Who says You Shouldn’t Run VMs and Containers Together?

Ian Jolliffe
Wind River

Glenn Seiler
Wind River
Agenda

- Making the Case for Both VMs and Containers
- State of Container Usage in Telco today
- Options for running VMs and Containers
- An example: StarlingX – Distributed Edge Cloud
Ok, We All Know Containers ‘Rule the Edge’

- Best where low latency, resiliency, and portability are extremely important
- Run the maximum amount of particular applications on a minimum of servers
- Containers are useful for deploying short-lived and ephemeral services
- Models and applications where a problem can be divided into small sets of tasks
But There is A Management Issue with Containers

An order-of-magnitude increase in the number of components that must be managed and monitored.

Too many ‘services’ increases mgmt overhead.

**Average Container Lifetimes**

- **Orchestrated:** 0.5 days
- **Unorchestrated:** 6 days
- **All Containers:** 2 days

**ORCHESTRATED CONTAINERS CHURN 12X FASTER**

Source: Datadog
So When Do We Want to Run VMs (or VNFs)

- Running multiple applications on servers and/or have a wide variety of operating systems
- If state is required
- Networking and performance accelerators such as CPU Pinning, NUMA, DPDK, SR-IOV
- Don’t need to update or replace the workload often
- Workloads that need to look at every single packet; including service chaining
- A tighter security model than what is available for containers today

“82% of respondents indicated they are deploying or plan to execute VNFs on uCPE located at customer sites”

Michael Howard   IHS Markit
Some Facts About Container Adoption in Telco

- Multi-tenant servers represented 62% of total servers in 2018 survey
- Telcos and enterprises run containers in only 5% of their multi-tenant servers
- 16% of multitenant telco servers will have a container-OS in 2022
- Hypervisors running virtual machines that contain a container OS accounted for 18% hypervisors deployed

Source: 2018 Multi-Tenant Server Software Market Tracker
So you want to "transform" to CNFS
#1: “Containers in a VM” Hybrid Architecture

Containers are hosted by VMs

**PROS:**
- Supports COEs and Containers
- Maintains “Pet” approach for critical VMs including containers in a VM
  - Live migration, monitoring & fault recovery
  - Scaling of and load balancing across hosts

**CONS:**
- Heavyweight at the edge
- Complex; lot of moving parts
#2: Bare metal Containers in a VM Hybrid Architecture

Provides a bare metal environment within OpenStack (with Ironic)

**PROS:**
- Flexible, proven method
- Containers running Natively; No Hypervisor
- Containers managed by COE
- Maintain VM/OpenStack Investment

**CONS:**
- Complex
- Heavy control plane for Edge
- Can’t extend some benefits of VMs to containers
#3: Kata Containers architecture

Provides a bare metal environment within (OpenStack) VMs

**PROS:**
- Increased isolation
- Managed by Kubernetes
- Behaves like a container

**CONS:**
- Consumes more resources
#4: Bare Metal Containers with Openstack/VMs

Containerize OpenStack

- On top of a bare metal Kubernetes cluster
- Kubernetes manages the lifecycle of the containerized OpenStack application

VMs and Containers managed the same through Kubernetes and treated as equal citizens

Only run OpenStack (in a container) when needed for VMs

Kubernetes cluster available for non-OpenStack end user applications

Lighter weight approach to deploying containers
An Example: StarlingX; a Container and VM Solution

- Leverages the benefits of Cloud Native
- Incorporates the best VM technology
- With a focus on Operations and Day 2 management at the Edge

Plus much, much more.....

[Diagram showing the combination of Cloud Native Computing Foundation and StarlingX]
What is StarlingX

- Upstream project launched under the OpenStack Foundation in May 2018
- Focus on the Edge use cases
- 2nd release Aug/2019 - Achieved
- Community metrics last 90 days:
  - Contributors from more than 10 organizations
  - 1000 emails on Mailing List
  - 584 commits merged
  - 130 people have submitted changes
Use StarlingX to Deploy Containers

StarlingX
- Virtualized Infrastructure Manager
- Infrastructure Orchestration

Kubernetes
- Helm
- Armada
- Docker Registry

- Host Management
- Configuration Management
- Fault Management
- Software Management
- Service Management

- IPMI
- Horizon
- CEPH
- Keystone
- etcd
- Calico
- Docker
Use StarlingX to Deploy OpenStack and VMs

OpenStack control plane in containers

Kubernetes

Armada

Helm

Docker Registry

Virtual Machine

IPMI

Virtual Machine

Virtual Machine

Virtual Machine

OpenStack control plane in containers

Host Management

Configuration Management

Fault Management

Service Management

Software Management

Infrastructure Orchestration

Virtualized Infrastructure Manager
No Matter what Topology – have your cake and eat it too

Minimum-Footprint Edge Solution
Single server

Highly-Available Edge Solution
Two servers

Frame-Level Solution
4 – 100 servers

Low Footprint
1:1 protected pair of servers

Rack/Mini-DataCenter
Summary

- VM’s are very relevant and needed in Telco
- Containers and Kubernetes enable new use cases, highly dynamic environments at the edge
- There are many options to run VMs and Containers
- StarlingX is a project that enables both of the above in an easy to consume manner