Red Hat A-MQ 7.0 A-MQ

Introduction to A-MQ 7.0

Red Hat Customer Content Services

Red Hat A-MQ 7.0 A-MQ

Introduction to A-MQ 7.0

Legal Notice

Abstract

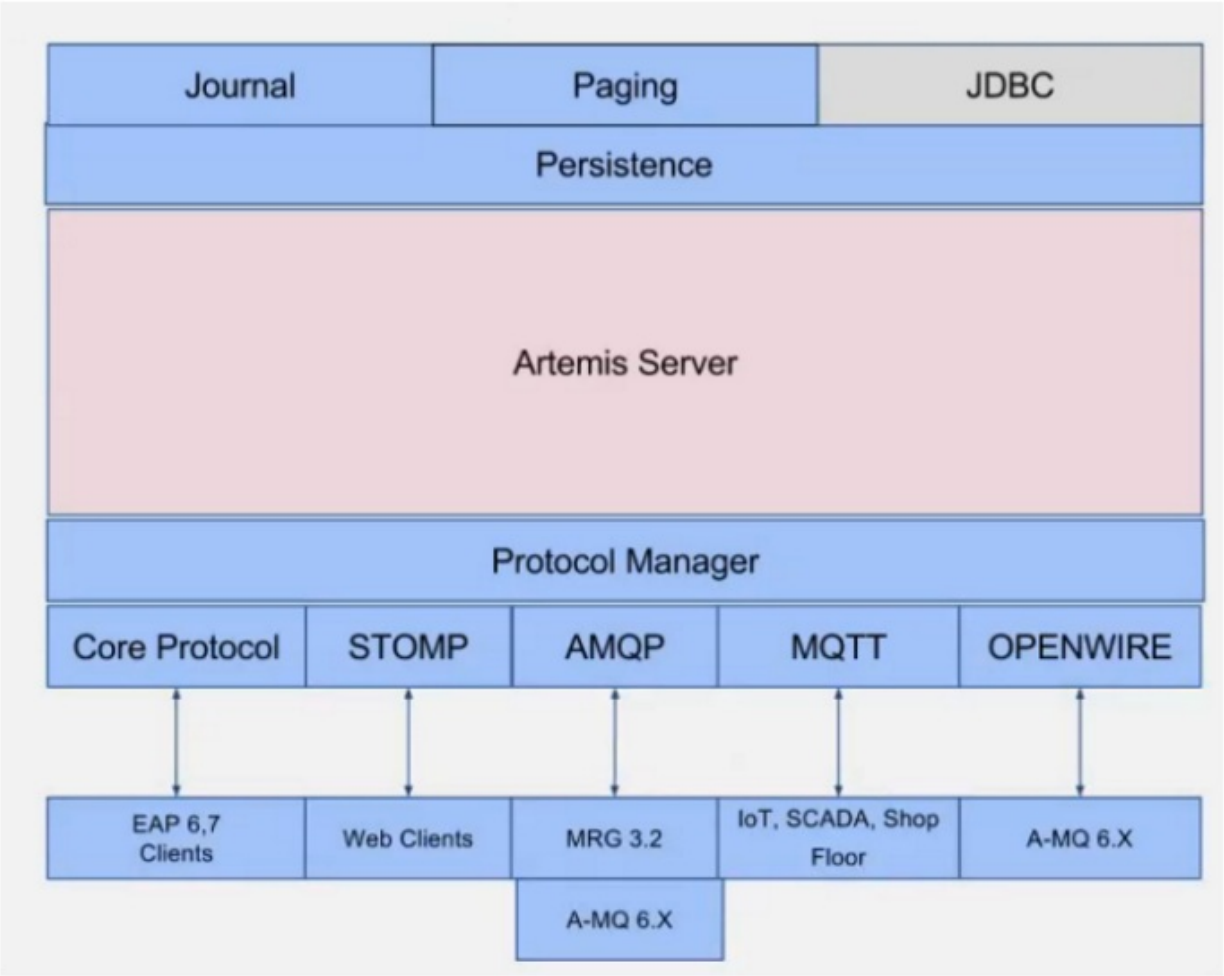
This guide provides introduction to A-MQ.

Table of Contents

CHAPTER 1. INTRODUCTION TO A-MQ 7.0

1.1. WHAT IS A-MQ 7.0?

A-MQ 7.0 is a JMS 2.0 compliant, multi-protocol, embeddable, high performance, clustered, asynchronous messaging system. It consists of a broker, router and client-side libraries that enable remote communication among distributed client applications. A-MQ 7.0 provides numerous connectivity options and can communicate with a wide variety of non-JMS clients through its support of the AMQP, MQTT, OpenWire, HortnQ Core, Artemis Core and STOMP wire protocols. The architectural overview of A-MQ 7 is shown below.



Note

A-MQ 7.0 does not support JDBC in the Alpha release.

1.1.1. A-MQ 7.0 Broker

A-MQ 7.0 Broker handles the exchange of messages between messaging clients. It does so by managing the transport connections used for communication with messaging clients, coordinating with other brokers, managing the database for persistent messages, monitoring and managing various components of the messaging system, and so on. The broker is the heart of a messaging system.

A-MQ 7.0 Broker is based on Apache ActiveMQ Artemis, which is an open source project for an asynchronous messaging system.

1.1.2. A-MQ 7.0 Client

A-MQ 7.0 client is a suite of AMQP 1.0 APIs embeddable in an application that can connect to a broker over one or more wire protocols.

Client applications send or receive messages. Message producers create and send messages. Message consumers receive and process them. JMS clients use the JMS API to interact with the broker. Non-JMS clients use any of A-MQ's other client APIs to interact with the broker.

1.1.3. A-MQ 7.0 Interconnect

A-MQ 7.0 Interconnect, based on Apache Qpid Dispatch Router, is a technology that provides a number of benefits to messaging systems that use the AMQP protocol.

A-MQ 7.0 Interconnect can be deployed as a single instance or in networks of interconnected routers with flexible topology. The router network provides an interconnect layer between messaging-based applications or clients and message brokers. The router network forwards messages to their destinations based on the high-level semantics of the AMQP protocol.

1.1.4. Messages

Messages are the means by which client applications transmit business data and events. Messages can contain either textual or binary payloads. They also contain metadata, which provides additional information about the message. Applications can use the metadata programmatically to modify or fine tune message delivery or administratively to monitor the health of the messaging system.

Messaging is loosely coupled exchange of messages between applications. It is location transparent and can be reliable or unreliable.

1.1.5. Wire Protocols

Wire protocols describe the data format between a client and broker. A-MQ 7.0 supports the following wire protocols.

- AMQP
- MQTT
- STOMP
- OpenWire
- HornetQ Core
- Artemis Core

The following table lists the protocol compatibility matrix.

Table 1.1. A-MQ 7.0 protocol compatibility matrix (Messaging APIs)

COMPONENT	LANGUAGES	PLATFORMS	PROTOCOLS
ActiveMQ CMS	C++	Linux, Windows, Solaris	OpenWire, STOMP
ActiveMQ JMS	Java, Scala	JVM	OpenWire
ActiveMQ NMS	C#, VB.NET	.NET	OpenWire, AMQP 1.0, MQTT, STOMP
Qpid JMS	Java, Scala	JVM	AMQP 1.0
Qpid Messaging C++	C++	Linux	AMQP 1.0, AMQP 0-10
Proton Python	Python	Linux, Windows	AMQP 1.0
Proton C++	C++	Linux, Windows	AMQP 1.0
Rhea	JavaScript	Linux, Windows	AMQP 1.0

Table 1.2. A-MQ 7.0 protocol compatibility matrix (Servers)

COMPONENT	PLATFORMS	PROTOCOLS
ActiveMQ 5 broker	JVM	AMQP 1.0, OpenWire, MQTT, STOMP
Artemis broker	JVM	AMQP 1.0, OpenWire, Core, MQTT, STOMP
Dispatch router	Linux	AMQP 1.0

1.2. WHY USE A-MQ 7.0?

The benefits of using A-MQ 7.0 are:

- Open source software. A-MQ 7.0 is licensed using the Apache Software License v2.0 to minimize barriers to adoption.
- Written in Java. Runs on any platform with a Java 8 or higher runtime.
- Full feature set with unique features.
- Minimal third party dependencies. Run A-MQ 7.0 in either stand-alone mode, integrated in a JEE application server or embedded inside your own product.
- Seamless High Availability (HA). A-MQ 7.0's HA solution features automatic client failover to guarantee zero message loss or duplication in the event of server failure.
- High-performance journal provides persistent messaging performance at non-persistent rates. A-MQ 7.0 handles its own persistence, rather than relying on a database or other third party persistence engine.
- Flexible clustering. Create clusters of servers that can load-balance messages. Link distributed clusters over unreliable connections to form a global network. Configure routing of messages in a highly flexible way.
- Multi-protocol. Supports AMQP, MQTT, STOMP, OpenWire, HornetQ Core and Artemis Core.
- Embeddable. Can be embedded in JBoss EAP 7.0, Fuse 7.0 or in your own applications.