
KubeCon
CloudNativeCon
North America 2024

KubeCon | CloudNativeCon

North America 2024

SIG Network Intro and Update

November 13, 2024

KubeCon | CloudNativeCon

North America 2024

sig-network-policy-api

Nadia Pinaeva (Red Hat); Shaun Crampton (Tigera)

SIG Network Policy API

Sub working group of Kubernetes SIG Network

Work through **N**etwork **P**olicy **E**nhancement **P**roposals (**NPEP**s)

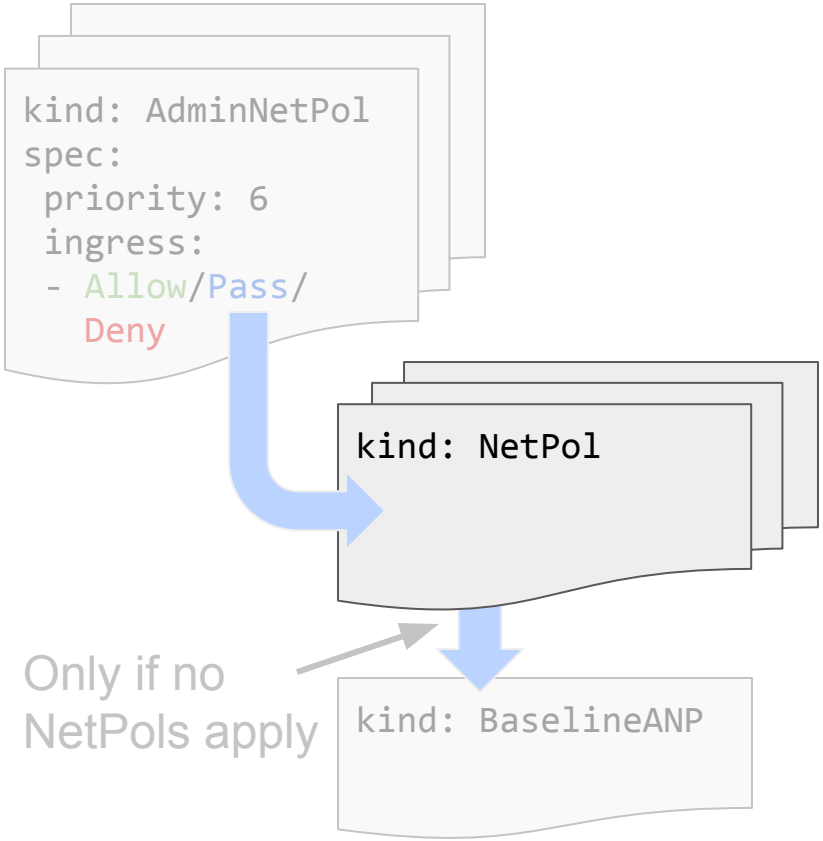
Focus: maintaining/developing network policy APIs & tools:

- 🪨 NetworkPolicy
- 🚀 AdminNetworkPolicy (ANP)
  🚀 BaselineAdminNetworkPolicy (BANP)
  🚧 Tenant isolation
- 🔧 Policy Assistant

ANP and BANP active, Tenant isolation work in progress.

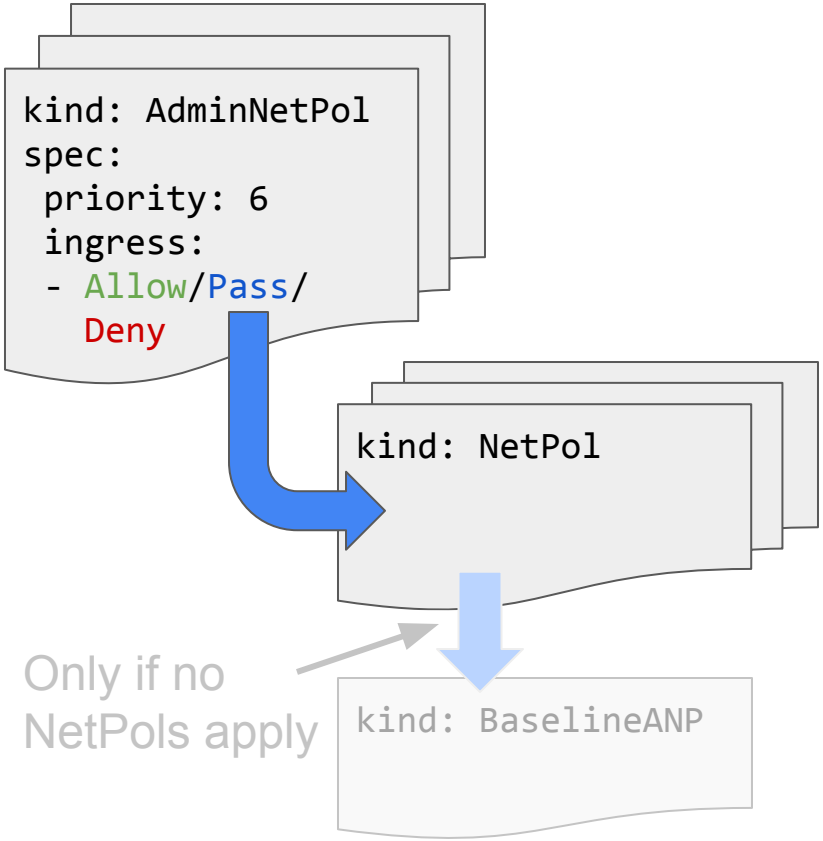
Why new policy APIs?

- NetworkPolicy opt-in and allow-only
- Works OK for developers
- But… cluster admin wants to:
  - Set cluster-wide security posture
  - Enforce strict "guard rails"
    - Prod ← 👮 → staging
  - Lock those down with RBAC
  - Allow "system" traffic at cluster level
  - Isolate "tenants"
  - Own as little policy as possible!



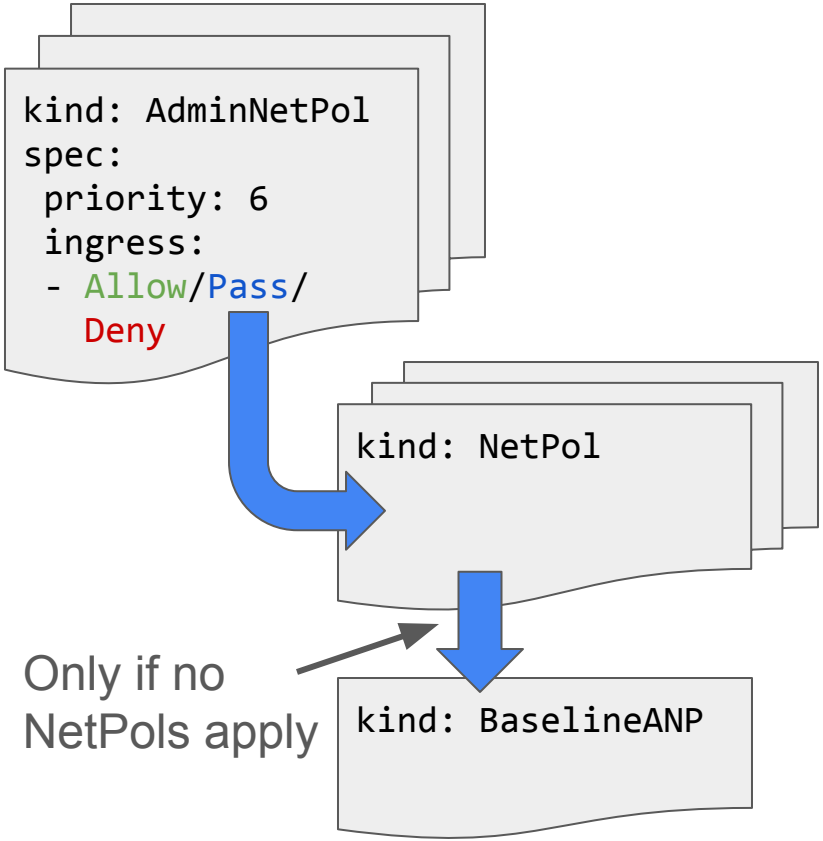
AdminNetworkPolicy

- **AdminNetworkPolicy**
  - Global "tier" ahead of NetworkPolicy
  - **No** implicit deny; no match □ fall through
  - **Explicit** Allow/Pass/Deny
- **BaselineAdminNetworkPolicy**
  - Singleton policy
  - Applies only if no NetPols apply to Pod
    - Replaces "allow by default"
- **Recent NPEPs:**
  - **#122:** 🚧 Tenant isolation (Nadia)
  - **#126:** ✅ CIDR/node matches (Surya)
  - **#133:** ✅ FQDN support (Rahul)
- **New CRD API**:
  - https://network-policy-api.sigs.k8s.io/api-overview/



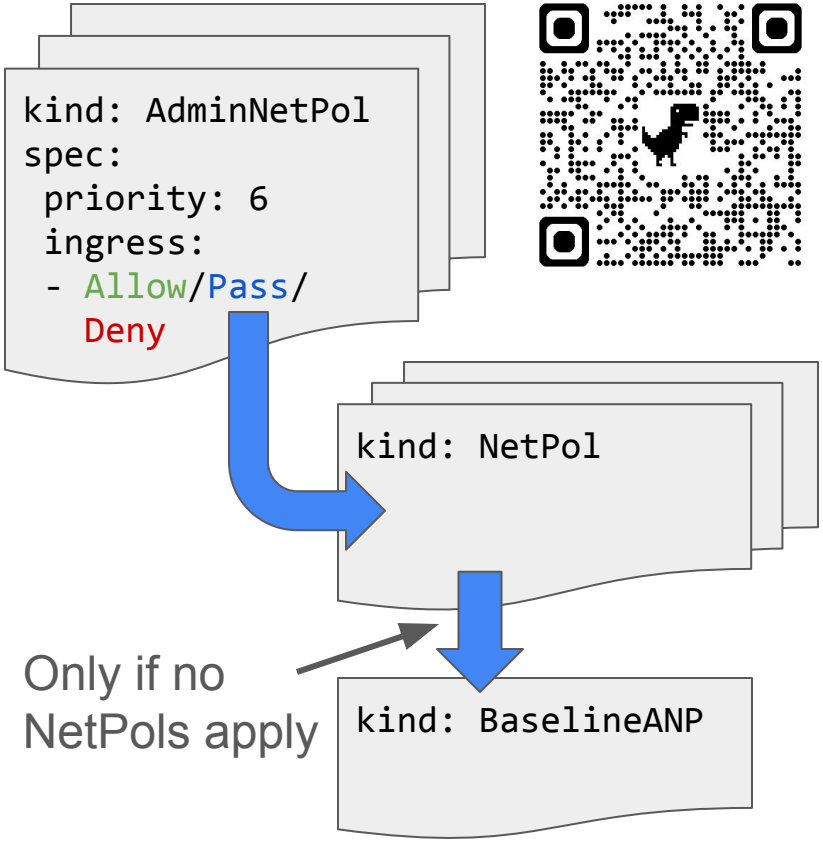
BaselineAdminNetworkPolicy

- **AdminNetworkPolicy**
  - Admin "tier" ahead of NetworkPolicy
  - **No** implicit deny; no match □ fall through
  - **Explicit** Allow/Pass/Deny
- **BaselineAdminNetworkPolicy**
  - Singleton policy
  - Applies only if no NetPols apply to Pod
    - Replaces "allow by default"
- **Recent NPEPs:**
  - **#122:** 🚧 Tenant isolation (Nadia)
  - **#126:** ✅ CIDR/node matches (Surya)
  - **#133:** ✅ FQDN support (Rahul)
- **New CRD API**:
  - https://network-policy-api.sigs.k8s.io/api-overview/



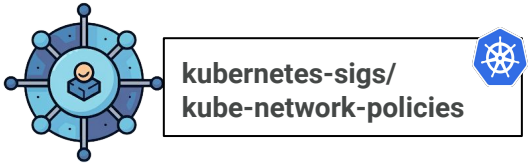
Recent work

- **AdminNetworkPolicy**
  - Admin "tier" ahead of NetworkPolicy
  - **No** implicit deny; no match □ fall through
  - **Explicit** Allow/Pass/Deny
- **BaselineAdminNetworkPolicy**
  - Singleton policy
  - Applies only if no NetPols apply to Pod
    - Replaces "allow by default"
- **Recent NPEPs:**
  - **#122:** 🚧 Tenant isolation (Nadia)
  - **#126:** ✅ CIDR/node matches (Surya)
  - **#133:** ✅ FQDN support (Rahul)
- **New CRD API**:
  - https://network-policy-api.sigs.k8s.io/api-overview/



AdminNetworkPolicy API: Impl. status

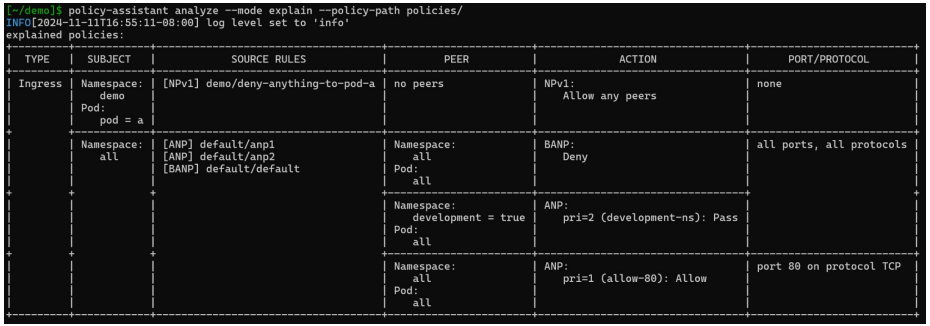
- Closing in on beta API level…
- Five implementations so far; **new:** kube-network-policies, Calico, Kube-OVN



- On roadmap:

1. Cilium tracker: https://github.com/cilium/cilium/issues/23380

Policy Assistant



- Policy simulation tool
- Can get policy/pods/etc from
  - Cluster
  - Local files
- Recent highlights:
  - Read policy from cluster or files
  - JSON input for traffic specs
  - Walkthrough mode completed
    https://github.com/kubernetes-sigs/network-policy-api/blob/main/cmd/policy-assistant/examples/demos/walkthrough/README.md
  - Release!
    https://github.com/kubernetes-sigs/network-policy-api/releases/tag/v0.0.1-policy-assistant
- Shout outs to contributors:
  - Hunter https://github.com/huntergregory
  - Gabriel https://github.com/gabrielggg
  - Nikola https://github.com/Peac36

Policy Assistant: examples (1)

Analysing TCP port 80 between two workloads

```
[~/demo]$ kubectl get deployments,pods -n demo
NAME                READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/a   1/1     1            1           6h4m

NAME                     READY   STATUS    RESTARTS   AGE
pod/a-6b45b6bccc-z89mn   2/2     Running   0          6h4m
pod/b                    2/2     Running   0          17h
[~/demo]$
[~/demo]$ policy-assistant analyze --mode walkthrough --policy-path policies/ --src-workload demo/deployment/a --dst-workload demo/pod/b --protocol TCP --port 80
INFO[2024-11-11T16:53:30-08:00] log level set to 'info'
verdict walkthrough:
+----------------------------------------+---------+----------------------+-----------------------------+
|                TRAFFIC                 | VERDICT |  INGRESS WALKTHROUGH |      EGRESS WALKTHROUGH     |
+----------------------------------------+---------+----------------------+-----------------------------+
| demo/deployment/a -> demo/pod/b:80 (TCP) | Allowed | [ANP] Allow (allow-80) | no policies targeting egress |
+----------------------------------------+---------+----------------------+-----------------------------+
```

Specifying multiple traffic paths with JSON

```
[~/demo]$ policy-assistant analyze --mode walkthrough --policy-path policies/ --traffic-path traffic.json
INFO[2024-11-11T16:54:31-08:00] log level set to 'info'
verdict walkthrough:
+-----------------------------------------------+---------+-----------------------------------------------------------------------------+-----------------------------+
|                    TRAFFIC                    | VERDICT |                          INGRESS WALKTHROUGH                                |      EGRESS WALKTHROUGH     |
+-----------------------------------------------+---------+-----------------------------------------------------------------------------+-----------------------------+
| demo/deployment/a -> demo/pod/b:80 (TCP)      | Allowed | [ANP] Allow (allow-80)                                                      | no policies targeting egress |
+-----------------------------------------------+---------+-----------------------------------------------------------------------------+                             +
| demo/deployment/a -> demo/pod/b:81 (TCP)      | Denied  | [ANP] No-Op -> [BANP] Deny (baseline-deny)                                  |                             |
+-----------------------------------------------+         +-----------------------------------------------------------------------------+                             +
| demo2/[app=nginx] -> demo/deployment/a:81 (TCP) |       | [ANP] Pass (development-ns) -> [NPv1] Dropped (demo/deny-anything-to-pod-a) |                             |
+-----------------------------------------------+---------+-----------------------------------------------------------------------------+-----------------------------+
```

Policy Assistant: examples (2)

Full "explain" for a set of policies

```
[~/demo]$ policy-assistant analyze --mode explain --policy-path policies/
INFO[2024-11-11T16:55:11-08:00] log level set to 'info'
explained policies:
+---------+------------+-----------------------------------+----------------------+-------------------------------+--------------------------+
|  TYPE   |  SUBJECT   |           SOURCE RULES            |         PEER         |            ACTION             |      PORT/PROTOCOL       |
+---------+------------+-----------------------------------+----------------------+-------------------------------+--------------------------+
| Ingress | Namespace: | [NPv1] demo/deny-anything-to-pod-a | no peers             | NPv1:                         | none                     |
|         |    demo    |                                   |                      |    Allow any peers            |                          |
|         | Pod:       |                                   |                      |                               |                          |
|         |    pod = a |                                   |                      |                               |                          |
+         +------------+-----------------------------------+----------------------+-------------------------------+--------------------------+
|         | Namespace: | [ANP] default/anp1                | Namespace:           | BANP:                         | all ports, all protocols |
|         |    all     | [ANP] default/anp2                |    all               |    Deny                       |                          |
|         |            | [BANP] default/default            | Pod:                 |                               |                          |
|         |            |                                   |    all               |                               |                          |
+         +            +                                   +----------------------+-------------------------------+                          +
|         |            |                                   | Namespace:           | ANP:                          |                          |
|         |            |                                   |    development = true|    pri=2 (development-ns): Pass|                         |
|         |            |                                   | Pod:                 |                               |                          |
|         |            |                                   |    all               |                               |                          |
+         +            +                                   +----------------------+-------------------------------+--------------------------+
|         |            |                                   | Namespace:           | ANP:                          | port 80 on protocol TCP  |
|         |            |                                   |    all               |    pri=1 (allow-80): Allow    |                          |
|         |            |                                   | Pod:                 |                               |                          |
|         |            |                                   |    all               |                               |                          |
+---------+------------+-----------------------------------+----------------------+-------------------------------+--------------------------+
```

KubeCon | CloudNativeCon

North America 2024

kube-proxy

Dan Winship (Redhat); Antonio Ojea (Google); Daman Arora (Broadcom)

kube-proxy nftables backend

- Originally alpha in 1.29, now beta in 1.31 (expected GA in 1.33)
- Mostly compatible with iptables (but better!)
  - Requires Linux 5.13+ (e.g. a mid-2021-or-later distro)
- Several bugfixes, performance improvements, and minor features since alpha.
  - Fixes to LoadBalancer handling and kube-proxy startup (Quan Tian)
  - Improved handling of traffic to unused service IPs/ports (Daman Arora)
  - NodePorts accepted on primary IPs only (`@nayihz`)
  - Partial/incremental sync support (Nadia Pinaeva)

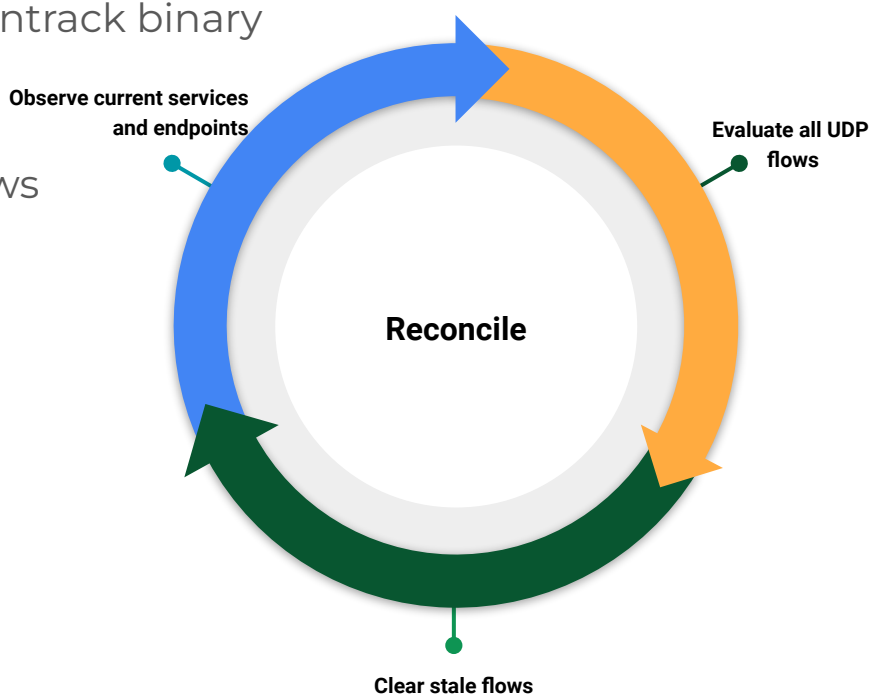
kube-proxy nftables backend

- Drops some “bad” iptables features, so not 100% compatible
  - In 1.31 the iptables mode exposes some metrics to help you figure out if you are using iptables-specific features:
    - `kubeproxy_iptables_localhost_nodeports_accepted_packets_total` indicates if you are using localhost NodePorts
    - `kubeproxy_iptables_ct_state_invalid_dropped_packets_total` indicates if you need to set “`conntrack.tcpBeLiberal`”

- Talk tomorrow at 2:30, “How the Tables Have Turned: Kubernetes Says Goodbye to IPTables”

Connection Tracking Improvements

- Netlink for interaction with kernel
  - Removes dependency on user-space conntrack binary
  - Improves performance
    - No process forking
    - Single dump call to clear all stale flows
- Conntrack Reconciler
  - Existing Problems
    - Cleanup is a best-effort
    - Cleanup is edge-triggered
    - Cleanup logic is complex



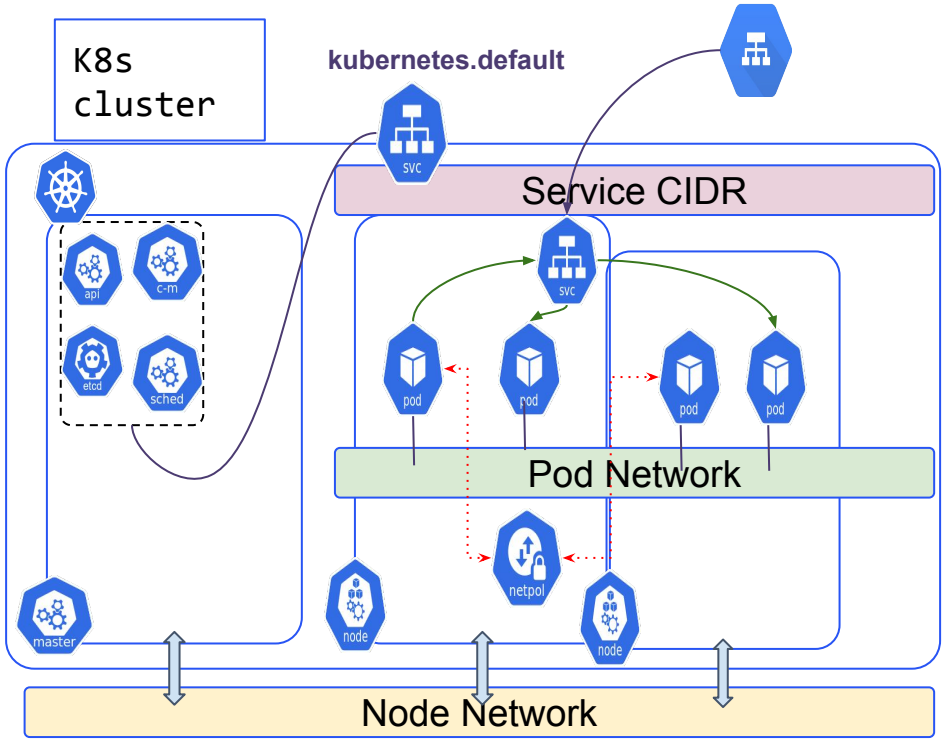
KubeCon | CloudNativeCon

North America 2024

Multiple Service CIDRs

Antonio Ojea (Google)

Multiple Service CIDRs (beta v1.31)



- Service Network was defined as a kube-apiserver flag and limited because of the implementation (etcd bitmap
- Services ClusterIPs are immutable once set*
- Multiple apiservers must agree on the service CIDR range
- Adding new ServiceCIDRs allow users to expand the existing range assigned to Services

KubeCon

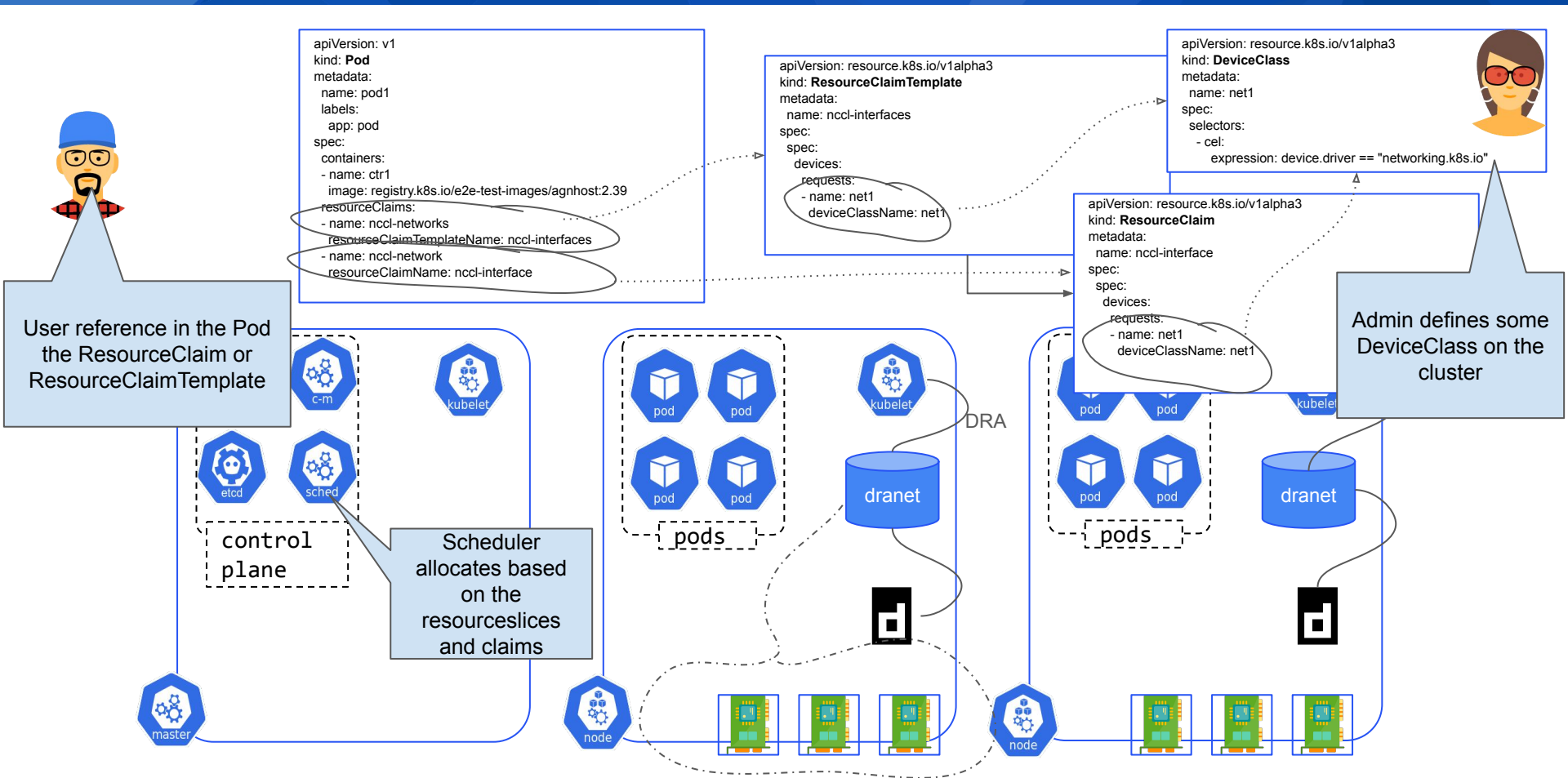
CloudNativeCon

North America 2024

Kubernetes Network Drivers: DRA

Antonio Ojea (Google)

Networking High Level API: DRA



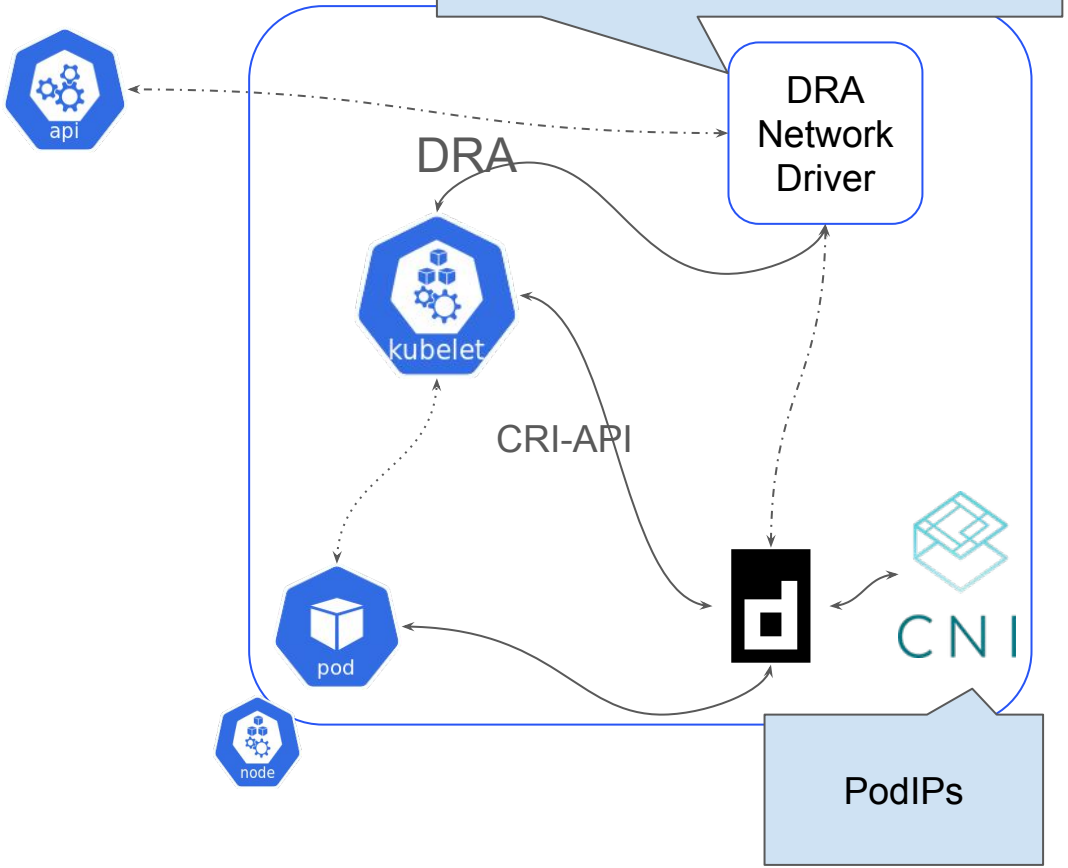
Kubernetes Network Drivers

- Secondary Network Interfaces
- Additional network functionality
- …

Kubernetes Network Drivers use **DRA** to expose Networks resources at the Node level that can be referenced by all the **Pod** (or all containers)

The network driver, before the **RunPodSandbox** is called, receives the **NodePrepareResources** rpc with the **Devices** and **Configuration** to use with the **Container Runtime**.

This allows to keep **backwards compatibility** and **expand** the existing network plugins.



KubeCon | CloudNativeCon

North America 2024

Questions?

Please leave feedback on sched:

KubeCon | CloudNativeCon

North America 2024