



VMware Tanzu RabbitMQ

Upgrade & Migration Guide

Introduction

RabbitMQ is a powerful and widely used messaging and streaming broker that enables fast, reliable, and versatile communication for applications, making it ideal for distributed microservices, real-time data processing, and highly scalable systems.

VMware Tanzu RabbitMQ builds on this foundation by delivering enterprise-ready packaging, enhanced security hardening, exclusive advanced features, and tested upgrade flows, with support for deployment on Kubernetes, as a virtual appliance (OVA), or via traditional Linux packages. Tanzu RabbitMQ is a logical upgrade from the community open source version not least because it is managed and built by the same team. Additionally the list of Tanzu specific continues to grow with each release.

Just as RabbitMQ's highly durable **Quorum Queues (QQ)**, designed with **data safety as the top priority**, replaced **Classic Mirrored Queues (CMQ)** to bring stronger reliability and operational consistency, migrating from open-source RabbitMQ to VMware Tanzu RabbitMQ unlocks a suite of **Tanzu-exclusive features**, including:

- **Warm Standby Replication** – enables continuous schema and data replication to downstream clusters for fast failover and disaster recovery, with features like pausing/resuming sync and x.509 cert-based authentication [VMware Blogsdocs.vmware.com](https://blogs.vmware.com).
- **Intra-Cluster Compression** – compresses data traffic between RabbitMQ nodes, reducing cloud inter-zone bandwidth costs docs.vmware.com.
- **Ready-to-Run OCI Container Image** – curated dependencies and “safety by default,” optimized for seamless deployment across environments docs.vmware.com.
- **Production-Safe Defaults** – built-in safe memory-to-disk ratios (e.g., disk free limit matches memory), improved crash-dump handling, up-to-date cryptographic standards like TLS 1.2+ [VMware Blogsdocs.vmware.com](https://blogs.vmware.com).

Migrating to Tanzu RabbitMQ also ensures predictable upgrade lifecycles, enterprise support, and safe adoption of advanced features like **Khepri**, RabbitMQ's next-generation metadata store.

When it comes to upgrading, the **standard method should be in-place rolling upgrades**, allowing nodes to be upgraded sequentially with minimal downtime. Yet, for large version jumps where rolling

upgrades aren't supported, a **Blue-Green strategy** offers a safer upgrade path by maintaining continuous service availability during migration.

1. Installation Methods

Before diving into upgrades, it's important to know the supported installation methods. The commands differ, but the goal is the same: run RabbitMQ in a stable, reproducible environment.

A. Kubernetes – Helm (OCI Chart)

Helm is the package manager for Kubernetes. It packages resources into *charts*, making installs and upgrades repeatable.

Why use it?

- Rapid deployment
- Easy version management
- Rolling upgrades supported by Kubernetes

TechDocs (Install):

[Installation using Helm – VMware Tanzu RabbitMQ on Kubernetes \(v4.1\)](#)

Example:

None

```
helm registry login
rabbitmq-helmoci.packages.broadcom.com -u
username -p <your-token>
```

```
helm upgrade --install trmq
oci://rabbitmq-helmoci.packages.broadcom.
com/tanzu-rabbitmq/tanzu-rabbitmq
--version <VERSION> --namespace messaging
--create-namespace -f values.yaml
```

B. Kubernetes – Carvel (kctrl / kapp-controller)

Carvel is a suite of tools for app lifecycle management on Kubernetes. `kctrl` is the CLI to install *packages* into clusters.

Why use it?

- Native to Tanzu Kubernetes Grid (TKG)
- Integrates with Tanzu packaging repository
- Supports multi-cluster consistency

TechDocs (Install):

[Installation using Carvel – VMware Tanzu RabbitMQ on Kubernetes \(v4.1\)](#)

Example:

None

```
kctrl package repository add -r
tanzu-rabbitmq -n
kapp-controller-packaging-global --url
<CARVEL_REPO_URL>

kctrl package install -i trmq -p
tanzu-rabbitmq.<domain> -v <VERSION> -n
messaging --values-file values.yaml
```

C. OVA (Virtual Appliance)

An OVA is a pre-packaged virtual machine image, ready to run on vSphere or other hypervisors.

Why use it?

- Turnkey deployment
- Minimal Kubernetes knowledge required
- Includes pre-configured RabbitMQ

TechDocs (Install):

[VMware Tanzu RabbitMQ Virtual Machine \(OVA\) Installation \(v4.1\)](#)

Flow: Download → Import into vSphere (or compatible hypervisor) → Power on → Configure via CLI or management UI

D. RPM (Linux)

RPM is the package format for RHEL, CentOS, and related distros.

TechDocs (Install):

[Installing RabbitMQ on RPM-based Linux](#)

Why use it?

- Fits traditional VM or bare-metal deployments
- Works with existing OS-level monitoring and automation

Highlights:

- Ensure Erlang/OTP compatibility.
- Use official RabbitMQ repositories.

Example:

None

```
dnf install -y erlang rabbitmq-server
```

2. Dependencies Before Upgrading RabbitMQ Erlang/OTP

- Each RabbitMQ version supports specific Erlang versions — check the compatibility matrix: [Erlang Compatibility Matrix](#).
- Upgrade Erlang first if required.

Example:

None

```
sudo yum install  
erlang-<COMPATIBLE_VERSION>
```

Other dependencies

- OpenSSL 1.1+ or 3.x
- Updated `glibc` and `libstdc++`
- Disk space & RAM for metadata store migration
- Kubernetes API version compatibility for K8s installs

3. Checking RabbitMQ Status

Kubernetes:

None

```
kubectl -n messaging get pods -l  
app.kubernetes.io/name=rabbitmq  
kubectl exec -it <pod> --  
rabbitmq-diagnostics status  
kubectl exec -it <pod> -- rabbitmqctl  
cluster_status
```

OVA / RPM:

None

```
sudo systemctl status rabbitmq-server  
sudo rabbitmq-diagnostics status  
sudo rabbitmqctl cluster_status
```

Enable UI:

None

```
sudo rabbitmq-plugins enable  
rabbitmq_management
```

Access via: <http://<host>:15672/>

4. Pre-Upgrade Checklist

- Read Release Notes — Understand changes in RabbitMQ and Erlang/OTP.
- Verify Erlang/OTP and dependency compatibility
- Confirm cluster health (no alarms, nodes reachable)
- Backup definitions:

None

```
rabbitmqadmin export  
/tmp/definitions.json
```

- Backup persistent storage (stop and copy data directory)
- Follow supported upgrade paths
- In Kubernetes: verify chart/package version compatibility

5. Upgrade Flows

Upgrades in RabbitMQ should ideally be performed using **in-place rolling upgrades**, where nodes are sequentially upgraded while the cluster continues serving traffic.

This method minimizes downtime and disruption, making it the **recommended standard upgrade path** for Tanzu RabbitMQ in most production environments.

Note: Rolling upgrades are supported only when the version step is compatible. If the version gap is too large or rolling upgrades are not supported for the target version, a **Blue-Green upgrade strategy** should be followed instead to ensure service continuity.

A. Kubernetes – Helm

TechDocs (Upgrade):

[Upgrade using Helm – VMware Tanzu RabbitMQ on Kubernetes \(v4.1\)](#)

None

```
helm upgrade --install trmq
oci://rabbitmq-helmoci.packages.broadcom.
com/tanzu-rabbitmq/tanzu-rabbitmq
--version <NEW_VERSION> -n messaging -f
values.yaml
```

```
kubectl -n messaging rollout status
statefulset/trmq
```

B. Kubernetes – Carvel

TechDocs (Upgrade):

[Upgrade using Carvel – VMware Tanzu RabbitMQ on Kubernetes \(v4.1\)](#)

None

```
kctrl package installed update -i trmq -n
messaging --version <NEW_VERSION>
--values-file values.yaml
```

C. OVA (Virtual Appliance)

For OVA, **Blue-Green strategy is recommended..**

TechDocs (Blue-Green):

[Blue-Green Upgrade – RabbitMQ](#)

Flow:

1. Deploy **Green OVA** (new version) alongside **Blue OVA** (current).
2. Export definitions from Blue:

None

```
rabbitmqadmin export  
/tmp/definitions.json
```

3. Import into Green:

None

```
rabbitmqadmin import  
/tmp/definitions.json
```

4. Set up Federation to sync queues:

None

```
rabbitmqctl set_parameter  
federation-upstream blue  
'{"uri":"amqp://blue-host"}'  
  
rabbitmqctl set_policy --apply-to queues  
migrate-all ".*"  
'{"federation-upstream":"blue"}'
```

5. Cut over traffic to Green.
6. Decommission Blue after validation.

D. RPM

None

```
sudo rabbitmqctl stop
sudo rpm -Uvh
tanzu-rabbitmq-server-<NEW_VERSION>.<arch
>.rpm
sudo rabbitmqctl start
```

6. Blue-Green Upgrade Strategy (General)

- Deploy a parallel **Green** cluster
- Migrate configuration & definitions
- Use Federation/Shovel for queue data migration
- Redirect clients to Green
- Retire Blue once stable

7. Khepri Database in RabbitMQ

Starting with RabbitMQ **4.0**, **Khepri** is fully supported but **Mnesia** remains the default metadata store in versions 4.0 and 4.1.

Khepri must be explicitly enabled using the `khepri_db` feature flag.

Benefits of Khepri:

- Faster and safer crash recovery
- Handles large metadata volumes efficiently
- More predictable upgrade behavior

Check metadata store in use:

None

```
rabbitmqctl eval
'application:get_env(rabbit,
schema_db_dir).'
```

8. Post-Upgrade Validation

None

```
rabbitmqctl status
rabbitmqctl list_feature_flags
rabbitmqctl cluster_status
rabbitmq-diagnostics check_local_alarms
rabbitmq-diagnostics check_port_listener
```

Test:

- Client connections
- Queue throughput
- Management UI access

9. Migrating from OSS to VMware Tanzu RabbitMQ

It is often overlooked that the team at Broadcom own, manage and maintain open source RabbitMQ. It is this same team that builds on this to produce the commercial Tanzu RabbitMQ builds. Whether deploying on Kubernetes using the Tanzu RabbitMQ Helm chart or deploying on bare metal using the raw components exclusively available in a tarball. Tanzu RabbitMQ takes the out of the box open source RabbitMQ to the next level and beyond, with robust, production-ready artifacts that empower teams to better manage messaging workloads at scale with greater confidence.

Why migrate:

Migrating from open-source RabbitMQ to VMware Tanzu RabbitMQ unlocks significant advantages:

- **Enterprise-grade readiness** — Tanzu RabbitMQ includes validated, production-ready images (OVA and OCI) with built-in security, dependency validation, and hardened defaults that minimize operational risk.
- **Enhanced resilience and data safety** — Includes features like Warm Standby Replication for fast failover, automated multi-site replication with declarative config, and intra-cluster compression to optimize network usage.
- **Strong governance and security posture** — Features such as vault integration for credential management, TLS 1.2+ enforced by default, FIPS compliance, and enterprise SLAs ensure secure and reliable operations.

- **Simplified deployment and support** — Tanzu RabbitMQ delivers day-zero-ready deployments with curated images, access to VMware engineering resources, and 24/7 enterprise-grade support—reducing the complexity of installing, upgrading, and securing the platform.
- **Future-proof enhancements** — Native improvements such as AMQP 1.0 support over WebSockets, MQTTv5 for IoT scale, JMS integration, performance optimizations for Quorum Queues, stream filtering, and a revamped metadata store (Khepri) offer ongoing operational and development benefits.

Steps:

1. Export definitions from OSS:

None

```
rabbitmqadmin export  
/tmp/definitions.json
```

2. Perform the Migration:

- **Preferred:** Use an in-place rolling upgrade if the version step is supported — upgrade nodes sequentially while the cluster remains online.
- **Fallback:** If rolling upgrades aren't possible (e.g., due to a major version gap), adopt a Blue-Green migration by running OSS and Tanzu clusters in parallel and syncing traffic/state.

3. Import definitions into VMware Tanzu RabbitMQ:

None

```
rabbitmqadmin import  
/tmp/definitions.json
```

4. Switch clients.
5. Decommission OSS RabbitMQ (in-case of blue-green upgrade).

Conclusion

Upgrading RabbitMQ means checking Erlang/OTP compatibility, backing up, following the right upgrade path, and validating results.

VMware Tanzu RabbitMQ offers enhanced stability, tested packaging, enterprise-grade support, and exclusive features like **Warm Standby Replication**, **Intra-Cluster Compression**, **AMQP 1.0 over WebSocket** and **Full Khepri Support**—delivering improved resilience, security, observability, and scalability that go beyond open-source RabbitMQ.

Copyright © 2025 Broadcom. All rights reserved.

The term "Broadcom" refers to Broadcom Inc. and/or its subsidiaries. For more information, go to www.broadcom.com. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies. Broadcom reserves the right to make changes without further notice to any products or data herein to improve reliability, function, or design. Information furnished by Broadcom is believed to be accurate and reliable. However, Broadcom does not assume any liability arising out of the application or use of this information, nor the application or use of any product or circuit described herein, neither does it convey any license under its patent rights nor the rights of others. Item No: vmw-bc-flyr-2p-a4-word-2025 Dec-24