

VCAP-VCF ARCHITECT DESIGN EXERCISES · EXAM 3V0-12.26

Work exercise 1 to exercise 146 and you have designed a VCF 9 private cloud on paper

Exam 3V0-12.26, Advanced VMware Cloud Foundation 9 Architect. This exam tests design, not operation: its twenty objectives sit in sections 1, 2 and 3 of the blueprint, and sections 4 and 5 have no testable objectives at all. Every exercise below produces a written artefact. The code at the right of each one is the objective it practises.

Dr Pranay Jha · drpranayjha.com

A. REQUIREMENTS, CONSTRAINTS, ASSUMPTIONS AND RISKS

Everything in this exam rests on telling these four apart. Do this section before any other.

1. Take a real project you worked on and write ten statements the customer actually made, in their words. 1.1
2. Classify each of those ten as a business requirement or a technical requirement, and defend any you found hard to place. 1.1
3. Write the test you used to decide: a business requirement says what the organisation needs, a technical requirement says what the solution must do. 1.1
4. Now classify the same ten as requirement, constraint, assumption or risk. Several will move. 1.2
5. Write the one-line test for each of the four, so you can apply it under time pressure. 1.2
6. Take three constraints and write what each one removes from the design space. 1.2
7. Take three assumptions and write what happens to the design if each proves false. 1.2
8. Convert one assumption into a risk, and write the mitigation that would let you close it. 1.2
9. Find a statement in your list that is really two requirements joined by "and", and split it. 1.1
10. Find a statement that sounds like a requirement but is actually a solution, and rewrite it as the need behind it. 1.1
11. Write a requirement that is not measurable, then rewrite it so it can be verified. 1.1

B. AMPRS AS A DESIGN LENS

Availability, manageability, performance, recoverability and security. The exam uses these five as its vocabulary, so you should too.

12. Write one sentence defining each of the five qualities, without using the word itself in the definition. 1.3
13. For each quality, write the question you would ask a customer to elicit it. 1.3
14. Take one design decision and write its effect on all five qualities, including the ones it harms. 1.3
15. Find a pair of qualities that conflict in a real design, and write how you would resolve the conflict. 1.3
16. Classify twenty requirements from a scenario against the five qualities, and note any that belong to none of them. 1.3
17. Write what distinguishes availability from recoverability, with an example of a design that has one and not the other. 1.3
18. Write what distinguishes manageability from monitoring, since the exam treats them as different objectives. 1.3

C. DESIGN DECISIONS AND JUSTIFICATION

A design decision without a justification is an opinion. This is the single most examinable skill in the blueprint.

19. Write a design decision in a fixed structure: the decision, the justification, the implication, and the requirement it satisfies. 1.4
20. Take ten decisions from a past design and rewrite them all in that structure. 1.4
21. For each decision, trace it back to a specific requirement, constraint or assumption. Delete any that trace to nothing. 1.4

- | | | |
|-----|--|-----|
| 22. | Write three decisions where the justification is a trade-off rather than a best practice. | 1.4 |
| 23. | Take one decision and write the alternative you rejected, with the reason you rejected it. | 1.4 |
| 24. | Find a decision in your own past work that was actually a default nobody questioned, and justify it properly or change it. | 1.4 |
| 25. | Write a decision whose implication is a new risk, and record that risk in the register. | 1.4 |
| 26. | Produce a one-page decision register for a small design, with every decision numbered and traced. | 1.4 |

D. DESIGN VALIDATION

How you prove the design meets the requirements before anyone builds it.

- | | | |
|-----|--|-----|
| 27. | Write what a design validation strategy contains and who it is for. | 1.5 |
| 28. | Take ten requirements and write the test that would prove each one is met. | 1.5 |
| 29. | Separate the tests that can only be run after build from the ones that can be checked on paper. | 1.5 |
| 30. | Write a validation plan covering functional verification, performance testing and failure testing. | 1.5 |
| 31. | Define the entry and exit criteria for validation, so there is an agreed definition of done. | 1.5 |
| 32. | Write what you do when a validation test fails: change the design, change the requirement, or accept a risk. | 1.5 |
| 33. | Produce a requirement to test traceability matrix for one small design. | 1.5 |

E. VCF ARCHITECTURE OPTIONS

Section 2 asks you to choose between architectures from a scenario. Build the comparison before you need it.

- | | | |
|-----|---|-----|
| 34. | List the VCF deployment architectures available in VCF 9 and write one line on what each is for. | 2.1 |
| 35. | Write the difference between a consolidated and a standard architecture, and the point at which you move from one to the other. | 2.1 |
| 36. | Write what a VCF instance, a fleet and a region mean, and how they nest. | 2.1 |
| 37. | Given a single-site scenario with 200 workloads, choose an architecture and justify it in three sentences. | 2.1 |
| 38. | Given a two-region scenario with a data residency requirement, choose an architecture and justify it. | 2.1 |
| 39. | Write what changes in the design when a second region is added later rather than planned from the start. | 2.1 |
| 40. | State the management domain sizing drivers, and what grows it as the estate grows. | 2.1 |
| 41. | Write when you would separate workload domains and when you would add clusters to an existing one. | 2.1 |

F. RECOMMENDED AGAINST SUPPORTED

A design can be supported and still be wrong. The exam tests whether you know the difference.

- | | | |
|-----|--|-----|
| 42. | Write the difference between a supported configuration and a recommended one, in one sentence. | 2.2 |
|-----|--|-----|

- | | | |
|-----|--|-----|
| 43. | Find three configurations that are supported but not recommended, and write why you would avoid each. | 2.2 |
| 44. | Write where you would check support status for a given combination, and how current that source is. | 2.2 |
| 45. | Take a design where a constraint forces a supported but not recommended choice, and write the risk you record. | 2.2 |
| 46. | Write what you do when the customer demands a configuration that is not supported at all. | 2.2 |
| 47. | Write how the interoperability matrix and the compatibility guide differ, and when you use each. | 2.2 |
| 48. | Produce a short list of version and interoperability checks you would run before signing off any VCF design. | 2.2 |

G. VCF ADVANCED SERVICES

Knowing which service answers which need, and when none of them do.

- | | | |
|-----|--|-----|
| 49. | List the VCF Advanced Services and write the use case for each in one line. | 2.3 |
| 50. | Given a requirement for self-service application deployment, say which service answers it and why. | 2.3 |
| 51. | Given a requirement for container workloads alongside virtual machines, say which service answers it. | 2.3 |
| 52. | Given a requirement for private AI workloads, say which service answers it and what it demands of the physical design. | 2.3 |
| 53. | Write what each Advanced Service adds to the management domain in resource terms. | 2.3 |
| 54. | Find a requirement that sounds like it needs an Advanced Service but does not, and justify leaving it out. | 2.3 |

H. DISCOVERY AND ANALYSIS

Section 3 begins here. Everything downstream is only as good as this.

- | | | |
|-----|--|-----|
| 55. | Write the ten questions you would ask in a first discovery workshop, in the order you would ask them. | 3.1 |
| 56. | Take a one-page customer brief and extract every requirement from it, numbered. | 3.1 |
| 57. | Identify what the brief does not say that you must ask about before designing. | 3.1 |
| 58. | Separate the stated requirements from the implied ones, and get the implied ones confirmed in writing. | 3.1 |
| 59. | Write the business objectives behind the requirements, in the customer language rather than yours. | 3.1 |
| 60. | Prioritise the requirements as must, should and could, and record who agreed the priority. | 3.1 |
| 61. | Find two requirements that conflict, and write the question you would ask to resolve them. | 3.1 |

I. CONCEPTUAL MODEL

The layer most people skip. The exam does not.

- | | | |
|-----|---|-----|
| 62. | Write what a conceptual model contains, and what it deliberately leaves out. | 3.2 |
| 63. | Produce a conceptual model for a scenario: entities, their relationships, and the qualities required of each. | 3.2 |
| 64. | Map every business objective to an element of your conceptual model. | 3.2 |
| 65. | Write why the conceptual model contains no product names, and what breaks if you put them in. | 3.2 |
| 66. | Take a conceptual model and write the questions it raises that the logical design must answer. | 3.2 |
| 67. | Present the model as one diagram a business sponsor could read without help. | 3.2 |

J. LOGICAL DESIGN

What the solution does, still without deciding what it is built from.

- | | | |
|-----|--|-----|
| 68. | Write the difference between logical and physical design, with an example of each for the same component. | 3.3 |
| 69. | Produce a logical design for compute: clusters, their purpose, their sizing drivers, with no hardware named. | 3.3 |
| 70. | Produce a logical design for storage: policies, tiers, protection levels, with no vendor named. | 3.3 |
| 71. | Produce a logical design for networking: segments, routing boundaries, isolation requirements, with no device named. | 3.3 |
| 72. | Produce a logical design for management and operations components. | 3.3 |
| 73. | Trace every element of the logical design back to the conceptual model. | 3.3 |
| 74. | Find a logical design element that has crept into physical detail, and move it to the right layer. | 3.3 |

K. PHYSICAL DESIGN

The layer that gets built. Every number here needs a reason.

- | | | |
|-----|---|-----|
| 75. | Turn your logical compute design into a physical one: host specification, host count, cluster layout. | 3.4 |
| 76. | Show the calculation behind the host count, including the failure capacity you reserved. | 3.4 |
| 77. | Turn your logical storage design into a physical one: architecture, policies, capacity with overhead shown. | 3.4 |
| 78. | Turn your logical network design into a physical one: switches, uplinks, VLANs, gateways, address allocation. | 3.4 |
| 79. | Size the management domain and list every appliance in it with its deployment size. | 3.4 |
| 80. | Write the physical design decisions that were forced by a constraint rather than chosen. | 3.4 |
| 81. | Produce the bill of materials implied by your physical design. | 3.4 |
| 82. | Check every physical number against a published maximum, and record where you are close to one. | 3.4 |

L. DESIGN FOR AVAILABILITY

The first of the five quality objectives, each examined in its own right.

- | | | |
|-----|---|-----|
| 83. | Write the availability requirement as a number, and state what it is measured against. | 3.5 |
| 84. | Design cluster sizing and admission control to meet that number, showing the capacity cost. | 3.5 |
| 85. | Decide where you need stretched clusters and where you do not, and justify both. | 3.5 |
| 86. | Write the single points of failure remaining in your design, and whether each is accepted or mitigated. | 3.5 |
| 87. | Design availability for the management components, not only the workloads. | 3.5 |
| 88. | Write what your design does when a whole site is lost, step by step. | 3.5 |
| 89. | Record the availability design decisions with their trade-off against cost. | 3.5 |

M. DESIGN FOR MANAGEABILITY

How the thing is run after you leave.

- | | | |
|-----|--|-----|
| 90. | Write what manageability covers in this blueprint, and how it differs from monitoring. | 3.6 |
| 91. | Design the lifecycle approach: how components are patched and upgraded, and in what order. | 3.6 |
| 92. | Design for configuration consistency, including how drift is detected and corrected. | 3.6 |
| 93. | Design the operational model: who does what, and which tasks are self-service. | 3.6 |
| 94. | Write the manageability cost of each major design decision you have made. | 3.6 |
| 95. | Find a decision that improved performance and made the design harder to run, and record the trade. | 3.6 |

N. DESIGN FOR PERFORMANCE

Numbers, and the evidence behind them.

- | | | |
|------|--|-----|
| 96. | Write the performance requirements as measurable figures, not adjectives. | 3.7 |
| 97. | Design compute sizing to meet them, stating the consolidation ratio you assumed and why. | 3.7 |
| 98. | Design storage to meet a stated latency and throughput requirement. | 3.7 |
| 99. | Design the network to meet a stated throughput requirement, including east-west traffic. | 3.7 |
| 100. | Identify the first bottleneck your design would hit under growth, and at what point. | 3.7 |
| 101. | Write what you would measure after build to prove the performance requirements are met. | 3.7 |
| 102. | Record where you traded performance for cost, and get it agreed. | 3.7 |

O. DESIGN FOR RECOVERABILITY

What happens after something is lost, and how long it takes.

- | | | |
|------|---|-----|
| 103. | Write the recovery point and recovery time objectives, per workload tier, as numbers. | 3.8 |
| 104. | Design backup for the workloads to meet those numbers. | 3.8 |
| 105. | Design backup and restore for the management components, including the restore order. | 3.8 |

- | | | |
|------|---|-----|
| 106. | Design disaster recovery between regions, and state what is replicated and what is rebuilt. | 3.8 |
| 107. | Write the recovery runbook at a level a second engineer could follow. | 3.8 |
| 108. | Write how you would test recovery, and how often. | 3.8 |
| 109. | State the difference between your availability design and your recoverability design in one sentence. | 3.8 |

P. DESIGN FOR SECURITY

Security as a design layer, not a checklist at the end.

- | | | |
|------|---|-----|
| 110. | Extract the security and compliance requirements from a scenario, including the ones implied by the industry. | 3.9 |
| 111. | Design network segmentation and the firewall model, and say where each rule is enforced. | 3.9 |
| 112. | Design encryption: what is encrypted at rest, what in transit, and where the keys live. | 3.9 |
| 113. | Design certificate management across the fleet, including renewal. | 3.9 |
| 114. | Design role separation so that no single administrator can act unobserved. | 3.9 |
| 115. | Design logging and audit retention to meet the compliance requirement. | 3.9 |
| 116. | Write the residual security risks and who accepted them. | 3.9 |

Q. DESIGN FOR IDENTITY MANAGEMENT

Its own objective in this blueprint, so treat it as its own design.

- | | | |
|------|--|------|
| 117. | Write the identity requirements: who signs in, to what, and from where. | 3.10 |
| 118. | Design the identity provider topology and its availability. | 3.10 |
| 119. | Design group to role mapping across the fleet, by group and never by user. | 3.10 |
| 120. | Design multi-factor authentication and state where it applies and where it cannot. | 3.10 |
| 121. | Design the break-glass path for when the identity provider is unavailable. | 3.10 |
| 122. | Write what happens to access when a person leaves, and who performs it. | 3.10 |

R. CONSUMPTION STRATEGY

How the platform is actually used by the people it was built for.

- | | | |
|------|---|------|
| 123. | Write who consumes this platform and in what units: virtual machines, namespaces, applications. | 3.11 |
| 124. | Design the tenancy model: organizations, projects, namespaces, and the boundaries between them. | 3.11 |
| 125. | Design the self-service catalogue and what a consumer may and may not change. | 3.11 |
| 126. | Design quotas and the policy for what happens when one is exhausted. | 3.11 |
| 127. | Design the chargeback or showback model and the figures it rests on. | 3.11 |
| 128. | Design the request and approval flow, including where it should not apply. | 3.11 |
| 129. | Write how consumption changes the capacity plan, since self-service makes demand unpredictable. | 3.11 |

S. MONITORING STRATEGY

The last design objective, and the one that tells you whether the rest worked.

- | | | |
|------|---|------|
| 130. | Write what the monitoring strategy must answer, and for whom. | 3.12 |
| 131. | Design what is collected, at what interval, and how long it is retained. | 3.12 |
| 132. | Design alerting: which conditions page someone, which raise a ticket, and which are informational only. | 3.12 |
| 133. | Design capacity monitoring and the lead time it must give you before you run out. | 3.12 |
| 134. | Design cost monitoring and who receives it. | 3.12 |
| 135. | Design the dashboards for three audiences: the operator, the service owner, and the sponsor. | 3.12 |
| 136. | Write how monitoring proves each AMPRS quality is being met in production. | 3.12 |

T. A FULL DESIGN, AGAINST THE CLOCK

Do this section twice in the final week, from a scenario you have not seen before.

- | | | |
|------|---|------|
| 137. | From a one-page brief, extract and number every requirement, constraint, assumption and risk. 25 minutes. | 1.2 |
| 138. | Produce the conceptual model from those requirements. 20 minutes. | 3.2 |
| 139. | Produce the logical design for compute, storage and network. 40 minutes. | 3.3 |
| 140. | Produce the physical design with sizing shown. 40 minutes. | 3.4 |
| 141. | Write twenty design decisions with justification, implication and traceability. 35 minutes. | 1.4 |
| 142. | Write the availability and recoverability design against stated numbers. 25 minutes. | 3.5 |
| 143. | Write the security and identity design. 25 minutes. | 3.9 |
| 144. | Write the consumption and monitoring strategy. 25 minutes. | 3.11 |
| 145. | Produce the validation plan and the traceability matrix. 20 minutes. | 1.5 |
| 146. | Present the whole design in ten minutes to someone who will challenge it. | 1.4 |