Improve Performance and Security for Containers using Kuryr and Cilium

Michal Rostecki
Software Engineer
mrostecki@suse.com

Rossella Sblendido
Team Lead Networking
rsblendido@suse.com
BPF

From https://github.com/cilium/cilium
Cilium architecture

From https://github.com/cilium/cilium
Introducing Kuryr

The two Kuryr-Kubernetes components depicted with all the main components they interact with.
Packet traversal vanilla Kuryr
Why Cilium + Kuryr?

- Support for Network Policies in Kubernetes
- Using BPF as an underlying mechanism for Network Policies
Integration: how it works

- Kuryr-kubernetes controller is running as a deployment
- Kuryr-kubernetes CNI plugin is *not* used
- Cilium CNI plugin is used instead
Integration: challenges

- Cilium CNI plugin had to be extended to:
  - read the OVS bridge name and information about allocated IP from pod annotations
  - create OVS VIFs
How the integration works

User → Neutron → Kuryr ctrl → K8S API → Kubelet → Cilium CNI

Create pod

Request port creation

Add annotation with Neutron port data

Create pod API object

Create pod sandbox and containers

Call CNI plugin to set up netns

Read pod annotation with Neutron port data

Create OVS VIF

Create Cilium endpoint
Packet traversal with Cilium

![Diagram of packet traversal with Cilium](image-url)
Demo time!
Future work

- Provide support for Kubernetes Services
- Provide support for load balancing (preferably by using Cilium BPF programs)
- Gating upstream