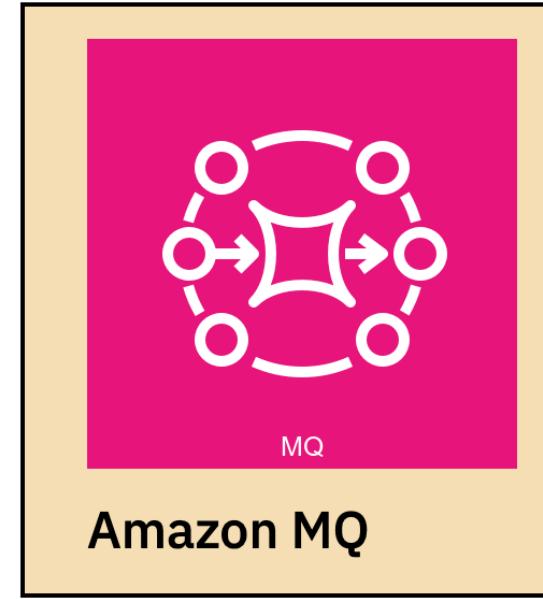


MQ

Amazon MQ

Table of Contents



- 1. What is Amazon MQ?
- 2. Difference between Amazon SQS/SNS and MQ
- 3. Brokers
 - 4. Maintaining an Amazon MQ Broker
 - 5. Active MQ
 - 6. Amazon MQ for ActiveMQ Broker Types
 - 7. Amazon MQ single-instance broker
 - 8. Amazon MQ single-instance broker - with EBS
 - 9. Amazon MQ active/standby broker for high availability
 - 10. Cross-Region data replication for Amazon MQ for ActiveMQ
- 11. RabbitMQ
 - 12. Amazon MQ for RabbitMQ Broker Types
 - 13. Single-instance broker - RabbitMQ
 - 14. Cluster deployment for high availability

AWS Services

EC2 Instances (Applications)



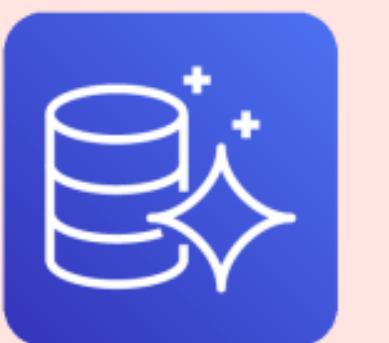
3. Send/Receive messages

S3 Bucket (Data Storage)



4. Stores message data

Aurora Database (Metadata Storage)



5. Stores metadata

Route 53 (DNS Service)

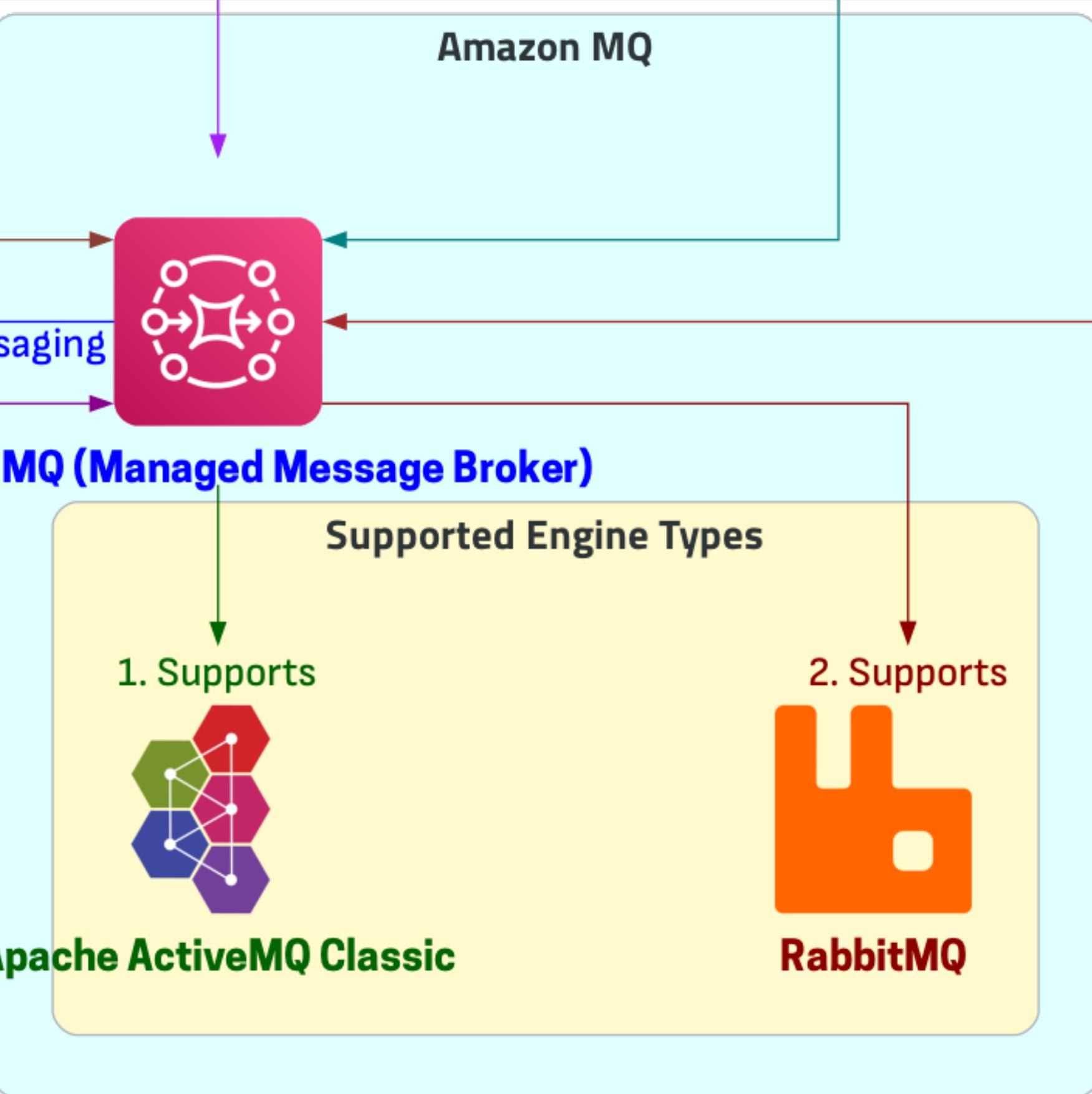


6. Provides DNS

Elastic Load Balancing



7. Distributes traffic



1. Managed message broker service

Easy migration to cloud

Enables communication between applications

2. Message broker functionality

Connects software applications and components

Supports various programming languages

Works with different operating systems

Uses formal messaging protocols

AWS Services

EC2 Instances (Applications)



3. Send/Receive messages

S3 Bucket (Data Storage)



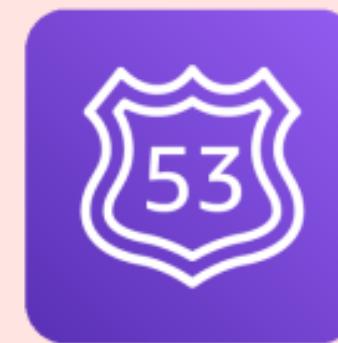
4. Stores message data

Aurora Database (Metadata Storage)



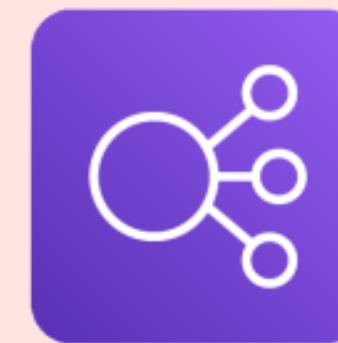
5. Stores metadata

Route 53 (DNS Service)

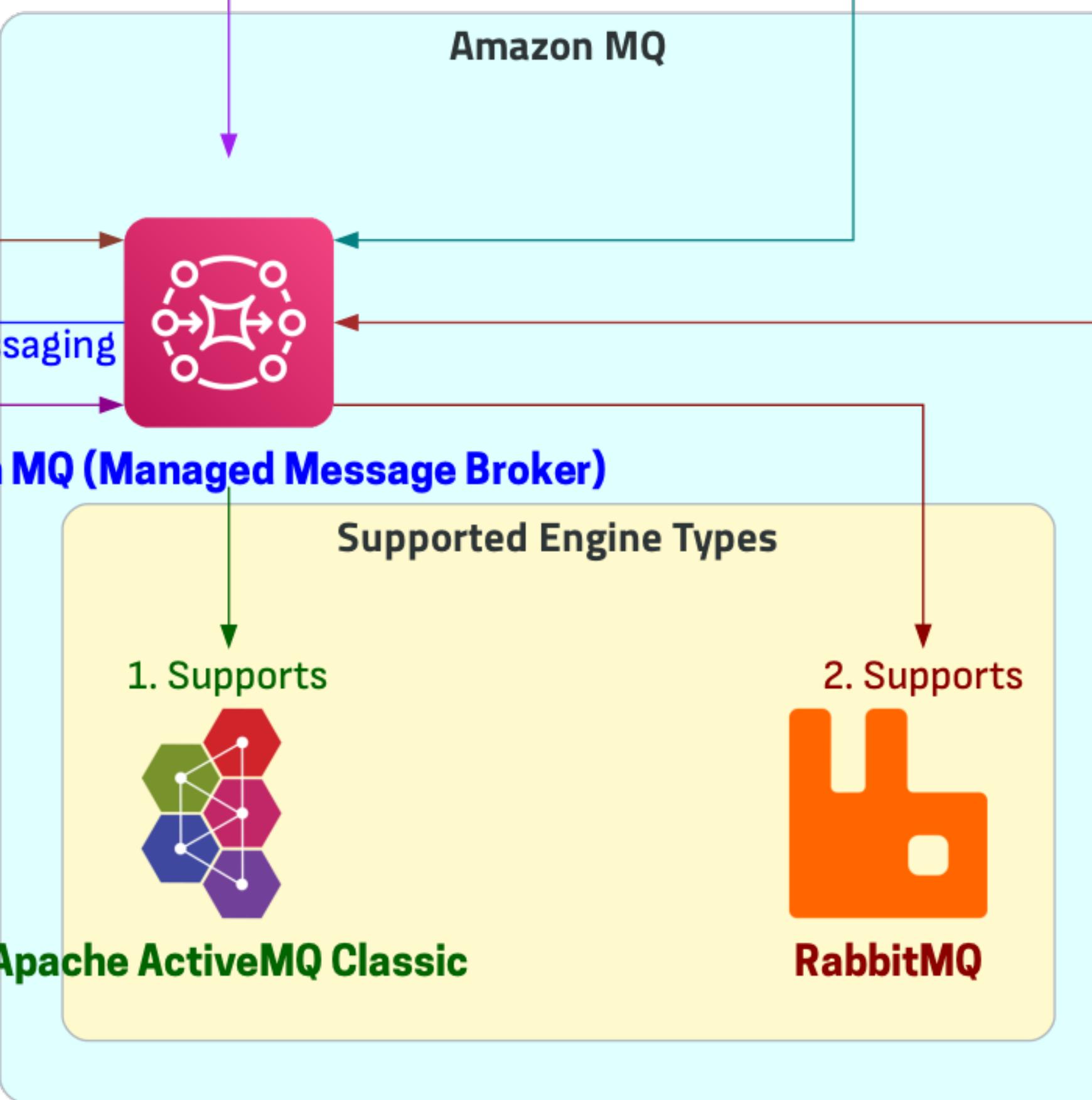


6. Provides DNS

Elastic Load Balancing



7. Distributes traffic



3. 🚂 Supported engine types

Apache ActiveMQ Classic

RabbitMQ

4. 🎨 Compatibility

👉 Works with existing applications and services

5. 🛡️ Managed service benefits

🚫 No need to manage own messaging system

🔧 No operation required

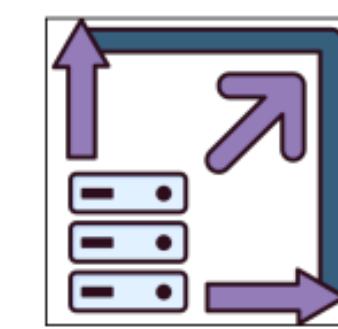
⌚ No maintenance needed

Difference between Amazon SQS/SNS and MQ

Amazon SQS (Simple Queue Service)



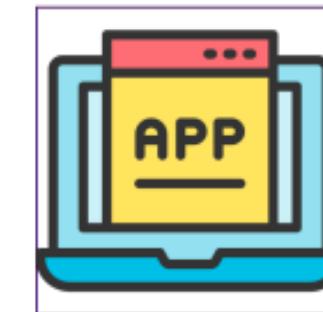
6a. Simple API



4a. Provides

High Scalability

7. Beneficial for



5a. Use for queues



New Applications

6b. Simple API

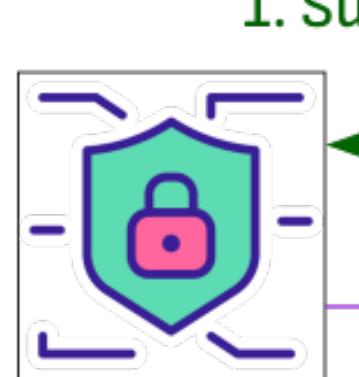
5b. Use for topics

Amazon SNS (Simple Notification Service)



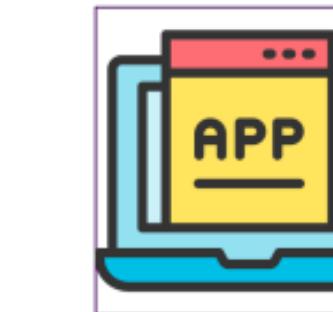
Amazon MQ (Managed Message Broker)

3. Migrate to



1. Supports

JMS, AMQP, MQTT, OpenWire, STOMP



2. Compatible with

Existing Applications (Migration Scenario)

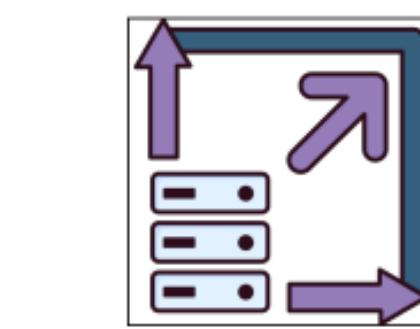
1. 🚂 Amazon MQ	
🔧 Managed message broker service	🛠 APIs: JMS
⚡ Compatibility with popular brokers	🌐 Protocols: AMQP, MQTT, OpenWire, STOMP
🔄 Recommended for migrations	🏭 Existing applications 🔗 Reliant on specific APIs/protocols

1. Supports

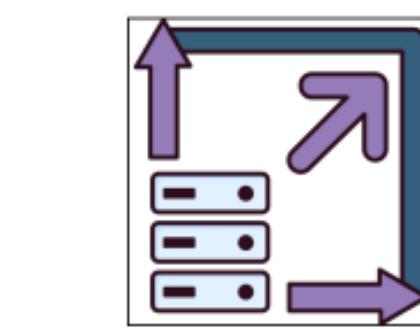
2. Compatible with

3. Migrate to

4b. Provides

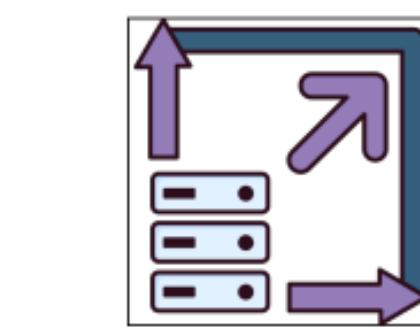


High Scalability



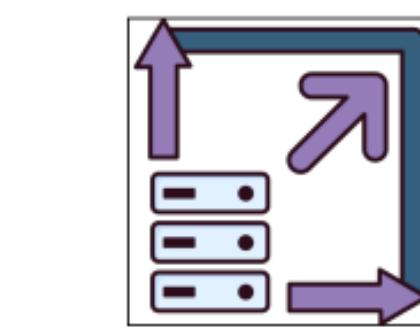
High Scalability

4b. Provides



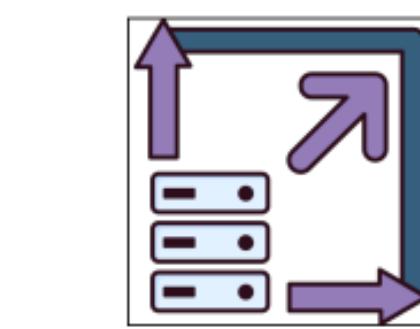
High Scalability

4b. Provides



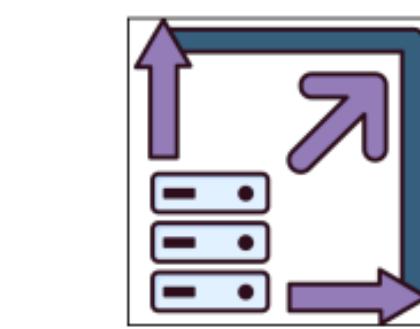
High Scalability

4b. Provides



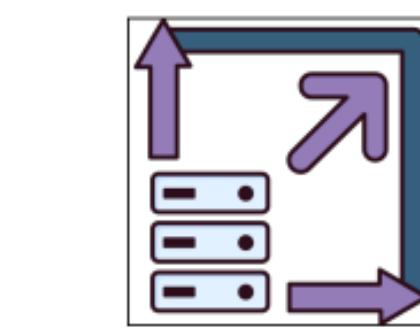
High Scalability

4b. Provides



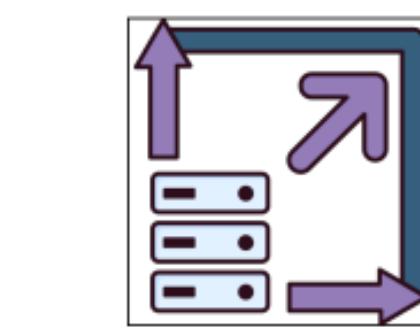
High Scalability

4b. Provides



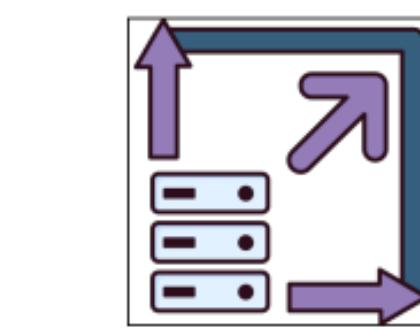
High Scalability

4b. Provides



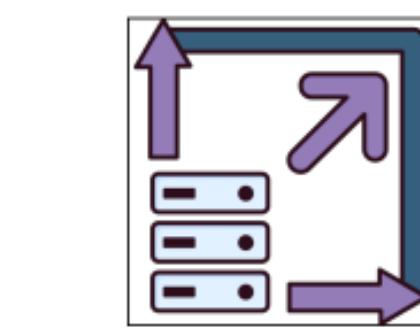
High Scalability

4b. Provides



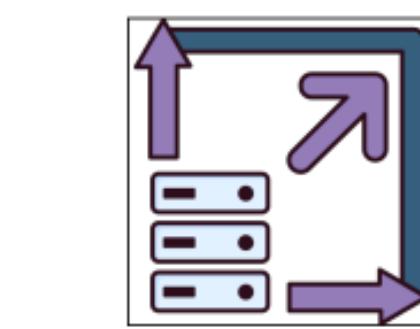
High Scalability

4b. Provides



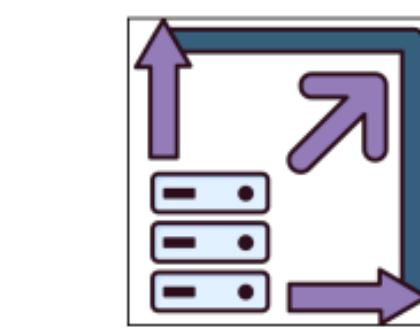
High Scalability

4b. Provides



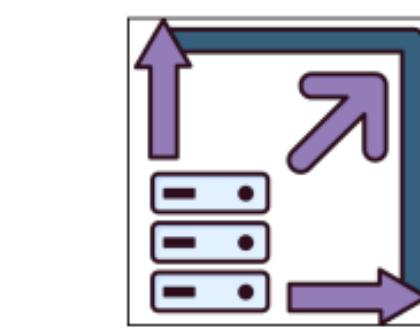
High Scalability

4b. Provides



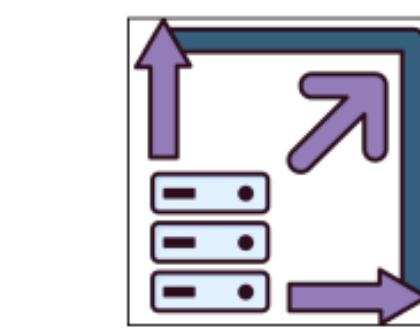
High Scalability

4b. Provides



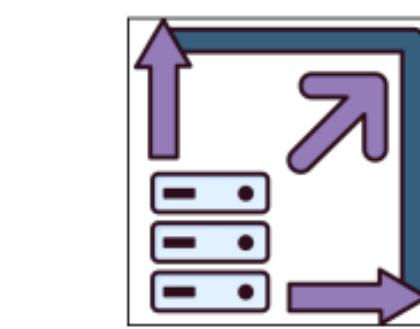
High Scalability

4b. Provides



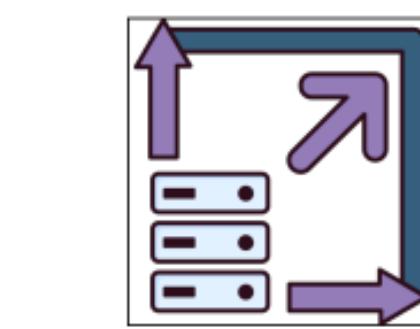
High Scalability

4b. Provides



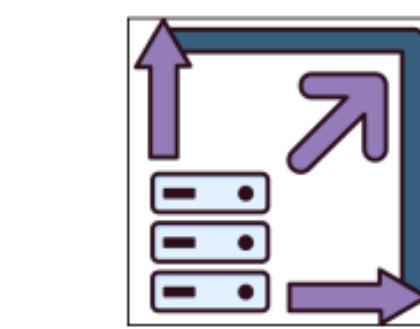
High Scalability

4b. Provides



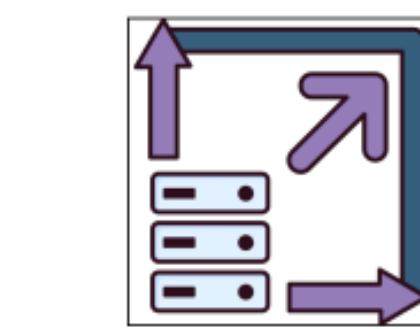
High Scalability

4b. Provides



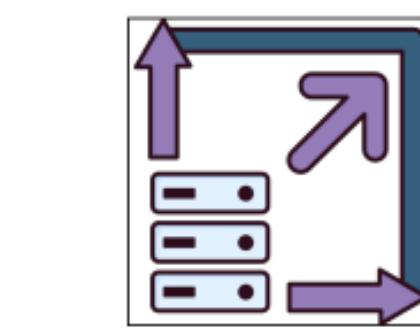
High Scalability

4b. Provides



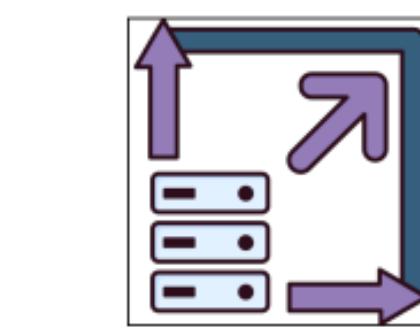
High Scalability

4b. Provides



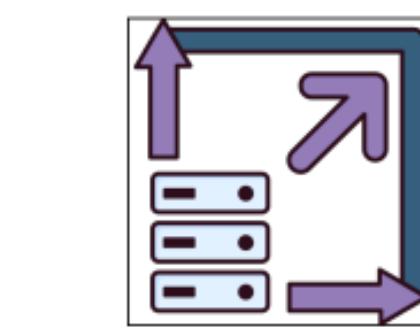
High Scalability

4b. Provides



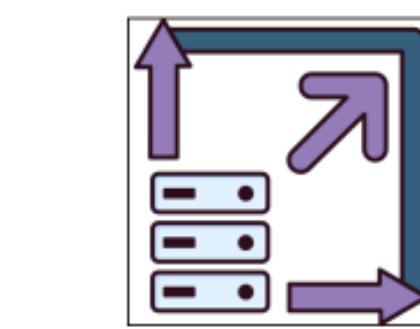
High Scalability

4b. Provides



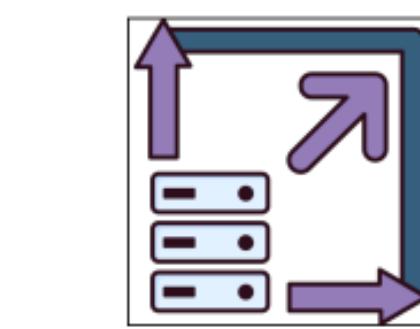
High Scalability

4b. Provides



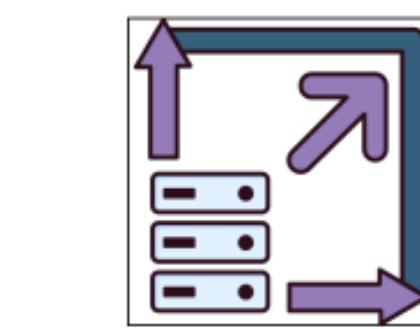
High Scalability

4b. Provides



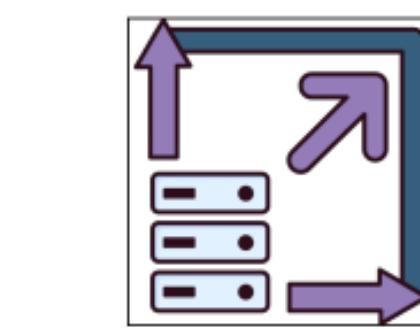
High Scalability

4b. Provides



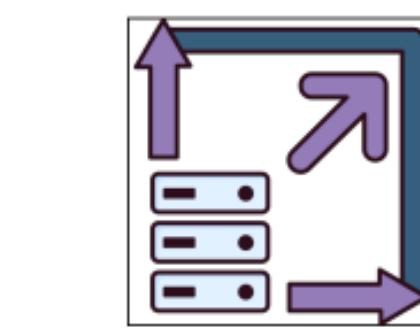
High Scalability

4b. Provides



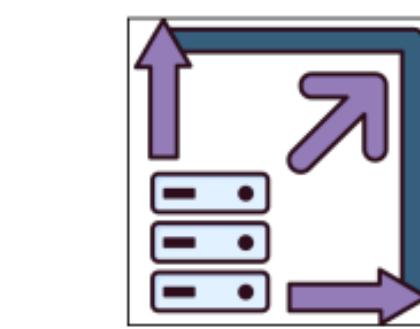
High Scalability

4b. Provides



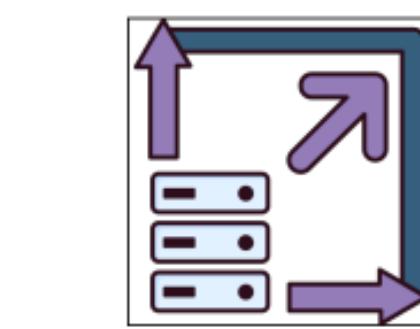
High Scalability

4b. Provides



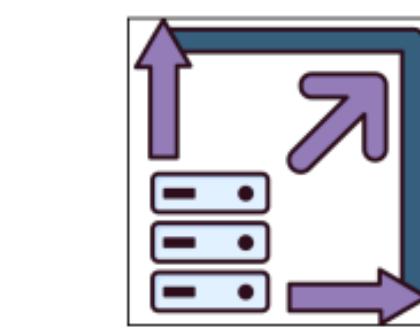
High Scalability

4b. Provides



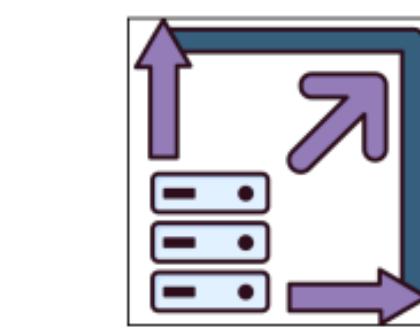
High Scalability

4b. Provides



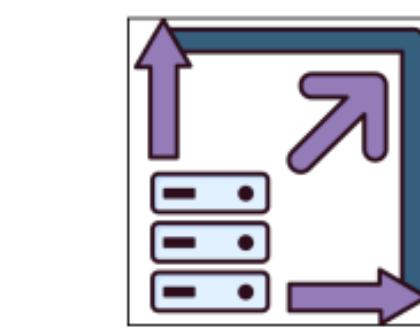
High Scalability

4b. Provides



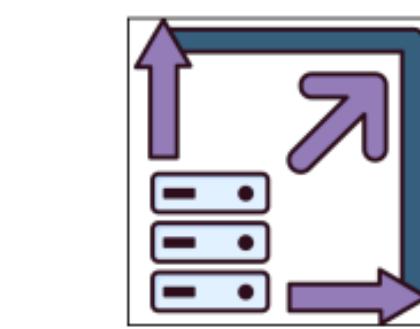
High Scalability

4b. Provides



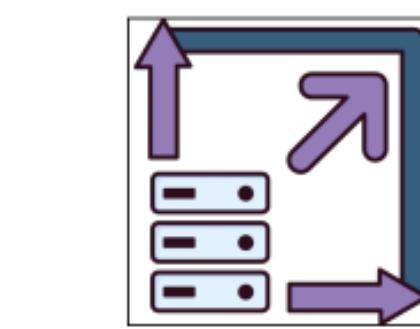
High Scalability

4b. Provides



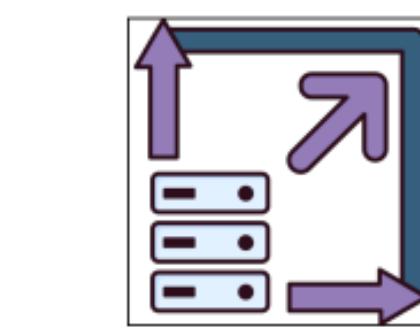
High Scalability

4b. Provides



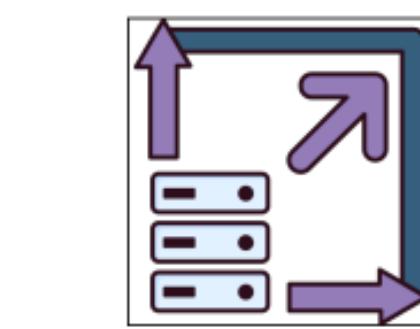
High Scalability

4b. Provides



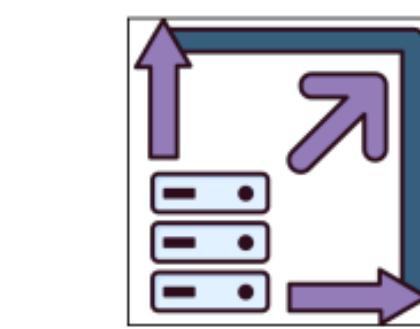
High Scalability

4b. Provides



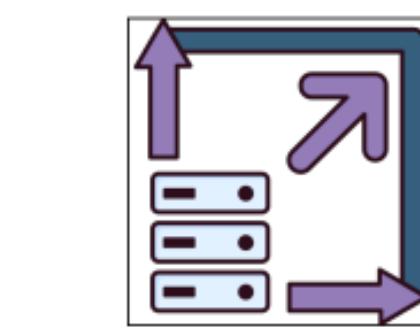
High Scalability

4b. Provides



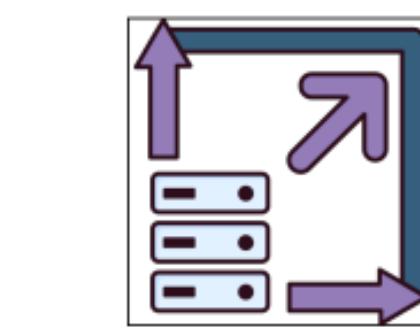
High Scalability

4b. Provides



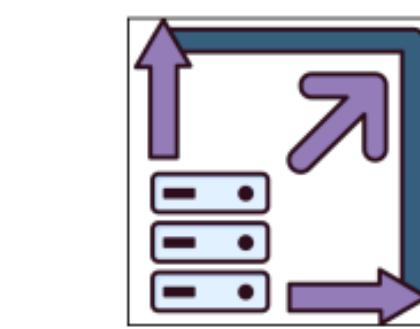
High Scalability

4b. Provides



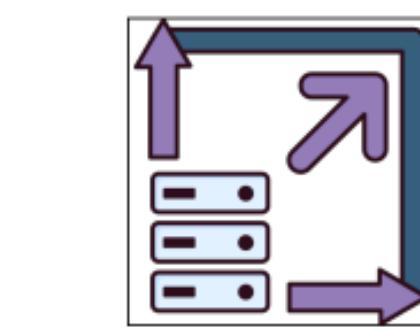
High Scalability

4b. Provides



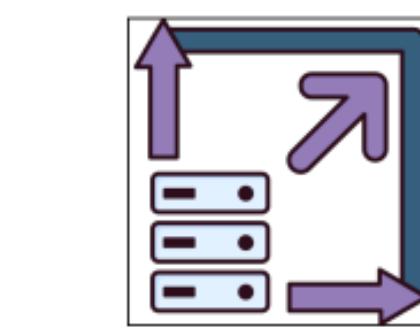
High Scalability

4b. Provides



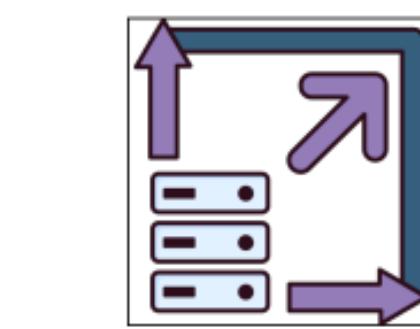
High Scalability

4b. Provides



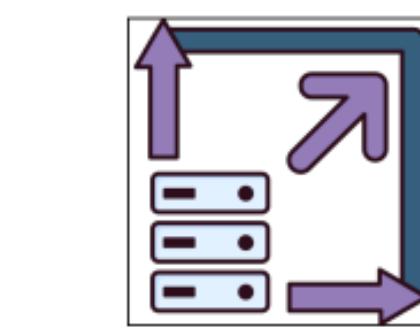
High Scalability

4b. Provides



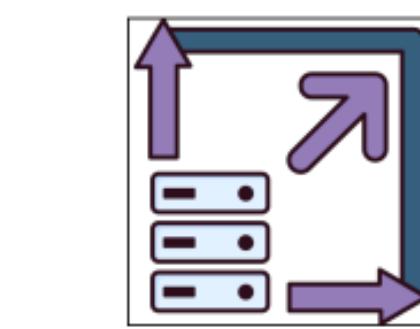
High Scalability

4b. Provides



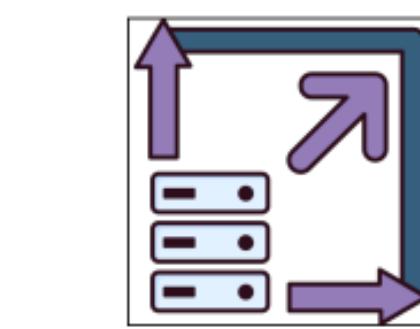
High Scalability

4b. Provides



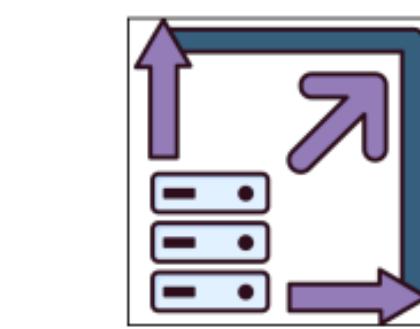
High Scalability

4b. Provides



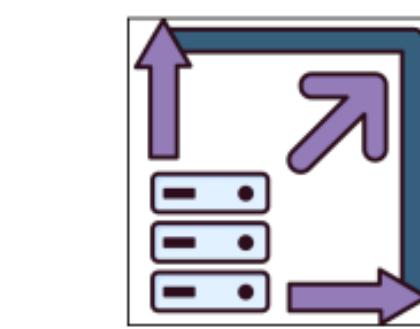
High Scalability

4b. Provides



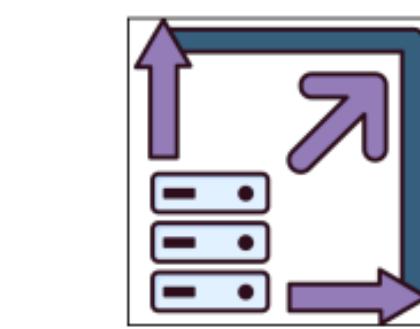
High Scalability

4b. Provides



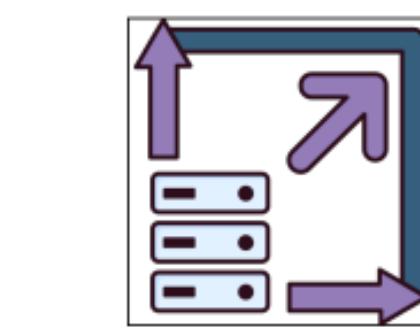
High Scalability

4b. Provides



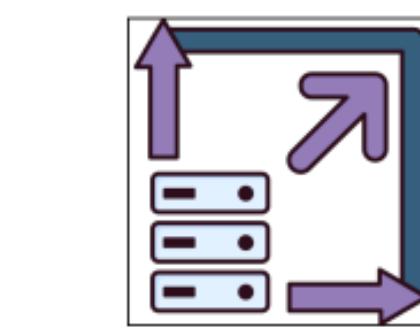
High Scalability

4b. Provides



High Scalability

4b. Provides

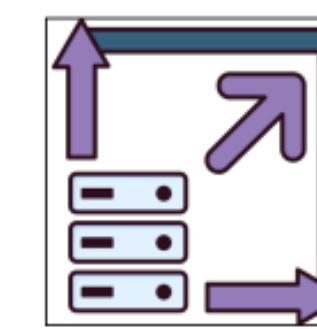


Difference between Amazon SQS/SNS and MQ

Amazon SQS (Simple Queue Service)



6a. Simple API



4a. Provides

High Scalability

7. Beneficial for



5a. Use for queues



New Applications

6b. Simple API

5b. Use for topics



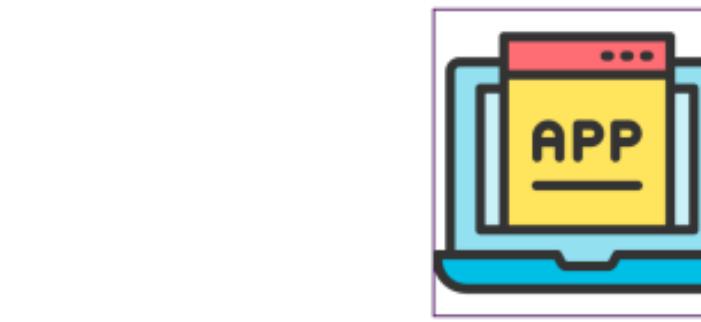
Amazon MQ (Managed Message Broker)

3. Migrate to



1. Supports

JMS, AMQP, MQTT, OpenWire, STOMP



2. Compatible with

Existing Applications (Migration Scenario)

2. 📡 Amazon SQS & SNS	
Queue and topic services	
📈 Highly scalable	🚀 Nearly unlimited scalability
🔧 Simple to use	🛠 No broker setup required ⚡ Simple APIs
NEW Recommended for new applications	

Maintaining an Amazon MQ Broker

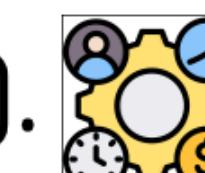
 Amazon MQ periodically performs maintenance on the hardware, operating system, or engine software of a message broker. 

Maintenance Duration

- Maintenance can last up to **two hours**
- Duration depends on scheduled operations for your message broker

Minimizing Downtime



To minimize downtime during maintenance, select a broker deployment mode with high availability across multiple Availability Zones (AZ). 

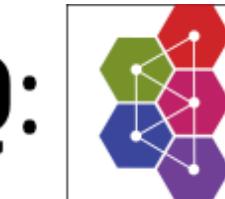


Multi-AZ Deployment Modes



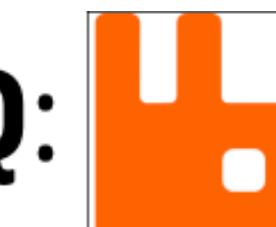
Amazon MQ offers different Multi-AZ deployment modes based on the broker engine type:

1. Amazon MQ for ActiveMQ:



- Provides **active/standby** deployments for high availability
- Maintenance performed one instance at a time
- At least one instance remains available
- Option to configure a network of brokers with scattered maintenance windows

2. Amazon MQ for RabbitMQ:



- Offers **cluster** deployments for high availability
- Maintenance performed one node at a time
- Keeps at least two running nodes at all times

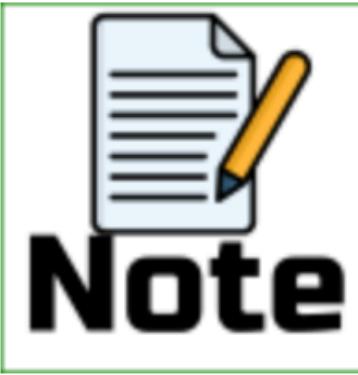
JUL
17

Scheduling Maintenance



You can schedule maintenance:

- Once a week
- At a specified time
- Window lasts up to two hours



Note

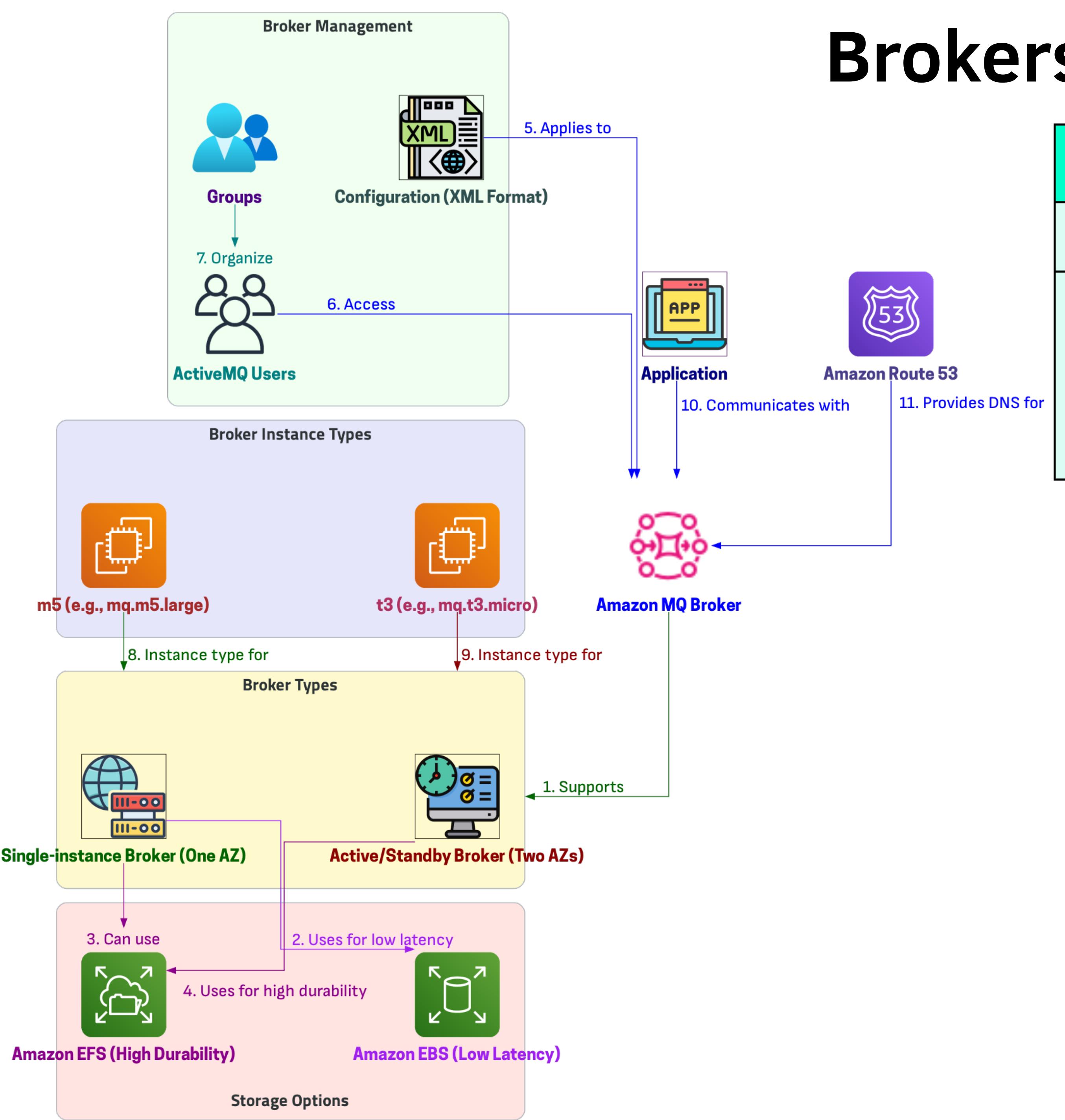
This schedule sets the window for Amazon MQ to start and complete maintenance actions.



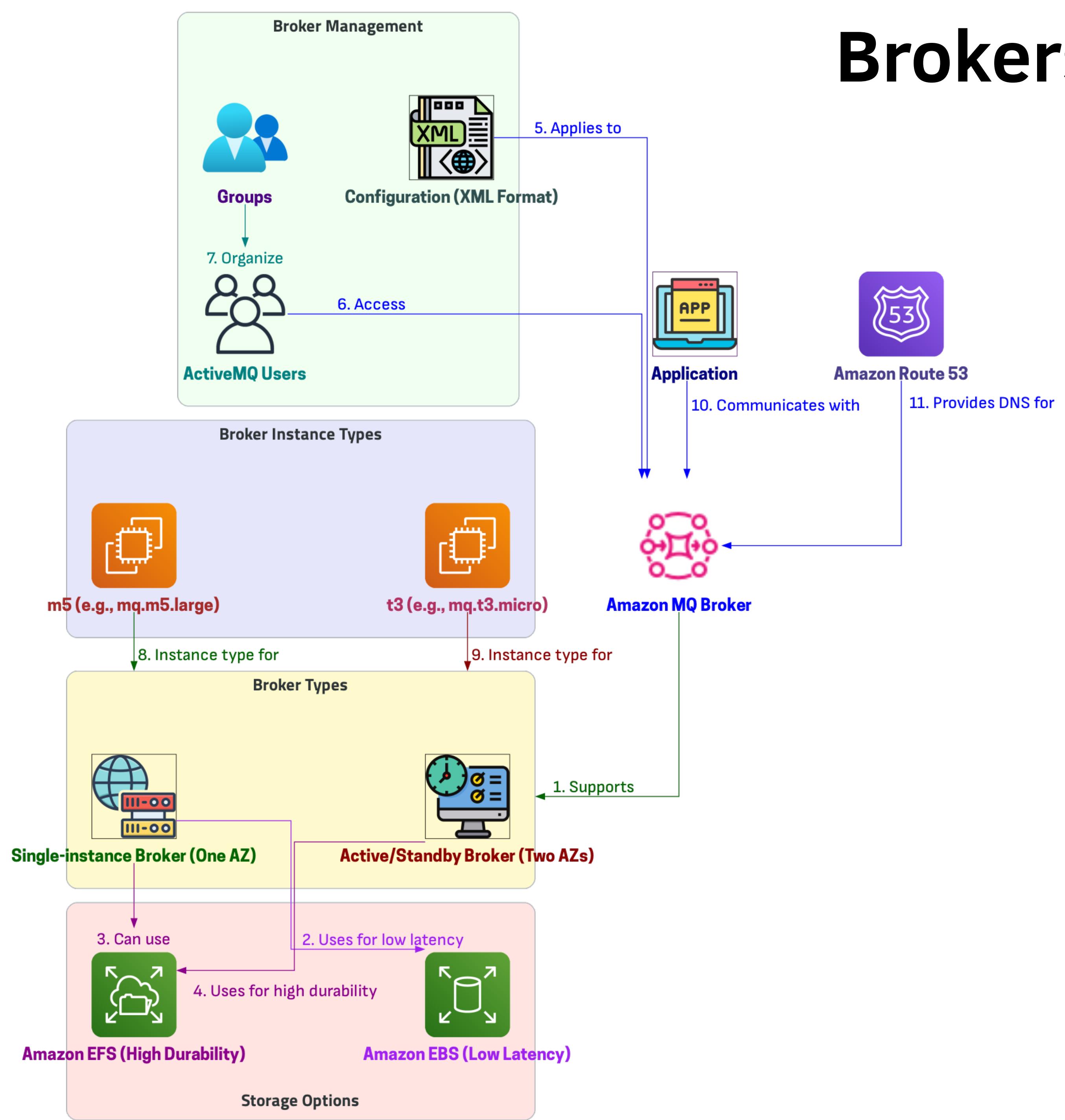
Important

Active MQ

Brokers

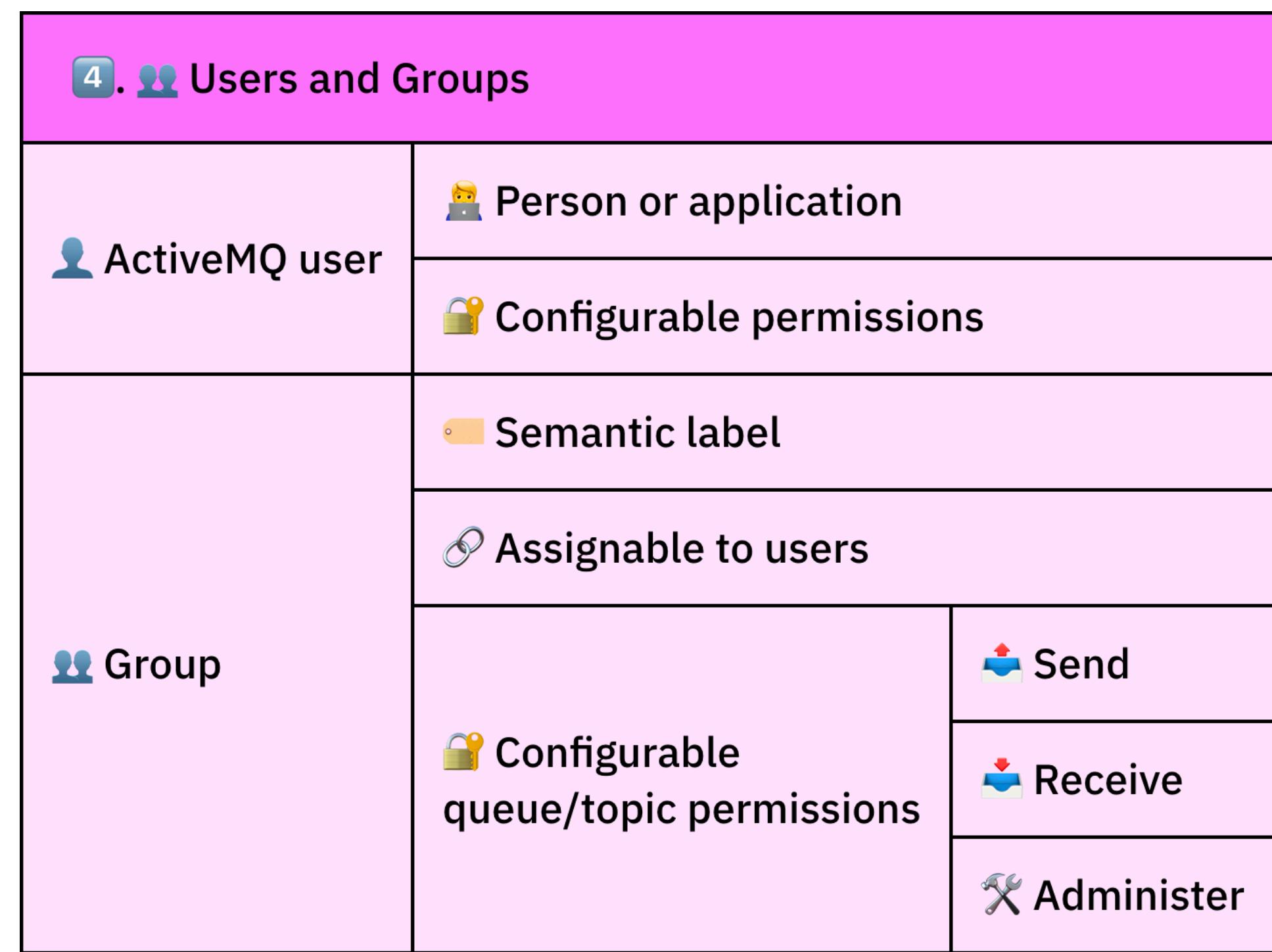
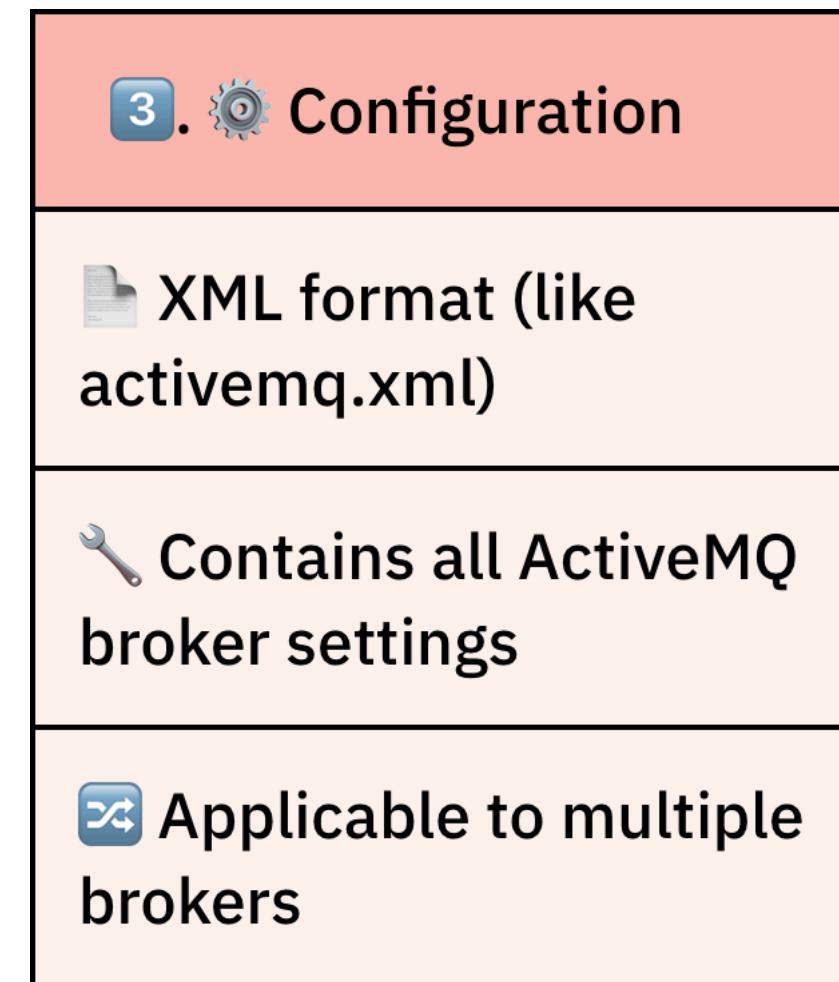
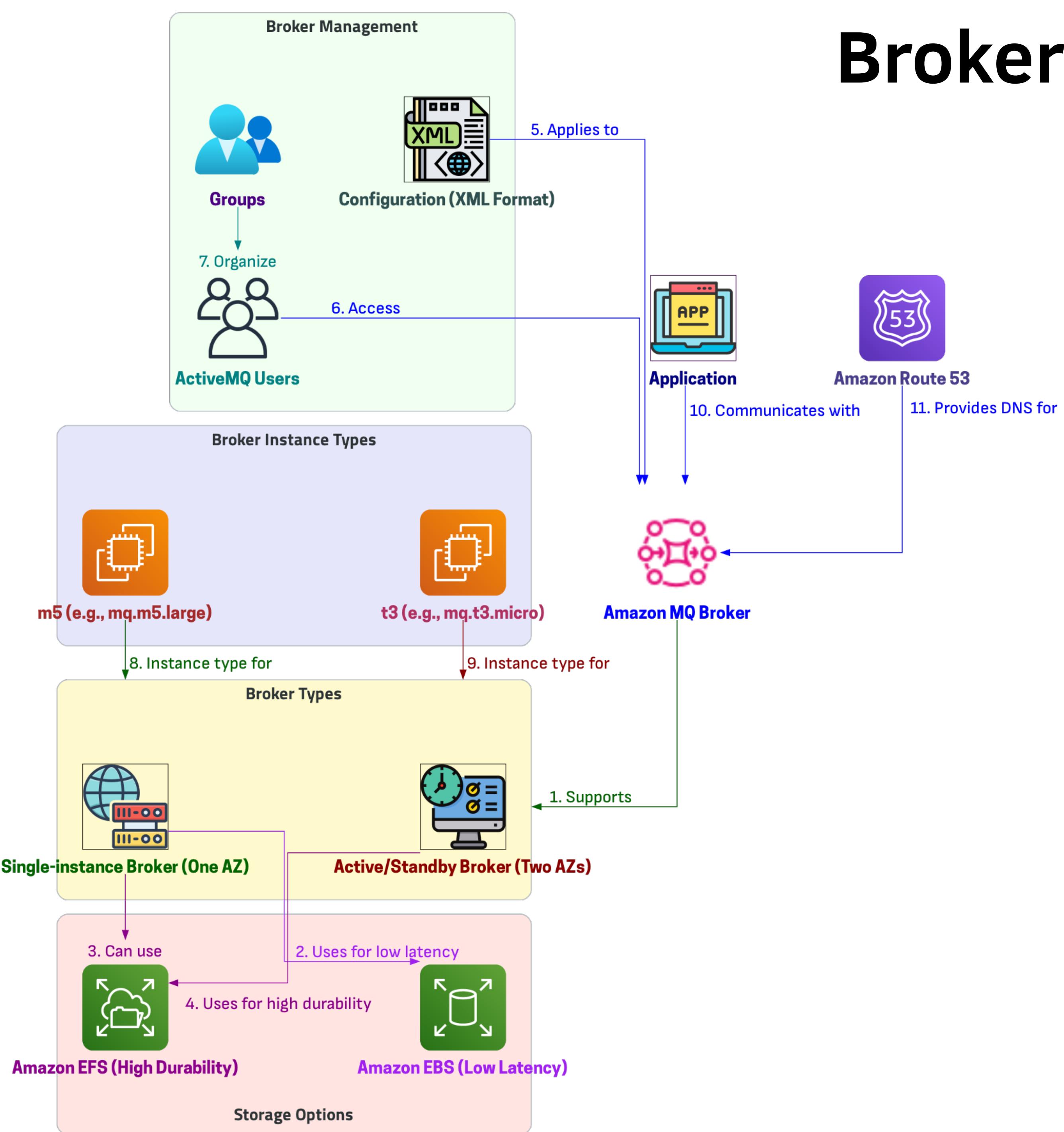


Brokers

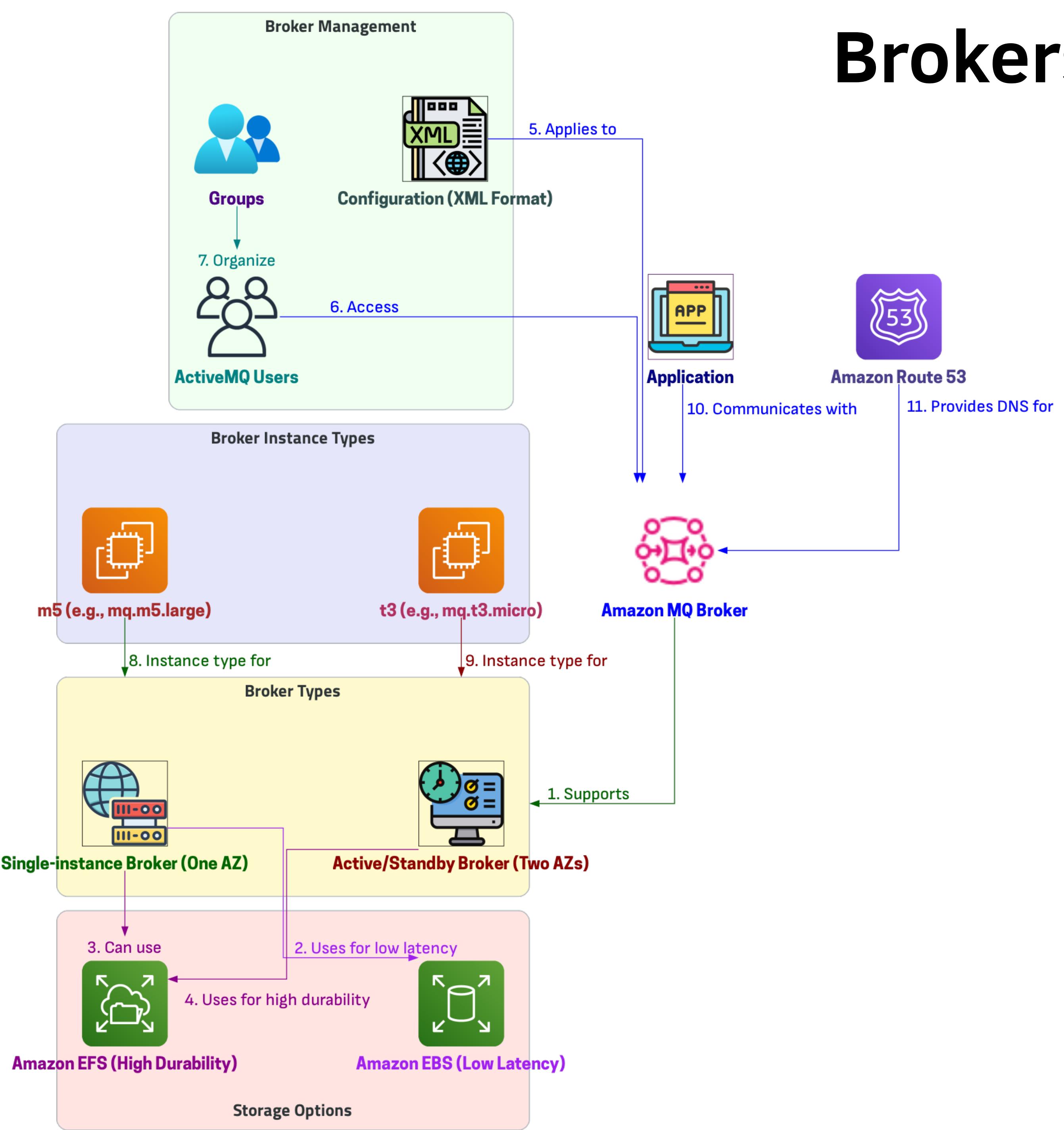


2. ⚡ Broker Types	
Single-instance broker	<ul style="list-style-type: none"> 1 One broker, one Availability Zone 🔗 Communicates with application 💾 Uses EBS or EFS storage
Active/standby broker	<ul style="list-style-type: none"> 2 Two brokers, two Availability Zones 🔄 Redundant pair configuration 🔗 Synchronous communication with application 💾 Uses EFS storage

Brokers



Brokers



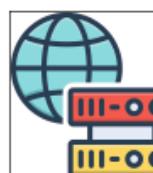
5. Storage Options	
Amazon EFS (default)	High durability Replication across multiple AZs
Amazon EBS	Low latency High throughput

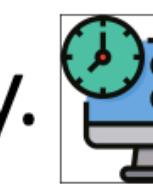


Amazon MQ for ActiveMQ Broker Types



Amazon MQ for ActiveMQ offers two types of broker deployments:

1.  **Single-instance brokers:** A standalone broker deployment. 

2.  **Active/standby brokers:** A redundant pair of brokers for high availability. 



 For both deployment modes, Amazon MQ ensures **high durability** by storing data redundantly.



Amazon MQ single-instance broker

1. Architecture

1 One broker in one Availability Zone

🔗 Communicates with application

2. Storage Options

Amazon EFS

🔒 High durability

↑ High availability

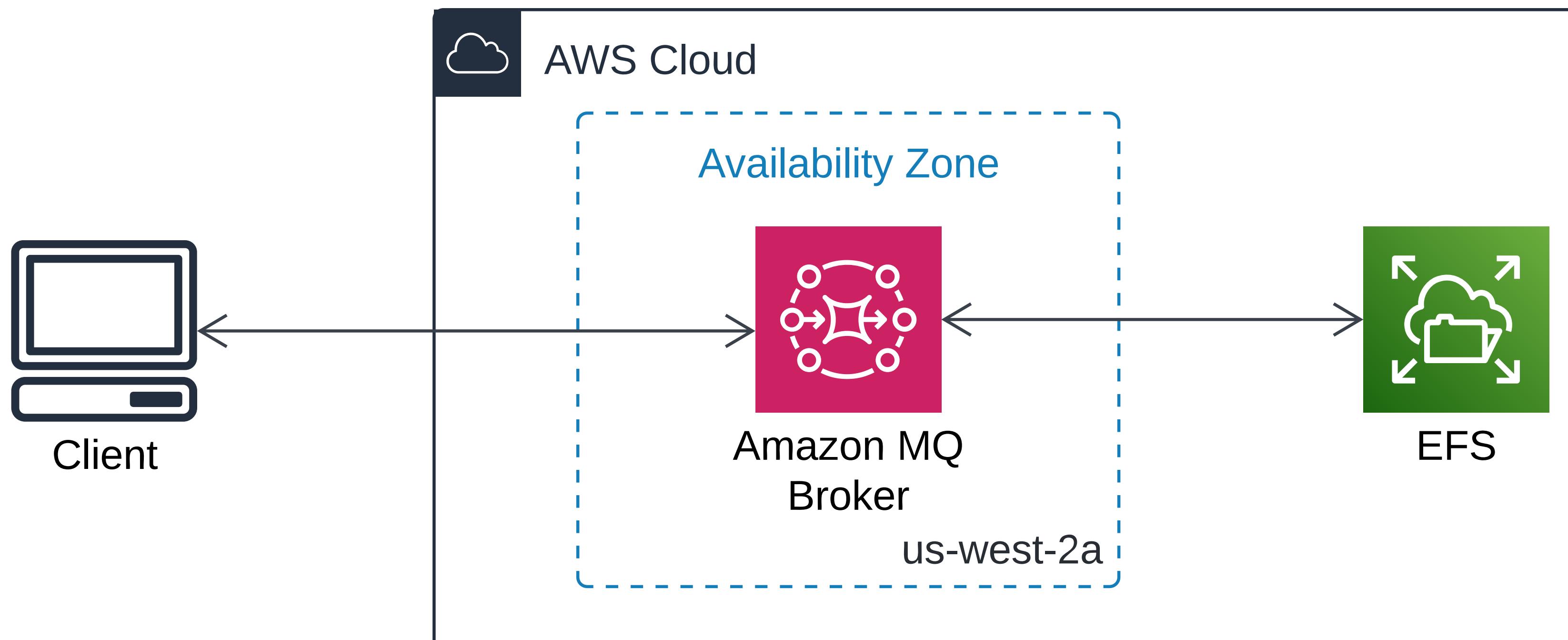
🌐 Data redundancy across multiple AZs

Amazon EBS

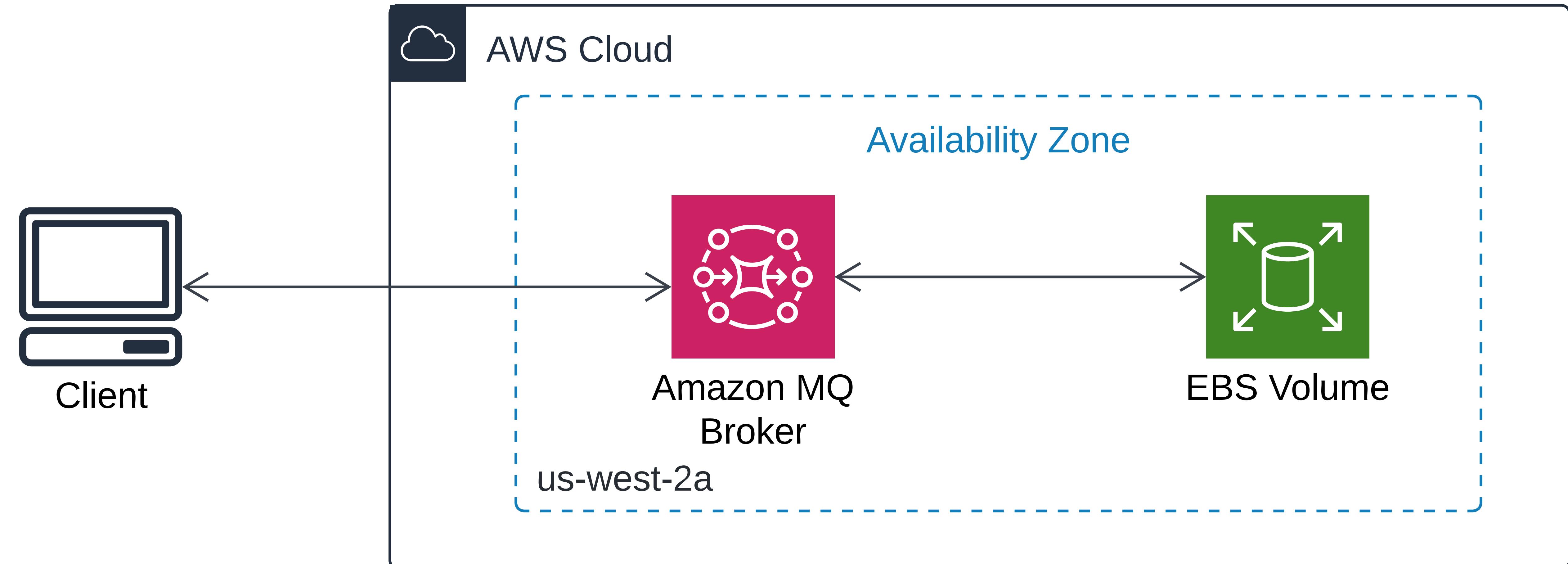
块级存储

⚡ Low latency

🚀 High throughput



Amazon MQ single-instance broker - with EBS



Amazon MQ active/standby broker for high availability

1. Architecture

Two brokers in different Availability Zones

Redundant pair configuration

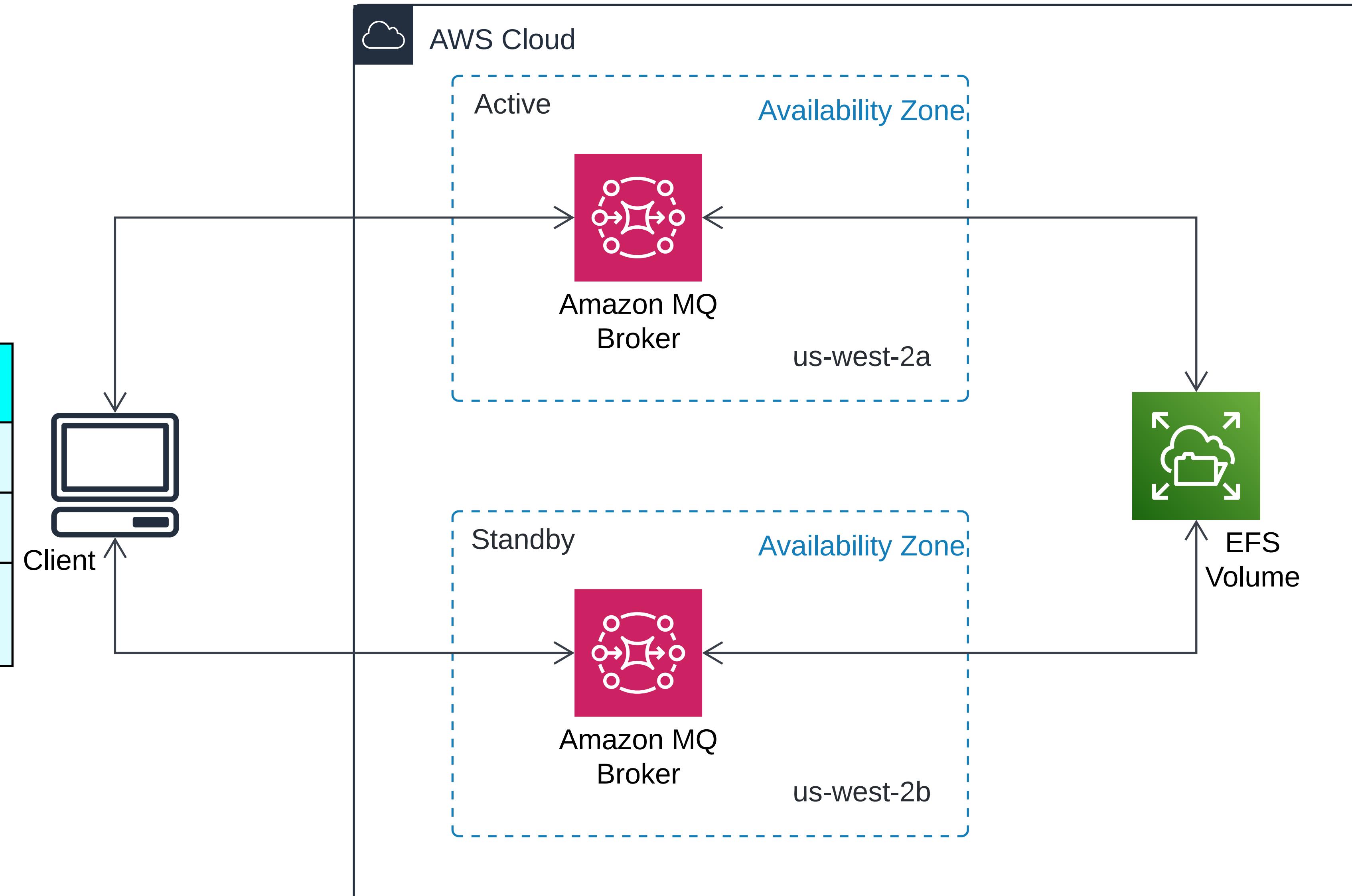
Synchronous communication

- With application
- With Amazon EFS

2. Storage

Amazon EFS volumes

- High durability
- High availability
- Data redundancy across multiple AZs



Amazon MQ active/standby broker for high availability

3. ⚙ Failover Process

💡 One active, one standby broker

🔄 Automatic switch on malfunction/maintenance

⌚ Short transition time

🕒 Quick reboot (few seconds)

5. 🔍 Wire-level Protocol Endpoints

🔢 Two endpoints per protocol

⚡ One active in each pair

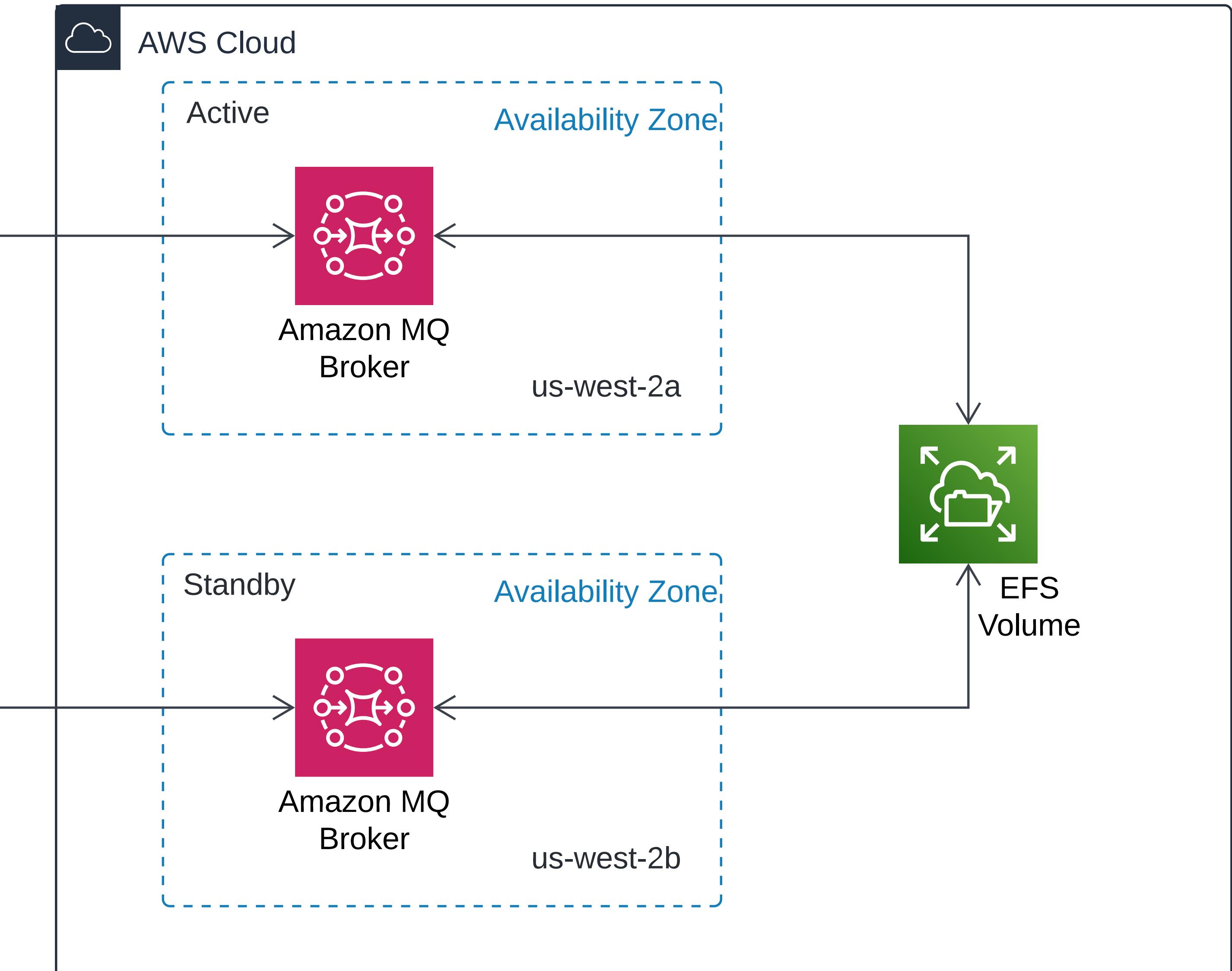
🔤 -1 and -2 suffixes for redundant pairs

🚗 Failover Transport for connection flexibility

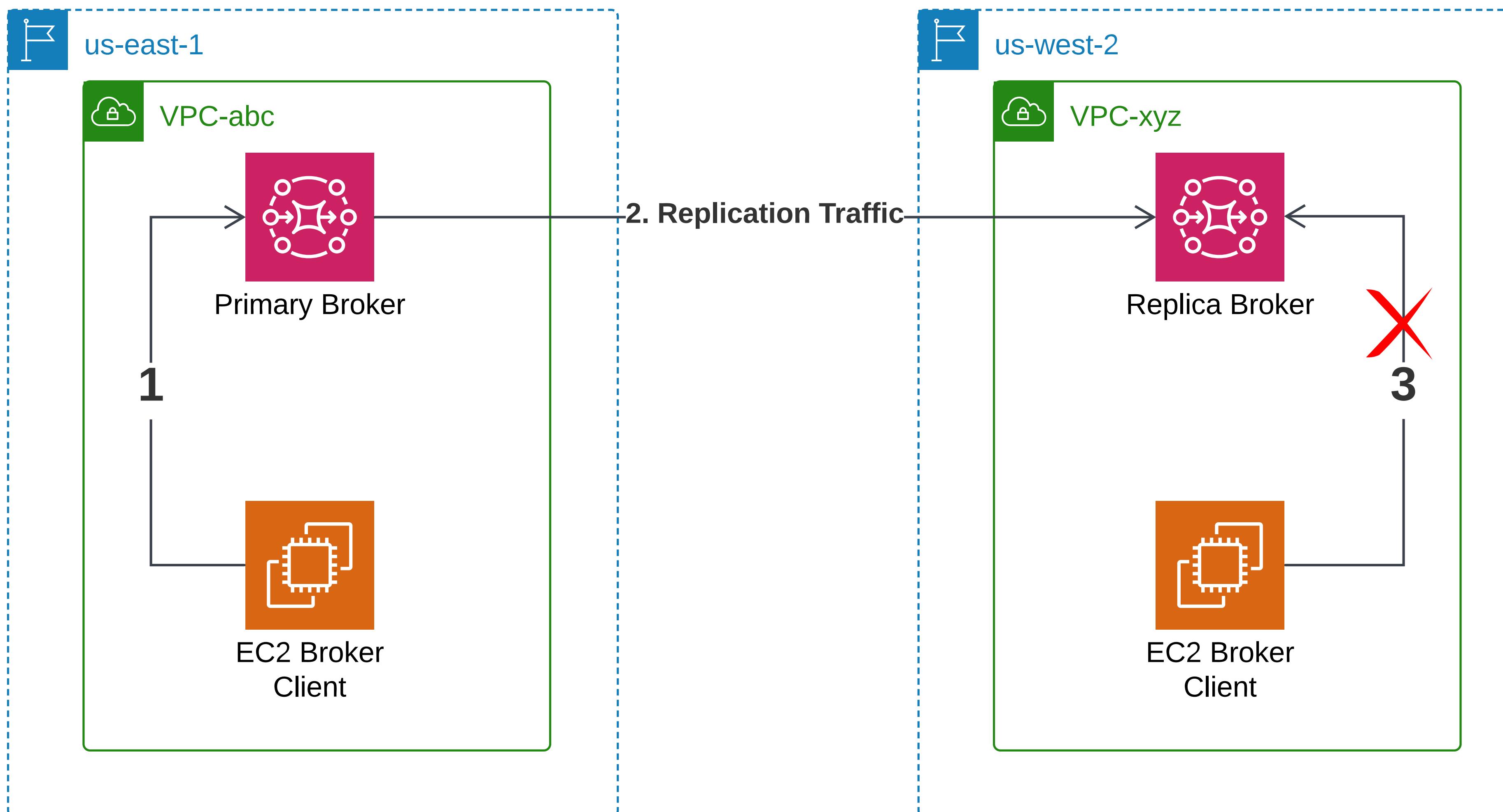
4. 🖥 ActiveMQ Web Console

🔗 Two URLs provided

🟢 Only one active at a time

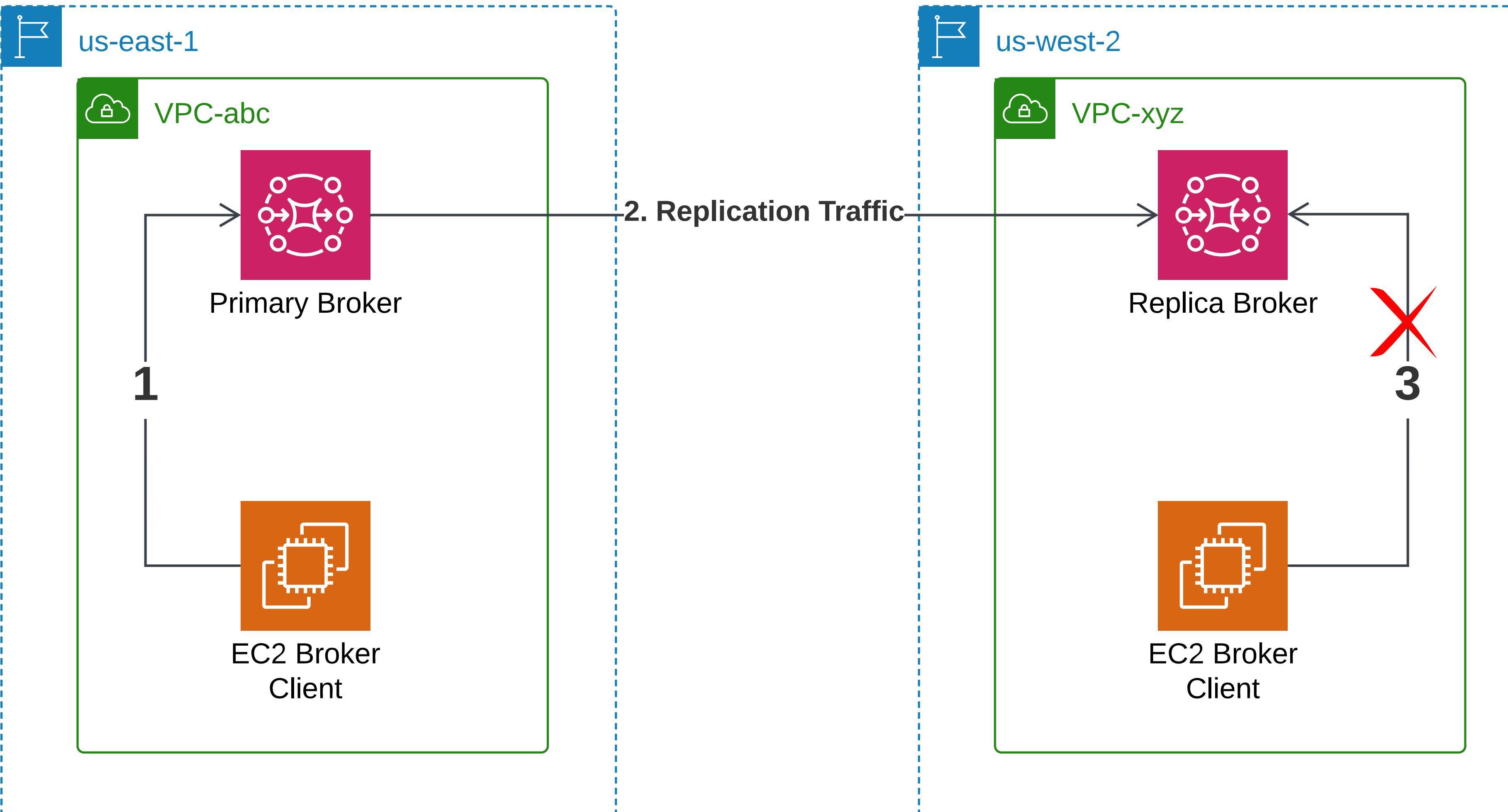


Cross-Region data replication for Amazon MQ for ActiveMQ



- | |
|----------------------------------|
| 1. CRDR Feature |
| Asynchronous message replication |
| Primary Region to Replica Region |
| 2. Failover Process |
| Amazon MQ API request |
| Promote replica to primary |
| Demote primary to replica |

Cross-Region data replication for Amazon MQ for ActiveMQ



3. 🏗️ Architecture	
● Primary Region	🔄 Redundant pair: active/standby
● Replica Region	🔄 Redundant pair: active/standby
4. 🛡️ Disaster Recovery	
✗ Primary broker failure	
☒ Switchover or failover	
⟳ Role reversal	⬆️ Replica promoted to primary ⬇️ Primary demoted to replica

RabbitMQ



Amazon MQ for RabbitMQ Broker Types



Amazon MQ for RabbitMQ offers two types of broker deployments:

Single-Instance Broker



1. Comprised of **one broker** in **one Availability Zone**



2. Positioned behind a **Network Load Balancer (NLB)**



3. The broker communicates with:

- Your application



- An Amazon EBS storage volume



Cluster Deployment



1. A logical grouping of **three RabbitMQ broker nodes**



2. Positioned behind a **Network Load Balancer**



3. Distributed across **multiple Availability Zones (AZ)**



4. Nodes share:

- Users



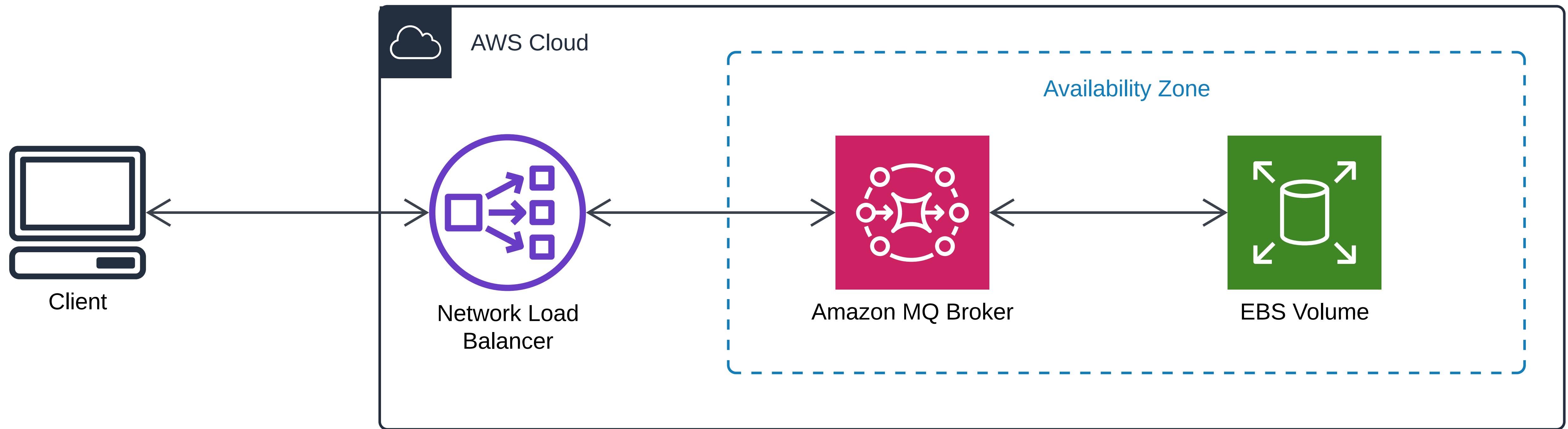
- Queues



- Distributed state



Single-instance broker - RabbitMQ



1. Architecture
1 One broker in one Availability Zone
2 Behind Network Load Balancer (NLB)
Communicates with application

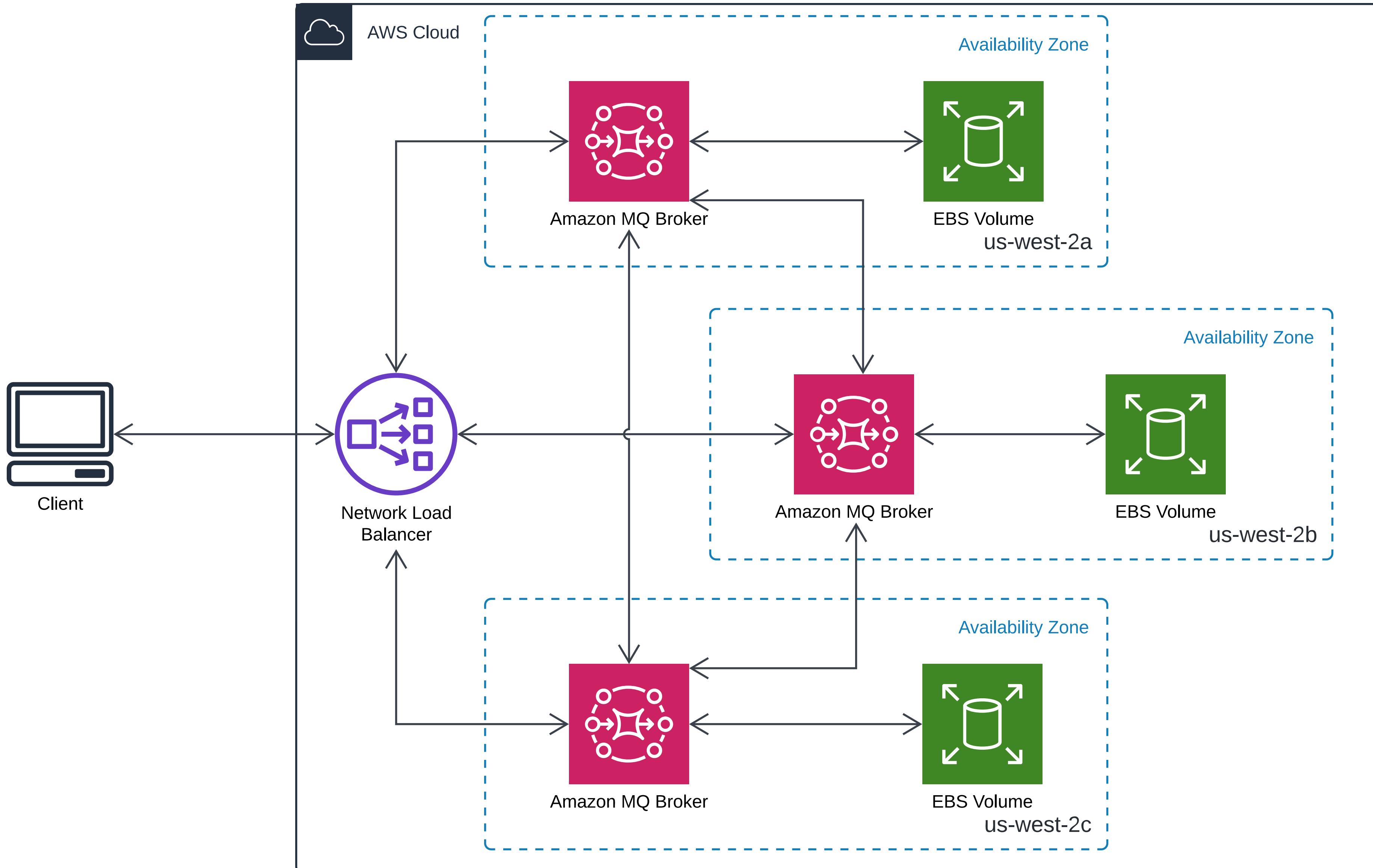
2. Storage
Amazon EBS volume
Block level storage
Low-latency
High throughput

3. Network Load Balancer benefits	
Unchanging broker endpoint	During maintenance
EC2 hardware failures	For applications
Continuous connectivity	For users

Cluster deployment for high availability

1. Architecture

- 3 Three RabbitMQ broker nodes
- Behind Network Load Balancer
- Across multiple Availability Zones
- Shared users, queues, distributed state



Cluster deployment for high availability

2. High Availability (HA) Setup

Automatic management of broker policies

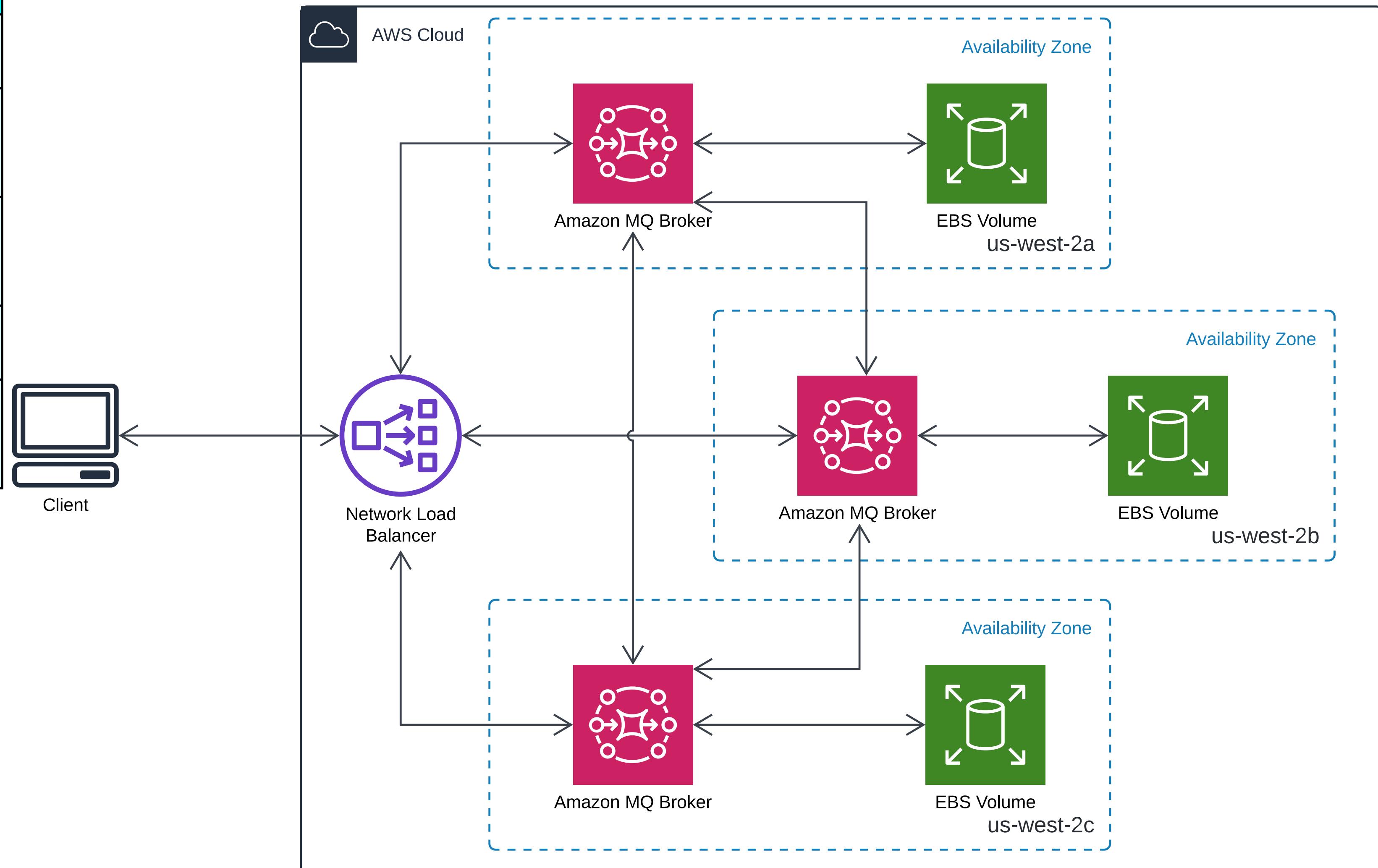
1 One main node per queue

Operations applied to main, then propagated

Default system policy

ha-mode: all

ha-sync-mode: automatic



Cluster deployment for high availability

3. Maintenance Process

1 One node at a time

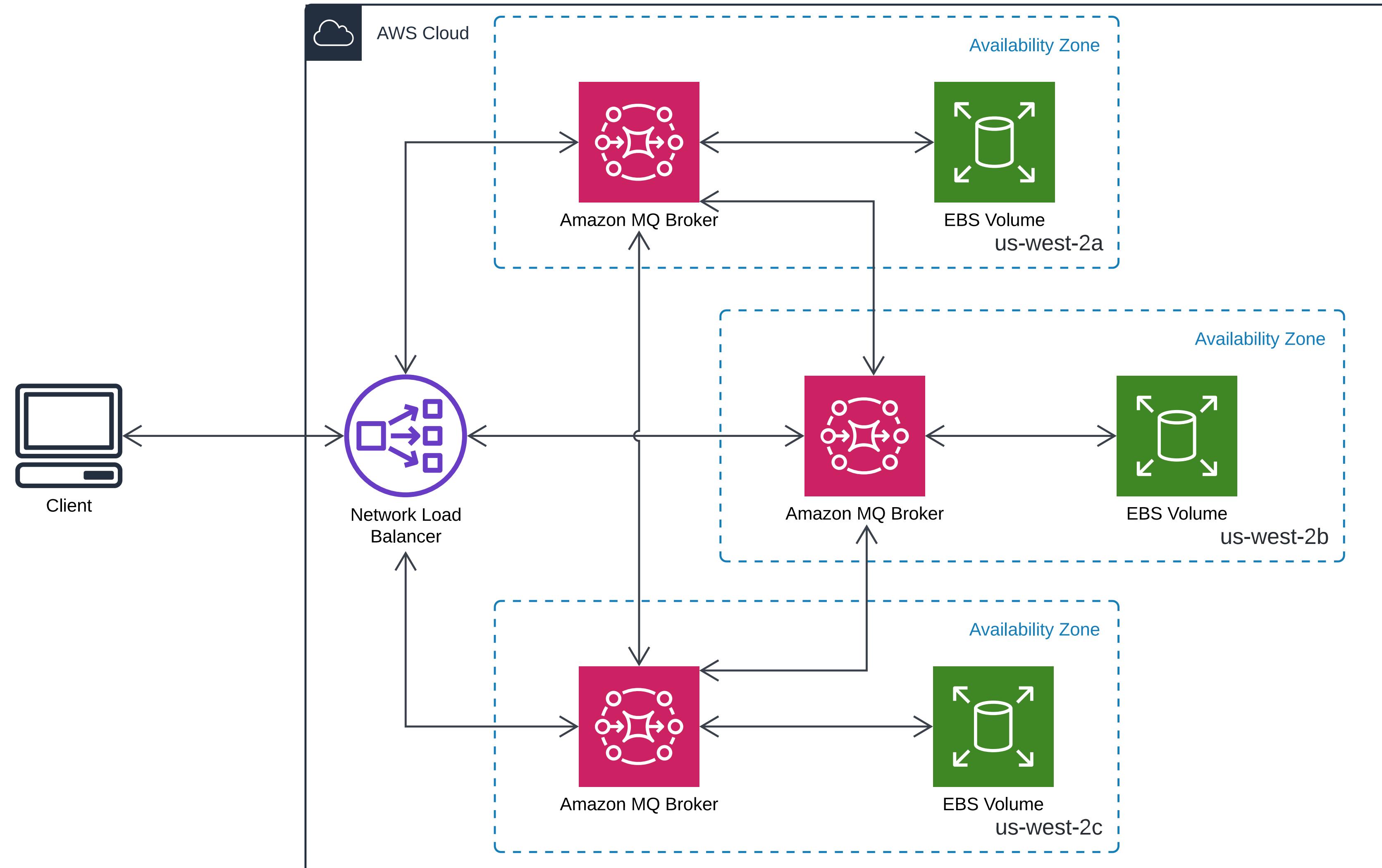
2 Minimum two running nodes always

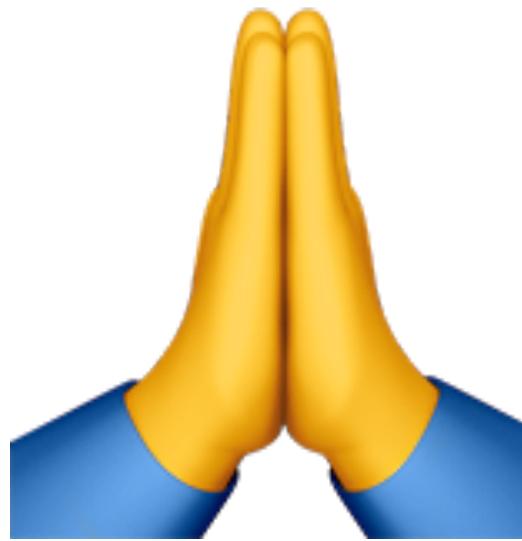
⚡ Client connections severed, need reconnection

⟳ Queue synchronization
on node rejoin

🚫 Blocks other queue
operations

💡 Mitigation: Keep
queues short





**Thanks
for
Watching**