

cert-manager

Project Update: Beyond 2026

Tim Ramlot & Maël Valais
Palo Alto Networks



cert-manager

X.509 certificate management for Kubernetes and OpenShift



14K+

GitHub Stars

500+

Contributors

1 million+

daily downloads

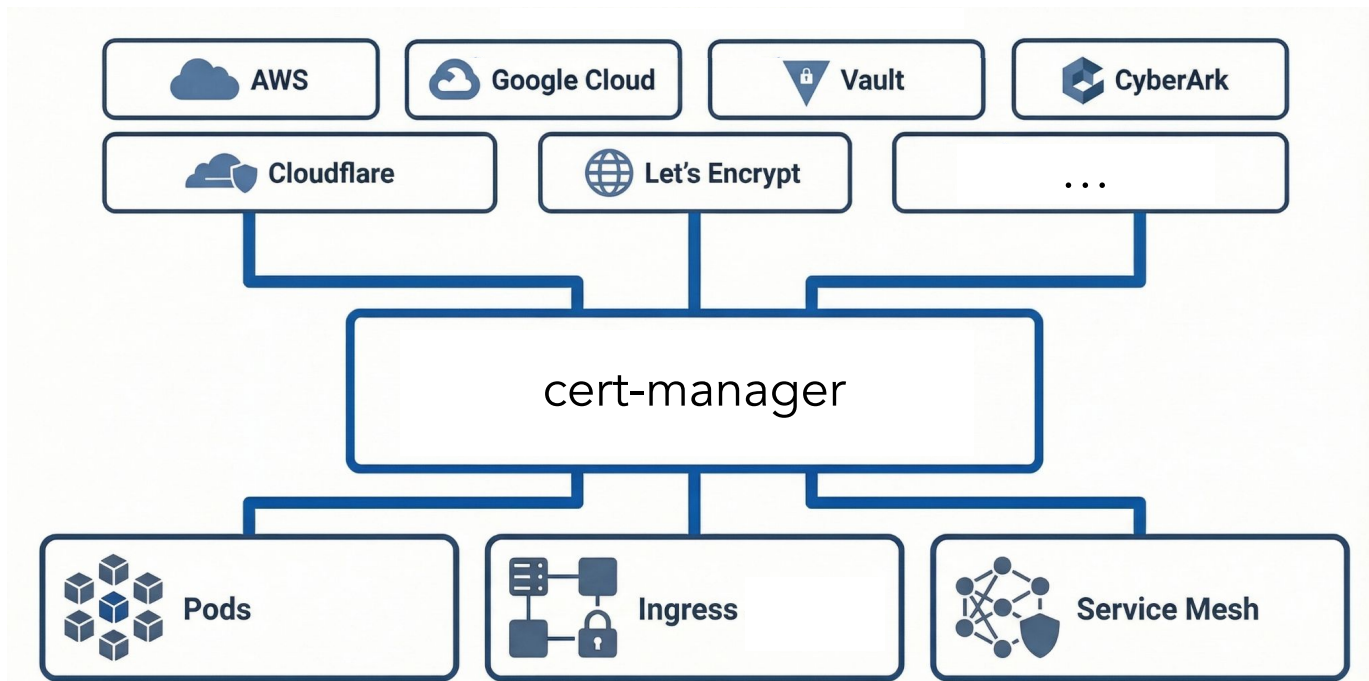


**CNCF Graduated
project**



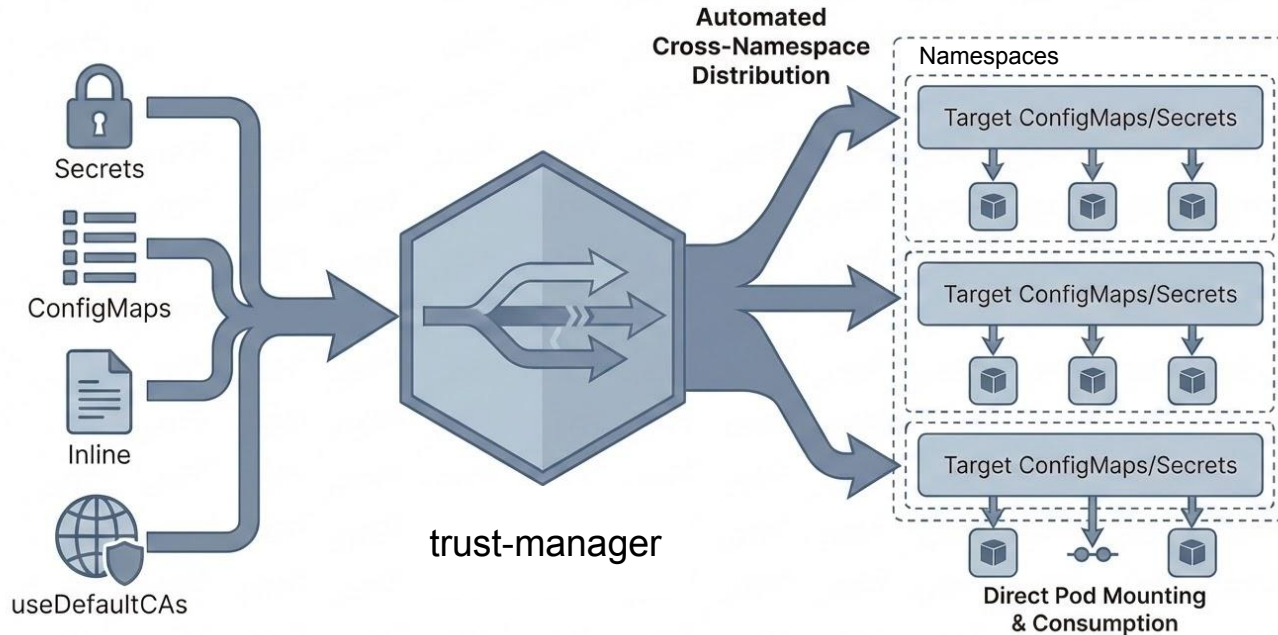
cert-manager / [cert-manager](#)

X.509 certificate issuance automation for Kubernetes



cert-manager / trust-manager

trust bundle composition and distribution in Kubernetes



Governance

@munnerz (Apple)

@joshvanl (Diagrid)

@wallrj (Palo Alto Networks)

@jakexks (G-Research)

@maelvls (Palo Alto Networks)

@sgtcodfish (Palo Alto Networks)

@inteon (Palo Alto Networks)

@thatsmrtalbot (Epic Games)

@erikgb (Zenior)

@hjoshi123 (Walmart) 🎉

New maintainer!!



Latest releases

cert-manager v1.20.0

- ListenerSet support
- Helm chart image configuration
- Azure private zones (DNS-01)
- You can now have very big certificates
- Lots happening in background: [server-side apply](#)

trust-manager v0.22.0

- Helm chart image configuration
- Skip non-CA certificates
- Lots happening to welcome the new ClusterBundle resource

<https://cert-manager.io/announcements/2025/09/05/trust-manager-clusterbundle-future/>

Helm chart image configuration

Before, wasn't easy to swap to use your own registry

```
1  image:
2    registry: artifactory.domain.com/docker
3    repository: jetstack/cert-manager-controller
4  acmesolver:
5    image:
6      registry: artifactory.domain.com/docker
7      repository: jetstack/cert-manager-acmesolver
8  webhook:
9    image:
10     registry: artifactory.domain.com/docker
11     repository: jetstack/cert-manager-webhook
12  cainjector:
13    image:
14     registry: artifactory.domain.com/docker
15     repository: jetstack/cert-manager-cainjector
16  startupapicheck:
17    image:
18     registry: artifactory.domain.com/docker
19     repository: jetstack/cert-manager-startupapicheck
```

Helm chart image configuration

Today, it is as simple as this:

```
1 | imageRegistry: artifactory.domain.com/docker
```

```
helm template cert-manager oci://quay.io/jetstack/charts/cert-manager --version v1.20.0 \  
--set imageRegistry="<registry>" \  
--set imageNamespace="<namespace>"
```

...

```
image: "<registry>/<namespace>/cert-manager-cainjector:v1.20.0"  
image: "<registry>/<namespace>/cert-manager-controller:v1.20.0"  
image: "<registry>/<namespace>/cert-manager-webhook:v1.20.0"  
image: "<registry>/<namespace>/cert-manager-startupapicheck:v1.20.0"
```

+ also works for our other Helm charts (trust-manager, approver-policy ...)

ListenerSet: Migrate off Ingress NGINX with Gateway API

And now it's urgent!

The screenshot shows a web browser window displaying the Kubernetes Blog. The browser's address bar shows the URL `kubernetes.io/blog/2025/11/11/ingress-nginx-retirement/`. The page header features the Kubernetes logo and navigation links: Documentation, Kubernetes Blog (active), Training, Careers, Partners, Community, Versions, and English. Below the header, there is a promotional banner for KubeCon + CloudNativeCon Europe 2026, with the text: "Join us for four days of incredible opportunities to collaborate, learn and share with the cloud native community. Buy your ticket now! 23 - 26 March | Amsterdam, The Netherlands". The main content area features the article title "Ingress NGINX Retirement: What You Need to Know" by Tabitha Sable (Kubernetes SRC) dated Tuesday, November 11, 2025. The article text states: "To prioritize the safety and security of the ecosystem, Kubernetes SIG Network and the Security Response Committee are announcing the upcoming retirement of Ingress NGINX. Best-effort maintenance will continue until March 2026. Afterward, there will be no further releases, no bugfixes, and no updates to resolve any security vulnerabilities that may be discovered. Existing deployments of Ingress NGINX will continue to function and installation artifacts will remain available." On the right side of the article, there are links for RSS Feed, Submit a Post, @kubernetes.io, @Kubernetesio, on GitHub, #kubernetes-users, Stack Overflow, and Forum. On the left side, there is a search bar and a sidebar menu with a date filter set to 2025, listing "Kubernetes 1.35: Enhanced Debugging with Versioned z-pages APIs" and "Kubernetes v1.35: Watch".

Browser tabs: Ingress NGINX Retirement: W x +

Address bar: `kubernetes.io/blog/2025/11/11/ingress-nginx-retirement/`

Navigation: Documentation, Kubernetes Blog, Training, Careers, Partners, Community, Versions, English

KubeCon + CloudNativeCon Europe 2026

Join us for four days of incredible opportunities to collaborate, learn and share with the cloud native community.
Buy your ticket now! 23 - 26 March | Amsterdam, The Netherlands

Ingress NGINX Retirement: What You Need to Know

By **Tabitha Sable (Kubernetes SRC)** | Tuesday, November 11, 2025

To prioritize the safety and security of the ecosystem, Kubernetes SIG Network and the Security Response Committee are announcing the upcoming retirement of [Ingress NGINX](#). Best-effort maintenance will continue until March 2026. Afterward, there will be no further releases, no bugfixes, and no updates to resolve any security vulnerabilities that may be discovered. **Existing deployments of Ingress NGINX will continue to function and installation artifacts will remain available.**

Search: Search this site

2026

2025

- Kubernetes 1.35: Enhanced Debugging with Versioned z-pages APIs
- Kubernetes v1.35: Watch

RSS Feed

Submit a Post

@kubernetes.io

@Kubernetesio

on GitHub

#kubernetes-users

Stack Overflow

Forum

ListenerSet: Migrate off Ingress NGINX with Gateway API

Had this issue opened for a while...

The screenshot shows a GitHub issue page for the 'cert-manager' repository. The issue title is 'Create certificate based on HTTPRoute configuration #7473'. It is categorized as a 'Feature' and has a parent ticket 'ListenerSet' (#7839). The issue was opened by 'BeyondEvil' on Dec 17, 2024, and edited by 'maelvls'. The description asks if the feature request is related to a problem and describes two problems: 1. It separates the responsibility between the Cluster Operator role and the Developer role. 2. It avoids having to use wildcard certificates for Gateways. The issue also mentions 'Role separation' and the responsibility of the Cluster Operator and Developer roles for the Gateway and HTTPRoute resources respectively. The current functionality causes a blurring of those lines at best, and a bottleneck at worst - as the Gateway needs updating with a specific listener for each publically exposed workload. The right sidebar shows the issue is assigned to 'maelvls' and has the label 'kind/feature'.

Create certificate based on HTTPRoute configuration #7473

Open Feature Parent: Top-level ticket: ListenerSet #7839

BeyondEvil opened on Dec 17, 2024 · edited by maelvls

Is your feature request related to a problem? Please describe.
This feature solves two problems.

1. It separates the responsibility between the Cluster Operator role and the Developer role
2. It avoids having to use wildcard certificates for Gateways

Role separation

The Gateway (gateway.networking.k8s.io/v1) resource is the responsibility of the Cluster Operator whilst the HTTPRoute (gateway.networking.k8s.io/v1) is the responsibility of the Developer role.

The current functionality causes a blurring of those lines at best, and a bottleneck at worst - as the Gateway needs updating with a specific listener for each publically exposed workload.

Assignees: maelvls

Labels: kind/feature

Type: Feature

Projects: No projects

49



✗ Migrating from Ingress to Gateway API without ListenerSet

Before migrating to Gateway API
(single multi-tenant ingress controller)

After migrating to Gateway API



Gateway

```
metadata:
  name: gw-123
  annotations:
    cert-manager.io/cluster-issuer: letsencrypt-prod
spec:
  gatewayClassName: your-gateway-class
  listeners:
    - hostname: example.com
      name: example
      port: 443
      protocol: HTTPS
      tls:
        certificateRefs:
          - name: example-tls
```

✗ No longer self-service:
developer has to
ask Cluster Operator

Ingress

```
metadata:
  annotations:
    cert-manager.io/cluster-issuer: letsencrypt-prod
spec:
  ingressClassName: nginx
  tls:
    - hosts:
        - example.com
      secretName: example-tls
```

```
rules:
  - host: example.com
    http:
      paths: [{pathType: Prefix, path: /, backend: {}}]
```

HTTPRoute

```
spec:
  hostnames:
    - example.com
  parentRefs:
    - name: gw-123
  rules:
    - matches:
        - path: {type: PathPrefix, value: /}
      backendRefs: [{name: ping, port: 80}]
```



✅ Migrating from Ingress to Gateway API using ListenerSet


Cluster
operators

Before migrating to Gateway API
(single multi-tenant ingress controller)

After migrating to Gateway API


Application
developers

Ingress

```
metadata:
  annotations:
    cert-manager.io/cluster-issuer: letsencrypt-prod
spec:
  ingressClassName: nginx

  tls:
  - hosts:
    - example.com
    secretName: example-tls

  rules:
  - host: example.com
    http:
      paths: [{pathType: Prefix, path: /, backend: {}}]
```

Gateway

```
metadata:
  name: gw-123
spec:
  gatewayClassName: your-gateway-class
  listeners: []
  allowedListeners:
  - from: Same
```

ListenerSet

```
metadata:
  annotations:
    cert-manager.io/cluster-issuer: letsencrypt-prod
spec:
  parentRef:
    name: gw-123
  listeners:
  - hostname: example.com
    name: example
    protocol: HTTPS
    port: 443
    tls:
      mode: Terminate
      certificateRefs:
      - kind: Secret
        name: example-tls
```



Stays self-service

HTTPRoute

Top-tier security

Protect against supply chain attacks



TOFU: digest matching all downloads



fast security updates



avoid unmaintained/
unnecessary dependencies



Protect against direct attacks



code scanning/
linting



automated and
verifiable releases



security audits

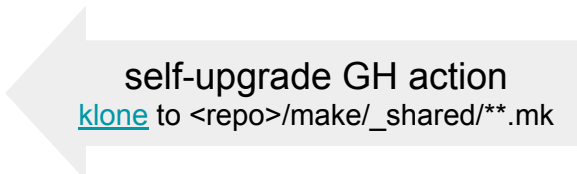


governance

Top-tier security across 11 repositories - HOW?

github.com/cert-manager/...

- **cert-manager**
- **trust-manager**
- **approver-policy**
- **istio-csr**
- **csi-driver**
- **csi-driver-spiffe**
- **google-cas-issuer**
- **cmctl**
- **issuer-lib**
- **openshift-routes**
- **csi-lib**



self-upgrade GH action
[clone](#) to <repo>/make/_shared/**/*.mk

github.com/cert-manager/

makefile-modules

- modules/tools
- modules/go
- modules/helm
- modules/licenses
- modules/oci-build
- modules/oci-publish
- modules/repository-base
- ...

+ renovate across all projects

Top-tier security: Protect against **supply chain** attacks

digest matching all downloads (trust on first use/ TOFU)

- go - go.dev/blog/supply-chain
- every downloaded binary's digest is checked:
cert-manager/makefile-modules/blob/main/modules/tools/00_mod.mk
- digests for GH actions & OCI images

fast security updates

- automated go, image & tool upgrades:
eg. cert-manager/makefile-modules/pull/560
- automated go dependency upgrades:
eg. cert-manager/cert-manager/pull/8627
- sometimes solve dependency CVE before scanners notice:
eg. go1.25.7 update - cert-manager/makefile-modules/pull/538

avoid unmaintained/ unnecessary dependencies

- minimal base images - cert-manager/base-images
- renovate “abandoned dependencies” - cert-manager/cert-manager/issues/8309



Top-tier security: Protect against **direct** attacks

code scanning/ linting (security + best practices)

- golangci-lint - [golangci-lint.run](#); [.golangci.override.yaml](#)
- govulncheck - [pkg.go.dev/golang.org/x/vuln/cmd/govulncheck](#)
- helm-tool linting - [github.com/cert-manager/helm-tool](#)
- boilersuite - [cert-manager/boilersuite](#)
- new: kubeapi linter (KAL) - [kubernetes-sigs/kube-api-linter](#)

automated and verifiable releases

- artifacts signed by pipeline
<https://cert-manager.io/docs/contributing/signing-keys/> + WIP

governance

- healthy set of maintainers: [maintainers.csv](#)
- trust in our contributors: [GOVERNANCE.md](#)

frequent audits

- cert-manager security audit - [AdaLogics-2023-cert-manager-audit-report.pdf](#)
- fuzzing audit - [AdaLogics-2025-cert-manager-fuzzing-audit-report.pdf](#)



Top-tier security: controlplane's hardening guide (2026)

65

ID: Title	Priority	Mitigated threats	Recommendation details	NIST SP 800-53 800-53 Revision 5
R-ARCH-01: Externalized and Tiered PKI Architecture	High	T-01 - Local CA collapse via exposed root CA private key T-03 - PKI trust domain collapse and workload impersonation via unencrypted private key storage T-10 - Service disruption by local CA rotation	Architect cert-manager with an offline or HSM-protected root CA and in-cluster intermediate CAs. Apply Intermediate CA name constraints to restrict certificate issuance to intended namespaces or domains, reducing blast radius. Integrate with enterprise PKI, Vault/OpenBao, or cloud KMS-backed issuers. Avoid using in-cluster CAs as trust anchors.	SC-12: Cryptographic Key Establishment and Management SC-17: Public Key Infrastructure SC-28: Protection of Information at Rest
R-ARCH-02: Certificate Issuance Governance and Policy Enforcement	High	T-02 - Policy bypass and cross-service trust abuse via enabled auto-approver T-11 - Persistent lateral movement and impersonation via long-lived non-revocable certificates T-13 - Ingress creation allows circumvention of certificate approval policies	Treat certificate issuance as a privileged identity function. Enforce SAN, usage, duration, and issuer constraints via <code>CertificateRequestPolicy</code> , or admission controls, (such as Gatekeeper and Kyverno). Align approval workflows with organizational trust boundaries. Enforce company-wide rules on certificate lifetime, aligned with industry direction and best practices. Ensure only trusted and maintained Issuers are used.	AC-3: Access Enforcement AC-6: Least Privilege CM-6: Configuration Settings
R-RBAC-01: Certificate Lifecycle Separation of Duties	High	T-02 - Policy bypass and cross-service trust abuse via enabled auto-approver	Separate certificate request, approval, and issuer administration responsibilities. Avoid allowing one identity to request and approve certificates. Restrict <code>CertificateRequest</code> creation to	AC-1: Policy and Procedures AC-5: Separation of Duties AC-6: Least Privilege

FUTURE work

- separate cert-manager/cert-manager issuers & migrate to issuer-lib
[cert-manager/issuer-lib/pull/332](#)
- webhook-cert-lib: replace cert-manager dynamic webhook CA and cainjector
[cert-manager/webhook-cert-lib/pull/129](#)
- make ACME issuer async
[cert-manager/cert-manager/pull/8071](#), [cert-manager/cert-manager/pull/7805](#), ...
- add image digests to Helm charts
[cert-manager/helm-tool/pull/104](#)
- fully migrate cert-manager to SSA
[cert-manager/cert-manager/pull/7908](#)
- renewal window support
[cert-manager/cert-manager/pull/8258](#); [cert-manager/cert-manager/pull/7399](#)
- ACME ARI support
[cert-manager/cert-manager/pull/8574](#)

Where can you find us?



<https://github.com/cert-manager/cert-manager>



cert-manager KubeCon In-Person Kiosk



daily meetings: <https://cert-manager.io/docs/contributing/#meetings>



slack: <https://cert-manager.io/docs/contributing/#slack>