

— GUIDE · RABBITMQ

RabbitMQ Federation vs Shovel Operating Runbook: Plugin Setup, Links and States

Which of the two to run for a given job, why all your links restarted at once, how to read link and shovel states, and the settings and patches to check before a link carries production traffic. Checked against rabbitmq.com on 29 October 2026, together with the 4.x release notes and security advisories.



By **Scott Sternloff**, Senior Enterprise Architect

LinkedIn · Updated October 29, 2026 · 12 min read · 10 sections

— FEDERATION FOLLOWS TOPOLOGY, SHOVEL MOVES WHAT YOU NAME

Federation keeps an ongoing relationship: a federated exchange replays what its upstream receives, and a federated queue pulls messages only when its local consumers need them. **Shovel** moves messages from a named source to a named destination unconditionally, always in one direction, which makes it the tool for migrations and drains.

- **Links restart for documented reasons.** Queue federation links run on the queue leader's node and are recreated after every leader election; exchange links and shovels move when their node stops. A rolling restart therefore restarts them all.
- **The CLI only sees one node.** `rabbitmqctl federation_status` reports the links on the node you ask, so check every node or use the HTTP API.
- **Patch level matters for credentials.** 2026 advisories fixed shovel URI passwords readable by monitoring users and a federation upstream that skipped vhost authorization.

What this runbook is for

This page is the operating checklist for links you already run or are about to run: choosing the mechanism, reading their state, setting the defaults and patching. Choosing federation or shovel as a disaster recovery design, with RPO and failover drills, is covered in the federation vs shovel DR article and the disaster recovery guide; this page does not repeat that comparison.

Pick the mechanism by the job, not by habit

Match the job to the row below; when two rows fit, prefer the one that writes the movement down explicitly. The docs recommend dynamic shovels over static ones unless you have a reason, because static shovels live in `advanced.config` and need a node restart to create or delete.

JOB	USE	WHY
Copy a flow published to an exchange in another cluster	Federated exchange	Replays messages published upstream to matching bindings downstream; a new downstream binding starts receiving without touching the link.
One logical queue across several clusters	Federated queue	Moves messages only when the upstream has no local consumers, so they go to where the consuming capacity is.
Move everything from queue A to queue or exchange B	Dynamic shovel	Moves unconditionally and is explicit about source and destination.
Drain a queue on a cluster you are retiring	Dynamic shovel with src-delete-after	queue-length measures the source queue at start and deletes the shovel after moving that many; src-delete-after-duration gives it a time limit.
Move between an AMQP 1.0 broker and RabbitMQ	Shovel (amqp10)	Source and destination can use different protocols.
Move between queues or vhosts in the same cluster	Local shovel (4.2 and later)	Uses an internal API rather than a network connection, so no TLS; the same authentication checks still apply.
Traffic in both directions	Federation, defined at both ends	A shovel is always one-way; queue federation can be bi- or N-directional, but each direction is its own link.
Clusters with over 100 ms p99 RTT or packet loss between them	Federation or shovel, not one stretched cluster	The clustering guide marks that range as not recommended for clustering and points to these two instead.

Table • Federation or shovel, by job (rabbitmq.com federation and shovel guides, 4.4). Scroll sideways on small screens.

Why did all my links restart?

Usually because cluster membership or leadership changed, which the docs describe as expected behaviour; the real risk is the one link that did not come back. Work down the list to find which cause applies.

1. **A leader election moved the queue.** Queue federation links start on the node holding the downstream queue's leader and are recreated on the new leader after every election. A rolling restart, a node added or removed, or `rabbitmq-queues rebalance` all trigger elections.
2. **A node stopped or crashed.** Exchange federation links and shovels run on one node each and restart on another when it goes.
3. **A policy changed.** A link that decides it is no longer needed after a policy change stops. Only one policy applies to an object, and among equal priorities one is picked at random, so a new policy can silently replace the one that enabled federation.
4. **The upstream kept failing.** Links reconnect after reconnect-delay (1 second by default) and give up if failures come too fast, generally one every few seconds. Raising reconnect-delay lets a large set of links ride out a flapping upstream.
5. **The source disappeared.** A link that cannot find its source exchange or queue stops trying.
6. **TLS stopped verifying.** From Erlang 26, TLS client peer verification is on by default, and TLS links without a client certificate and key fail to connect.
7. **Someone restarted them.** Before the fixes for GHSA-g5v3-w5xg-62q2 and GHSA-7jc3-73v6-rjvc, a user with only the monitoring tag could restart federation links and delete or restart dynamic shovels. 4.3.0 made both operations require policymaker.
8. **A node was shutting down on a pre-4.3 release.** 4.3.0 stopped federation links restarting during plugin or node shutdown, which had delayed shutdown on nodes with hundreds of links.

In our support work, federation is a recurring theme, and the costly part is almost never the restart: it is that nobody had a list of which links should exist, so the missing one surfaced days later as missing messages.

Read link and shovel state from every node

Check state cluster-wide, because the federation CLI reports only the node it runs on. The commands:

```
rabbitmqctl federation_status          # links on THIS node only
curl -u <user> https://<node>:15672/api/federation-links  # cluster-wide, needs r
rabbitmqctl shovel_status --formatter=pretty_table
rabbitmqctl list_parameters --vhost <vhost>              # what SHOULD exist
```

STATE	APPLIES TO	MEANING	ACTION
starting	Federation link, shovel	Starting or trying to connect to its endpoints	Stuck here: check reachability, credentials and TLS from the node running it; auth failures are logged there.
running	Federation link, shovel	Connected and moving messages	Compare count against your inventory, not just the state.
shutdown, with an error	Federation link	Link stopped; the error says why	Read the error, fix the upstream or policy, then let it reconnect.
terminated, with a reason	Shovel	Stopped or hit an exception	Read the reason; restart with <code>rabbitmqctl restart_shovel</code> once fixed.
Missing from the status	Federation link	No upstream, no matching policy, or the link gave up	Inspect upstreams, then policies for conflicting priorities, then node logs: the order the docs recommend.

Table • Federation link and shovel states and what to do (rabbitmq.com federation and shovel guides). Scroll sideways on small screens.

Check the defaults before a link carries production traffic

Most link defaults favour safety over speed, but three of them can lose or pile up messages, so review each row before go-live.

SETTING	DEFAULT	WHAT TO CHECK
ack-mode (federation and shovel)	on-confirm	Safest and slowest. on-publish and no-ack can lose messages on network or broker failure; use them only where that is acceptable.
prefetch-count / src-prefetch-count	1000	Higher can raise throughput up to a point, at the cost of link memory.
reconnect-delay	1 second	Raise it where many links share one upstream, so they tolerate higher failure rates.
Upstream queue-type (exchange federation)	classic	The internal upstream queue is single-replica. Set quorum (available since 3.13.1); changing it recreates the queue unless resource-cleanup-mode is never.
resource-cleanup-mode	Delete upstream queue when the link stops	Messages buffered upstream go with it. Set never if a link must keep collecting while you change its configuration.
expires and message-ttl (upstream queue)	none	With no expiry the upstream buffer grows for as long as the link is down. Bounding it means accepting loss; decide which you want.
max-hops (exchange federation)	1	Raise only for deliberate multi-hop topologies; messages can still be duplicated over multiple routes.
src-delete-after (shovel)	never	Use queue-length for one-off drains, or src-delete-after-duration (minimum 60 seconds by default) for a time limit.
URIs	One endpoint	List several: both federation and a shovel pick one at random on each attempt until one endpoint connects.

Table • Federation and dynamic shovel settings worth checking (rabbitmq.com federation reference and dynamic shovel guide, 4.4). Scroll sideways on small screens.

The operating checklist

Run this list when a link goes in, after every upgrade or membership change, and at each quarterly review. The first four items are AceMQ practice; the rest come from the docs and advisories.

1. **Keep an inventory of every upstream, upstream set, federation policy and shovel.** Diff `rabbitmqctl list_parameters` and `rabbitmqctl list_policies` against it after any membership change, because that is the only way to see a link that failed to come back.
2. **Alert on link state, not only node health.** A down link breaks nothing locally and can run unnoticed for days.
3. **Give each link its own least-privilege user.** A failure then names itself in the logs of the node running the link.
4. **Re-check every link after an upgrade.** Rolling upgrades cause the leader elections that recreate queue federation links.
5. **Give federation policies explicit priorities.** Avoid policies that apply to all, so another policy cannot displace the one that enables federation.
6. **Manage upstream sets with the CLI.** The HTTP API does not support upstream sets.
7. **Test a definitions export before you depend on it.** On older releases, `rabbitmqctl export_definitions` failed with badarg when a federation-upstream-set held more than one upstream (GitHub issue #11612, fixed for the 3.13.5 milestone).
8. **Configure client certificates on TLS links.** Erlang 26 and later verify peers by default.
9. **Keep DEBUG logging off on nodes running shovels unless you are patched.** GHSA-9wm4-9m6g-w38x (DEBUG logs) and GHSA-q44f-9vc5-grh7 (crash reports) both wrote decrypted shovel URLs to logs on unpatched builds.

0. **Patch before you trust monitoring users near link credentials.** GHSA-6gmw-wxch-cwvc let monitoring users read shovel source and destination passwords through the HTTP API (fixed in 3.13.19, 4.0.24, 4.1.15, 4.2.10 and 4.3.5), and GHSA-x5h5-588r-cv55 exposed AMQP 1.0 shovel URI passwords through shovel status (fixed in 3.13.15, 4.0.20, 4.1.11, 4.2.6 and 4.3.0). We recommend rotating link credentials after patching.
1. **Patch federation upstreams that cross vhosts.** GHSA-42pc-678q-v8qj let a policymaker on one vhost drain messages from another through a direct-connection upstream, and with the default ack-mode the source messages were consumed. Fixed in 3.13.18, 4.0.24, 4.1.14, 4.2.9 and 4.3.3.

What changed for links in 4.2, 4.3 and 4.4

If your links were set up on 3.13 or earlier, these are the 4.x changes that alter how they behave.

- **4.2: local shovels.** A third shovel protocol, `local`, consumes and publishes inside the same cluster through an internal API.
- **4.3: faster, quieter shutdown.** Federation links and their connections stop in parallel and no longer restart during plugin or node shutdown.
- **4.3: policymaker required.** Restarting a federation link and deleting a shovel through the management API now need the policymaker tag.
- **4.3: src-consumer-name.** An optional property sets the shovel's consumer tag or AMQP 1.0 link identifier, which makes it identifiable on the source.
- **4.4: rejected messages go back to the source.** When the destination rejects a message, for example a full queue with reject-publish, a quorum source dead-letters it once it passes the delivery limit instead of redelivering it to the shovel indefinitely, which was a 4.3.0 regression. A classic source has no delivery limit, so the message keeps returning until the destination accepts it.

- **4.4: AMQP 1.0 shovel fixes.** Settlement on reject or release, duplicate forwarding after out-of-order confirms, and a local shovel with src-delete-after moving more than its configured count were all fixed.
- **4.4: stricter URI parsing.** Unsupported `auth_mechanism` or `verify` values in federation and shovel URIs are now rejected when the URI is parsed rather than at connection time, so a typo fails the definition instead of the link.

Enable the federation plugin and the shovel plugin

Neither mechanism does anything until its plugin is on, and a cluster needs the federation plugin enabled on every node.

```
rabbitmq-plugins enable rabbitmq_federation
rabbitmq-plugins enable rabbitmq_federation_management    # management plugin: link
rabbitmq-plugins enable rabbitmq_shovel
rabbitmq-plugins enable rabbitmq_shovel_management        # Admin > Shovel Status a
```

Federation links use AMQP 0-9-1 connections between the clusters. It is built for brokers in different administrative domains, such as different data centers, and it assumes the connection between them will fail from time to time. Use the `amqps` scheme and port 5671 for TLS, with certificates passed as URI query parameters, rather than `amqp` and port 5672. A shovel's TLS settings apply only when its URI uses `amqps`.

Shovel configuration: src-uri, dest-uri and the queue at each end

A dynamic shovel is a runtime parameter, so you can create, change and delete it without a node restart. One shovel consumes every message from its source and publishes it to its destination. This example consumes from a queue on the local broker and publishes to a destination queue on a remote one.

```
rabbitmqctl set_parameter shovel my-shovel \  
  '{"src-uri": "amqp://", "src-queue": "source-queue",  
    "dest-uri": "amqp://remote-server", "dest-queue": "target-queue",  
    "dest-queue-args": {"x-queue-type": "quorum"}}'
```

- **src-uri and dest-uri.** The source and destination broker connection strings. Each can be a single URI or a list.
- **Queue or exchange, never both.** Specify either `src-queue` or `src-exchange` at the source, and either `dest-queue` or `dest-exchange` at the destination. With a source exchange the shovel declares an exclusive queue and binds it, and `src-exchange-key` sets the routing key for that binding.
- **Routing on the destination.** With `dest-exchange`, `dest-exchange-key` sets the routing key; if it is not set, the original message's routing key is used.
- **Requeue and redelivery after a stop.** When a shovel stops, its in-flight messages are requeued at the source, so they are consumed again after the restart. Make the consumer at the destination tolerate duplicates.
- **Watching it.** `rabbitmqctl shovel_status`, `GET /api/shovels` and the Admin > Shovel Status page all give the list of shovels and their state. Throughput is bounded by `src-prefetch-count`, the prefetch limit on unacknowledged messages, which defaults to 1000.

Federation and shovel on Amazon MQ for RabbitMQ (AWS)

Amazon MQ for RabbitMQ supports both plugins, with limits that depend on whether a broker is private, meaning reachable only from inside its VPC. Check them before you plan a link to or from an AWS broker; the rules below are from the Amazon MQ plugin documentation.

- **Shovel:** only dynamic shovels are supported, and a shovel cannot target a private broker as its destination.

- **Federation:** you configure it on the downstream broker, and it cannot be configured when the upstream queue or exchange is on a private broker. The supported pairs are public to public, or a public upstream feeding a private downstream.
- **So for a link into AWS from a private network:** if the AWS broker is private and the on-premises side is the source, a shovel cannot deliver into it. Federate from a public upstream, or make the AWS broker public, and confirm which fits your network policy.

The destination URI for an Amazon MQ broker uses `amqps` and port 5671, so the TLS note in section 7 applies.

Frequently asked questions

Should I use federation or shovel in RabbitMQ?

Federation for an ongoing relationship that should follow the topology: a federated exchange copies what is published upstream, and a federated queue pulls messages when local consumers need them. Shovel for an explicit, one-way move from a named source to a named destination, such as a migration or draining a queue on a cluster you are retiring.

Why did all my RabbitMQ federation links restart?

Queue federation links run on the node holding the downstream queue's leader and are recreated after every leader election, and exchange links move when their node stops. Rolling restarts, adding or removing nodes and rebalancing all cause this. Policy changes, repeated upstream failures and, on unpatched builds, a monitoring-tag user can also restart them.

Why does rabbitmqctl federation_status show no links?

It only reports links on the node you ran it on. Run it on every node or query `GET /api/federation-links`. If a link is missing cluster-wide, check that the upstream exists, that the federation policy is the one actually applied (equal-priority policies are picked at random), and the node logs.

How do I make a RabbitMQ shovel stop after it drains a queue?

Set `src-delete-after` to `queue-length` on a dynamic shovel: it measures the source queue length when it starts and deletes itself after transferring that many messages. An integer value sets the count directly, and cannot be combined with `ack-mode no-ack`. `src-delete-after-duration` sets a time limit instead, with a 60-second minimum by default.

Is RabbitMQ federation bidirectional?

Each federation link is one-way: an upstream defined on cluster B pulls from cluster A. Two-way movement needs links defined at both ends. Queue federation can form bi- or N-directional graphs this way; a shovel is always unidirectional.

How do I enable the RabbitMQ federation plugin?

Run `rabbitmq-plugins enable rabbitmq_federation`, on every node if you have a cluster, and add `rabbitmq_federation_management` if you want links in the management UI and HTTP API. Then define an upstream and a policy that matches the exchanges or queues to federate; each match starts a link.

What are src-uri and dest-uri in a RabbitMQ shovel?

They are the connection URIs of the source and destination brokers in a dynamic shovel definition. Each is paired with a queue or an exchange at that end (src-queue or src-exchange, dest-queue or dest-exchange, one of each pair, not both). A URI can be a list, in which case the shovel picks one at random until an endpoint connects.

Does Amazon MQ for RabbitMQ support federation and shovel?

Yes, with limits. Only dynamic shovels are supported, and a shovel cannot use a private broker as its destination. Federation cannot be configured when the upstream is on a private broker. Public-to-public, or a public upstream to a private downstream, is supported.

Related