

Cloud Platform Validation Spec

Notes for contributors:

- ***This doc is under development and open for all inputs.***
- This docs tries to define and cover all checks that can be run on a cloud platform to validate its state/health.
- Validation checks targets to ensures that all cloud software components are healthy/configured as described in PDF as desired cloud state.
- All validation checks combined can be considered as a *kind of ping utility for a cloud platform*.
- Functional tests or Performance Benchmarks are out of scope for this list.
- SDV(Sub Project of CIRV) implements or will implement all checks defined by this doc.

Table of Contents

- [Table of Contents](#)
- [RA1 validation checks](#)
- [RA2 validation checks](#)

RA1 validation checks

Currently defined Cloud Deployment Types:

- **OOK** - OpenStack on Kubernetes
- **OOO** - OpenStack on OpenStack
- **OAC** - OpenStack as Containers (without Kubernetes)
- **OAV** - OpenStack as VMs

Test Suites are used to group multiple checks.

Sl. No.	Test Suite	Test /Check Name	Cloud Deployment Types	Description/Details	Notes/References
1.	platform	pod_health_check	OOK, OAC	Checks health of all overcloud components running as pod in Kubernetes cluster(in OOK deployment). Pass: All components are healthy. Fail: One or more components are unhealthy.	
2.	storage	ceph_health_check	OOK, OAC, OOO, OAV	Checks health of all components of ceph cluster configured for OpenStack.	
3.	observability	prometheus_check	OOK	Check health endpoints(https: "/-/healthy") and readiness endpoint("/-/ready") of prometheus. Pass: On pass of both healthy and ready check. Fail: If readiness of healthy not true.	
4.	observability	prometheus_alert_manager_check	OOK	Check whether Alert Manager is ok by sending https request to "/-/healthy" and "/-/ready" endpoints of the alert manager.	
5.	observability	grafana_check	OOK	Checks whether Grafana is healthy by sending request at /api/health endpoint.	
6.	observability	elasticsearch_check	OOK	Checks health of elasticsearch cluster by sending https request at "/_cluster/health" endpoint of the Elasticsearch cluster.	
7.	observability	kibana_check	OOK	Kibana Dashboard health check, checks health using status at "/api/status" endpoint.	
8.	observability	nagios_check	OOK	Check whether Nagios api is reachable and gives https_OK	
9.	observability	elasticsearch_exporter_check	OOK	Check whether elasticsearch exporter is exporting prometheus metrics at "/metrics"	

10.	observability	fluentd_exporter_check	OOK	Check whether fluentd exporter is exporting prometheus metrics at "/metrics"	
11.	network	physical_network_check	OOK, OAC, OOO, OAV	Checks network mappings in ml2.conf against PDF.	
12.	compute	reserved_vnf_cores_check	OOK, OAC, OOO, OAV	Checks vcpu_pin_set configurations in nova against the required PDF value for reserved cores.	
13.	compute	isolated_cores_check	OOK, OAC, OOO, OAV	checks isolcpus configuration against required value in PDF.	
14.	network	vswitch_pmd_cores_check	OOK, OAC, OOO, OAV	Evaluates pmd-cpu-mask in vswitch against required cores in PDF.	
15.	network	vswitch_dpdk_lcores_check	OOK, OAC, OOO, OAV	Evaluates dpdk-lcore-mask in vswitch against required cores in PDF.	
16.	compute	os_reserved_cores_check	OOK, OAC, OOO, OAV	Calculates os_reserved_cores using formula: os_reserved_cores = all_cores - (reserved_vnf_cores + vswitch_pmd_cores + vswitch_dpdk_lcores) and compares against required os_reserved cores in PDF.	
17.	compute	nova_scheduler_filters_check	OOK, OAC, OOO, OAV		
18.	compute	cpu_allocation_ratio_check	OOK, OAC, OOO, OAV		
19.	platform	api_version_check	OOK, OAC, OOO, OAV		
20.	network	mtu_check	OOK, OAC, OOO, OAV		
21.	platform	ntp_check	OOK, OAC, OOO, OAV		
22.	network	sriov_vfs_check	OOK, OAC, OOO, OAV		
23.	security	pod_linux_capabilities_allowed_check	OOK		
24.	security	privileged_pod_allowed_check	OOK		
25.	security	pod_host_volume_mount_check	OOK		
26.	security	pod_host_network_check	OOK		
27.	security	mgmt_api_access_check	OOK		
28.	compute	cpu_manager_policy_check	OOK	Checks whether the actual cpu-manager-policy in the config of Kubelet on each node is same as the desired cpu-manager-policy defined in PDF. Also, if the policy is set to static then check further that desired cpu-manager-reconcile-period and full-pcpus-only in PDF matches the actual configurations in Kubelet.	For better CPU affinity Telcos may be interested to configure cpu-manager-policy as static with required options. https://kubernetes.io/docs/tasks/administer-cluster/cpu-management-policies/
29.	compute	topology_manager_policy_check	OOK	Check-a: Checks whether topology-manager-scope(pod or container) set on kubelets on all nodes are same as required by PDF. Check-b: Checks whether topology-manager-policy(no node, best-efforts, restricted or single-numa-node) set on kubelets on all nodes are same as required by PDF. This check is considered passed if both Check-a and Check-b pass.	To best utilize NUMA architecture for better performance Telco may be interested to configure topology-manager-policy in kubelets. https://kubernetes.io/docs/tasks/administer-cluster/topology-manager/

30.	network	cni_check	OOK		
31.	platform	device_plugin_check	OOK		
32.		service_mesh_check	OOK		
33.		ingress_egress_check	OOK		
34.	platform	kubevirt_check	OOK		
35.		helm_check	OOK		
36.	platform	readiness_probe_check	OOK	Checks whether the readiness probe is configured for all overcloud components deployed as pods on undercloud Kubernetes.	For undercloud K8s to ensure that overcloud is ready all overcloud component should define the required readiness check for them. Note: this check only checks whether the probes are defined rather than checking the readiness of each pod, as it is already covered by <code>pod_health_check</code>
37.	platform	startup_probe_check	OOK		
38.	platform	liveliness_probe_check	OOK	Checks whether the liveness probe is configured for all overcloud components deployed as pods on undercloud Kubernetes.	https://kubernetes.io/docs/tasks/configure-pod-container/configure-liveness-readiness-startup-probes/ Note: this check only checks whether the probes are defined rather than checking the liveness of each pod, as it is already covered by <code>pod_health_check</code>

RA2 validation checks

Sl. No.	Test suite	Test/Check Name	Cloud Deployment Types	Description/Details	Notes /References
1.	security	network_policy_check		Checks whether default policy is used to deny all ingress and egress traffic, & unselected pods are isolated	
2.	security	encryption_check		Checks whether external key management systems are in use for encryption of secrets	
3.	security	access_control_check		Checks whether role based access control (RBAC) is enabled	