

Never Lose a Message Again: Building Rock-Solid Event-Driven Streaming Architectures with

VMware Tanzu RabbitMQ and Spring

VMware Tanzu – Data Services



VMware Tanzu

Infrastructure for running modern apps and backing services with consistent, conformant Kubernetes everywhere.



GemFire

Fast In-Memory data store for Caching, Transactional and NoSQL support powered by Apache Geode

I need a fast data store



I need to replatform a relational database

SQL

Relational MySQL or Postgres database for Transactional or Analytic data processing



Data Management

Management for Tanzu Data Services instances



I need to drive analytic value of our tons of existing data

Greenplum

Massively Parallel Processing (MPP) Postgres for Big Data store for analytics, Machine Learning and Artificial Intelligence



I need reliable messaging delivery

Rabbit MQ

High throughput broker for reliable messaging delivery



I need flexible and manageable data integrations

Spring Cloud Data Flow

Data integration orchestration service for dynamically building data pipelines

Features

- ✓ Cloud deployed backing-services
- ✓ On-Premise and Multi-Cloud
- ✓ Self – Service
- ✓ Scaling
- ✓ HA - Fault Tolerant
- ✓ Based on open source
- ✓ World Class Support

RabbitMQ

Message Broker

Reliable Messaging

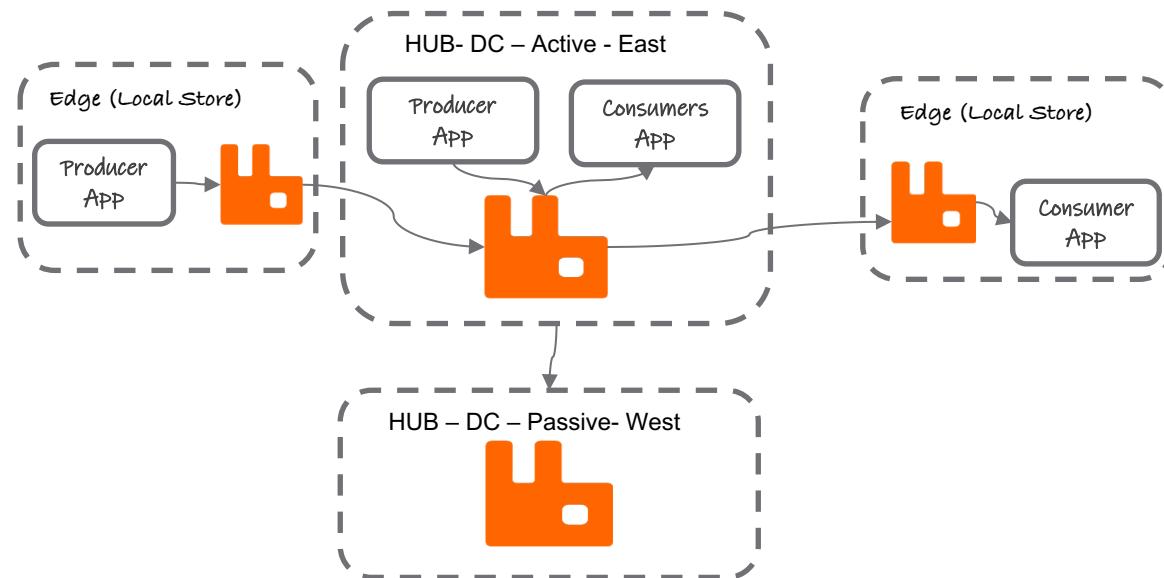
I need reliable
messaging delivery

Tanzu RabbitMQ

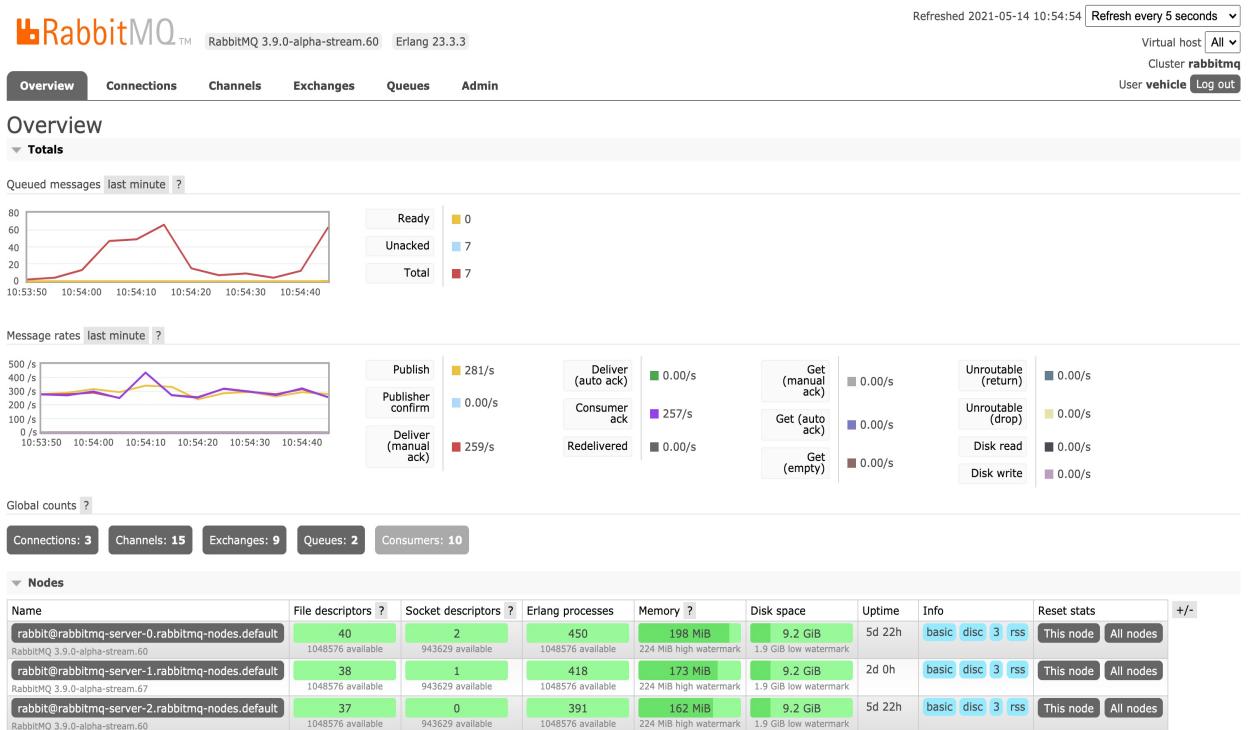
Message Broker

I need reliable messaging delivery

- Increased resiliency**
 - Resilience messaging with redundancy replication
 - High availability deployment as clusters
 - wide area network deliverable disaster recovery
 - Multi-site replication
- User-friendly implementation**
 - Cloud based deployment options
 - Built-in management dashboard user interface



```
apiVersion: rabbitmq.com/v1
kind: RabbitmqCluster
metadata:
  name: rabbitmq
spec:
  replicas: 3
  image: registry.pivotal.io/rabbitmq/vmware-tanzu-rabbitmq
```



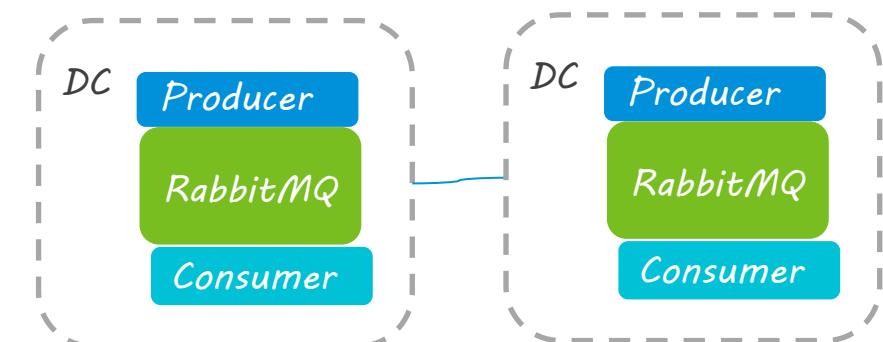
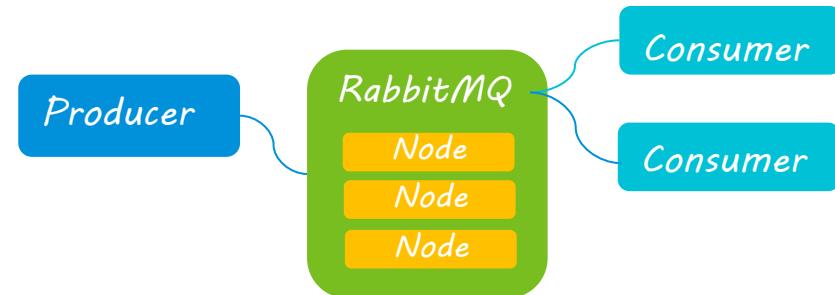
Use Cases

Example Requirements

- As a user I need **FAST** Low latency message processing
 - In memory messaging support
 - Time sensitive messages with strict Service Level Agreements (SLA)
 - Scale out architecture
 - Increase number of consumers to increase throughput of the producers
- As a developer I need a simple **user-friendly** implementation
 - Most popular open-source message broker
 - Develop cross-language messaging with your favorite programming languages, like Java, .NET, PHP, Python, JavaScript, Ruby, Go, and more.
 - Built-in management dashboard user interface
 - Out of box monitoring and alarms
- As a user developer, I need flexible routing
 - Abstraction between producer exchanges and one or more consumer's queues
 - Message routing based on application provided keys (routing keys) and or message headers patterns
 - Distributed local data center architecture or cross data center Wide Area Network replication

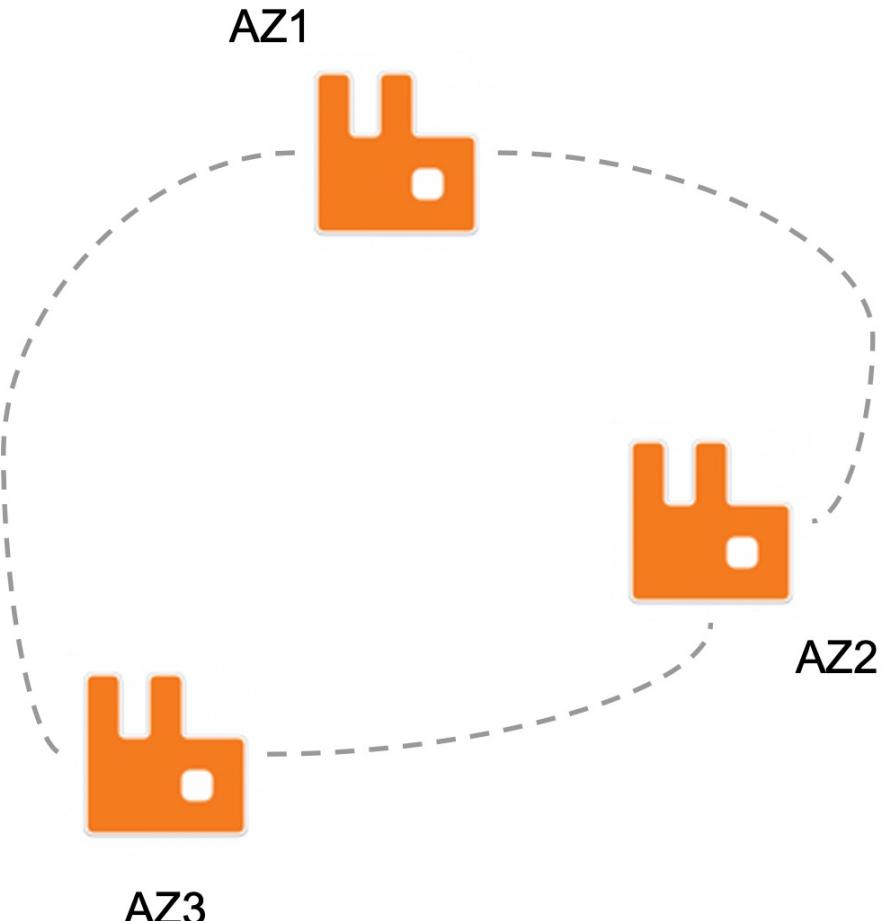
I need reliable messaging delivery

RabbitMQ™



Tanzu RabbitMQ: Compression

Reduce IaaS costs in your high availability cloud deployments



- When would you use this plugin?
 - High availability replicated queues across Azs
- What benefits would you expect to see?
 - IaaS cost savings
- How does it work?
 - RabbitMQ nodes communicate with their peers and CLI tools using dedicated TCP connections, optionally [protected with TLS](#).
 - <https://www.rabbitmq.com/clustering-compression.html>
 - Uses zstd by default, which has the best CPU to latency & compression ratio balance

Tanzu RabbitMQ: Schema Sync Plugin

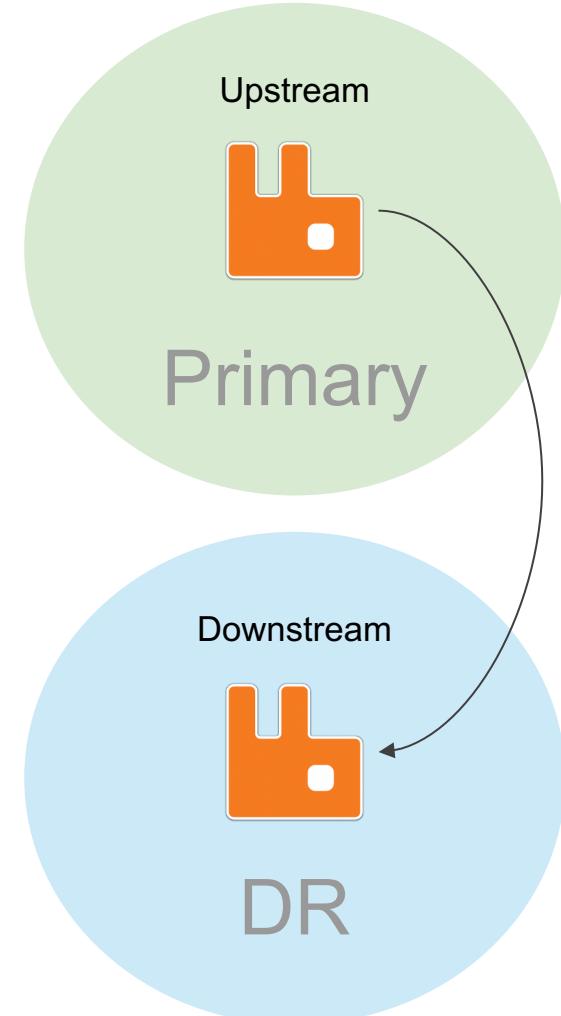
Set up a leader-follower relationship between RabbitMQ clusters

When would you use this plugin?

- You need to easily set up and maintain a “hot standby” RabbitMQ in a different location for failover

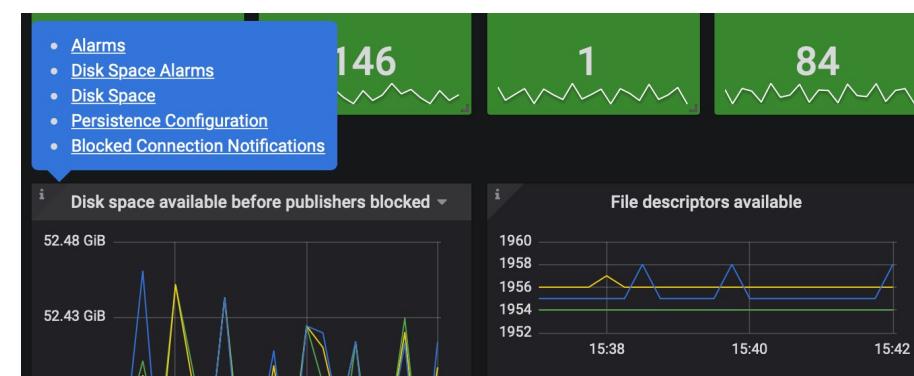
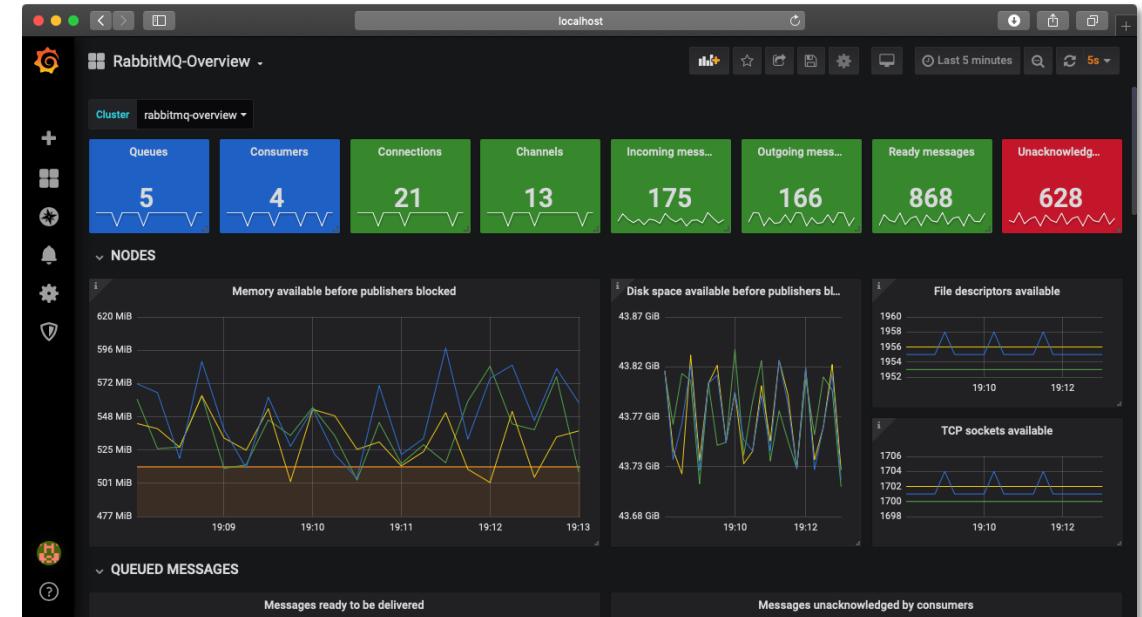
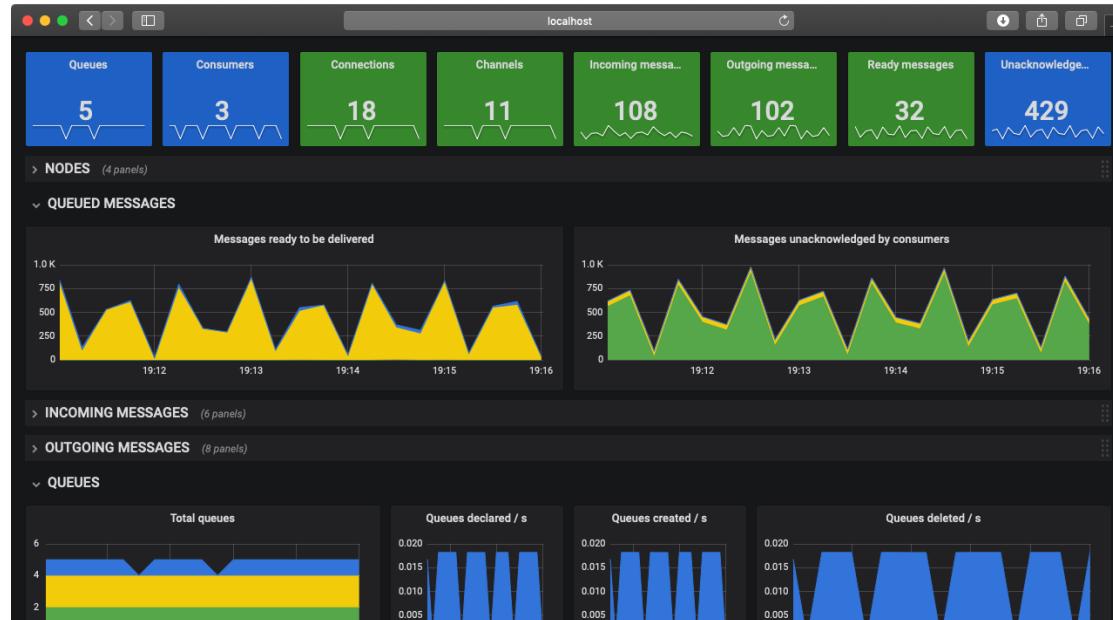
What benefits would you expect to see?

- Increased resiliency (reduced TTR, multiple downstreams possible)
- Reduced time to configure and maintain, saving costs



RabbitMQ Observability

RabbitMQ is observable out of the box, utilizing the most widely adopted open metrics

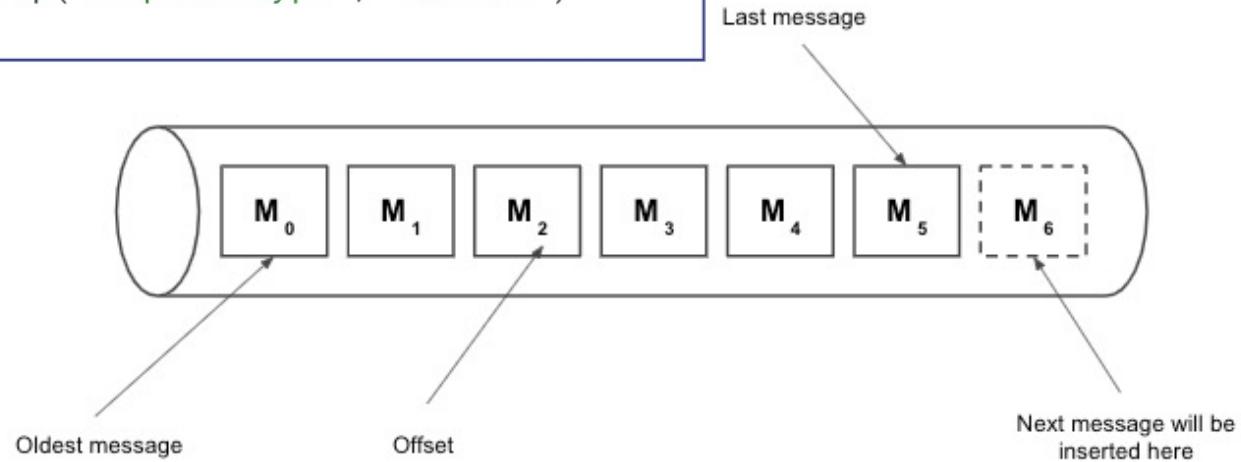


Messaging Streaming

Replay Messages

- Kafka like event logging
- **Large fan-outs:** when several consumer applications need to read the same messages.
- **Replay / Time-traveling:** when consumer applications need to read the whole history of data or from a given point in a stream.
- **Throughput performance:** when higher throughput than with other protocols (AMQP, STOMP, MQTT) is required.
- **Large logs:** when large amount of data need to be stored, with minimal in-memory overhead.

```
channel.queueDeclare(  
    "my-stream",  
    true,          // durable  
    false, false, // not exclusive, not auto-delete  
    Collections.singletonMap("x-queue-type", "stream")  
)
```

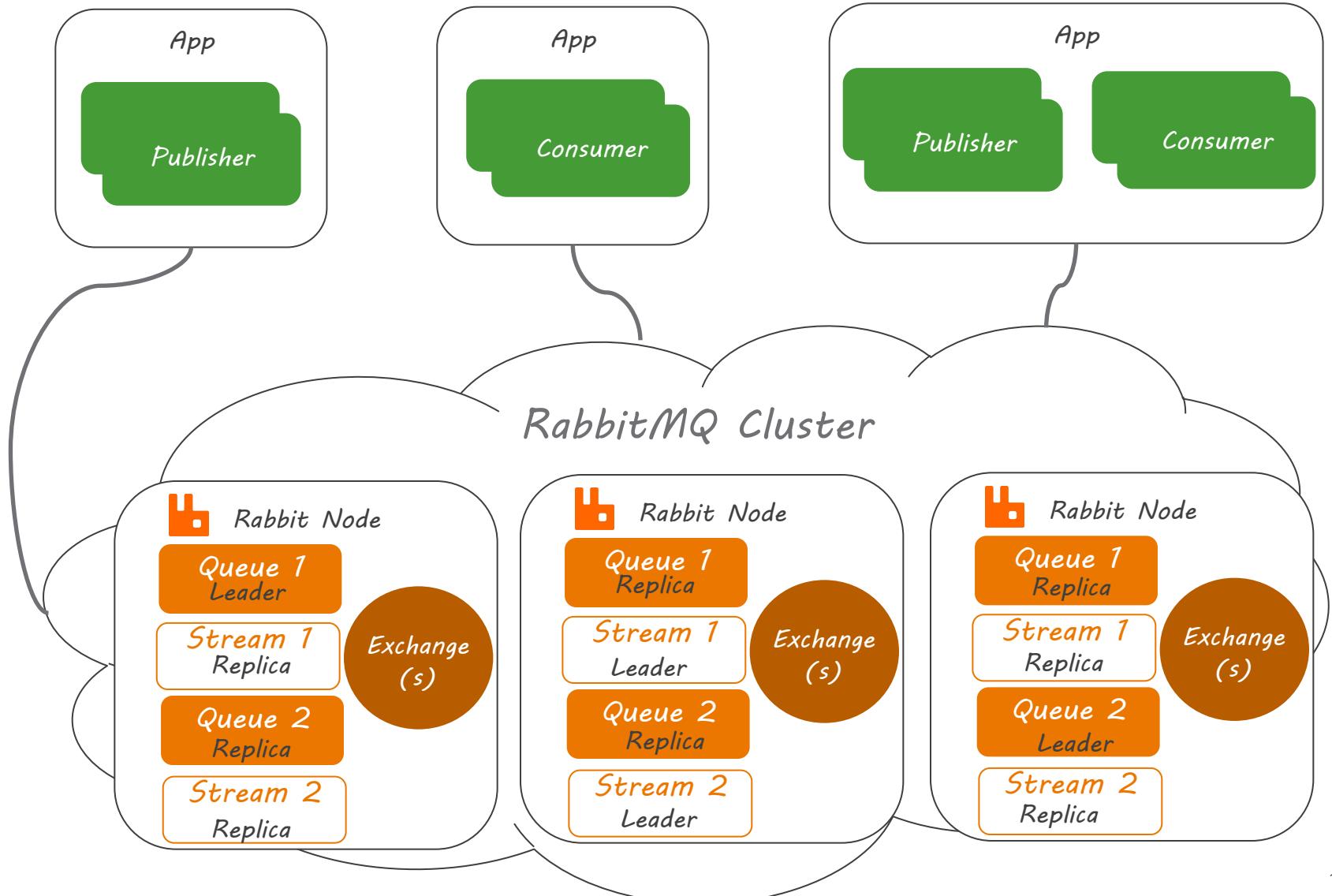


```
channel.basicConsume(  
    "my-stream",  
    false, // not auto-ack  
    Collections.singletonMap("x-stream-offset", 0),  
    (s, delivery) -> { }, // delivery callback  
    s -> { } // cancel callback  
)
```

RabbitMQ

Scalability/Reability

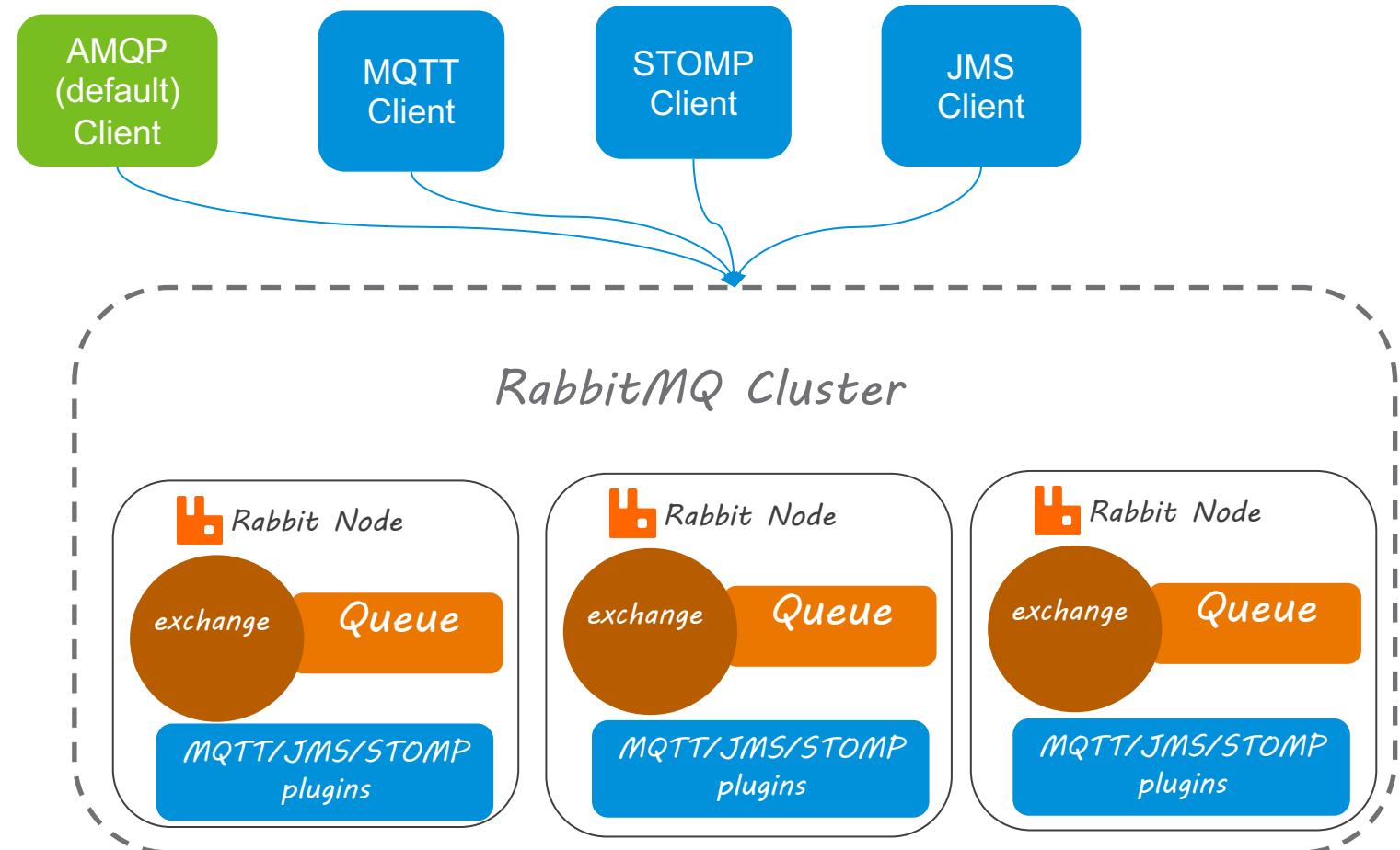
- An odd number of cluster nodes are recommended (ex: 1, 3, 5, 7) by several features that require a consensus between cluster member
- A client can connect to any node.
- Nodes will route operations to the leader.
- Preferred Quorum queues over classic for improved reliability.
- Streams used for high throughput and or replay messages
- Add more queues for Scability across the cluster



RabbitMQ Protocols

RabbitMQ supports various interfaces using plugins that ships in the core distribution.

- Supports TLS connections
- User authentication/authorization
- Store messages in Classic Queues, Quorum Queues or Streams



Spring Cloud Data Flow

Build, Deploy, and Monitor streaming and batch data pipelines

- Spring Cloud Data Flow for VMware Tanzu automates the deployment of data pipelines backed by cloud native applications

Spring Cloud Stream

- Spring Cloud Stream is a framework for building highly scalable event-driven microservices connected with shared messaging systems.

Dashboard

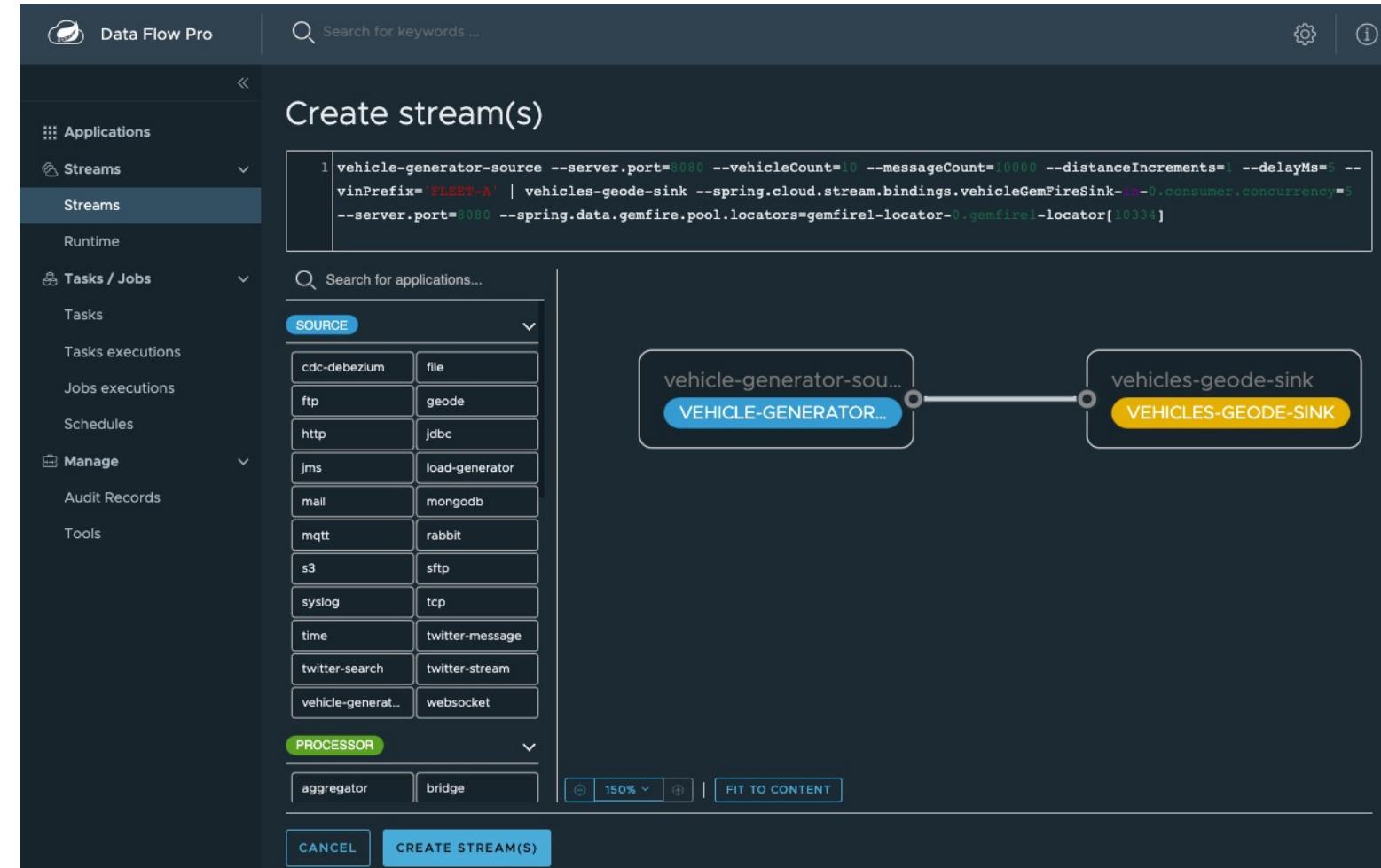
- GUI for managing data pipelines

DSL

- Pipeline definitions language similar to UNIX commands
 - Ex: file | s3

REST API & shell interface

```
dataflow:>stream list
```



Exercises

Lab 1 – Kubernetes install

Lab 2 - Create a RabbitMQ Cluster with HA

Lab 3 – Provisioning using the RabbitMQ Topology Operation

Lab 4 – Messaging

Lab 5 - Spring DataFlow