Stonetusker Insights · Technical note on Kubernetes and platform engineering

Ingress NGINX No Longer Gets Security Fixes: Check Your Clusters in 30 Minutes

The Kubernetes project stopped maintaining Ingress NGINX in March 2026. Your clusters still run, which makes the risk easy to miss. This note shows how to check, what the result means and how to plan the move.

For: Teams that run Kubernetes with the community Ingress NGINX controller.

Skip if: You do not run Kubernetes.

By Subeesh Sivanandan, Founder, Stonetusker Systems · Updated 29 October 2026 · About 3 minutes to read · PDF version

The short version

- Ingress NGINX received its last maintenance in March 2026. The project publishes no more releases, bug fixes or security updates.[1]
- Existing installations keep running, so nothing fails on the day. The risk arrives with the next published vulnerability.
- About half of cloud native environments relied on this controller, according to research cited by the Kubernetes committees.[2]

- Two commands show whether you run it. The number of custom snippets tells you how long the move will take.

March 2026
Last maintenance[1]

About 50%
Cloud native environments that relied on it[2]

9.8 out of 10
Severity of the 2025 IngressNightmare flaw[4]

Why it matters now

Ingress handles every request from your customers. In March 2025 researchers at Wiz disclosed a group of flaws in Ingress NGINX called IngressNightmare. The worst one (CVE-2025-1974) scored 9.8 out of 10. It allowed an attacker with access to the pod network to run code in the controller and read secrets across the cluster.[4] Versions 1.12.1 and 1.11.5 fixed it.

The next flaw of this kind will stay open, because the project has ended. Some vendors sell patched builds of the retired project. They work as a bridge with an end date.

If you use the managed add-on on Azure Kubernetes Service, Microsoft provides critical security patches for it through November 2026.[5]

Check your clusters in 30 minutes

Step 1. Find the controller

Run this with cluster administrator permissions on every cluster, including staging and edge clusters. The Kubernetes committees published this command for this purpose.[2]

```
kubectl get pods --all-namespaces --selector
app.kubernetes.io/name=ingress-nginx
```

If pods come back, the cluster runs the retired controller.

Step 2. Confirm through the IngressClass

Custom Helm installations sometimes change the pod label, so an empty result in step 1 needs this second check.

```
kubectl get ingressclass -o custom-
columns=NAME:.metadata.name,CONTROLLER:.spec.controller
```

`k8s.io/ingress-nginx` means the retired community controller. `nginx.org/ingress-controller` means the F5 NGINX Ingress Controller, which is a separate product. Older installations may rely only on the `kubernetes.io/ingress.class` annotation, so run both checks.

Step 3. Count the annotations

This lists every Ingress NGINX annotation in use and how often it appears.

```
kubectl get ingress -A -o json \
  | jq -r '.items[].metadata.annotations // {} | keys[]' \
  | grep '^nginx.ingress.kubernetes.io/' | sort | uniq -c | sort -rn
```

What the result means

What you find	What it means	Typical effort
Common annotations such as `rewrite-target` and `ssl-redirect`	ingress2gateway translates these automatically, except rewrites that use capture groups such as `$1`	One to two weeks for a single cluster

What you find	What it means	Typical effort
`proxy-body-size`, `use-regex`, or authentication annotations such as `auth-url`	Not translated, or translated with behaviour that differs between implementations	Testing and a design decision for each one
Annotations that end in `snippet`	Raw NGINX configuration inside an Ingress object	Engineering work for each one; four to eight weeks per cluster group

The number of snippets predicts the timeline better than the number of services. The migration tool ingress2gateway 1.0, released in March 2026, translates more than 30 common Ingress NGINX annotations and flags the ones that need manual work.[3]

The move, step by step

1. **Inventory.** Run the checks above on every cluster and export all Ingress objects with their annotations to a file.
2. **Choose the destination.** Pick a Gateway API implementation for each cluster, and agree who owns Gateways and who owns Routes.
3. **Translate.** Run `ingress2gateway print --providers=ingress-nginx --all-namespaces` and read every warning. Treat the output as a first draft.
4. **Run both in parallel.** Install the new implementation with its own load balancer address and test each host with real requests, including redirects, headers and regular expression paths.
5. **Move one host at a time.** Shift traffic one hostname at a time, starting with the lowest risk, with a one-step rollback. Remove Ingress NGINX when its traffic reaches zero.

Most teams move to the Kubernetes Gateway API, the direction the Kubernetes project recommends.[1] Supported implementations include Envoy Gateway, kgateway, Istio, Cilium, NGINX Gateway Fabric, Traefik and Kong. A large estate can switch to another Ingress

controller first as a bridge. The F5 NGINX Ingress Controller uses different annotations, so a switch to it is also a migration.

What is still open

- Managed Kubernetes providers and third-party Helm charts may still install Ingress NGINX as an add-on or a dependency. Please confirm the plan and timing with each provider you use.
- Paid extended support offerings differ in scope and duration.
- ingress2gateway improves with each release. The annotation coverage in this note reflects version 1.0.

A note to forward to your team

> Ingress NGINX check
>
> The Kubernetes project stopped security fixes for Ingress NGINX in March 2026. Our clusters keep running, and the next vulnerability will stay open.
>
> Please run the three checks in this note on every cluster by [date] and reply with three numbers:
> 1. Clusters that run the controller
> 2. Ingress objects that use it
> 3. Annotations that end in "snippet"
>
> Owner: [name]
> Note: https://stonetusker.com/insights/ingress-nginx-retirement.html

Questions

Will our controller stop working?

The controller keeps running and the images remain available. What has ended is maintenance, so any new vulnerability stays open.

Is the F5 NGINX Ingress Controller the same product?

It is a separate project maintained by F5, with its own annotations. It is a valid destination, and moving to it is also a migration.

Do we have to move to Gateway API?

Another supported Ingress controller is a reasonable bridge. Gateway API is the direction of the Kubernetes project, so plan to arrive there.

How long does the move take?

A single cluster with host and path routing usually takes one to two weeks. Larger estates with custom snippets and regular expression routes usually take four to eight weeks per cluster group.

Want a second opinion?

Run the check yourself first. If you would like a second opinion, reply with three numbers only: the clusters that run the controller, the Ingress objects that use it and the annotations that end in snippet. No cluster names, configuration or command output are needed. Within five working days you receive a short written plan with a recommended destination, an order of work and a time estimate. It is free and confidential, and a call is optional.

Ask for a second opinion

The button opens an email to subeesh@stonetusker.com with the subject and the numbers to fill in. You can also use the contact page, or reply to the email or LinkedIn message that brought you here.

Next page for your situation

- Bitnami images: the other common component that stopped receiving fixes.
- Checklist for developer experience teams: eight plain questions and the dates that matter.

Sources

1. Kubernetes: Ingress NGINX Retirement, What You Need to Know (November 2025)
2. Kubernetes Steering and Security Response Committees: Ingress NGINX statement (January 2026)
3. Kubernetes: Announcing Ingress2Gateway 1.0 (March 2026)
4. Wiz: IngressNightmare, CVE-2025-1974
5. Microsoft Learn: Migrate from the NGINX application routing add-on to Gateway API on Azure Kubernetes Service

About the author

Subeesh Sivanandan is the Founder and CEO of Stonetusker Systems. He has worked for more than 27 years in release engineering, DevOps, platform engineering & embedded Linux. His career includes serving as the Principal Engineer & Manager of Release Engineering at Stryker, alongside earlier roles at Nokia, VeriSign, and IP Infusion. LinkedIn profile.

Revision history (recent 3 updates)

Version 2.1, 29 October 2026. Every fact on this page is checked against a minimum of two sources each month.

Version	Date	Change
1.4	16 October 2026	Rewritten as an article for quick readers
2.0	25 October 2026	Simpler layout and shorter text; the review offer now asks for three numbers only

Version	Date	Change
2.1	29 October 2026	Who-this-is-for box, email offer with the numbers listed, and next pages