

305-300 Latest Examprep, 305-300 Examcollection Questions Answers



BONUS!!! Download part of ExamsTorrent 305-300 dumps for free: <https://drive.google.com/open?id=1-V9cZssNrpr3TvaAdq6ns-1XsA4Ig5Nq>

Every browser such as Chrome, Mozilla Firefox, MS Edge, Internet Explorer, Safari, and Opera supports this format of LPIC-3 Exam 305: Virtualization and Containerization (305-300) mock exam. You can attempt the LPIC-3 Exam 305: Virtualization and Containerization (305-300) test multiple times to relieve exam stress and boosts confidence. Besides Windows, ExamsTorrent Lpi 305-300 web-based practice exam works on iOS, Android, Linux, and Mac.

To prepare for the LPI 305-300 Certification Exam, candidates should have a solid foundation in virtualization and containerization technologies. They should also have practical experience working with virtualization and containerization software, such as Docker, Kubernetes, and VMware. Candidates may also benefit from taking training courses or studying exam preparation materials to brush up on the exam's topics and objectives.

>> 305-300 Latest Examprep <<

305-300 Examcollection Questions Answers, Valid 305-300 Test Pattern

To help you learn with the newest content for the 305-300 preparation materials, our experts check the updates status every day, and their diligent works as well as professional attitude bring high quality for our 305-300 practice materials. You may doubtful if you are newbie for our 305-300 training engine, free demos are provided for your reference. The free demo of 305-300 exam questions contains a few of the real practice questions, and you will love it as long as you download and check it.

Lpi LPIC-3 Exam 305: Virtualization and Containerization Sample Questions (Q17-Q22):

NEW QUESTION # 17

If a Dockerfile contains the following lines:

```
WORKDIR /
```

```
RUN cd /tmp
```

```
RUN echo test > test
```

where is the file test located?

- A. /tmp/test on the system running docker build.
- B. test in the directory holding the Dockerfile.
- C. /test within the container image.

- D. /ting/test within the container image.
- E. /root/tesc within the container image.

Answer: C

Explanation:

Explanation

The WORKDIR instruction sets the working directory for any subsequent RUN, CMD, ENTRYPOINT, COPY and ADD instructions that follow it in the Dockerfile¹. The RUN instruction executes commands in a new layer on top of the current image and commits the results². The RUN cd command does not change the working directory for the next RUN instruction, because each RUN command runs in a new shell and a new environment³. Therefore, the file test is created in the root directory (/) of the container image, not in the /tmp directory. References:

* Dockerfile reference: WORKDIR

* Dockerfile reference: RUN

* difference between RUN cd and WORKDIR in Dockerfile

NEW QUESTION # 18

Which operating systems are compatible with cloud-init? (Select all that apply)

- **A. Linux**
- B. Windows
- C. macOS
- D. Android

Answer: A

Explanation:

Cloud-init is an initialization and configuration system designed primarily for Linux-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 called Cloudbase-Init, which is not cloud-init itself. macOS and Android do not support cloud-init.

Therefore, the correct answer is C (Linux).

NEW QUESTION # 19

How does Packer interact with system images?

- **A. 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.**
- B. 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.
- C. 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.
- 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 has to be installed within the target image and is executed during the image's first boot in order to execute preparation tasks.

Answer: A

Explanation:

Explanation

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 # 20

Which of the following kinds of data can cloud-init process directly from user-data? (Choose three.)

- A. ISO images to boot from
- B. Base64-encoded binary files to execute
- C. cloud-config declarations in YAML
- D. Lists of URLs to import
- E. Shell scripts to execute

Answer: C,D,E

Explanation:

Explanation

Cloud-init is a tool that allows users to customize the configuration and behavior of cloud instances during the boot process. Cloud-init can process different kinds of data that are passed to the instance via user-data, which is a mechanism provided by various cloud providers to inject data into the instance. Among the kinds of data that cloud-init can process directly from user-data are:

- * Shell scripts to execute: Cloud-init can execute user-data that is formatted as a shell script, starting with the `#!/bin/sh` or `#!/bin/bash` shebang. The script can contain any commands that are valid in the shell environment of the instance. The script is executed as the root user during the boot process¹².

- * Lists of URLs to import: Cloud-init can import user-data that is formatted as a list of URLs, separated by newlines. The URLs can point to any valid data source that cloud-init supports, such as shell scripts, cloud-config files, or include files. The URLs are fetched and processed by cloud-init in the order they appear in the list¹³.

- * cloud-config declarations in YAML: Cloud-init can process user-data that is formatted as a cloud-config file, which is a YAML document that contains declarations for various cloud-init modules. The cloud-config file can specify various aspects of the instance configuration, such as hostname, users, packages, commands, services, and more. The cloud-config file must start with the `#cloud-config` header¹⁴.

The other kinds of data listed in the question are not directly processed by cloud-init from user-data. They are either not supported, not recommended, or require additional steps to be processed. These kinds of data are:

- * ISO images to boot from: Cloud-init does not support booting from ISO images that are passed as user-data. ISO images are typically used to install an operating system on a physical or virtual machine, not to customize an existing cloud instance. To boot from an ISO image, the user would need to attach it as a secondary disk to the instance and configure the boot order accordingly⁵.

- * Base64-encoded binary files to execute: Cloud-init does not recommend passing binary files as user-data, as they may not be compatible with the instance's architecture or operating system.

Base64-encoding does not change this fact, as it only converts the binary data into ASCII characters. To execute a binary file, the user would need to decode it and make it executable on the instance⁶.

References:

- * User-Data Formats - cloud-init 22.1 documentation
- * User-Data Scripts
- * Include File
- * Cloud Config
- * How to Boot From ISO Image File Directly in Windows
- * How to run a binary file as a command in the terminal?.

NEW QUESTION # 21

When using direct Linux booting to start Linux within a KVM virtual machine, which KVM parameter is used to specify parameters for the Linux kernel?

(Specify ONLY the option name without any values or parameters.)

Answer:

Explanation:

-append

Explanation:

When using direct kernel booting with KVM/QEMU, the `-append` option is used to pass kernel command-line parameters to the Linux

kernel. This allows administrators to specify boot-time options such as root filesystem location, console settings, or debugging flags. Official QEMU documentation lists `-append` as the correct parameter for kernel arguments, making this answer correct.

NEW QUESTION # 22

.....

Most people said the process is more important than the result, but as for 305-300 exam, the result is more important than the process, because it will give you real benefits after you obtain 305-300 exam certification in your career in IT industry. If you have made your decision to pass the exam, our 305-300 exam software will be an effective guarantee for you to Pass 305-300 Exam. Maybe you are still doubtful about our product, it doesn't matter, but if you try to download our free demo of our 305-300 exam software first, you will be more confident to pass the exam which is brought by our ExamsTorrent.

305-300 Examcollection Questions Answers: <https://www.examstorrent.com/305-300-exam-dumps-torrent.html>

- [illegible]

What's more, part of that ExamsTorrent 305-300 dumps now are free: <https://drive.google.com/open?id=1V9cZssNrpr3TvaAdq6ns-1XsA4Ig5Nq>