DUMPSBOSS.

VMware vSphere with Tanzu Specialist

VMware 5V0-23.20

Version Demo

Total Demo Questions: 14

Total Premium Questions: 144

Buy Premium PDF

https://dumpsboss.co

support@dumpsboss.co

support@dumpsboss.co
dumpsboss.co

Topic Break Down

Topic	No. of Questions
Topic 1, Architectures and Technologies	38
Topic 2, VMware Products and Solutions	27
Topic 3, Planning and Designing	8
Topic 4, Installing, Configuring, and Setup	29
Topic 5, Performance-tuning, Optimization and Upgrades	6
Topic 6, Troubleshooting and Repairing	1
Topic 7, Administrative and Operational Tasks	35
Total	144

QUESTION NO: 1

Which command displays the storage limits that have been set together with the amount of resources consumed?

A. kubect1 get resourcequotas

B. kubect1 config get-resourcequotas limits

C. kubect1 list resourcequotas

D. kubect1 describe resourcequotas

ANSWER: D

Explanation:

`kubectl describe resourcequota` is the appropriate command because it returns the detailed status of a ResourceQuota, including both the configured quota ceilings and current consumption. In its output, quota entries are presented with `Used` and `Hard` values. For storage-related quotas, this can include values such as total persistent-volume-claim storage requests, persistent-volume-claim counts, or storage-class-specific request limits. Comparing `Used` with `Hard` directly shows how much of each enforced storage resource limit has been consumed and how much capacity remains available to workloads in the namespace. A ResourceQuota is namespace-scoped, so the command should be run against the intended namespace, for example with `-n <namespace>`, and a specific quota name can be supplied when required. Kubernetes documents ResourceQuota status as containing the enforced hard limits and the currently used aggregate resources; the describe output exposes those values in an administrator-friendly format. See the Kubernetes Resource Quotas documentation and the kubectl describe reference.

QUESTION NO: 2

An administrator is tasked with increasing the amount of CPU and memory in an existing Tanzu Kubernetes cluster.

Which change must the administrator complete to ensure the cluster scales successfully when updating the YAML definition?

A. Manually update the CPU and memory of the nodes

B. Update the Virtual Machine Class Type

C. Increase the number of worker nodes

D. Increase the number of control plane nodes

ANSWER: B

Explanation:

Updating the Virtual Machine Class Type is required because a Tanzu Kubernetes cluster's control-plane and worker nodes are deployed as virtual machines whose CPU, memory, and related resource configuration are defined by the selected virtual machine class. In the cluster YAML, the VM class is specified for the control plane and worker node pools. To increase the compute capacity of the individual cluster nodes, the administrator changes those class references to a class that provides the required larger CPU and memory allocation, then applies the updated manifest. Tanzu Kubernetes Grid reconciles the desired state and replaces or updates cluster nodes as needed to use the requested sizing. VM classes also determine whether CPU and memory are reserved, depending on whether a guaranteed or best-effort class is used; therefore, the chosen class must be available in the target namespace and compatible with the desired workload requirements. This approach keeps the cluster under Kubernetes and Tanzu lifecycle management rather than making unsupported, out-of-band modifications to node virtual machines. VMware's vSphere Supervisor documentation describes VM classes as the mechanism for defining VM resource sizing and their availability to Tanzu workloads. See the VMware vSphere Supervisor documentation and the Kubernetes workload management documentation.

QUESTION NO: 3

An organization is preparing to deploy vSphere with Tanzu and will be using the vSphere Networking stack.

How should the administrator allocate management network IP addresses for the Kubernetes Control Plane within the Supervisor Cluster?

A. Five IP addresses are required, one for each of the Control Plane VMs. one for the floating IP address of the Control Plane VM, and one spare for performing rolling cluster upgrades

B. Four IP addresses are required, one for each of the Control Plane VMs and one spare for performing rolling cluster upgrades

C. Three IP addresses are required, one for each of the Control Plane VMs

D. Six IP addresses are required, one for each of the Control Plane VMs, one for the floating IP address of the Control Plane VM. one for performing rolling cluster upgrades and one for the image Registry VM.

ANSWER: A

Explanation:

Five IP addresses are required, one for each of the Control Plane VMs. one for the floating IP address of the Control Plane VM, and one spare for performing rolling cluster upgrades. A Supervisor Cluster deployed with the vSphere networking stack uses three Kubernetes control plane virtual machines for its highly available control plane. Each control plane VM needs a management-network address. The cluster additionally requires a floating management IP address, which provides a stable endpoint for control-plane availability and can move between control-plane VMs as needed. A fifth, unused address must be reserved so that vCenter Server can create a replacement control plane VM during a rolling upgrade without an IP-address conflict. Therefore, the administrator must provide a block of five consecutive static management IP addresses for the Supervisor control plane. This allocation is specifically for the Kubernetes control plane and is separate from addresses planned for workload networks, load-balancer services, ingress, or other infrastructure components. VMware documents the control-plane address requirements as part of Supervisor networking configuration and planning. See the VMware Supervisor networking configuration documentation and the Supervisor planning and preparation documentation.

QUESTION NO: 4

Which three characteristics are true of Control Plane VMs? (Choose three.)

A. They can be resized by administrators directly through vCenter Inventory View.

B. They each run the Spherelet.

C. They each expose the Kubernetes API.

D. They do not run any Kubernetes Pods.

E. They are connected to a Management portgroup.

F. They are deployed via a vCenter Service.

ANSWER: C E F

Explanation:

Control Plane VMs are the virtual machines that host the Supervisor's Kubernetes control-plane components. Each Control Plane VM runs a Kubernetes API server, providing a control-plane endpoint; the Supervisor's configured control-plane endpoint and load-balancing design make that API available to clients with high availability. The Control Plane VMs are created and lifecycle-managed by the vCenter Server Workload Control Plane service when a Supervisor is enabled, rather than being manually deployed as ordinary workload VMs. They are also attached to the selected management network or management port group so that vCenter Server, ESXi hosts, and other required management-plane services can communicate with the Supervisor control plane. These networking settings are part of Supervisor configuration and must provide the required management connectivity and IP addressing for the control-plane nodes. VMware describes the Supervisor architecture, including its control-plane VMs and API services, in the vSphere Supervisor documentation. Configuration requirements for the management network and Supervisor deployment are also documented in the Supervisor configuration and management guide.

QUESTION NO: 5

What is the proper way to delete a Persistent Volume Claim?

A. By using the kubectl delete persistentvolumeclaim command

B. By using the kubectl remove pvc command

C. Through the SPBM policy engine using the vSphere Client

D. By unmounting the volume from the VM and deleting it from the vSphere datastore

ANSWER: A

Explanation:

By using the `kubectl delete persistentvolumeclaim` command is the proper Kubernetes-native method for deleting a PersistentVolumeClaim (PVC). A PVC is a namespaced Kubernetes API resource, so it should be managed through the Kubernetes API rather than by directly manipulating its backing storage in vSphere. For example, an administrator can run `kubectl delete persistentvolumeclaim <claim-name> -n <namespace>`, or use the short resource name, `kubectl delete pvc <claim-name> -n <namespace>`. Kubernetes then processes the deletion request and coordinates the associated storage lifecycle through the configured CSI driver.

This workflow preserves Kubernetes resource state and allows protection mechanisms, such as finalizers, to complete required cleanup safely before the claim is removed. The eventual handling of the bound PersistentVolume and its backing vSphere storage is controlled by the PersistentVolume reclaim policy and the storage provisioner. In environments using the vSphere CSI driver, Kubernetes communicates with vSphere to perform any required storage operations rather than requiring an administrator to delete the backing disk manually. See the Kubernetes documentation for PersistentVolumeClaims and the command reference for kubectl delete.

QUESTION NO: 6

How does Kubernetes implement the vSphere storage policy in vSphere with Tanzu?

A. Storage class

B. Paravirtual CSI

C. Static Persistent Volume

D. Persistent Volume

ANSWER: A

Explanation:

Storage class is correct because vSphere with Tanzu exposes vSphere storage policies assigned to a workload namespace as Kubernetes storage classes. This gives Kubernetes users a native way to request persistent storage with the placement, performance, availability, and data-service requirements defined by the underlying vSphere policy. A developer selects the storage class in a persistent volume claim; Kubernetes then uses the vSphere integration to dynamically provision an appropriate persistent volume and its backing virtual disk in accordance with that policy.

The mapping also makes policy-controlled storage consumable without requiring application teams to interact directly with vCenter Server. When creating the Kubernetes storage-class name, vSphere with Tanzu normalizes the source policy name by converting uppercase characters to lowercase and replacing spaces with hyphens. Consequently, using lowercase policy names without spaces helps make the resulting storage-class names predictable. This behavior connects vSphere Storage Policy Based Management with the standard Kubernetes storage consumption workflow. See the VMware vSphere with Tanzu documentation at Broadcom TechDocs and the Kubernetes overview of Storage Classes.

QUESTION NO: 7

Which three elements are part of the Container Runtime for ESXi (CRX)? (Choose three.)

A. Image Service

B. Kube Proxy

C. Linux kernel

D. Customized VMX

E. Spherelet

F. Container Engine

ANSWER: A C E

Explanation:

Container Runtime for ESXi (CRX) is the purpose-built, lightweight execution environment that allows a vSphere Pod to run directly on an ESXi host while retaining VM-style isolation. Its minimal **Linux kernel** supplies the guest operating-system functionality needed to execute container workloads without deploying a full general-purpose virtual machine. The **Image Service** is responsible for obtaining and preparing the container images required for the workload, allowing the pod runtime to access the images specified by Kubernetes. **Spherelet** is VMware's ESXi-integrated Kubernetes node agent; it communicates with the Kubernetes control plane and coordinates the lifecycle and execution of vSphere Pods on the host. Together, these components provide the core CRX architecture: a compact kernel environment, image handling, and Kubernetes-aware pod management integrated with ESXi. This design is foundational to vSphere with Tanzu's ability to run Kubernetes workloads as isolated vSphere Pods. VMware's vSphere with Tanzu documentation describes vSphere Pods and their ESXi-integrated runtime model in the vSphere with Tanzu Configuration and Management documentation; additional architecture context is available in VMware's vSphere with Tanzu overview.

QUESTION NO: 8

A Pod remains in the Pending state after a developer deploys an application to a Tanzu Kubernetes cluster.

Which two actions should the administrator take to identify the scheduling issue? (Choose two.)

A. Run kubectl describe pod for the Pending Pod.

B. Review the ResourceQuota objects in the namespace.

C. Delete the Supervisor control plane virtual machines.

D. Manually move the worker node virtual machines to another datastore.

E. Remove the StorageClass from the vSphere Namespace.

ANSWER: A B

Explanation:

Running `kubectl describe pod` provides detailed Pod status and an Events section. Scheduler events commonly identify the reason that a Pod cannot be placed, such as insufficient CPU or memory, unsatisfied node affinity rules, unbound persistent volumes, or taints that the Pod does not tolerate.

Reviewing the namespace ResourceQuota objects is also appropriate. A namespace quota can prevent a workload from being admitted or scheduled when its requested CPU, memory, object count, or storage consumption would exceed the configured limit. The command `kubectl get resourcequota` identifies quota objects, and `kubectl describe resourcequota` displays their configured hard limits and current usage. See the Kubernetes documentation for debugging Pods and Resource Quotas.

QUESTION NO: 9

An administrator needs to limit resource consumption within a vSphere with Tanzu namespace.

Which three actions should be taken to meet this goal? (Choose three.)

A. Limit the capacity of the storage able to be consumed.

B. Change the size of the Supervisor Control Plane nodes.

C. Limit the number of load balancers able to be deployed.

D. Limit the CPU and Memory able to be consumed.

E. Limit the number of Operators able to be deployed.

F. Limit the number of vSphere Pods able to be provisioned.

ANSWER: A D F

Explanation:

vSphere with Tanzu uses namespace resource limits to establish a bounded allocation of Supervisor resources for the workloads deployed by namespace users. “Limit the CPU and Memory able to be consumed” is required because CPU and memory limits define the compute capacity available to the namespace. This prevents its workloads from consuming more than the amount reserved for that namespace from the Supervisor’s backing vSphere resources.

“Limit the capacity of the storage able to be consumed” is also required. A namespace storage limit controls the aggregate storage capacity that workloads in the namespace can request and consume, supporting predictable datastore usage and tenant isolation.

“Limit the number of vSphere Pods able to be provisioned” is the applicable object-count limit. vSphere with Tanzu namespaces can have a maximum number of vSphere Pods, which restricts the scale of pod-based workloads that namespace members can create. Together, these limits govern compute, storage, and vSphere Pod consumption at the namespace boundary. VMware documents namespace resource limits as part of Supervisor namespace configuration; see the vSphere with Tanzu configuration documentation and the Kubernetes Resource Quotas documentation for the underlying quota model.

QUESTION NO: 10

Which two container network interfaces (CNIs) are supported with Tanzu Kubernetes clusters created by the Tanzu Kubernetes Grid Service? (Choose two)

A. NSX-T

B. Weave Net

C. Flannel

D. Antrea

E. Calico

ANSWER: D E

Explanation:

Tanzu Kubernetes clusters created through the Tanzu Kubernetes Grid Service support **Antrea** and **Calico** as their container network interface implementations. A CNI supplies the pod-network connectivity required for Kubernetes workloads and integrates network-policy capabilities used to control traffic between pods and services. Antrea is the default CNI for Tanzu Kubernetes Grid Service clusters in the relevant vSphere with Tanzu releases. It is Kubernetes-native networking software built on Open vSwitch and provides both connectivity and Kubernetes NetworkPolicy enforcement. Calico is also supported as an alternative CNI, allowing administrators to select it when its networking and policy model better fits organizational requirements. The CNI selection is specified as part of the Tanzu Kubernetes cluster configuration and should be planned before deployment because it defines fundamental cluster networking behavior. VMware’s vSphere

with Tanzu documentation explicitly identifies Antrea and Calico as the supported CNI choices for Tanzu Kubernetes Grid Service-provisioned clusters. See the VMware vSphere with Tanzu CNI documentation and the Antrea project documentation.

QUESTION NO: 11

Which object helps maintain copies of a vSphere pod?

A. ReplicaSets

B. Network Policies

C. Namespaces

D. Persistent Volume

ANSWER: A

Explanation:

ReplicaSets are Kubernetes workload-controller objects that maintain a specified number of identical Pod replicas. A ReplicaSet uses a label selector to identify the Pods it manages and continuously compares the actual number of matching Pods with the desired replica count. If a managed Pod is deleted, fails, or is otherwise no longer present, the ReplicaSet creates another Pod from its Pod template so that the requested number of copies is restored. This provides redundancy and availability for containerized workloads running in a vSphere with Tanzu environment, just as it does in any conformant Kubernetes cluster. In practical application deployments, ReplicaSets are commonly created and managed indirectly through a Deployment, which adds declarative rollout and update capabilities while relying on ReplicaSets to maintain the desired Pod count. The Kubernetes documentation defines a ReplicaSet as the controller responsible for ensuring that a stable set of replica Pods runs at any given time. See the Kubernetes ReplicaSet documentation and VMware's vSphere with Tanzu workload deployment documentation for related Kubernetes workload-management guidance.

QUESTION NO: 12

Which two actions should a developer take before deploying a workload to a Tanzu Kubernetes cluster from a local YAML file? (Choose two.)

A. Authenticate to the Tanzu Kubernetes cluster.

B. Select the context for the target Tanzu Kubernetes cluster.

C. Create a new vSphere Distributed Switch.

D. Deploy an additional Supervisor control plane VM.

E. Assign a storage policy directly to each ESXi host.

ANSWER: A B

Explanation:

Authenticate to the Tanzu Kubernetes cluster is required so that kubectl has valid credentials and a kubeconfig context for the target cluster. In a vSphere with Tanzu environment, a developer can use the `kubectl vsphere login` workflow to obtain access based on assigned vCenter Single Sign-On and namespace permissions.

Select the context for the target Tanzu Kubernetes cluster is also required. A kubeconfig can contain entries for the Supervisor and multiple Tanzu Kubernetes clusters. The developer should select the intended cluster context before applying the YAML file, ensuring that the workload is created in the correct cluster and namespace.

Creating a new vSphere Distributed Switch or modifying the Supervisor control plane is not a developer deployment task. Kubernetes documents kubeconfig contexts in Configure Access to Multiple Clusters.

QUESTION NO: 13

Which corresponding object is automatically created in the embedded Registry Service when a new Namespace is created in the Supervisor Cluster?

A. Pod

B. Project

C. Container

D. Image

ANSWER: B

Explanation:

Project is correct because the embedded Registry Service integrates the Supervisor Cluster's namespace model with its image-registry authorization model. When a vSphere Namespace is created and associated with the Registry Service, vSphere automatically provisions a corresponding registry project for that namespace. The project provides a scoped location in which container images can be stored and managed for workloads deployed in that namespace. This automatic association enables namespace users to authenticate to the registry and work with images under permissions governed by the namespace's access controls, rather than requiring an administrator to manually create and align registry locations for every namespace. The resulting project also gives workload teams a predictable image path based on their namespace when configuring Kubernetes manifests, image pulls, and image publishing workflows. This behavior is part of the Registry Service integration described in the vSphere with Tanzu administration documentation, which covers configuring namespaces and registry access. See the vSphere with Tanzu Configuration and Management documentation and the vSphere with Tanzu documentation portal for Registry Service and Supervisor Namespace administration guidance.

QUESTION NO: 14

Which container network interface (CNI) is supported with Tanzu Kubernetes clusters created by the Tanzu Kubernetes Grid Service? (Choose two)

A. NSX-T

B. WeaveNet

C. Antrea

D. Cillium

ANSWER: A C

Explanation:

NSX-T and Antrea are supported networking choices for Tanzu Kubernetes clusters deployed through the Tanzu Kubernetes Grid Service in the vSphere with Tanzu context covered by this question. NSX-T provides the infrastructure-integrated networking and security capabilities used for Tanzu workloads, including logical networking, routing, load-balancing integration, and Kubernetes-aware policy enforcement through its container networking components. This enables workload clusters to participate in the NSX-T networking design configured for the Supervisor environment.

Antrea is VMware-originated open-source Kubernetes networking and security software built on Open vSwitch. It implements Kubernetes networking requirements and provides Kubernetes NetworkPolicy enforcement, while also supporting Antrea-specific policy and observability capabilities. Tanzu Kubernetes Grid uses Antrea as its standard CNI implementation in applicable Tanzu Kubernetes cluster deployments, making it an appropriate supported answer alongside NSX-T for this exam objective. The exact available networking configuration can vary by vSphere with Tanzu release and deployment architecture, so administrators should always validate the supported topology for their installed release. See the Tanzu Kubernetes Grid networking documentation and the Antrea documentation.