Specifying HPA Capability Requirements using TOSCAbased VNF Descriptors

Some VNF suppliers make use of the underlying hardware platform capabilities in order to accelerate performance and optimize throughput of their VNF products. In order to ensure proper instantiation and operation of such VNFs, the VNF descriptor (VNFD) is used to describe VNF-specific hardware platform (HPA) capability requirements that will be matched against the capabilities of the underlying hardware infrastructure resources, in order to ensure that appropriate resources are used for instantiation.

Specifying HPA requirements

There are five different types of HPA capability requirements that can be specified in the VNFdescriptor - CPU, memory, storage, network and NUMA. Each requirement type is specified using a TOSCA hash map containing a list of capabilities and their corresponding desired configuration values. The names of HPA capabilities and the format of their configuration values are **managed outside of ONAP**, using the <u>HPA Capability Registry maintained by</u> <u>ETSI NFV ISG</u>, and are meant to be opaque from the TOSCA orchestrator and parser perspectives. The following table describes where within the VNFD the HPA requirement hash maps are defined and where within the VNFD they are used.

HPA Requirements	Attribute Name	Defined in	Used in
CPU	vdu_cpu_requirements	tosca.datatypes.nfv.VirtualCpu	tosca.nodes.nfv.VDU.Compute
Memory	vdu_memory_requirements	tosca.datatypes.nfv.VirtualMemory	tosca.nodes.nfv.VDU.Compute
NIC	network_interface_requirements	tosca.datatypes.nfv.VirtualNetworkInterfaceRequirements	tosca.nodes.nfv.VduCp
			tosca.nodes.nfv.VnfExtCp
Storage	vdu_storage requirements	map	tosca.nodes.nfv.VDU.VirtualStorage
NUMA	logical_node_requirements	tosca.datatypes.nfv.LogicalNodeData	tosca.nodes.nfv.VDU.Compute

HPA Requirement Key/Value Pair Entry Format

HPA capability requirements are specified using a set of TOSCA hash maps within the VNFD. Each entry in the hash map fully defines a single HPA capability requirement. Within a given hash map entry, the "key" portion of the entry contains the HPA capability name and the "value" portion contains structured data representing the the attributes of the corresponding configuration value.

The "key" portion of each array entry contains a string specifying a particular HPA capability name. The "value" portion of each array entry is a structured field containing information used to interpret the requested capability configuration. The capability values are defined as TOSCA strings, and their content can vary from a single attribute to complex permutations of simple attributes, lists and hash maps. Content and structure of capability values are defined using JSON schemas that can be used to parse the capability value strings in order to determine the desired state of HPA capabilities.

All capability value strings contain a set of pre-defined attributes that are used refine the how the requirements should be treated and what configuration value schema should be used for its interpretation. The following table describes that pre-defined attributes and their purpose:

Array Entry	Values	
<key></key>	A string specifying the name of a given HPA capability.	
<value></value>	A map of key-value pairs describing the configuration value of a given capability, where the key-value pairs are defined as follows	

key	type	value
schem aVersi on	string	A string describing the version the schema to be used for validation and interpretation of the <configuration-value>.</configuration-value>
schem aSele ctor	string	An Identifier for the particular schema to use.
hardw arePla tform	string	A string describing what hardware platform associated with a given HPA capability. For vendor-neutral HPA capabilities a value of "generic" is used.
mand atory	string	A boolean value expressed in string type specifying whether the requested capability is mandatory for for proper operation of the VNF. The value of "true" indicated that a given HPA capability must be present. The value of "false" indicates that the VNF can function with or without the capability.
config uratio nValue	string	A string describing the configuration value of a given capability, as defined by the schema.