

— GUIDE · RABBITMQ

RabbitMQ Erlang Version Compatibility: Version, Erlang and Patch-Owner Matrix

One row per RabbitMQ series from 3.8 to 4.4: which Erlang it runs on, the last public patch, the newer patch numbers the security advisories cite, when each support window closes, and who ships you a fix today. Checked against rabbitmq.com on 29 October 2026, with patch numbers taken from the RabbitMQ security advisories.



By **Scott Sternloff**, Senior Enterprise Architect

LinkedIn · Updated October 29, 2026 · 9 min read · 10 sections

— FIND YOUR ROW, THEN READ THREE COLUMNS

4.4 is current (4.4.0, released 20 Oct 2026), with community support to 31 Jul 2027. 4.3 keeps community support to 31 Jan 2027. Every other series is outside community support and gets fixes only through a commercial arrangement.

- **Erlang:** 4.4 runs on Erlang 27 to 29. Erlang 26 is end of life, which leaves 3.13 and 4.0.1 to 4.0.3 with no supported runtime.
- **Patch number:** the last public patch is not the last patch. The 25 October 2026 advisories cite fixes in 4.3.7, 4.2.12, 4.1.17 and 3.13.21, none of which is on the public release page.
- **Patch owner:** by rabbitmq.com's indicative dates, Broadcom commercial support is still open for 3.13, 4.1, 4.2, 4.3 and 4.4, and closed for 4.0 on 30 Sep 2026. AceMQ extended support covers RabbitMQ back to 3.8.x.

What this page is for

This is a lookup table for a planning meeting: find your series, read off the Erlang you need, the patch level you should be on, and who is on the hook for the next CVE. For the long-form explanation of RabbitMQ end-of-life dates and what end of support means for an audit, read the end-of-life article; for the Erlang 26 background, read the Erlang 26 article. This page does not repeat either.

The matrix: series, Erlang, patch level and patch owner

Read across your row. The patch-owner column is computed from rabbitmq.com's community and commercial end dates as of 29 October 2026; rabbitmq.com calls its commercial dates indicative and treats the Broadcom support portal as the contractual record, so plan with this table and commit against your subscription terms.

SERIES	LAST PUBLIC PATCH	NEWER PATCH CITED IN AN ADVISORY	ERLANG/OTP	COMMUNITY SUPPORT ENDS	COMMUNITY SUPPORT ENDS (INDICATED)
4.4	4.4.0 (20 Oct 2026)	None above the public patch	27.0 to 29.x	31 Jul 2027	31 Oct 2027
4.3	4.3.6 (16 Sep 2026)	4.3.7 (GHSA-9h6g-46wc-m2f5, 25 Oct 2026)	27.0 to 28.x on 4.3.6; 27.0 to 27.x on 4.3.3 to 4.3.5; 26.2 to 27.x on 4.3.0 to 4.3.2	31 Jan 2027	30 Apr 2027
4.2	4.2.11 (16 Sep 2026)	4.2.12 (GHSA-9h6g-46wc-m2f5, 25 Oct 2026)	27.0 to 27.x on 4.2.9 and later; 26.2 to 27.x on 4.2.0 to 4.2.8	31 Jul 2026	30 Jun 2026
4.1	4.1.8 (22 Jan 2026)	4.1.17 (GHSA-9h6g-46wc-m2f5, 25 Oct 2026)	26.2 to 27.x (4.1.0 to 4.1.4 as listed)	31 Jan 2026	30 Apr 2026
4.0	4.0.9 (14 Apr 2025)	4.0.24 (GHSA-6gmw-wxch-cwvc, 18 Aug 2026)	26.2 to 27.x on 4.0.4 and later; 26.2.x only on 4.0.1 to 4.0.3	30 Apr 2025	30 Sep 2025

SERIES	LAST PUBLIC PATCH	NEWER PATCH CITED IN AN ADVISORY	ERLANG/OTP	COMMUNITY SUPPORT ENDS	COMMUNITY SUPPORT ENDS (INDICATED)
3.13	3.13.7 (26 Aug 2024)	3.13.21 (GHSA-9h6g-46wc-m2f5, 25 Oct 2026)	26.0 to 26.2.x (Erlang 26 is end of life)	30 Sep 2024	31 Dec 2024
3.12	3.12.14 (6 May 2024)	None above the public patch	25.0 to 26.2.x on 3.12.10 and later as listed	29 Feb 2024	30 Jun 2024
3.11	3.11.28 (22 Dec 2023)	None above the public patch	25.0 to 25.3.x	30 Jun 2023	30 Jun 2023
3.10	3.10.25 (18 Jul 2023)	None above the public patch	24.3.4.8 to 25.3.x on 3.10.19 and later	30 Sep 2022	31 Dec 2022
3.9	3.9.29 (9 Mar 2023)	None above the public patch	24.3.4.2 to 25.2 on 3.9.28 and 3.9.29	31 Jan 2023	31 Jul 2023

SERIES	LAST PUBLIC PATCH	NEWER PATCH CITED IN AN ADVISORY	ERLANG/OTP	COMMUNITY SUPPORT ENDS	COMMERCIAL SUPPORT ENDS (INDICATED)
3.8	3.8.35 (9 Jul 2022)	None above the public patch	23.2 to 24.3 on 3.8.29 and later	31 Jan 2022	31 Jul 2022

Table • RabbitMQ series 3.8 to 4.4: patch levels, Erlang range, support windows and who patches it (checked 29 October 2026). Scroll sideways on small screens.

The last public patch is not the last patch

Once a series leaves community support, its public patch number stops moving while commercial patching carries on, so reading the public number as "latest" can leave you several patches behind while you believe you are current.

The advisories show it directly. The 18 August 2026 batch (for example GHSA-6gmw-wxch-cwvc) fixes issues in 3.13.19, 4.0.24 and 4.1.15, when the last public patches of those series are 3.13.7, 4.0.9 and 4.1.8. The 25 October 2026 batch (for example GHSA-9h6g-46wc-m2f5) cites 4.3.7, 4.2.12, 4.1.17 and 3.13.21. 4.2 is a useful edge case: it left community support on 31 July 2026, two more public patches followed (4.2.10 and 4.2.11), and the advisories already cite 4.2.12.

Only state a patch number you can source. A subscriber's exact current patch comes from the Broadcom portal; the advisory column above is the floor, not the ceiling.

Erlang: pick a version both ends of the move support

The Erlang you can run is set by the RabbitMQ patch, not the series, so check the exact patch against the matrix before you touch either.

- **Erlang 29 is supported from 4.4.0 and 4.3.7.** The docs list 4.3.7 with Erlang 29 although the public release page tops out at 4.3.6.
- **Erlang 28 is fully supported from 4.4.0 and 4.3.6.** On earlier 4.2 and 4.3 patches the docs allow Erlang 28 for brand-new clusters only, because upgrading a Khepri cluster from Erlang 27 to 28 can hit a known issue in mixed-version clusters, which is what a rolling upgrade is.
- **Erlang 27 is the floor from 4.2.9 and 4.3.3.** A 3.13 cluster cannot reach those patches in one move, because 3.13 tops out at Erlang 26.2.x.
- **Erlang 26 is end of life.** 3.13 and 4.0.1 to 4.0.3 run only on it.
- **Keep one major Erlang version across the cluster.** The docs highly recommend it, RabbitMQ refuses to cluster nodes with a potentially incompatible combination, and they recommend identical versions outside a reasonably long upgrade window.

In our upgrade work we sequence every step so that each node runs an Erlang version supported by both the RabbitMQ version it is leaving and the one it is moving to. That keeps the move rolling instead of turning it into a full stop.

Why the Erlang version is part of the RabbitMQ version

RabbitMQ is written in Erlang and runs on Erlang/OTP, which is why every release series carries an Erlang range and why an Erlang upgrade is a broker change, not an operating system patch. The protocol listeners run in that same runtime: AMQP 0-9-1 (the Advanced Message Queuing Protocol) and, since 4.0, AMQP 1.0 are core protocols, while MQTT (3.1, 3.1.1 and 5.0), STOMP and RabbitMQ Streams ship as

plugins, and the management plugin adds the HTTP API. One Erlang version therefore sits under every client connection to the node, whatever protocol it uses.

When a node joins a cluster, RabbitMQ checks Erlang's internal protocol versions and distributed libraries, and it refuses to cluster a potentially incompatible combination.

Erlang 25: which RabbitMQ versions still list it

Erlang 25 appears only in the older rows of the compatibility table, all of them outside community support.

- **3.11:** Erlang 25.0 to 25.3.x, so Erlang 25 is its newest runtime.
- **3.12:** Erlang 25.0 as the minimum, running up to 26.2.x on 3.12.10 and later as listed in the matrix.
- **3.10.19 and later:** 24.3.4.8 to 25.3.x.
- **No 4.x release:** the lowest Erlang any 4.x release accepts is 26.2 (4.0, 4.1, and the earlier 4.2 and 4.3 patches); 4.2.9 and later, 4.3.3 and later, and 4.4 need 27.0.

A RabbitMQ 3.11 user cannot reach Erlang 26 without leaving 3.11, because its maximum is 25.3.x. If you still run Erlang 25, check the exact build: the RabbitMQ docs note that Erlang 25 before 25.0.2 is affected by CVE-2022-37026, a critical-severity issue (CVSS 9.8).

Where the installed Erlang comes from

The Erlang version on a node is whatever package was installed, so ask where it came from before you plan an upgrade. Team RabbitMQ provides Debian and Ubuntu repositories, and for RPM-based systems its zero-dependency Erlang package, which follows the Erlang/OTP patch release schedule closely. Erlang is also bundled in the RabbitMQ Docker image. Distribution repositories often carry

heavily split packages and older Erlang versions unless backports are enabled, which is a common reason a node is on a different Erlang from its peers.

Three support windows, three different owners

Most version arguments come from treating three separate windows as one. They overlap, they end on different dates, and only one of them reaches the 3.8 to 3.12 range.

1. **Community support (the RabbitMQ project).** Public bug fixes and security patches, best effort, with no response commitment. Each series gets it for months, not years: 4.1 left it on 31 January 2026, 4.2 on 31 July 2026, and 4.3 leaves it on 31 January 2027.
2. **Commercial support (Tanzu RabbitMQ, from Broadcom).** Patches and fixes on a contracted response time through the Broadcom portal, until the commercial end date in the matrix. The dates do not run in version order: 3.13 is covered to 31 December 2029 while 4.0 closed on 30 September 2026.
3. **AceMQ extended support.** CVE patches backported to the build you are running, on open-source or Tanzu RabbitMQ, with a 15-minute P1 response. AceMQ extended support covers RabbitMQ back to 3.8.x, which includes series that sit outside the commercial window entirely.

How to read your own row in five minutes

Run two commands on any node and you have everything the matrix needs.

```
rabbitmq-diagnostics status  
rabbitmqctl cluster_status
```

- **Exact RabbitMQ patch and Erlang version on every node.** Mixed patches are normal mid-upgrade and a finding if they have been there for weeks.
- **Compare your patch with both patch columns.** Below the public patch means you are missing public fixes; between the public and advisory columns means you depend on a commercial build to close the gap.
- **Check the patch-owner column against your contract.** If it says commercial and you have no subscription, nobody is patching you.
- **Check the Erlang column against what is installed.** An Erlang outside the listed range is unsupported even when the broker runs.

If you cannot upgrade this quarter

Staying put is a valid decision when it is a managed one. Frozen change windows, certified applications, air-gapped estates and vendors who have not qualified a newer release are the usual reasons we see. What turns it into a risk is having no one contracted to ship the next fix.

For 3.13, 4.1 and 4.2 that can be a Broadcom subscription; for 4.0 and 3.12 back to 3.8 the commercial window has closed and the route is extended support. If you are ready to move instead, the upgrade runbook covers the hop sequence, and the upgrade service runs it with you.

Frequently asked questions

Which RabbitMQ versions are still supported?

As of 29 October 2026, RabbitMQ 4.4 (released 20 October 2026) and 4.3 are inside community support, to 31 July 2027 and 31 January 2027. By rabbitmq.com's indicative dates, Broadcom commercial support also covers 4.2, 4.1 and 3.13. 4.0 and 3.12 or older are outside both upstream windows; AceMQ extended support covers RabbitMQ back to 3.8.x.

Which Erlang version does RabbitMQ 4.4 need?

Erlang/OTP 27.0 up to 29.x. Erlang 29 is supported from 4.4.0 and 4.3.7, Erlang 28 is fully supported from 4.4.0 and 4.3.6, and Erlang 26 is end of life.

Why is my RabbitMQ patch older than the one in the security advisory?

Because public patches stop soon after a series leaves community support while commercial patching continues. The 25 October 2026 advisories cite fixes in 4.3.7, 4.2.12, 4.1.17 and 3.13.21, none of which is on the public release page, so a build on the last public patch is behind those fixes.

Can RabbitMQ 3.13 run on Erlang 27?

No. The compatibility matrix lists 3.13.0 to 3.13.7 with Erlang 26.0 to 26.2.x, and Erlang 26 is end of life. Moving to Erlang 27 means moving to 4.x.

Can I get security patches for RabbitMQ 3.8 to 3.12?

Not from the project or a Broadcom subscription, because both windows have closed for those series. AceMQ extended support covers RabbitMQ back to 3.8.x and backports CVE patches to the build you are running.

Does RabbitMQ 4.x support Erlang 25?

No. The 4.x series start at Erlang 26.2 on 4.0, 4.1 and the earlier 4.2 and 4.3 patches, and at 27.0 on 4.2.9 and later, 4.3.3 and later, and 4.4. Erlang 25 is the newest runtime only on 3.11 and an option on early 3.12 builds, both outside community support.

What happens if RabbitMQ nodes run different Erlang versions?

RabbitMQ checks Erlang's internal protocol versions and libraries when a node joins and refuses to cluster a potentially incompatible combination. The docs highly recommend one major Erlang version across the cluster, and identical versions outside a reasonably long upgrade window.

Related