

VCAP-VKS LAB TASKS · EXAM 3V0-24.25

Work task 1 to task 170 and you have built, operated, broken and fixed VKS

Exam 3V0-24.25, Advanced VMware Cloud Foundation 9.0 vSphere Kubernetes Service. Tasks only, in build order. The code at the right of each task is the exam objective it covers.

Dr Pranay Jha · drpranayjha.com

A. DESIGN AND PLANNING

Write each answer down. One page total.

1. A customer has no NSX licence. State the workload network and the load balancer you will use for the Supervisor. 1.3
2. State the load balancer that will be used on an NSX VPC deployment where Avi has not been deployed. 2.2
3. List every valid workload network and load balancer pairing available in VCF 9.0. 1.3
4. A design must publish 300 virtual services. State the Avi Controller size and the minimum number of Service Engines per Supervisor. 3.1
5. State what changes about Supervisor functionality if the Avi Controller is deployed in the smallest size. 3.1
6. Draw the ingress path and the egress path for a pod in a VPC-backed namespace. Name every hop. 3.3
7. State how a workload in a namespace is given a predictable source address when it reaches an external database. 3.3
8. For each of these, decide VM or container and give one line of justification: a per-socket licensed legacy application, a stateless web front end, a GPU training job, a vendor appliance. 1.1
9. State what a zonal Supervisor across three vSphere Zones protects against, and what it requires of your storage policies. 1.3
10. List everything that must exist in a vSphere Namespace before a developer can deploy a virtual machine into it. 2.1
11. State which decisions made at Supervisor enablement cannot be changed afterwards without redeployment. 3.4
12. Write the enablement sequence for a Supervisor as an ordered list of steps, from prerequisites to first namespace. 3.4

B. SUPERVISOR ENABLEMENT

13. Enable a Supervisor on a single vSphere cluster using vDS networking and the Foundation Load Balancer. 4.1
14. Record the management network, workload network, and the ingress and egress ranges you assigned. 2.2
15. Verify the Supervisor is running and record how many control plane virtual machines were created. 2.1
16. Enable a Supervisor using NSX segment networking with the NSX load balancer. 4.1
17. Enable a Supervisor using NSX VPC networking with Avi. 4.1
18. Configure a zonal Supervisor across three vSphere Zones. 4.2
19. Configure a single-node Supervisor suitable for an edge site. 2.1
20. Download the Kubernetes CLI tools from the Supervisor and install both the kubectl vSphere plugin and the VCF CLI. 4.5
21. Create a VCF CLI context to the Supervisor and confirm you can list namespaces. 4.5
22. Log in to the same Supervisor using kubectl vsphere login and compare the contexts you now hold. 4.5
23. Change the Supervisor control plane size and record what the change affects. 2.1
24. Locate the Supervisor certificate and record where it would be replaced. 2.5b

C. NAMESPACES AND ZONES

- | | | |
|-----|--|-----|
| 25. | Create a vSphere Namespace named prod-apps. | 4.2 |
| 26. | Assign a storage policy to prod-apps and confirm the resulting StorageClass is visible from kubectl. | 2.3 |
| 27. | Bind two VM classes to prod-apps, one small and one large. | 4.2 |
| 28. | Associate a content library with prod-apps. | 2.5 |
| 29. | Apply a CPU, memory and storage limit to prod-apps. | 4.2 |
| 30. | Apply an object limit that caps the number of VKS clusters allowed in prod-apps. | 4.2 |
| 31. | Create a second namespace named dev-apps with a different storage policy and a single VM class. | 4.2 |
| 32. | From the Resources view of prod-apps, list every object type a namespace can contain. | 4.2 |
| 33. | Verify that a VM class bound to prod-apps is not usable from dev-apps. | 5.2 |
| 34. | Delete dev-apps and record what happened to the objects inside it. | 4.2 |
| 35. | Recreate dev-apps and confirm it is placed across your zones as expected. | 4.2 |

D. IDENTITY AND ACCESS

- | | | |
|-----|---|------|
| 36. | Grant a user view permission on prod-apps and confirm what they can and cannot do. | 2.4 |
| 37. | Grant a second user edit permission on prod-apps. | 2.4 |
| 38. | Grant a group owner permission on prod-apps. | 2.4 |
| 39. | Configure an external identity provider as the identity source and log in as a federated user. | 2.4 |
| 40. | Log in to a VKS cluster as a vCenter Single Sign-On user with kubectl. | 2.4 |
| 41. | Retrieve the cluster administrator kubeconfig from the namespace secret and use it to access the cluster. | 2.4 |
| 42. | Create a Kubernetes RoleBinding inside a VKS cluster that grants a Single Sign-On user the edit role in one namespace only. | 2.4 |
| 43. | Apply a Pod Security policy or admission configuration to a VKS cluster namespace and prove a privileged pod is refused. | 2.5b |

E. CONTENT LIBRARIES AND KUBERNETES RELEASES

- | | | |
|-----|--|-----|
| 44. | Create a subscribed content library for Kubernetes releases and synchronise it. | 2.5 |
| 45. | List every Kubernetes release now available to prod-apps. | 2.5 |
| 46. | Create a local content library and import a Kubernetes release into it manually. | 2.5 |
| 47. | Attach the local library to a namespace and confirm the release is offered there. | 2.5 |
| 48. | Document the full sequence for making Kubernetes releases available at an air-gapped site. | 2.5 |
| 49. | Identify a Kubernetes release that is not compatible with your current Supervisor version and record how you can tell. | 5.2 |
| 50. | Create a content library for virtual machine images and import one image for use with VM Service. | 2.5 |

F. VM SERVICE AND VSPHERE PODS

- | | | |
|-----|--|------|
| 51. | Create a virtual machine in prod-apps through VM Service using a YAML manifest. | 4.3 |
| 52. | Attach a second network interface and a persistent volume to that virtual machine. | 4.3 |
| 53. | Set a cloud-init user data block on a VM Service virtual machine so it is configured at first boot. | 4.3 |
| 54. | Expose the VM Service virtual machine through a load balancer service. | 4.12 |
| 55. | Deploy a vSphere Pod in prod-apps. | 4.3 |
| 56. | Record where the vSphere Pod runs and where a VKS cluster pod runs, and state the difference. | 1.1 |
| 57. | Delete the vSphere Pod and confirm the host-level object is removed. | 4.3 |
| 58. | Attempt to create a virtual machine using a VM class that is not bound to the namespace, and record the error. | 5.2 |

G. SUPERVISOR SERVICES

- | | | |
|-----|---|------|
| 59. | List the Supervisor Services currently installed. | 4.4 |
| 60. | Download and install the Harbor Supervisor Service. | 4.4 |
| 61. | Log in to Harbor and create a project. | 4.4 |
| 62. | Install the Contour Supervisor Service. | 4.4 |
| 63. | Install the External DNS Supervisor Service and configure it against a DNS zone. | 4.4 |
| 64. | Install the Velero Supervisor Service. | 4.4 |
| 65. | Install the CA Cluster Issuer service and issue a certificate from it. | 2.5b |
| 66. | Upgrade one Supervisor Service to a newer version. | 4.10 |
| 67. | Uninstall one Supervisor Service and confirm its objects are removed. | 4.4 |
| 68. | Record where you check that a service version is compatible with your Supervisor version. | 4.10 |

H. VKS CLUSTER PROVISIONING

- | | | |
|-----|--|------|
| 69. | Write a VKS cluster manifest for a cluster named vks-01 with one control plane node and two workers. | 4.14 |
| 70. | Provision vks-01 in prod-apps using kubectl. | 4.5 |
| 71. | Monitor the provisioning and list the cluster, machine and virtual machine objects as they appear. | 4.5 |
| 72. | Log in to vks-01 and confirm all nodes are Ready. | 4.5 |
| 73. | Export a kubeconfig for vks-01 using the VCF CLI cluster plugin. | 4.5 |
| 74. | Provision a second cluster vks-02 using the VCF CLI rather than kubectl. | 4.5 |
| 75. | Provision a cluster with three control plane nodes and record the difference in the machine objects. | 4.5 |
| 76. | Provision a cluster with two node pools that use different VM classes. | 4.5 |
| 77. | Provision a cluster that uses Calico instead of the default CNI. | 2.5b |
| 78. | Provision a cluster with a specified default StorageClass. | 4.11 |

- | | | |
|------------|---|------|
| 79. | Provision a cluster with an additional trusted certificate authority supplied in the specification. | 2.5b |
| 80. | Delete vks-02 and confirm every object it created has been removed. | 4.5 |

I. VKS NETWORKING AND CERTIFICATES

- | | | |
|------------|--|------|
| 81. | Expose an application in vks-01 using a Service of type LoadBalancer and record the address assigned. | 4.12 |
| 82. | Deploy an ingress controller in vks-01 and publish an application through it over HTTPS. | 4.12 |
| 83. | Create a NetworkPolicy that denies all traffic to a namespace, then allow one source. | 2.5b |
| 84. | Replace the certificate presented by an application with one signed by your own certificate authority. | 2.5b |
| 85. | Prove that a pod in vks-01 can resolve cluster DNS and reach an external address. | 5.2 |
| 86. | Record which load balancer served your LoadBalancer service and what its limits are. | 3.1 |
| 87. | Configure egress so that traffic leaving the namespace uses a known source address. | 3.3 |
| 88. | Deploy a workload that requires layer 7 routing rules and confirm whether your load balancer can serve it. | 3.1 |

J. STORAGE

- | | | |
|------------|---|------|
| 89. | List the StorageClasses available inside vks-01 and state where each came from. | 2.3 |
| 90. | Create a PersistentVolumeClaim that binds dynamically, and attach it to a pod. | 4.11 |
| 91. | Write a file to the volume, delete the pod, recreate it, and prove the data survived. | 4.11 |
| 92. | Expand the PersistentVolumeClaim and confirm the filesystem inside the pod grew. | 4.11 |
| 93. | Create a static PersistentVolume and bind a claim to it by hand. | 4.11 |
| 94. | Create a claim that requests a StorageClass that does not exist, and record the event produced. | 5.2 |
| 95. | Change the reclaim policy on a volume and record what now happens when the claim is deleted. | 4.11 |
| 96. | Add a second storage policy to prod-apps and verify a new StorageClass appears. | 2.3 |
| 97. | In a zonal Supervisor, deploy a workload to a zone whose datastores do not match the storage policy, and record the result. | 2.3 |

K. WORKLOADS, REGISTRY AND PACKAGES

- | | | |
|-------------|---|-----|
| 98. | Push a container image to Harbor. | 4.8 |
| 99. | Create a registry secret in a VKS namespace and deploy a workload that pulls from Harbor using it. | 4.8 |
| 100. | Delete the registry secret and record the exact pod event produced. | 5.3 |
| 101. | Deploy a workload from a registry whose certificate authority is not trusted, and record the error. | 5.3 |
| 102. | Make that registry trusted for every node of the cluster and prove the pull now succeeds. | 5.3 |

- | | | |
|------|---|------|
| 103. | Add a package repository to vks-01 and list the packages it offers. | 4.8 |
| 104. | Install a standard package into vks-01 and verify it is running. | 4.8 |
| 105. | Upgrade that package to a newer version. | 4.8 |
| 106. | Uninstall the package and confirm its resources are gone. | 4.8 |
| 107. | Deploy an application with a Deployment, a Service and an Ingress in one manifest, and reach it from outside the cluster. | 4.12 |

L. LIFECYCLE, SCALING AND AUTOSCALING

- | | | |
|------|---|------|
| 108. | Scale the worker node pool of vks-01 from two nodes to four. | 4.5 |
| 109. | Scale the pool back to two and record which nodes were removed. | 4.5 |
| 110. | Change the VM class used by the worker node pool and record whether nodes were replaced or edited in place. | 4.6 |
| 111. | Perform a rolling update of vks-01 to a newer Kubernetes release. | 4.6 |
| 112. | During the update, record the point at which an old node is removed. | 4.6 |
| 113. | Enable the cluster autoscaler on a node pool with a minimum of two and a maximum of five nodes. | 4.7 |
| 114. | Create enough pending workload to force the autoscaler to add a node. | 4.7 |
| 115. | Remove the workload and confirm the autoscaler scales the pool back down. | 4.7 |
| 116. | Upgrade the autoscaler configuration to a different minimum and maximum. | 4.7 |
| 117. | Deploy a pod with no resource requests, make it unschedulable, and record why the autoscaler does not help. | 4.7 |
| 118. | Add a HorizontalPodAutoscaler to a deployment and drive load into it until it scales. | 5.5 |
| 119. | Upgrade the Supervisor itself and record the order in which components are updated. | 4.10 |

M. BACKUP, RESTORE AND SNAPSHOTS

- | | | |
|------|---|------|
| 120. | Configure Velero against external S3-compatible object storage. | 4.13 |
| 121. | Back up a single namespace in vks-01, including its persistent volumes. | 4.13 |
| 122. | Delete that namespace and restore it from the backup. | 4.13 |
| 123. | Restore the same backup into a namespace with a different name. | 4.13 |
| 124. | Create a scheduled backup that runs nightly and confirm the schedule object exists. | 4.13 |
| 125. | Point the backup location at a bucket that does not exist and record how the failure appears. | 4.13 |
| 126. | Create a snapshot in a VKS cluster and restore from it. | 4.9 |
| 127. | State what a snapshot does not protect you against, and which task in this section does. | 4.9 |

N. SERVICE MESH

- | | | |
|------|--|-----|
| 128. | Install a service mesh into vks-01. | 3.6 |
| 129. | Enable the mesh for one namespace and confirm what was injected into each pod. | 3.6 |
| 130. | Prove that traffic between two services is now mutually authenticated. | 3.6 |

- | | | |
|------|--|-----|
| 131. | Shift traffic between two versions of a service using the mesh. | 3.6 |
| 132. | State what the mesh gives you that your ingress controller does not. | 3.6 |

O. TROUBLESHOOTING

Create each fault yourself where you can, then diagnose it from the symptom alone.

- | | | |
|------|--|------|
| 133. | A VKS cluster stays in Creating. Find the blocking condition and state the cause. | 5.1 |
| 134. | Remove the VM class binding from a namespace, provision a cluster, and diagnose the failure. | 5.2 |
| 135. | Remove the storage policy from a namespace, create a claim, and diagnose the failure. | 5.2 |
| 136. | Request a Kubernetes release that is not in the content library and diagnose the failure. | 5.2 |
| 137. | A namespace will not reach a ready state. Identify where you look first. | 5.1 |
| 138. | A LoadBalancer service has no external address. Identify the cause and state which load balancer is in play. | 5.1 |
| 139. | Exhaust the ingress address range and record how the next service fails. | 5.1 |
| 140. | A pod cannot resolve DNS inside a VKS cluster. Diagnose it. | 5.2 |
| 141. | A pod cannot reach the internet but can reach other pods. Diagnose it. | 5.2 |
| 142. | Diagnose an ImagePullBackOff caused by a missing registry secret. | 5.3 |
| 143. | Diagnose an ImagePullBackOff caused by an untrusted certificate authority. | 5.3 |
| 144. | Diagnose an application failing with a certificate error while image pulls on the same node succeed. | 5.3 |
| 145. | Collect the logs of the controllers responsible for cluster lifecycle on the Supervisor. | 5.1 |
| 146. | Open a shell on a VKS cluster node using the node SSH key from the namespace secret. | 5.1 |
| 147. | From a node, inspect the running containers and the images present. | 5.2 |
| 148. | From a node, find the log that shows whether the node finished its initial configuration. | 5.2 |
| 149. | Stall a cluster upgrade by removing available capacity, then recover it. | 5.4 |
| 150. | Stall a cluster upgrade with a PodDisruptionBudget, then recover it. | 5.4 |
| 151. | A Supervisor Service upgrade fails. State what you check before retrying. | 4.10 |
| 152. | Single Sign-On login to a VKS cluster fails. Regain cluster administrator access. | 5.1 |
| 153. | Produce a full diagnostic bundle for one namespace in a VKS cluster. | 5.1 |

P. OPTIMISATION

- | | | |
|------|--|-----|
| 154. | Identify the three pods consuming the most memory across all namespaces. | 5.5 |
| 155. | Find a container that is being CPU throttled while its node has spare capacity, and correct it. | 5.5 |
| 156. | Add accurate requests and limits to a workload that has none, and show the scheduling behaviour change. | 5.5 |
| 157. | Find a workload that cannot be scheduled and state whether the fix is the pod, the node pool, or the VM class. | 5.5 |
| 158. | Deploy monitoring into a VKS cluster and produce a chart of node and pod utilisation. | 5.5 |
| 159. | Forward VKS cluster logs to an external system. | 5.5 |

- 160.** Using your own data, recommend a node pool size and VM class for the workload you have been running.

5.5

Q. END TO END, AGAINST THE CLOCK

Do this in one sitting, from nothing, in under two hours. Repeat until it is routine.

- | | | |
|-------------|--|------|
| 161. | Enable a Supervisor with a stated networking model and load balancer. | 4.1 |
| 162. | Create a namespace with a storage policy, two VM classes, a content library and a quota. | 4.2 |
| 163. | Grant one user edit access to the namespace. | 2.4 |
| 164. | Provision a VKS cluster with two node pools and a trusted certificate authority. | 4.5 |
| 165. | Install Harbor and push an image to it. | 4.4 |
| 166. | Deploy an application from that image with a persistent volume and an ingress, reachable over HTTPS. | 4.12 |
| 167. | Enable autoscaling on the worker node pool. | 4.7 |
| 168. | Back the application namespace up to external object storage. | 4.13 |
| 169. | Upgrade the cluster to a newer Kubernetes release with no loss of service. | 4.6 |
| 170. | Delete the namespace and restore the application from backup. | 4.13 |