Kubernetes plans to remove kube-proxy's IPVS mode: how to prepare your clusters

18 November 2026 · 5 min read · blogs.iictp.org/kubernetes-kube-proxy-ipvs-removal-nftables

In a post on the Kubernetes blog dated 16 November 2026, Dan Winship of Red Hat, writing for SIG Network, says the project now plans to remove the IPVS backend of kube-proxy in Kubernetes 1.40. The backend was deprecated as of Kubernetes 1.35, and all IPVS users are told to plan a move to nftables or iptables.

TL;DR The short version

- Kubernetes plans to remove kube-proxy's IPVS mode, and IPVS users are told to plan a move to nftables or iptables.
- IPVS fixed iptables performance problems, but the kernel IPVS API proved a poor fit for the Kubernetes Services API.
- Find which proxy mode each cluster uses, including managed ones, and on Kubernetes 1.36 or later with Linux 5.13 or later the project recommends nftables.

Read online, listen to it, or change the text size: https://blogs.iictp.org/kubernetes-kube-proxy-ipvs-removal-nftables

What was reported

- The IPVS backend was built to fix iptables performance problems: connection latency that grew with the number of Services, and full ruleset rewrites whenever a Service or endpoint changed.
- The post says it met that goal, but the kernel IPVS API proved a poor match for the Kubernetes Services API, and the backend never implemented every edge case correctly.
- IPVS has no batch update operation, so large sets of changes could be slower than with iptables. The backend also fell back on iptables for features beyond basic load balancing.
- That dependency matters for the future: the post says Red Hat plans to remove the iptables API from Red Hat Enterprise Linux, which would also end the IPVS backend for Kubernetes users on RHEL.
- Most IPVS schedulers did not deliver their promise here either, the post explains, because kube-proxy instances on different nodes do not share scheduler state.

What a practitioner should do

- Find out which proxy mode each cluster uses, including managed and vendor-supplied clusters, and record the result.
- On Kubernetes 1.36 or later with Linux kernel 5.13 or later, the project recommends nftables. It says nftables is feature-complete and performs very slightly better than IPVS, and that fixes after 1.33 were not backported to

releases before 1.36.

- With kubeadm, the post describes changing the mode in the kube-proxy ConfigMap from ipvs to nftables and restarting the kube-proxy pods. Convert any IPVS-specific settings to their nftables equivalents as well.
- If you cannot use nftables, iptables is the fallback. The project says it has improved greatly since 2017, and recommends rebooting each node after this change rather than only restarting kube-proxy.
- Test the change in a non-production cluster first, and check anything that relied on IPVS behaviour or on inspecting IPVS state during troubleshooting.

The wider lesson

A component can succeed at its original goal and still become a liability if it depends on an interface that does not fit. Deprecation notices are a planning input, and the cheapest migration is the one scheduled before the deadline. That long view belongs to the Architecture & Systems Design domain of the TPBOK.

> Facts in this post were read from the sources listed below on 18 November 2026; this is a developing story or product and details may have changed since.

Sources and further reading

1. Kubernetes Blog (Dan Winship, Red Hat). Scheduling the removal of the kube-proxy IPVS backend. 16 November 2026. kubernetes.io/blog/2026/11/16/ipvs-removal/

Published by IICTP, an independent technical credential body, on 18 November 2026. The current version of this post is at https://blogs.iictp.org/kubernetes-kube-proxy-ipvs-removal-nftables. More guides: blogs.iictp.org.