

Tanzu RabbitMQ for Kubernetes

Self-Service Access to Easily Managed RabbitMQ Clusters

FEATURES

- Rapid, self-service provisioning
- Automated operations
- Rolling upgrades with no downtime
- Extensive security features
- VMware expert support

KEY BENEFITS

Easily support cloud-native application development patterns

- Event-driven architectures
- Microservices
- Continuous deployment

Your choice of clouds

- Simpler to integrate apps across public cloud services or a combination of private and public clouds
- Place the right workloads on the right cloud and avoid vendor lock-in

Self-service deployment with automated operations

- Simplify operations of large-scale messaging services, even in multi-cluster Kubernetes environments
- Keep your workloads properly isolated
- Simplify lifecycle management to reduce your risk and shift your focus to more strategic work.

Tanzu RabbitMQ is now available with a custom operator that makes the widely popular messaging broker easy for developers to access and consistent for platform operators to manage with any certified Kubernetes runtime.

Self-Service Access to Easily Managed RabbitMQ Clusters

When it comes to building distributed applications, messaging software serves as the central nervous system integrating far-flung instances. As developers express an increasing appetite for shipping code with high velocity, they don't want to become bogged down with procurement, installation, and provisioning of their messaging systems. Developers also want a high degree of automation to avoid getting tangled up in the drudgery of management and operations. Meanwhile, platform operations teams can help these application development teams most effectively when the messaging services they provision and operate are stable, predictable, and require the minimum amount of configuration.

Emerging as a de-facto standard for cloud-native messaging, Tanzu RabbitMQ delivers efficient communication among all of your apps and services. Tanzu RabbitMQ is a fast and dependable message server that supports a wide range of use cases, including reliable integration, content-based routing and global data delivery, and high-volume monitoring and data ingestion.

With Tanzu RabbitMQ for Kubernetes, now developers can provision both the Tanzu RabbitMQ and the open-source RabbitMQ message brokers with simple commands on top of any Kubernetes cluster. The operator works automatically with the Kubernetes runtime to maintain the desired cluster state. Developers can also simplify connections between applications, deploy more scalable systems, and ship faster. At the same time, platform operators get a simplified provisioning and management experience without getting bogged down in tedious configuration and deployment details.

Enterprise messaging services, now available on Kubernetes

Based on the widely popular open source RabbitMQ messaging system, Tanzu RabbitMQ is designed to work seamlessly with Tanzu Editions and also to run on any certified Kubernetes runtime, including OpenShift. Deploying Tanzu RabbitMQ on Kubernetes provides an easy, automated way to deploy and manage RabbitMQ brokers for reliable event and message distribution.

With Tanzu RabbitMQ on Kubernetes, you can run the same messaging services across data centers, public cloud, and hybrid cloud for a consistent experience. Your workloads remain properly isolated

and secure, no matter where you run them. Deploy, scale, patch, and upgrade all of your clusters with reduced downtime.

Expert engineering knowledge codified in software

Tanzu RabbitMQ features a Kubernetes operator to automate the entire cluster lifecycle:

Configure and deploy Tanzu RabbitMQ or open-source RabbitMQ clusters, with best practice defaults and plugins for wide-scale enterprise service (enterprise features and support only available with Tanzu RabbitMQ)

Plugins for monitoring clusters using Prometheus and Grafana (end eventually Tanzu Observability)

Scaling and recovery to declared state

Upgrade and reconfigure clusters with zero downtime

The Tanzu RabbitMQ operator codifies the engineering knowledge needed to deploy and operate Tanzu RabbitMQ. It also automates operational flows using Kubernetes primitives (e.g., stateful sets and persistent volumes), which allows specific RabbitMQ configuration options and commands to be managed via the Kubernetes API.

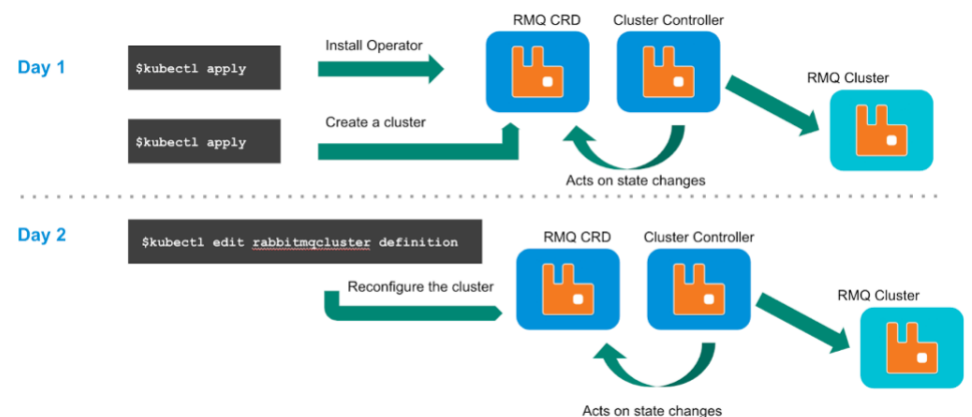


Figure 1: How the Tanzu RabbitMQ operator reconciles state

The only requirement is a certified Kubernetes runtime; everything else is handled by the Tanzu RabbitMQ operator. The operator specifies the desired cluster deployment state. It consists of a custom resource definition (CRD) and a *custom controller*. A custom resource extends the basic capabilities of Kubernetes and can be managed the same as any other Kubernetes object. A CRD file defines these objects in YAML for Kubernetes to create and watch. Controllers (control loops) watch and regulate the state of the custom Kubernetes objects defined in the CRD files. As a result, the Kubernetes API can orchestrate specific application lifecycle tasks and reconcile the actual cluster state to the desired state without manual intervention.

Why use Tanzu RabbitMQ for Kubernetes?

- Remove developer friction with self-service message broker deployment

- Ease the load on platform operations with expert knowledge and automation built in to the Tanzu RabbitMQ operator
- Improve governance with an allowed set of configurations, in the form of easy to read and maintain declarative YAML manifests.
- Simplify deployments with messaging as code: use GitOps to control your declarative cluster configuration, and the operator will automatically reconcile any changes to the configuration across your fleet of clusters. You can also control the timing for each cluster upgrade or reconfiguration
- Faster time to scale: containers start much faster than virtual machines
- Faster time to recovery: Tanzu Rabbit MQ for Kubernetes use controller loops to watch and automatically heal your cluster
- Deploy where it makes sense for your business: on any cloud, virtualization infrastructure, or even on bare metal
- Get full observability: expose metrics in Prometheus format, and annotate them for discoverability by observability software ingesting Prometheus sources
- Zero-downtime upgrades: Tanzu RabbitMQ for Kubernetes increases your business continuity providing automated, specific upgrades between Tanzu RabbitMQ versions without the need to bring down the clusters.

Rolling upgrades made easier

Rolling upgrades of a running deployment can be performed by simply updating the Kubernetes deployment manifest. Once this happens, the cluster operator component will act on each node in turn to:

- Safely stop message processing, shifting queues to other nodes as appropriate
- Update the node to the new version of Tanzu RabbitMQ
- Restart the Tanzu RabbitMQ nodes
- Rebalance replicated queues to improve resilience

The operator covers a variety of upgrade scenarios, among them:

- Placing specific deployments on a deny list from automatic upgrade
- Upgrading the operator, then upgrading deployments not on the deny list
- Maintaining multiple versions in the fleet

Users will periodically upgrade the operator to get the latest features. The Tanzu RabbitMQ operator is upgradeable using a single command and involves **no downtime for any of the managed deployments**. Newer versions of the operator maintain backwards compatibility to manage existing deployments.

Security

Security is a major concern in both cloud and shared infrastructure environments. Tanzu RabbitMQ for Kubernetes helps developers secure their applications for zero-trust configurations using a variety of measures:

- Running the operator under the service account (non-root)
- Limiting the scope to watch resources in specific namespaces
- Running clusters under a non-privileged daemon user

- Implementing zero-trust mTLS-encrypted communication between clients and the deployment
- Implementing optional zero-trust mTLS-encrypted communications between nodes
- Implementing Oauth support

Supported by VMware experts

Tanzu RabbitMQ for Kubernetes was created by VMware experts with years of experience deploying it in enterprise environments in order to make going to production safer and easier. Enterprise support plans give you peace of mind around the clock.

Where do I start?

Tanzu RabbitMQ for Kubernetes is ready for you today! [Download it from the Tanzu Network](#) and be sure to have a look at the Tanzu RabbitMQ for Kubernetes [quick start guide](#) and [the documentation](#).