Juniper JN0-214 JNCIA-Cloud Practice Questions

Shared by Frazier on 19-10-2026

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

Your organization manages all of its sales through the Salesforce CRM solution.

In this scenario, which cloud service model are they using?

Options:

A- Storage as a Service (STaas)
B- Software as a Service (Saa
C- Platform as a Service (Paa)
D- Infrastructure as a Service (IaaS)

Answer:

B

Explanation:

Cloud service models define how services are delivered and managed in a cloud environment. The three primary models are:

Infrastructure as a Service (IaaS): Provides virtualized computing resources such as servers, storage, and networking over the internet. Examples include Amazon EC2 and Microsoft Azure Virtual Machines.

Platform as a Service (PaaS): Provides a platform for developers to build, deploy, and manage applications without worrying about the underlying infrastructure. Examples include Google App Engine and Microsoft Azure App Services.

Software as a Service (SaaS): Delivers fully functional applications over the internet, eliminating the need for users to install or maintain software locally. Examples include Salesforce CRM, Google Workspace, and Microsoft Office 365.

In this scenario, the organization is using Salesforce CRM, which is a SaaS solution. Salesforce provides a complete customer relationship management (CRM) application that is accessible via a web browser, with no need for the organization to manage the underlying infrastructure or application code.

Why SaaS?

No Infrastructure Management: The customer does not need to worry about provisioning servers, databases, or networking components.

Fully Managed Application: Salesforce handles updates, patches, and maintenance, ensuring the

application is always up-to-date.

Accessibility: Users can access Salesforce CRM from any device with an internet connection.

JNCIA Cloud Reference:

The JNCIA-Cloud certification emphasizes understanding the different cloud service models and their use cases. SaaS is particularly relevant in scenarios where organizations want to leverage pre-built applications without the complexity of managing infrastructure or development platforms.

For example, Juniper's cloud solutions often integrate with SaaS platforms like Salesforce to provide secure connectivity and enhanced functionality. Understanding the role of SaaS in cloud architectures is essential for designing and implementing cloud-based solutions.

Juniper JNCIA-Cloud Study Guide: Cloud Service Models

Salesforce CRM Documentation

Question 2

Question Type: MultipleChoice

Click to the Exhibit button.

```
(kolla-toolbox)[user@openstack /]$ openstack server list
+--------------------------------------+--------+--------+---------------+-------+-------------+
| ID                                   | Name   | Status | Networks      | Image | Flavor      |
+--------------------------------------+--------+--------+---------------+-------+-------------+
| 56774543-6fcb-4ed1-b8d0-d1c8884e4626 | myvSRX | ACTIVE | VN-A=10.1.0.3 | vSRX3 | vSRX-Flavor |
+--------------------------------------+--------+--------+---------------+-------+-------------+
(kolla-toolbox)[user@openstack /]$
```

Referring to the exhibit, which two statements are correct? (Select two.)

Options:

A- The myvSRX instance is using a default image.
B- The myvSRX instance is a part of a default network.
C- The myvSRX instance is created using a custom flavor.
D- The myvSRX instance is currently running.

Answer:

C, D

Explanation:

The openstack server list command provides information about virtual machine (VM) instances in the OpenStack environment. Let's analyze the exhibit and each statement:

Key Information from the Exhibit:

The output shows details about the myvSRX instance:

Status: ACTIVE (indicating the instance is running).

Networks: VN-A-10.1.0.3 (indicating the instance is part of a specific network).

Image: vSRX3 (indicating the instance was created using a custom image).

Flavor: vSRX-Flavor (indicating the instance was created using a custom flavor).

Option Analysis:

A . The myvSRX instance is using a default image.

Incorrect: The image name vSRX3 suggests that this is a custom image, not the default image provided by OpenStack.

B . The myvSRX instance is a part of a default network.

Incorrect: The network name VN-A-10.1.0.3 indicates that the instance is part of a specific network, not the default network.

C . The myvSRX instance is created using a custom flavor.

Correct: The flavor name vSRX-Flavor indicates that the instance was created using a custom flavor, which defines the CPU, RAM, and disk space properties.

D . The myvSRX instance is currently running.

Correct: The ACTIVE status confirms that the instance is currently running.

Why These Statements?

Custom Flavor: The vSRX-Flavor name clearly indicates that a custom flavor was used to define the instance's resource allocation.

Running Instance: The ACTIVE status confirms that the instance is operational and available for use.

JNCIA Cloud Reference:

The JNCIA-Cloud certification emphasizes understanding OpenStack commands and outputs,

including the openstack server list command. Recognizing how images, flavors, and statuses are represented is essential for managing VM instances effectively.

For example, Juniper Contrail integrates with OpenStack Nova to provide advanced networking features for VMs, ensuring seamless operation based on their configurations.

OpenStack CLI Documentation: openstack server list Command

Juniper JNCIA-Cloud Study Guide: OpenStack Compute

Question 3

Question Type: MultipleChoice

Which component of Kubernetes runs on each node maintaining network rules?

Options:

A- container runtime
B- kube-proxy
C- kubelet
D- kube controller

Answer:

B

Explanation:

Kubernetes components work together to ensure seamless communication and network functionality within the cluster. Let's analyze each option:

A . container runtime

Incorrect: The container runtime (e.g., containerd, cri-o) is responsible for running containers on worker nodes. It does not maintain network rules.

B . kube-proxy

Correct: kube-proxy is a Kubernetes component that runs on each node and maintains network rules to enable communication between services and pods. It ensures proper load balancing and routing of traffic.

C . kubelet

Incorrect: The kubelet is responsible for managing the state of pods and containers on a node. It does not handle network rules.

D . kube controller

Incorrect: The kube controller manages the desired state of the cluster, such as maintaining the correct number of replicas. It does not directly manage network rules.

Why kube-proxy?

Network Rules: kube-proxy implements iptables or IPVS rules to route traffic between services and pods, ensuring seamless communication.

Load Balancing: It provides basic load balancing for services, distributing traffic across available pods.

JNCIA Cloud Reference:

The JNCIA-Cloud certification covers Kubernetes networking, including the role of kube-proxy. Understanding how kube-proxy works is essential for managing network communication in Kubernetes clusters.

For example, Juniper Contrail integrates with Kubernetes to enhance networking capabilities, leveraging kube-proxy for service-level traffic management.

Kubernetes Documentation: kube-proxy

Juniper JNCIA-Cloud Study Guide: Kubernetes Networking

Question 4

Question Type: MultipleChoice

Which container runtime engine is used by default in OpenShift?

Options:

A- containerd
B- cri-o
C- Docker
D- runC

Answer:

B

Explanation:

OpenShift uses a container runtime engine to manage and run containers within its Kubernetes-based environment. Let's analyze each option:

A . containerd

Incorrect:

While containerd is a popular container runtime used in Kubernetes environments, it is not the default runtime for OpenShift. OpenShift uses a runtime specifically optimized for Kubernetes workloads.

B . cri-o

Correct:

CRI-O is the default container runtime engine for OpenShift. It is a lightweight, Kubernetes-native runtime that implements the Container Runtime Interface (CRI) and is optimized for running containers in Kubernetes environments.

C . Docker

Incorrect:

Docker was historically used as a container runtime in earlier versions of Kubernetes and OpenShift. However, OpenShift has transitioned to CRI-O as its default runtime, as Docker's architecture is not directly aligned with Kubernetes' requirements.

D . runC

Incorrect:

runC is a low-level container runtime that executes containers. While it is used internally by higher-level runtimes like containerd and cri-o, it is not used directly as the runtime engine in OpenShift.

Why CRI-O?

Kubernetes-Native Design: CRI-O is purpose-built for Kubernetes, ensuring compatibility and performance.

Lightweight and Secure: CRI-O provides a minimalistic runtime that focuses on running containers efficiently and securely.

JNCIA Cloud Reference:

The JNCIA-Cloud certification covers container runtimes as part of its curriculum on container

orchestration platforms. Understanding the role of CRI-O in OpenShift is essential for managing containerized workloads effectively.

For example, Juniper Contrail integrates with OpenShift to provide advanced networking features, leveraging CRI-O for container execution.

OpenShift Documentation: CRI-O Runtime

Juniper JNCIA-Cloud Study Guide: Container Runtimes

Question 5

Question Type: MultipleChoice

Which Linux protection ring is the least privileged?

Options:

A- 0
B- 1
C- 2
D- 3

Answer:

D

Explanation:

In Linux systems, the concept of protection rings is used to define levels of privilege for executing processes and accessing system resources. These rings are part of the CPU's architecture and provide a mechanism for enforcing security boundaries between different parts of the operating system and user applications. There are typically four rings in the x86 architecture, numbered from 0 to 3:

Ring 0 (Most Privileged): This is the highest level of privilege, reserved for the kernel and critical system functions. The operating system kernel operates in this ring because it needs unrestricted access to hardware resources and control over the entire system.

Ring 1 and Ring 2: These intermediate rings are rarely used in modern operating systems. They can be utilized for device drivers or other specialized purposes, but most operating systems, including Linux, do not use these rings extensively.

Ring 3 (Least Privileged): This is the least privileged ring, where user-level applications run. Applications running in Ring 3 have limited access to system resources and must request services from the kernel (which runs in Ring 0) via system calls. This ensures that untrusted or malicious code cannot directly interfere with the core system operations.

Why Ring 3 is the Least Privileged:

Isolation: User applications are isolated from the core system functions to prevent accidental or intentional damage to the system.

Security: By restricting access to hardware and sensitive system resources, the risk of vulnerabilities or exploits is minimized.

Stability: Running applications in Ring 3 ensures that even if an application crashes or behaves unexpectedly, it does not destabilize the entire system.

JNCIA Cloud Reference:

The Juniper Networks Certified Associate - Cloud (JNCIA-Cloud) curriculum emphasizes understanding virtualization, cloud architectures, and the underlying technologies that support them. While the JNCIA-Cloud certification focuses more on Juniper-specific technologies like Contrail, it also covers foundational concepts such as virtualization, Linux, and cloud infrastructure.

In the context of virtualization and cloud environments, understanding the role of protection rings is important because:

Hypervisors often run in Ring 0 to manage virtual machines (VMs).

VMs themselves run in a less privileged ring (e.g., Ring 3) to ensure isolation between the guest operating systems and the host system.

For example, in a virtualized environment like Juniper Contrail, the hypervisor (e.g., KVM) manages the execution of VMs. The hypervisor operates in Ring 0, while the guest OS and applications within the VM operate in Ring 3. This separation ensures that the VMs are securely isolated from each other and from the host system.

Thus, the least privileged Linux protection ring is Ring 3 , where user applications execute with restricted access to system resources.

Juniper JNCIA-Cloud Study Guide: Virtualization Basics

x86 Architecture Protection Rings Documentation

Question 6

Question Type: MultipleChoice

Which OpenStack object enables multitenancy?

Options:

A- role
B- flavor
C- image
D- project

Answer:

D

Explanation:

Multitenancy is a key feature of OpenStack, enabling multiple users or organizations to share cloud resources while maintaining isolation. Let's analyze each option:

A . role

Incorrect: A role defines permissions and access levels for users within a project. While roles are important for managing user privileges, they do not directly enable multitenancy.

B . flavor

Incorrect: A flavor specifies the compute, memory, and storage capacity of a VM instance. It is unrelated to enabling multitenancy.

C . image

Incorrect: An image is a template used to create VM instances. While images are essential for deploying VMs, they do not enable multitenancy.

D . project

Correct: A project (also known as a tenant) is the primary mechanism for enabling multitenancy in OpenStack. Each project represents an isolated environment where resources (e.g., VMs, networks, storage) are provisioned and managed independently.

Why Project?

Isolation: Projects ensure that resources allocated to one tenant are isolated from others, enabling secure and efficient resource sharing.

Resource Management: Each project has its own quotas, users, and resources, making it the foundation of multitenancy in OpenStack.

JNCIA Cloud Reference:

The JNCIA-Cloud certification emphasizes understanding OpenStack's multitenancy model, including the role of projects. Recognizing how projects enable resource isolation is essential for managing shared cloud environments.

For example, Juniper Contrail integrates with OpenStack Keystone to enforce multitenancy and network segmentation for projects.

OpenStack Keystone Documentation

Juniper JNCIA-Cloud Study Guide: OpenStack Multitenancy.

Question 7

Question Type: MultipleChoice

Which two statements about Kubernetes are correct? (Select two.)

Options:

A- Kubernetes is compatible with the container open container runtime.
B- Kubernetes requires the Docker daemon to run Docker containers.
C- A container is the smallest unit of computing that you can manage with Kubernetes.
D- A Kubernetes cluster must contain at least one control plane node.

Answer:

A, C

Explanation:

Kubernetes is an open-source container orchestration platform that automates the deployment, scaling, and management of containerized applications. Let's analyze each statement:

A . Kubernetes is compatible with the container open container runtime.

Correct: Kubernetes supports the Open Container Initiative (OCI) runtime standards, which ensure compatibility with various container runtimes like containerd, cri-o, and others. This flexibility allows Kubernetes to work with different container engines beyond just Docker.

B . Kubernetes requires the Docker daemon to run Docker containers.

Incorrect: While Kubernetes historically used Docker as its default container runtime, it no longer depends on the Docker daemon. Instead, Kubernetes uses the Container Runtime

Interface (CRI) to interact with container runtimes like containerd or cri-o. Docker's runtime has been replaced by containerd in most modern Kubernetes deployments.

C . A container is the smallest unit of computing that you can manage with Kubernetes.

Correct: In Kubernetes, a container represents the smallest deployable unit of computing. Containers encapsulate application code, dependencies, and configurations. Kubernetes manages containers through higher-level abstractions like Pods, which are groups of one or more containers.

D . A Kubernetes cluster must contain at least one control plane node.

Incorrect: While a Kubernetes cluster typically requires at least one control plane node to manage the cluster, this statement is incomplete. A functional Kubernetes cluster also requires at least one worker node to run application workloads. Both control plane and worker nodes are essential for a fully operational cluster.

Why These Answers?

Compatibility with OCI Runtimes: Kubernetes' support for OCI-compliant runtimes ensures flexibility and avoids vendor lock-in.

Containers as Smallest Unit: Understanding that containers are the fundamental building blocks of Kubernetes is crucial for designing and managing applications in a Kubernetes environment.

JNCIA Cloud Reference:

The JNCIA-Cloud certification covers Kubernetes as part of its container orchestration curriculum. Understanding Kubernetes architecture, compatibility, and core concepts is essential for deploying and managing containerized applications in cloud environments.

For example, Juniper Contrail integrates with Kubernetes to provide advanced networking and security features for containerized workloads. Proficiency with Kubernetes ensures seamless operation of cloud-native applications.

Kubernetes Documentation: Container Runtimes

Juniper JNCIA-Cloud Study Guide: Kubernetes

Question 8

Question Type: MultipleChoice

Your organization manages all of its sales through the Salesforce CRM solution.

In this scenario, which cloud service model are they using?

Options:

A- Storage as a Service (STaas)
B- Software as a Service (Saa
C- Platform as a Service (Paa)
D- Infrastructure as a Service (IaaS)

Answer:

B

Explanation:

Cloud service models define how services are delivered and managed in a cloud environment. The three primary models are:

Infrastructure as a Service (IaaS): Provides virtualized computing resources such as servers, storage, and networking over the internet. Examples include Amazon EC2 and Microsoft Azure Virtual Machines.

Platform as a Service (PaaS): Provides a platform for developers to build, deploy, and manage applications without worrying about the underlying infrastructure. Examples include Google App Engine and Microsoft Azure App Services.

Software as a Service (SaaS): Delivers fully functional applications over the internet, eliminating the need for users to install or maintain software locally. Examples include Salesforce CRM, Google Workspace, and Microsoft Office 365.

In this scenario, the organization is using Salesforce CRM, which is a SaaS solution. Salesforce provides a complete customer relationship management (CRM) application that is accessible via a web browser, with no need for the organization to manage the underlying infrastructure or application code.

Why SaaS?

No Infrastructure Management: The customer does not need to worry about provisioning servers, databases, or networking components.

Fully Managed Application: Salesforce handles updates, patches, and maintenance, ensuring the application is always up-to-date.

Accessibility: Users can access Salesforce CRM from any device with an internet connection.

JNCIA Cloud Reference:

The JNCIA-Cloud certification emphasizes understanding the different cloud service models and their use cases. SaaS is particularly relevant in scenarios where organizations want to leverage pre-built applications without the complexity of managing infrastructure or development platforms.

For example, Juniper's cloud solutions often integrate with SaaS platforms like Salesforce to

provide secure connectivity and enhanced functionality. Understanding the role of SaaS in cloud architectures is essential for designing and implementing cloud-based solutions.

Juniper JNCIA-Cloud Study Guide: Cloud Service Models

Salesforce CRM Documentation

To Get Premium Files for JN0-214 Visit

https://www.p2pexams.com/products/jn0-214

For More Free Questions Visit

https://www.p2pexams.com/juniper/pdf/jn0-214

