TECHNICAL WHITE PAPER: **November**

Tanzu RabbitMQ on vSphere Kubernetes Service on VMware Cloud Foundation

Reference Architecture

Table of contents

Executive Summary

The validation of Tanzu RabbitMQ on VMware vSphere Kubernetes Service (VKS) platform represents a critical milestone in modern cloud-native infrastructure deployment. By selecting VMware Cloud Foundation, the private cloud of choice, enterprises can accelerate the adoption of modern applications such as RabbitMQ with VKS by using their cloud admins current skillsets.

Below are benefits of using VKS with modern applications:

Lower TCO: With VKS organizations have the ability to reduce silos, leverage existing tools and skill sets without having to retrain staff and/or change existing processes. Utilizing unified lifecycle management across infrastructure components to stay up to date with the most recent patches and minimizing security risks.

Operational Simplicity: VKS is engineered for unparalleled operational simplicity, leveraging the familiarity of existing vSphere tools, skills, and workflows. This design philosophy significantly reduces the learning curve for IT teams and streamlines management processes. With VKS, organizations benefit from automated cluster provisioning, which accelerates deployment times and minimizes manual configuration errors. Furthermore, its robust capabilities extend to automated upgrades and comprehensive lifecycle management. This integrated approach ensures consistent operations, reduces overhead, and frees up valuable resources to focus on innovation rather than infrastructure maintenance.

Run and Manage Kubernetes at Scale: Effortlessly deploy and manage Kubernetes clusters at scale, leveraging a built-in, Cloud Native Computing Foundation (CNCF) certified Kubernetes distribution. VKS provides fully automated lifecycle management, streamlining operations from initial setup to ongoing maintenance and upgrades. This comprehensive approach ensures that organizations can harness the power of Kubernetes for their containerized applications with unparalleled efficiency and reliability, without the complexities typically associated with large-scale Kubernetes deployments.

VMware vSphere Kubernetes Service

VMware vSphere Kubernetes Service (VKS) is the Kubernetes runtime built directly into VMware Cloud Foundation (VCF). With CNCF certified Kubernetes, VKS enables platform engineers to deploy and manage Kubernetes clusters while leveraging a comprehensive set of cloud services in VCF. Cloud admins benefit from the support for N-2 Kubernetes versions, enterprise grade security, and simplified lifecycle management for modern apps adoption.

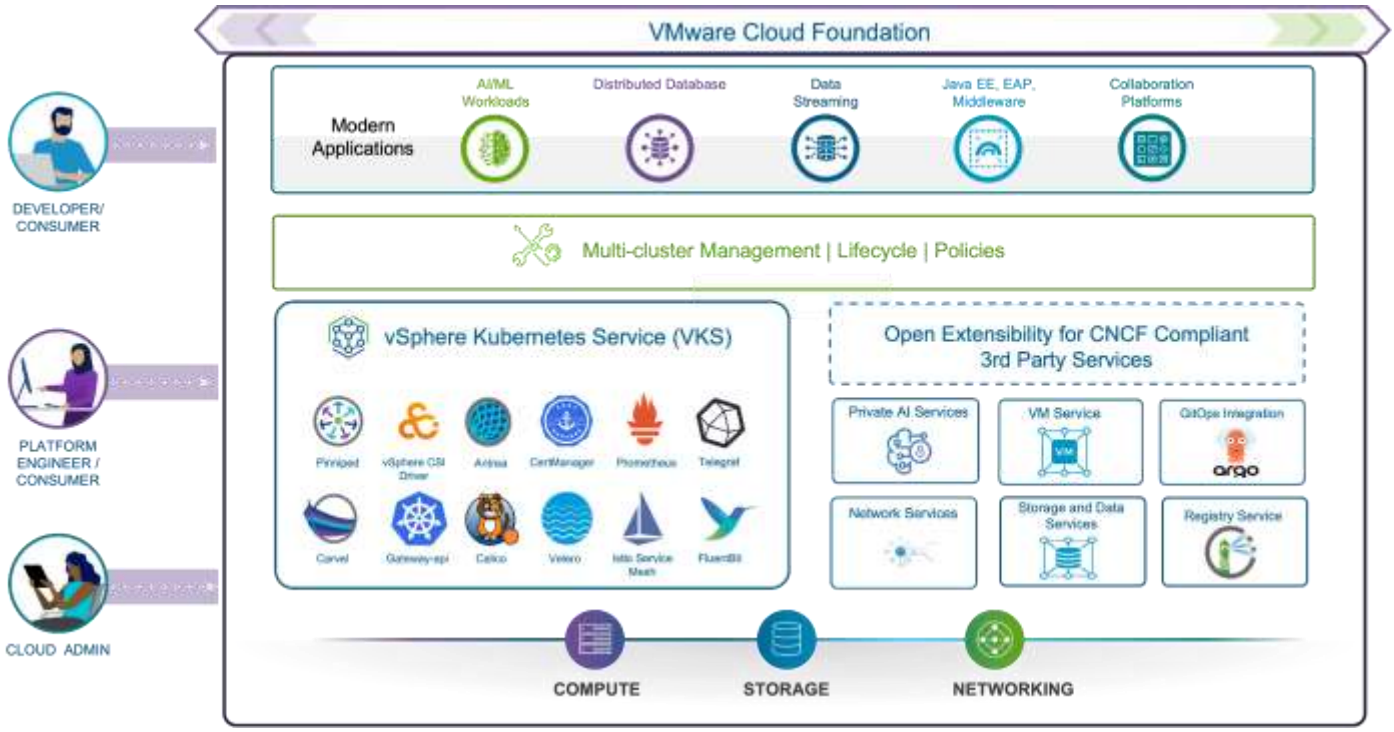


Figure 1: VKS on VCF Ecosystem

Tanzu RabbitMQ

RabbitMQ is a fast and dependable open-source message server, which supports a wide range of use cases including reliable integration, content-based routing and global data delivery, and high-volume monitoring and data ingestion.

Emerging as the de facto standard for cloud messaging, RabbitMQ is used for efficient communication between servers, apps, and devices, and creates lasting value by enabling rapid development of modern decentralized app and data architectures that can scale with your business needs.

Tanzu RabbitMQ includes the following features exclusive to Tanzu RabbitMQ on Kubernetes:

- Warm Standby Replication
- Intra-cluster Compression
- Forward proxy support in OAuth 2.0 plug-in
- Change LDAP plugin configuration at runtime

Please refer to official Tanzu RabbitMQ documentation for details.

https://techdocs.broadcom.com/us/en/vmware-tanzu/data-solutions/tanzu-rabbitmq-on-kubernetes/4-2/tanzu-rabbitmq-kubernetes/tanzu-rabbitmq-features.html

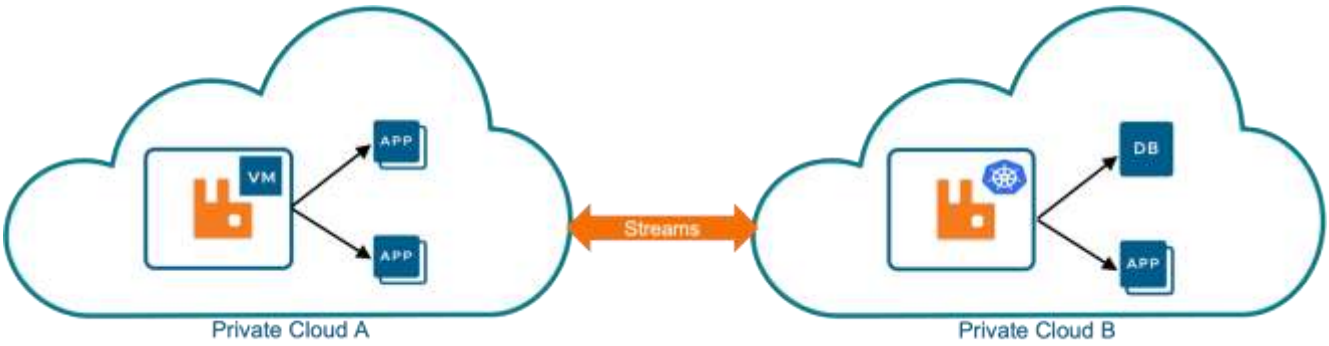


Figure 2: Tanzu RabbitMQ Workflow

Solution Architecture

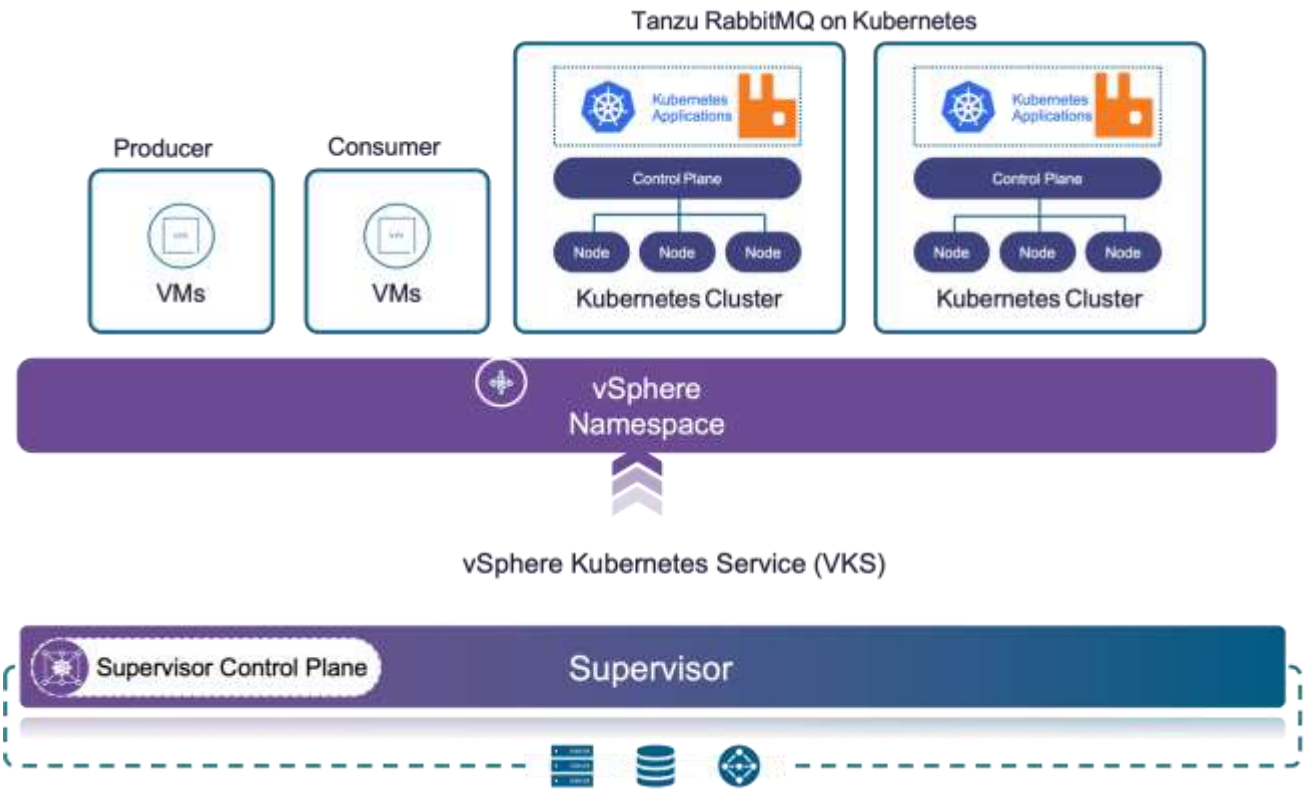


Figure 3: Tanzu RabbitMQ on VKS

Below is the list of software versions used during this validation.

Component	Version	Notes

Tanzu RabbitMQ for Kubernetes	4.2	
vSphere Kubernetes Service	3.4	
vSphere Cloud Foundation	9.0	
Kubernetes	1.33	VKS Kubernetes Release
Helm	3.19	

Solution Validation

Below we provide the installation and validation steps for Tanzu RabbitMQ on VKS using Helm. Please review carefully as the steps below may differ for your environment.

Install VMware Tanzu RabbitMQ Operator on Kubernetes using Helm

Please refer to VMware Tanzu official documents:

https://techdocs.broadcom.com/us/en/vmware-tanzu/data-solutions/tanzu-rabbitmq-on-kubernetes/4-2/tanzu-rabbitmq-kubernetes/installation-using-helm.html

Validate Connectivity to VMware Tanzu RabbitMQ Cluster

1. Create a user namespace where the RabbitMQ cluster will be created.

```
kubectl create ns rabbitmq
```

2. Create a secret tanzu-rabbitmq-registry-creds in the user namespace using the repo at *rabbitmq.packages.broadcom.com* prior to deploying the RabbitMQ cluster. Refer to the Tanzu official documents on how to obtain your username and token.

https://techdocs.broadcom.com/us/en/vmware-tanzu/data-solutions/tanzu-rabbitmq-on-kubernetes/3-13/tanzu-rabbitmq-kubernetes/installation.html

```
kubectl create secret docker-registry tanzu-rabbitmq-registry-creds \
--docker-server "rabbitmq.packages.broadcom.com" \
--docker-username "<username>" \
--docker-password "<your-token>" \
-n rabbitmq
```

3. Create rmq-cluster.yaml for deploying a RabbitMQ cluster

```
apiVersion: rabbitmq.com/v1beta1
kind: RabbitmqCluster
metadata:
  name: my-tanzu-rabbit
  namespace: rabbitmq
spec:
  replicas: 1
  imagePullSecrets:
  - name: tanzu-rabbitmq-registry-creds
  service:
    type: LoadBalancer
```

4. Create a RabbitMQ cluster using the command below

```
kubectl apply -f rmq-cluster.yaml
rabbitmqcluster.rabbitmq.com/my-tanzu-rabbit created


# Confirm if the cluster is up and running
kubectl get all -n rabbitmq
NAME                            READY   STATUS    RESTARTS   AGE
pod/my-tanzu-rabbit-server-0    1/1     Running   0          71s

NAME                             TYPE           CLUSTER-IP     EXTERNAL-IP     PORT(S)
AGE
service/my-tanzu-rabbit          LoadBalancer   10.103.31.88   10.138.169.32
5672:32025/TCP,15672:32192/TCP,15692:30912/TCP   71s
service/my-tanzu-rabbit-nodes    ClusterIP      None           <none>          4369/TCP,25672/TCP
71s

NAME                                       READY   AGE
statefulset.apps/my-tanzu-rabbit-server    1/1     71s

NAME                                           ALLREPLICASREADY   RECONCILESUCCESS   AGE
rabbitmqcluster.rabbitmq.com/my-tanzu-rabbit   True               True               71s


kubectl get rmq -n rabbitmq
NAME              ALLREPLICASREADY   RECONCILESUCCESS   AGE
my-tanzu-rabbit   True               True               3m31s
```

5. To test the connectivity to the RabbitMQ server, first create a RabbitMQ user using the RabbitMQ server pod as follows:

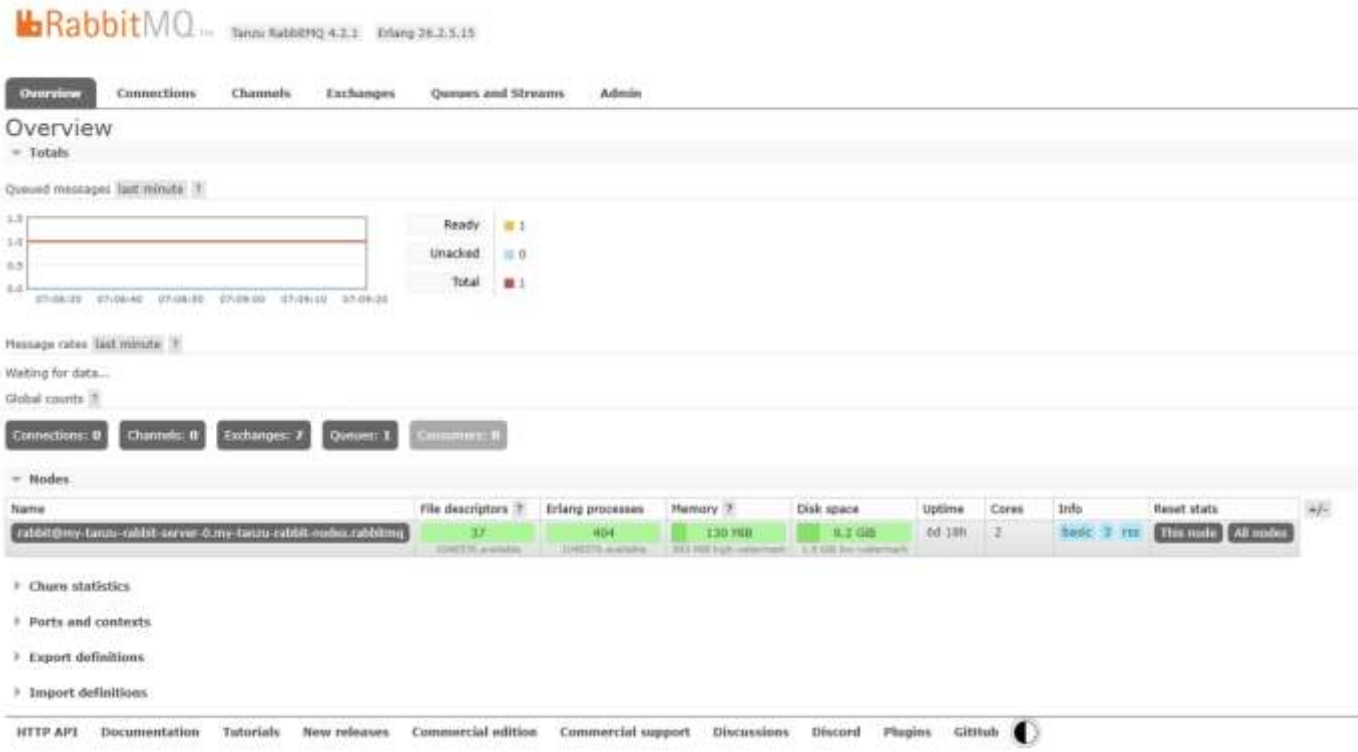
```
kubectl -n rabbitmq exec -it my-tanzu-rabbit-server-0 -- bash
rabbitmqctl add_user rmquser1 rmquser1
rabbitmqctl set_user_tags rmquser1 administrator
rabbitmqctl set_permissions rmquser1  ".*" ".*" ".*"
```

6. Refer to https://github.com/rabbitmq/omq and use *omq* CLI to publish and consume sample messages. Note the example IP 10.138.169.32 is retrieved from the LoadBalancer output in the previous step.

```
sudo apt install golang-go
go install github.com/rabbitmq/omq@main
go/bin/omq amqp --uri amqp://guest:guest@10.138.169.32:5672/ --queues quorum --time 5s
2025/11/21 17:53:55 INFO Declaring queues...
2025/11/21 17:53:55 INFO consumer started id=1 terminus=/queues/omq-1
```

```
2025/11/21 17:53:55 INFO publisher started id=1 rate=unlimited destination=/queues/omq-1
2025/11/21 17:53:56 published=198/s consumed=198/s
2025/11/21 17:53:57 published=257/s consumed=257/s
2025/11/21 17:53:58 published=256/s consumed=256/s
2025/11/21 17:53:59 published=251/s consumed=251/s
2025/11/21 17:54:00 published=256/s consumed=256/s
2025/11/21 17:54:00 TOTAL PUBLISHED messages=1227 confirmed=1227 returned=0 rate=245.11/s
2025/11/21 17:54:00 TOTAL CONSUMED messages=1227 rate=245.11/s
```

7. For a UI validation for RabbitMQ, browse to http://10.138.169.32:15672. Login with the previously created username/password rmquser1/rmquser1



Conclusion

The adoption of vSphere Kubernetes Service (VKS) on VMware Cloud Foundation (VCF) for deploying Tanzu RabbitMQ shifts the management paradigm from infrastructure heavy lifting to application-focused operations. VKS offers RabbitMQ customers a deeply integrated, enterprise-grade foundation, leveraging Cloud Native Storage (CNS) for robust, persistent volume management crucial for message queue integrity and state. This integration drastically simplifies tasks like scaling, failover, and hardware lifecycle management, allowing for higher density and more efficient resource utilization. Ultimately, by standardizing RabbitMQ deployments on VKS and VCF, organizations gain the operational agility of Kubernetes combined with the reliability and governance of VMware Cloud Foundation, ensuring their critical messaging infrastructure is both resilient and easily scalable to meet modern demand spikes.