

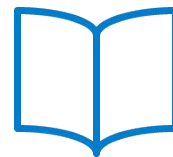


VNFs in a CNF environment

Monika Antoniak, Piotr Skamruk
CodiLime

Agenda

- Who are we?
- Business use case - use existing VNFs in a containerized set-up
- Technical solution to the problem
- Q&A



Who we are?

Who we are

- CodiLime has been providing networking engineering services since 2011
- As part of our R&D efforts and based on our expertise with CNFs and VNFs, we have decided to explore this topic further
- Today we are presenting the working example

Business use case

Business case

What if...

- You love the lightness of containers and use them on a daily basis
- You value the flexibility and resilience of the overall solution
- You want a simple way to deploy things
- You enjoy using kubernetes to manage your resources
- ...and you use a business critical network function as a blackbox VM



What can you do to get all of that?

A step back: VNFs and CNFs

VNF (Virtual Network Function): a well-known way to realize network functions in virtualized or cloud environments

Software is provided as a VM image that cannot be changed into a container image, or is based on an operating system other than Linux

VNF examples: vFW, vBNG, vEPC

CNF (Containerized Network Function): a new way of providing required network functionality with the help of containers

Software is distributed as a container image and can be managed using container-management tools (like docker images in kubernetes env)

CNF examples: vCPE/cCPE, LDAP, DHCP

Back to business

Goal: a converged setup for running containerized and VM-based network functions

- using a single user interface - the same type of definition for both containers and VMs) & controlled by a single API (kubernetes)
- without the need to have Kubernetes on OpenStack, or OpenStack on Kubernetes
- without the need to maintain libvirt installation on top of Kubernetes by hand
- without the need to use libvirt per VM, with custom object definitions for VMs needed by KubeVirt

Technical solution to the problem

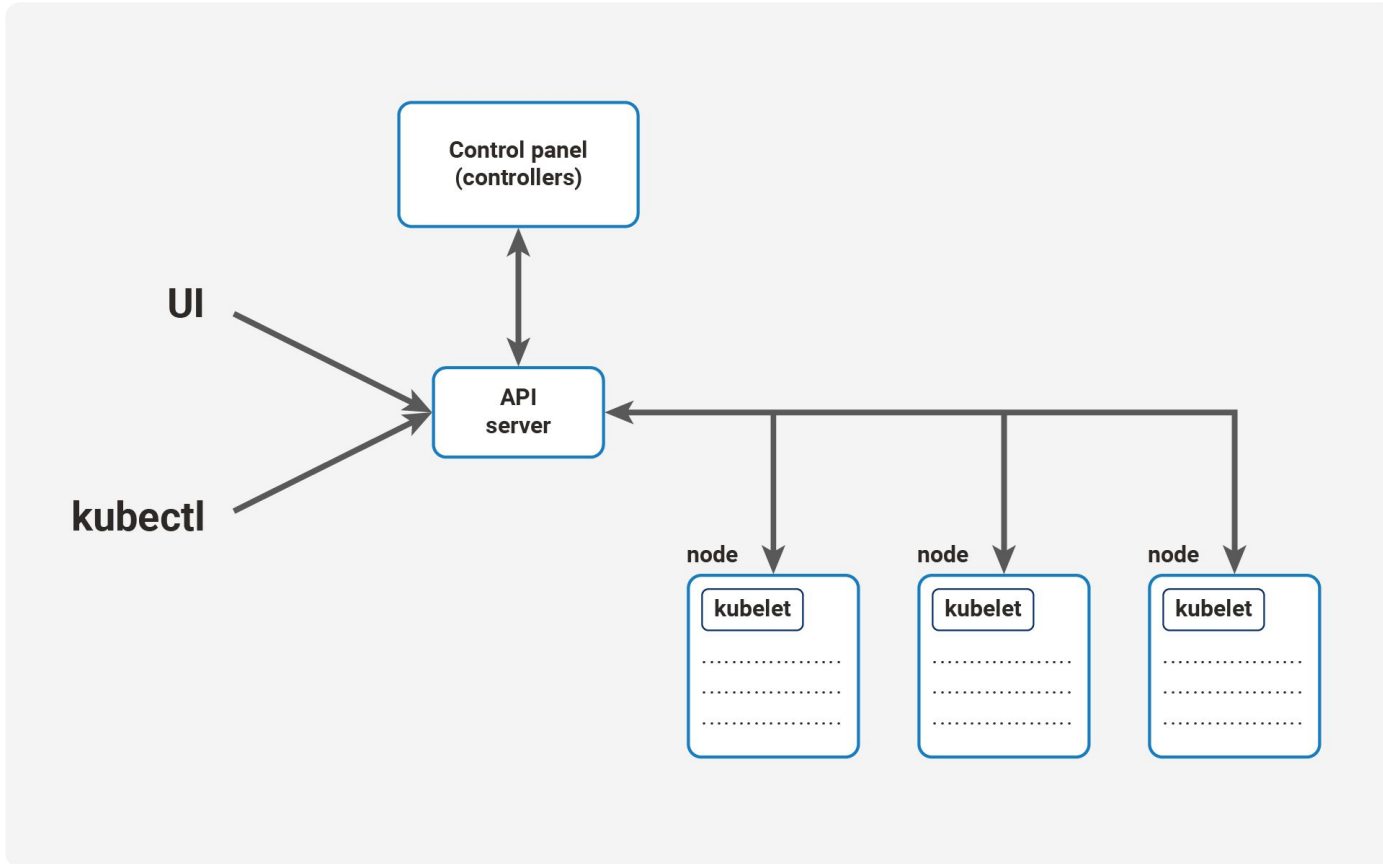
Technical Solution

How to achieve a defined goal in a technically mature way:

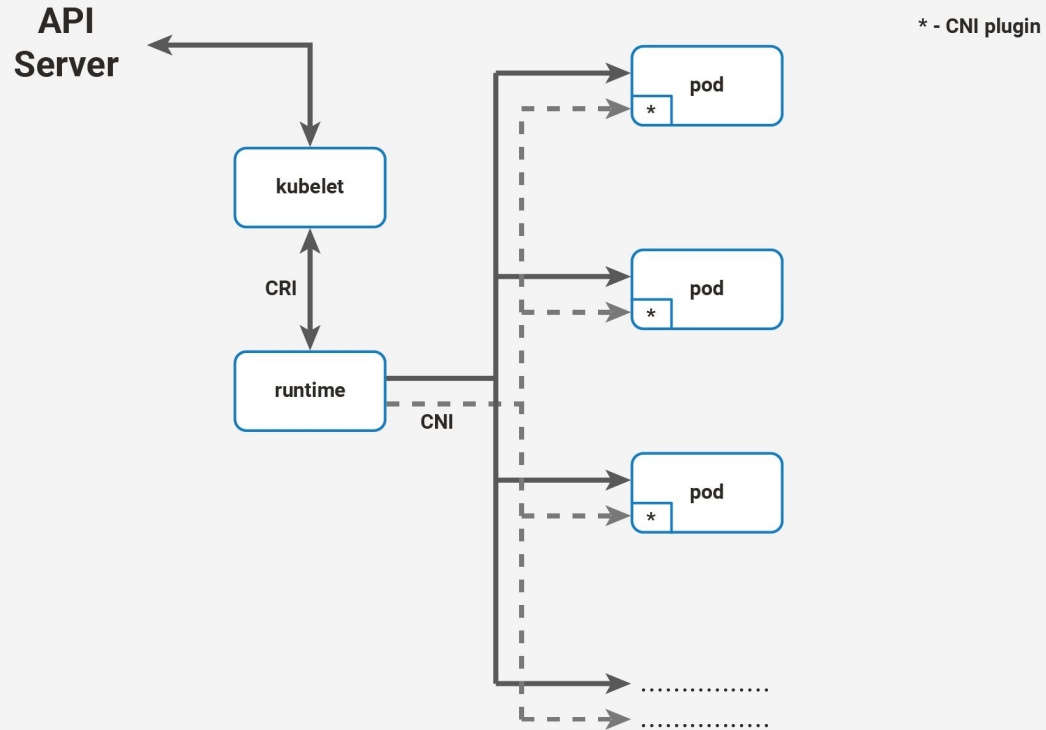
- Use Kubernetes & leverage its CRI mechanism to run VMs as pods
- Configure nodes to use criproxy/Virtlet
- Describe the VM using a pod definition passing the URL to a qcow2 image in place of the usual docker image for a container
- Run such a pod as a part of deployment/replicaset/daemonset...
- You can use this solution in a wide variety of cloud provider setups!



How Kubernetes works



Whats on worker nodes



What is Virtlet?

- CRI-native
- same resource limits as for containers
- same network configuration using CNI (note: highly recommended use of SR-IOV)
- runs qcow2 base images using libvirt
- single libvirt instance per node
- As it's CRI, all VMs will be seen in k8s as pods, so it can be part of other pods using native objects



Virtlet demo

Running virtlet using

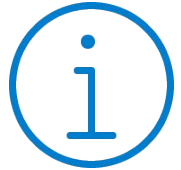
<https://github.com/Mirantis/virtlet/blob/master/deploy/demo.sh>

running VNF defined as pod, using kubectl +
definition of VM in YAML



Where to look for more information

- <https://codilime.com/>
- <https://codilime.com/vnfs-in-cnfs/>
- <https://www.mirantis.com/blog/kubevirt-vs-virtlet-comparison-better/>
- <https://docs.virtlet.cloud>



Q&A



Thank You

Follow Up Links:

kubernetes.codilime.com



Backup

Who we are?

Monika's background is in networking. She spent over 15 years in telcos designing, prototyping, testing and implementing solutions. She has worked with different vendors and thus looked at different solutions and different points of view. To keep abreast of rapidly evolving technology, she has broadened her view and added SDN & NFV to the picture to be able to work with complete, automated network solutions.

Her goal is always to start from a business perspective and then use the technology that best fits a project's aims. Working at the intersection of actual deployment support and within an R&D department allows her to reach her goals with the freedom of using new technologies and the practical approach that comes from real-world use cases.

Who we are?

Piotr is a long-time GNU/Linux and Forth language enthusiast, sys administrator and sys developer. He has worked on kernel sources, backend apps and even on frontends in a wide variety of languages. At Intel he did the kvm flavor for CoreOS RKT, enabling it to run containers on VMs. At Mirantis he was on the Virtlet core team, writing a Kubernetes runtime that made it possible to run VM workloads, based on QCOW2 images as Kubernetes native pods. He's a CNI Maintainer.

Now at CodiLime, he's working on networking solutions for Kubernetes.