Open Source
MANO

VNF ONBOARDING QUESTIONNAIRE

Version 1.0
September 30, 2016

This OSM VNF Onboarding questionnaire is for VNF creators who intend to onboard one or more VNFs to the OSM Platform. Answers to these questions are used to inform and guide the onboarding process.

NOTE: This document is based on “RIFT.io VNF Onboarding Questionnaire v3.3”, © RIFT.io, 2016, used with permission.

Contents

COMPANY INFORMATION

1. Company name
2. Company technical contact

   Email

   Phone

VIRTUAL NETWORK FUNCTION INFORMATION

3. VNF product name:
4. VNF software version:
5. Provide a high-level description of the function of the VNF:
6. How is the VNF typically used as part of a network service?
7. Provide instructions for how to download a copy of your software images and product admin / configuration manuals.

   Alternatively, attach a copy when you return this questionnaire to RIFT.io.
8. Describe how licensing works (if any)
9. Describe how to verify that the VNF is functioning properly.

   Provide a minimal sanity-test procedure and expected results.
10. Has the VNF been booted under OpenStack?
11. If yes, what version?
12. Has the VNF been booted under KVM?
13. If yes, what version?

14. Has the VNF been booted under VMware?
15. If yes, what version?
16. Has the VNF been booted on AWS?
17. If yes, any special considerations?
18. Has the VNF been booted under containers?
19. If yes, what version?
20. Does the VNF have a separate Management Interface from the Default Interface?
21. Do all of the VNF’s Management Interfaces have DHCP enabled?
22. If no, please explain:
23. Does the VNF support a Native Application dashboard GUI?
24. If yes, describe how to use/access the GUI (e.g., port # and credentials):
25. Does the VNF support SSH management?
26. If SSH support is enabled by default, what are the default credentials? Can the credentials be supplied via cloud-init?
27. If not enabled by default, how is support enabled and configured?
28. Does the VNF support HTTP/HTTPS management?
29. If HTTP/S support is enabled by default, what are the default credentials and port?
30. If HTTP/S not enabled by default, how is support enabled and configured?

31. Does the application have one or more existing JuJu Charms? If so, provide and overview of their capabilities.

32. If NETCONF support is enabled by default, what are the default credentials?

33. If not enabled by default, how is NETCONF support enabled and configured?

34. How can the OSM determine if the VNF is ready to accept configuration post resource orchestration? Is there a preferred configuration mechanism/protocol?

VIRTUAL MACHINE INFORMATION

35. How many VMs are in the VNF?

36. If more than one VM, are there startup ordering dependencies?

37. If there are startup ordering dependencies, explain the issues in detail:

38. How many physical hosts are required to support a single instance of this VNF?

39. If the number of hosts is variable due to scaling, provide a desired call model and number of hosts required to support the call model:

CONSTITUENT VIRTUAL MACHINES

For each constituent VM in the VNF, provide the following information.

40. VM name: ______________________

41. VM image file name: ______________________

42. VM software image format: (iso, qcow, qcow2): ______________________

43. Number of vCPUs required: ______________________

44. Amount of memory required in MB: ______________________

45. Amount of block disk space for primary VM disk image required in GB and required I/O-ops/sec: ______________________

46. Describe any additional block storage requirements for the constituent VM: ______________________

    a. Name of volume: ______________________

    b. Base file image name/type: ______________________

    c. Size in GB: ______________________

    d. Required IO-ops/second: ______________________

    e. Volume driver type: ISCSI, NFS, ZFS, FiberChannel: ______________________

    f. Any other requirements/assumptions: ______________________

47. Is a specific VM flavor required? If yes, what is the name, and what characteristics are associated with the VMs? ______________________

48. I/O requirements ______________________

    a. Number of interface: ______________________

    b. Type of each interface (PCI-passthrough, SR-IOV, Virt-io, e100 emulation, etc.): ______________________

    c. Are there any specific NIC assumptions? ______________________

49. Specific hardware dependencies: ______________________

50. Configuration requirements:
51. Is there support for cloud-init?
52. What configuration parameters are required?

    Provide example template in YAML format.
53. Does any configuration need to be applied post resource orchestration?
54. If yes, what configuration is required and when should it be applied? How are interface roles(LAN/WAN/MGMT) mapped to allocated devices?

CONSTITUENT VM MONITORING PARAMETERS

For each constituent VM in the VNF, provide the following information for each monitoring parameters accessible via HTTP/HTTPS to be shown in the Launchpad GUI:

55. Protocol and port number:
56. Username/password credentials:
57. Corresponding endpoint URL:
58. Display name:
59. Short description:
60. Access method:
    (e.g., JSONPATH, OBJECTPATH)
61. Access method parameters:
    (e.g., ‘json_path’ : ‘$.system.mem_total’)
62. Value-type of monitoring parameter:
    (e.g., string, integer, decimal)
63. Range of possible values:
64. Display units (if applicable):
    (e.g., MB, KB, %, etc.)
65. Display widget for monitoring parameter *(e.g., counter, gauge)*

66. Min/max frequency at which the parameter should be fetched: ______________________

____

ENHANCED PLATFORM AWARENESS (EPA) OPTIMIZATION PARAMETERS

67. For each constituent VM, note the EPA parameters to be used during instantiation. ______________________

Note: For guidance, refer to details in the following tables.

The tables below are based on OSM R1 Data model that will be enhanced with additional EPA parameter support in subsequent releases.

68. Guest EPA (vnfd:vdu:guest-epa)

ID	Type	Cardinality	Description
`trusted-execution`	Boolean	1	This VM should be allocated from trusted pool.
`mempage-size`	enum	0..1	Memory page allocation size. If a VM requires hugepages, it should choose LARGE or SIZE_2MB or SIZE_1GB. If the VM prefers hugepages it should chose PREFER_LARGE. ▪ LARGE: Require hugepages (either 2MB or 1GB) ▪ SMALL: Doesn't require hugepages ▪ SIZE_2MB: Requires 2MB hugepages ▪ SIZE_1GB: Requires 1GB hugepages ▪ PREFER_LARGE: Application prefers hugepages
`cpu-pinning-policy`	enum	0..1	CPU pinning policy describes association between virtual CPUs in guest and the physical CPUs in the host. ▪ DEDICATED: Virtual CPUs are pinned to physical CPUs ▪ SHARED: Multiple VMs may share the same physical CPUs. ▪ ANY : Any policy is acceptable for the VM

ID	Type	Cardinality	Description
`cpu-thread-pinning-policy`	enum	0..1	CPU thread pinning policy describes how to place the guest CPUs when the host supports hyper threads: ▪ AVOID: Avoids placing a guest on a host with threads. ▪ SEPARATE: Places vCPUs on separate cores, and avoids placing two vCPUs on two threads of same core. ▪ ISOLATE: Places each vCPU on a different core, and places no vCPUs from a different guest on the same core. ▪ PERFER: Attempts to place vCPUs on threads of the same core.
`pcie-device`	list	0..1	Information of PCIe acceleration devices. See 69. PCIe acceleration devices (vnfd:vdu:guest-epa:pcie-device).
`numa-policy`	choice	1	Specifies whether the VM is NUMA aware. When NUMA aware, the numa-node-policy container captures the details about the numa-node-policy. Choices are: node-cnt, mem-policy, node. See 70. NUMA node policy information (vnfd:vdu:guest-epa:numa-policy).

69. PCIE ACCELERATION DEVICES (VNFD:VDU:GUEST-EPA:PCIE-DEVICE)

ID	Type	Cardinality	Description
`device-id`	string	1	Device identifier
`count`	uint64	1	Number of devices to attach to the VM.

70. NUMA NODE POLICY INFORMATION (VNFD:VDU:GUEST-EPA:NUMA-POLICY)

ID	Type	Cardinality	Description
`node-cnt`	uint16	1	Number of NUMA nodes to expose to the VM.
`mem-policy`	enum	1	This policy specifies how the memory should be allocated in a multi-node scenario. ▪ STRICT: The memory must be allocated strictly from the memory attached to the NUMA node. ▪ PREFERRED: The memory should be allocated preferentially from the memory attached to the NUMA node
`node`	list	0..n	List of numa nodes. See 71. NUMA node information (vnfd:vdu:guest-epa:numa-policy:numa-nodes).

71. NUMA node information (vnfd:vdu:guest-epa:numa-policy:numa-nodes)

ID	Type	Cardinality	Description
id	uint64	1	NUMA node identification. Typically 0 or 1.
vpcu	uint64	1	Number of VPCUs to allocate on this NUMA node.
memory-mb	integer	1	Memory size expressed in MB for this NUMA node.
om-numa-type	choice	1	Openmano Numa type selection: CORES, PAIRED-THREADS, THREADS. See 72. OpenMANO NUMA type (vnfd:vdu:guest-epa:numa-policy:numa-nodes:om-numa-types).

72. OpenMANO NUMA type (vnfd:vdu:guest-epa:numa-policy:numa-nodes:om-numa-types)

ID	Type	Cardinality	Description
num-cores	uint8	1	Number of cores.
paired-threads	container	1	Container paired-threads. See 73. Container paired threads (vnfd:vdu:guest-epa:numa-policy:numa-nodes:om-numa-types:paired-threads).
threads	uint8	1	Number of threads.

73. Container paired threads (vnfd:vdu:guest-epa:numa-policy:numa-nodes:om-numa-types:paired-threads)

ID	Type	Cardinality	Description
paired-thread-ids	list	0..n	List of thread pairs to use in case of paired-thread numa. max-elements 16. ▪ thread-a (key, uint8) ▪ thread-b (uint8)

74. vSwitch EPA (vnfd:vdu:vswitch-epa)

ID	Type	Cardinality	Description
ovs-acceleration	enum	0..1	Specifies Open vSwitch acceleration mode. ▪ MANDATORY: OVS acceleration is required ▪ PREFERRED: OVS acceleration is preferred ▪ DISABLED: OVS acceleration is disabled.
ovs-offload	enum	0..1	Specifies Open vSwitch hardware offload mode. ▪ MANDATORY: OVS offload is required ▪ PREFERRED: OVS offload is preferred ▪ DISABLED: OVS offload is disabled

75. Hypervisor EPA (vnfd:vdu:hypervisor-epa)

ID	Type	Cardinality	Description
type	enum	0..1	Specifies the type of hypervisor. KVM: KVM XEN: XEN.
version	string	0..1	Version of the hypervisor.

76. Host EPA (vnfd:vdu:host-epa)

ID	Type	Cardinality	Description
cpu-model	enum	0..1	Host CPU model. Examples include: SandyBridge, IvyBridge. The following values are supported: PREFER_WESTMERE, REQUIRE_WESTMERE, PREFER_SANDYBRIDGE, REQUIRE_SANDYBRIDGE, PREFER_IVYBRIDGE, REQUIRE_IVYBRIDGE, PREFER_HASWELL, REQUIRE_HASWELL, PREFER_BROADWELL, REQUIRE_BROADWELL, PREFER_NEHALEM, REQUIRE_NEHALEM, PREFER_PENRYN, REQUIRE_PENRYN, PREFER_CONROE, REQUIRE_CONROE
cpu-arch	enum	0..1	Host CPU architecture. The following enums are supported: PREFER_X86, REQUIRE_X86, PREFER_X86_64, REQUIRE_X86_64, PREFER_I686, REQUIRE_I686, PREFER_IA64, REQUIRE_IA64, PREFER_ARMV7, REQUIRE_ARMV7, PREFER_ARMV8, REQUIRE_ARMV8.
cpu-vendor	enum	0..1	Host CPU vendor. The following enums are supported: PREFER_INTEL, REQUIRE_INTEL, PREFER_AMD, REQUIRE_AMD.
cpu-socket-count	enum	0..1	Number of sockets on the host. The following enums are supported: PREFER_ONE, PREFER_TWO, REQUIRE_ONE, REQUIRE_TWO.
cpu-core-count	uint64	0..1	Number of cores on the host.

ID	Type	Cardinality	Description
cpu-feature	enum	0..1	List of CPU features. The following enums are supported: PREFER_AES; REQUIRE_AES; PREFER_CAT; REQUIRE_CAT; PREFER_CMT; REQUIRE_CMT; PREFER_DDIO; REQUIRE_DDIO. ▪ AES: CPU supports advanced instruction set for AES (Advanced Encryption Standard). ▪ CAT: Cache Allocation Technology (CAT) allows an Operating System, Hypervisor, or similar system management agent to specify the amount of L3 cache (currently the last-level cache in most server and client platforms) space an application can fill (as a hint to hardware functionality, certain features such as power management may override CAT settings). ▪ CMT: Cache Monitoring Technology (CMT) allows an Operating System, Hypervisor, or similar system management agent to determine the usage of cache based on applications running on the platform. The implementation is directed at L3 cache monitoring (currently the last-level cache in most server and client platforms). ▪ DDIO: Intel Data Direct I/O (DDIO) enables Ethernet server NICs and controllers talk directly to the processor cache without a detour via system memory. This enumeration specifies if the VM requires a DDIO capable host. NOTE: Provide both require and prefer options for these features.
om-cpu-model-string	string	1	OpenMANO CPU model string
om-cpu-feature	string	1	OpenMANO CPU features

ADVANCED TOPICS

77. Does the VNF require its own specific VNFM?

    a. If yes, does the specific VNFM have a MANO aligned Or-Vnfm interface?

    b. What format is the Or-Vnfm interface (REST, TOSCA, etc.)?

    c. Which resource allocation model does the specific VNFM use – NFVO allocation or VNFM allocation?

78. Does the VNF support Service Function Chaining? Is it port-based or wireline-protocol (NSH) based?

79. How should VM failures be handled?

80. What auto-scaling support is desired?

81. What other VNF lifecycle events need to be handled?

82. Does the VNF require an Element Manager? If so, is there a Ve-Vnfm-em aligned interface?