includingLXD, which provides enhanced management features and APIs.
The other options are incorrect because they do not align with established Linux container terminology.
Official Linux container documentation consistently defines LXC asLinux Containers, making optionBthe correct answer.

NEW QUESTION: 22
In order to use the optiondom0_memto limit the amount of memory assigned to the Xen Domain-0, where must this option be specified?
A. In the bootloader configuration, when Xen is booted.
B. In any of Xen's global configuration files.
C. In its .config file, when the Domain-0 kernel is built.
D. In the configuration file /etc/xen/Domain-0.cfg, when Xen starts.
E. In its Makefile, when Xen is built.
Answer: (SHOW ANSWER)
The option dom0_mem is used to set the initial and maximum memory size of the Domain-0, which is the privileged domain that starts first and manages the unprivileged domains (DomU) in Xen. The option dom0_mem must be specified in the bootloader configuration, such as GRUB or GRUB2, when Xen is booted. This ensures that the Domain-0 kernel can allocate memory for storing memory metadata and network related parameters based on the boot time amount of memory. If the option dom0_mem is not specified in the bootloader configuration, the Domain-0 will use all the available memory on the host system by default, which may cause performance and security issues. References:
* Managing Xen Dom0#s CPU and Memory
* Xen Project Best Practices
* Dom0 Memory - Where It Has Not Gone

NEW QUESTION: 23
Which container orchestration platform is often associated with automated scaling and load balancing?
A. Docker Compose

B. Kubernetes
C. Amazon ECS
D. LXC
Answer: B (LEAVE A REPLY)
Kubernetesis the container orchestration platform most strongly associated withautomated scaling and load balancing. According to container orchestration documentation, Kubernetes provides built-in features such as Horizontal Pod Autoscaling, service-based load balancing, self-healing, and declarative deployment models.
Docker Compose and LXC are not orchestration platforms at scale, and while Amazon ECS supports scaling, Kubernetes is the industry-standard platform most commonly referenced for these capabilities.
Therefore, the correct answer isB.

NEW QUESTION: 24
Which of the following devices exist by default in an LXC container? (Choose three.)
A. /dev/log
B. /dev/console
C. /dev/urandom
D. /dev/kmem
E. /dev/root
Answer: (SHOW ANSWER)
LXC (Linux Containers) is a lightweight virtualization technology that allows multiple isolated Linux systems (containers) to run on the same host. LXC uses Linux kernel features such as namespaces, cgroups, and AppArmor to create and manage containers. Each container has its own file system, network interfaces, process tree, and resource limits. However, containers share the same kernel and hardware with the host, which makes them more efficient and faster than full virtualization.
By default, an LXC container has a minimal set of devices that are needed for its operation. These devices are created by the LXC library when the container is started, and are removed when the container is stopped. The default devices are:
* /dev/log: This is a Unix domain socket that connects to the syslog daemon on the host. It allows the container to send log messages to the host's system log1.
* /dev/console: This is a character device that provides access to the container's console. It is usually connected to the host's terminal or a file. It allows the container to interact with the user or the host's init system12.
* /dev/urandom: This is a character device that provides an unlimited source of pseudo-random numbers. It is used by various applications and libraries that need randomness, such as cryptography, UUID generation, and hashing13.
The other devices listed in the question do not exist by default in an LXC container. They are either not needed, not allowed, or not supported by the container's namespace or cgroup configuration. These devices are:

* /dev/kmem: This is a character device that provides access to the kernel's virtual memory. It is not needed by the container, as it can access its own memory through the /proc filesystem. It is also not allowed by the container, as it would expose the host's kernel memory and compromise its security4.
* /dev/root: This is a symbolic link that points to the root device of the system. It is not supported by the container, as it does not have a separate root device from the host. The container's root file system is mounted from a directory, an image file, or a loop device on the host5.
:
Linux Containers - LXC - Manpages - lxc.container.conf.5
Linux Containers - LXC - Getting started
Random number generation - Wikipedia
/dev/kmem - Wikipedia
Linux Containers - LXC - Manpages - lxc.container.conf.5

NEW QUESTION: 25
Which of the following mechanisms are used by LXC and Docker to create containers? (Choose three.)
A. Linux Capabilities
B. Kernel Namespaces
C. Control Groups
D. POSIXACLs
E. File System Permissions
Answer: A,B,C (LEAVE A REPLY)
LXC and Docker are both container technologies that use Linux kernel features to create isolated environments for running applications. The main mechanisms that they use are:
* Linux Capabilities: These are a set of privileges that can be assigned to processes to limit their access to certain system resources or operations. For example, a process with the CAP_NET_ADMIN capability can perform network administration tasks, such as creating or deleting network interfaces. Linux capabilities allow containers to run with reduced privileges, enhancing their security and isolation.
* Kernel Namespaces: These are a way of creating separate views of the system resources for different processes. For example, a process in a mount namespace can have a different file system layout than the host or other namespaces. Kernel namespaces allow containers to have their own network interfaces, process IDs, user IDs, and other resources, without interfering with the host or other containers.
* Control Groups: These are a way of grouping processes and applying resource limits and accounting to them. For example, a control group can limit the amount of CPU, memory, disk I/O, or network bandwidth that a process or a group of processes can use. Control groups allow containers to have a fair share of the system resources and prevent them from exhausting the host resources.

POSIX ACLs and file system permissions are not mechanisms used by LXC and Docker to create containers.
They are methods of controlling the access to files and directories on a file system, which can be applied to any process, not just containers.
:
LXC vs Docker: Which Container Platform Is Right for You?
LXC vs Docker: Why Docker is Better in 2023 | UpGuard
What is the Difference Between LXC, LXD and Docker Containers
lxc - Which container implementation docker is using - Unix & Linux Stack Exchange

NEW QUESTION: 26
Which of the following tasks is performed by Vagrant?
A. It automates the installation of a virtual machine according to a configuration file describing the desired VM
B. It monitors the functionality of a virtual machine and restarts the VM in case of failure
C. It is a programming interface used to create reports from collected performance data
D. It is a hypervisor optimized for embedded ARM systems
E. It migrates virtual machines automatically between host systems
Answer: (SHOW ANSWER)
Vagrant is a tool designed toautomate the creation and provisioning of development virtual machines using a declarative configuration file called aVagrantfile. According to virtualization documentation, Vagrant simplifies setting up reproducible environments by automatically installing and configuring virtual machines based on this file.
Vagrant does not perform monitoring, reporting, hypervisor functions, or automatic VM migration. It integrates with providers such as VirtualBox, KVM, and VMware but is not itself a hypervisor.
Therefore, the correct answer isA.

NEW QUESTION: 27
Which of the following statements are true regarding VirtualBox?
A. It is a hypervisor designed as a special kernel that is booted before the first regular operating system starts.
B. It only supports Linux as a guest operating system and cannot run Windows inside a virtual machine.
C. It requires dedicated shared storage, as it cannot store virtual machine disk images locally on block devices of the virtualization host.
D. It provides both a graphical user interface and command line tools to administer virtual machines.
E. It is available for Linux only and requires the source code of the currently running Linux kernel to be available.
Answer: D (LEAVE A REPLY)

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

NEW QUESTION: 28
Which cloud management tools are known for their infrastructure-as-code (IaC) approach? (Select all that apply)
A. Puppet
B. Terraform
C. AWS CloudFormation
D. Ansible
Answer: (SHOW ANSWER)
Infrastructure as Code (IaC) is an approach where infrastructure is defined and managed usingmachine- readable configuration files. According to DevOps and cloud documentation,Puppet, Terraform, AWS CloudFormation, and Ansibleall support IaC principles. Terraform and AWS CloudFormation are declarative IaC tools used to provision cloud infrastructure. Puppet and Ansible are configuration management and automation tools that also enable infrastructure definition through code.
All listed tools are widely recognized in IaC workflows, makingA, B, C, and Dcorrect.

NEW QUESTION: 29
What is a container image?
A. A lightweight virtual machine
B. An operating system kernel
C. A physical server
D. A snapshot of an application and its dependencies
Answer: D (LEAVE A REPLY)
Acontainer imageis astatic, immutable packagethat contains an application along with all its required dependencies, libraries, and configuration files. Containerization documentation defines a container image as a snapshot of an application and its runtime environment, which can be instantiated as a running container.
Container images do not include a full operating system kernel; instead, containers share the host system's kernel. This design makes containers lightweight and fast to deploy compared to traditional virtual machines.

Therefore, the correct answer isC.

NEW QUESTION: 30
Ifdocker stackis to be used to run a Docker Compose file on a Docker Swarm, how are the images referenced in the Docker Compose configuration made available on the Swarm nodes?
A. docker stack builds the images locally and copies them to only those Swarm nodes which run the service.
B. docker stack passes the images to the Swarm master which distributes the images to all other Swarm nodes.
C. docker stack instructs the Swarm nodes to pull the images from a registry, although it does not upload the images to the registry.
D. docker stack transfers the image from its local Docker cache to each Swarm node.
E. docker stack triggers the build process for the images on all nodes of the Swarm.
Answer: C (LEAVE A REPLY)
Docker stack is a command that allows users to deploy and manage a stack of services on a Docker Swarm cluster. A stack is a group of interrelated services that share dependencies and can be orchestrated and scaled together. A stack is typically defined by a Compose file, which is a YAML file that describes the services, networks, volumes, and other resources of the stack. To use docker stack to run a Compose file on a Swarm, the user must first create and initialize a Swarm cluster, which is a group of machines (nodes) that are running the Docker Engine and are joined into a single entity. The Swarm cluster has one or more managers, which are responsible for maintaining the cluster state and orchestrating the services, and one or more workers, which are the nodes that run the services.
When the user runs docker stack deploy with a Compose file, the command parses the file and creates the services as specified. However, docker stack does not build or upload the images referenced in the Compose file to any registry. Instead, it instructs the Swarm nodes to pull the images from a registry, which can be the public Docker Hub or a private registry. The user must ensure that the images are available in the registry before deploying the stack, otherwise the deployment will fail. The user can use docker build and docker push commands to create and upload the images to the registry, or use an automated build service such as Docker Hub or GitHub Actions. The user must also make sure that the image names and tags in the Compose file match the ones in the registry, and that the Swarm nodes have access to the registry if it is private. By pulling the images from a registry, docker stack ensures that the Swarm nodes have the same and latest version of the images, and that the images are distributed across the cluster in an efficient way.
The other options are not correct. Docker stack does not build the images locally or on the Swarm nodes, nor does it copy or transfer the images to the Swarm nodes. Docker stack also does not pass the images to the Swarm master, as this would create a bottleneck and a single point of failure. Docker stack relies on the registry as the source of truth for the images, and delegates the image pulling to the Swarm nodes. References:
* Deploy a stack to a swarm | Docker Docs1

* docker stack deploy | Docker Docs2
* docker build | Docker Docs3
* docker push | Docker Docs4

NEW QUESTION: 31
Which operating systems are compatible with cloud-init? (Select all that apply)
A. Windows
B. macOS
C. Linux
D. Android
Answer: C (LEAVE A REPLY)
Cloud-initis an initialization and configuration system designed primarily forLinux-based cloud instances.
According to official cloud-init documentation, it is supported on a wide range of Linux distributions, including Ubuntu, Red Hat Enterprise Linux, CentOS, Debian, and SUSE.
Windows systems use a separate project calledCloudbase-Init, which is not cloud-init itself.
macOS and Android do not support cloud-init.
Therefore, the correct answer isC (Linux).

NEW QUESTION: 32
Which of the following statements are true regarding resource management for full virtualization? (Choose two.)
A. The hygervisor may provide fine-grained limits to internal elements of the guest operating system such as the number of processes.
B. The hypervisor provides each virtual machine with hardware of a defined capacity that limits the resources of the virtual machine.
C. Full virtualization cannot pose any limits to virtual machines and always assigns the host system's resources in a first-come-first-serve manner.
D. All processes created within the virtual machines are transparently and equally scheduled in the host system for CPU and I/O usage.
E. It is up to the virtual machine to use its assigned hardware resources and create, for example, an arbitrary amount of network sockets.

Answer: B,E (LEAVE A REPLY)
Resource management for full virtualization is the process of allocating and controlling the physical resources of the host system to the virtual machines running on it. The hypervisor is the software layer that performs this task, by providing each virtual machine with a virtual hardware of a defined capacity that limits the resources of the virtual machine. For example, the hypervisor can specify how many virtual CPUs, how much memory, and how much disk space each virtual machine can use. The hypervisor can also enforce resource isolation and prioritization among the virtual machines, to ensure that they do not interfere with each other or consume more resources than they are allowed to. The hypervisor cannot provide fine-grained limits to internal elements of the guest operating system, such as the number of processes, because the hypervisor does not have access to the internal state of the guest operating system. The guest operating system is responsible for managing its own resources within the virtual hardware provided by the hypervisor. For example, the guest operating system can create an arbitrary amount of network sockets, as long as it does not exceed the network bandwidth allocated by the hypervisor. Full virtualization can pose limits to virtual machines, and does not always assign the host system's resources in a first-come-first-serve manner. The hypervisor can use various resource management techniques, such as reservation, limit, share, weight, and quota, to allocate and control the resources of the virtual machines. The hypervisor can also use resource scheduling algorithms, such as round-robin, fair-share, or priority-based, to distribute the resources among the virtual machines according to their needs and preferences. All processes created within the virtual machines are not transparently and equally scheduled in the host system for CPU and I/O usage. The hypervisor can use different scheduling policies, such as proportional-share, co-scheduling, or gang scheduling, to schedule the virtual CPUs of the virtual machines on the physical CPUs of the host system. The hypervisor can also use different I/O scheduling algorithms, such as deadline, anticipatory, or completely fair queuing, to schedule the I/O requests of the virtual machines on the physical I/O devices of the host system. The hypervisor can also use different resource accounting and monitoring mechanisms, such as cgroups, perf, or sar, to measure and report the resource consumption and performance of the virtual machines.
References:
* Oracle VM VirtualBox: Features Overview
* Resource Management as an Enabling Technology for Virtualization - Oracle
* Introduction to virtualization and resource management in IaaS | Cloud Native Computing Foundation

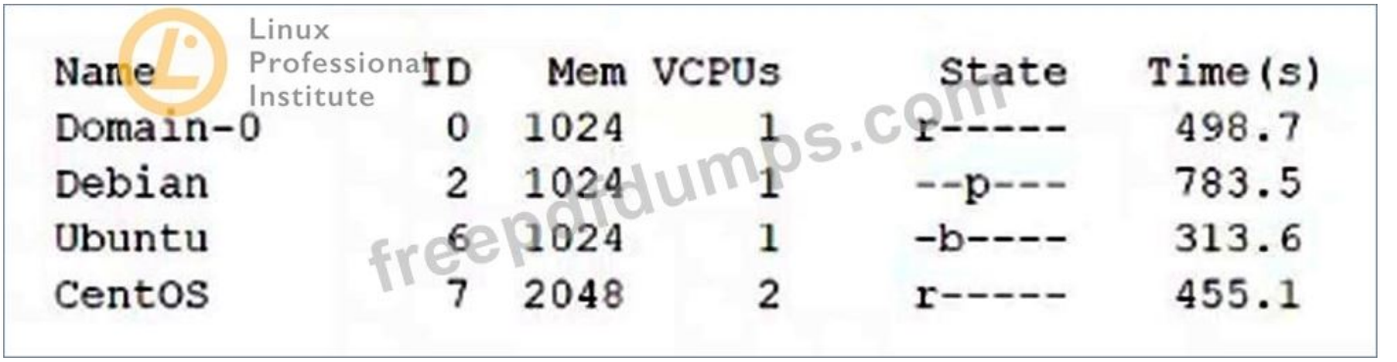
NEW QUESTION: 33
Which CPU flag indicates the hardware virtualization capability on an AMD CPU?
A. HVM
B. VIRT
C. SVM
D. PVM
E. VMX

Answer: C (LEAVE A REPLY)
The CPU flag that indicates the hardware virtualization capability on an AMD CPU is SVM. SVM stands for Secure Virtual Machine, and it is a feature of AMD processors that enables the CPU to run virtual machines with hardware assistance. SVM is also known as AMD-V, which is AMD's brand name for its virtualization technology. SVM allows the CPU to support a hypervisor, which is a software layer that creates and manages virtual machines. A hypervisor can run multiple virtual machines on a single physical machine, each with its own operating system and applications. SVM improves the performance and security of virtual machines by allowing the CPU to directly execute privileged instructions and handle memory access, instead of relying on software emulation or binary translation. SVM also provides nested virtualization, which is the ability to run a virtual machine inside another virtual machine. To use SVM, the CPU must support it and the BIOS must enable it. The user can check if the CPU supports SVM by looking for the svm flag in the /proc/cpuinfo file or by using the lscpu command. The user can also use the virt-host-validate command to verify if the CPU and the BIOS are properly configured for hardware virtualization123. References:
* How to check if CPU supports hardware virtualization (VT technology)1
* Processor support - KVM3
* How to Enable Virtualization in BIOS for Intel and AMD4

NEW QUESTION: 34
What happens when the following command is executed twice in succession?
docker run -tid -v data:/data debian bash
A. The container resulting from the second invocation can only read the content of /data/ and cannot change it.
B. Each container is equipped with its own independent data volume, available at /data/ in the respective container.
C. Both containers share the contents of the data volume, have full permissions to alter its content and mutually see their respective changes.
D. The original content of the container image data is available in both containers, although changes stay local within each container.
E. The second command invocation fails with an error stating that the volume data is already associated with a running container.
Answer: C (LEAVE A REPLY)
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 container12. 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.

NEW QUESTION: 35
Which of the following statements is true regarding the following output ofxl list:



A. Both Debian and Ubuntu require xl commands to start running.
B. The domain with ID 2 uses Para virtualization.
C. CentOS is the domain which has consumed the most CPU time.
D. Ubuntu is idle or waiting for I/O.
E. It is necessary to use the xl command to change Ubuntu's state to running.
Answer: D (LEAVE A REPLY)
The output of xl list shows the state of the domains. The domain with ID 6, Ubuntu, has a state of "b-". This means that the domain is blocked, which means it is idle or waiting for I/O.
https://xenbits.xen.org/docs/unstable/man/xl.1.html

NEW QUESTION: 36
How does Packer interact with system images?
A. Packer has to be installed within the target image and is executed during the image's first boot in order to execute preparation tasks.
B. Packer installs a client within the image which has to be run periodically via cron in order to retrieve the latest template from the Packer server and apply it locally.
C. Packer periodically connects through the network to the Packer daemons of all running Packer images in order to re-apply the whole template to the running instance.
D. Packer downloads and extracts an image in order to make changes to the image's file system, repack the modified image and upload it again.
E. Packer creates an instance based on a source image, prepares the instance through a network connection and bundles the resulting instance as a new system image.
Answer: E (LEAVE A REPLY)

Packer is a tool that automates the creation of identical machine images for multiple platforms from a single source configuration. Packer works by creating an instance based on a source image, which is a pre-existing image that serves as a starting point. Packer then connects to the instance through a network connection, such as SSH or WinRM, and runs various commands and scripts to install and configure software within the instance. Packer then shuts down the instance and creates a new system image from it, which can be used to launch new instances. Packer supports many platforms, such as AWS, Azure, VMware, Docker, and others.
Packer does not install any software or run any daemon within the target image, nor does it periodically connect to the running instances to re-apply the template. Packer also does not modify the source image directly, but creates a new image from the modified instance.
References:
* Packer by HashiCorp
* HashiCorp Packer - Build Automated Machine Images
* Introduction | Packer | HashiCorp Developer

NEW QUESTION: 37
In an IaaS cloud, what is a common method for provisioning new computing instances with an operating system and software?
A. Each new instance is connected to the installation media of a Linux distribution and provides access to the installer by logging in via SSH.
B. Each new instance is created based on an image file that contains the operating system as well as software and default configuration for a given purpose.
C. Each new instance is a clone of another currently running instance that includes all the software, data and state of the original instance.
D. Each new instance is connected via a VPN with the computer that started the provisioning and tries to PXE boot from that machine.
E. Each new instance contains a minimal live system running from a virtual CD as the basis from which the administrator deploys the target operating system.
Answer: B (LEAVE A REPLY)
In an IaaS cloud, the most common method for provisioning new computing instances is to use an image file that contains a pre-installed operating system and software. This image file is also known as a machine image, a virtual appliance, or a template. The image file can be customized for a specific purpose, such as a web server, a database server, or a development environment. The image file can be stored in a repository or a library that is accessible by the cloud provider or the user. When a new instance is requested, the cloud provider copies the image file to a virtual disk and attaches it to the instance. The instance then boots from the virtual disk and runs the operating system and software from the image file. This method is faster and more efficient than installing the operating system and software from scratch for each new instance. It also ensures consistency and reliability across multiple instances that use the same image file. References:
* LPI Virtualization and Containerization Exam Objectives, Topic 305.1: Virtualization Concepts and Theory, Objective: Describe the concept of machine images and templates

* LPI Virtualization and Containerization Study Guide, Chapter 1: Virtualization Concepts and Theory, Section: Machine Images and Templates
* LPI LPIC-3 305 Certification Sample Questions and Practice Exam, Question 10: In an IaaS cloud, what is a common method for provisioning new computing instances with an operating system and software?

NEW QUESTION: 38
What is oVirt?
A. An extension to the Linux Kernel used to provide container virtualization similar to LXC and OpenVZ
B. A comprehensive management infrastructure for Linux-based virtualization
C. An approach used to eliminate the need for virtualization called Zero-Virt
D. A Linux-based hypervisor similar to KVM and Xen
E. A library that provides access to several different virtualization technologies in a common manner
Answer: B (LEAVE A REPLY)
oVirt is anopen-source virtualization management platformthat provides a comprehensive infrastructure for managingLinux-based virtualization environments. According to official oVirt and Red Hat documentation, oVirt acts as amanagement layer on top of KVM, offering centralized control over hosts, virtual machines, storage, and networking.
oVirt itself isnot a hypervisor; instead, it relies onKVMas the underlying virtualization technology. It provides both aweb-based management interfaceand REST APIs, enabling administrators to manage large- scale virtualization deployments efficiently. Features include VM lifecycle management, live migration, high availability, storage domain management, and integration with directory services.
Option E describeslibvirt, not oVirt. Options A, C, and D do not match oVirt's documented purpose.
Therefore, the correct answer isB.

NEW QUESTION: 39
Which of the following tools is not typically used for VM provisioning and image creation?
A. Kubernetes
B. Vagrant
C. Packer
D. cloud-init
Answer: A (LEAVE A REPLY)
VM provisioning and image creation tools are used tobuild, configure, and initialize virtual machine images. According to virtualization documentation,Packeris specifically designed for automated image creation,Vagrantis used for provisioning reproducible development environments, andcloud-initis used to configure instances at first boot.

Kubernetes, however, is acontainer orchestration platform, not a VM provisioning or image creation tool.
It manages containerized workloads after infrastructure has already been provisioned.
Documentation clearly separatesinfrastructure provisioning toolsfromapplication orchestration platforms
. As such, Kubernetes is not typically used for VM provisioning or image creation.
Therefore, the correct answer isA.

NEW QUESTION: 40
Which of the following statements is true regarding networking with libvirt?
A. Libvirt's network functionality is limited to connecting virtual machines to a physical network interface of the host system.
B. Libvirt assiqns the same MAC address to all virtual machines and isolates their network interfaces at the link layer.
C. Libvirt networks appear, by default, as standard Linux bridges in the host system.
D. Libvirt requires a dedicated network interface that may not be used by the host system.
E. Libvirt supports exactly one virtual network and connects all virtual machines to it.
Answer: (SHOW ANSWER)
Libvirt supports creating and managing various types of virtual networks that can be used to connect virtual machines to each other or to the external network. One of the common types of virtual networks is the NAT- based network, which uses network address translation (NAT) to allow virtual machines to access the outside world through the host's network interface. By default, libvirt creates a NAT-based network called 'default' when it is installed and started. This network appears as a standard Linux bridge device on the host system, named virbr0. The bridge device has an IP address of 192.168.122.1/24 and acts as a gateway and a DHCP server for the virtual machines connected to it. The bridge device also has iptables rules to forward and masquerade the traffic from and to the virtual machines. The virtual machines connected to the 'default' network have their own IP addresses in the 192.168.122.0/24 range and their own MAC addresses generated by libvirt. The virtual machines can communicate with each other, with the host, and with the external network through the bridge device and the NAT mechanism12.
The other statements in the question are false regarding networking with libvirt. Libvirt's network functionality is not limited to connecting virtual machines to a physical network interface of the host system. Libvirt can also create isolated networks that do not have any connection to the outside world, or routed networks that use static routes to connect virtual machines to the external network without NAT3.
Libvirt does not assign the same MAC address to all virtual machines and isolate their network interfaces at the link layer. Libvirt assigns a unique MAC address to each virtual machine and allows them to communicate with each other at the network layer4. Libvirt does not require a dedicated network interface that may not be used by the host system. Libvirt can share the host's network interface with the virtual machines using NAT or bridging, or it can pass a physical network interface to a virtual machine exclusively using PCI passthrough5. Libvirt does not

support exactly one virtual network and connect all virtual machines to it. Libvirt supports creating and managing multiple virtual networks with different names and configurations, and connecting virtual machines to different networks according to their needs6. References:
* libvirt: Virtual Networking
* libvirt: NAT forwarding (aka "virtual networks")
* libvirt: Routed network
* libvirt: MAC address
* libvirt: PCI passthrough of host network devices
* [libvirt: Network XML format]

NEW QUESTION: 41
Which of the following statements about the commandlxc-checkpointis correct?
A. It creates a clone of a container.
B. It doubles the memory consumption of the container.
C. It only works on stopped containers.
D. It writes the status of the container to a file.
E. It creates a container image based on an existing container.
Answer: D (LEAVE A REPLY)
The command lxc-checkpoint is used to checkpoint and restore containers. Checkpointing a container means saving the state of the container, including its memory, processes, file descriptors, and network connections, to a file or a directory. Restoring a container means resuming the container from the saved state, as if it was never stopped. Checkpointing and restoring containers can be useful for various purposes, such as live migration, backup, debugging, or snapshotting. The command lxc-checkpoint has the following syntax:
lxc-checkpoint {-n name} {-D path} [-r] [-s] [-v] [-d] [-F]
The options are:
* -n name: Specify the name of the container to checkpoint or restore.
* -D path: Specify the path to the file or directory where the checkpoint data is dumped or restored.
* -r, --restore: Restore the checkpoint for the container, instead of dumping it. This option is incompatible with -s.
* -s, --stop: Optionally stop the container after dumping. This option is incompatible with -r.
* -v, --verbose: Enable verbose criu logging. Only available when providing -r.
* -d, --daemon: Restore the container in the background (this is the default). Only available when providing -r.
* -F, --foreground: Restore the container in the foreground. Only available when providing -r.
The command lxc-checkpoint uses the CRIU (Checkpoint/Restore In Userspace) tool to perform the checkpoint and restore operations. CRIU is a software that can freeze a running application (or part of it) and checkpoint it to a hard drive as a collection of files. It can then use the files to restore and run the application from the point it was frozen at1.

The other statements about the command lxc-checkpoint are not correct. It does not create a clone or an image of a container, nor does it double the memory consumption of the container. It can work on both running and stopped containers, depending on the options provided.
References:
* Linux Containers - LXC - Manpages - lxc-checkpoint.12
* lxc-checkpoint(1) - Linux manual page - man7.org3
* CRIU4

NEW QUESTION: 42
What is the default provider of Vagrant?
A. lxc
B. hyperv
C. virtualbox
D. vmware_workstation
E. docker
Answer: (SHOW ANSWER)
Vagrant is a tool that allows users to create and configure lightweight, reproducible, and portable development environments. Vagrant supports multiple providers, which are the backends that Vagrant uses to create and manage the virtual machines. By default, VirtualBox is the default provider for Vagrant.
VirtualBox is still the most accessible platform to use Vagrant: it is free, cross-platform, and has been supported by Vagrant for years. With VirtualBox as the default provider, it provides the lowest friction for new users to get started with Vagrant. However, users can also use other providers, such as VMware, Hyper- V, Docker, or LXC, depending on their preferences and needs. To use another provider, users must install it as a Vagrant plugin and specify it when running Vagrant commands. Users can also change the default provider by setting the VAGRANT_DEFAULT_PROVIDER environmental variable. References:
* Default Provider - Providers | Vagrant | HashiCorp Developer1
* Providers | Vagrant | HashiCorp Developer2
* How To Set Default Vagrant Provider to Virtualbox3

NEW QUESTION: 43
Which of the following commands executes a command in a running LXC container?
A. lxc-accach
B. lxc-batch
C. lxc-run
D. lxc-enter
E. lxc-eval
Answer: A (LEAVE A REPLY)
The command lxc-attach is used to execute a command in a running LXC container. It allows the user to start a process inside the container and attach to its standard input, output, and error

streams1. For example, the command lxc-attach -n mycontainer -- ls -lh /home will list all the files and directories in the /home directory of the container named mycontainer1. The other options are not valid LXC commands. The command lxc- batch does not exist. The command lxc-run is an alias for lxc-start, which is used to start a container, not to execute a command in it2. The command lxc-enter is also an alias for lxc-attach, but it is deprecated and should not be used3. The command lxc-eval is also not a valid LXC command. References:
* 1: Executing a command inside a running LXC - Unix & Linux Stack Exchange.
* 2: lxc-start: start a container. - SysTutorials.
* 3: lxc-attach: start a process inside a running container. - SysTutorials.

NEW QUESTION: 44
Where are paravirtualized device drivers installed?
A. In the Guest OS
B. No special drivers are required for paravirtualization
C. Compiled into the hypervisor
D. In the Host OS
Answer: A (LEAVE A REPLY)
Paravirtualization is a virtualization technique where the guest operating system is aware that it is running in a virtualized environment and cooperates directly with the hypervisor. According to Xen and virtualization documentation, this cooperation is achieved through paravirtualized device drivers, which must be installed inside the guest operating system.
These specialized drivers replace traditional hardware drivers and communicate directly with the hypervisor using optimized interfaces. This reduces the overhead of hardware emulation and improves performance, particularly for disk and network I/O. Because of this design, the guest OS must either be modified or explicitly include support for paravirtualized drivers.
Option B is incorrect because paravirtualization explicitly requires special guest-side drivers. Option C is incorrect because drivers are not compiled into the hypervisor; they reside in the guest. Option D is also incorrect because host OS drivers do not provide paravirtualized functionality to guests.
Virtualization notes consistently state that paravirtualized drivers are installed and run within the guest OS, making option A the correct answer.

NEW QUESTION: 45
Which of the following is a benefit of virtualization?
A. All of the above
B. Improved security
C. Decreased resource utilization
D. Simplified hardware management
Answer: D (LEAVE A REPLY)

One of the core benefits of virtualization issimplified hardware management. Virtualization abstracts physical hardware into software-defined resources, making it easier to manage, allocate, migrate, and maintain systems.
Option C is incorrect because virtualizationincreases, not decreases, resource utilization. Since one option is incorrect, "All of the above" cannot be correct. While improved security is also a benefit, the most accurate single answer in this context issimplified hardware management. Therefore, the correct answer isD.

NEW QUESTION: 46
When using Packer to create machine images, what are some common sources for base images? (Select all that apply)
A. Physical servers
B. Images from untrusted sources
C. Custom images created by the organization
D. Official operating system images
Answer: (SHOW ANSWER)
Packer is an image automation tool used to create identical machine images for multiple platforms from a single configuration. According to virtualization and containerization documentation, Packer typically uses trusted and standardized base imagesas the starting point for image creation. The most common sources includeofficial operating system imagesprovided by vendors such as Red Hat, Ubuntu, or cloud service providers, andcustom images created internally by an organization.
Using official images ensures that the base system is secure, properly maintained, and compliant with vendor standards. Organizations often build custom base images to enforce internal security policies, hardening standards, and preinstalled software, which Packer then extends into environment-specific images.
Physical servers are not considered common or practical base sources for Packer, as Packer operates on machine image artifacts rather than cloning live hardware systems. Images from untrusted sources are explicitly discouraged in virtualization documentation due to security and compliance risks.
Therefore, documentation clearly supportscustom organizational imagesandofficial operating system imagesas the correct and recommended base sources for Packer image creation.

NEW QUESTION: 47
In the command vzctl _____ 105 /usr/bin/apt-get install wget, which subcommand of vzctl is missing in order to install wget in the OpenVZ container 105?
Answer:
exec
Explanation:
In OpenVZ virtualization, thevzctlcommand is used to manage containers. According to OpenVZ documentation, theexecsubcommand is required to execute a commandinside a running container.
The correct command syntax is:
vzctl exec 105 /usr/bin/apt-get install wget
The exec subcommand runs the specified command in the context of the container identified by its container ID (CTID), in this case105. Without exec, the command would not be executed inside the container environment.
Therefore, the missing and correct subcommand isexec.

NEW QUESTION: 48
What is the purpose of a .dockerignore file?
A. It lists files existing in a Docker image which should be excluded when building a derivative image.
B. It specifies files that Docker does not submit to the Docker daemon when building a Docker image
C. It exists in the root file system of containers that should ignore volumes and ports provided by Docker.
D. It must be placed in the top level directory of volumes that Docker should never attach automatically to a container
E. It specifies which parts of a Dockerfile should be ignored when building a Docker image.
Answer: B (LEAVE A REPLY)
The purpose of a .dockerignore file is to specify files that Docker does not submit to the Docker daemon when building a Docker image. A .dockerignore file is a text file that contains a list of files or directories that should be excluded from the build context, which is the set of files and folders that are available for use in a Dockerfile. By using a .dockerignore file, you can avoid sending files or directories that are large, contain sensitive information, or are irrelevant to the Docker image to the daemon, which can improve the efficiency and security of the build process. The other options are incorrect because they do not describe the function of a .dockerignore file. Option A is wrong because a .dockerignore file does not affect the files existing in a Docker image, but only the files sent to the daemon during the build. Option C is wrong because a .

dockerignore file does not exist in the root file system of containers, but in the same directory as the Dockerfile. Option D is wrong because a .dockerignore file does not affect the volumes that Docker attaches to a container, but only the files included in the build context. Option E is wrong because a .dockerignore file does not affect the parts of a Dockerfile that are executed, but only the files available for use in a Dockerfile.
References:
* What are .dockerignore files, and why you should use them?
* Dockerfile reference | Docker Docs
* How to use .dockerignore and its importance - Shisho Cloud

NEW QUESTION: 49
Which of the following statements are true regarding a Pod in Kubernetes? (Choose two.)
A. All containers of a Pod run on the same node.
B. Pods are always created automatically and cannot be explicitly configured.
C. A Pod is the smallest unit of workload Kubernetes can run.
D. When a Pod fails, Kubernetes restarts the Pod on another node by default.
E. systemd is used to manage individual Pods on the Kubernetes nodes.
Answer: A,C (LEAVE A REPLY)
A Pod in Kubernetes is a collection of one or more containers that share the same network and storage resources, and a specification for how to run the containers. A Pod is the smallest unit of workload Kubernetes can run, meaning that it cannot be divided into smaller units. Therefore, option C is correct. All containers of a Pod run on the same node, which is the smallest unit of computing hardware in Kubernetes. A node is a physical or virtual machine that hosts one or more Pods. Therefore, option A is also correct. Pods are not always created automatically and cannot be explicitly configured. Pods can be created manually using YAML or JSON files, or using commands like kubectl run or kubectl create. Pods can also be created automatically by higher-level controllers, such as Deployment, ReplicaSet, or StatefulSet. Therefore, option B is incorrect. When a Pod fails, Kubernetes does not restart the Pod on another node by default. Pods are ephemeral by nature, meaning that they can be terminated or deleted at any time. If a Pod is managed by a controller, the controller will create a new Pod to replace the failed one, but it may not be on the same node.
Therefore, option D is incorrect. systemd is not used to manage individual Pods on the Kubernetes nodes.
systemd is a system and service manager for Linux operating systems that can start and stop services, such as docker or kubelet. However, systemd does not interact with Pods directly. Pods are managed by the kubelet service, which is an agent that runs on each node and communicates with the Kubernetes control plane.
Therefore, option E is incorrect. References:
* Pods | Kubernetes
* What is a Kubernetes pod? - Red Hat
* What's the difference between a pod, a cluster, and a container?

* What are Kubernetes Pods? | VMware Glossary
* Kubernetes Node Vs. Pod Vs.Cluster: Key Differences - CloudZero

NEW QUESTION: 50
Which statement is true regarding the Linux kernel module that must be loaded in order to use QEMU with hardware virtualization extensions?
A. It must be loaded into the kernel of the host system only if the console of a virtual machine will be connected to a physical console of the host system
B. It must be loaded into the kernel of each virtual machine that will access files and directories from the host system's file system.
C. It must be loaded into the Kernel of the host system in order to use the visualization extensions of the host system's CPU
D. It must be loaded into the kernel of the first virtual machine as it interacts with the QEMU bare metal hypervisor and is required to trigger the start of additional virtual machines
E. It must be loaded into the kernel of each virtual machine to provide Para virtualization which is required by QEMU.
Answer: C (LEAVE A REPLY)
The Linux kernel module that must be loaded in order to use QEMU with hardware virtualization extensions is KVM (Kernel-based Virtual Machine). KVM is a full virtualization solution that allows a user space program (such as QEMU) to utilize the hardware virtualization features of various processors (such as Intel VT or AMD-V). KVM consists of a loadable kernel module, kvm.ko, that provides the core virtualization infrastructure and a processor specific module, kvm-intel.ko or kvm-amd.ko. KVM must be loaded into the kernel of the host system in order to use the virtualization extensions of the host system's CPU. This enables QEMU to run multiple virtual machines with unmodified Linux or Windows images, each with private virtualized hardware. KVM is integrated with QEMU, so there is no need to load it into the kernel of each virtual machine or the first virtual machine. KVM also does not require paravirtualization, which is a technique that modifies the guest operating system to communicate directly with the hypervisor, bypassing the emulation layer. References:
* Features/KVM - QEMU
* Kernel-based Virtual Machine
* KVM virtualization on Red Hat Enterprise Linux 8 (2023)

NEW QUESTION: 51
Which of the following commands lists all differences between the disk images vm1-snap.img and vm1.img?
A. virt-delta -a vm1-snap.img -A vm1.img
B. virt-cp-in -a vm1-snap.img -A vm1.img
C. virt-cmp -a vm1-snap.img -A vm1.img
D. virt-history -a vm1-snap.img -A vm1.img
E. virt-diff -a vm1-snap.img -A vm1.img

Answer: (SHOW ANSWER)
The virt-diff command-line tool can be used to list the differences between files in two virtual machines or disk images. The output shows the changes to a virtual machine's disk images after it has been running. The command can also be used to show the difference between overlays1. To specify two guests, you have to use the -a or -d option for the first guest, and the -A or -D option for the second guest. For example: virt-diff -a old.img -A new.img1. Therefore, the correct command to list all differences between the disk images vm1- snap.img and vm1.img is: virt-diff -a vm1-snap.img -A vm1.img. The other commands are not related to finding differences between disk images. virt-delta is a tool to create delta disks from two disk images2. virt- cp-in is a tool to copy files and directories into a virtual machine disk image3. virt-cmp is a tool to compare two files or directories in a virtual machine disk image4. virt-history is a tool to show the history of a virtual machine disk image5. References:
* 21.13. virt-diff: Listing the Differences between Virtual Machine Files ...
* 21.14. virt-delta: Creating Delta Disks from Two Disk Images ...
* 21.6. virt-cp-in: Copying Files and Directories into a Virtual Machine Disk Image ...
* 21.7. virt-cmp: Comparing Two Files or Directories in a Virtual Machine Disk Image ...
* 21.8. virt-history: Showing the History of a Virtual Machine Disk Image ...

NEW QUESTION: 52
Which command in the KVM monitor restores a snapshot?
Answer:
loadvm
Explanation:
In KVM and QEMU-based virtualization environments, the QEMU monitor provides an interactive interface for managing virtual machine runtime operations. According to KVM documentation, the command used to restore a previously saved snapshot within the monitor is loadvm.
Snapshots capture the state of a virtual machine at a specific point in time, including CPU state, memory, and disk state (depending on configuration). The loadvm command allows administrators to revert a virtual machine back to that saved state, which is especially useful for testing, debugging, and recovery scenarios.
This command is typically used in conjunction with the savevm command, which creates snapshots. The functionality is supported primarily with disk formats such as QCOW2, which allow snapshot capabilities.
Therefore, the correct and documented command is loadvm.

NEW QUESTION: 53
What is a container image?
A. A physical server
B. A lightweight virtual machine
C. A snapshot of an application and its dependencies
D. An operating system kernel

Answer: C (LEAVE A REPLY)
Acontainer imageis astatic, immutable packagethat contains an application along with all its required dependencies, libraries, and configuration files. Containerization documentation defines a container image as a snapshot of an application and its runtime environment, which can be instantiated as a running container.
Container images do not include a full operating system kernel; instead, containers share the host system's kernel. This design makes containers lightweight and fast to deploy compared to traditional virtual machines.
Therefore, the correct answer isC.

NEW QUESTION: 54
Which of the following commands boots a QEMU virtual machine using hardware virtualization extensions?
A. qvirt -create -drive file=debian.img -cdrom debian.iso -m 1024 -boot d -driver hvm
B. vm -kvm -drive file=debian.img -cdrom debian.iso -m 1024 -boot d
C. qemu-hw -create -drive file=debian.img -cdrom debian.iso -m 1024 -boot d
D. qemu -accel kvm -drive file-debian.img -cdrom debian.iso -m 1024 -boot d
E. qvm start -vmx -drive file=debian.img -cdrom debian.iso -m 1024 -boot d
Answer: D (LEAVE A REPLY)
The correct command to boot a QEMU virtual machine using hardware virtualization extensions is qemu - accel kvm -drive file-debian.img -cdrom debian.iso -m 1024 -boot d. This command uses the -accel option to specify the hardware accelerator to use, which in this case is kvm. KVM is a full virtualization solution for Linux on x86 hardware containing virtualization extensions (Intel VT or AMD-V)1. The -drive option specifies the disk image file to use, which in this case is debian.img. The -cdrom option specifies the ISO image file to use as a CD-ROM, which in this case is debian.iso. The -m option specifies the amount of memory to allocate to the virtual machine, which in this case is 1024 MB. The -boot option specifies the boot order, which in this case is d, meaning to boot from the CD-ROM first. References:
https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html
/virtualization_deployment_and_administration_guide/sect-troubleshooting-
enabling_intel_vt_x_and_amd_v_virtualization_hardware_extensions_in_bios
https://fedoraproject.org/wiki/Virtualization

NEW QUESTION: 55
After creating a new Docker network using the following command:
docker network create --driver bridge isolated_nw
which parameter must be added todocker createin order to attach a container to the network?
A. --eth0=isolated_nw
B. --alias=isolated_nw
C. --ethernet=isolated_nw
D. --network=isolated_nw

E. --attach=isolated_nw

Answer: (SHOW ANSWER)

To attach a container to a network when creating it, the --network flag must be used with the name of the network as the argument. The --network flag specifies the network mode for the container. By default, the network mode is bridge, which means the container is connected to the default bridge network. However, if a custom network is created, such as isolated_nw in this case, the container must be explicitly attached to it using the --network flag. For example, to create a container named web1 and attach it to the isolated_nw network, the command would be:
docker create --name web1 --network isolated_nw nginx
The other options are not valid parameters for docker create. The --eth0, --ethernet, and --attach flags do not exist. The --alias flag is used to specify an additional network alias for the container on a user-defined network, but it does not attach the container to the network. References:
* docker network create | Docker Documentation1
* docker create | Docker Documentation
* Networking overview | Docker Docs2

NEW QUESTION: 56

Xen is a:

A. Type 2 hypervisor

B. Type 1 hypervisor

C. Application virtualization platform

D. Virtualization management tool

Answer: B (LEAVE A REPLY)

Xen is classified as a Type 1 (bare-metal) hypervisor, meaning it runs directly on the system hardware rather than on top of a host operating system. Virtualization documentation defines Type 1 hypervisors as having direct control over CPU, memory, and hardware resources, providing better performance and stronger isolation compared to Type 2 hypervisors.
In a Xen-based system, the hypervisor is loaded first during boot. A privileged management domain, known as Domain 0 (Dom0), is then started to manage hardware access and virtual machine lifecycle operations.
This architecture clearly differentiates Xen from hosted (Type 2) hypervisors such as VirtualBox. Xen is not merely a management tool or application virtualization platform; it is a full hypervisor used in enterprise, cloud, and service provider environments. Therefore, the correct answer is B.