KubeCon CloudNativeCon North America 2026

#KubeCon #CloudNativeCon

SIG Network Intro and Updates

Bowei Du · Guilherme Cassolato · Surya Seetharaman · Austin Abro



Who we are

KubeCon CloudNativeCon North America 2026



Bowei Du

@bowei
Google



Guilherme Cassolato

@guicassolato
Red Hat



Surya Seetharaman

@tssurya
Red Hat



Austin Abro

@AustinAbro321
Defense Unicorns

Agenda

1. **What is SIG Network?** Scope, people and subprojects
2. **Kubernetes core** KEP graduations, deprecations and kube-proxy
3. **What's new in the subprojects** Highlights from across the SIG
4. **Subproject Status Survey 2026** What we learned and what we're doing about it
5. **Becoming a contributor** Using AI to become a new k8s contributor
6. **Get involved** Where to find us, how to contribute

KubeCon CloudNativeCon
North America 2026

What is SIG Network?



What SIG Network owns

Services & kube-proxy

Service, EndpointSlice, load balancing, iptables / nftables / IPVS

Network Policy

NetworkPolicy and ClusterNetworkPolicy

DNS

Cluster DNS spec, kube-dns, node-local-dns, external-dns

Ingress & Gateway API

North-south and east-west traffic routing APIs, incl. AI inference

Pod networking & IPAM

Dual-stack, IP allocation, node IPAM, multi-network

Network drivers & DRA

DRANET, CNI-DRA driver, device-backed networking

Who runs the SIG

Chairs

Run operations and processes

Bowei Du
@bowei · Google

Guilherme Cassolato
@guicassolato · Red Hat

Michael Zappa
@mikezappa87 · Microsoft

Technical Leads

Technical direction, subproject lifecycle

Antonio Ojea
@aojea · Google

Dan Winship
@danwinship · Red Hat

Tim Hockin
@thockin · Google

Subproject leads

Projects with formal leads

Gateway API
Ricardo Katz · Rob Scott · Nick Young

Network Policy
Dan Winship · Nadia Pinaeva · Surya Seetharaman

Multi-Network
Lionel Jouin · Maciej Skrocki

22 subprojects at a glance

3 Growing / In Progress	7 Healthy & stable	6 Maintenance mode	4 Winding down	2 Follow-up needed
Gateway API	Network Policy	knftables	ingress-nginx (archived)	ip-masq-agent
DRANET	Gateway API Inference Extension	iptables-wrappers	kubernetes-network-drivers	ingress-controller-conformance
Kube Agentic Networking (early-stage)	External DNS	node-ipam-controller	cluster-proportional-autoscaler	
Multi-Network	kindnet	nat64	cluster-proportional-vertical-autoscaler	
	ingress-gce	node-local-dns		
	cni-dra-driver	kube-dns (deprecated)		

Health as self-reported by maintainers in the 2026 subproject survey


KubeCon
CloudNativeCon
North America 2026
Kubernetes core

KEP graduations & deprecations

Graduated to GA

KEP-4762 Pod hostname override (FQDN)
GA in v1.37

KEP-5311 Relaxed Service name validation
GA in v1.37

KEP-4858 Strict IP / CIDR validation
GA in v1.38*

New & promoted

KEP-6032 nftables localhost NodePorts
Alpha v1.37 → Beta, on by default, v1.38*

KEP-6313 Network-isolated (hermetic) Pods
Alpha targeted for v1.38*

KEP-5343 nftables as default kube-proxy mode
Accepted; default flips in a later release

Deprecated

KEP-5707 Service.spec.externalIPs
Deprecated in v1.36

KEP-5495 kube-proxy ipvs mode
KubeProxyIPVS gate deprecated in v1.37

KEP-4974 core/v1 Endpoints
apiserver service proxy moved to EndpointSlice

Retired feature gates removed: TopologyAwareHints, ServiceTrafficDistribution, PreferSameTrafficDistribution, MultiCIDRServiceAllocator, RelaxedDNSSearchValidation. ** v1.38 is still in development*

kube-proxy nftables & localhost NodePorts

iptables mode

- NodePorts reachable on 127.0.0.1
- Needs route_localnet=1 on the node
- Long-standing security footgun

nftables mode

- GA since v1.33
- No localhost NodePorts out of the box
- Planned to become the default (KEP-5343)

What's changing

- Localhost NodePort functionality is opt-in with nftables
- Kube metrics track if the cluster is reliant on localhost NodePorts
- Configure kube-proxy with localhost in NodePort addresses to migrate



KEP-6032: Alpha in v1.37 → Beta, on by default, in v1.38
Under the hood: knftables v0.0.22 reads rules via netlink (NFTablesNetlink gate, on by default since v1.37)

KubeCon CloudNativeCon
North America 2026

What's new in the subprojects



Gateway API: three releases in 2026

v1.5 February 2026

- **Standard:** TLSRoute, ListenerSet, CORS filter, frontend mTLS, backend client certs
- **v1:** ReferenceGrant

v1.6 June 2026

- **Standard:** TCPRoute, UDPRoute
- **Experimental:** XBackend (v1alpha1)
- Controller Matching Wizard, driven by conformance reports

v1.7 October 2026

- **Standard:** Retries, route rule-name uniqueness, BackendTLSPolicy SAN / TLS options
- **Experimental:** XTelemetryPolicy + OTel attributes, listener filters, address routability, ClusterTrustBundle

Decomposing Service: XBackend + EndpointSelector (KEP-6116, proposed) — a joint effort with core SIG Network

Ingress, inference & agents

ingress2gateway

v1.0 → v1.2

- v1.0 in March, days before ingress-nginx was archived
- Providers split from Emitters; new Traefik provider
- Emitters for Envoy Gateway, kgateway, agentgateway, Airlock

Inference Extension

v1.4 → v1.6

- InferencePool v1; v1alpha2 deprecated in v1.5
- v1.6 (Aug): EPP and BBR moved to llm-d
- Lightweight reference EPP (lwepp) stays in-repo

Kube Agentic Networking

v0.1 → v0.2

- v0.1 (Jun): v1alpha1 XAccessPolicy — MCP tool authorization, CEL, ExternalAuth
- v0.2 (Sep): MCP parameter-level authorization, OTel decision tracing

Deep dive: Gateway API & Kube Agentic Networking talk — Wed, Nov 11, 3:10–3:40 pm · Salt Palace, Level 2, 250 D-F

Network Policy: subproject updates

API releases

v0.2.0 · Apr 2026

- ClusterNetworkPolicy v1alpha2 — the first CNP release
- Unifies AdminNetworkPolicy and BaselineAdminNetworkPolicy into Admin and Baseline tiers
- v0.2.1 patch release in October

kube-network-policies

v1.1.0 → v1.1.3

- Reference implementation of the API
- Full ClusterNetworkPolicy v1alpha2 support
- Pod-to-Node egress policies; non-disruptive nftables updates

NPEP process

Working well

- Enhancements proposed as NPEPs
- “Even new contributors can file them”
- Close collaboration with Gateway API

Healthy and stable. Main pain point: getting API-review time slots — one of the survey’s action items

ClusterNetworkPolicy v1beta1

One cluster-scoped API for admins

1. **Admin tier**
   evaluated before namespace NetworkPolicy — Accept / Deny / Pass
2. **NetworkPolicy**
   owned by app teams, per namespace
3. **Baseline tier**
   cluster defaults, evaluated after NetworkPolicy

Road to beta

v1alpha1
AdminNetworkPolicy + BaselineAdminNetworkPolicy

v1alpha2
ClusterNetworkPolicy, released in v0.2.0 (April 2026)

v1beta1 — next
ClusterNetworkPolicy

Getting ready for v1beta1

- **FQDN egress** (NPEP-133) limited to the Admin tier
- **NamedPort** and **Nodes** support
- **Hairpin traffic:** capability model + conformance

Across the subprojects



DRANET

Five releases in 2026 (v1.1 → v1.5): IPVLAN subinterfaces, Whereabouts IPAM, VRF, RDMA / InfiniBand across 6 clouds



Multi-Network

Core PodNetwork API proposal being re-aligned with DRANET; reference implementation prototyped on kindnet



External DNS

v0.21 → v0.24 (Oct): Gateway API v1 sources, ListenerSet, FIPS images; heading to v1.0



kindnet

Donated to kubernetes-sigs; the reference CNI running across Kubernetes CI



cni-dra-driver

Exploring how CNI fits into DRA; helped graduate KEP-4817 to GA



ingress-gce

Splitting the Ingress, L4 and NEG controllers into distinct codebases

Quiet plumbing & lifecycle changes

Stable, thinly staffed

knftables netlink support landed in v0.0.22

iptables-wrappers underpins distroless-iptables image

node-ipam-controller feature complete (KEP-2593)

nat64 stable, scope is right

node-local-dns standalone module; 1.27.x shipped in Sept.

Lifecycle



ingress-nginx
Archived March 24, 2026, with SIG support; migrate with ingress2gateway



kube-dns
Deprecated; CVE fixes possible through v1.40; use CoreDNS



kubernetes-network-drivers
Archive candidate; work consolidating into DRANET



CPA / CPVA
Proposed move to SIG Autoscaling

KubeCon CloudNativeCon
North America 2026

Subproject Status Survey 2026



Why we ran a subproject survey

We surveyed all 22 SIG Network subprojects (July–Aug 2026) to understand each project's real state, meet the people maintaining them, and see where the SIG helps — or should.

- One pre-filled form per project
- Distributed by Slack DM + email
- ~10–15 min per maintainer

22
subprojects surveyed

32
responses

18/22
projects responded

77%
with substantive answers

What we heard: healthy, but rebalancing



Flagships thriving
Gateway API, Network Policy, Multi-Network, Inference Extension, DRANET all active and shipping



AI/ML is the new center of gravity
and it's creating overlap between projects racing to serve it



Thin maintainership
many utility projects run on 1–2 people; library projects unsure what the SIG is for them



Winding down
ingress-nginx archived, kube-dns deprecated, kubernetes-network-drivers → archive, CPA/CPVA → maybe SIG Autoscaling



Two recurring asks
faster API review and better SIG-wide communication

4.2

/ 5

average rating of SIG Network support

In maintainers' own words

"SIG Network supported us with retiring the project and making decisions or coming up with communications we wouldn't have been able to on our own."

— ingress-nginx

"The fact nobody even knows it runs on all the CI systems and does not have weird network bugs is the best metric."

— kindnet

"The project gets no support at all from SIG Network… [or] is 100% supported by SIG Network, via me acting as SIG Network's representative. I'm not sure which is correct."

— knftables

"K8s does not like to be experimented on… providing unreasonable standards for this level of APIs."

— Multi-Network, on API review

What we're doing about it

1. **AI-networking coordination** Gateway API · Inference Extension · Kube Agentic Networking · WG AI Gateway
2. **API-review throughput** especially for experimental APIs
3. **Lifecycle cleanup** archive, deprecate or relocate the tail
4. **Tackle the bus factor** recruit maintainers for at-risk projects
5. **Close the communication gap** publicize the SIG meeting, rotate project updates

Full report: sig-network/survey-2026/report.md

KubeCon CloudNativeCon
North America 2026

Becoming a contributor



Using AI to become a new k8s contributor

Where I started

- I maintain Zarf, which relies heavily on localhost NodePorts
- Found an aligned issue, spoke with maintainers, followed up in the sig-network call.

How AI helped

- Understanding Transactions, proxy lifecycle, networking, test patterns
- Quick implementation to discover gaps in the KEP

Lessons & guardrails

- Challenge every decision, filter decisions to maintainers when unsure
- Breaking changes are complex
- KEPs add time, but improved the implementation massively

Why it matters here: many subprojects run on 1–2 maintainers. Lowering the barrier for new contributors is how we fix the bus factor.

Get involved

Join us on Slack

#sig-network

kubernetes.slack.com · scan the code →

SIG meeting Thursdays at 09:00 PT, every two weeks

Mailing list groups.google.com/a/kubernetes.io/g/sig-network

Community page git.k8s.io/community/sig-network

Maintainers wanted: node-local-dns · Kube Agentic Networking · External DNS

kubernetes.slack.com/archives/C09QYUH5W

KubeCon CloudNativeCon North America 2026

Thank you! Questions?

