Support for K8S (Kubernetes) based Cloud regions

Kubernetes based Cloud-region support

Code Name: K8S

This page tracks the related materials/discussions/etc related to cloud regions that are controlled by Kubernetes.

As starting point, this effort has started as small subgroup of multicloud as task force.As the efforts evolve, logistics would be revised. Maybe this task force would be promoted to a independent group or an independent project.

Meetings

- #coe Team ONAP7, Friday UTC 3:00pm from Jan 16/17, 2018. weekly meeting

Slides/links

- Slides presented to architecture subcommittee to start this project as PoC : K8S_for_VNFs_And_ONAP_Support.pptx

Version 2 of slides are here:

K8S_for_VNFs_And_ONAP_Support_v2.pptx

K8s Plugin progress slide:

Kubernetes.pptx

Project (This is sub project of Multi-Cloud as decided by Architecture subcommittee & Multi-Cloud team)

Project name: K8S based cloud-region support

code name: K8S

Repository name: multicloud/k8s

Background:

As of ONAP R2, ONAP can only orchestrate VNFs across cloud regions that have Openstack and its variations as VIM. Supported Multi-Cloud plugins include VMWare VIO, Windriver Titanium, Openstack Newton and Openstack Ocata. VMWare VIO and Windriver Titanium are vendor supported Openstack projects with additional features.

As of ONAP R2, ONAP can only deploy VM workload types.

Need for containers are felt for following reasons

- Resource constrained Edges : Power and space constraints limit number of compute nodes that can be deployed in edges.
- Ubiquity of container support : Many cloud providers are now supporting containerized workload using Kubernetes.
- Micro Service Architecture based solutions: Containers are considered to be ideal solution to implement micro-services.

Kubernetes is one of the popular container orchestrators. K8S is supported by GCE, AZURE and AWS and will be supported by Akraino Edge stack that enable edge clouds.

K8S is also being enhanced to support VM workload types too. This helps cloud-regions that need to support VMs while moving some workloads to containers. For security reason, cloud-regions may continue to support VM types for security reasons and hence there is need for VIM that support both VMs and containers.

Since same K8S instance can orchestrate both VM and container workload types, same compute nodes can be leveraged for both VMs and containers.

Telco and CSPs are seeing similar need to deploy networking applications as containers

There are few differences between containers for Enterprise applications and networking applications. Networking applications are of three types - Management applications, Control plane applications and data plane applications. Management and control plane applications are similar to Enterprise applications, but data plane applications different in following aspects:

- Multiple virtual network interfaces
- Multiple IP addresses
- SRIOV networking support
- Programmable virtual switch (for service function chaining, to tap the traffic for visibility etc..)

Since, both VMs and container workloads are used for networking applications, there would be need for

- Sharing the networks across VMs and containers.
- Sharing the volumes across VMs and containers.

Proposal 1 & Proposal 2 feedback from community

Please see slides attached in slides/links section on detailed information on Proposal 1 and Proposal 2.

In summary:

Proposal 1: All the orchestration information is represented as TOSCA based service templates. There is no cloud technology specific artifacts. All the information about VNF, VDU, VL and Volumes are represented as per ETSI SOL specifications. In this proposal, ONAP with the help of Multi-Cloud translates the TOSCA representation of VNF/VDU/VL/Volume to Cloud technology specific API information before issuing cloud technology specific API calls.

Proposal 2: In this proposal, service orchestration information is represented in TOSCA grammar, but majority of VNFD/VDU/VL/Volume information is represented in cloud-technology specific artifacts. These artifacts are part of the VNF portion of CSAR. Since, cloud region selected is unknown during the VNF onboarding, multiple cloud-technology specific artifacts would need to be part of CSAR. Right artifact is chosen at run time by ONAP. Cloud technology specific artifacts are : HOT in case of Openstack, ARM in case of Azure, K8S deployment information in case of K8S etc ..

Though Proposal 1 is ideal, for pragmatic reasons Proposal 2 is chosen in this POC. Few reasons are given here:

- TOSCA way of representing Kubernetes deployment is not popular today. Many container vendors are comfortable in providing K8S deployment yaml files or Helm charts.
- Current TOSCA standards don't expose all the capabilities of K8S. TOSCA based container standards can take lot of time in standardizing and agreed upon by the community. But, there is a need now to support K8S based cloud regions.
- K8S community is very active and features supported by various K8S providers can differ from one another. VNF vendors are aware of that and create VNF images that take advantage of those features. It is felt that ONAP should not hinder these deployments.

Project Description:

The effort will investigate/drive a way to allow ONAP to deploy/manage Network Services/VNFs over cloud regions that support K8S orchestrator.

This project enhances ONAP to support VNFs as containers in addition to VNFs as VMs.

This project is treated as POC project for Casablanca.

Though it may identify the changes required in VNF package, SDC, OOF, SO and Modeling, it will not try to enhance these projects as part of this effort.

This project is described using API capabilities and its functionality.

- Create a Multi-Cloud plugin service that interacts with Cloud regions supporting K8S
  - VNF Bring up:
    - API: Exposes API to upper layers in ONAP to bring up VNF.
      - Currently Proposal 2 (Please see the attached presentation and referenced in Slides/Links section) seems to be the choice.
      - Information expected by this plugin :
        - K8S deployment information (in the form understood by K8S), which is opaque to rest of ONAP. This information is normally expected to be provided as part of VNF onboarding in CSAR) and some information (variables values) are created part of every instantiation.
          - TBD - Is this artifact passed to Multi-Cloud as reference or is it going to be passed as immediate data from the upper layers of ONAP.
          - Note that there could be multiple artifacts for a given deployment. For example, EdgeXFoundry requires multiple deployment yaml files (one for each type of container) and multiple service yaml files. Since, there could be many of them, ordering in which they get executed is important. Hence, it is required that the priority order is given in a separate yaml artifact, which is interpreted only by K8S plugin.
          - Supported K8S yaml artifacts - Deployment (POD, Daemonset, Stateful set)(, Service, Persistent volume). Others for for future releases.
          - Kubernetes templates (artifacts) and variables:
            - Since many instances of deployments can be instantiated using same CSAR, it is necessary that they are brought in different namespaces. Namespace name is expected to be part of variables (which is different for different instances).
            - And also any template information starting with $.... is replaced from the variables.
        - Metadata information collected by upper layers of ONAP
          - Cloud region ID
          - Set of compute profiles (One for each VDU within the VNF).
          - TBD - Is there anything to be passed
    - Functionality:
      - Instantiate VNFs that only consist of VMs.
      - Instantiate VNFs that only consist of containers
      - Instantiate multiple VNFs (some VNFs realized as VMs and some VNFs realized as containers) that communicate with each other on same networks (External Connection Points of various VNFs could be on the same network)
      - Reference to newly brought up VNF is stored in A&AI (Which is needed when VNF needs to be brought down, modify the deployment)
      - TBD - Should it populate A&AI with reference to each VM & Container instance of VNF? Or one reference to entire VNF instance good enough? Assuming that there is a need for storing reference to each VM/Container instance in A&AI, some exploration is required to see whether this information is made available by K8S API or should it watch for the events from K8S?
      - TBD - Is there any other information that this plugin is expected to populate A&AI DB (IP address of each VM /Container instance) and anything else?
  - VNF Bring down:
    - API: Exposes API function to upper layers in ONAP to terminate VNF
    - Functionality: Based on the request coming from upper ONAP layer, it will terminate the VNF that was created earlier.
  - Scaling within VNF:
    - It leaves the decision of scaling-out and scaling-in of services of the VNF to the K8S controller at the cloud-region.
    - (TBD) - How the configuration life cycle management be taken care?
      - Should the plugin watch for new replicas being created by K8S and inform APPC, which in turn sends the configuration?
      - Or should we let the new instance that is being brought up talk to APP-C or anything else and let it get the latest configuration?
  - Healing & Workload Movement (Not part of Casablanca)
    - No API is expected as it is assumed that K8S master at the cloud region will take care of this.
    - TBD - Is there any information to be populated in the A&AI when some healing or workload movement occurs that the cloud-region.
  - VNF scaling: (Not part of Casablanca)
    - API : Scaling of entire VNF
      - Similar to VNF bring up.
  - Create Virtual Link:
    - API : Exposes API to create virtual link
      - Meta data
      - Opaque information (Since OVN + SRIOV are chosen, opaque information passed to it is amenable to create networks and subnets as per the OVN/SRIOV Controller capabilities)
      - Reference to the newly created network is added to the A&AI.
      - If network already exists, it is expected that use count is incremented.
    - Functionality:
      - Creates network if it does not exist.
      - Using OVN/SRIOV CNI API, it will populate remote DHCP/DNS Servers.
      - TBD : Need to understand OVN controller and SRIOV controller capabilities and figure out the functionality of this API in this plugin.
  - Delete Virtual Link:
    - API : Exposes API to delete virtual network
    - Functionality:
      - If there is no reference to this network (if use count is 0), then using OVN/SRIOV controllers, it deletes the virtual network.
  - Create persistent volume
    - Create volume that needs to exist across VNF life cycle.
  - Delete persistent volume

- Delete volume

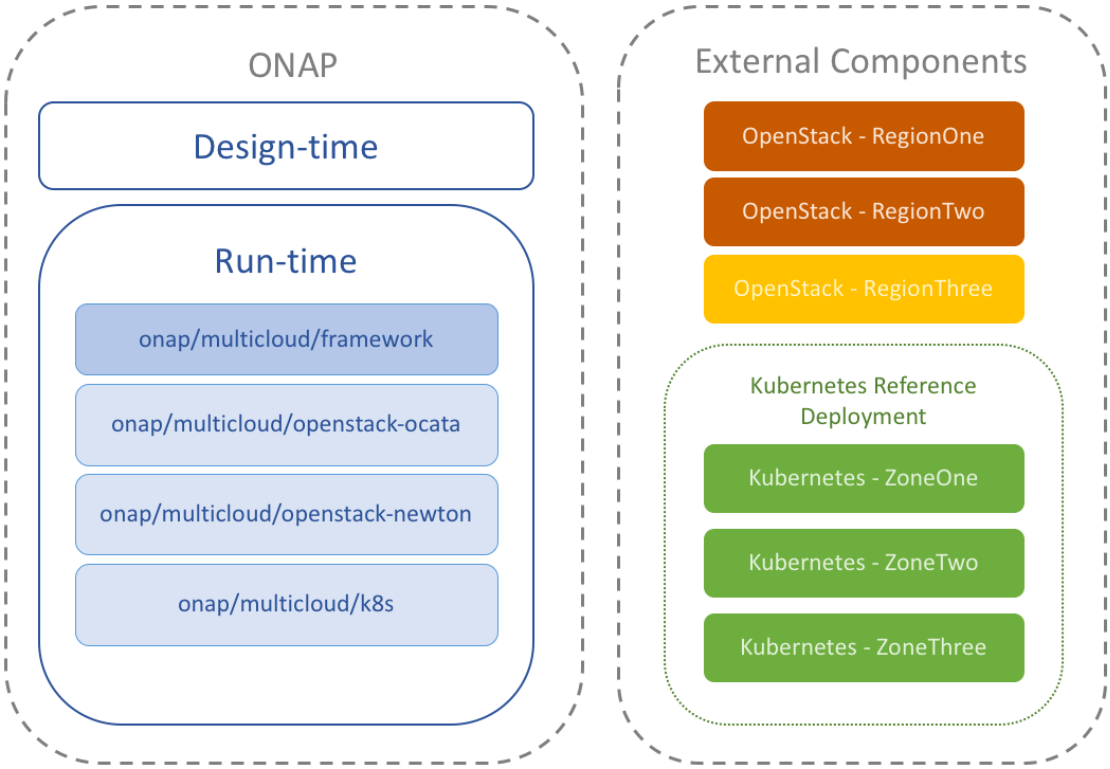
Work Items

Two major efforts as part of this activity

1. Kubernetes Reference Deployment
2. K8S plugin in Multi-Cloud/VIM

Kubernetes Reference Deployment

Offers Ansible playbooks for installing a Kubernetes Deployment with additional components required for ONAP MultiCloud plugin. Its temporal repository is https://github.com/electrocucaracha/krd



Activities:

Activity (Non ONAP related, but necessary to prove K8S plugin)	Owner	Status
Add K8S installation scripts	Victor Morales	Done
Add flannel Networking support	Victor Morales	Done
Add OVN ansible playbook	Victor Morales	Done
Create functional test to validate OVN operability	Victor Morales	In progress
Add Virtlet ansible playbook	Victor Morales	Done
Create functional test to validate Virtlet operability	Victor Morales	In progress
Prove deployment with EdgeXFoundry containers with flannel network	ramamani yeleswarapu	

Prove deployment with one VM and container sharing flannel network	ramamani yeleswarapu	
Prove deployment with one VM and container sharing CNI network	ramamani yeleswarapu	
Add Multus CNI ansible playbook	ramamani yeleswarapu	In progress
Create functional test to validate Multus CNI operability	ramamani yeleswarapu	
Prove deployment with one VM (firewall VM) and container (simple router container) sharing two networks (both from OVN)	ramamani yeleswarapu	
Prove deployment with one VM and container sharing two networks (one from OVN and another from Flannel	ramamani yeleswarapu	
Document how the usage of the project	Victor Morales	In progress
Add Node Feature Discovery for Kubernetes	Victor Morales	
Create functional test for NFD	Victor Morales	

MultiCloud/Kubernetes Plugin

Translates the ONAP runtime instructions into Kubernetes RESTful API calls. Its temporal repository is https://github.com/shank7485/k8-plugin-multicloud

Activities:

Activity	Owner	Status
Create a layout for the project	Shashank Kumar Shankar	Done
Create a README file with the basic installation instructions	Shashank Kumar Shankar	Done
Define the initial swagger API	Shashank Kumar Shankar	Done
Implement /vnf_instances POST endpoint	Victor Morales	Done
Implement the Create method for VNFInstanceClient struct	Victor Morales	Done
Implement /vnf_instances GET endpoint	Victor Morales	Done
Implement the List method for VNFInstanceClient struct	Victor Morales	Done
Implement /vnf_instances/{name} GET endpoint	Victor Morales	In progress
Implement the Get method for VNFInstanceClient struct	Victor Morales	In progress
Implement /vnf_instances/{name} PATCH endpoint	Shashank Kumar Shankar	In progress
Implement the Get method for VNFInstanceClient struct	Shashank Kumar Shankar	In progress
Implement /vnf_instances/{name} DELETE endpoint	Shashank Kumar Shankar	Done
Implement the Delete method for VNFInstanceClient struct	Shashank Kumar Shankar	Done
Create the struct for the Creation response	Victor Morales	
Create the struct for the List response	Victor Morales	
Create the struct for the Get response	Victor Morales	
K8S Plugin API definition towards rest of ONAP for compute	Shashank Kumar Shankar	

K8S Plugin API definition towards rest of ONAP for networking	Shashank Kumar Shankar	
K8S plugin API definition towards rest of ONAP for storage (May not be needed)	Shashank Kumar Shankar	
Merge KRD and plugin repo and upload into the ONAP official repo	Victor Morales	
SO Simulator for compute	Shashank Kumar Shankar	

K8S plugin for compute
 	
Instantiation time:
 	
▪ Loading artifacts based on the order
▪ For each artifact
  ▪ Updating loaded artifact based on API information.
  ▪ Updating loaded artifact based on variables
  ▪ Making calls to K8S (Getting endpoint to talk to from ESR registered repo)		

Testing with K8S reference deployment with hardcoded flannel configuration at the site (Using EdgeXFoundry) - Deployment yaml files to be part of K8S plugin (uploaded manually)	ramamani yeleswarapu	
K8S Plugin implementation for OVN	Ritu Sood	
SO simulator for network		
Testing with K8S reference deployment with OVN networking (using EdgeXFoundry)	Victor Morales	
Testing with K8S reference deployment with OVN with VM and containers having multiple interfaces		
K8S plugin - Artifact distribution Client to receive artifacts from SDC (Mandatory - On demand artifact download, pro-active storage is stretch goal)		
Above test scenario without harcoding yaml files in K8S plugin		
K8s plugin - Download Kube Config file form AAI and use it to authenticate/operate with a Kubernetes cluster	Shashank Kumar Shankar	
K8s plugin - Add an endpoint to render Swagger file	Shashank Kumar Shankar	

Note: Once above list is decided, appropriate JIRA stories will be created.

Projects that may be impacted

Project	Possible impact	Workaround	owner	Status
SO	Ability to call generic VNF API	Until SO is enhanced to support ▪ TOSCA orchestartion ▪ VNF level Abstract API SO will be simulated to test the K8S plugin and reference deployment	SO simulation owner: ???	
SDC	May not be any impact, but need to see if there any impact ▪ for adding new artifacts. ▪ supporting download requests on specific artifcacts		Owner : Libo	
A&AI AND ESR	May not be any impact, but need to see whether any schema changes are required • Add Kubeconfig related data on per cloud-region basis. Check whether any existing fields in cloud-region can be used to store this information or introduce new attributes in the schema (under ESR)		Owner : Shashank and Dileep	

MSB /ISTIO	No impact on MSB. But fixes required to do following: Integration with ISTIO CA to have the certificate enrolled for communicating with other ONAP servceis Also to communicate with remote K8S master.			

Activities that are in scope for phase1 (Stretch goals)

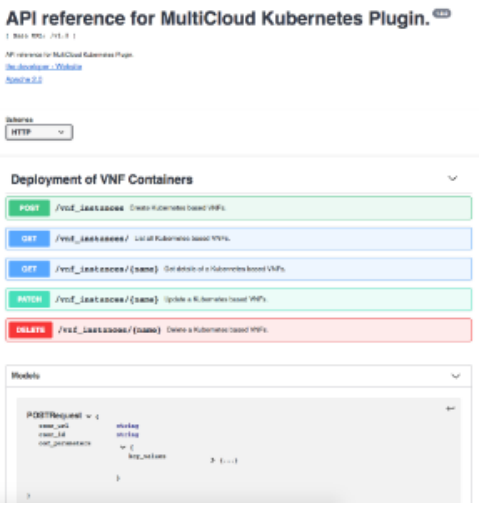
Activity	Owner	Status
K8S node-feature discovery and population of A&AI DB with the features		
Support for Cloud based CaaS (IBM, GCP to start with)		

Scope

- Support for K8S based sites (others such as Dockerswarm, Mesos are not in the scope of Casablanca)
- Support for OVN and flannel based networks in sites
- Support for virtlet to bring up VM based workloads (Others such as Kubevirt is for future)
- Support for bare-metal containers using docker run time (Kata containers support will be taken care later)
- Multiple virtual network support
- Support for multiple interfaces to VMs and containers.
- Proving using VFW VM, Simple router container and EdgeXFoundy containers.
- Support for K8S deployment and other yaml files as artifacts (Helm charts and pure TOSCA based container deployment representation is beyond Casablanca)
- Integration with ISTIO CA (for certificate enrolment)

API/Interfaces

Swagger API:



Key Project Facts:

This project will be subproject of Multicloud project.

NOTE: if this effort is sub-project of multicloud as ARC committee recommended, this will be same to multicloud's.

Facts	Info
PTL (first and last name)	same to multicloud project
Jira Project Name	same to multicloud project
Jira Key	same to multicloud project
Project ID	same to multicloud project
Link to Wiki Space	Support for K8S (Kubernetes) based Cloud regions

Release Components Name:

Note: refer to existing project for details on how to fill out this table

Components Name	Components Repository name	Maven Group ID	Components Description
container	multicloud/container	org.onap.multicloud.container	container orchestration engine cloud infrastructure

Resources committed to the Release:

Note 1: No more than 5 committers per project. Balance the committers list and avoid members representing only one company.

Note 2: It is critical to complete all the information requested, that will help to fast forward the onboarding process.

Role	First Name Last Name	Linux Foundation ID	Email Address	Location
committer	Victor Morales	electrocucaracha	victor.morales@intel.com	PT(pacific time zone)
contributors	munish agarwal		Munish.Agarwal@ericsson.com	
	Ritu Sood	ritusood	ritu.sood@intel.com	PT(pacific time zone)
	Shashank Kumar Shankar		shashank.kumar.shankar@intel.com	PT(pacific time zone)
	ramamani yeleswarapu		ramamani.yeleswarapu@intel.com	PT(pacific time zone)

	Kiran Kamineni		kiran.k.kamineni@intel.com	PT(pacific time zone)
	Bin Hu	bh526r	bh526r@att.com	
	libo zhu			
	Manjeet Singh Bhatia	manjeets	manjeet.s.bhatia@intel.com	PT(pacific time zone)
	Phuoc Hoang	hoangphuocbk	phuoc.hc@dcn.ssu.ac.kr	
	Mohamed ElSerngawy	melserngawy	mohamed. elserngawy@kontron.com	EST
	Komer Poodari	kpoodari	kpoodari@berkeley.edu	PST
	ramki krishnan	ramkri123	ramki krishnan	PST
Interested (will attend my first on 20180206) - part of oom and logging projects	Michael O'Brien	michaelobrien	frank.obrien@amdocs.com	EST (GMT-5)