How to Bring your Virtual Machine VNF to Container World?

Tomofumi Hayashi, Red Hat
Disclaimer

The content set forth herein is Red Hat confidential information and does not constitute in any way a binding or legal agreement or impose any legal obligation or duty on Red Hat.

This information is provided for discussion purposes only and is subject to change for any or no reason.
The Motivation is...

Containerized app brings more agility/efficiency/flexibility than virtualized app...

So what about NFV?
Is It Easy to Make a VNF Container?

- Physical Router
  - Console
  - Mgmt Interface
  - Kernel (modified)

- Virtualized Router
  - Console
  - Mgmt Interface
  - Kernel (modified)

- Containerized Router
  - Console
  - Mgmt Interface
  - Generic Kernel
CNF = Container Network Function? or Cloud Native Network Function?

From [https://www.cncf.io/about/charter/](https://www.cncf.io/about/charter/), cloud native systems should be:

(a) **Container packaged**.
(b) Dynamically managed.
(c) Micro-services oriented.
Agenda:

- Network Device Functions for Containers
  - Data Plane
  - User Interface
  - Orchestration
- Open Source Projects for Container Network Functions
Network Device Functions for Containers VNF?

- **Orchestration**
- **User Interface**
- **Control Plane**
- **Data Plane**

- Deploy
- Configuration & Operation
- Telemetry
- Multiple networks in Kubernetes
- SR-IOV
Data Plane (Multiple Networks in Kubernetes)

The Kubernetes Pod always have one interface to connect Kubernetes networks. But sometimes VNFs want to use multiple interfaces

• To serve L2 network functions (e.g. vCPE use-case)
• To isolate networks from other Pod/Users
Data Plane (Multiple Networks in Kubernetes)

There are two working groups in K8s community, under network-SIG:
- Network Plumbing WG ([meeting agenda/info](#))
- Network Service Mesh WG ([meeting agenda/info](#))
Data Plane (Multiple Networks in Kubernetes)

<<<<<< They have talk/tutorial in ONS!!! >>>>>>>

- Network Plumbing WG
  - Tutorial: Tutorial: NFV features in Kubernetes” at G102 (right now!)
- Network Service Mesh WG
  - Talk: “Network Service Mesh: An Attempt to Reimagine NFV in a Cloud-Native Fashion” Tomorrow (Sep 26, 14:30 - 15:00, G106/7)
Multiple Interface in Kubernetes

The Core Concept is....

- All traffic goes through eth0
  - (Liveness and Readiness) Probes
  - Communication between API and Pod
  - User Traffic
Multiple Interface in Kubernetes (cont'd)

The Core Concept is....

- User Traffic
- (Liveness and Readiness) Probes
- Communication between API and Pod
Data Plane (Multiple Networks in Kubernetes)

There are two working groups in K8s community, under network-SIG:

- **Network Plumbing WG**
- **Network Service Mesh WG**
Data Plane (Multiple networks in Kubernetes)

Network Plumbing WG

- Making de-facto standard document/specs
  - Kubernetes Network Custom Resource Definition De-facto Standard Version 1
  - V2 currently under development...
- Implement multus-cni as its reference implementation
  - meta-plugin to multiplex network CNI plugins
Multiple Interface in Kubernetes (cont'd)

multus-cni example

apiVersion: "k8s.cni.cncf.io/v1"
kind: NetworkAttachmentDefinition
metadata:
  name: foobar
spec:
  config: '{ "type": "vlan", (snip) }'

Kubernetes servers (api, kubelet so on)

Network attachments

Pod
  - net0
  - eth0
  kind: Pod
  … (snip)…
  annotations:
    k8s.v1.cni.cncf.io/networks: foobar

default network

another network

vlan
Data Plane (Multiple Networks in Kubernetes)

There are two working groups in K8s community, under network-SIG:

- Network Plumbing WG
- Network Service Mesh WG
Data Plane (Multiple networks in Kubernetes)

Network Service Mesh WG

- Provide network service (L2, L3 and others) into Kubernetes from scratch
- Interacts with Device Plugin API (DPAPI) without CNI
  - Provide a brand new network framework in Kubernetes
- Implementation: [github.com/ligato/networkservicemesh](http://github.com/ligato/networkservicemesh)
Network Device Functions for Containers VNF?

- **Orchestration**
- **User Interface**
- **Control Plane**
- **Data Plane**

- **Deploy**
  - Configuration & Operation
  - Telemetry

- Multiple networks in Kubernetes
- SR-IOV and userspace
SR-IOV

- [https://github.com/hustcat/sriov-cni](https://github.com/hustcat/sriov-cni)
  - without any resource management...
- [https://github.com/intel/sriov-network-device-plugin](https://github.com/intel/sriov-network-device-plugin)
  - CNI plugin + device plugin for resource management
- Network Service Mesh
  - device plugin only
SR-IOV

- [https://github.com/intel/sriov-network-device-plugin](https://github.com/intel/sriov-network-device-plugin)
  - resource management with Device Plugin API (DPAPI)
    - Step 1) Before the pod launch, Device Plugin allocates VFs
    - Step 2) Its CNI plugin configures VF, given from Device Plugin
  - Mainly discuss at
    - k8s/resource-management working group
    - network plumbing working group

Note: [https://github.com/zshi-redhat/virt-network-device-plugin](https://github.com/zshi-redhat/virt-network-device-plugin) provides SR-IOV emulation with virtio_net for PoC/Demo
Userspace

https://github.com/intel/userspace-cni-network-plugin (active)

- In very early development phase
- Create virtual interface (other than veth)
- Connect to virtual switch
  - OvS-DPDK
    - vhostuser interface
  - VPP
    - memif interface
Network Device Functions for Containers VNF?

- Orchestration
  - Deploy
- User Interface
  - Configuration & Operation
  - Telemetry
- Control Plane
- Data Plane
  - Multiple networks in Kubernetes
  - SR-IOV and userspace
Configuration and Operation: Gap

Strongly related to its lifecycle:

- **Container**
  - Stateless
  - Read once, no changed (delete and launch again if config change)
- **Network device**
  - STATEFUL!
  - Changed on-demand
Configuration and Operation (cont’d)

Infra in Kubernetes:

• **Custom Resources**
• **Admission Controllers/Dynamic Admission Control**
• (skipped: Overlay Mount Filesystem, provide ARG in Pod...)

Infra in Networking:

• RESTCONF/NETCONF/YANG for modeling
• gRPC/ssh/http for transport
Configuration ... (cont’d) - Infra in Kubernetes

Custom Resources:

- Create Original ‘Resource’ object in Kubernetes
- User can create/modify through k8s API
- multus-cni uses custom resources as following:

```yaml
apiVersion: "k8s.cni.cncf.io/v1"
kind: NetworkAttachmentDefinition
metadata:
  name: foobar
```

Pod

annotations:
  k8s.v1.cni.cncf.io/networks: foobar
Admission Controllers/Dynamic Admission Control:

- Intercepts requests for Kubernetes API (to create custom resource, for example) after auth, before its persisted
- ValidatingAdmissionWebhook is used to hook the request and do validation
Configuration and Operation (cont’d)

Infra in Kubernetes:

- Custom Resources
- Admission Controllers/Dynamic Admission Control
- (skipped: Overlay Mount Filesystem, provide ARG in Pod...)

Infra in Networking:

- RESTCONF/NETCONF/YANG for modeling
- gRPC/ssh/http for transport
Configuration... in Networking

- IETF netmod WG: NETCONF, RESTCONF/YANG
  - IETF netmod WG provides data models in YANG
  - NETCONF/RESTCONF uses ssh/http(s)/TLS/SOAP/TLS for transport
- OpenConfig: NETCONF, RESTCONF, gRPC/YANG
  - OpenConfig provides common data models in YANG
  - OpenConfig also defines gNMI (gRPC Network Management Interface)
Configuration and Operation (cont’d)

Talk: Beyond the Command Line: Programming Network Devices with gRPC and OpenConfig

(Day3, September 27, 16:45 - 17:15, G106/107)
Network Device Functions for Containers VNF?

- **Orchestration**
  - Deploy

- **User Interface**
  - Configuration & Operation
  - Telemetry

- **Control Plane**
  - Multiple networks in Kubernetes

- **Data Plane**
  - SR-IOV and userspace
Telemetry

Infra in Container:
- http/https (for Prometheus)

Infra in Networking:
- YANG-PUSH in IETF netconf wg
- gNMI, Streaming Telemetry in OpenConfig
- **VES (VNF Event Stream)** in OPNFV
- and so on (vendor specific way and yeah, we have SNMP!)
Telemetry

Additional consideration in case of container environment:

- Should we provide all information for each container?
  - Some info is host specific, not container specific.
- Is container telemetry suitable for Telco?
  - Prometheus exporter (TLS with nginx)
  - Prometheus exporter consumes TCP ports....
Network Device Functions for Containers VNF?

- Orchestration
- User Interface
- Control Plane
- Data Plane

- Deploy
- Configuration & Operation
- Telemetry
- Multiple networks in Kubernetes
- SR-IOV and userspace
Deploy

Kubernetes and its community provide following tools:

• **Helm**
• **Operator Framework**
Deploy

**Helm**

- Package manager for Kubernetes
- Package, 'charts', provides several Kubernetes resources
  - Pod
  - Configmap
  - Service
  - Deployment
Deploy

**Operator Framework**

- Framework to manage Kubernetes native applications
  - can be used for deployment as well as automation
  - e.g: etcd-operators, prometheus-operators
- Operator creates custom resource to manage applications
- Each operators has associated 'operator pod'
  - to watch custom resource objects and
  - to keep apps based on its custom resources (e.g. # of replica)
Deploy Operator Framework (in case of etcd-operator)

Operator Pod:

check CRD object

watch pod and change it as CRD's spec

Kubernetes api-server

etcd pod1  etcd pod2  etcd pod3

apiVersion: "etcd.database.coreos.com/v1beta2"
kind: EtcdCluster
metadata:
  name: example-etcd-cluster
  .... (snip)
spec:
  size: 3
Open Source Projects for Container VNF

Container VNF:
- Container4NFV project in OPNFV
- Metaswitch's Clearwarter docker integration
- Metaswitch's Clearwater Kubernetes integration by Intel

Cloud native VNF:
- Clover in OPNFV
Wrap-up:

• Network Device Functions for Containers
  • Data Plane
  • User Interface
  • Orchestration
• Open Source Projects for Container Network Functions
Thank you! Questions?

Slides available at https://onseu18.sched.com/

@s1061123
Click to add Title

• Click to add Text