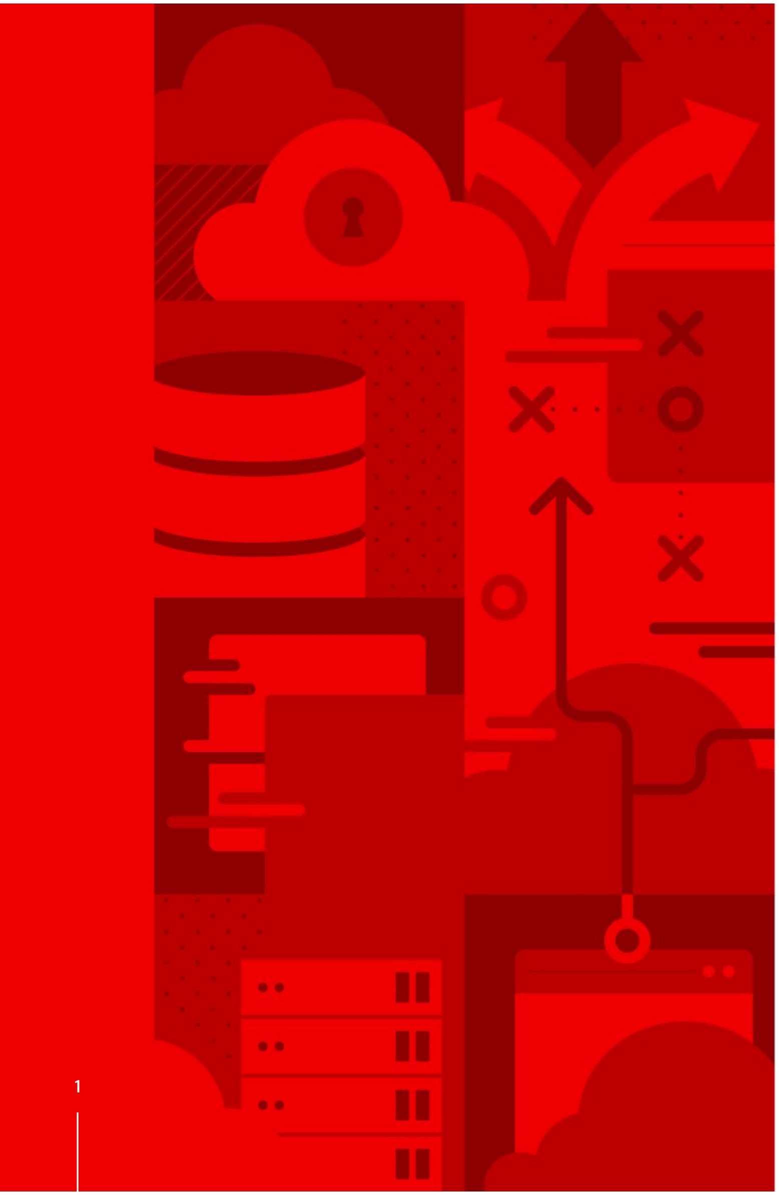


Multi-network in Kubernetes: No batteries included

FOSDEM 2024

Miguel Duarte & Douglas Smith
Engineering @ Red Hat, Inc.

Hi, it's Miguel & Doug!

@dougbtv

Doug Smith

- Technical lead for OpenShift Network Plumbing Team in OpenShift Engineering
- Multus CNI maintainer
- Network Plumbing Working Group member
- Blog: https://dougbtv.com

@maiqueb

Miguel Duarte

- Software engineer in OpenShift Virt Networking
- Network Plumbing Working Group member
- Blog: https://maiqueb.github.io

What’s on the docket?

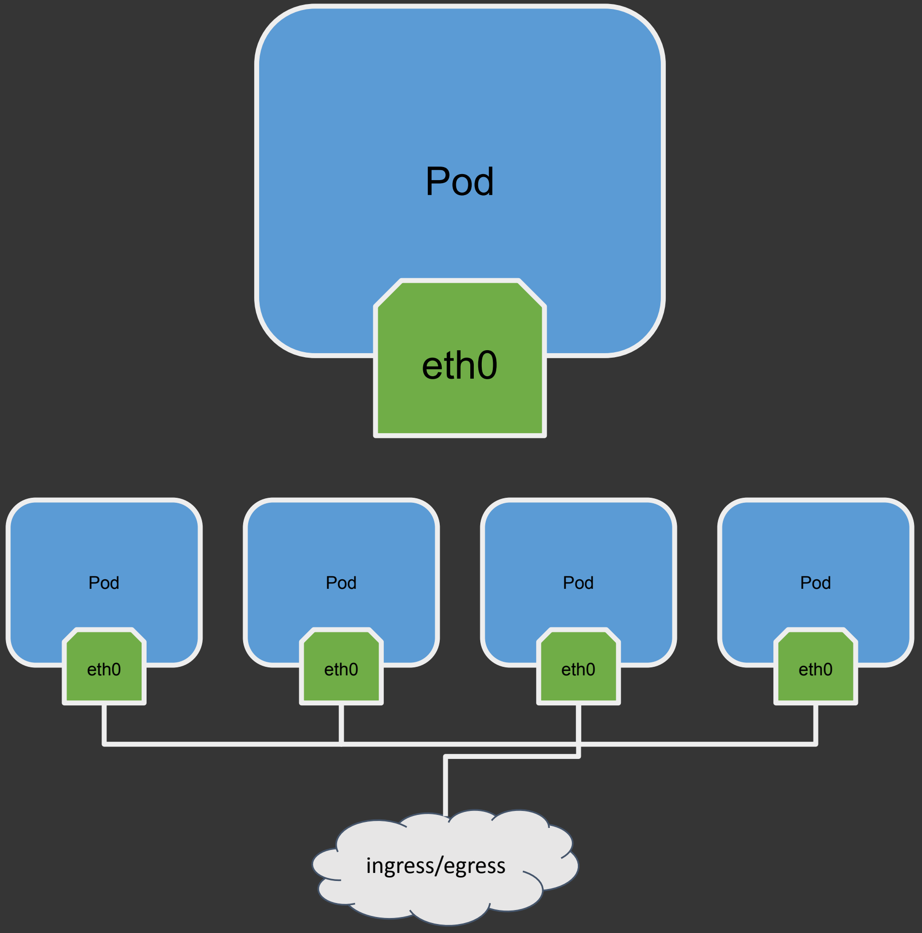
Agenda

- What is Multi-networking in K8s?
- Problem statement.
- What’s the (current) solution
- What does the future look like?
- Lessons Learned
- Watch the demo @ home

What is Multi-networking in Kubernetes?

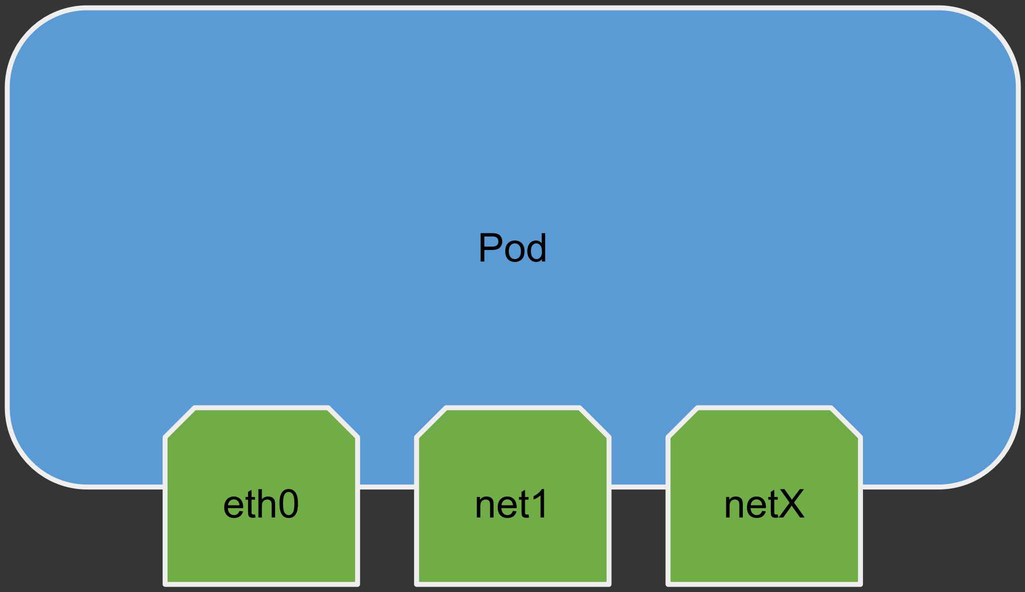
One Lone Network Interface...

Today, by default, we have just one network interface on each pod, that pod is promised network connectivity among pods in the network...



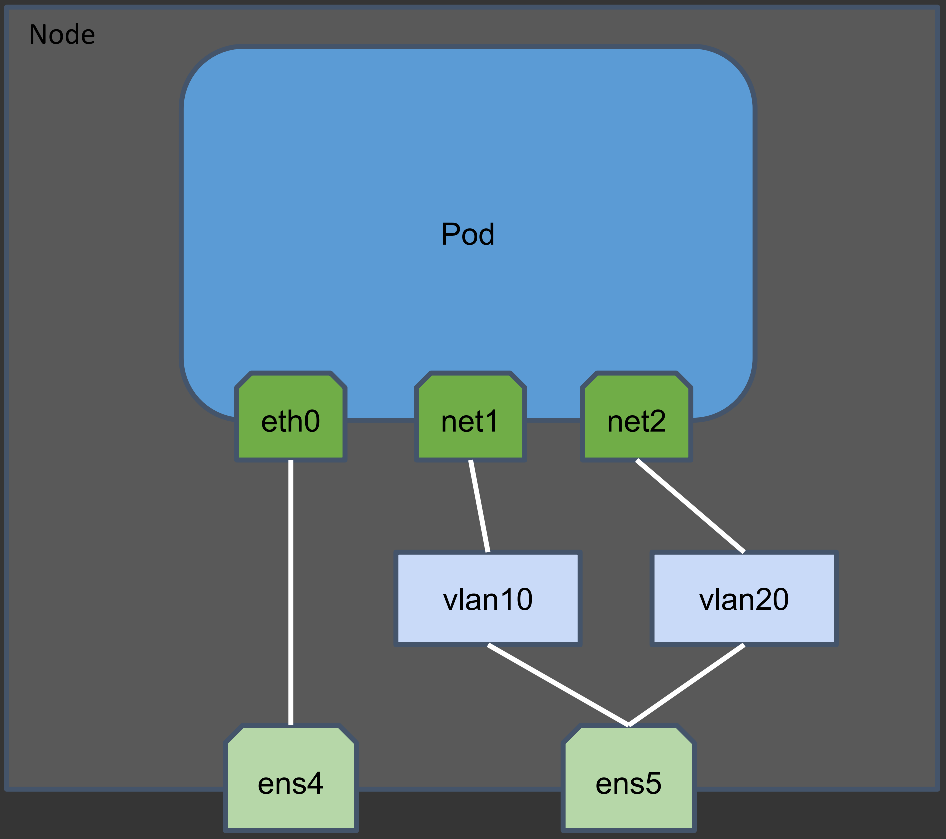
ALL THE NETWORKS!

With multi-networking, we can provide advanced connectivity to complex network topologies, in order to isolate traffic, connect to existing networks, and more.



Example Use Case: Network Isolation

As a Cloud Native Network Function (CNF) vendor I require an additional interface to be provisioned into the Kubernetes Pod. Each of these interfaces has to be in an isolated network for performance and security concerns. The isolation has to be done on a Layer-2.



Problem Statement.

Batteries not included.

Every way to achieve this today requires add-ons.

Nothing is natively in Kubernetes.

User Experience Challenges.

It's just not easy to use, or to debug today.

It requires arcane knowledge.

What's the current solution?

What's the current solution?

Multus CNI

- Out of tree CNI multiplexer (runs multiple CNI plugins)
- Uses CRDs (api extensions) to “trick the platform”
- Requires that you have an “add on” to the platform.
- Mixes JSON and YAML (approximately.)

K8s Native Multi Networking

- Builds on and extends the K8s native API
- Does not require an add on for the API
- Implementation agnostic
- But, it will require an add-on for the implementation.

What's the current solution?

Network Plumbing Working Group / Multus CNI

CNI JSON inside YAML and "out-of-tree"

```
---
kind: Pod
metadata:
  name: pod1
  annotations:
    k8s.cni.cncf.io/networks: '{"name":
"media-net", "interface": "net1"}'
spec:
  [...]
```

```
---
kind: NetworkAttachmentDefinition
[...]
spec:
  conf: '{
  "type": "bridge",
  "name": "media-net",
  "ipam": {"type": "static"}
}'
```

K8s Native Multi-networking / PodNetwork

All YAML and all "in tree"

```
---
kind: Pod
metadata:
  name: pod1
spec:
  networks:
  - podNetworkName: media-net
    interfaceName: net1
  [...]
```

```
---
apiVersion: v1
kind: PodNetwork
metadata:
  name: media-net
spec:
  provider: "myprovider.io/application"
```

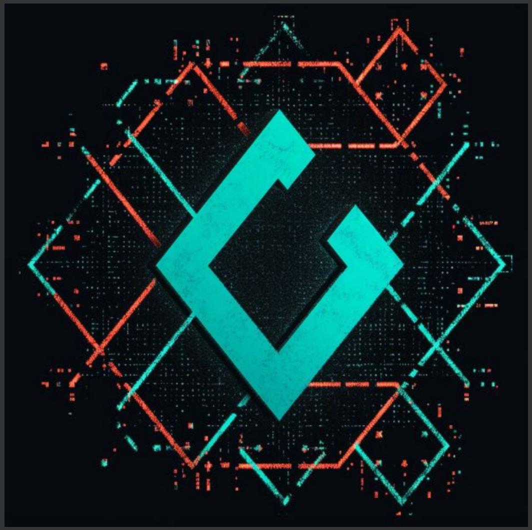
What does the future look like?

What we're focused on...

We might still have Multus...

We might have K8s Native Multi Networking...

We see some potential in...



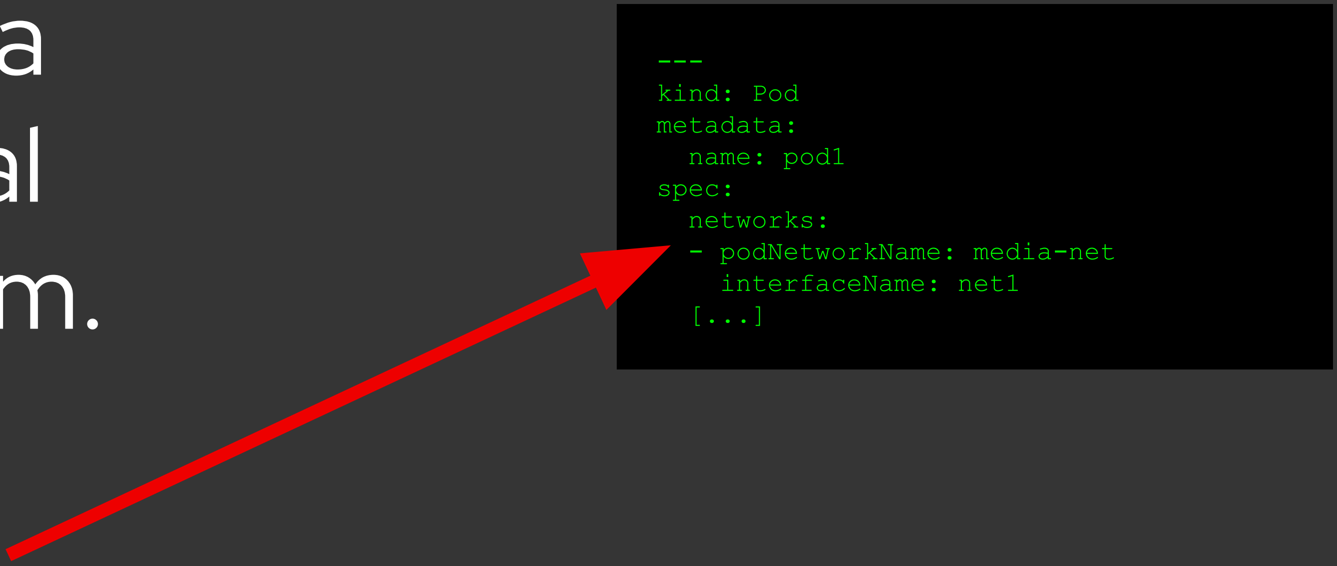
We might get CNI 2.0

We might have KNI: The Kubernetes Network Interface

Lessons Learned

People problems are the hardest ones.

This is a political problem.



```
---
kind: Pod
metadata:
  name: pod1
spec:
  networks:
  - podNetworkName: media-net
    interfaceName: net1
  [...]
```

Yep, we learned a few things.

Lessons Learned

APIs are FOREVER

Scope creep is difficult, especially at scale.

People and political problems are harder than technical ones.

Watch the demo @ Home

Thank you !!! Questions ?