Software Delivery Validation(SDV)

Overview

This will be the 'software' counterpart of this work. The validation of the software configuration, should be performed against the requirements, that is preferably defined in a machine-readable format. Hence, this project relies heavily on an implementation of PDF 2.0 defined in CNTT.

Team

Name	Affiliation	Remarks
Sridhar K. N. Rao Sridhar Rao	Spirent	CIRV-SDV Lead
Parth Yadav Parth Yadav	Delhi University	LFN Intern
Ashwin Nayak	NITK	LFN Intern
Shivam Balikondwar Shivam Balikondwar	PICT	Student Volunteer

Implementation of PDF 2.0 in JSON

This implementation will be "SPECIFIC" to support software validation, and multiple installers/installations (Airship, TripleO, Fuel, etc). In this implementation, there will be parameters that are not part of CNTT-PDF. The implementation has two parts:

1. PDF-Template in JSON.
2. Site-specific PDF in JSON

The PDF-Template includes "extrapolation" information, which will help to create site-specific pdfs. The template will be filled by the user and extrapolated by the tool to create complete site-specific PDF.

The below figure summarizes the organization of the template.

blocked URL

Architecture

There will be two versions with single code-base.

Version-1

The below figure provides the software architecture of the SDV. The user only runs 'valid' program, and select one or more validations.

blocked URL

Version-2

The containerized version.

blocked URL

Testcases

The initial testcases can be found here: http://testresults.opnfv.org/test/#/projects/cirv

Pre-Deployment Software Validation: Hyperlinks

The below table and excel-sheet summarizes the links for which the connectivity is to be checked. Along with the connectivity, the 'Latency' can also be noted.

PS: The table (sources) and the excel (images) are specific to Airship. For other deployments, the components and corresponding repo-names and tags may vary.

Images

Airship-Images.xlsx

Sources

Sl. No.	Component	Repository-Name	Tags	URL to use
1	Neutron	openstack-helm/neutron	d2abe39d498f48c4721e26aca19e81189bc8891b	https://opendev.org/openstack/openstack-helm
2	helm-toolkit	openstack-helm-infra/helm-toolkit	03580ec90afa162c166661acf27f351b83565375	https://opendev.org/openstack/openstack-helm-infra
3	nova	openstack-helm/nova	d2abe39d498f48c4721e26aca19e81189bc8891b	https://opendev.org/openstack/openstack-helm
4	openvswitch	openstack-helm-infra/openvswitch	d0b32ed88ad652d9c2226466a13bac8b28038399	https://opendev.org/openstack/openstack-helm-infra
5	calico, libvirt, mariadb, memcached, rabbitmq, postgresql, ceph-client, ceph-mon, ceph-osd, ceph-provisioners,	openstack-helm-infra/< >	03580ec90afa162c166661acf27f351b83565375	https://opendev.org/openstack/openstack-helm-infra
6	apiserver, calico/etcd, controller-manager, coredns, etcd, haproxy, ingress, proxy, scheduler, promenade	charts/< >	64807416b71958e31156ef7a50e169813acc4e15	https://opendev.org/airship/promenade
7	barbican, cinder, glance, heat, horizon, keystone, tempest	openstack-helm-infra/< >	52c132b9356695bf455ae25ec78cef9f532f9fe8	https://opendev.org/openstack/openstack-helm
8	tiller	charts/tiller	50384e47c762438b9e39abe4677f3c29f3c09184	https://opendev.org/airship/armada
9	calico-utility, ceph-utility, compute-utility, etcd-utility, mysqlclient-utility, openstack-utility, postgresql-utility	charts/<>	9c2038cb59bfbb3ff5c3bbf11c7001d621437b98	https://opendev.org/airship/porthole

Pre-Deployment Software Validation: Configuration

Apart from ensuring that the requirements are met, this validation helps in minimizing/eliminating any deployment errors, drives test-automation, and checks for consistency to achieve efficient automation. The below picture summarizes the scope (in red dashed rectangle) of the software validation.

blocked URL

This software validation can be: (a) 'Pre-Deployment' Software Validation (b) NW Links Validation (c) Post-Deployment Software Validation

The below table summarizes validation of different configurations. The table uses Airship and its configurations as an example.

Sl-No	Validation-Parameter	expected value	File To look at (Applies to AIRSHIP ONLY)	"Key" to match (Applies to AIRSHIP ONLY)	Requirement Level
1	OS	ubuntu xenial	gobal/software/charts/ucp/drydock/maas.yaml type/cntt/profiles/host/*	default_os: image:	Mandatory
2	HugePage Configuration	size: 1G count: 32 (minimum)	profiles: type/cntt/profiles/hardware/* site/<site_name>/profiles/hardware/*	hugepages: dpdk: size: count:	Mandatory
3	CPU Isolation	Ex: 4-43, 48-87	same as above	cpuset: kvm: vcpu_pin_set	Mandatory
4	Openstack Version	ocata/pike/stein	No Single Explicit Location. global/software/config/versions.yaml includes the mention of the version	osh: <service_name>: <service_component>: *ocata*	Mandatory
5	Openstack Services ?	ingress-controller, ceph-config,, mariadb, rabbitmq, memcached, keystone, radosgw, glance, cinder, compute-kit, heat, horizon	type/cntt/software/manifests/full-site.yaml global/software/manifests/full-site.yaml		Mandatory ?
6	Virtual-Switch	openvswitch openvswitch-dpdk	Config only - Version as supported by OSH site/<podname>/software/charts/osh/openstack-compute-kit/chart-group.yaml	chart_group: - openvswitch-dpdk - neutron-ovsdpdk	Mandatory
7	Version of the Manifests	1.7	site/<site-name>/site-definitions.yaml	revision:	Mandatory
8	Node-Names ?	list of all the names.	site/<site_name>/baremetal/nodes.yaml	name:	Mandatory ?
10	bootstrap protocol ?	pxe	profiles: type/cntt/profiles/hardware/* site/<site_name>/profiles/hardware/*	bootstrap_protocol:	Mandatory ?

Sl-No	Validation-Parameter	expected value	File To look at (Applies to AIRSHIP ONLY)	"Key" to match (Applies to AIRSHIP ONLY)	Requirement Level
1	Hypervisor	kvm	global/software/charts/osh/openstack-compute-kit/nova.yaml	virt_type:	Optional
2	Container Engine	docker	global/software/config/versions.yaml	repositories: docker:	Optional
3	Container Management	kubernetes	NA	NA	Optional
4	k8S components	proxy, container-networking, dns, etcd, haproxy, core	type/cntt/software/manifests/full-site.yaml global/software/manifests/full-site.yaml		Optional
5	Ceph Components	mds, mon, osd, rgw, mgr	type/cntt/profiles/genesis.yaml ---------- type/cntt/software/config/endpoints.yaml	labels: dynamic: <ceph-*>: enabled ---------- data: <tenant-ceph-*>:	Optional
6	K8S Networking	calico	No single explicit location.		Optional
7	LMA Components - Client Side	promethues-openstack-exporter fluentbit	type/cntt/software/manifests/full-site.yaml global/software/manifests/full-site.yaml		Optional
8	LMA Components - Server Side	infra-logging, infra-monitoring, infra-dashboards	type/cntt/software/manifests/full-site.yaml global/software/manifests/full-site.yaml		Optional
9	Special NIC Drivers	ixgbe or i40e	baremetal/bootactions/*.yaml	The complete file.	Optional
10	Users				Optional
11	UCP Components	ceph-update, ceph-config, core, keystone, divingbell, armada, deckhand, drydock-scaled, promenade, shipyard,	type/cntt/software/manifests/full-site.yaml global/software/manifests/full-site.yaml		Optional
12	Tenant Ceph Components	tenant-ceph (mgr, mon, rgw)	type/cntt/profiles/genesis.yaml ---------- type/cntt/software/config/endpoints.yaml	labels: dynamic: <tenant-ceph-*>: enabled ---------- data: <tenant-ceph-*>:	Optional
13	Post-Dep N/W Config Script				Optional
14	Site-Type	cntt	site/<site-name>/site-definitions.yaml	site_type:	Optional
15	Version of the Manifests	1.7	site/<site-name>/site-definitions.yaml	revision:	Optional
16	OVS DPDk Configuration	Non-Datapath Threads, PMD Threads OVS-Bridge name	profiles: type/cntt/profiles/hardware/* site/<site_name>/profiles/hardware/* ---------- type/cntt/network/common-addresses-ovsdpdk.yaml	cpu_sets: dpdk-lcore-mask: pmd-cpu-mask: ---------- bridge_for_ovsdpdk	Optional

Network-Links Validation

Some of the options we have:

1. Run LLDPtool on compute and control nodes – easiest to implement, but, it can get complex if the number of nodes are more.
2. LLDP on TORs - Reuse the solutions that used by ‘data centers’ – opensource and scalable. It may take slightly more time.

We propose to use Option-1. The first problem is topology representation. If PDF 2.0 does not include this in its description, we will use DOT-Specified network graph .

Post-Deployment Software Validation : State

Sl. No.	Validation Type	Description
1	Container or POD Status	Ensure container status is OK. Detect failed containers and raise an error
2	Required Container ports are open	Ensure relevant docker containers are up and running, with ports open to listen
3	ntp	Verify all deployed nodes have their clock synchronized
4	OVS-DPDK Configuration	Validates OVS DPDK PMD cores from all NUMA nodes
5	RabbitMQ Limits	Make sure the rabbitmq file descriptor limits are set to reasonable values
6	Service Status	Detect services status on the target host and fails if we find a failed service.
7	SELinux	Ensure we don't have any SELinux denials on the system
8	TLS configuration	Ensure endpoints are secured.
9	Storage (Ceph) Health Check	Health should be OK

Post-Deployment Software Validation: Security

Sl. No.	Validation Type	Description
1	Security: Right File Permissions	https://docs.openstack.org/security-guide/checklist.html
2	Security: Right Configurations	https://docs.openstack.org/security-guide/checklist.html

Readouts

SDVState readouts

Readout	Author	Slides	Notes
SDVState announcement at LFN Mentorship Project Presentation	Parth Yadav	? Unknown Attachment	