

VCAP VCF SUPPORT TASKS · EXAM 3V0-13.26

Work task 1 to task 214 and you have broken and fixed every part of VCF

Exam 3V0-13.26, VMware Certified Advanced Professional in VCF Support. The mirror of the Administrator exam: sections 1 to 4 of the blueprint have no testable objectives, and all thirty sit under Troubleshoot and Optimise, of which twenty-nine are distinct. Section A builds the method, B to F are compute, G to I are storage, J is Automation, K to N are Supervisor and VKS, O to T are the Operations stack, U to W are NSX, and X is timed drills with faults somebody else created. Almost every task asks you to create the fault first. The code at the right of each task is the objective it practises.

Dr Pranay Jha · drpranayjha.com

A. THE TROUBLESHOOTING TOOLKIT

Objective 5.5. Know what each tool is for before you need one in a hurry.

- | | | |
|----|--|-----|
| 1. | List every tool VCF gives you for troubleshooting, and write one line on each. | 5.5 |
| 2. | Record which tool answers what is broken, and which answers what changed. | 5.5 |
| 3. | Record which tool answers what is slow, and which answers who did it. | 5.5 |
| 4. | Record where each tool gets its data from, and how fresh that data is. | 5.5 |
| 5. | Record which tools need something configured before they are useful. | 5.5 |
| 6. | Record which require a licence or an edition you may not have. | 5.5 |
| 7. | Build one table: the symptom, the first tool, and the second tool. | 5.5 |
| 8. | Record what you would use if the management plane itself is down. | 5.5 |
| 9. | Write the order you open things in, and keep it where you will see it. | 5.5 |

B. ESX HOST FAULTS

Objective 5.6. The layer everything else sits on.

- | | | |
|-----|--|-----|
| 10. | List the ways an ESX host can be unhealthy without being down. | 5.6 |
| 11. | Record where host logs live and which log covers what. | 5.6 |
| 12. | Disconnect a host from vCenter and diagnose it from the symptoms. | 5.6 |
| 13. | Record the difference between not responding, disconnected and isolated. | 5.6 |
| 14. | Break host time synchronisation and record what starts failing. | 5.6 |
| 15. | Fill a host log partition and record what the host does. | 5.6 |
| 16. | Diagnose a host that will not exit maintenance mode. | 5.6 |
| 17. | Record how you would check hardware health from VCF. | 5.6 |
| 18. | Record what a purple diagnostic screen gives you and what to collect. | 5.6 |
| 19. | Record the order you would work a host that is unresponsive but still running workloads. | 5.6 |

C. VCENTER FAULTS

Objective 5.7. When the thing you troubleshoot with is the thing that is broken.

- | | | |
|-----|--|-----|
| 20. | Record the vCenter services and which ones everything else depends on. | 5.7 |
| 21. | Record where vCenter logs live and which to read first. | 5.7 |
| 22. | Stop a non-critical vCenter service and record what stopped working. | 5.7 |
| 23. | Diagnose a vCenter that is up but will not accept logins. | 5.7 |
| 24. | Record the certificate problems that present as a login failure. | 5.7 |
| 25. | Fill the vCenter database partition in a lab and record the behaviour. | 5.7 |
| 26. | Record how to check vCenter health and backup status in VCF. | 5.7 |
| 27. | Record what still works in the workload domain while vCenter is down. | 5.7 |
| 28. | Record the recovery options, in order of how much you would lose. | 5.7 |

D. CLUSTER FAULTS

Objective 5.8. HA, DRS, EVC and the things that stop a cluster behaving.

- | | | |
|-----|---|-----|
| 29. | Record what HA actually protects against, and what it does not. | 5.8 |
|-----|---|-----|

30.	Break HA network heartbeating and record what the cluster reports.	5.8
31.	Diagnose an HA configuration error on one host.	5.8
32.	Record what admission control does when it refuses a power-on.	5.8
33.	Cause an admission control failure deliberately and read the error.	5.8
34.	Diagnose a DRS cluster that is not balancing, and record why.	5.8
35.	Record how an affinity rule can prevent a migration, and find one that does.	5.8
36.	Record what EVC does and the error when a machine cannot move because of it.	5.8
37.	Record the order you would work a cluster where nothing will power on.	5.8

E. VIRTUAL MACHINE FAULTS

Objective 5.9. The layer the complaint actually arrives at.

38.	Record the common reasons a virtual machine will not power on.	5.9
39.	Cause a power-on failure and read the error to its root cause.	5.9
40.	Diagnose a machine that will not migrate, and record the reason given.	5.9
41.	Record what an orphaned or inaccessible machine means and how each happens.	5.9
42.	Create an inaccessible machine and recover it.	5.9
43.	Diagnose a machine with no network connectivity, from the virtual side only.	5.9
44.	Record where a machine logs its own problems, and read one.	5.9
45.	Diagnose a machine that is slow but shows no alarms.	5.9
46.	Record what VMware Tools state affects, and what it does not.	5.9
47.	Record the five questions you would ask the person reporting the fault.	5.9

F. OPTIMISING DOMAINS AND WORKLOADS

Objectives 5.10 and 5.11. Where troubleshooting turns into tuning.

48.	Record the four resources you optimise, and the metric that matters for each.	5.10
49.	Find the busiest cluster in a workload domain and record what is driving it.	5.10
50.	Record what CPU ready tells you and at what level it becomes a problem.	5.10
51.	Record what memory ballooning and swapping each indicate.	5.10
52.	Record what storage latency figures are acceptable, and where you measure them.	5.10
53.	Produce one optimisation recommendation for the domain, with evidence.	5.10
54.	Find an oversized machine and record the evidence that it is oversized.	5.11
55.	Find a machine that is contending, and identify what it is contending with.	5.11
56.	Record what you would change on that machine, and what you would not.	5.11
57.	Make one change, measure before and after, and record whether it worked.	5.11

G. MONITORING STORAGE

Objectives 5.1 and 5.2. The same wording appears as 5.30, which is a duplicate.

58.	List the tools in VCF for monitoring vSAN and record what each shows.	5.1
59.	Run the vSAN health checks and record every category.	5.1
60.	Open vSAN performance and record what is available per cluster, host, disk and machine.	5.1
61.	Record what the vSAN capacity view reports beyond used and free.	5.1

- | | | |
|-----|--|-----|
| 62. | Record which vSAN metric would exonerate storage from a performance complaint. | 5.1 |
| 63. | Monitor a non-vSAN datastore and record the metrics available. | 5.2 |
| 64. | Record what you can see about the array from VCF, and what you cannot. | 5.2 |
| 65. | Record the visibility boundary, and what you must ask the storage team for. | 5.2 |
| 66. | Build one table: the storage symptom, and which tool shows it. | 5.2 |

H. VSAN FAULTS

Objective 5.3. Create each fault, then diagnose it.

- | | | |
|-----|--|-----|
| 67. | Write your diagnostic order for a vSAN performance complaint. | 5.3 |
| 68. | Break vSAN networking on one host and diagnose it blind. | 5.3 |
| 69. | Record which health checks turned red, in what order, and how quickly. | 5.3 |
| 70. | Create a non-compliant object and resolve it. | 5.3 |
| 71. | Cause a resynchronisation and work out why it is running from the evidence. | 5.3 |
| 72. | Record how to tell a necessary resynchronisation from a worrying one. | 5.3 |
| 73. | Fill a vSAN datastore past a threshold and record what happens, in order. | 5.3 |
| 74. | Record what a failed capacity device looks like and the recovery steps. | 5.3 |
| 75. | Record what differs between OSA and ESA when diagnosing a device failure. | 5.3 |
| 76. | Diagnose an object that is inaccessible, and record the difference from non-compliant. | 5.3 |

I. NON-VSAN STORAGE FAULTS

Objective 5.4. Different tools, different language, same pressure.

- | | | |
|-----|---|-----|
| 77. | Break one path to a datastore and record the failover, and how long it took. | 5.4 |
| 78. | Record what an all paths down event is and what causes it. | 5.4 |
| 79. | Record what permanent device loss is and how the host behaves differently. | 5.4 |
| 80. | Record how you would tell those two apart from the logs alone. | 5.4 |
| 81. | Diagnose a datastore that will not mount on one host only. | 5.4 |
| 82. | Record the usual causes of a host-specific storage fault, in order of likelihood. | 5.4 |
| 83. | Record where multipathing state is checked and what healthy looks like. | 5.4 |
| 84. | Record the evidence you would bring to the storage team, and what it proves. | 5.4 |

J. MONITORING VCF AUTOMATION

Objectives 5.12, 5.13 and 5.14. Three objectives on watching the tenant platform.

- | | | |
|-----|---|------|
| 85. | Confirm VCF Operations is collecting from VCF Automation, and record the last collection. | 5.12 |
| 86. | Find the VCF Automation objects in Operations and record what is reported for each. | 5.12 |
| 87. | Record which Automation failure would be visible here before a tenant reported it. | 5.12 |
| 88. | Monitor Provider Management: record what healthy, degraded and at capacity look like. | 5.13 |
| 89. | Record what you would check every morning on the provider side. | 5.13 |
| 90. | Monitor an All Apps organization: projects, deployments and quota consumption. | 5.14 |
| 91. | Find an organization close to its quota and record what the tenant will see next. | 5.14 |
| 92. | Diagnose a failed deployment using Operations rather than the Automation console. | 5.14 |

93. Record which question you take to Operations and which to the Automation portal. 5.14

K. SUPERVISOR AND VKS FAULTS

Objective 5.15. Provisioning, connectivity and namespace errors.

94. Record the stages a VKS cluster passes through while being provisioned. 5.15
95. Record where a stuck provisioning job reports its reason. 5.15
96. Diagnose a VKS cluster that never finishes provisioning. 5.15
97. Record the usual causes: content library, networking, capacity, class, certificates. 5.15
98. Break Supervisor connectivity and record what still works and what does not. 5.15
99. Diagnose a namespace that will not accept a workload. 5.15
100. Record what to check when kubectl cannot reach the Supervisor at all. 5.15
101. Record which logs on which component you would collect for this class of fault. 5.15
102. Write the diagnostic order for a cluster that will not provision. 5.15

L. CONTAINERS, REGISTRIES AND TRUSTED CAS

Objective 5.16. The faults that look like Kubernetes and are not.

103. Record what happens, in order, when a pod cannot pull its image. 5.16
104. Cause an image pull failure and read the event to its root cause. 5.16
105. Record the difference between an authentication failure and a certificate failure here. 5.16
106. Configure a registry with an untrusted certificate and diagnose the result. 5.16
107. Record where a trusted CA is supplied to the Supervisor and to a VKS cluster. 5.16
108. Fix the trust and confirm the pull succeeds. 5.16
109. Record what to check when only some nodes can pull an image. 5.16
110. Record the three commands you would run first for any container start failure. 5.16

M. FAILED VKS UPGRADES

Objective 5.17. Restart or recover, which is its own objective for a reason.

111. Record the stages of a VKS cluster upgrade. 5.17
112. Record where upgrade progress and failure are reported. 5.17
113. Record the usual causes of a stalled upgrade. 5.17
114. Record what state a cluster is left in when an upgrade fails partway. 5.17
115. Record how to restart a failed upgrade, and what that does. 5.17
116. Record when restarting is wrong and recovery is needed instead. 5.17
117. Record what you would collect before attempting either. 5.17

N. CLUSTER PERFORMANCE AND SCALING

Objective 5.18. Optimising with the monitoring you already have.

118. Record what monitoring is available for a VKS cluster and where it comes from. 5.18
119. Find a cluster under pressure and record which resource is short. 5.18
120. Record the scaling options available and what each one changes. 5.18
121. Record the difference between scaling the cluster and scaling the workload. 5.18

- | | | |
|------|---|------|
| 122. | Scale a cluster and record how long it took and what it cost. | 5.18 |
| 123. | Record what autoscaling needs configured before it will work. | 5.18 |
| 124. | Record what you would check before concluding a cluster needs more nodes. | 5.18 |
| 125. | Produce one scaling recommendation with the evidence behind it. | 5.18 |

O. VCF OPERATIONS INFRASTRUCTURE

Objective 5.19. Cloud proxies, clusters, availability and adapters.

- | | | |
|------|---|------|
| 126. | Draw the VCF Operations deployment and label every node and collector. | 5.19 |
| 127. | Record what a cloud proxy does and what breaks when it cannot reach the cluster. | 5.19 |
| 128. | Diagnose a cloud proxy that shows as disconnected. | 5.19 |
| 129. | Record what a collector group gives you and what happens when a member fails. | 5.19 |
| 130. | Diagnose an adapter instance that has stopped collecting. | 5.19 |
| 131. | Record the difference between an adapter that is failing and one that is just slow. | 5.19 |
| 132. | Record how high availability behaves in an Operations cluster, and its limits. | 5.19 |
| 133. | Record where Operations reports its own health, and what to check first. | 5.19 |
| 134. | Record what you would do when Operations itself is the thing that is broken. | 5.19 |

P. CONFIGURATION ISSUES THROUGH OPERATIONS

Objective 5.20. The what-changed question.

- | | | |
|------|--|------|
| 135. | Open Config Drift and record what it compares against. | 5.20 |
| 136. | Record where the baseline came from and who can change it. | 5.20 |
| 137. | Change a setting deliberately and find it reported as drift. | 5.20 |
| 138. | Record the detection delay and whether anybody was told. | 5.20 |
| 139. | Use Operations to find what changed on an object around a known time. | 5.20 |
| 140. | Record what compliance benchmarks add to this picture. | 5.20 |
| 141. | Diagnose a problem whose root cause is a configuration change. | 5.20 |
| 142. | Record how you would prove a change caused an incident, with evidence. | 5.20 |

Q. VCF OPERATIONS FOR NETWORKS

Objective 5.21. Flows, paths and the physical world.

- | | | |
|------|---|------|
| 143. | Confirm it is collecting, and record from which sources. | 5.21 |
| 144. | Trace the path between two machines end to end. | 5.21 |
| 145. | Record what the flow analysis shows that NSX alone does not. | 5.21 |
| 146. | Diagnose a connectivity problem using the path view. | 5.21 |
| 147. | Record what it tells you about a firewall rule that is never hit. | 5.21 |
| 148. | Record what a missing data source makes invisible. | 5.21 |
| 149. | Use it to answer one real question about east-west traffic. | 5.21 |
| 150. | Record when you reach for this product rather than NSX or Logs. | 5.21 |

R. WORKLOAD ISSUES THROUGH OPERATIONS

Objective 5.22. Starting from the complaint rather than the component.

- | | | |
|------|---|------|
| 151. | Open the Troubleshooting Workbench on a workload and record what it shows. | 5.22 |
| 152. | Use it to find what changed around the time of a complaint. | 5.22 |
| 153. | Work a real or staged slow-workload complaint using the workbench alone. | 5.22 |
| 154. | Record which evidence it gave you and which you had to find elsewhere. | 5.22 |
| 155. | Record what a business application view adds when one tier is unhealthy. | 5.22 |
| 156. | Diagnose a workload whose problem turns out to be a neighbour. | 5.22 |
| 157. | Record how you would prove a workload problem is not the platform. | 5.22 |
| 158. | Write the order you would work a slow virtual machine complaint in this tool. | 5.22 |

S. VCF OPERATIONS FOR LOGS

Objective 5.23. The what exactly happened question.

- | | | |
|------|--|------|
| 159. | Confirm Logs is receiving events and list the sources. | 5.23 |
| 160. | Record which VCF components send to it and which do not. | 5.23 |
| 161. | Run an interactive query and narrow it to one event type. | 5.23 |
| 162. | Take a real incident and find its first error in the logs. | 5.23 |
| 163. | Record how you would correlate across two components by timestamp. | 5.23 |
| 164. | Install a content pack and record what it added. | 5.23 |
| 165. | Create a log event alert query and trigger it. | 5.23 |
| 166. | Record what you would check when the logs show nothing at all. | 5.23 |
| 167. | Record which question goes to Logs and which to Operations. | 5.23 |

T. OPERATIONS ALERTS

Objective 5.24. The alerts themselves can be the problem.

- | | | |
|------|---|------|
| 168. | Open an alert definition and record its symptoms, impact and recommendations. | 5.24 |
| 169. | Record why an alert might fire when nothing is wrong. | 5.24 |
| 170. | Record why an alert might not fire when something is. | 5.24 |
| 171. | Find an alert that is firing and work it to a root cause or a false positive. | 5.24 |
| 172. | Change a threshold in a policy and confirm the alert state changed. | 5.24 |
| 173. | Diagnose a notification that was never delivered. | 5.24 |
| 174. | Record where delivery failures are visible. | 5.24 |
| 175. | Record how you would tune a noisy alert without hiding a real problem. | 5.24 |

U. NSX: THE RIGHT TOOL, AND INFRASTRUCTURE FAULTS

Objectives 5.25 and 5.26.

- | | | |
|------|---|------|
| 176. | List the tools available for troubleshooting NSX and what each is for. | 5.25 |
| 177. | Record which tool answers is it configured, and which answers is traffic flowing. | 5.25 |
| 178. | Record when you would use Traceflow, and what it cannot tell you. | 5.25 |
| 179. | Record when port mirroring or a packet capture is the only answer. | 5.25 |
| 180. | Build one table: the NSX symptom, and the first tool to open. | 5.25 |

181.	Record the NSX management and control plane components and what each failure costs.	5.26
182.	Diagnose a transport node that shows as down.	5.26
183.	Break the tunnel between two transport nodes and diagnose it from the symptoms.	5.26
184.	Record what an MTU mismatch looks like, and why it is often found late.	5.26
185.	Diagnose an edge node that is not forwarding.	5.26

V. CONNECTIVITY AND ROUTING

Objective 5.27. The largest class of real support calls.

186.	Record the path a packet takes between two machines on the same segment.	5.27
187.	Record the path when they are on different segments behind one Tier-1.	5.27
188.	Record the path from a workload to the physical world.	5.27
189.	Break connectivity at the segment level and diagnose it.	5.27
190.	Break it at the Tier-1 level and diagnose it.	5.27
191.	Break the Tier-0 uplink or its routing and diagnose it.	5.27
192.	Diagnose a distributed firewall rule that is silently dropping traffic.	5.27
193.	Record how you would tell a firewall drop from a routing failure.	5.27
194.	Diagnose asymmetric routing, and record how it presents.	5.27
195.	Write the diagnostic order for two machines that cannot reach each other.	5.27

W. ECMP AND THE PACKET WALK

Objectives 5.28 and 5.29. Describe objectives, and simulations.

196.	Write what ECMP is and the problem it solves.	5.28
197.	Record what ECMP requires of the Tier-0 and of the physical network.	5.28
198.	Record how ECMP relates to high availability, and where they differ.	5.28
199.	Record what happens to an existing flow when a path is lost.	5.28
200.	Record what stateful services do to ECMP, and why that matters.	5.28
201.	Narrate the north-south packet walk out loud, in under two minutes, with no notes.	5.29
202.	Narrate the east-west packet walk between two segments the same way.	5.29
203.	Draw both walks and mark the point where a failure would be invisible to the guest.	5.29

X. AGAINST THE CLOCK

Do this section twice in the final week, with faults somebody else created.

204.	Diagnose an ESX host fault you did not cause. 15 minutes.	5.6
205.	Diagnose a virtual machine that will not power on. 10 minutes.	5.9
206.	Diagnose a vSAN object that is inaccessible. 20 minutes.	5.3
207.	Diagnose a datastore that will not mount on one host. 15 minutes.	5.4
208.	Diagnose a VKS cluster that will not provision. 25 minutes.	5.15
209.	Diagnose an image pull failure caused by trust. 15 minutes.	5.16
210.	Diagnose a cloud proxy that has stopped collecting. 15 minutes.	5.19
211.	Diagnose two machines that cannot reach each other across a Tier-1. 20 minutes.	5.27
212.	Find the first error of an incident in the logs. 15 minutes.	5.23

213. Narrate the north-south packet walk from memory. 5 minutes.

5.29

214. Produce one optimisation recommendation with evidence. 20 minutes.

5.10