ONAP MultiCloud/K8s Casablanca

Victor Morales

https://about.me/electrocucaracha
Casablanca’s Accomplishments

1. MultiCloud/K8S plugin (https://github.com/onap/multicloud-k8s/tree/master/src/k8splugin)
3. OVN4NFVK8S (https://github.com/opnfv/ovn4nv-k8s-plugin)
4. vFirewall Use case:
   b. VMs (https://github.com/onap/multicloud-k8s/blob/master/vagrant/tests/integration_vFW.sh)
MultiCloud/K8S plugin

ONAP Multi-Cloud plugin written in Go programming language which offers an API for interacting with Cloud regions supporting Kubernetes.

Requirements:
• Go 1.11
• Docker
• docker-compose

Installation:

$ mkdir -p /opt/{kubeconfig,consul/config}
$ cp $HOME/.kube/config /opt/kubeconfig/krd
$ export KUBE_CONFIG_DIR=/opt/kubeconfig
$ git clone https://git.onap.org/multicloud/k8s/
$ cd deployments
$ ./build.sh
$ docker-compose up -d

API

POST - /v1/vnf_instances/
GET - /v1/vnf_instances/{cloudRegionID}/{namespace}
DELETE - /v1/vnf_instances/{cloudRegionID}/{namespace}/{externalVNFID}
GET - /v1/vnf_instances/{cloudRegionID}/{namespace}/{externalVNFID}
POST - /v1/vnf_instances/

Create VNF

1. CSAR ID
   Cloud Region ID
   OOF Parameters
   Network Parameters

2. Download specific Kubeconfig and available Network Annotations

3. Extract CSAR to get KBs yamls

4. Validate parameters and add them to the KBs yamls

5. Pass the updated yamls to cluster

AAI

KBS Plugin

KBS Cloud Region

Multicloud CSAR directory
Kubernetes Reference Deployment (KRD)

Offers a reference for deploying a Kubernetes cluster that satisfies the requirements of ONAP multicloud/k8s plugin. Its ansible playbooks allow to provision a deployment on Bare-metal or Virtual Machines.
KRD provisioning process

1. **vagrant up installer**
   - generate host.ini
   - libvirt VM creation

2. **install_k8s**
   - get kubespray source code
   - ansible-playbook cluster.yml

3. **install_addons**
   - configure-virtlet.yml
   - configure-ovn4nfv.yml

4. **install_plugin**
   - _print_kubernetes_info
vFirewall ONAP Use Case

It is composed of three virtual functions (VFs):

- **Packet generator**: Sends packets to the traffic sink through the firewall. This includes a script that periodically generates different volumes of traffic.
- **Firewall**: Reports the volume of traffic passing through to the ONAP DCAE collector.
- **Traffic sink**: Displays the traffic volume that lands at the sink using the link [http://192.168.20.250:667](http://192.168.20.250:667) through your browser and enable automatic page refresh by clicking the "Off" button. You can see the traffic volume in the charts.
VMs Provisioning

packetgen

1. kubernetes.io/target-runtime: virtlet.cloud
2. Ubuntu image?
   - Yes: pull image
   - No: libvirt VM creation
3. cloud-init execution
4. packetgen script execution
5. install dependencies
6. download ONAP artifacts
7. start vpp  
8. start honeycomb

firewall

1. kubernetes.io/target-runtime: virtlet.cloud
2. Ubuntu image?
   - Yes: pull image
   - No: libvirt VM creation
3. cloud-init execution
4. firewall script execution
5. install dependencies
6. download ONAP artifacts
7. start vpp
8. start vpp_measurement_reporter

https://github.com/electrocucaracha/vFW-demo
Containers
Provisioning

Prebaked Images

Sink image?

pull image

dockershim creation

add route

Darkstat image?

pull image

dockershim creation

start darkstat

https://github.com/electrocucaracha/cFW-demo
Q & A