VNF Deployment Attributes

Cisco Elastic Services Controller (ESC) is a Virtual Network Functions Manager (VNFM), performing lifecycle management of Virtual Network Functions (VNFs). ESC provides agentless and multi-vendor VNF management by

provisioning the virtual services and monitoring their health and load. ESC provides the flexibility to define rules for monitoring, and associate actions to be triggered based on the outcome of these rules. Based on the monitoring results, ESC performs scale in or scale out on the VNFs. It also supports automatic VM recovery when a VM fails.

The following sections below list the deployment (dep.xml) attributes.

Deployment Container

Attributes	Data Type	Description
tenant	list	List of tenants
name	string	Name of the tenant
managed_resource	boolean	If true, the tenant is created, used and deleted. If false, tenant is only used by ESC. The default is true
extensions	container	
extension	list	Tenant level associated extensions, e.g. quota
deployments (For Deploying VNFs without Service Registration)		
deployment	list	List of deployment
name	string	Name of the deployment
extensions	container	
extension	list	Deployment level associated extensions, e.g. VMWARE_VCD_PARAMS
locators	container	Contains VIM-specific resource locator properties for VMware multi VDC (support for single VIM deployments)
datacenter	string	Specifies the datacenter where the deployment will be done. Supported only in VMware
placement	list	Placement policy specification. Specifying it as a list allows us to define different placement policies among different combination of vm_groups
placement_group	list	Placement group policy specification. This policy will allow defining the placement policy and the VM group that will be part of this policy
policies	container	Describes different policies that can be specified that will affect the way VMs are brought up
policy	list	The policy list that contains a list of conditions and a list of actions

name	string	Specifies the unique name (within deployment) of the policy
conditions	container	The lifecycle stage conditions to trigger actions
condition	list	List of conditions on which this policy relies
name	string	Specifies condition on which this policy relies ESC provides a list of supported conditions
actions	container	The actions that will be triggered if conditions are satisfied
action	list	List of actions which this policy triggers
name	string	Specifies name of the action to be triggered. Some action names are pre-defined in ESC
type	Enum: • SCRIPT • PRE_DEFINED	Specifies the type of action
properties	container	Contains a list of name/value <property>
file_locators	container	List of file locators
file_locator	list	Each file locator is to specify the local or remote file that can be referenced in LCS, Day 0 or KPI Section. The remote_file can be used with authentication support using file server
local_file	container	File path on local file system
remote_file	container	Name of the File Server definition to which to connect
deployment_groups		
anti_affinity_group	string	Specifies the name of the anti-affinity group that this deployment pertains to. A deployment can pertain to zero or multiple anti_affinity_groups. Supported in Openstack only

VM Groups

vm_group		
vm_group	list	This section allows you to define properties such as number of interfaces, type of monitoring, monitoring frequency, type of events, scaling mechanism, elasticity properties, and so on for each VM in this group. This represents a type of VM. For example, if one needs two webservers in a deployment, only one VM instance is defined and number of instances is set to 2 in the scaling section. If there are two types of VMs, for example a webserver and a database server, then such a service will have two vm_groups: one for webserver and another for database server

name	string	Describes the name of the VM group
placement	container	Placement policy specification. Specifying it as a list allows us to define different placement policies among different combination of vm_groups
locator	container	Indicates multi VIM capable configuration with VIM explicitly defined
vim_id	string	Specifies the ESC defined id of the target VIM to which to apply the configuration
vim_project	string	Specifies the name of the target project to which to apply the configuration
vim_region	string	Specifies the name of the target region to which to apply the configuration (AWS Only)
vim_vdc	string	Specifies the name of the target virtual datacenter to which to apply the configuration (Cloud Director Only)
type	Enum: • affinity • anti_affinity • zone_host • vm_policy	• affinity: all instances on the same host • anti_affinity: all instances on different hosts • zone_host: specify a zone or host • vm_policy: specify a VM placement policy (Cloud Director 10.x+ only)
enforcement	Enum: • strict • loose	Whether the VIM hint must be enforced or should be tried
host	string	Host on which the VMs of group specified above should be deployed on
zone	string	Zone on which the VMs of group specified above should be deployed on
policy	string	The Cloud Director 10.x+ VM placement policy name or HREF that determines VM placement
bootup_time	integer	Time in seconds that the VM takes to perform a cold boot. ESC waits for bootup_time and in this time frame if VM does not come up due to any reason, ESC starts recovery timer
reboot_time	integer	Time in seconds that the VM takes to perform a normal reboot. If not specified, it will use bootup_time value
		Time in seconds for the VM to perform a normal warm reboot. ESC waits for recovery wait time and then starts the recovery as defined in the dep.xml (reboot, redeploy or reboot/Redeploy). Reboot and deploy actions may be performed three times (or as per config) before ESC decides if VM is deployed or not deployed
recovery_policy	container	Specifies the type of recovery policy

action_on_recovery	Enum: • REBOOT_THEN_REDEPLOY (default) • REBOOT_ONLY • REDEPLOY_ONLY	Specifies the type of recovery policy
recovery_type	Enum: • AUTO (default) • MANUAL	The type of recovery
max_retries	Integer (default=3)	The number of recovery attempts
image	string	Refers to the pre-existing image or template on the VIM. Image term is applicable to the OpenStack environment and Template is applicable to the VMware VIM
flavor	string	Refers to the pre-existing flavor on the VIM. This is applicable only on the OpenStack environment
vim_vm_name	string	User specified name for the VM on the VIM
software_version	string	Software version of the VM group. The software version along with life cycle stages will enable the VM or VNF to perform software upgrade
floating_ip	type:escname	Floating ip

volumes

If size and sizeunit is provided, ESC will create the volume else it will find the volume in the VIM with the given details

name	string	Specifies the display name
volid	string	Specifies the order in which the out-of- band volume is attached
bus	Enum: • ide • scsi • virtio	Specifies the bus type of the volume to be attached
type	string	(Optional) Specify the type to match the volume with the type provided by ESC
size	integer	(Optional) Size of the Volume
sizeunit	Enum: • MiB • GiB • TiB • PiB	Size units
boot_index	integer	Specify the boot order for bootable volumes

Interface (list)

Specifies number of interfaces and properties for each interface. The order of the interfaces specified here does not correspond to the order of the interfaces in the VM.

Interface list can be choice between creating a new interface or port or use an existing port. In case of new interface, ESC will create the interface while creating the VM or VNF. As part of this, user could configure the interface to be DHCP or static and other port configurations.

In case of existing port, user will be able to add the port to VM or VNF but cannot add additional configuration

nicid	integer	Logical ID for the interfaces. This is used later in the KPI section to link on which nic the monitoring should happen
vim_interface_name	string	User specified name for the interface on the VIM
floating_ip	type:escnetname	Whether floating ip needs to be associated or not or needs to assign a given floating ip
model	Enum: • e1000; • ne2k_pci • virtio (default) • i82551 • i82557b • i82559er • rtl8139 • pcnet	In case of virtual: e1000 or virtio. In case of passthrough: Model of the NIC. This will be specific to the data center. Datacenters may have NICs that support virtual functions from different vendors, like Intel, Cisco etc
mac_address	ietf-macaddress	Static MAC address for this interface
network	string	Network to which this interface needs to be attached

Choice: single_subnet_or_multiple

Choose between subnet/ip_address or addresses. The container addresses was added to support multiple subnet/ip_address combination. This is needed for a VNF to support dual stack interface or multiple IP configuration per interface

	Choice: single		
subnet		string	Subnet within the network to where the port needs to be created (name or UUID)
ip_address		ietf-ipaddress	Static IP address for this interface
	Choice: multiple		
addresses		container	Support dual stack interface or multiple IP configuration per interface
address		list	List of subnet and ip_address
address_id		uint16	Id for the address in address list

floating_ip	type:escnetname	Whether floating ip needs to be associated or not or needs to assign a given floating ip
subnet	string	Subnet name or uuid for allocating IP to this port
ip_address	ietf-ipaddress	Static IP address for this specific subnet

End Choice: single_subnet_or_multiple

type	Enum: • virtual (default) • passthru • direct • macvtap • access • trunk	Interface type; configures Single Root I/O Virtualization. Note: Setting the type of interface as direct configures SR-IOV
port_security_enabled	boolean	Whether the port security is enabled at port level
public_ip_address	ietf-ipaddress	Public IP address for this interface
security_groups	container	Containter for security group(s) set for this instance
security_group	string	IP filter rules that determine access control for the VM instance
allowed_address_pairs	container	The allowed address is allows one to specify arbitrary mac_address/ip_address(cidr) pairs that are allowed to pass through a port regardless of subnet
network	list	Network allowed on this interface
name	string	Network name or uuid
address	list	Allowed address on this interface
ip_address	ietf-ipaddress	Ip address or Subnet address for this network
netmask	ietf-ipaddress	Netmask for the subnet address
ip_prefix	string	Prefix length for subnet address, represented by integer. For IPv4 address range is 0 - 32. For IPv6 address range is 0 - 128
port	string	The interface refers to an existing port
pci_id	string	To specify the PCI device for the interface. For VMware vSphere only
Monitoring		
monitoring data	list	Specify the monitoring rules that will be used to configure the monitor module within ESC

event_name	string	A user defined event name. Corresponding event name should exist in the rules section. Monitor module informs the event generator when the event has to be triggered, e.g. VM_ALIVE
monitoring_agent	uri	It specifies the monitor for VNF, e.g. local MONA (default if omitted) or distributed MONA
nicid	integer	Interface that should be used to monitor the metric. This is used when proxy is not used. Ensure that interface with this ID is specified earlier
address_id	integer	Address within the interface that should be used to monitor the metric. This is used when proxy is not used and nicid interface has multiple addresses. Ensure that an address with this ID is specified earlier
metric_value	string	Threshold value that should be checked by monitor module
metric_cond	Enum • GT • LT • EQ • GE • LE	Supported conditional operators for the metric value
metric_type	integer	Supported metric types are INT8, UINT8, INT16, UINT16, INT32, UINT32, FLOAT, DOUBLE, STRING
metric_occurrences_true	integer	Number of successive polling cycles monitoring module finds the condition to be true before sending an event
metric_occurrences_false	integer	Number of successive polling cycles monitoring module finds the condition to be false before sending an event
metric_collector	container	This section provides information about the metrics that needs to be monitored and at what frequency should the monitoring happen
type	string	Type that monitor module should monitor. Example: ICMP Ping. These are the types that are supported by the monitoring module. List of all supported names is monitor module dependent and the reader is advised to refer to the documentation of the monitor module used in a specific implementation
		Frequency with which the metric should be polled by the monitor module
polling_unit	Enum: • minutes • seconds	Units of poll frequency in seconds or minutes
rules		
admin_rules	container	These are the rules that an administrator specifies when the service is registered. This action is taken for each and every deployment of the service

rule	list	Actions that should be taken by ESC or by some other module on behalf of ESC when an event is triggered by the event. Every rule will have a name and an action script associated with it. The action script is a URL from where ESC downloads the script and executes when and event corresponding
event_name	string	Corresponding event name must be present in the monitoring section
action	string	Action associated with the above event. Values are ALWAYS log, TRUE servicebooted.sh, FALSE recover auto healing. There is a specific format for this, and the description must be updated with more useful information
scaling		
scaling	container	Specifies how many instances of a particular type of VM needs to be instantiated and whether elastic scale in and scale out is required
min_active	integer	Describes the minimum number of VMs in the deployment. Irrespective of what the load is on these VMs, ESC ensures at least the minimum number of service VMs will always be running
max_active	integer	Describes the maximum number of active VMs to be activated by ESC. New VMs are activated when the load increases
elastic	boolean (default=true)	Request elastic scale-in and scale-out
static_ip_address_pool	string	Lists the IP addresses
extensions		
extension	container	VM group level extensions, applicable for vCenter/vSphere deployments only.
name	string	The extension name; currently must be: "vmware_vsphere_placement" to affect vCenter/vSphere deployments. All other values will be silently ignored.
properties	container	Specifies 1-N child properties against vCenter/vSphere deployments. Each child is a list of name/value <property>
name	string	Can be either: "folder" or "resource_pool" or both if specifying two property children.
value	string	The applicable value. For "folder", this is the folder name to place the VM under. It will be created if it does not exist, or used if it does exist. For "resource_pool", this is the resource pool to place the VM under and must pre-exist, else an error occurs.
config_data		

configuration	container	This enables to pass day-0 configuration data into the service VM. There are two ways: File, and inline data. In either case a CDROM is created with the contents of the configuration data and is attached to the VM. file_locator_name is newly added to refer to the file locator defined at deployment
config_type	string (default='CREATE_ISO')	Defines how ESC will process the configuration data. If value is not present, ESC will pass the files from configuration list without attempting any conversion
config_options	container	Defines the options required to create the ISO file
configuration	list	This represents a list of configuration files/templates. This being a list allows one to specify multiple day-0 configurations
Trunks (list) -- More than one trunk may be defined under trunks		
name	string	Name of the trunk. This will be the name of the trunk in Openstack
parent_port	string	Name of the parent port
parent_nicid	string	This specifies the NIC ID as defined under <interfaces>, in this example, 0. The port created for this NIC will be used as the parent port for the trunk.
subports	container	Wrapper for one or more subport elements
subport	list	Wraps the subport definition. At least one subport must be defined
name	string	Name of the subport. This will be the name of the subport in Openstack
port_id	string	ID of this subport in VIM
network	string	Name or ID of the Openstack network to associate with this port. May be an external or ephemeral network defined for the deployment

segmentation_type	string	Defaults to vlan. See Openstack API documentation for other possible values Only vlan is currently supported as a segmentation-type
segmentation_id	unit16	The VLAN ID to assign to this subport

Ephemeral Networks

Network		
name	string	Name of the network
shared	boolean (default=true)	True if the network is shared among other tenants
admin_state	boolean	Specifies whether the admin state is up or down. Set to true for up and false for down. If down, the network does not forward packets
router_external	boolean	Indicates whether this network is externally accessible
provider_physical_net work	string	Specifies the name of the physical network over which the virtual network is implemented
provider_network_type	string	Specifies the physical mechanism by which the Virtual network is implemented
provider_segmentation _id	int	ID or tunnel ID of the underlying physical network
vlan_id	int	(VMWare only) Specifies the VLAN id to assign to this port group
number_of_ports	int	(VMWare only) Specifies the number of ports to allocate on the port group
switch_name	string	(VMWare only) Specifies the name of the switch
locators	container	Contains VIM-specific resource locator properties for VMware multi VDC (support for single VIM deployments)
multi_vim_locators	container	Locate the vim in a multi vim environment.
vim_network_name	string	Specifies the name of the network to be created on the VIM and must have a VIM id associated with it
subnet		
name	string	Name of the subnet
ipversion	string	IP Version – IPv4 or IPv6
dhcp	boolean	Are IP address for the VMs on this network are to be allocated by DHCP

address	etf-inet-types:ip-address	Subnet address for this network
netmask	etf-inet-types:ip-address	Subnet mask represented by IP address
ip_prefix	string	Subnet mask represented by IP prefix
gateway	etf-inet-types:ip-address	Default GW for the network. If unspecified ESC will try to determine the gateway for the
no_gateway	Boolean	Ignore the gateway setting

Images

The table below lists the Image (image.xml) attributes.

Image		
name	string	Name of the image
src	string	Indicates to ESC the source of the image. It could be either a URL from which ESC will download the image (http://...), or a file location path on the ESC VM itself (file://...)
disk_format	Enum: • qcow2 • raw • vmdk	Describes the format of the disk
container_format	string	The container format, e.g. 'bare'
serial_console	boolean	Set to true if the image has serial console
disk_bus	Enum: • ide • scsi • virtio	Root disk bus
visibility	string (default='public')	Specifies whether image should be created as 'public' or 'private'
checksum	string	SHA checksum for the file
checksum_algorithm	Enum: • sha256sum • sha512sum	Algorithm to use to validate the provided checksum
remove_src_on_completion	boolean (default=false)	true if image source file should be deleted after image registration is complete
locators	container	Contains VIM-specific resource locator properties
datacenter	string	Specifies the datacenter where the image will be created. Supported only in VMWare
locator	container	Indicates multi VIM capable configuration with VIM explicitly defined
vim_id	string	Specifies the ESC defined id of the target VIM to which to apply the configuration

vim_project	string	Specifies the name of the target project to which to apply the configuration
vim_region	string	Specifies the name of the target region to which to apply the configuration (AWS Only)
vim_vdc	string	Specifies the name of the target virtual datacenter to which to apply the configuration (Cloud Director Only)
metadata	container	Metadata properties are described in this section
src	ietf-inet-types:uri	URL of the metadata file from where it can be downloaded by ESC
checksum	string	SHA checksum for the file
checksum_algorithm	string	Algorithm to use to validate the provided checksum
properties	container	An optional properties list can be included in the image creation payload. This is only supported for creation of images on Openstack. Image properties will be stored in Openstack as image metadata. Property name and values are not validated by ESC

Flavor

The table below lists the Flavor (flavor.xml) attributes.

Flavor		
name	string	Name of the flavor.
vcpus	integer	Number of virtual CPUs per VM instance.
memory_mb	integer	Amount of memory in Mega Bytes per VM instance
root_disk_mb	integer	Virtual root disk size in gigabytes. This is an ephemeral disk the base image is copied into
ephemeral_disk_mb	integer	Specifies the size of a secondary ephemeral data disk. This is an empty, unformatted disk
swap_disk_mb	integer	Optional swap space allocation for the instance
properties	container	List of property name/value pairs
name	string	Name/key of property
value	string	Specifies the value of a property
encrypted_value	tailf:aes-cfb-128-encrypted-string	Specifies the encrypted value of a property

Ephemeral Volumes

The table below lists the Volume attributes.

Volume		
name	string	Name of the Volume
size	integer	Size of the Volume
sizeunit	Enum: • MiB • GiB • TiB • PiB	Size units
image	string	Name or UUID of the source image
type	string	Allows to provide scheduling to a specific back-end, and also can be used to specify specific information for a back-end storage device upon which to act upon

Routers

The table below lists the router attributes.

Router		
routers	container	
router	list	
name	string	Name of the Router to be created
description	string	Description of the Router
admin_state	boolean	Specifies whether the admin state is up or down. Set to true for up and false for down.
external_network	string	External gateway of the router
snat_enable	boolean	SNAT enabled or not
distribution	boolean	distributed : true indicates a distributed router
interfaces	container	
interface	list	
subnet	type: escroutername	Subnet name of the interface
port_id	type: escroutername	port id of the interface
static_routes	container	
route	list	
route_name	string	RouteName for identifying routes
destination	ipv4-cidr	Destination CIDR added for the static-routes
next_hop	escipaddr	next_hop added for the static-routes

opdata_routers

The table below lists the router attributes.

opdata_router		
routers	container	
router	list	
name	string	Name of the Router to be created
router_id	ietf-yang-types:uuid	ID of the router
description	string	Description of the Router
admin_state	boolean	Specifies whether the admin state is up or down. Set to true for up and false for down.
external_network	string	External gateway of the router
snat_enable	boolean	SNAT enabled or not
distributed	boolean	distributed : true indicates a distributed router
interfaces	container	
interface	list	
subnet	type: escroutername	Subnet name of the interface
port_id	type: escroutername	port id of the interface
static_routes	container	
route	list	
route_id	string	Internal route id of the route
destination	ipv4-cidr	Destination CIDR added for the static-routes
next_hop	escipaddr	next_hop added for the static-routes