KubeCon CloudNativeCon
North America 2025

The Community-Driven Evolution of the Kubernetes Network Driver

Antonio Ojea
Staff Software Engineer, Google
Lionel Jouin
Software Engineer, Red Hat

#KubeCon #CloudNativeCon



A New Frontier for Networking

Kubernetes is the standard, but a **new wave of demanding workloads** is pushing our networking models to the limit.
The rise of **AI**, **Telco**, and **HPC** requires a more dynamic, intelligent approach.



A New Ecosystem Emerges



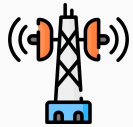
AI/ML

High-performance drivers for AI/ML workloads providing maximum throughput, low latency, and efficient hardware use.



HPC

Drivers for tight node sync, ultra-low-latency, high precision computing, interconnects, and massive parallelism.



Telco

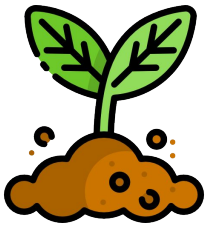
Advanced drivers powering 5G/6G networks with multi-homing, network slicing, QoS, and acceleration.

MPI, RDMA, DPDK, SR-IOV, GPU-DIRECT

The Journey: A Winding Road

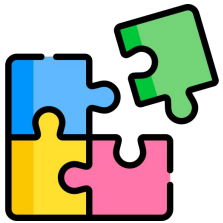
This wasn't a single project, but a **convergence of many ideas, experiments, and lessons from across the ecosystem**.

Lessons from Our Community



Organic Innovation

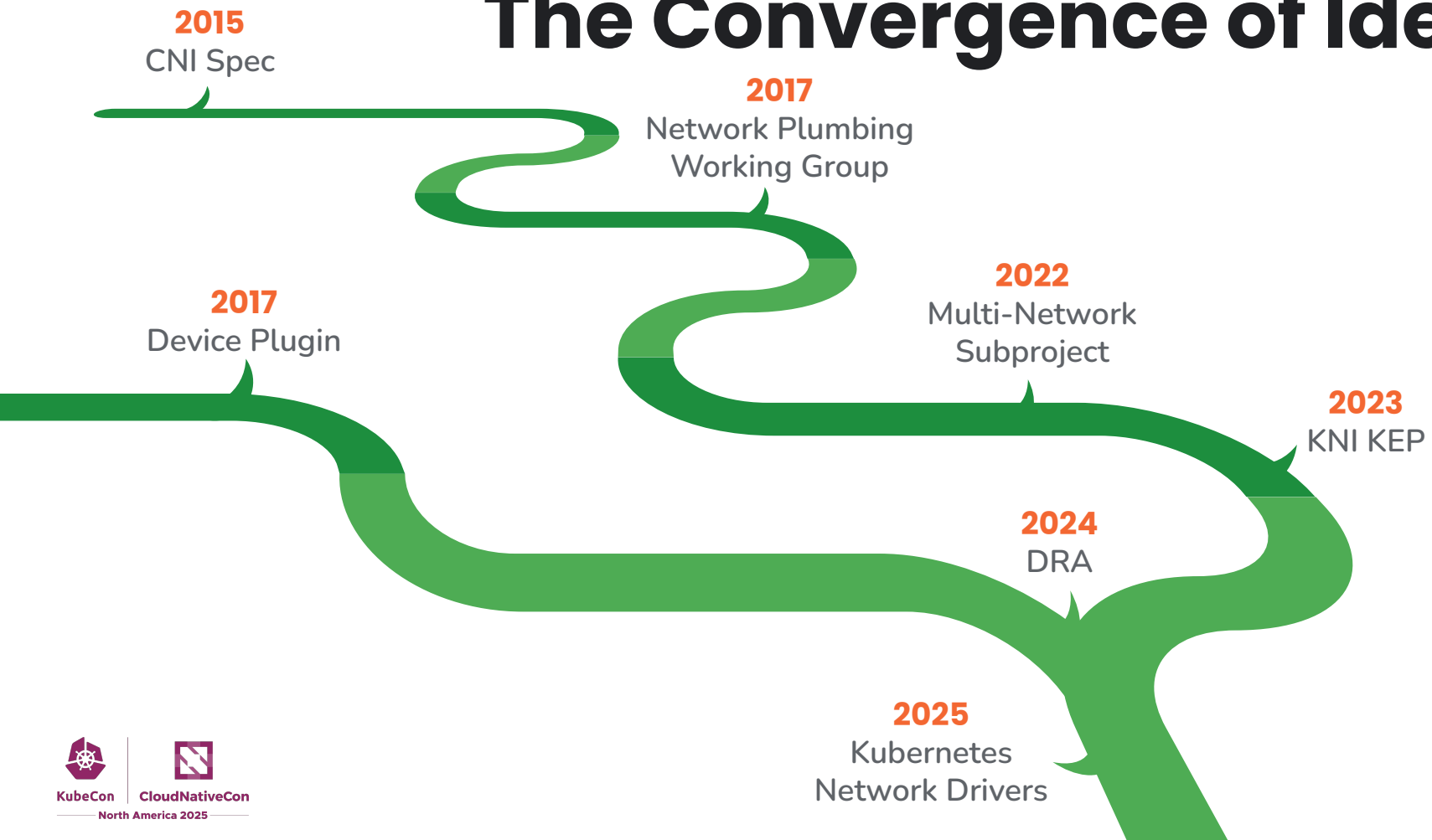
When faced with limits, the community always finds a way. We saw this with annotations and custom controllers—brilliant, but often non-portable solutions that created fragmentation.



A Familiar Pattern

This journey of innovation leading to standardization is one we've seen before. The impact of Kubernetes on containerized applications and standardization of cloud native workloads.

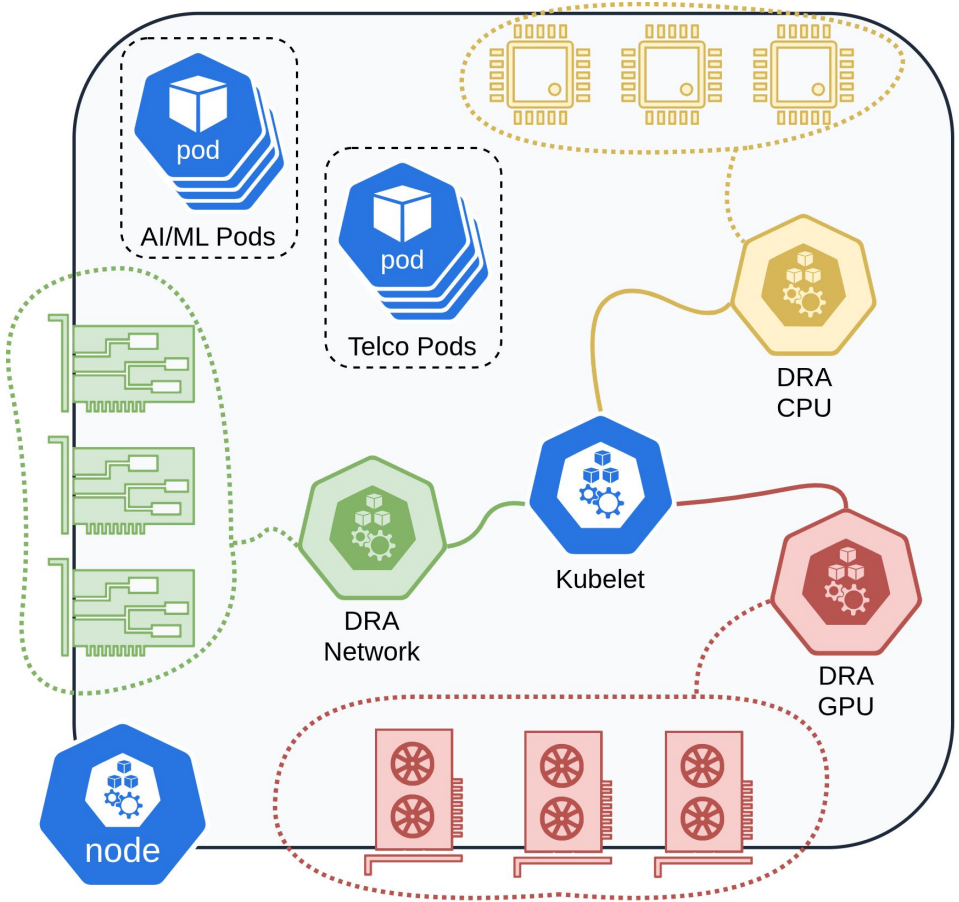
The Convergence of Ideas



A Common Language

DRA

Dynamic Resource Allocation



Kubernetes Network Driver



Problem: The Topology Lottery

- Kubernetes networking is topology-blind.
- Severe performance bottlenecks, and unpredictable results.



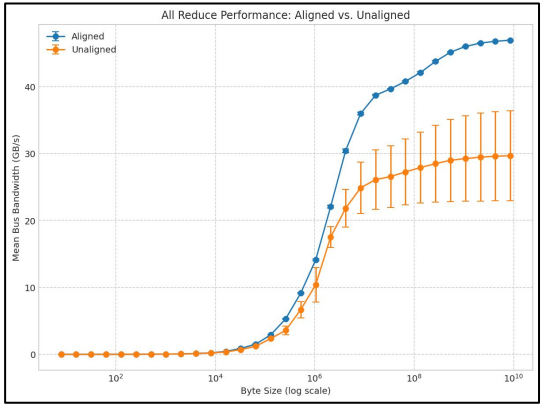
Solution: Topology-Aware Scheduling

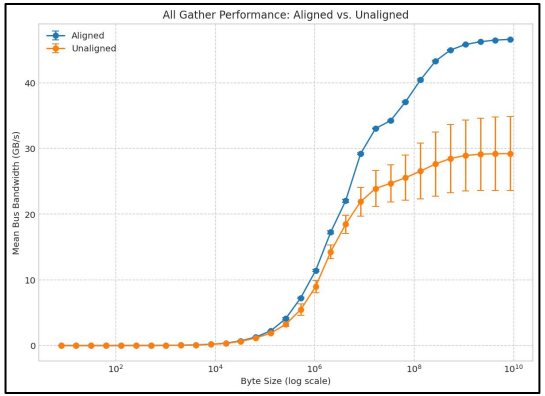
- DRA: For expressive, topology-aware resource claims.
- NRI: For composable runtime hooks that replace fragile plugin chains.



Result: Predictable, Maximum Performance*

- Up to a 59.6% increase in NCCL All Gather Benchmark bandwidth.
- Up to a 58.1% increase in NCCL All Reduce Benchmark bandwidth.
- Eliminated performance variance by solving the 'topology lottery'.





KubeCon CloudNativeCon North America 2025

*The Kubernetes Network Driver Model: A Composable Architecture for High-Performance Networking, Antonio Ojea, 2025 IEEE 50th Conference on Local Computer Networks (LCN)

Live Demo: The Declarative Way

```yaml
# pod.yaml
---
apiVersion: v1
kind: Pod
metadata:
  name: my-ai-pod
spec:
  resourceClaims:
  - name: perf-net
    source:
      resourceClaimTemplateName:
        high-perf-net-claim
```



github.com/LionelJouin/kubecon-na-2025-knd

One More Thing...

Today, we want to celebrate the donation of the first **production-grade**, **AI-focused** Kubernetes Network Driver to the community.

DRANET



Kubernetes Network Driver based on DRA



Topology-Aware Network Resources



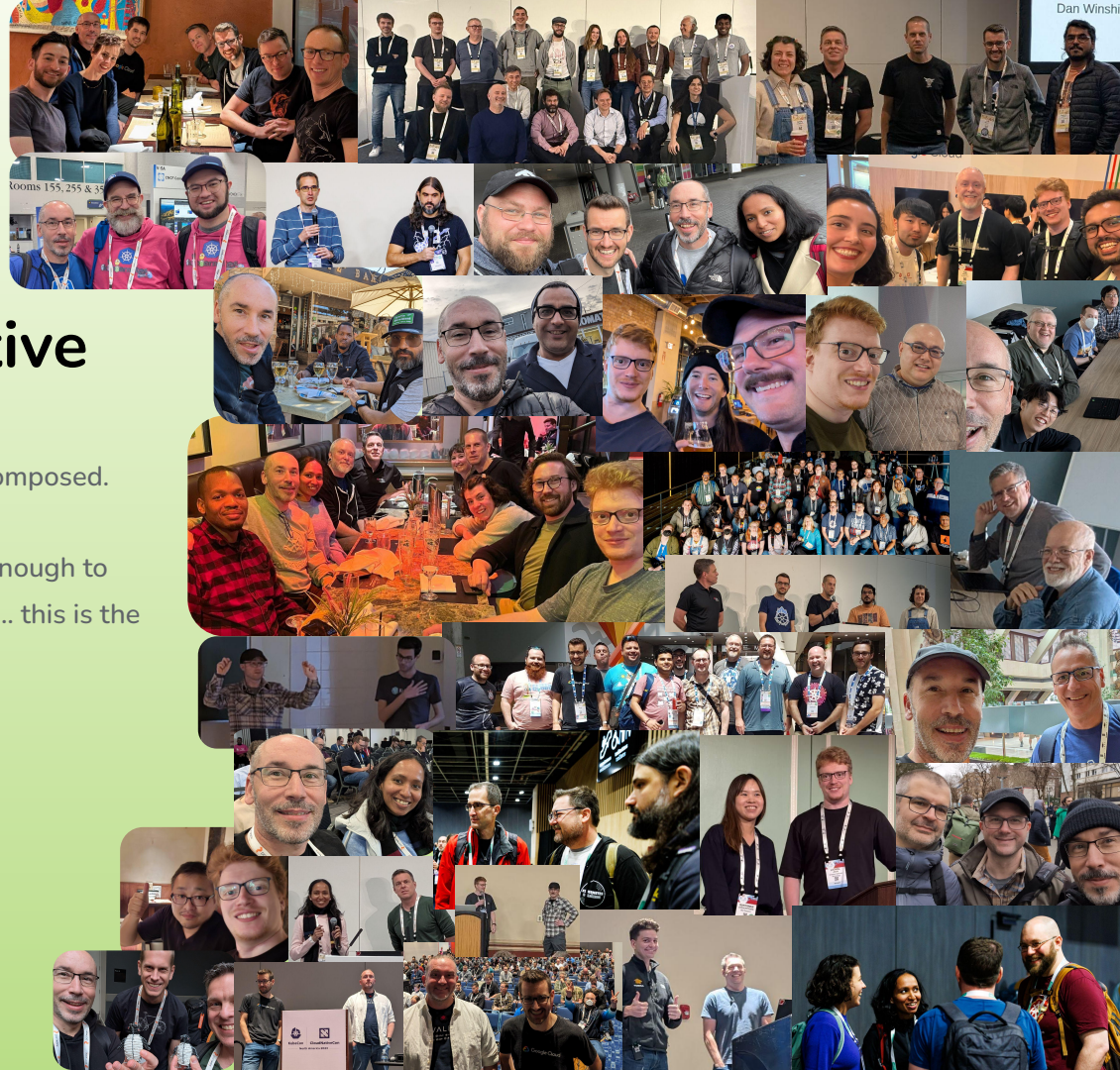
Enhanced Networking for the AI/ML Era

dranet.dev

The Future is Collaborative

The real power isn't in one driver, but in how they can be composed. Imagine your Pod requesting a specific GPU or CPU and a high-performance network. Kubernetes is now *intelligent* enough to **compose** these resources*, optimizing the entire data path ... this is the **next frontier ... a galaxy of drivers**.

*CPU, Memory, GPU, FPGA, Smart NICs, DPU, TPU, Storage...

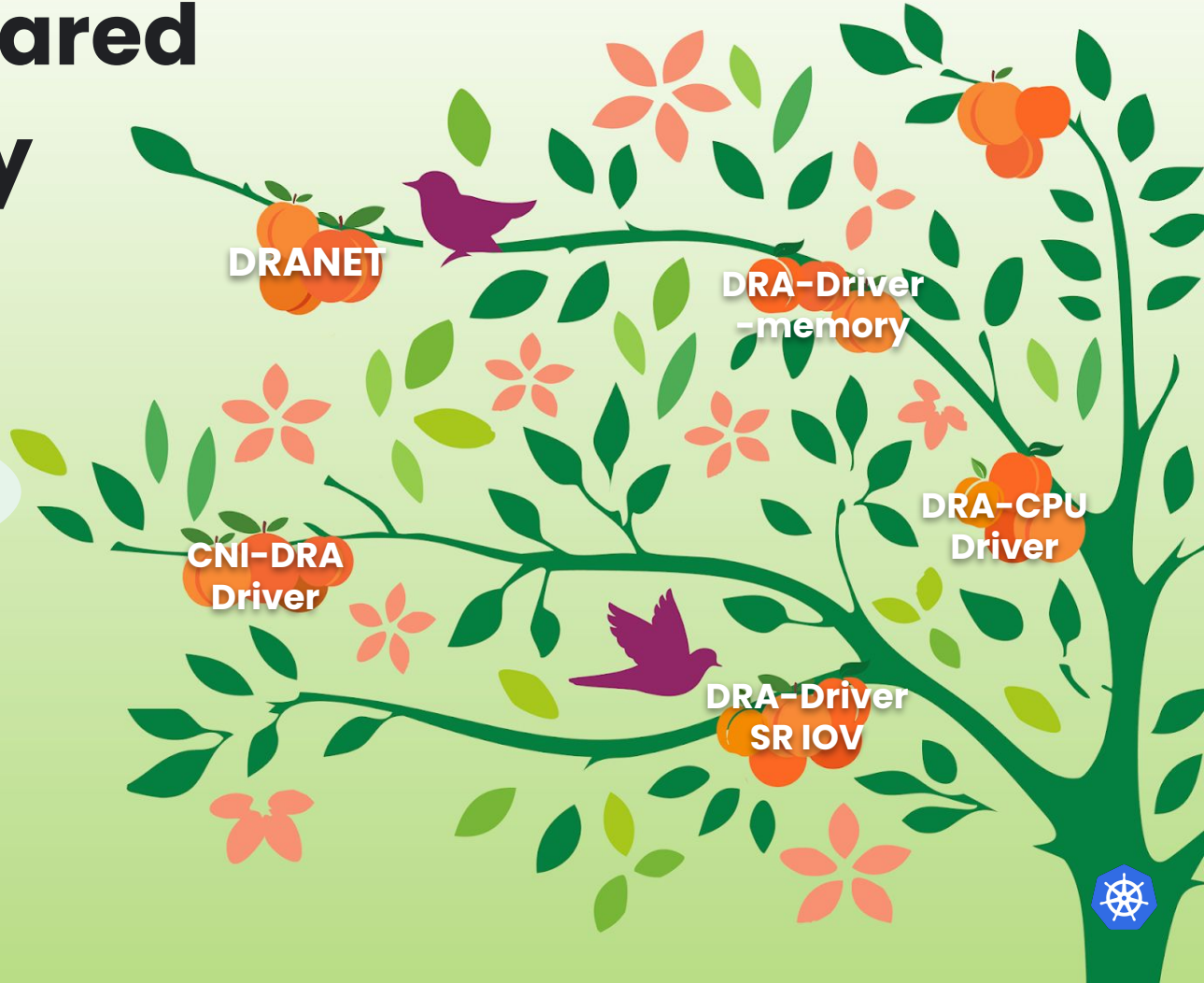


Join Our Shared Journey

SIG-Network

Multi-Network Subproject

WG Device Management



Thank You

Let's build the future, together.



linktr.ee/kubecon.na.2025.knd

```
~$ kubectl get resourceslices
NAME                                                        NODE                DRIVER                                  POOL                AGE
kind-control-plane-kubecon-na-2025-knd.example.com-zjlmg    kind-control-plane  kubecon-na-2025-knd.example.com  kind-control-plane  17s
kind-worker-kubecon-na-2025-knd.example.com-86lgn           kind-worker         kubecon-na-2025-knd.example.com  kind-worker         17s
kind-worker2-kubecon-na-2025-knd.example.com-zcs5b          kind-worker2        kubecon-na-2025-knd.example.com  kind-worker2        17s

~$ kubectl get resourceslices $(kubectl get resourceslices | grep kind-worker- | awk '{print $1}') -o yaml
apiVersion: resource.k8s.io/v1
kind: ResourceSlice
metadata:
  name: kind-worker-kubecon-na-2025-knd.example.com-86lgn
spec:
  devices:
  - attributes:
      kubecon-na-2025-knd.example.com/interfaceName:
        string: dummy0
      kubecon-na-2025-knd.example.com/macAddress:
        string: 3a:73:ad:dd:5a:52
      kubecon-na-2025-knd.example.com/mtu:
        int: 9000
      kubecon-na-2025-knd.example.com/type:
        string: dummy
    name: dummy0
  - attributes:
      kubecon-na-2025-knd.example.com/interfaceName:
        string: dummy1
      kubecon-na-2025-knd.example.com/macAddress:
        string: 7a:c4:c1:c1:4f:b7
      kubecon-na-2025-knd.example.com/mtu:
        int: 1500
      kubecon-na-2025-knd.example.com/type:
        string: dummy
    name: dummy1
  driver: kubecon-na-2025-knd.example.com
  nodeName: kind-worker
  pool:
    generation: 1
    name: kind-worker
    resourceSliceCount: 1
```

https://asciinema.org/a/5pZfcshIYv1zwTlOSJkHmRPvB

```
~$ cat docs/demo/pod.yaml
---
apiVersion: resource.k8s.io/v1
kind: ResourceClaimTemplate
metadata:
  name: high-perf-net-claim
spec:
  spec:
    devices:
      requests:
      - name: high-perf-net-claim
        exactly:
          deviceClassName: dummy
          allocationMode: ExactCount
          count: 1
          selectors:
          - cel:
              expression: device.attributes["kubecon-na-2025-knd.example.com"].mtu >= 9000
---
apiVersion: v1
kind: Pod
metadata:
  name: demo
  labels:
    app: demo
spec:
  containers:
  - name: alpine
    image: alpine:latest
    imagePullPolicy: IfNotPresent
    command:
    - sleep
    - infinity
  resourceClaims:
  - name: high-perf-net-claim
    resourceClaimTemplateName: high-perf-net-claim

~$ kubectl apply -f docs/demo/pod.yaml
resourceclaimtemplate.resource.k8s.io/high-perf-net-claim created
pod/demo created
```

https://asciinema.org/a/5pZfcshIYv1zwTlOSJkHmRPvB

```
~$ kubectl get pods -o wide
NAME                          READY   STATUS    RESTARTS   AGE    IP           NODE                 NOMINATED NODE   READINESS GATES
demo                          1/1     Running   0          7s     10.244.1.2   kind-worker          <none>           <none>
kubecon-na-2025-knd-hfc2p     1/1     Running   0          40s    172.18.0.2   kind-worker2         <none>           <none>
kubecon-na-2025-knd-qmqr2     1/1     Running   0          40s    172.18.0.3   kind-worker          <none>           <none>
kubecon-na-2025-knd-rjzd5     1/1     Running   0          40s    172.18.0.4   kind-control-plane   <none>           <none>
```

https://asciinema.org/a/5pZfcshIYv1zwTlOSJkHmRPvB

```
~$ kubectl get resourceclaim $(kubectl get resourceclaim | grep demo-high-perf-net-claim | awk '{print $1}') -o
yaml
apiVersion: resource.k8s.io/v1
kind: ResourceClaim
metadata:
  annotations:
    resource.kubernetes.io/pod-claim-name: high-perf-net-claim
  creationTimestamp: "2025-11-05T20:18:43Z"
  finalizers:
  - resource.kubernetes.io/delete-protection
  generateName: demo-high-perf-net-claim-
  name: demo-high-perf-net-claim-k8brc
  namespace: default
  ownerReferences:
  - apiVersion: v1
    blockOwnerDeletion: true
    controller: true
    kind: Pod
    name: demo
    uid: 0ff80613-3f71-4a88-9eda-20fcecc9cbf7
  resourceVersion: "723"
  uid: afa0c8a3-b005-42aa-874d-b4454a59e505
spec:
  devices:
    requests:
    - exactly:
        allocationMode: ExactCount
        count: 1
        deviceClassName: dummy
        selectors:
        - cel:
            expression: device.attributes["kubecon-na-2025-knd.example.com"].mtu >=
              9000
      name: high-perf-net-claim
status:
  allocation:
    devices:
      results:
      - device: dummy0
        driver: kubecon-na-2025-knd.example.com
        pool: kind-worker
        request: high-perf-net-claim
    nodeSelector:
      nodeSelectorTerms:
      - matchFields:
        - key: metadata.name
          operator: In
          values:
          - kind-worker
  devices:
  - conditions: null
    device: dummy0
    driver: kubecon-na-2025-knd.example.com
    networkData:
      hardwareAddress: 3a:73:ad:dd:5a:52
      interfaceName: dummy0
      ips:
      - fe80::3873:adff:fedd:5a52/64
    pool: kind-worker
  reservedFor:
  - name: demo
    resource: pods
    uid: 0ff80613-3f71-4a88-9eda-20fcecc9cbf7
```

https://asciinema.org/a/5pZfcshIYv1zwTlOSJkHmRPvB

```
~$ kubectl exec -it demo -- ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: tunl0@NONE: <NOARP> mtu 1480 qdisc noop state DOWN qlen 1000
    link/ipip 0.0.0.0 brd 0.0.0.0
3: eth0@if6: <BROADCAST,MULTICAST,UP,LOWER_UP,M-DOWN> mtu 1500 qdisc noqueue state UP qlen
1000
    link/ether 36:4d:78:03:c8:98 brd ff:ff:ff:ff:ff:ff
    inet 10.244.1.2/24 brd 10.244.1.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fd00:10:244:1::2/64 scope global
       valid_lft forever preferred_lft forever
    inet6 fe80::344d:78ff:fe03:c898/64 scope link
       valid_lft forever preferred_lft forever
4: dummy0: <BROADCAST,NOARP,UP,LOWER_UP> mtu 9000 qdisc noqueue state UNKNOWN qlen 1000
    link/ether 3a:73:ad:dd:5a:52 brd ff:ff:ff:ff:ff:ff
    inet6 fe80::3873:adff:fedd:5a52/64 scope link
       valid_lft forever preferred_lft forever
```

https://asciinema.org/a/5pZfcshIYv1zwTlOSJkHmRPvB