

VCAP VCF AUTOMATION TASKS · EXAM 3V0-21.25

Work task 1 to task 178 and you have built a platform other people can consume

Exam 3V0-21.25, VMware Certified Advanced Professional in VCF Automation. The blueprint publishes six objectives with forty-three sub-objectives underneath them, and thirty-five of those sit in section 4. Sections A and B are the platform underneath, C and D are written design work, E to H are the provider portal, I to M are the inside of an organization, N to T are consumption, Orchestrator and monitoring, and U is timed drills. The code at the right of each task is the objective it practises.

Dr Pranay Jha · drpranayjha.com

A. SUPERVISOR COMPONENTS

Objective 2.1. The platform VCF Automation sits on, so start here even if you came for the automation.

1. Draw a vSphere Supervisor and name every component it deploys. 2.1
2. Record how many control plane virtual machines a Supervisor creates and what each one does. 2.1
3. Write what a vSphere Namespace is, and why it is two things at once. 2.1
4. List what must be attached to a namespace before anything can be deployed into it. 2.1
5. Write the difference between a vSphere Pod, a VM created by VM Service, and a VKS cluster. 2.1
6. Record what a Supervisor Service is and name two of them. 2.1
7. Write what a vSphere Zone is and what a three-zone Supervisor protects against. 2.1
8. Record which Supervisor decisions cannot be changed after enablement. 2.1
9. Write how VCF Automation consumes the Supervisor, and which objects appear on each side. 2.1

B. SUPERVISOR NETWORKING

The second half of objective 2.1, and the part that decides what Automation can offer.

10. Write the three workload networking options for a Supervisor and the load balancer choices each allows. 2.1
11. Record which networking option VCF Automation expects for the full set of services. 2.1
12. Draw the NSX objects created per namespace on a segment-backed Supervisor. 2.1
13. Draw the NSX objects created for a VPC-backed Supervisor. 2.1
14. Compare the object count of both models at a hundred namespaces. 2.1
15. Write where ingress and egress addresses come from in each model. 2.1
16. Record what a Transit Gateway does in the VPC model. 2.1
17. Write what changes for Automation if the Supervisor was built on vDS networking only. 2.1

C. TENANCY MODELS AND ORGANIZATION TYPES

Objective 3.1, the design section. Three sub-objectives, and all of them are choosing questions.

18. List the VCFA tenancy deployment models and write one line on what each is for. 3.1
19. Write which model you would choose for a single internal IT department, and why. 3.1
20. Write which model you would choose for a service provider with external customers, and why. 3.1
21. Write the difference between an All Apps organization and a VM Apps organization in one sentence each. 3.1
22. List what an All Apps organization can offer that a VM Apps organization cannot. 3.1
23. Given a tenant who only runs virtual machines, choose an organization type and justify it. 3.1
24. Given a tenant who needs Kubernetes namespaces and self-service networking, choose and justify. 3.1
25. Write what it costs to change organization type later, and design so you do not have to. 3.1

D. REGIONS, CLOUD ZONES AND QUOTAS

The third sub-objective of 3.1, and the vocabulary everything else is built from.

- | | | |
|-----|---|-----|
| 26. | Write what a Region is in VCF Automation and what it maps to underneath. | 3.1 |
| 27. | Write what a Cloud Zone is and how it relates to a Region. | 3.1 |
| 28. | Write what a Quota controls and at which level it is set. | 3.1 |
| 29. | Draw Region, Cloud Zone, Organization and Project as nested objects, so the containment is unambiguous. | 3.1 |
| 30. | Write which of these the provider creates and which the tenant creates. | 3.1 |
| 31. | Write what happens to a deployment when the quota it draws from is exhausted. | 3.1 |
| 32. | Write what the remaining functional components are: flavors, VM classes, images and extensibility. | 3.1 |
| 33. | Design the Region and Cloud Zone layout for a two-site estate, and justify it. | 3.1 |

E. THE PROVIDER MANAGEMENT PORTAL

Objective 4.1 begins. Eleven sub-objectives, the largest single block outside Consumption.

- | | | |
|-----|---|-----|
| 34. | Sign in to the Provider Management portal and record what a provider administrator can see. | 4.1 |
| 35. | Configure the portal basics and record every setting that applies fleet-wide. | 4.1 |
| 36. | Configure the identity provider for the portal and sign in as a federated user. | 4.1 |
| 37. | Record what breaks if the identity provider is unreachable, and what still works. | 4.1 |
| 38. | Confirm the local provider account still works, and record where its credentials live. | 4.1 |
| 39. | Map a directory group to a provider role and test what that role can and cannot do. | 4.1 |
| 40. | Create a custom provider role and prove the boundary of one privilege you excluded. | 4.1 |

F. REGIONS AND PROVIDER CONTENT LIBRARIES

Still objective 4.1. What the provider makes available before any tenant exists.

- | | | |
|-----|---|-----|
| 41. | Create a Region in the Provider Management portal and record what it needs first. | 4.1 |
| 42. | Associate infrastructure with the Region and confirm it reports as available. | 4.1 |
| 43. | Create a second Region and record what is shared between them and what is not. | 4.1 |
| 44. | Create a provider content library and record where its storage comes from. | 4.1 |
| 45. | Publish an image into the provider content library. | 4.1 |
| 46. | Add a Kubernetes release to the library and confirm it becomes available to the Supervisor. | 4.1 |
| 47. | Record how an organization gets access to a provider library, and what the tenant sees. | 4.1 |
| 48. | Remove an item from the library and record what happens to deployments that used it. | 4.1 |

G. RIGHTS BUNDLES, DATA SERVICES, CERTIFICATES AND BRANDING

The remaining provider-level configuration in objective 4.1.

- | | | |
|-----|--|-----|
| 49. | Create a rights bundle and record what a rights bundle actually grants. | 4.1 |
| 50. | Use a rights bundle to give an organization access to VCF Operations Orchestrator. | 4.1 |

- | | | |
|-----|--|-----|
| 51. | Confirm from inside the organization that the Orchestrator access arrived. | 4.1 |
| 52. | Configure VCF Data Services in the Provider Management portal. | 4.1 |
| 53. | Record what Data Services offers a tenant and what it requires of the platform. | 4.1 |
| 54. | Configure provider-level certificate management and record which services present which certificate. | 4.1 |
| 55. | Replace a provider certificate and list everything that must be re-registered afterwards. | 4.1 |
| 56. | Configure organization branding and record what a tenant sees change. | 4.1 |
| 57. | Record which branding elements are set by the provider and which by the tenant. | 4.1 |

H. CREATING, ENABLING AND DELETING ORGANIZATIONS

The last of objective 4.1, and the most consequential actions in the portal.

- | | | |
|-----|--|-----|
| 58. | Create a new organization and record every decision the wizard asks for. | 4.1 |
| 59. | Record which of those decisions cannot be changed afterwards. | 4.1 |
| 60. | Assign infrastructure to the organization and prove it cannot see another organization. | 4.1 |
| 61. | Enable the Provider Consumption Organization and record what it is for. | 4.1 |
| 62. | Deploy one workload through the PCO and write how it differs from a tenant organization. | 4.1 |
| 63. | Record who should use the PCO and who should not. | 4.1 |
| 64. | Delete a test organization and record what the platform warns you about first. | 4.1 |
| 65. | Record what is removed on deletion and what is left behind. | 4.1 |

I. ORGANIZATION IDENTITY AND ACCESS CONTROL

Objective 4.2 begins. The tenant side of identity.

- | | | |
|-----|--|-----|
| 66. | Configure an identity provider inside a VCF All Apps organization. | 4.2 |
| 67. | Record why a tenant would want its own identity source rather than the provider one. | 4.2 |
| 68. | Map a directory group to an organization role and sign in as a member. | 4.2 |
| 69. | Configure the three common roles: organization owner, project member, viewer. | 4.2 |
| 70. | Create a custom organization role and test exactly what it can and cannot do. | 4.2 |
| 71. | Write the two layers of access: organization role, and project membership. | 4.2 |
| 72. | Record what happens to a user who is in the organization but in no project. | 4.2 |

J. INTEGRATIONS, CERTIFICATES AND SECRETS

Still objective 4.2. What the organization connects to and what it holds.

- | | | |
|-----|--|-----|
| 73. | Add the Git integration to a VCF All Apps organization and record what it enabled. | 4.2 |
| 74. | Add the Orchestrator integration and confirm it appears for project members. | 4.2 |
| 75. | Remove an integration and record what stops working. | 4.2 |
| 76. | Configure a certificate within the organization and record what presents it. | 4.2 |
| 77. | Create a secret in the organization and record where it can be used. | 4.2 |
| 78. | Record who can read a secret and who cannot, and test it. | 4.2 |
| 79. | Rotate a secret and confirm whatever used it still works. | 4.2 |

K. PROJECTS AND GOVERNANCE POLICIES

Where tenants are actually bounded, and the heart of objective 4.2.

- | | | |
|-----|--|-----|
| 80. | Create a Project and assign users, a region and a quota. | 4.2 |
| 81. | Sign in as a project member and confirm they see only their own project. | 4.2 |
| 82. | Create a second project and prove isolation between the two. | 4.2 |
| 83. | Exhaust a project quota and record the exact error a user receives. | 4.2 |
| 84. | Create a governance policy restricting which machine sizes may be deployed. | 4.2 |
| 85. | Test it from a user account with a disallowed size, and record the error. | 4.2 |
| 86. | Create a lease policy, deploy something, and watch the lease expire. | 4.2 |
| 87. | Record whether the owner was warned before expiry, and who can extend a lease. | 4.2 |
| 88. | Create an approval policy and approve a request as a different person. | 4.2 |
| 89. | Record where the audit trail of an approval is kept. | 4.2 |
| 90. | Write where you would deliberately not apply an approval policy, and why. | 4.2 |

L. NAMESPACE CLASSES

Its own sub-objective in 4.2, and the bridge from Automation to the Supervisor.

- | | | |
|-----|---|-----|
| 91. | Create a Namespace Class and record everything it defines. | 4.2 |
| 92. | Set the CPU, memory and storage limits on the class. | 4.2 |
| 93. | Select the storage classes and VM classes the namespace may use. | 4.2 |
| 94. | Deploy a namespace from the class as a project member. | 4.2 |
| 95. | Open the resulting vSphere Namespace in vCenter and confirm the limits came from the class. | 4.2 |
| 96. | Write what a Namespace Class controls that a Project does not. | 4.2 |
| 97. | Change the class and record what happens to namespaces already deployed from it. | 4.2 |

M. VPCS, CONNECTIVITY PROFILES AND IP BLOCKS

The networking sub-objectives of 4.2, including the ones easy to miss.

- | | | |
|------|---|-----|
| 98. | Create a VPC from within the organization. | 4.2 |
| 99. | Open NSX and list the objects created underneath it. | 4.2 |
| 100. | Create a VPC connectivity profile at organization level and record what it controls. | 4.2 |
| 101. | Attach two VPCs to the same profile and confirm they drew non-overlapping addresses. | 4.2 |
| 102. | Create a Private Transit Gateway IP Block and record what consumes it. | 4.2 |
| 103. | Create an IP Quota and record what it limits. | 4.2 |
| 104. | Exhaust an IP quota and record the error and where a provider would check it. | 4.2 |
| 105. | Write the difference between an IP block and an IP quota in one sentence each. | 4.2 |
| 106. | Record the regional network configuration an organization needs before any VPC can exist. | 4.2 |

N. ORGANIZATION CONTENT LIBRARIES AND IMAGES

Objective 4.3 begins, the largest objective in the blueprint at fifteen sub-objectives.

- | | | |
|------|--|-----|
| 107. | Create an Organizational Content Library within a VCF All Apps organization. | 4.3 |
| 108. | Record how it differs from the provider content library. | 4.3 |
| 109. | Upload an image into the organization library. | 4.3 |
| 110. | Record the image formats accepted and the one you would standardise on. | 4.3 |
| 111. | Deploy something from the uploaded image to prove the library is usable. | 4.3 |
| 112. | Create a Subscription to a published library and confirm items arrive. | 4.3 |
| 113. | Record the difference between downloading content immediately and on demand. | 4.3 |
| 114. | Change something in the source library and confirm the subscriber picks it up. | 4.3 |

O. DEPLOYING RESOURCES

Still 4.3. What a tenant actually does with the platform.

- | | | |
|------|--|-----|
| 115. | Deploy a virtual machine within a VCF All Apps organization. | 4.3 |
| 116. | Deploy a namespace and record which class it came from. | 4.3 |
| 117. | Deploy a VPC or network resource and record what was created underneath. | 4.3 |
| 118. | Deploy a Kubernetes cluster if the organization offers one, and record how long it took. | 4.3 |
| 119. | Record which resource types your organization can deploy and which it cannot. | 4.3 |
| 120. | Record where a deployment appears afterwards and what a user can do with it. | 4.3 |

P. CUSTOM PROPERTY GROUPS, RESOURCES AND ACTIONS

Three separate sub-objectives of 4.3, and where extensibility begins.

- | | | |
|------|---|-----|
| 121. | Create a Custom Property Group and record what it is for. | 4.3 |
| 122. | Use the property group in a deployment and confirm the values arrived. | 4.3 |
| 123. | Record the difference between a property group and a plain input on a blueprint. | 4.3 |
| 124. | Create a Custom Resource and record what it lets a tenant deploy that was not there before. | 4.3 |
| 125. | Record what a Custom Resource needs behind it to actually do anything. | 4.3 |
| 126. | Create a Custom Resource Action and attach it to a resource type. | 4.3 |
| 127. | Run the custom action against a deployed resource and confirm the result. | 4.3 |
| 128. | Write the difference between a Custom Resource Action and a Day 2 Action. | 4.3 |

Q. BLUEPRINTS AND THE SERVICE CATALOG

The centre of the Consumption objective.

- | | | |
|------|---|-----|
| 129. | Create a Blueprint that deploys a single resource. | 4.3 |
| 130. | Add inputs to the blueprint and record which a requester may change. | 4.3 |
| 131. | Version the blueprint and record what versioning gives you. | 4.3 |
| 132. | Publish the blueprint to the Service Catalog and share it with a project. | 4.3 |
| 133. | Sign in as an ordinary project member and request it from the catalog. | 4.3 |
| 134. | Confirm the requester could change only what the blueprint exposed. | 4.3 |

- | | | |
|------|--|-----|
| 135. | Create a blueprint that deploys more than one resource with a dependency between them. | 4.3 |
| 136. | Break a blueprint deliberately, read the validation error, and fix it. | 4.3 |
| 137. | Record where a failed deployment reports its reason, and the first place you would look. | 4.3 |
| 138. | Withdraw a catalog item and record what happens to existing deployments. | 4.3 |

R. DAY 2 ACTIONS

Its own sub-objective, and the part tenants use most after the first week.

- | | | |
|------|--|-----|
| 139. | List the Day 2 Actions available on a deployed virtual machine. | 4.3 |
| 140. | Resize a deployment using a Day 2 Action and confirm the change. | 4.3 |
| 141. | Use a Day 2 Action to change the lease on a deployment. | 4.3 |
| 142. | Record which Day 2 Actions require approval and which do not. | 4.3 |
| 143. | Restrict a Day 2 Action with a policy and test it from a user account. | 4.3 |
| 144. | Record where Day 2 Actions are audited. | 4.3 |
| 145. | Destroy a deployment through a Day 2 Action and confirm what was cleaned up. | 4.3 |

S. VCF OPERATIONS ORCHESTRATOR

Five sub-objectives of 4.3 on their own. Treat Orchestrator as its own subject.

- | | | |
|------|--|-----|
| 146. | Deploy and configure VCF Operations Orchestrator for a VCF All Apps organization. | 4.3 |
| 147. | Confirm the organization can reach it, using the rights bundle from section G. | 4.3 |
| 148. | Identify the components of an Orchestrator workflow and write what each one does. | 4.3 |
| 149. | Write the difference between a workflow, an action and a policy in Orchestrator. | 4.3 |
| 150. | Create a Workflow that takes an input and produces an output. | 4.3 |
| 151. | Add a decision element to the workflow and test both branches. | 4.3 |
| 152. | Run the workflow and read its execution log, including a failed run. | 4.3 |
| 153. | Create an Action and call it from a workflow. | 4.3 |
| 154. | Record when you would write an Action rather than put the logic in the workflow. | 4.3 |
| 155. | Install a VCF Operations Orchestrator Plug-in. | 4.3 |
| 156. | Configure the plug-in and use one of its elements in a workflow. | 4.3 |
| 157. | Record what a plug-in adds, and what it costs in upgrade dependency. | 4.3 |
| 158. | Call a workflow from a blueprint or a Custom Resource Action, joining the two halves of this exam. | 4.3 |

T. MONITORING WITH VCF OPERATIONS

Objective 5.1, the whole of section 5 in three sub-objectives.

- | | | |
|------|---|-----|
| 159. | Confirm VCF Operations is collecting from VCF Automation. | 5.1 |
| 160. | Find the VCF Automation objects in VCF Operations and record what is reported about each. | 5.1 |
| 161. | Monitor Provider Management: what is healthy, what is degraded, what is at capacity. | 5.1 |
| 162. | Monitor an All Apps organization: its projects, deployments and quota consumption. | 5.1 |
| 163. | Build an alert on an Automation condition and trigger it deliberately. | 5.1 |

- | | | |
|------|---|-----|
| 164. | Build a dashboard a provider administrator would actually use. | 5.1 |
| 165. | Build a view that shows a tenant what they are consuming. | 5.1 |
| 166. | Record which question each product answers: Operations, Operations for Logs, the portal itself. | 5.1 |
| 167. | Record what you would monitor to know a deployment failed before the tenant tells you. | 5.1 |

U. AGAINST THE CLOCK

Do this section twice in the final week, from a clean start, timing yourself.

- | | | |
|------|--|-----|
| 168. | Create an organization, assign infrastructure, and sign in as its administrator. 20 minutes. | 4.1 |
| 169. | Configure the organization identity provider and map two groups to two roles. 20 minutes. | 4.2 |
| 170. | Create a project with a quota, a governance policy and an approval policy. 25 minutes. | 4.2 |
| 171. | Create a Namespace Class and deploy a namespace from it. 20 minutes. | 4.2 |
| 172. | Create a VPC, a connectivity profile and an IP quota. 25 minutes. | 4.2 |
| 173. | Create a content library, upload an image, and deploy from it. 20 minutes. | 4.3 |
| 174. | Build a blueprint with inputs, publish it, and request it as an ordinary user. 30 minutes. | 4.3 |
| 175. | Build an Orchestrator workflow with an input, a decision and an action. 30 minutes. | 4.3 |
| 176. | Perform three Day 2 Actions on a deployment and record where each was audited. 15 minutes. | 4.3 |
| 177. | Build a monitoring view for a provider and one for a tenant. 20 minutes. | 5.1 |
| 178. | Write the Region, Cloud Zone, Organization, Project and Namespace Class hierarchy from memory. 10 minutes. | 3.1 |