TECHNICAL WHITE PAPER
Feb 2025

VMware Tanzu RabbitMQ OVA and vSphere vMotion

Compatibility, Benefits and Best Practices

Table of contents

Executive Summary:

VMware Tanzu RabbitMQ OVA simplifies the deployment of RabbitMQ as a pre-configured virtual appliance. However, many organizations leveraging VMware environments seek clarity on the compatibility and benefits of using Tanzu RabbitMQ OVA with vSphere vMotion. This white paper explores the integration, benefits, and best practices for ensuring seamless vMotion operation of Tanzu RabbitMQ OVA.

Introduction:

In dynamic enterprise environments, maintaining high availability and minimizing downtime during infrastructure changes is critical. Tanzu RabbitMQ, a high-performance message broker, can be deployed as an OVA in VMware vSphere to simplify setup. VMware's vMotion technology enables live migration of virtual machines (VMs) without service interruption. Organizations often questioned whether Tanzu RabbitMQ OVA could handle the complexities of live migrations. Extensive testing has now proven that Tanzu RabbitMQ OVA migrates seamlessly in VMware vSphere, providing operational resilience and flexibility.

Problem Statement:

Organizations previously expressed concerns about the reliability of Tanzu RabbitMQ OVA when used with vSphere vMotion, particularly under high workloads and with active queue processing. However, recent validations have confirmed that Tanzu RabbitMQ OVA with Khepri database enabled works effectively with vSphere vMotion, ensuring:

- Uninterrupted message flow during migrations.
- Consistent service availability, even under high workloads.
- Reliable data integrity and operational resilience.

These issues necessitate a clear approach in performing vMotion migration of Tanzu RabbitMQ OVA in vSphere.

Proposed Solution:

By deploying a Tanzu RabbitMQ OVA with the Khepri database and adhering to best practices for configuration and resource allocation, organizations can ensure a seamless and successful vMotion migration of the Tanzu RabbitMQ OVA:

- Zero Downtime: Ensure message broker availability during live migrations.
- Data Integrity: Maintain queue consistency and prevent message loss.
- Operational Efficiency: Perform infrastructure upgrades and maintenance without disrupting services.

Tanzu RabbitMQ and Khepri:

Note:

Khepri is a fully supported metadata store starting with the RabbitMQ 4.0.x release. However, RabbitMQ 3.13.x includes only experimental support for Khepri and is not ready for use in production. Additionally, it is not possible to upgrade from 3.13.x to 4.x or beyond if Khepri was enabled on the 3.13.x node(s).

Warning:

Users considering enabling Khepri in RabbitMQ 3.13.x should carefully evaluate the potential complexities and ensure compatibility before proceeding.

During vMotion migration, partial network partitioning may occur among the nodes. RabbitMQ provides robust network partition recovery strategies to mitigate potential disruptions. RabbitMQ adheres to the Consistency and Partition Tolerance (CP) model from the CAP theorem, ensuring strong data integrity while remaining resilient to network partitions. Additionally, Khepri's Raft-based approach to failure handling effectively manages network partitions, enhancing overall system resilience.

Consistency (C):

- Quorum Queues & Khepri Metadata Store:
  - Uses Raft consensus to ensure updates (messages, metadata) are consistently replicated.
- Publisher Confirms & Transactions:
  - Guarantees messages are either fully committed or discarded, preventing partial updates.
- Ordered Log Replication:
  - Khepri's Raft-based log ensures nodes replay missed updates in a deterministic manner.

Partition Tolerance (P):

Partition tolerance means that RabbitMQ continues to function even if network partitions cause some nodes to become unreachable. Key partition tolerance strategies include:

- Leader Election in Khepri (Raft Algorithm):
  - If the leader node becomes unavailable due to a network partition, a new leader is elected from the available followers using the Raft leader election process.
  - The old leader steps down upon rejoining to prevent conflicts.
- Network Partition Handling Strategies:
  - RabbitMQ provides different partition handling strategies (pause_minority, pause_if_all_down, autoheal, and ignore), to manage partitions effectively.
  - The default pause_minority strategy ensures consistency by pausing nodes in the minority partition, preventing them from making conflicting updates.
- Delayed Synchronization:
  - Partitioned nodes sync with the leader upon reconnection, ensuring state consistency and avoiding split-brain scenarios.

Implementation Steps:

- Deploy Tanzu RabbitMQ OVA with Khepri database:
  - Import the Tanzu RabbitMQ OVA into the vSphere environment
  - Configure network settings to ensure stable connectivity
  - Use persistent storage to store RabbitMQ data to prevent data loss during migrations
  - Enable Khepri database

- Enable vMotion Compatibility:
  - Ensure Tanzu RabbitMQ virtual machine is hosted on shared storage (e.g., vSAN, NFS) to facilitate live / vMotion migration.
  - Verify that the VMkernel network adapter supports vMotion traffic.
  - Configure CPU and memory reservations to minimize resource contention.

- Tested Live Migration:
  - Performed multiple instances of vMotion migration (change of compute resource) of Tanzu RabbitMQ OVA in a controlled environment.
  - Monitored RabbitMQ's performance and logs during the migration to detect potential issues.
  - Observations:
    - The vMotion process was successful even after a longer duration of migration time (in Fig (a), the vMotion had consumed 14 minutes).
    - No interruptions in message processing or service availability were detected during or after the migration. Refer Fig (b) and Fig (c).
  - Successful vMotion test results:
    - Idle Cluster: Result: **Pass**
    - Cluster under load: Result:**Pass**
    - Active Quorum Queues: Result: **Pass**
    - Active Streams: Result: **Pass**

- Environment Details:
  - Tanzu RabbitMQ Version: 4.0.5
  - vSphere: 8.0 U3
  - Cluster Configuration: 3-node RabbitMQ cluster (with Khepri database)
  - Load Details:
    - 3 active queues:
      - Classic queue
      - Quorum queue (10 producers and 10 consumers)
      - Stream queue (20 producers and 10 consumers)
    - Message Rate: 10,000 messages per second per node.
    - Messages: ~6 million

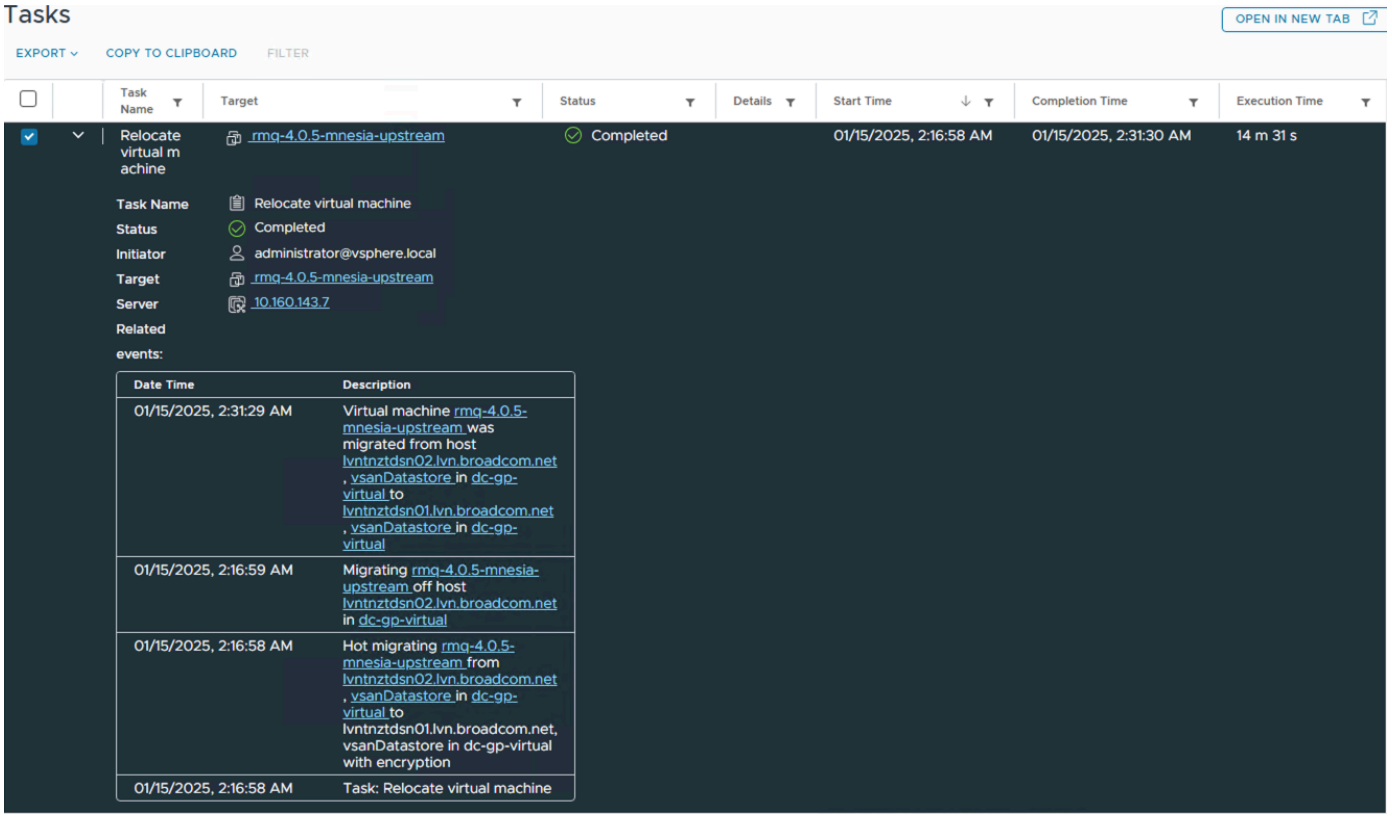
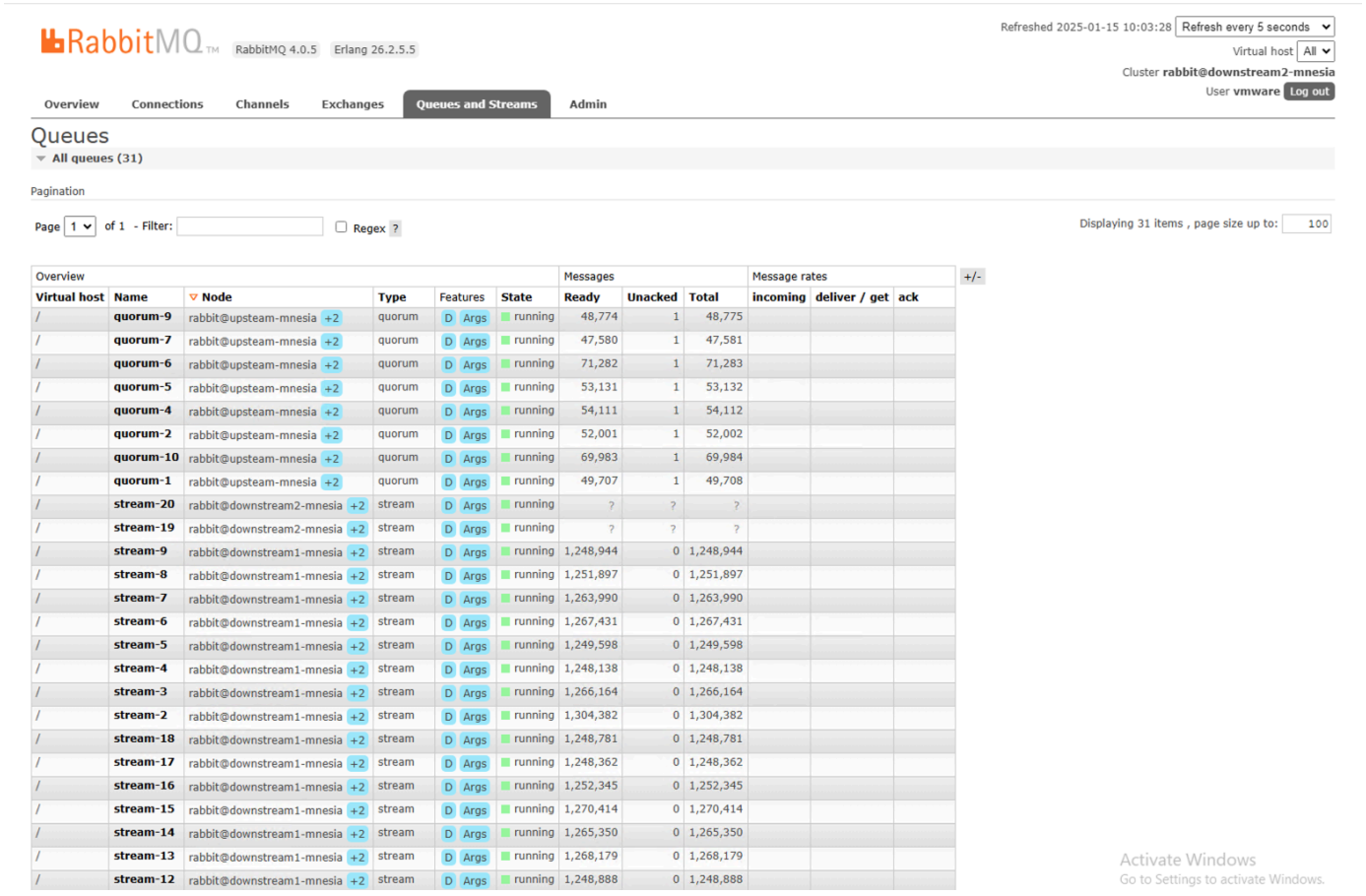


Fig (a) - vMotion migration event of Tanzu RabbitMQ OVA.



RabbitMQ™ RabbitMQ 4.0.5 Erlang 26.2.5.5

Refreshed 2025-01-15 10:03:28 Refresh every 5 seconds
Virtual host All
Cluster rabbit@downstream2-mnesia
User vmware Log out

Overview | Connections | Channels | Exchanges | Queues and Streams | Admin

Queues

All queues (31)

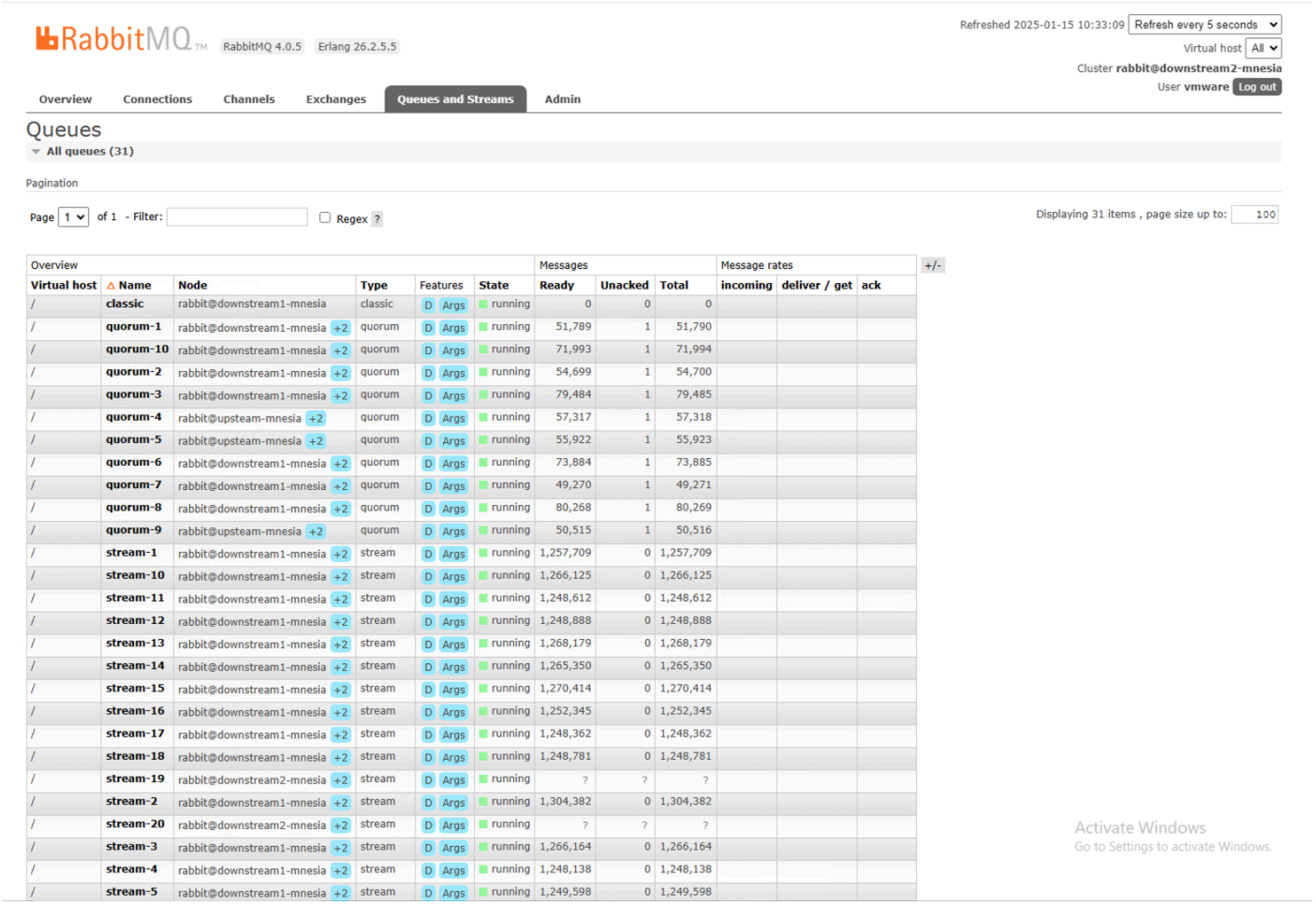
Pagination

Page 1 of 1 - Filter: Regex ?

Displaying 31 items , page size up to: 100

Overview						Messages			Message rates			+/-
Virtual host	Name	Node	Type	Features	State	Ready	Unacked	Total	incoming	deliver / get	ack	
/	quorum-9	rabbit@upsteam-mnesia +2	quorum	D Args	running	48,774	1	48,775				
/	quorum-7	rabbit@upsteam-mnesia +2	quorum	D Args	running	47,580	1	47,581				
/	quorum-6	rabbit@upsteam-mnesia +2	quorum	D Args	running	71,282	1	71,283				
/	quorum-5	rabbit@upsteam-mnesia +2	quorum	D Args	running	53,131	1	53,132				
/	quorum-4	rabbit@upsteam-mnesia +2	quorum	D Args	running	54,111	1	54,112				
/	quorum-2	rabbit@upsteam-mnesia +2	quorum	D Args	running	52,001	1	52,002				
/	quorum-10	rabbit@upsteam-mnesia +2	quorum	D Args	running	69,983	1	69,984				
/	quorum-1	rabbit@upsteam-mnesia +2	quorum	D Args	running	49,707	1	49,708				
/	stream-20	rabbit@downstream2-mnesia +2	stream	D Args	running	?	?	?				
/	stream-19	rabbit@downstream2-mnesia +2	stream	D Args	running	?	?	?				
/	stream-9	rabbit@downstream1-mnesia +2	stream	D Args	running	1,248,944	0	1,248,944				
/	stream-8	rabbit@downstream1-mnesia +2	stream	D Args	running	1,251,897	0	1,251,897				
/	stream-7	rabbit@downstream1-mnesia +2	stream	D Args	running	1,263,990	0	1,263,990				
/	stream-6	rabbit@downstream1-mnesia +2	stream	D Args	running	1,267,431	0	1,267,431				
/	stream-5	rabbit@downstream1-mnesia +2	stream	D Args	running	1,249,598	0	1,249,598				
/	stream-4	rabbit@downstream1-mnesia +2	stream	D Args	running	1,248,138	0	1,248,138				
/	stream-3	rabbit@downstream1-mnesia +2	stream	D Args	running	1,266,164	0	1,266,164				
/	stream-2	rabbit@downstream1-mnesia +2	stream	D Args	running	1,304,382	0	1,304,382				
/	stream-18	rabbit@downstream1-mnesia +2	stream	D Args	running	1,248,781	0	1,248,781				
/	stream-17	rabbit@downstream1-mnesia +2	stream	D Args	running	1,248,362	0	1,248,362				
/	stream-16	rabbit@downstream1-mnesia +2	stream	D Args	running	1,252,345	0	1,252,345				
/	stream-15	rabbit@downstream1-mnesia +2	stream	D Args	running	1,270,414	0	1,270,414				
/	stream-14	rabbit@downstream1-mnesia +2	stream	D Args	running	1,265,350	0	1,265,350				
/	stream-13	rabbit@downstream1-mnesia +2	stream	D Args	running	1,268,179	0	1,268,179				
/	stream-12	rabbit@downstream1-mnesia +2	stream	D Args	running	1,248,888	0	1,248,888				

Fig (b) - Before vMotion migration.



RabbitMQ™ RabbitMQ 4.0.5 Erlang 26.2.5.5

Refreshed 2025-01-15 10:33:09 Refresh every 5 seconds
Virtual host All
Cluster rabbit@downstream2-mnesia
User vmware Log out

Overview | Connections | Channels | Exchanges | Queues and Streams | Admin

Queues

All queues (31)

Pagination

Page 1 of 1 - Filter: Regex ?

Displaying 31 items , page size up to: 100

Overview						Messages			Message rates			+/-
Virtual host	Name	Node	Type	Features	State	Ready	Unacked	Total	incoming	deliver / get	ack	
/	classic	rabbit@downstream1-mnesia	classic	D Args	running	0	0	0				
/	quorum-1	rabbit@downstream1-mnesia +2	quorum	D Args	running	51,789	1	51,790				
/	quorum-10	rabbit@downstream1-mnesia +2	quorum	D Args	running	71,993	1	71,994				
/	quorum-2	rabbit@downstream1-mnesia +2	quorum	D Args	running	54,699	1	54,700				
/	quorum-3	rabbit@downstream1-mnesia +2	quorum	D Args	running	79,484	1	79,485				
/	quorum-4	rabbit@upsteam-mnesia +2	quorum	D Args	running	57,317	1	57,318				
/	quorum-5	rabbit@upsteam-mnesia +2	quorum	D Args	running	55,922	1	55,923				
/	quorum-6	rabbit@downstream1-mnesia +2	quorum	D Args	running	73,884	1	73,885				
/	quorum-7	rabbit@downstream1-mnesia +2	quorum	D Args	running	49,270	1	49,271				
/	quorum-8	rabbit@downstream1-mnesia +2	quorum	D Args	running	80,268	1	80,269				
/	quorum-9	rabbit@upsteam-mnesia +2	quorum	D Args	running	50,515	1	50,516				
/	stream-1	rabbit@downstream1-mnesia +2	stream	D Args	running	1,257,709	0	1,257,709				
/	stream-10	rabbit@downstream1-mnesia +2	stream	D Args	running	1,266,125	0	1,266,125				
/	stream-11	rabbit@downstream1-mnesia +2	stream	D Args	running	1,248,612	0	1,248,612				
/	stream-12	rabbit@downstream1-mnesia +2	stream	D Args	running	1,248,888	0	1,248,888				
/	stream-13	rabbit@downstream1-mnesia +2	stream	D Args	running	1,268,179	0	1,268,179				
/	stream-14	rabbit@downstream1-mnesia +2	stream	D Args	running	1,265,350	0	1,265,350				
/	stream-15	rabbit@downstream1-mnesia +2	stream	D Args	running	1,270,414	0	1,270,414				
/	stream-16	rabbit@downstream1-mnesia +2	stream	D Args	running	1,252,345	0	1,252,345				
/	stream-17	rabbit@downstream1-mnesia +2	stream	D Args	running	1,248,362	0	1,248,362				
/	stream-18	rabbit@downstream1-mnesia +2	stream	D Args	running	1,248,781	0	1,248,781				
/	stream-19	rabbit@downstream2-mnesia +2	stream	D Args	running	?	?	?				
/	stream-2	rabbit@downstream1-mnesia +2	stream	D Args	running	1,304,382	0	1,304,382				
/	stream-20	rabbit@downstream2-mnesia +2	stream	D Args	running	?	?	?				
/	stream-3	rabbit@downstream1-mnesia +2	stream	D Args	running	1,266,164	0	1,266,164				
/	stream-4	rabbit@downstream1-mnesia +2	stream	D Args	running	1,248,138	0	1,248,138				
/	stream-5	rabbit@downstream1-mnesia +2	stream	D Args	running	1,249,598	0	1,249,598				

Fig (c) - After vMotion migration.

- Monitor and Troubleshoot:
    - Deploy monitoring tools (e.g., Prometheus, Grafana) to observe RabbitMQ's performance.
    - Analyze logs for errors or warnings related to vMotion events.

Benefits:

- **Enhanced Flexibility:** Live migrations allow administrators to perform maintenance or hardware upgrades without downtime.
- **Improved Resilience:** Persistent storage and clustering ensure RabbitMQ remains operational during migrations.
- **Optimized Resource Utilization:** vMotion ensures efficient workload distribution across the vSphere cluster.

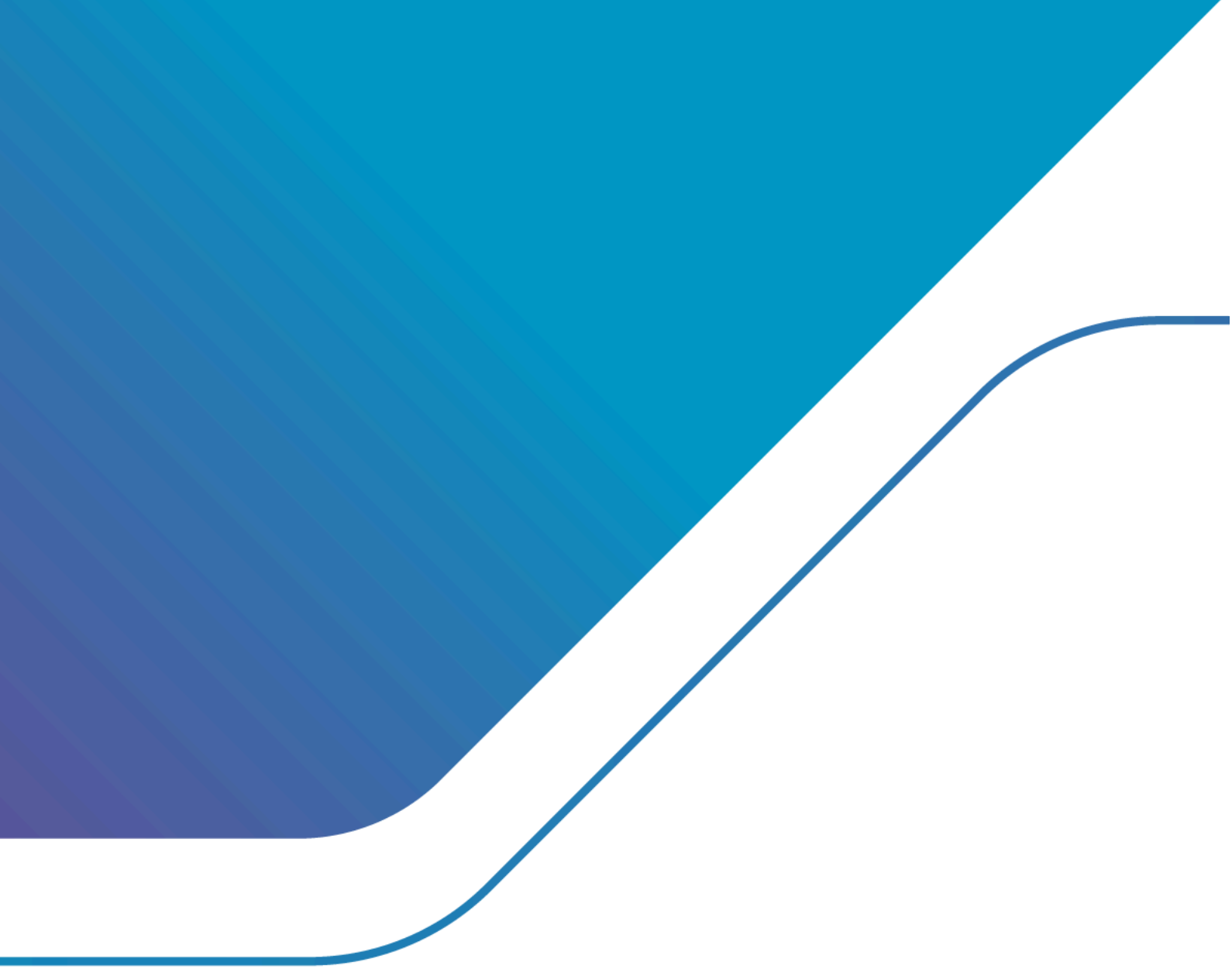
Conclusion:

With proper planning and adherence to best practices, the risks associated with live migrations can be eliminated providing a successful vMotion migration of Tanzu RabbitMQ OVA in vSphere. Thus ensuring reliable message delivery and operational continuity during and after vMotion migration.

Furthermore, RabbitMQ's network partition recovery strategies and Khepri's automatic leader election enable smooth failover and recovery, minimizing service disruptions, providing a resilient and highly available messaging infrastructure for enterprise environments.

References:

- RabbitMQ Official Site: https://www.rabbitmq.com/
- Khepri database: https://www.rabbitmq.com/docs/metadata-store#khepri
- Tanzu RabbitMQ OVA Configuration Guide: https://techdocs.broadcom.com/us/en/vmware-tanzu/data-solutions/tanzu-rabbitmq-ova/4-0/tanzu-rabbitmq-ova-virtual-machine/overview.html
- VMware vSphere Documentation: https://www.vmware.com/products/vsphere.html
- VMware vMotion Documentation: https://docs.vmware.com/en/VMware-vSphere/6.7/com.vmware.vsphere.vcenterhost.doc/GUID-D19EA1CB-5222-49F9-A002-4F8692B92D63.html



Item No: vmw-bc-wp-tech-a4-word-2025 Dec-24