September 25 - 27, 2018
Amsterdam, The Netherlands

ons

EUROPE

OPEN NETWORKING //
Integrate, Automate, Accelerate

ons
EUROPE
OPEN NETWORKING //
Integrate, Automate, Accelerate

September 25 - 27, 2018
Amsterdam, The Netherlands

Bringing Userspace Networking to CNI

Billy McFall, Red Hat

Userspace Networking:

Userspace CNI github

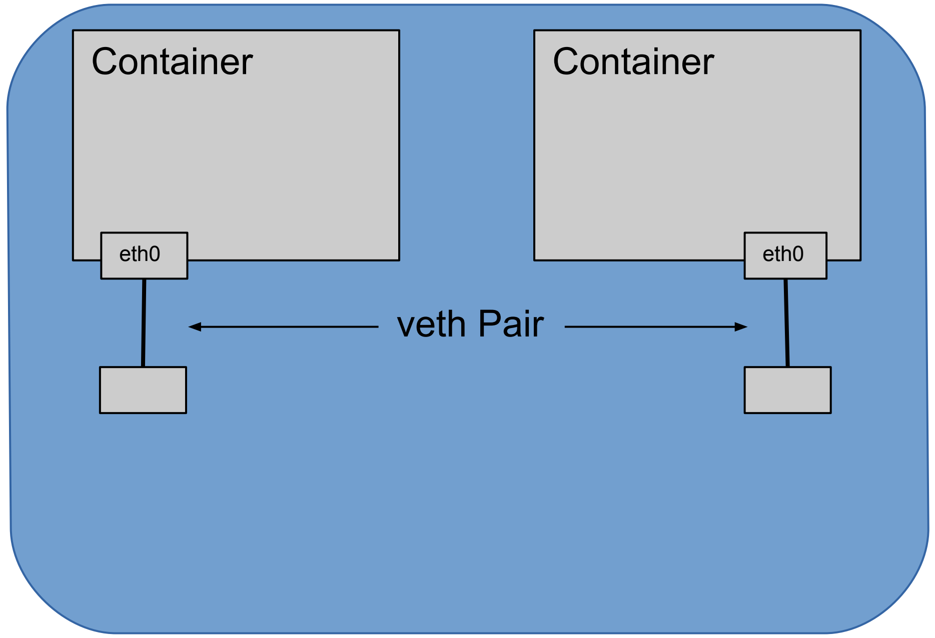
- Default packet processing on Linux distributions occur in the kernel.
- Userspace Networking - Packet processing occurs in software outside the kernel.
  - Kernel no longer manages NIC.
  - Higher packet processing rates and lower latency at cost of higher CPU usage.
- Userspace Networking currently being deployed in a Virtual Machine environment.

- **How to use Userspace Networking in a Cloud-Native based architecture?**

ONS EUROPE
OPEN NETWORKING // Integrate, Automate, Accelerate

Userspace CNI github

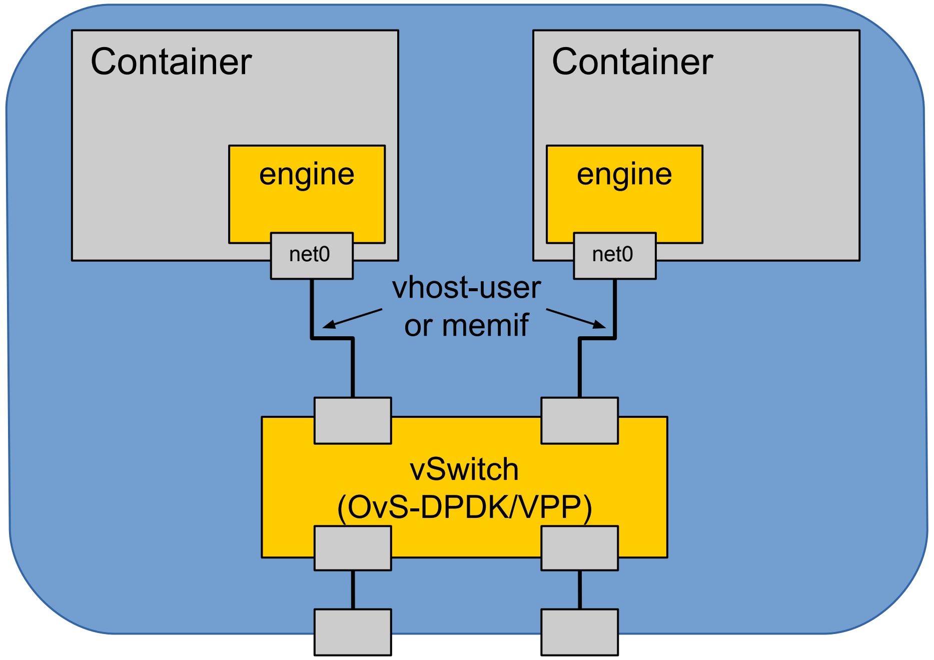
Traditional CNI:



- veth pair created.
- IP applied through namespace.
- eth0 shows up in container and ready to use.
- Nothing needed in container.
- Only one interface in container, so no question as to which interface to use for what.

Userspace CNI github

Userspace Networking in Container:



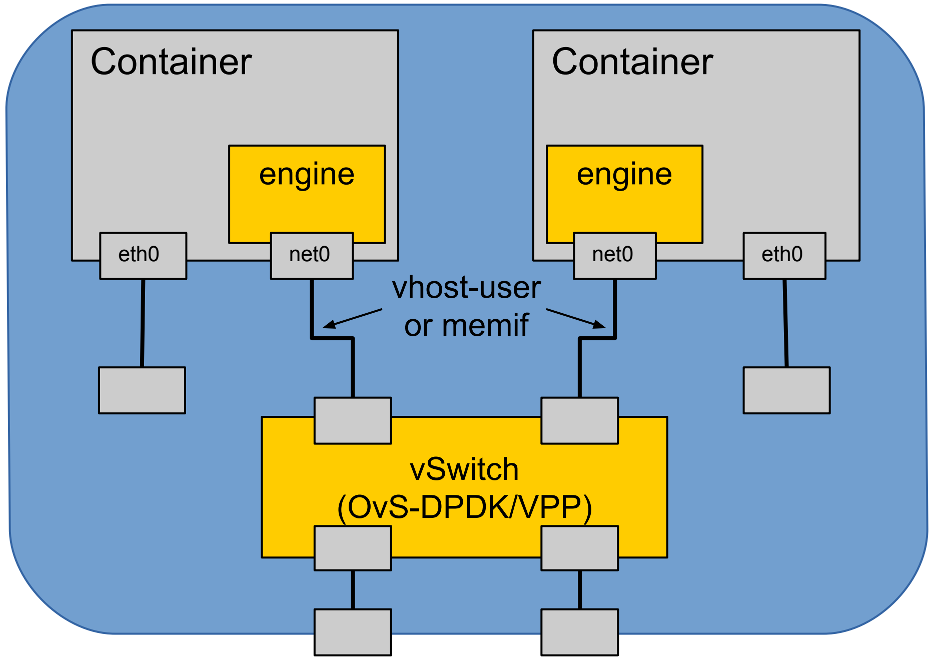
Interfaces (vhost-user/memif) don’t show up in container as straight Ethernet interface.

- Interface needs to be created in vSwitch on host.
- Interface needs to be added to network in vSwitch on host.
- Interface needs to be created in container.
- Interface needs to be added network in container.

ons EUROPE
OPEN NETWORKING // Integrate, Automate, Accelerate

Userspace CNI:

Userspace CNI github



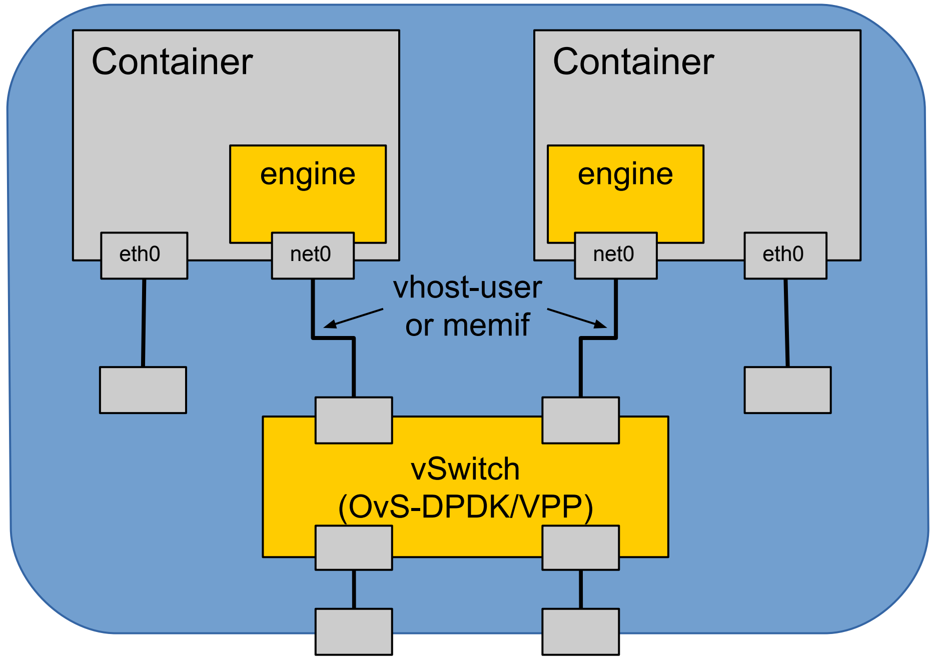
Userspace CNI is being developed by Intel/Nokia/Red Hat to introduce Userspace Networking to Containers. Userspace CNI:

- Interacts with the local vSwitch to create userspace interfaces on host and add interface to network on host.
- Using K8s Admission Controller to pass JSON structured data to container for consumption of interface and network provisioning.
- Implemented for OvS-DPDK and VPP vSwitches.
- Used with Multus for multiple interfaces.

ons EUROPE
OPEN NETWORKING //
Integrate, Automate, Accelerate

Userspace CNI github

Userspace CNI:



Userspace CNI Advantages:

- Higher packet processing speeds and lower latency.
- Not limited to IP traffic into container.
- Additional tunneling protocols can be passed into container or terminated on local vSwitch.

Userspace CNI github

Thank You!

For more information or to help contribute:

- Userspace CNI:

  https://github.com/intel/userspace-cni-network-plugin
- Multus:

  https://github.com/intel/multus-cni

ons EUROPE
OPEN NETWORKING //
Integrate, Automate, Accelerate

Appendix: Userspace CNI Current and Future

Userspace CNI github

Current:

- VPP, using a GO API, provisions local VPP instance. Formats and writes container data to file for container to read. Currently, directory containing file is mounted in the container. Sample app reads configuration and applies data to local VPP instance in container. Sample docker image: https://hub.docker.com/r/bmcfall/vpp-centos-userspace-cni/
- OvS-DPDK provisions local OvS-DPDK instance using a python script that executes ‘ovs-vsctl’ commands. No data currently written to container, but VPP implementation will become more generic shortly.

Future:

- Bring OvS-DPDK up to par with VPP implementation.
- Define spec on how JSON configuration data into container is formatted. Ease consumption of interface in container.
- Add Device Plugin for NUMA Management, node discovery, and dynamic provisioning. Still use CNI for interface plumbing.
- Update Multus to support Admission Controller, needed to automate feeding of configuration file into container.
- DaemonSet that installs and runs vSwitch in a container on host.