



KubeCon



CloudNativeCon

Europe 2026

#KubeCon #CloudNativeCon

Kubernetes Network Driver Unpacked: Modularity, Trade-offs and the Road Ahead

Sebastian Scheinkman *Principal Software Engineer, Red Hat*

Antonio Ojea *Software Engineer, Google*

Lionel Jouin *Software Engineer, Red Hat*

Sunyanan Choochotkaew *Staff Research Scientist, IBM*





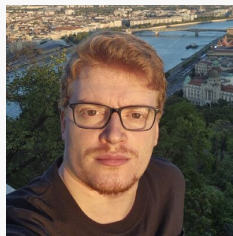
Sebastian Scheinkman



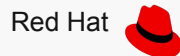
Principal Software Engineer

Github: [SchSeba](#)

Maintainer of the Network Plumbing Working Group, Multus, SR-IOV network operator, SR-IOV Device Plugin, CNI's and DRA-Driver for SR-IOV



Lionel Jouin



Software Engineer

Github: [LionelJouin](#)

CNI Maintainer.
Multi-Network Sub-project Lead.



Antonio Ojea



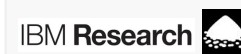
Senior Staff Software Engineer

Github: [aojea](#)

Tech Lead in Kubernetes SIG Network and Testing and member of the Kubernetes Steering Committee



Sunyanan Choochootkaew



Staff Research Scientist

Github: [sunya-ch](#)

Maintainer of Multi-NIC CNI on Vela
DRA Consumable Capacity KEP
DRA driver for virtual net-dev
DRA driver for Spyre



KubeCon



CloudNativeCon

Europe 2026

DRA, Solving Real Challenges



KubeCon

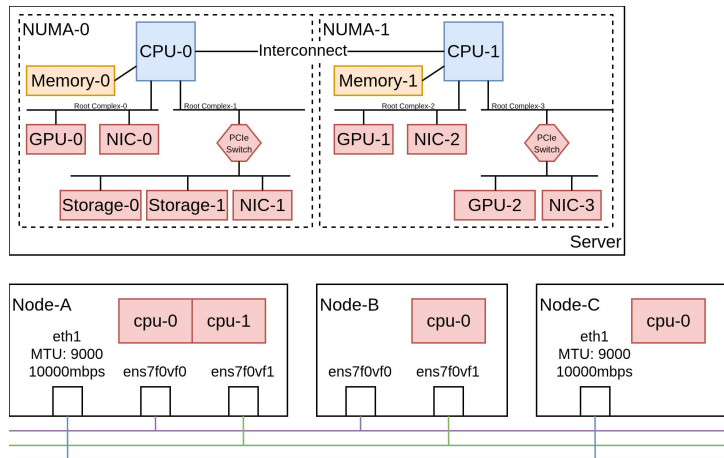


CloudNativeCon

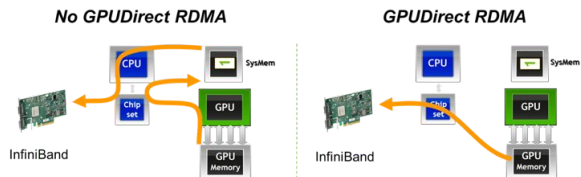
Europe 2026

Topology-aware scheduling

Native Resources resources for Telco use cases



GPUDirect RDMA



Source:
An Introduction to CUDA-Aware MPI
Nvidia Developer: Technical Blog

Expressiveness of Requests

```
selectors:  
- cel:  
  expression:  
    device.attributes[ "resource.kubernetes.io" ][ "pciBusID" ]  
    == "0008:06:00.0"  
- cel:  
  expression:  
    device.attributes[ "dra.net" ][ "rdmaDevice" ] ==  
    "mlx5_2"
```

Modernization of APIs

- Standard Network Attachment
- Closer integration of Scheduling And Configuration
- Validation, Authorisation...

Remove dependency on centralized networking component

Single Driver VS Constellation of Drivers



KubeCon



CloudNativeCon

Europe 2026

Monolith DRA Driver

Security (extra Permissions to cover all cases)

Complexity in discovery / Complexity in Configuration

Concrete example: *CNI-DRA-Driver*

- Requires extra CNI features (e.g. Validation)
- Scheduling not integrated
- Overlapping with CNI feature: Templating / RuntimeConfig
- Wide usage of CNI (Macvlan, OVN-Kubernetes...)
- Adds an extra API Layer

Constellation of DRA Drivers

Overlapping Resource Discovery

Common Features

- IPAM, Routing, Interface configuration...

Granularity of the Driver

- Per feature
 - RDMA, DPDK...
- Per type of device
 - MACVLAN, Host-Device, VF, SF...
- Per type of network
 - Host Device Network, Overlay Network...

Modeling Network using DRA



KubeCon



CloudNativeCon

Europe 2026

KEP-4815 - Partitionable Devices - Beta in v1.36

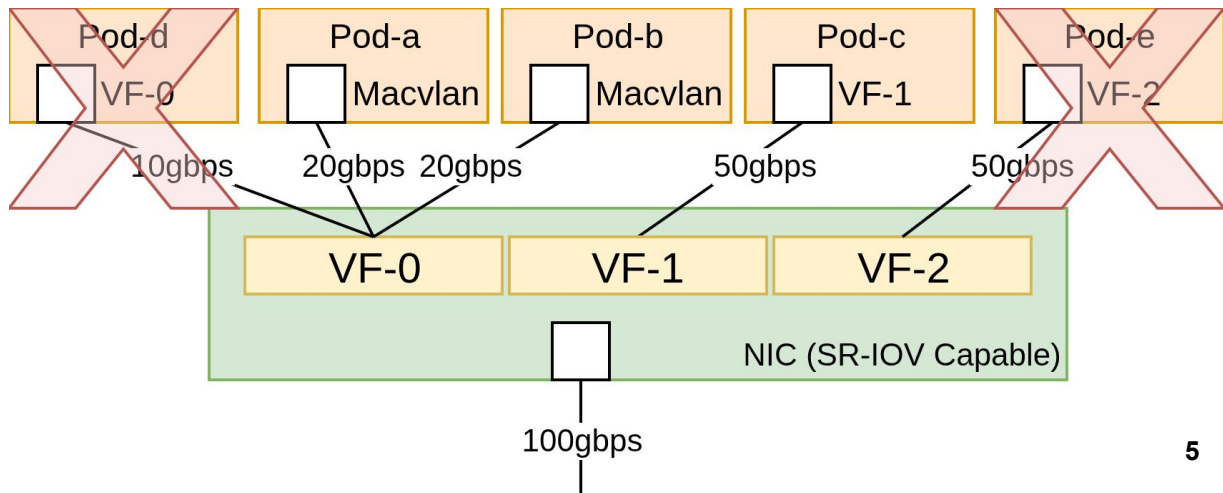
Add support for partitionable devices to DRA

KEP-5075 - Consumable Capacity - Beta in v1.36

Enable DRA to support dynamically provisioned devices

KEP-5941 - Shared Consumable Capacities Across Related Devices - Planned for v1.37

Allow DRA drivers to define parent-scoped shared capacities that are consumed per allocation by related child devices, with scheduler accounting that prevents aggregate over-allocation



Migration and Legacy Solutions



KubeCon



CloudNativeCon

Europe 2026

KEP-5004 - Handle extended resource requests via DRA Driver

Device Plugin to DRA

Multus support for DRA information

Multi-NIC CNI to DRA migration

CNI 2.0 - Fully integrated solution instead of multi-components

- DRANET / SR-IOV (CNIs, DRA, Operator, Multus...)

```
apiVersion: resource.k8s.io/v1
kind: DeviceClass
metadata:
  name: gpu.example.com
spec:
  selectors:
  - cel:
      expression: device.driver == 'dra.example.com' &&
        device.attributes["dra.example.com"].pfName == "eno1"
  extendedResourceName: example.com/port1
---
apiVersion: v1
kind: Pod
metadata:
  name: pod
  namespace: default
  labels:
    app: ext-resource-test
spec:
  terminationGracePeriodSeconds: 2
  containers:
  - name: test
    ...
  resources:
    requests:
      example.com/port1: "1"
    limits:
      example.com/port1: "1"
```

DRA as a First Step, What's Next?



KubeCon



CloudNativeCon

Europe 2026

Multi-Network

KEP-3698 - Multi-Network

```
apiVersion: multinet.networking.k8s.io/v1alpha1
kind: NetworkClass
metadata:
  name: ovn-kubernetes
spec:
  implementationType:
    group: k8s.ovn.org
    version: v1
    kind: UserDefinedNetwork
```

```
// PodNetwork identifies a specific pod network instance.
type PodNetwork struct {
    // Class identifies the NetworkClass responsible for the
    // pod network.
    Class string

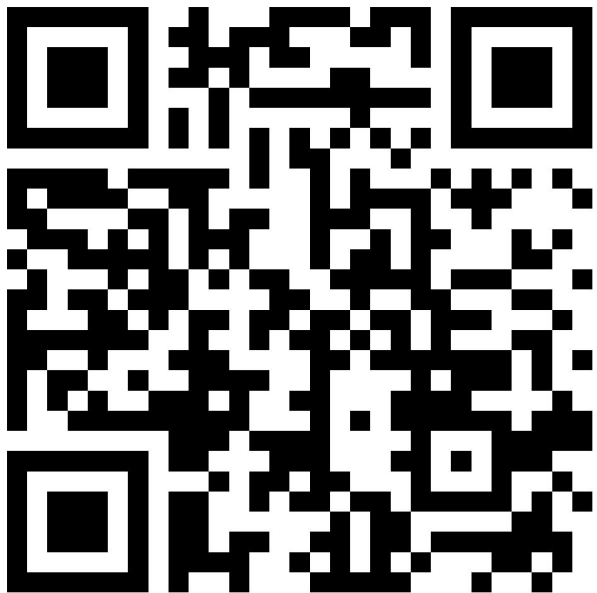
    // Name identifies the pod network object.
    Name string

    // Name identifies the namespace of the pod network object.
    // Optional if the pod network object is a non-namespace
    // object.
    Namespace *string
}
```

Impact of existing functionalities (Existing CNI for primary network)

- How to integrate with all features (DNS, Kube API, Probes, Services, Network Policies, Gateway API...)

How Do I Contribute?



linktr.ee/kubecon.eu.2026.knd

Slack Channels / Meeting Notes / Zoom meeting / Google Group

- SIG-network
- Multi Network
- WG Device Management



KubeCon



CloudNativeCon

Europe 2026

Q&A

