Number: JN0-214
Passing Score: 800
Time Limit: 120
File Version: 3.0

Exam Code: JN0-214

Exam Name: Cloud, Associate

Exam A

QUESTION 1
Which two statements describe a multitenant cloud? (Choose two.)

A. Tenants are aware of other tenants using their shared resources.
B. Servers, network, and storage are separated per tenant.
C. The entities of each tenant are isolated from one another.
D. Multiple customers of a cloud vendor have access to their own dedicated hardware.

Correct Answer: C, D
Section:
Explanation:
A multitenant cloud is a cloud architecture where multiple customers (tenants) share the same physical infrastructure or platform while maintaining logical isolation. Let's analyze each statement:
A . Tenants are aware of other tenants using their shared resources.
Incorrect: In a multitenant cloud, tenants are logically isolated from one another. While they may share underlying physical resources (e.g., servers, storage), they are unaware of other tenants and cannot access their data or applications. This isolation ensures security and privacy.
B . Servers, network, and storage are separated per tenant.
Incorrect: In a multitenant cloud, resources such as servers, network, and storage are shared among tenants. The separation is logical, not physical. For example, virtualization technologies like hypervisors and software-defined networking (SDN) are used to create isolated environments for each tenant.
C . The entities of each tenant are isolated from one another.
Correct: Logical isolation is a fundamental characteristic of multitenancy. Each tenant's data, applications, and configurations are isolated to prevent unauthorized access or interference. Technologies like virtual private clouds (VPCs) and network segmentation ensure this isolation.
D . Multiple customers of a cloud vendor have access to their own dedicated hardware.
Correct: While multitenancy typically involves shared resources, some cloud vendors offer dedicated hardware options for customers with strict compliance or performance requirements. For example, AWS offers 'Dedicated Instances' or 'Dedicated Hosts,' which provide dedicated physical servers for specific tenants within a multitenant environment.
JNCIA Cloud
Reference:
The Juniper Networks Certified Associate - Cloud (JNCIA-Cloud) curriculum discusses multitenancy as a key feature of cloud computing. Multitenancy enables efficient resource utilization and cost savings by allowing multiple tenants to share infrastructure while maintaining isolation.
For example, Juniper Contrail supports multitenancy by providing features like VPCs, network overlays, and tenant isolation. These capabilities ensure that each tenant has a secure and independent environment within a shared infrastructure.
NIST Cloud Computing Reference Architecture
Juniper JNCIA-Cloud Study Guide: Multitenancy

QUESTION 2
What are the two characteristics of the Network Functions Virtualization (NFV) framework? (Choose two.)
A It implements virtualized tunnel endpoints
B. It decouples the network software from the hardware.
C. It implements virtualized network functions
D. It decouples the network control plane from the forwarding plane.

A.

Correct Answer:
Section:

Explanation:
Network Functions Virtualization (NFV) is a framework designed to virtualize network services traditionally run on proprietary hardware. NFV aims to reduce costs, improve scalability, and increase flexibility by decoupling network functions from dedicated hardware appliances. Let's analyze each statement:
A . It implements virtualized tunnel endpoints.
Incorrect: While NFV can support virtualized tunnel endpoints (e.g., VXLAN gateways), this is not a defining characteristic of the NFV framework. Tunneling protocols are typically associated with SDN or overlay networks rather than NFV itself.
B . It decouples the network software from the hardware.
Correct: One of the primary goals of NFV is to separate network functions (e.g., firewalls, load balancers, routers) from proprietary hardware. Instead, these functions are implemented as software running on standard servers or virtual machines.
C . It implements virtualized network functions.
Correct: NFV replaces traditional hardware-based network appliances with virtualized network functions (VNFs). Examples include virtual firewalls, virtual routers, and virtual load balancers. These VNFs run on commodity hardware and are managed through orchestration platforms.
D . It decouples the network control plane from the forwarding plane.
Incorrect: Decoupling the control plane from the forwarding plane is a characteristic of Software-Defined Networking (SDN), not NFV. While NFV and SDN are complementary technologies, they serve different purposes. NFV focuses on virtualizing network functions, while SDN focuses on programmable network control.
JNCIA Cloud
Reference:
The JNCIA-Cloud certification covers NFV as part of its discussion on cloud architectures and virtualization. NFV is particularly relevant in modern cloud environments because it enables flexible and scalable deployment of network services without reliance on specialized hardware.
For example, Juniper Contrail integrates with NFV frameworks to deploy and manage VNFs, enabling service providers to deliver network services efficiently and cost-effectively.
ETSI NFV Framework Documentation
Juniper JNCIA-Cloud Study Guide: Network Functions Virtualization

QUESTION 3
What is the name of the Docker container runtime?

A. docker_cli
B. containerd
C. dockerd
D. cri-o

Correct Answer: B
Section:
Explanation:
Docker is a popular containerization platform that relies on a container runtime to manage the lifecycle of containers. The container runtime is responsible for tasks such as creating, starting, stopping, and managing containers. Let's analyze each option:
A . docker_cli
Incorrect: The Docker CLI (Command Line Interface) is a tool used to interact with the Docker daemon (dockerd). It is not a container runtime but rather a user interface for managing Docker containers.
B . containerd
Correct: containerd is the default container runtime used by Docker. It is a lightweight, industry-standard runtime that handles low-level container management tasks, such as image transfer, container execution, and lifecycle management. Docker delegates these tasks to containerd through the Docker daemon.
C . dockerd
Incorrect: dockerd is the Docker daemon, which manages Docker objects such as images, containers, networks, and volumes. While dockerd interacts with the container runtime, it is not the runtime itself.
D . cri-o
Incorrect: cri-o is an alternative container runtime designed specifically for Kubernetes. It implements the Kubernetes Container Runtime Interface (CRI) and is not used by Docker.
Why containerd?
Industry Standard: containerd is a widely adopted container runtime that adheres to the Open Container Initiative (OCI) standards.
Integration with Docker: Docker uses containerd as its default runtime, making it the correct answer in this context.
JNCIA Cloud

Reference:
The JNCIA-Cloud certification emphasizes understanding containerization technologies and their components. Docker and its runtime (containerd) are foundational tools in modern cloud environments, enabling lightweight, portable, and scalable application deployment.
For example, Juniper Contrail integrates with container orchestration platforms like Kubernetes, which often use containerd as the underlying runtime. Understanding container runtimes is essential for managing containerized workloads in cloud environments.
Docker Documentation: Container Runtimes
Open Container Initiative (OCI) Standards
Juniper JNCIA-Cloud Study Guide: Containerization

QUESTION 4
Which command should you use to obtain low-level information about Docker objects?

A. docker info <OBJECT_NAME>
B. docker inspect <OBJECT_NAME>
C. docker container <OBJECT_NAME>
D. docker system <OBJECT_NAME>

Correct Answer: B
Section:
Explanation:
Docker provides various commands to manage and interact with Docker objects such as containers, images, networks, and volumes. To obtain low-level information about these objects, the docker inspect command is used.
Let's analyze each option:
A . docker info <OBJECT_NAME>
Incorrect: The docker info command provides high-level information about the Docker daemon itself, such as the number of containers, images, and system-wide configurations. It does not provide detailed information about specific Docker objects.
B . docker inspect <OBJECT_NAME>
Correct: The docker inspect command retrieves low-level metadata and configuration details about Docker objects (e.g., containers, images, networks, volumes). This includes information such as IP addresses, mount points, environment variables, and network settings. It outputs the data in JSON format for easy parsing and analysis.
C . docker container <OBJECT_NAME>
Incorrect: The docker container command is a parent command for managing containers (e.g., docker container ls, docker container start). It does not directly provide low-level information about a specific container.
D . docker system <OBJECT_NAME>
Incorrect: The docker system command is used for system-wide operations, such as pruning unused resources (docker system prune) or viewing disk usage (docker system df). It does not provide low-level details about specific Docker objects.
Why docker inspect?
Detailed Metadata: docker inspect is specifically designed to retrieve comprehensive, low-level information about Docker objects.
Versatility: It works with multiple object types, including containers, images, networks, and volumes.
JNCIA Cloud
Reference:
The JNCIA-Cloud certification covers Docker as part of its containerization curriculum. Understanding how to use Docker commands like docker inspect is essential for managing and troubleshooting containerized applications in cloud environments.
For example, Juniper Contrail integrates with container orchestration platforms like Kubernetes, which rely on Docker for container management. Proficiency with Docker commands ensures effective operation and debugging of containerized workloads.
Docker Documentation: docker inspect Command
Juniper JNCIA-Cloud Study Guide: Containerization

QUESTION 5
You are asked to provision a bare-metal server using OpenStack.
Which service is required to satisfy this requirement?

A. Ironic
B. Zun
C. Trove
D. Magnum

Correct Answer: A
Section:
Explanation:
OpenStack is an open-source cloud computing platform that provides various services for managing compute, storage, and networking resources. To provision a bare-metal server in OpenStack, the Ironic service is required.
Let's analyze each option:
A . Ironic
Correct: OpenStack Ironic is a bare-metal provisioning service that allows you to manage and provision physical servers as if they were virtual machines. It automates tasks such as hardware discovery, configuration, and deployment of operating systems on bare-metal servers.
B . Zun
Incorrect: OpenStack Zun is a container service that manages the lifecycle of containers. It is unrelated to bare-metal provisioning.
C . Trove
Incorrect: OpenStack Trove is a Database as a Service (DBaaS) solution that provides managed database instances. It does not handle bare-metal provisioning.
D . Magnum
Incorrect: OpenStack Magnum is a container orchestration service that supports Kubernetes, Docker Swarm, and other container orchestration engines. It is focused on containerized workloads, not bare-metal servers.
Why Ironic?
Purpose-Built for Bare-Metal: Ironic is specifically designed to provision and manage bare-metal servers, making it the correct choice for this requirement.
Automation: Ironic automates the entire bare-metal provisioning process, including hardware discovery, configuration, and OS deployment.
JNCIA Cloud
Reference:
The JNCIA-Cloud certification covers OpenStack as part of its cloud infrastructure curriculum. Understanding OpenStack services like Ironic is essential for managing bare-metal and virtualized environments in cloud deployments.
For example, Juniper Contrail integrates with OpenStack to provide networking and security for both virtualized and bare-metal workloads. Proficiency with OpenStack services ensures efficient management of diverse cloud resources.
OpenStack Documentation: Ironic Bare-Metal Provisioning
Juniper JNCIA-Cloud Study Guide: OpenStack Services

QUESTION 6
Which two statements are correct about an underlay network? (Choose two.)

A. An underlay network can be built using either Layer 2 or Layer 3 connectivity.
B. A Layer 3 underlay network uses routing protocols to provide IP connectivity.
C. The underlay network is the virtual network used to connect multiple virtual machines (VMs).
D. The underlay network is built using encapsulations tunnels.

Correct Answer: A, B
Section:
Explanation:
An underlay network refers to the physical or logical network infrastructure that provides the foundation for overlay networks in cloud environments. It handles the actual transport of data between devices and serves as the backbone for cloud architectures. Let's analyze each statement:
A . An underlay network can be built using either Layer 2 or Layer 3 connectivity.
Correct: Underlay networks can operate at both Layer 2 (switching) and Layer 3 (routing). For example:
Layer 2: Uses Ethernet switching to forward traffic within a single broadcast domain.
Layer 3: Uses IP routing to forward traffic across multiple subnets or networks.
B . A Layer 3 underlay network uses routing protocols to provide IP connectivity.

Correct: In a Layer 3 underlay network, routing protocols like OSPF, BGP, or EIGRP are used to exchange routing information and ensure IP connectivity between devices. This is common in large-scale cloud environments where scalability and segmentation are critical.
C . The underlay network is the virtual network used to connect multiple virtual machines (VMs).
Incorrect: The underlay network is the physical or logical infrastructure that supports the overlay network. The overlay network, on the other hand, is the virtual network used to connect VMs, containers, or other endpoints. The underlay provides the foundation, while the overlay adds abstraction and flexibility.
D . The underlay network is built using encapsulations tunnels.
Incorrect: Encapsulation tunnels (e.g., VXLAN, GRE) are used in overlay networks, not underlay networks. The underlay network provides the physical or logical transport layer, while the overlay network uses tunnels to create virtualized network segments.
Why These Answers?
Layer 2 and Layer 3 Flexibility: The underlay network must support both switching and routing to accommodate diverse workloads and topologies.
Routing Protocols in Layer 3: Routing protocols are essential for scalable and efficient IP connectivity in Layer 3 underlay networks.
JNCIA Cloud
Reference:
The JNCIA-Cloud certification covers underlay and overlay networks as part of its discussion on cloud architectures. Understanding the distinction between underlay and overlay networks is crucial for designing and managing cloud environments.
For example, Juniper Contrail uses an underlay network to provide the physical connectivity required for overlay networks. The underlay ensures reliable and scalable transport, while the overlay enables flexible virtualized networking.
Juniper JNCIA-Cloud Study Guide: Underlay and Overlay Networks
Network Virtualization Documentation

QUESTION 7
Which two statements are correct about Network Functions Virtualization (NFV)? (Choose two.)

A. the NFV framework explains how VNFs fits into the whole solution.
B. The NFV Infrastructure (NFVI) is a component of NFV.
C. The NFV Infrastructure (NFVI) is not a component of NFV.
D. The NFV framework is defined by the W3C.

Correct Answer: A, B
Section:
Explanation:
Network Functions Virtualization (NFV) is a framework designed to virtualize network services traditionally run on proprietary hardware. It decouples network functions from dedicated hardware appliances and implements them as software running on standard servers or virtual machines. Let's analyze each statement:
A . The NFV framework explains how VNFs fit into the whole solution.
Correct: The NFV framework provides a structured approach to deploying and managing Virtualized Network Functions (VNFs). It defines how VNFs interact with other components, such as the NFV Infrastructure (NFVI), Management and Orchestration (MANO), and the underlying hardware.
B . The NFV Infrastructure (NFVI) is a component of NFV.
Correct: The NFV Infrastructure (NFVI) is a critical part of the NFV architecture. It includes the physical and virtual resources (e.g., compute, storage, networking) that host and support VNFs. NFVI acts as the foundation for deploying and running virtualized network functions.
C . The NFV Infrastructure (NFVI) is not a component of NFV.
Incorrect: This statement contradicts the NFV architecture. NFVI is indeed a core component of NFV, providing the necessary infrastructure for VNFs.
D . The NFV framework is defined by the W3C.
Incorrect: The NFV framework is defined by the European Telecommunications Standards Institute (ETSI), not the W3C. ETSI's NFV Industry Specification Group (ISG) established the standards and architecture for NFV.
Why These Answers?
Framework The NFV framework provides a comprehensive view of how VNFs integrate into the overall solution, ensuring scalability and flexibility.
NFVI Role: NFVI is essential for hosting and supporting VNFs, making it a fundamental part of the NFV architecture.
JNCIA Cloud
Reference:
The JNCIA-Cloud certification covers NFV as part of its cloud infrastructure curriculum. Understanding the NFV framework and its components is crucial for deploying and managing virtualized network functions in cloud

environments.
For example, Juniper Contrail integrates with NFV frameworks to deploy and manage VNFs, enabling service providers to deliver network services efficiently and cost-effectively.
ETSI NFV Framework Documentation
Juniper JNCIA-Cloud Study Guide: Network Functions Virtualization

QUESTION 8
Which component of a software-defined networking (SDN) controller defines where data packets are forwarded by a network device?

A. the operational plane
B. the forwarding plane C the management plane
C. the control plane

Correct Answer:
Section:
Explanation:
Software-Defined Networking (SDN) separates the control plane from the data (forwarding) plane, enabling centralized control and programmability of network devices. Let's analyze each option:
A . the operational plane
Incorrect: The operational plane is not a standard term in SDN architecture. It may refer to monitoring or management tasks but does not define packet forwarding behavior.
B . the forwarding plane
Incorrect: The forwarding plane (also known as the data plane) is responsible for forwarding packets based on rules provided by the control plane. It does not define where packets are forwarded; it simply executes the instructions.
C . the management plane
Incorrect: The management plane handles device configuration, monitoring, and administrative tasks. It does not determine packet forwarding paths.
D . the control plane
Correct: The control plane is responsible for making decisions about where data packets are forwarded. In SDN, the control plane is centralized in the SDN controller, which calculates forwarding paths and communicates them to network devices via protocols like OpenFlow.
Why the Control Plane?
Centralized Decision-Making: The control plane determines the optimal paths for packet forwarding and updates the forwarding plane accordingly.
Programmability: SDN controllers allow administrators to programmatically define forwarding rules, enabling dynamic and flexible network configurations.
JNCIA Cloud
Reference:
The JNCIA-Cloud certification emphasizes understanding SDN architecture and its components. The separation of the control plane and forwarding plane is a foundational concept in SDN, enabling scalable and programmable networks.
For example, Juniper Contrail serves as an SDN controller, centralizing control over network devices and enabling advanced features like network automation and segmentation.
Open Networking Foundation (ONF) SDN Architecture
Juniper JNCIA-Cloud Study Guide: Software-Defined Networking

QUESTION 9
Which cloud automation tool uses YAML playbook to install software and tools on servers?

A. Python
B. Ansible
C. Terraform
D. Heat

Correct Answer: B
Section:
Explanation:
Cloud automation tools streamline the deployment and management of software, tools, and infrastructure in cloud environments. Let's analyze each option:

A . Python
Incorrect: Python is a general-purpose programming language, not a cloud automation tool. While Python scripts can be used for automation, it is not specifically designed for this purpose.
B . Ansible
Correct: Ansible is a popular automation tool that uses YAML-based playbooks to define and execute tasks. It automates the installation of software, configuration management, and application deployment on servers. Ansible's simplicity and agentless architecture make it widely adopted in cloud environments.
C . Terraform
Incorrect: Terraform is an infrastructure-as-code (IaC) tool used to provision and manage cloud infrastructure (e.g., virtual machines, networks, storage). It uses HashiCorp Configuration Language (HCL), not YAML, for defining configurations.
D . Heat
Incorrect: Heat is an orchestration tool in OpenStack that uses YAML templates to define and deploy cloud resources. While it supports YAML, it is specific to OpenStack and focuses on infrastructure provisioning rather than server-level software installation.
Why Ansible?
YAML Playbooks: Ansible uses YAML-based playbooks to define tasks, making it easy to read and write automation scripts.
Agentless Architecture: Ansible operates over SSH, eliminating the need for agents on target servers.
Versatility: Ansible can automate a wide range of tasks, from software installation to configuration management.
JNCIA Cloud
Reference:
The JNCIA-Cloud certification covers automation tools as part of its cloud operations curriculum. Tools like Ansible are essential for automating repetitive tasks and ensuring consistency in cloud environments.
For example, Juniper Contrail integrates with Ansible to automate the deployment and configuration of network services, enabling efficient management of cloud resources.
Ansible Documentation: YAML Playbooks
Juniper JNCIA-Cloud Study Guide: Automation Tools

QUESTION 10
What is the role of overlay tunnels in an overlay software-defined networking (SDN) solution?

A. The overlay tunnels provide optimization of traffic for performance and resilience.
B. The overlay tunnels provide load balancing and scale out for applications.
C. The overlay tunnels provide microsegmentation for workloads.
D. The overlay tunnels abstract the underlay network topology.

Correct Answer: D
Section:
Explanation:
In an overlay software-defined networking (SDN) solution, overlay tunnels play a critical role in abstracting the underlying physical network (underlay) from the virtualized network (overlay). Let's analyze each option:
A . The overlay tunnels provide optimization of traffic for performance and resilience.
Incorrect: While overlay tunnels can contribute to traffic optimization indirectly, their primary role is not performance or resilience. These aspects are typically handled by SDN controllers or other network optimization tools.
B . The overlay tunnels provide load balancing and scale out for applications.
Incorrect: Load balancing and scaling are functions of application-level services or SDN controllers, not the overlay tunnels themselves. Overlay tunnels focus on encapsulating traffic rather than managing application workloads.
C . The overlay tunnels provide microsegmentation for workloads.
Incorrect: Microsegmentation is achieved through policies and security rules applied at the overlay network level, not directly by the tunnels themselves. Overlay tunnels enable the transport of segmented traffic but do not enforce segmentation.
D . The overlay tunnels abstract the underlay network topology.
Correct: Overlay tunnels encapsulate traffic between endpoints (e.g., VMs, containers) and hide the complexity of the underlay network. This abstraction allows the overlay network to operate independently of the physical network topology, enabling flexibility and scalability.
Why This Answer?
Abstraction of Underlay: Overlay tunnels use encapsulation protocols like VXLAN, GRE, or MPLS to create virtualized networks that are decoupled from the physical infrastructure. This abstraction simplifies network management and enables advanced features like multi-tenancy and mobility.
JNCIA Cloud

Reference:
The JNCIA-Cloud certification covers overlay and underlay networks as part of its SDN curriculum. Understanding the role of overlay tunnels is essential for designing and managing virtualized networks in cloud environments. For example, Juniper Contrail uses overlay tunnels to provide connectivity between virtual machines (VMs) and containers, abstracting the physical network and enabling seamless communication across distributed environments.
Juniper JNCIA-Cloud Study Guide: Overlay Networks
Network Virtualization Documentation

QUESTION 11
Which two CPU flags indicate virtualization? (Choose two.)

A. lvm
B. vmx
C. xvm
D. kvm

Correct Answer: B, D
Section:
Explanation:
CPU flags indicate hardware support for specific features, including virtualization. Let's analyze each option:
A . lvm
Incorrect: LVM (Logical Volume Manager) is a storage management technology used in Linux systems. It is unrelated to CPU virtualization.
B . vmx
Correct: The vmx flag indicates Intel Virtualization Technology (VT-x), which provides hardware-assisted virtualization capabilities. This feature is essential for running hypervisors like VMware ESXi, KVM, and Hyper-V.
C . xvm
Incorrect: xvm is not a recognized CPU flag for virtualization. It may be a misinterpretation or typo.
D . kvm
Correct: The kvm flag indicates Kernel-based Virtual Machine (KVM) support, which is a Linux kernel module that leverages hardware virtualization extensions (e.g., Intel VT-x or AMD-V) to run virtual machines. While kvm itself is not a CPU flag, it relies on hardware virtualization features like vmx (Intel) or svm (AMD).
Why These Answers?
Hardware Virtualization Support: Both vmx (Intel VT-x) and kvm (Linux virtualization) are directly related to CPU virtualization. These flags enable efficient execution of virtual machines by offloading tasks to the CPU.
JNCIA Cloud
Reference:
The JNCIA-Cloud certification emphasizes understanding virtualization technologies, including hardware-assisted virtualization. Recognizing CPU flags like vmx and kvm is crucial for deploying and troubleshooting virtualized environments.
For example, Juniper Contrail integrates with hypervisors like KVM to manage virtualized workloads in cloud environments. Ensuring hardware virtualization support is a prerequisite for deploying such solutions.
Intel Virtualization Technology Documentation
KVM Documentation
Juniper JNCIA-Cloud Study Guide: Virtualization

QUESTION 12
Which statement about software-defined networking is true?

A. It must manage networks through the use of containers and repositories.
B. It manages networks by separating the data forwarding plane from the control plane.
C. It applies security policies individually to each separate node.
D. It manages networks by merging the data forwarding plane with the control plane.

Correct Answer: B
Section:

Explanation:
Software-Defined Networking (SDN) is a revolutionary approach to network management that separates the control plane from the data (forwarding) plane. Let's analyze each option:
A . It must manage networks through the use of containers and repositories.
Incorrect: While containers and repositories are important in cloud-native environments, they are not a requirement for SDN. SDN focuses on programmability and centralized control, not containerization.
B . It manages networks by separating the data forwarding plane from the control plane.
Correct: SDN separates the control plane (decision-making) from the data forwarding plane (packet forwarding). This separation enables centralized control, programmability, and dynamic network management.
C . It applies security policies individually to each separate node.
Incorrect: SDN applies security policies centrally through the SDN controller, not individually to each node. Centralized policy enforcement is one of the key advantages of SDN.
D . It manages networks by merging the data forwarding plane with the control plane.
Incorrect: Merging the forwarding and control planes contradicts the fundamental principle of SDN. The separation of these planes is what enables SDN's flexibility and programmability.
Why This Answer?
Separation of Planes: By decoupling the control plane from the forwarding plane, SDN enables centralized control over network devices. This architecture simplifies network management, improves scalability, and supports automation.
JNCIA Cloud
Reference:
The JNCIA-Cloud certification covers SDN as a core concept in cloud networking. Understanding the separation of the control and forwarding planes is essential for designing and managing modern cloud environments.
For example, Juniper Contrail serves as an SDN controller, centralizing control over network devices and enabling advanced features like network automation and segmentation.
Open Networking Foundation (ONF) SDN Architecture
Juniper JNCIA-Cloud Study Guide: Software-Defined Networking

QUESTION 13
Which type of virtualization provides containerization and uses a microservices architecture?

A. hardware-assisted virtualization
B. OS-level virtualization
C. full virtualization
D. paravirtualization

Correct Answer: B
Section:
Explanation:
Virtualization technologies enable the creation of isolated environments for running applications or services. Let's analyze each option:
A . hardware-assisted virtualization
Incorrect: Hardware-assisted virtualization (e.g., Intel VT-x, AMD-V) provides support for running full virtual machines (VMs) on physical hardware. It is not related to containerization or microservices architecture.
B . OS-level virtualization
Correct: OS-level virtualization enables containerization , where multiple isolated user-space instances (containers) run on a single operating system kernel. Containers are lightweight and share the host OS kernel, making them ideal for microservices architectures. Examples include Docker and Kubernetes.
C . full virtualization
Incorrect: Full virtualization involves running a complete guest operating system on top of a hypervisor (e.g., VMware ESXi, KVM). While it provides strong isolation, it is not as lightweight or efficient as containerization for microservices.
D . paravirtualization
Incorrect: Paravirtualization involves modifying the guest operating system to communicate directly with the hypervisor. Like full virtualization, it is used for running VMs, not containers.
Why OS-Level Virtualization?
Containerization: OS-level virtualization creates isolated environments (containers) that share the host OS kernel but have their own file systems, libraries, and configurations.
Microservices Architecture: Containers are well-suited for deploying microservices because they are lightweight, portable, and scalable.
JNCIA Cloud
Reference:
The JNCIA-Cloud certification emphasizes understanding virtualization technologies, including OS-level virtualization. Containerization is a key component of modern cloud-native architectures, enabling efficient deployment of microservices.

For example, Juniper Contrail integrates with Kubernetes to manage containerized workloads in cloud environments. OS-level virtualization is fundamental to this integration.
Docker Documentation: Containerization
Juniper JNCIA-Cloud Study Guide: Virtualization

QUESTION 14
Which feature of Linux enables kernel-level isolation of global resources?

A. ring protection
B. stack protector
C. namespaces
D. shared libraries

Correct Answer: C
Section:
Explanation:
Linux provides several mechanisms for isolating resources and ensuring security. Let's analyze each option:
A . ring protection
Incorrect: Ring protection refers to CPU privilege levels (e.g., Rings 0--3) that control access to system resources. While important for security, it does not provide kernel-level isolation of global resources.
B . stack protector
Incorrect: Stack protector is a compiler feature that helps prevent buffer overflow attacks by adding guard variables to function stacks. It is unrelated to resource isolation.
C . namespaces
Correct: Namespaces are a Linux kernel feature that provides kernel-level isolation of global resources such as process IDs, network interfaces, mount points, and user IDs. Each namespace has its own isolated view of these resources, enabling features like containerization.
D . shared libraries
Incorrect: Shared libraries allow multiple processes to use the same code, reducing memory usage. They do not provide isolation or security.
Why Namespaces?
Resource Isolation: Namespaces isolate processes, networks, and other resources, ensuring that changes in one namespace do not affect others.
Containerization Foundation: Namespaces are a core technology behind containerization platforms like Docker and Kubernetes, enabling lightweight and secure environments.
JNCIA Cloud
Reference:
The JNCIA-Cloud certification covers Linux fundamentals, including namespaces, as part of its containerization curriculum. Understanding namespaces is essential for managing containerized workloads in cloud environments.
For example, Juniper Contrail leverages namespaces to isolate network resources in containerized environments, ensuring secure and efficient operation.
Linux Kernel Documentation: Namespaces
Juniper JNCIA-Cloud Study Guide: Linux Features

QUESTION 15
Which two tools are used to deploy a Kubernetes environment for testing and development purposes? (Choose two.)

A. OpenStack
B. kind
C. oc
D. minikube

Correct Answer: B, D
Section:
Explanation:
Kubernetes is a popular container orchestration platform used for deploying and managing containerized applications. Several tools are available for setting up Kubernetes environments for testing and development purposes.
Let's analyze each option:
A . OpenStack

Incorrect: OpenStack is an open-source cloud computing platform used for managing infrastructure resources (e.g., compute, storage, networking). It is not specifically designed for deploying Kubernetes environments.
B . kind
Correct: kind (Kubernetes IN Docker) is a tool for running local Kubernetes clusters using Docker containers as nodes. It is lightweight and ideal for testing and development purposes.
C . oc
Incorrect: oc is the command-line interface (CLI) for OpenShift, a Kubernetes-based container platform. While OpenShift can be used to deploy Kubernetes environments, oc itself is not a tool for setting up standalone Kubernetes clusters.
D . minikube
Correct: minikube is a tool for running a single-node Kubernetes cluster locally on your machine. It is widely used for testing and development due to its simplicity and ease of setup.
Why These Tools?
kind: Ideal for simulating multi-node Kubernetes clusters in a lightweight environment.
minikube: Perfect for beginners and developers who need a simple, single-node Kubernetes cluster for experimentation.
JNCIA Cloud
Reference:
The JNCIA-Cloud certification covers Kubernetes as part of its container orchestration curriculum. Tools like kind and minikube are essential for learning and experimenting with Kubernetes in local environments.
For example, Juniper Contrail integrates with Kubernetes to provide advanced networking and security features for containerized workloads. Proficiency with Kubernetes tools ensures effective operation and troubleshooting.
Kubernetes Documentation: kind and minikube
Juniper JNCIA-Cloud Study Guide: Kubernetes

QUESTION 16
What are two Kubernetes worker node components? (Choose two.)

A. kube-apiserver
B. kubelet
C. kube-scheduler
D. kube-proxy

Correct Answer: B, D
Section:
Explanation:
Kubernetes worker nodes are responsible for running containerized applications and managing the workloads assigned to them. Each worker node contains several key components that enable it to function within a Kubernetes cluster. Let's analyze each option:
A . kube-apiserver
Incorrect: The kube-apiserver is a control plane component, not a worker node component. It serves as the front-end for the Kubernetes API, handling communication between the control plane and worker nodes.
B . kubelet
Correct: The kubelet is a critical worker node component. It ensures that containers are running in the desired state by interacting with the container runtime (e.g., containerd). It communicates with the control plane to receive instructions and report the status of pods.
C . kube-scheduler
Incorrect: The kube-scheduler is a control plane component responsible for assigning pods to worker nodes based on resource availability and other constraints. It does not run on worker nodes.
D . kube-proxy
Correct: The kube-proxy is another essential worker node component. It manages network communication for services and pods by implementing load balancing and routing rules. It ensures that traffic is correctly forwarded to the appropriate pods.
Why These Components?
kubelet: Ensures that containers are running as expected and maintains the desired state of pods.
kube-proxy: Handles networking and enables communication between services and pods within the cluster.
JNCIA Cloud
Reference:
The JNCIA-Cloud certification covers Kubernetes architecture, including the roles of worker node components. Understanding the functions of kubelet and kube-proxy is crucial for managing Kubernetes clusters and troubleshooting issues.
For example, Juniper Contrail integrates with Kubernetes to provide advanced networking and security features. Proficiency with worker node components ensures efficient operation of containerized workloads.

Kubernetes Documentation: Worker Node Components
Juniper JNCIA-Cloud Study Guide: Kubernetes Architecture

QUESTION 17
Which term identifies to which network a virtual machine interface is connected?

A. virtual network ID
B. machine access control (MAC)
C. Virtual Extensible LAN
D. virtual tunnel endpoint (VTEP)

Correct Answer: A
Section:
Explanation:
In cloud environments, virtual machines (VMs) connect to virtual networks to enable communication. Identifying the network to which a VM interface is connected is essential for proper configuration and isolation. Let's analyze each option:
A . virtual network ID
Correct: The virtual network ID uniquely identifies the virtual network to which a VM interface is connected. This ID is used to logically group VMs and ensure they can communicate within the same network while maintaining isolation from other networks.
B . machine access control (MAC)
Incorrect: The MAC address is a hardware identifier for a network interface card (NIC). While it is unique to each interface, it does not identify the network to which the VM is connected.
C . Virtual Extensible LAN (VXLAN)
Incorrect: VXLAN is a tunneling protocol used to create overlay networks in cloud environments. While VXLAN encapsulates traffic, it does not directly identify the network to which a VM interface is connected.
D . virtual tunnel endpoint (VTEP)
Incorrect: A VTEP is a component of overlay networks (e.g., VXLAN) that encapsulates and decapsulates traffic. It is used to establish tunnels but does not identify the virtual network itself.
Why Virtual Network ID?
Logical Isolation: The virtual network ID ensures that VMs are logically grouped into isolated networks, enabling secure and efficient communication.
Scalability: Virtual networks allow cloud environments to scale by supporting multiple isolated networks within the same infrastructure.
JNCIA Cloud
Reference:
The JNCIA-Cloud certification emphasizes understanding virtual networking concepts, including virtual networks and their identifiers. Virtual network IDs are fundamental to cloud architectures, enabling multi-tenancy and network segmentation.
For example, Juniper Contrail uses virtual network IDs to manage connectivity and isolation for VMs in cloud environments. Proper configuration of virtual networks ensures seamless communication and security.
Virtual Networking Documentation
Juniper JNCIA-Cloud Study Guide: Virtual Networks

QUESTION 18
Click the Exhibit button.

```
web-service.yaml
---
apiVersion: v1
kind: Service
metadata:
  name: web-service
spec:
  type: NodePort
  selector:
    app: web-app
  ports:
    - protocol: tcp
      port: 8080
      targetPort: 5000
      nodePort: 31000
```

Referring to the exhibit, which port number would external users use to access the WEB application?

A. 80
B. 8080
C. 31000
D. 5000

Correct Answer: C
Section:
Explanation:
The YAML file provided in the exhibit defines a Kubernetes Service object of type NodePort. Let's break down the key components of the configuration and analyze how external users access the WEB application:
Key Fields in the YAML File:
type: NodePort:
This specifies that the service is exposed on a static port on each node in the cluster. External users can access the service using the node's IP address and the assigned nodePort.
port: 8080:
This is the port on which the service is exposed internally within the Kubernetes cluster. Other services or pods within the cluster can communicate with this service using port 8080.
targetPort: 5000:
This is the port on which the actual application (WEB application) is running inside the pod. The service forwards traffic from port: 8080 to targetPort: 5000.
nodePort: 31000:
This is the port on the node (host machine) where the service is exposed externally. External users will use this port to access the WEB application.
How External Users Access the WEB Application:
External users access the WEB application using the node's IP address and the nodePort value (31000).
The Kubernetes service listens on this port and forwards incoming traffic to the appropriate pods running the WEB application.
Why Not Other Options?
A . 80: Port 80 is commonly used for HTTP traffic, but it is not specified in the YAML file. The service does not expose port 80 externally.
B . 8080: Port 8080 is the internal port used within the Kubernetes cluster. It is not the port exposed to external users.
D . 5000: Port 5000 is the target port where the application runs inside the pod. It is not directly accessible to external users.

Why 31000?
NodePort Service Type: The NodePort service type exposes the application on a high-numbered port (default range: 30000--32767) on each node in the cluster.
External Accessibility: External users must use the nodePort value (31000) along with the node's IP address to access the WEB application.
JNCIA Cloud
Reference:
The JNCIA-Cloud certification covers Kubernetes networking concepts, including service types like ClusterIP, NodePort, and LoadBalancer. Understanding how NodePort services work is essential for exposing applications to external users in Kubernetes environments.
For example, Juniper Contrail integrates with Kubernetes to provide advanced networking features, such as load balancing and network segmentation, for services like the one described in the exhibit.
Kubernetes Documentation: Service Types
Juniper JNCIA-Cloud Study Guide: Kubernetes Networking

QUESTION 19
You must provide tunneling in the overlay that supports multipath capabilities.
Which two protocols provide this function? (Choose two.)

A. MPLSoGRE
B. VXLAN
C. VPN
D. MPLSoUDP

Correct Answer: B, D
Section:
Explanation:
In cloud networking, overlay networks are used to create virtualized networks that abstract the underlying physical infrastructure. To support multipath capabilities , certain protocols provide efficient tunneling mechanisms.
Let's analyze each option:
A . MPLSoGRE
Incorrect: MPLS over GRE (MPLSoGRE) is a tunneling protocol that encapsulates MPLS packets within GRE tunnels. While it supports MPLS traffic, it does not inherently provide multipath capabilities.
B . VXLAN
Correct: VXLAN (Virtual Extensible LAN) is an overlay protocol that encapsulates Layer 2 Ethernet frames within UDP packets. It supports multipath capabilities by leveraging the Equal-Cost Multi-Path (ECMP) routing in the underlay network. VXLAN is widely used in cloud environments for extending Layer 2 networks across data centers.
C . VPN
Incorrect: Virtual Private Networks (VPNs) are used to securely connect remote networks or users over public networks. They do not inherently provide multipath capabilities or overlay tunneling for virtual networks.
D . MPLSoUDP
Correct: MPLS over UDP (MPLSoUDP) is a tunneling protocol that encapsulates MPLS packets within UDP packets. Like VXLAN, it supports multipath capabilities by utilizing ECMP in the underlay network. MPLSoUDP is often used in service provider environments for scalable and flexible network architectures.
Why These Protocols?
VXLAN: Provides Layer 2 extension and supports multipath forwarding, making it ideal for large-scale cloud deployments.
MPLSoUDP: Combines the benefits of MPLS with UDP encapsulation, enabling efficient multipath routing in overlay networks.
JNCIA Cloud
Reference:
The JNCIA-Cloud certification covers overlay networking protocols like VXLAN and MPLSoUDP as part of its curriculum on cloud architectures. Understanding these protocols is essential for designing scalable and resilient virtual networks.
For example, Juniper Contrail uses VXLAN to extend virtual networks across distributed environments, ensuring seamless communication and high availability.
VXLAN RFC 7348
MPLSoUDP Documentation
Juniper JNCIA-Cloud Study Guide: Overlay Networking

QUESTION 20
Which two statements about containers are true? (Choose two.)

A. Containers contain executables, libraries, configuration files, and an operating system.
B. Containers package the entire runtime environment of an application, including its dependencies.
C. Containers can only run on a system with a Type 2 hypervisor.
D. Containers share the use of the underlying system's kernel.

Correct Answer: B, D
Section:
Explanation:
Containers are a lightweight form of virtualization that enable the deployment of applications in isolated environments. Let's analyze each statement:
A . Containers contain executables, libraries, configuration files, and an operating system.
Incorrect: Containers do not include a full operating system. Instead, they share the host system's kernel and only include the application and its dependencies (e.g., libraries, binaries, and configuration files).
B . Containers package the entire runtime environment of an application, including its dependencies.
Correct: Containers bundle the application code, runtime, libraries, and configuration files into a single package. This ensures consistency across different environments and eliminates issues caused by differences in dependencies.
C . Containers can only run on a system with a Type 2 hypervisor.
Incorrect: Containers do not require a hypervisor. They run directly on the host operating system and share the kernel. Hypervisors (Type 1 or Type 2) are used for virtual machines, not containers.
D . Containers share the use of the underlying system's kernel.
Correct: Containers leverage the host operating system's kernel, which allows them to be lightweight and efficient. Each container has its own isolated user space but shares the kernel with other containers.
Why These Statements?
Runtime Environment Packaging: Containers ensure portability and consistency by packaging everything an application needs to run.
Kernel Sharing: By sharing the host kernel, containers consume fewer resources compared to virtual machines, which require separate operating systems.
JNCIA Cloud
Reference:
The JNCIA-Cloud certification emphasizes understanding containerization technologies, including Docker and Kubernetes. Containers are a fundamental component of modern cloud-native architectures.
For example, Juniper Contrail integrates with Kubernetes to manage containerized workloads, leveraging the lightweight and portable nature of containers.
Docker Documentation: Container Basics
Juniper JNCIA-Cloud Study Guide: Containerization

QUESTION 21
Which method is used to extend virtual networks between physical locations?

A. encapsulations
B. encryption
C. clustering
D. load-balancing

Correct Answer: A
Section:
Explanation:
To extend virtual networks between physical locations, a mechanism is needed to transport network traffic across different sites while maintaining isolation and connectivity. Let's analyze each option:
A . encapsulations
Correct: Encapsulation is the process of wrapping network packets in additional headers to create tunnels. Protocols like VXLAN, GRE, and MPLS are commonly used to extend virtual networks between physical locations by encapsulating traffic and transporting it over the underlay network.
B . encryption
Incorrect: Encryption secures data during transmission but does not inherently extend virtual networks. While encryption can be used alongside encapsulation for secure communication, it is not the primary method for extending networks.
C . clustering
Incorrect: Clustering refers to grouping multiple servers or devices to work together as a single system. It is unrelated to extending virtual networks between physical locations.

D . load-balancing
Incorrect: Load balancing distributes traffic across multiple servers or paths to optimize performance. While important for scalability, it does not extend virtual networks.
Why Encapsulation?
Tunneling Mechanism: Encapsulation protocols like VXLAN and GRE create overlay networks that span multiple physical locations, enabling seamless communication between virtual networks.
Isolation and Scalability: Encapsulation ensures that virtual networks remain isolated and scalable, even when extended across geographically dispersed sites.
JNCIA Cloud
Reference:
The JNCIA-Cloud certification covers overlay networking and encapsulation as part of its curriculum on cloud architectures. Understanding how encapsulation works is essential for designing and managing distributed virtual networks.
For example, Juniper Contrail uses encapsulation protocols like VXLAN to extend virtual networks across data centers, ensuring consistent connectivity and isolation.
VXLAN RFC 7348
GRE Tunneling Documentation
Juniper JNCIA-Cloud Study Guide: Overlay Networking

QUESTION 22

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

Correct Answer: C
Section:
Explanation:
This question seems to be asking about a Kubernetes component that is responsible for running containers. Let's analyze each option:
A . kubelet
Incorrect: The kubelet is responsible for managing the state of pods and containers on a worker node. It ensures that containers are running as expected but does not directly execute or run the containers.
B . kube-proxy
Incorrect: The kube-proxy manages network communication for services and pods by implementing load balancing and routing rules. It does not handle the execution of containers.
C . container runtime
Correct: The container runtime (e.g., containerd, cri-o) is the component that actually runs and manages containers on a Kubernetes node. It interacts with the operating system to start, stop, and manage containerized applications.
D . kube controller
Incorrect: The kube controller is part of the control plane and ensures that the desired state of the cluster (e.g., number of replicas) is maintained. It does not directly run containers.
Why Container Runtime?
Execution of Containers: The container runtime is responsible for pulling container images, starting containers, and managing their lifecycle.
Integration with Kubernetes: Kubernetes communicates with the container runtime through the Container Runtime Interface (CRI).
JNCIA Cloud
Reference:
The JNCIA-Cloud certification covers Kubernetes architecture, including the role of the container runtime. Understanding how containers are executed is essential for managing Kubernetes clusters.
For example, Juniper Contrail integrates with Kubernetes to provide networking and security for containerized workloads, relying on the container runtime to execute applications.
Kubernetes Documentation: Container Runtimes
Juniper JNCIA-Cloud Study Guide: Kubernetes Architecture

QUESTION 23

A. Pods are allowed to communicate if they are only in the default namespaces.
B. Pods are not allowed to communicate if they are in different namespaces.
C. Full communication between pods is allowed across nodes without requiring NAT.

D. Each pod has its own IP address in a flat, shared networking namespace.

Correct Answer: C, D
Section:
Explanation:
Kubernetes networking is designed to provide seamless communication between pods, regardless of their location in the cluster. Let's analyze each statement:
A . Pods are allowed to communicate if they are only in the default namespaces.
Incorrect: Pods can communicate with each other regardless of the namespace they belong to. Namespaces are used for logical grouping and isolation but do not restrict inter-pod communication.
B . Pods are not allowed to communicate if they are in different namespaces.
Incorrect: Pods in different namespaces can communicate with each other as long as there are no network policies restricting such communication. Namespaces do not inherently block communication.
C . Full communication between pods is allowed across nodes without requiring NAT.
Correct: Kubernetes networking is designed so that pods can communicate directly with each other across nodes without Network Address Translation (NAT). Each pod has a unique IP address, and the underlying network ensures direct communication.
D . Each pod has its own IP address in a flat, shared networking namespace.
Correct: In Kubernetes, each pod is assigned a unique IP address in a flat network space. This allows pods to communicate with each other as if they were on the same network, regardless of the node they are running on.
Why These Statements?
Flat Networking Model: Kubernetes uses a flat networking model where each pod gets its own IP address, simplifying communication and eliminating the need for NAT.
Cross-Node Communication: The design ensures that pods can communicate seamlessly across nodes, enabling scalable and distributed applications.
JNCIA Cloud
Reference:
The JNCIA-Cloud certification emphasizes Kubernetes networking concepts, including pod-to-pod communication and the flat networking model. Understanding these principles is essential for designing and managing Kubernetes clusters.
For example, Juniper Contrail provides advanced networking features for Kubernetes, ensuring efficient and secure pod communication across nodes.
Kubernetes Documentation: Networking Model
Juniper JNCIA-Cloud Study Guide: Kubernetes Networking

QUESTION 24
Which Kubernetes component guarantees the availability of ReplicaSet pods on one or more nodes?

A. kube-proxy
B. kube-scheduler
C. kube controller
D. kubelet

Correct Answer: C
Section:
Explanation:
Kubernetes components work together to ensure the availability and proper functioning of resources like ReplicaSets. Let's analyze each option:
A . kube-proxy
Incorrect: The kube-proxy manages network communication for services and pods by implementing load balancing and routing rules. It does not guarantee the availability of ReplicaSet pods.
B . kube-scheduler
Incorrect: The kube-scheduler is responsible for assigning pods to nodes based on resource availability and other constraints. While it plays a role in pod placement, it does not ensure the availability of ReplicaSet pods.
C . kube controller
Correct: The kube controller (specifically the ReplicaSet controller) ensures that the desired number of pods specified in a ReplicaSet are running at all times. If a pod crashes or is deleted, the controller creates a new one to maintain the desired state.
D . kubelet
Incorrect: The kubelet ensures that containers are running as expected on a node but does not manage the overall availability of ReplicaSet pods across the cluster.
Why Kube Controller?
ReplicaSet Management: The ReplicaSet controller within the kube controller manager ensures that the specified number of pod replicas are always available.

Self-Healing: If a pod fails or is deleted, the controller automatically creates a new pod to maintain the desired state.
JNCIA Cloud
Reference:
The JNCIA-Cloud certification covers Kubernetes control plane components, including the kube controller. Understanding the role of the kube controller is essential for managing the availability and scalability of Kubernetes resources.
For example, Juniper Contrail integrates with Kubernetes to provide advanced networking and security features, relying on the kube controller to maintain the desired state of ReplicaSets.
Kubernetes Documentation: ReplicaSet Controller
Juniper JNCIA-Cloud Study Guide: Kubernetes Control Plane

QUESTION 25
You want to limit the memory, CPU, and network utilization of a set of processes running on a Linux host.
Which Linux feature would you configure in this scenario?
You want to limit the memory, CPU, and network utilization of a set of processes running on a Linux host.
Which Linux feature would you configure in this scenario?

A. virtual routing and forwarding instances
B. network namespaces
C. control groups
D. slicing

Correct Answer: C
Section:
Explanation:
Linux provides several features to manage system resources and isolate processes. Let's analyze each option:
A . virtual routing and forwarding instances
Incorrect: Virtual Routing and Forwarding (VRF) is a networking feature used to create multiple routing tables on a single router or host. It is unrelated to limiting memory, CPU, or network utilization for processes.
B . network namespaces
Incorrect: Network namespaces are used to isolate network resources (e.g., interfaces, routing tables) for processes. While they can help with network isolation, they do not directly limit memory or CPU usage.
C . control groups
Correct: Control Groups (cgroups) are a Linux kernel feature that allows you to limit, account for, and isolate the resource usage (CPU, memory, disk I/O, network) of a set of processes. cgroups are commonly used in containerization technologies like Docker and Kubernetes to enforce resource limits.
D . slicing
Incorrect: 'Slicing' is not a recognized Linux feature for resource management. This term may refer to dividing resources in other contexts but is not relevant here.
Why Control Groups?
Resource Management: cgroups provide fine-grained control over memory, CPU, and network utilization, ensuring that processes do not exceed their allocated resources.
Containerization Foundation: cgroups are a core technology behind container runtimes like containerd and orchestration platforms like Kubernetes.
JNCIA Cloud
Reference:
The JNCIA-Cloud certification covers Linux features like cgroups as part of its containerization curriculum. Understanding cgroups is essential for managing resource allocation in cloud environments.
For example, Juniper Contrail integrates with Kubernetes to manage containerized workloads, leveraging cgroups to enforce resource limits.
Linux Kernel Documentation: Control Groups
Juniper JNCIA-Cloud Study Guide: Linux Features

QUESTION 26
Which Docker component builds, runs, and distributes Docker containers?

A. dockerd
B. docker registry
C. docker cli

D. container

Correct Answer: A
Section:
Explanation:
Docker is a popular containerization platform that includes several components to manage the lifecycle of containers. Let's analyze each option:
A . dockerd
Correct: The Docker daemon (dockerd) is the core component responsible for building, running, and distributing Docker containers. It manages Docker objects such as images, containers, networks, and volumes, and handles requests from the Docker CLI or API.
B . docker registry
Incorrect: A Docker registry is a repository for storing and distributing Docker images. While it plays a role in distributing containers, it does not build or run them.
C . docker cli
Incorrect: The Docker CLI (Command Line Interface) is a tool used to interact with the Docker daemon (dockerd). It is not responsible for building, running, or distributing containers but rather sends commands to the daemon.
D . container
Incorrect: A container is an instance of a running application created from a Docker image. It is not a component of Docker but rather the result of the Docker daemon's operations.
Why dockerd?
Central Role: The Docker daemon (dockerd) is the backbone of the Docker platform, managing all aspects of container lifecycle management.
Integration: It interacts with the host operating system and container runtime to execute tasks like building images, starting containers, and managing resources.
JNCIA Cloud
Reference:
The JNCIA-Cloud certification covers Docker as part of its containerization curriculum. Understanding the role of the Docker daemon is essential for managing containerized applications in cloud environments.
For example, Juniper Contrail integrates with Docker to provide advanced networking and security features for containerized workloads, relying on the Docker daemon to manage containers.
Docker Documentation: Docker Daemon
Juniper JNCIA-Cloud Study Guide: Containerization

QUESTION 27
You just uploaded a qcow2 image of a vSRX virtual machine in OpenStack.
In this scenario, which service stores the virtual machine (VM) image?

A. Glance
B. Ironic
C. Neutron
D. Nova

Correct Answer: A
Section:
Explanation:
OpenStack provides various services to manage cloud infrastructure resources, including virtual machine (VM) images. Let's analyze each option:
A . Glance
Correct: Glance is the OpenStack service responsible for managing and storing VM images. It provides a repository for uploading, discovering, and retrieving images in various formats, such as qcow2, raw, or ISO.
B . Ironic
Incorrect: Ironic is the OpenStack bare-metal provisioning service. It is used to manage physical servers, not VM images.
C . Neutron
Incorrect: Neutron is the OpenStack networking service that manages virtual networks, routers, and IP addresses. It does not store VM images.
D . Nova
Incorrect: Nova is the OpenStack compute service that manages the lifecycle of virtual machines. While Nova interacts with Glance to retrieve VM images for deployment, it does not store the images itself.
Why Glance?
Image Repository: Glance acts as the central repository for VM images, enabling users to upload, share, and deploy images across the OpenStack environment.

Integration with Nova: When deploying a VM, Nova retrieves the required image from Glance to create the instance.
JNCIA Cloud
Reference:
The JNCIA-Cloud certification covers OpenStack services, including Glance, as part of its cloud infrastructure curriculum. Understanding Glance's role in image management is essential for deploying and managing virtual machines in OpenStack.
For example, Juniper Contrail integrates with OpenStack Glance to provide advanced networking features for VM images stored in the repository.
OpenStack Glance Documentation
Juniper JNCIA-Cloud Study Guide: OpenStack Services

QUESTION 28
Click the Exhibit button.

```
lab@kollaopenstack:~/kolla-openstack$ openstack server show VM-A
+-------------------------------------+-----------------------------------------------------------+
| Field                               | Value                                                     |
+-------------------------------------+-----------------------------------------------------------+
| OS-DCF:diskConfig                   | AUTO                                                      |
| OS-EXT-AZ:availability_zone         | nova                                                      |
| OS-EXT-SRV-ATTR:host                | kollaopenstack                                            |
| OS-EXT-SRV-ATTR:hypervisor_hostname | kollaopenstack                                            |
| OS-EXT-SRV-ATTR:instance_name       | instance-00000001                                         |
| OS-EXT-STS:power_state              | Shutdown                                                  |
| OS-EXT-STS:task_state               | None                                                      |
| OS-EXT-STS:vm_state                 | stopped                                                   |
| OS-SRV-USG:launched_at              | 2021-07-13T13:39:28.000000                                |
| OS-SRV-USG:terminated_at            | None                                                      |
| accessIPv4                          |                                                           |
| accessIPv6                          |                                                           |
| addresses                           | public1=10.0.2.176                                        |
| config_drive                        |                                                           |
| created                             | 2021-07-13T13:38:50Z                                      |
| flavor                              | m1.tiny (1)                                               |
...
```

You have issued the openstack server show VM-A command and received the output shown in the exhibit.
To which virtual network is the VM-A instance attached?

A. m1.tiny
B. public1
C. Nova
D. kollaopenstack

Correct Answer: B
Section:
Explanation:
The openstack server show command provides detailed information about a specific virtual machine (VM) instance in OpenStack. The output includes details such as the instance name, network attachments, power state,

and more. Let's analyze the question and options:
Key Information from the Exhibit:
The addresses field in the output shows
public1=10.0.2.176
This indicates that the VM-A instance is attached to the virtual network named public1 , with an assigned IP address of 10.0.2.176 .
Option Analysis:
A . m1.tiny
Incorrect: m1.tiny refers to the flavor of the VM, which specifies the resource allocation (e.g., CPU, memory, disk). It is unrelated to the virtual network.
B . public1
Correct: The addresses field explicitly states that the VM-A instance is attached to the public1 virtual network.
C . Nova
Incorrect: Nova is the OpenStack compute service that manages VM instances. It is not a virtual network.
D . kollaopenstack
Incorrect: kollaopenstack appears in the output as the hostname or project name but does not represent a virtual network.
Why public1?
Network Attachment: The addresses field in the output directly identifies the virtual network (public1) to which the VM-A instance is attached.
IP Address Assignment: The IP address (10.0.2.176) confirms that the VM is connected to the public1 network.
JNCIA Cloud
Reference:
The JNCIA-Cloud certification emphasizes understanding OpenStack commands and outputs, including the openstack server show command. Recognizing how virtual networks are represented in OpenStack is essential for managing VM connectivity.
For example, Juniper Contrail integrates with OpenStack Neutron to provide advanced networking features for virtual networks like public1.
OpenStack CLI Documentation: openstack server show Command
Juniper JNCIA-Cloud Study Guide: OpenStack Networking

QUESTION 29
Which OpenStack service displays server details of the compute node?

A. Keystone
B. Neutron
C. Cinder
D. Nova

Correct Answer: D
Section:
Explanation:
OpenStack provides various services to manage cloud infrastructure resources, including compute nodes and virtual machines (VMs). Let's analyze each option:
A . Keystone
Incorrect: Keystone is the OpenStack identity service responsible for authentication and authorization. It does not display server details of compute nodes.
B . Neutron
Incorrect: Neutron is the OpenStack networking service that manages virtual networks, routers, and IP addresses. It is unrelated to displaying server details of compute nodes.
C . Cinder
Incorrect: Cinder is the OpenStack block storage service that provides persistent storage volumes for VMs. It does not display server details of compute nodes.
D . Nova
Correct: Nova is the OpenStack compute service responsible for managing the lifecycle of virtual machines, including provisioning, scheduling, and monitoring. It also provides detailed information about compute nodes and VMs, such as CPU, memory, and disk usage.
Why Nova?
Compute Node Management: Nova manages compute nodes and provides APIs to retrieve server details, including resource utilization and VM status.
Integration with CLI/REST APIs: Commands like openstack server show or nova hypervisor-show can be used to display compute node and VM details.

JNCIA Cloud
Reference:
The JNCIA-Cloud certification covers OpenStack services, including Nova, as part of its cloud infrastructure curriculum. Understanding Nova's role in managing compute resources is essential for operating OpenStack environments.
For example, Juniper Contrail integrates with OpenStack Nova to provide advanced networking and security features for compute nodes and VMs.
OpenStack Nova Documentation
Juniper JNCIA-Cloud Study Guide: OpenStack Services

QUESTION 30
Which OpenStack object enables multitenancy?

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

Correct Answer: D
Section:
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 31
You want to create a template that defines the CPU, RAM, and disk space properties that a VM will use when instantiated.
In this scenario, which OpenStack object should you create?

A. role
B. Image
C. project
D. flavor

Correct Answer: D
Section:
Explanation:
In OpenStack, a flavor defines the compute, memory, and storage properties of a virtual machine (VM) instance. Let's analyze each option:
A . role
Incorrect: A role defines permissions and access levels for users within a project. It is unrelated to defining VM properties.
B . Image
Incorrect: An image is a template used to create VM instances. While images define the operating system and initial configuration, they do not specify CPU, RAM, or disk space properties.
C . project
Incorrect: A project (or tenant) represents an isolated environment for managing resources. It does not define the properties of individual VMs.
D . flavor
Correct: A flavor specifies the CPU, RAM, and disk space properties that a VM will use when instantiated. For example, a flavor might define a VM with 2 vCPUs, 4 GB of RAM, and 20 GB of disk space.
Why Flavor?
Resource Specification: Flavors allow administrators to define standardized resource templates for VMs, ensuring consistency and simplifying resource allocation.
Flexibility: Users can select the appropriate flavor based on their workload requirements, making it easy to deploy VMs with predefined configurations.
JNCIA Cloud
Reference:
The JNCIA-Cloud certification covers OpenStack concepts, including flavors, as part of its cloud infrastructure curriculum. Understanding how flavors define VM properties is essential for managing compute resources effectively.
For example, Juniper Contrail integrates with OpenStack Nova to provide advanced networking features for VMs deployed using specific flavors.
OpenStack Nova Documentation: Flavors
Juniper JNCIA-Cloud Study Guide: OpenStack Compute

QUESTION 32
Which OpenStack node runs the network agents?

A. block storage
B. controller
C. object storage
D. compute

Correct Answer: B
Section:
Explanation:
In OpenStack, network agents are responsible for managing networking tasks such as DHCP, routing, and firewall rules. These agents run on specific nodes within the OpenStack environment. Let's analyze each option:
A . block storage
Incorrect: Block storage nodes host the Cinder service, which provides persistent storage volumes for virtual machines. They do not run network agents.
B . controller
Incorrect: Controller nodes host core services like Keystone (identity), Horizon (dashboard), and Glance (image service). While some networking services (e.g., Neutron server) may reside on the controller node, the actual network agents typically do not run here.
C . object storage
Incorrect: Object storage nodes host the Swift service, which provides scalable object storage. They are unrelated to running network agents.
D . compute
Correct: Compute nodes run the Nova compute service, which manages virtual machine instances. Additionally, compute nodes host network agents (e.g., L3 agent, DHCP agent, and metadata agent) to handle networking tasks for VMs running on the node.
Why Compute Nodes?
Proximity to VMs: Network agents run on compute nodes to ensure efficient communication with VMs hosted on the same node.
Decentralized Architecture: By distributing network agents across compute nodes, OpenStack achieves scalability and fault tolerance.
JNCIA Cloud

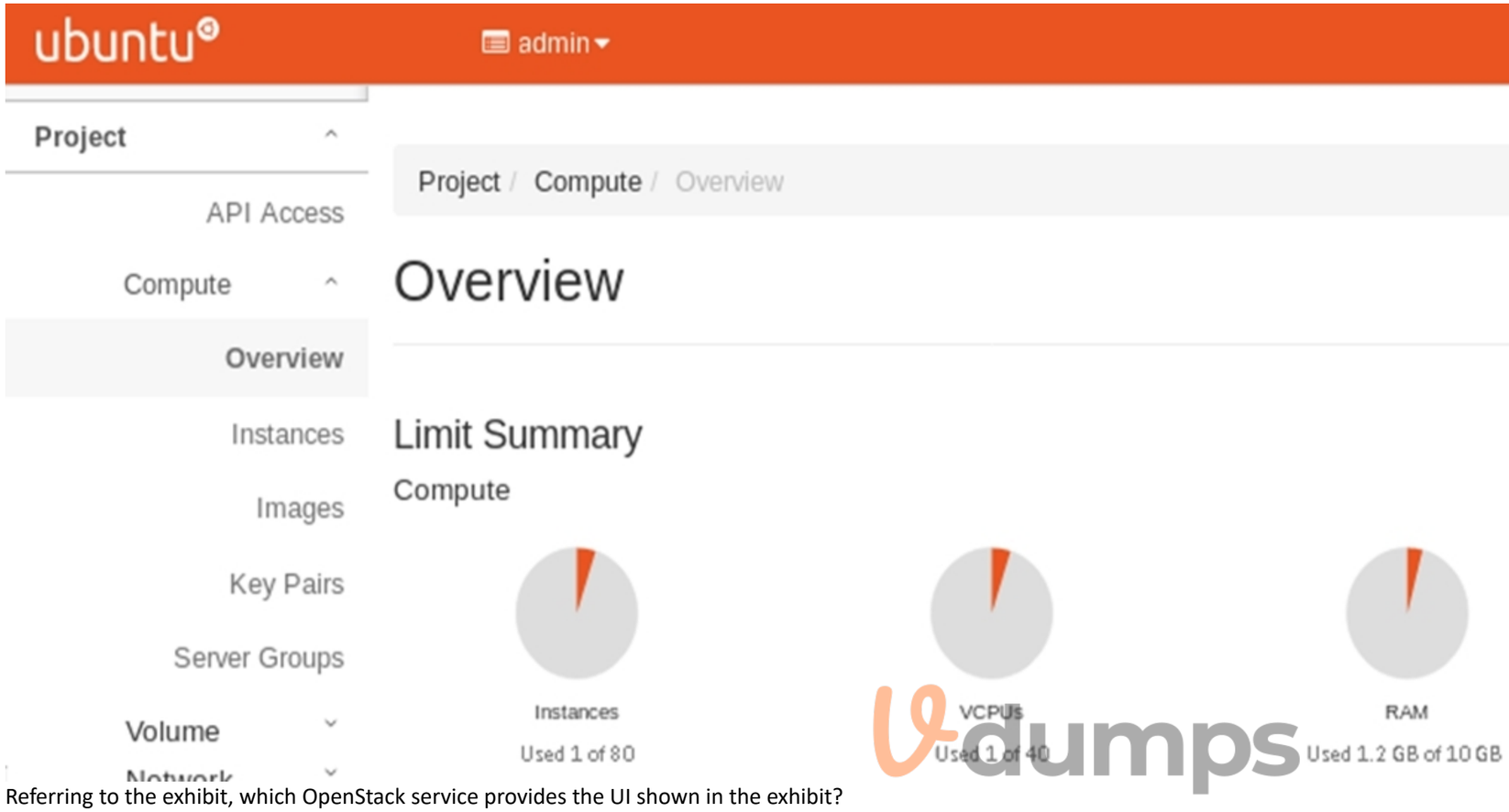
Reference:
The JNCIA-Cloud certification covers OpenStack architecture, including the roles of compute nodes and network agents. Understanding where network agents run is essential for managing OpenStack networking effectively.
For example, Juniper Contrail integrates with OpenStack Neutron to provide advanced networking features, leveraging network agents on compute nodes for traffic management.
OpenStack Neutron Documentation: Network Agents
Juniper JNCIA-Cloud Study Guide: OpenStack Networking

QUESTION 33
Your e-commerce application is deployed on a public cloud. As compared to the rest of the year, it receives substantial traffic during the Christmas season.
In this scenario, which cloud computing feature automatically increases or decreases the resource based on the demand?

A. resource pooling
B. on-demand self-service
C. rapid elasticity
D. broad network access

Correct Answer: C
Section:
Explanation:
Cloud computing provides several key characteristics that enable flexible and scalable resource management. Let's analyze each option:
A . resource pooling
Incorrect: Resource pooling refers to the sharing of computing resources (e.g., storage, processing power) among multiple users or tenants. While important, it does not directly address the automatic scaling of resources based on demand.
B . on-demand self-service
Incorrect: On-demand self-service allows users to provision resources (e.g., virtual machines, storage) without requiring human intervention. While this is a fundamental feature of cloud computing, it does not describe the ability to automatically scale resources.
C . rapid elasticity
Correct: Rapid elasticity is the ability of a cloud environment to dynamically increase or decrease resources based on demand. This ensures that applications can scale up during peak traffic periods (e.g., Christmas season) and scale down during low-demand periods, optimizing cost and performance.
D . broad network access
Incorrect: Broad network access refers to the ability to access cloud services over the internet from various devices. While essential for accessibility, it does not describe the scaling of resources.
Why Rapid Elasticity?
Dynamic Scaling: Rapid elasticity ensures that resources are provisioned or de-provisioned automatically to meet changing workload demands.
Cost Efficiency: By scaling resources only when needed, organizations can optimize costs while maintaining performance.
JNCIA Cloud
Reference:
The JNCIA-Cloud certification emphasizes the key characteristics of cloud computing, including rapid elasticity. Understanding this concept is essential for designing scalable and cost-effective cloud architectures.
For example, Juniper Contrail supports cloud elasticity by enabling dynamic provisioning of network resources in response to changing demands.
NIST Cloud Computing Reference Architecture
Juniper JNCIA-Cloud Study Guide: Cloud Characteristics

QUESTION 34
Click to the Exhibit button.



Referring to the exhibit, which OpenStack service provides the UI shown in the exhibit?

A. Nova
B. Neutron
C. Horizon
D. Heat

Correct Answer: C
Section:
Explanation:
The UI shown in the exhibit is the OpenStack Horizon dashboard. Horizon is the web-based user interface (UI) for OpenStack, providing administrators and users with a graphical interface to interact with the cloud environment. Through Horizon, users can manage resources like instances, networks, and storage, which is evident in the displayed metrics (Instances, VCPUs, RAM) for the project.

QUESTION 35
Which OpenStack service provides API client authentication?

A. Keystone
B. Nova
C. Heal
D. Neutron

Correct Answer: A

Section:
Explanation:
OpenStack is an open-source cloud computing platform that provides various services for managing infrastructure resources. Let's analyze each option:
A . Keystone
Correct: Keystone is the OpenStack service responsible for identity management and API client authentication . It provides authentication, authorization, and service discovery for other OpenStack services.
B . Nova
Incorrect: Nova is the OpenStack compute service that manages virtual machines and bare-metal servers. It does not handle authentication or API client validation.
C . Heat
Incorrect: Heat is the OpenStack orchestration service that automates the deployment and management of infrastructure resources using templates. It does not provide authentication services.
D . Neutron
Incorrect: Neutron is the OpenStack networking service that manages virtual networks, routers, and IP addresses. It is unrelated to API client authentication.
Why Keystone?
Authentication and Authorization: Keystone ensures that only authorized users and services can access OpenStack resources by validating credentials and issuing tokens.
Service Discovery: Keystone also provides a catalog of available OpenStack services and their endpoints, enabling seamless integration between components.
JNCIA Cloud
Reference:
The JNCIA-Cloud certification covers OpenStack services, including Keystone, as part of its cloud infrastructure curriculum. Understanding Keystone's role in authentication is essential for managing secure OpenStack deployments.
For example, Juniper Contrail integrates with OpenStack Keystone to authenticate and authorize network resources, ensuring secure and efficient operation.
OpenStack Keystone Documentation
Juniper JNCIA-Cloud Study Guide: OpenStack Services

QUESTION 36
You are asked to run a container in a Kubernetes environment.
What should you do to accomplish this task?

A. Create a JINJA2 template for the container and its resources.
B. Create a WYSYG definition for the container and its resources.
C. Define a YAML manifest for the container and its resources.
D. Define an XML configuration for the container and its resources.

Correct Answer: C
Section:
Explanation:
Kubernetes uses declarative configuration files to define and manage resources like containers, pods, and services. Let's analyze each option:
A . Create a JINJA2 template for the container and its resources.
Incorrect: JINJA2 is a templating language often used in automation tools like Ansible. While it can generate Kubernetes manifests, Kubernetes itself does not use JINJA2 templates natively.
B . Create a WYSYG definition for the container and its resources.
Incorrect: 'WYSYG' (What You See Is What You Get) is not a recognized format for Kubernetes configurations. Kubernetes relies on structured formats like YAML or JSON.
C . Define a YAML manifest for the container and its resources.
Correct: Kubernetes uses YAML (or JSON) manifests to define the desired state of resources, including containers, pods, and services. A YAML manifest specifies details like container images, resource limits, environment variables, and networking.
D . Define an XML configuration for the container and its resources.
Incorrect: Kubernetes does not use XML for defining resources. YAML is the standard format due to its readability and simplicity.
Why YAML Manifests?
Declarative Configuration: YAML manifests allow you to describe the desired state of your resources in a human-readable format.
Standard Practice: Kubernetes natively supports YAML for defining and deploying resources, making it the correct choice for this task.
JNCIA Cloud
Reference:

The JNCIA-Cloud certification emphasizes Kubernetes resource management, including YAML manifests. Understanding how to define and apply manifests is essential for deploying and managing containerized applications.
For example, Juniper Contrail integrates with Kubernetes to provide advanced networking features, relying on YAML manifests to configure resources.
Kubernetes Documentation: YAML Manifests
Juniper JNCIA-Cloud Study Guide: Kubernetes Resource Management

QUESTION 37
Your organization has legacy virtual machine workloads that need to be managed within a Kubernetes deployment.
Which Kubernetes add-on would be used to satisfy this requirement?

A. ADOT
B. Canal
C. KubeVirt
D. Romana

Correct Answer: C
Section:
Explanation:
Kubernetes is designed primarily for managing containerized workloads, but it can also support legacy virtual machine (VM) workloads through specific add-ons. Let's analyze each option:
A . ADOT
Incorrect: The AWS Distro for OpenTelemetry (ADOT) is a tool for collecting and exporting telemetry data (metrics, logs, traces). It is unrelated to running VMs in Kubernetes.
B . Canal
Incorrect: Canal is a networking solution that combines Flannel and Calico to provide overlay networking and network policy enforcement in Kubernetes. It does not support VM workloads.
C . KubeVirt
Correct: KubeVirt is a Kubernetes add-on that enables the management of virtual machines alongside containers in a Kubernetes cluster. It allows organizations to run legacy VM workloads while leveraging Kubernetes for orchestration.
D . Romana
Incorrect: Romana is a network policy engine for Kubernetes that provides security and segmentation. It does not support VM workloads.
Why KubeVirt?
VM Support in Kubernetes: KubeVirt extends Kubernetes to manage both containers and VMs, enabling organizations to transition legacy workloads to a Kubernetes environment.
Unified Orchestration: By integrating VMs into Kubernetes, KubeVirt simplifies the management of hybrid workloads.
JNCIA Cloud
Reference:
The JNCIA-Cloud certification covers Kubernetes extensions like KubeVirt as part of its curriculum on cloud-native architectures. Understanding how to integrate legacy workloads into Kubernetes is essential for modernizing IT infrastructure.
For example, Juniper Contrail integrates with Kubernetes and KubeVirt to provide networking and security for hybrid workloads.
KubeVirt Documentation
Juniper JNCIA-Cloud Study Guide: Kubernetes Extensions

QUESTION 38
Which technology is used to run VMs in an OpenShift cluster?

A. ESXi
B. OpenStack
C. Hyper-V
D. KubeVirt

Correct Answer: D
Section:
Explanation:

OpenShift is a Kubernetes-based container platform that supports both containerized and virtualized workloads. Let's analyze each option:
A . ESXi
Incorrect: ESXi is VMware's hypervisor for running virtual machines. While it is widely used in traditional virtualization environments, it is not integrated into OpenShift.
B . OpenStack
Incorrect: OpenStack is an open-source cloud computing platform used for managing infrastructure resources (e.g., compute, storage, networking). It is unrelated to running VMs in an OpenShift cluster.
C . Hyper-V
Incorrect: Hyper-V is Microsoft's hypervisor for running virtual machines. Like ESXi, it is not integrated into OpenShift.
D . KubeVirt
Correct: KubeVirt is the technology used to run virtual machines in an OpenShift cluster. It extends Kubernetes to support VM workloads alongside containers, enabling hybrid workload management.
Why KubeVirt?
Integration with OpenShift: KubeVirt is specifically designed to run VMs in Kubernetes-based environments like OpenShift.
Hybrid Workload Support: It allows organizations to manage both containers and VMs using the same Kubernetes APIs and tools.
JNCIA Cloud
Reference:
The JNCIA-Cloud certification covers OpenShift and its integration with Kubernetes extensions like KubeVirt. Understanding how to run VMs in OpenShift is essential for managing hybrid workloads in cloud-native environments.
For example, Juniper Contrail integrates with OpenShift and KubeVirt to provide networking and security for hybrid workloads.
KubeVirt Documentation
OpenShift Documentation: Virtualization
Juniper JNCIA-Cloud Study Guide: OpenShift and Kubernetes