

TESTPASSPORT

練習題



質 量 更 高

服 務 更 好

<https://www.testpassport.net>

免費一年更新服務

Exam : **3V0-24.25**

Title : Advanced VMware Cloud
Foundation 9.0 vSphere
Kubernetes Service

Version : DEMO

1.An administrator is tasked with making an existing vSphere Supervisor highly available by adding two additional vSphere Zones.

How should the administrator perform this task?

- A. You cannot add an existing Supervisor to a new vSphere Zone.
- B. Create a new multi-zone deployment and assign an existing vSphere cluster to it.
- C. Create a new vSphere Zone and add the Supervisor to the new vSphere Zone.
- D. Select Configure, select vSphere Zones, and click Add New vSphere Zone.

Answer: A

Explanation:

In VMware Cloud Foundation 9.0 and vSphere Supervisor architectures, the decision to deploy a Single-Zone or a Multi-Zone Supervisor is made at the time of initial enablement. A Single-Zone Supervisor is tied to a specific vSphere Cluster. A Multi-Zone Supervisor requires a minimum of three vSphere Zones (each mapped to a cluster) to be defined before the Supervisor is deployed so that the Control Plane VMs can be distributed for high availability.

Currently, there is no supported "in-place" migration path to convert a deployed Single-Zone Supervisor into a Multi-Zone Supervisor by simply adding zones later. If an organization requires the high availability provided by a three-zone architecture, the administrator must decommission the existing Single-Zone Supervisor and then re-enable the Supervisor Service using the Multi-Zone configuration wizard. This design ensures that the underlying Kubernetes Control Plane components are correctly instantiated with the necessary quorum and anti-affinity rules that can only be established during the initial "Workload Management" setup phase.

2.What three components run in a VMware vSphere Kubernetes Service (VKS) cluster? (Choose three.)

- A. Cloud Provider Implementation
- B. Container Network Implementation
- C. Cloud Provider Interface
- D. Container Storage Interface
- E. Cloud Storage Implementation
- F. Container Network Interface

Answer: A D F

Explanation:

VCF 9.0 explicitly lists the components that run in a VKS cluster and groups them into areas such as authentication/authorization, storage integration, pod networking, and load balancing. In that list, the documentation names: "Container Storage Interface Plugin" (a paravirtual CSI plug-in that integrates with CNS through the Supervisor), "Container Network Interface Plug-in" (a CNI plugin that provides pod networking), and "Cloud Provider Implementation" (supports creating Kubernetes load balancer services).

These three items map directly to the answer choices D (Container Storage Interface),F (Container Network Interface), and A (Cloud Provider Implementation). The same VCF 9.0 section also mentions an authentication webhook, but that component is not offered as a selectable option in this question, so the best three matches among the provided choices are the CSI, CNI, and cloud provider implementation entries that the document explicitly states are present inside a VKS cluster.

3.An administrator had deployed a Supervisor cluster on vSphere in a mult-zone-enabled environment

and now wants to create a zonal vSphere Namespace so that workloads can be scheduled across zones.

Drag and drop the six actions into the correct order from Configuration Option list on the left and place them into the Configuration Sequence on the right. (Choose six.)

Configuration Option

Select workload networking.
Assign the zonal storage policy.
Grant RBAC / permissions.
Create the vSphere Namespace
Assign the zones.
Define resource quotas / limits.

Configuration Sequence

Answer:

Configuration Option

Select workload networking.
Assign the zonal storage policy.
Grant RBAC / permissions.
Create the vSphere Namespace
Assign the zones.
Define resource quotas / limits.

Configuration Sequence

Create the vSphere Namespace
Assign the zones.
Select workload networking.
Assign the zonal storage policy.
Define resource quotas / limits.
Grant RBAC / permissions.

Explanation:

Configuration Sequence (in order):

Create the vSphere Namespace

Assign the zones

Select workload networking

Assign the zonal storage policy

Define resource quotas / limits

Grant RBAC / permissions

A zonal vSphere Namespace is created as a standard namespace first, then “zonalized” by associating it with one or more vSphere Zones so workloads can be scheduled according to zone placement rules.

You start by creating the namespace because it is the tenancy and governance container where networking, storage access, quotas, and permissions are applied. Next, you assign the zones, since zone association is what makes the namespace “zonal” and determines where Kubernetes workloads (and their node pools) are allowed to land.

With zones set, you configure workload networking, because namespaces must have the correct network attachment and IP behavior for the workloads that will be placed across the selected zones. Then you assign the zonal storage policy, ensuring that persistent volumes can be provisioned using storage that is valid /available for the zone placement model you selected. After networking and storage access are defined, you set resource quotas/limits (CPU, memory, storage) so multi-tenant consumption stays within governance boundaries. Finally, you grant RBAC/permissions so the right DevOps/users can consume the namespace and provision clusters/workloads under the enforced controls.

4.A VMware Administrator is tasked with implementing a backup and restore strategy using Velero and external object storage for the namespace 'myapp1'. Arrange the steps in the correct order of operations to enable Velero.

Run test backup: Velero backup create test-backup --include namespaces 'myapp'.	Apply BackupStorageLocation YAML.	Apply VolumeSnapshotLocation YAML.	Run the install command: Velero install -- provider aws --bucket <bucket> -- plugins velero/velero-plugin- for-vsphere:<ver> -- backup-location- config s3Url=

Answer:

Run test backup: Velero backup create test-backup --include namespaces 'myapp'.	Apply BackupStorageLocation YAML.	Apply VolumeSnapshotLocation YAML.	Run the install command: Velero install -- provider aws --bucket <bucket> -- plugins velero/velero-plugin- for-vsphere:<ver> -- backup-location- config s3Url=
Run the install command: Velero install -- provider aws --bucket <bucket> -- plugins velero/velero-plugin- for-vsphere:<ver> -- backup-location- config s3Url=	Apply BackupStorageLocation YAML.	Apply VolumeSnapshotLocation YAML.	Run test backup: Velero backup create test-backup --include namespaces 'myapp'.

Explanation:

Answer (Correct Order):

Run the install command: `velero install ... --provider aws --bucket <bucket> ... --plugins ... --backup-location-config ...`

Apply BackupStorageLocation YAML.

Apply VolumeSnapshotLocation YAML.

Run test backup: `velero backup create test-backup --include-namespaces "myapp1"`

The correct sequence follows Velero's operational model: install the Velero components first, then define where backups and snapshots are stored, and finally validate with a real backup. In VCF 9.0, the Velero

Plugin for vSphere installation command includes parameters for the object-store provider, bucket, and plugin images, which establishes the Velero control plane in the target namespace and prepares it to communicate with an S3-compatible store.

After the installation is in place, you apply theBackupStorageLocationconfiguration so Velero has a durable destination for Kubernetes backup metadata in the object store. This aligns with the VCF 9.0 guidance that backups upload Kubernetes metadata to the object store and require S3-compatible storage for backup/restore workflows.

Next, apply theVolumeSnapshotLocationso Velero knows how and where to create/track volume snapshots for stateful workloads. The VCF 9.0 install example explicitly includes snapshot/backup location configuration parameters, reflecting that both must be set for complete protection.

Finally, run a test backup scoped to the namespace (--include-namespaces=my-namespace) to confirm end-to-end functionality.

5.A VKS administrator is tasked to leverage day-2 controls to monitor, scale, and optimize Kubernetes clusters across multiple operating systems and workload characteristics.

What two steps should the administrator take? (Choose two.)

- A. Configure namespace quotas to set resource limits for CPU, memory, and storage.
- B. Disable Cluster Autoscaler to ensure resources in the pool are not depleted.
- C. Deploy Prometheus and Grafana to collect and display scrapeable metrics on nodes, pods, and applications.
- D. Set all VM Class limits to Compute Heavy to ensure worker nodes get all the resources needed.
- E. Ensure all node pools use the same Machine Deployment configuration for different workload characteristics.

Answer: A C

Explanation:

VCF 9.0 describes a vSphere Namespace as the control point where administrators define resource boundaries for workloads, explicitly stating that vSphere administrators can create namespaces and “configure them with specified amount of memory, CPU, and storage,” and that you can “set limits for CPU, memory, storage” for a namespace. This directly supports step A as a day-2 control to keep multi-tenant clusters governed and prevent resource contention across different teams and workload types. For monitoring and optimization, VCF 9.0 explains that day-2 operations include visibility into utilization and operational metrics for VKS clusters, noting that application teams can use day-2 actions and gain insights

into CPU and memory utilization and advanced metrics (including contention and availability) for VKS clusters. In addition, VCF 9.0 monitoring guidance for VKS clusters states that Telegraf and Prometheus must be installed and configured on each VKS cluster before metrics and object details are sent for monitoring, and that VCF Operations supports metrics collection for Kubernetes objects (namespaces, nodes, pods, containers) via Prometheus. Since the Prometheus stack commonly includes Grafana dashboards for visualization, deploying Prometheus + Grafana matches the required monitoring/optimization outcome in C.